

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інформаційних технологій та комп'ютерної інженерії
(назва факультету (відділення))
Кафедра програмного забезпечення
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка програмних засобів для веб-сервісу рекомендацій по торгівлі
криптовалютою з використанням оптимізованих гібридних моделей нейронних
мереж»

Виконав: студент II курсу, групи 2ПІ-23м
спеціальності 121 – Інженерія програмного забезпечення
(шифр і назва напрямку підготовки, спеціальності)

Керівник: Пасічнюк В. А.
(прізвище та ініціали)
К.Т.Н., доц. каф. ПЗ Коваленко О. О.
(прізвище та ініціали)
«09» грудня 2024 р.

Опонент: К.Т.Н., проф. каф. ОТ Захарченко С. М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
«09» грудня 2024 р.

Допущено до захисту

Завідувач кафедри ПЗ

Д.Т.Н., проф., Романюк О. Н.
«09» грудня 2024 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти II-й (магістерський)
Галузь знань 12 – Інформаційні технології
Спеціальність 121 – Інженерія програмного забезпечення
Освітньо-професійна програма – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
_____ Романюк О. Н.
«18» вересня 2024 р.

З А В Д А Н Н Я НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Пасічнюку Владиславу Андрійовичу

1. Тема роботи: Розробка програмних засобів для веб-сервісу рекомендацій по торгівлі криптовалютою з використанням оптимізованих гібридних моделей нейронних мереж.

Керівник роботи: Коваленко Олена Олексіївна, доц. каф. ПЗ, затверджені наказом вищого навчального закладу від «17» вересня 2024 р. № 310.

2. Строк подання студентом роботи 3 грудня 2024 р.

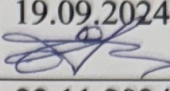
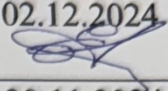
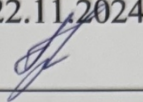
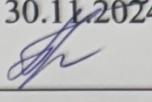
3. Вихідні дані до роботи: актуальні дані криптовалют з CoinMarketCap API; історичні дані криптовалют з AlphaVantage API; конфігурація оптимізованої гібридної моделі нейронної мережі; мова програмування Java; фреймворки та бібліотеки: Spring Framework, Thymeleaf, Deeplearning4j, Gson, Google Charts; розмір екрану 2560-1600px, кольоровий режим, тип графіку – лінійний горизонтальний.

4. Зміст розрахунково-пояснювальної записки: аналіз предметної галузі веб-сервісів рекомендацій по торгівлі криптовалютою; порівняльний аналіз аналогів; аналіз способів реалізації функціоналу прогнозування; аналіз типів нейронних мереж; постановка задач розробки; конфігурування гібридної моделі нейронної мережі; оптимізація моделі нейромережі; метод створення функціоналу прогнозування цін криптовалют; розробка архітектури програмного модуля; розробка алгоритмів функціонування системи; розробка програми для веб-сервісу рекомендацій по торгівлі криптовалютою; тестування програмного продукту; порівняльний аналіз точності прогнозування; економічна частина.

5. Перелік ілюстративного матеріалу: знімки екрану головних сторінок сайтів CoinMarketCap, CoinGecko, Blockonomi; схема порівняння RNN та LSTM моделей;

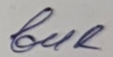
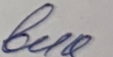
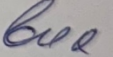
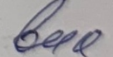
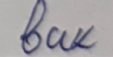
схема візуалізації роботи CNN мереж; діаграми класів, компонентів, діяльності та варіантів використання програми; блок-схеми алгоритмів роботи програми; схеми інтерфейсу веб-сервісу; знімки екрану середовищ розробки; фрагменти логічних структур; знімки екрану процесу взаємодії з розробленим веб-сервісом.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Коваленко О.О., к.т.н., доц. кафедри ПЗ	19.09.2024 	02.12.2024 
5	Причепя І.В., к.е.н., доц. кафедри ЕПВМ	22.11.2024 	30.11.2024 

7. Дата видачі завдання 19 вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів	Примітка
1	Аналіз стану веб-сервісів рекомендацій по торгівлі криптовалютою	20.09.2024 – 10.10.2024	
2	Методи конфігурування нейронної мережі та створення веб-сервісу	10.10.2024 – 21.10.2024	
3	Розробка програми для веб-сервісу рекомендацій по торгівлі криптовалютою	21.10.2024 – 15.11.2024	
4	Тестування програмного продукту	15.11.2024 – 22.11.2024	
5	Економічна частина	22.11.2024 – 30.11.2024	

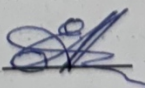
Студент


(підпис)

Пасічник В. А.

(прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи


(підпис)

Коваленко О. О.

(прізвище та ініціали)

АНОТАЦІЯ

УДК 336.744

Пасічнюк В. А. Розробка програмних засобів для веб-сервісу рекомендацій по торгівлі криптовалютою з використанням гібридних оптимізованих моделей нейронних мереж. Магістерська кваліфікаційна робота зі спеціальності 121 – інженерія програмного забезпечення, освітня програма – інженерія програмного забезпечення. Вінниця: ВНТУ, 2024. 149 с.

На укр. мові. Бібліогр.: 30 назв; рис.: 52; табл. 10.

У магістерській кваліфікаційній роботі проведено детальний аналіз предметної галузі веб-сервісів рекомендацій по торгівлі криптовалютою, здійснено порівняльний аналіз аналогів, розглянуто способи реалізації функціоналу прогнозування та проаналізовано різні типи моделей нейронних мереж. Таким чином, доведено доцільність розробки програмних засобів для веб-сервісу рекомендацій по торгівлі криптовалютою з використанням гібридних оптимізованих моделей нейронних мереж та сформовано перелік задач для розробки.

Було описано процес конфігурування гібридної моделі нейромережі та процес оптимізації значень вхідних параметрів. Також описано принципи роботи веб-сервісу та побудовано схеми для головних алгоритмів програми, діаграми класів, компонентів, діяльності та використання. Крім того, описано розробку структури інтерфейсу.

Обґрунтовано вибір мови програмування та середовища розробки. Детально описано етапи розробки програмного модуля та графічного інтерфейсу. Також проведено декілька етапів тестування програми та здійснено порівняльний аналіз точності прогнозування.

Для виявлення фінансової доцільності розробки, було виконано економічний аналіз, що передбачає розрахунки витрат та визначення суми потенційного прибутку.

Ключові слова: криптовалюта, торгівля, прогнозування, нейронна мережа, дані

ABSTRACT

Pasichniuk V. Development of software tools for the web service of cryptocurrency trading recommendations using optimized hybrid neural network models. Master's thesis in specialty 121 – software engineering, educational program – software engineering. Vinnytsia: VNTU, 2024. 149 p.

In Ukrainian. Bibliography: 30 titles; Figures: 52; Table 10.

In the master's qualification work, a detailed analysis of the subject area of web services of cryptocurrency trading recommendations was carried out, a comparative analysis of analogues was carried out, ways to implement the forecasting functionality were considered, and various types of neural network models were analyzed. Thus, the expediency of developing software tools for a web service of cryptocurrency trading recommendations using hybrid optimized neural network models is proved and a list of tasks for development is formed.

The process of configuring a hybrid neural network model and the process of optimizing the values of input parameters are described. The principles of the web service are also described, and diagrams for the main program algorithms, class, components, activities, and usage diagrams are constructed. In addition, the development of the interface structure is described.

The choice of programming language and development environment is substantiated. The stages of development of the program module and graphical interface are described in detail. Also, several stages of testing the program were carried out and a comparative analysis of forecasting accuracy was carried out.

In order to determine the financial feasibility of the development, an economic analysis was carried out, which involves calculating costs and determining the amount of potential profit.

Keywords: cryptocurrency, trading, forecasting, neural network, data

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ СТАНУ ВЕБ-СЕРВІСІВ РЕКОМЕНДАЦІЙ ПО ТОРГІВЛІ КРИПТОВАЛЮТОЮ	8
1.1 Аналіз предметної галузі	8
1.2 Порівняльний аналіз аналогів веб-сервісів рекомендацій по торгівлі криптовалютою	9
1.3 Аналіз способів реалізації функціоналу прогнозування.....	13
1.4 Аналіз типів моделей нейронних мереж	14
1.5 Постановка задач розробки.....	19
1.6 Висновки.....	20
2 МЕТОДИ КОНФІГУРУВАННЯ НЕЙРОННОЇ МЕРЕЖІ ТА СТВОРЕННЯ ВЕБ-СЕРВІСУ	21
2.1 Конфігурування гібридної моделі нейронної мережі	21
2.2 Оптимізація моделі нейромережі.....	23
2.3 Метод створення функціоналу прогнозування цін криптовалют	26
2.4 Розробка архітектури програмного модуля веб-сервісу рекомендацій по торгівлі криптовалютою	28
2.5 Розробка алгоритмів функціонування системи	34
2.6 Розробка структури інтерфейсу	37
2.7 Висновки.....	41
3 РОЗРОБКА ПРОГРАМИ ДЛЯ ВЕБ-СЕРВІСУ РЕКОМЕНДАЦІЙ ПО ТОРГІВЛІ КРИПТОВАЛЮТОЮ.....	43
3.1 Варіантний аналіз і обґрунтування вибору інструментів та технологій розробки.....	43

3.1.1	Варіантний аналіз і обґрунтування вибору мови програмування.....	43
3.1.2	Варіантний аналіз і обґрунтування вибору середовища розробки ...	47
3.2	Розробка програмного модуля	51
3.3	Розробка інтерфейсу користувача.....	55
3.4	Висновки.....	56
4	ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	58
4.1	Аналіз методів тестування	58
4.2	Тестування програми.....	61
4.3	Порівняльний аналіз точності прогнозування	73
4.4	Опис основних функціональних можливостей користувача.....	75
4.5	Висновки.....	79
5	ЕКОНОМІЧНА ЧАСТИНА	80
5.1	Оцінювання комерційного потенціалу розробки	80
5.2	Прогнозування витрат на виконання науково-дослідної роботи.....	87
5.3	Оцінка економічної ефективності науково-дослідної роботи.....	93
5.4	Розрахунок ефективності вкладених інвестицій та періоду їх окупності	94
5.5	Висновки.....	96
	ВИСНОВКИ	98
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	100
	ДОДАТКИ	103
	ДОДАТОК А. Технічне завдання.....	104
	ДОДАТОК Б. Протокол перевірки	108
	ДОДАТОК В. Лістинг коду	109
	ДОДАТОК Г. Ілюстративна частина	137

ВСТУП

Обґрунтування вибору теми дослідження. Криптовалюти стали невід'ємною частиною сучасного економічного простору, справляючи суттєвий вплив на фінансові системи по всьому світу. Вони дозволяють користувачам здійснювати операції з цифровими активами без необхідності залучення банків, що забезпечує швидкі та надійні міжнародні транзакції. Крім того, криптовалюти відкривають нові горизонти для інвесторів та підприємців, надаючи можливість збільшити свій капітал і залучати інвестиції зі всього світу.

Прогнозування ціни криптовалют є ключовим завданням для інвесторів, трейдерів та експертів, які прагнуть досягти успіху на цьому динамічному та швидкозмінному ринку. Зі зростанням обсягів торгів і ускладненням ринкових умов цей процес стає дедалі складнішим, вимагаючи застосування передових алгоритмів та методів машинного навчання для більш точних прогнозів.

Машинне навчання є однією з провідних технологій, що використовується для передбачення цін криптовалют [1]. Воно аналізує історичні дані і на основі виявлених тенденцій дає змогу прогнозувати майбутні цінові рухи. Це допомагає трейдерам та інвесторам приймати обґрунтовані рішення щодо купівлі або продажу активів, мінімізуючи ризики.

Серед методів машинного навчання особливо ефективними є нейронні мережі [2]. Вони імітують роботу людського мозку, що дозволяє їм навчатися на основі великих масивів даних і виявляти складні зв'язки та закономірності. Завдяки цьому нейронні мережі можуть обробляти величезні обсяги інформації, знаходити приховані залежності та здійснювати більш точні прогнози майбутніх змін на ринку.

Однією з головних переваг використання нейронних мереж для прогнозування цін криптовалют є їх здатність адаптовуватися до мінливих умов ринку. Це означає, що вони можуть своєчасно реагувати на нові тренди та коригувати свої прогнози відповідно до змін. Таким чином, інвестори отримують більш точні прогнози, що дозволяє знижувати ризики при ухваленні інвестиційних рішень.

Втім, слід пам'ятати, що криптовалюти відзначаються надзвичайною волатильністю, і їх ціни можуть різко змінюватися під впливом політичних та економічних факторів, новин у криптовалютній індустрії, а також технічних змін у блокчейн-протоколах. Тому, незважаючи на те, що машинне навчання та нейронні мережі мають великий потенціал для прогнозування цін, вони є лише одним з інструментів для прийняття рішень і не можуть гарантувати стовідсоткову точність.

На сьогоднішній день існує не так багато веб-сервісів, які надають доступ до перегляду актуальних та історичних даних криптовалют. Проте, жоден із них не має функціоналу прогнозування майбутніх цін.

Тому розробка власного веб-сервісу рекомендацій по торгівлі криптовалютою з використанням оптимізованих гібридних моделей нейронних мереж є актуальною та матиме перевагу над конкурентами на ринку.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

Мета і завдання дослідження. Метою дослідження є покращення процесів купівлі та продажу криптовалют за рахунок використання оптимізованих гібридних моделей нейронних мереж у програмному модулі, який здатен прогнозувати ціни криптовалют, базуючись на історичних даних зміни цін.

Відповідно до поставленої мети, в магістерській кваліфікаційній роботі потрібно вирішити такі задачі:

- проаналізувати стан веб-сервісів рекомендацій по торгівлі криптовалютою;
- визначити найбільш ефективний підхід для прогнозування цін криптовалют за визначеними критеріями;
- розробити методи створення функціоналу прогнозування цін криптовалют;
- розробити алгоритм отримання та процесингу історичних даних зміни цін криптовалют;
- створити алгоритм прогнозування ціни криптовалюти;
- провести конфігурування гібридної моделі нейронної мережі;

- здійснити оптимізацію вхідних параметрів в конфігурації гібридної моделі нейронної мережі;
- розробити інтерфейс користувача;
- розробити програмний модуль веб-сервісу рекомендацій по торгівлі криптовалютою з використанням оптимізованих гібридних моделей нейронних мереж;
- провести варіантний аналіз засобів та середовищ розробки;
- проаналізувати методи тестування програмного забезпечення;
- протестувати програмний модуль для виявлення потенційних проблем;
- провести порівняльний аналіз точності прогнозування.

Об'єкт дослідження – процес відслідковування актуальних та історичних даних криптовалют і прогнозування цін з використанням оптимізованих гібридних моделей нейронних мереж.

Предметом дослідження є алгоритми та засоби розробки програмних модулів для веб-сервісу рекомендацій по торгівлі криптовалютою, що використовує нейромережу.

Методи дослідження. У процесі дослідження застосовувалися: метод аналізу для формулювання аналітичних висновків за історичними даними, теорія нейромереж для прогнозування цін криптовалют, теорія чисел, методи об'єктно-орієнтованого програмування для проектування програмних модулів, комп'ютерне моделювання для аналізу та перевірки точності отриманих прогнозованих даних.

Новизна отриманих результатів:

1. Удосконалено метод створення функціоналу прогнозування цін криптовалют шляхом розробки підходу, який, на відміну від існуючих, передбачає використання нейронної мережі та візуалізацію результатів у форматі лінійного горизонтального графіка, що забезпечило зручний спосіб отримання орієнтовних майбутніх цін криптовалют, підвищуючи доступність і наочність прогнозів.

2. Покращено метод конфігурації моделей гібридних нейронних мереж, що вирізняється інтеграцією рекурентних (LSTM) і згорткових (CNN) шарів із використанням шарів активації (ReLU) та глобального усереднення. Додатково

оптимізовано вхідні параметри моделі, що дозволило ефективно застосовувати її для прогнозування фінансових показників.

Практична цінність отриманих результатів. Практична цінність одержаних результатів полягає в тому, що на основі отриманих в магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби для прогнозування цін криптовалют з використанням оптимізованих гібридних моделей нейронних мереж.

Особистий внесок здобувача. Усі наукові результати, викладені у магістерській кваліфікаційній роботі, отримані автором особисто. У друкованих працях, опублікованих у співавторстві, автору належать такі результати: [3] – аналіз бібліотек машинного навчання для мови Java, [4] – основні типи нейронних мереж, [5] – аналіз типів нейронних мереж для задачі прогнозування фінансових показників.

Апробація результатів роботи. Результати роботи доповідалися та обговорювалися на міжнародних та всеукраїнських конференціях: XVII Міжнародна науково-практична конференція “Інформаційні технології і автоматизація – 2024” (2024), XXIII Всеукраїнська науково-технічна конференція молодих вчених, аспірантів та студентів (2023), конференція “Молодь в науці: дослідження, проблеми, перспективи” (2023).

Публікації. Основні результати досліджень опубліковано в трьох наукових працях у збірниках матеріалів конференцій [3] [4] [5].

Структура та обсяг. Магістерська кваліфікаційна робота складається зі вступу, п'ятих розділів, висновків, списку літератури, що містить 30 найменувань, та 4 додатки. Робота містить 52 ілюстрації, 10 таблиць. Загальний обсяг роботи складає 149 сторінок, основний зміст викладено на 97 сторінках друкованого тексту.

1 АНАЛІЗ СТАНУ ВЕБ-СЕРВІСІВ РЕКОМЕНДАЦІЙ ПО ТОРГІВЛІ КРИПТОВАЛЮТОЮ

1.1 Аналіз предметної галузі

Сайти, присвячені тематиці криптовалютної торгівлі, є важливим джерелом даних для інвесторів і трейдерів, які працюють з цифровими активами. Функціональність таких платформ може варіюватися в залежності від особливостей кожного сайту, але в цілому, основними функціями є відображення актуальної вартості криптовалют, динаміки їх змін протягом дня, тижня, місяця, а також надання інтерактивних графіків.

Більшість платформ також пропонують розширені інструменти для аналізу ринку. Окрім базових даних, таких як поточна ціна криптовалют, користувачі можуть отримати доступ до інформації про ринкову капіталізацію, добові обсяги торгів, кількість монет в обігу та інші важливі метрики. Ці показники допомагають трейдерам оцінювати ринкову позицію активів та їхній потенціал [6].

Незважаючи на надання великої кількості аналітичних даних, більшість таких ресурсів уникають давати конкретні рекомендації щодо купівлі або продажу криптовалют. Основний акцент робиться на наданні користувачам інструментів і даних, таких як графіки, індикатори технічного аналізу, які дозволяють приймати обґрунтовані рішення самостійно. Деякі платформи також надають інсайти щодо поточних ринкових трендів або криптовалют, які мають потенціал для зростання, що може бути корисним для інвесторів, які шукають нові можливості для прибуткових вкладень.

Крім того, такі сайти можуть пропонувати корисні поради щодо вибору інструментів та торгових стратегій. Вони можуть включати в себе технічні індикатори, такі як індекси відносної сили або інші інструменти, які допомагають краще зрозуміти ринкові умови та передбачити потенційні рухи цін.

Подібні ресурси є незамінними для трейдерів та інвесторів, які прагнуть бути в курсі коливань цін на ринку криптовалют, щоб ефективно планувати свої інвестиційні

стратегії. Вони допомагають швидко реагувати на ринкові зміни і використовувати інформацію для ухвалення більш точних рішень.

Однак важливо пам'ятати, що будь-які інвестиційні рішення, пов'язані з криптовалютами, повинні базуватися на власному досвіді, знаннях та глибокому розумінні ринку. Торгівля криптовалютами залишається високоризиковою діяльністю через значну волатильність і непередбачуваність цього ринку. Тому рекомендується завжди дотримуватися обережності та використовувати надійні стратегії управління ризиками [7].

1.2 Порівняльний аналіз аналогів веб-сервісів рекомендацій по торгівлі криптовалютою

Існує безліч веб-сервісів, які пропонують рекомендації щодо торгівлі криптовалютою. Серед найпопулярніших представників варто виділити такі платформи, як:

- CoinMarketCap;
- CoinGecko;
- Blockonomi.

Порівняємо ці сервіси та розглянемо їхні сильні й слабкі сторони, а також зіставимо їх із власною розробкою.

CoinMarketCap (рис. 1.1) – один із найбільш відомих ресурсів у криптовалютній спільноті, який був запущений у 2013 році. Він швидко завоював популярність і нині є головним джерелом для отримання актуальних даних про ринок криптовалют. Платформа дозволяє користувачам відстежувати ціни, ринкову капіталізацію, обсяги торгів та інші важливі метрики для тисяч криптовалют [8].

Перевагами CoinMarketCap є наявність свіжих і докладних даних про криптовалютний ринок. Веб-сайт також пропонує інтерактивні віджети, що дозволяють користувачам швидко переглядати інформацію про найпопулярніші криптовалюти та нові активи, що з'являються на ринку. Однак, одним із недоліків сервісу є відсутність функціоналу для прогнозування цін, що обмежує можливості

користувачів у плануванні торговельних рішень. Крім того, інтерфейс може здаватися перевантаженим через велику кількість інформації, яку не завжди враховують у прийнятті рішень.

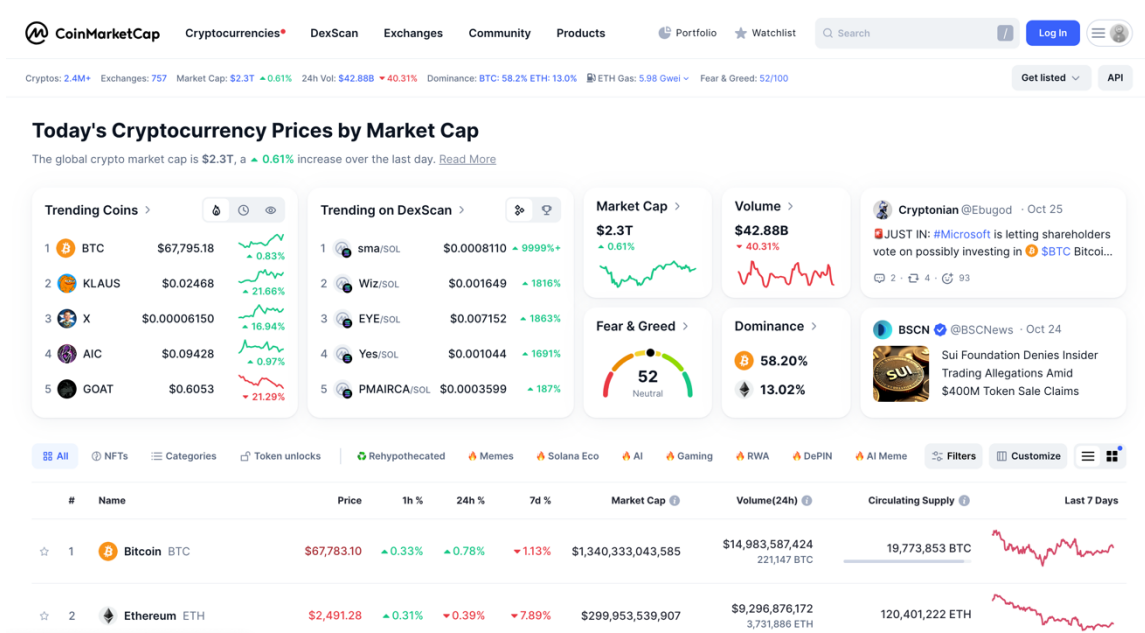


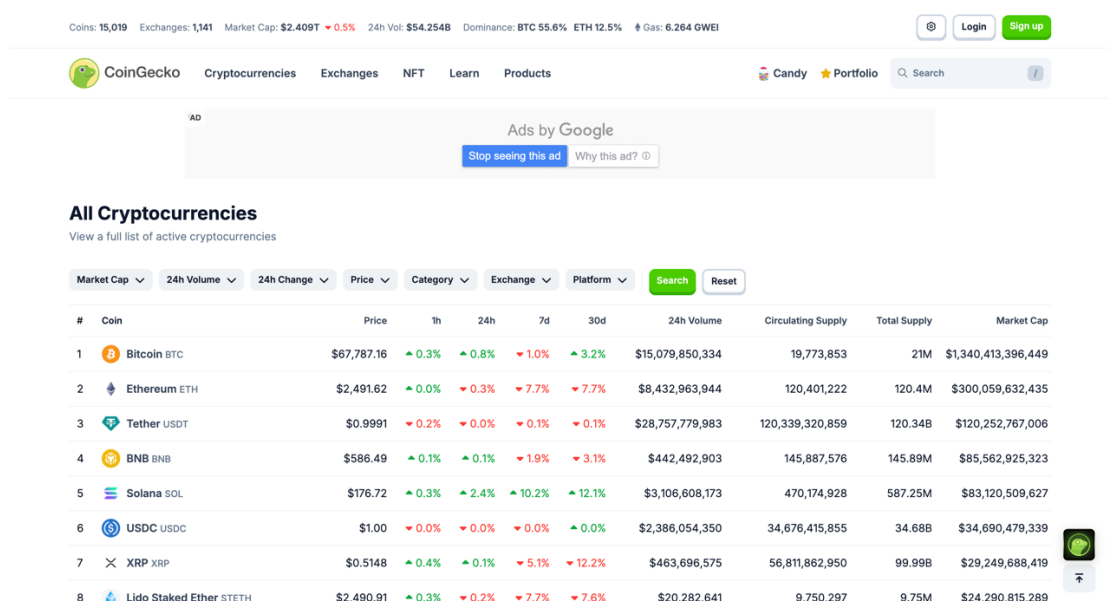
Рисунок 1.1 – Головна сторінка сайту CoinMarketCap

CoinGecko (рис. 1.2) – це ще один популярний сервіс, який відображає актуальні дані про ринок криптовалют. Він активно працює в цій сфері кілька років і виділяється тим, що має найбільшу команду розробників серед конкурентів, що дозволяє йому постійно вдосконалюватися. CoinGecko пропонує широкий набір аналітичних інструментів і може похвалитися швидким оновленням інформації [9].

Перевагами CoinGecko є простий і зрозумілий інтерфейс, який не перевантажує користувачів непотрібною інформацією, а також забезпечує своєчасне надання даних про ціни криптовалют. Однак, як і в CoinMarketCap, тут відсутня функція прогнозування цін, що може бути важливою для тих, хто шукає додаткові інструменти для аналізу ринку та прийняття рішень щодо інвестицій.

Bloktopi (рис. 1.3) також є ресурсом, який надає інформацію про криптовалютний ринок. Веб-сайт пропонує досить простий і зручний інтерфейс, що може сподобатися початківцям у крипторгівлі. Окрім стандартних ринкових даних,

одна з вкладок сайту містить навчальні матеріали, які можуть бути корисними для новачків, що тільки починають свій шлях у криптовалютному просторі [10].

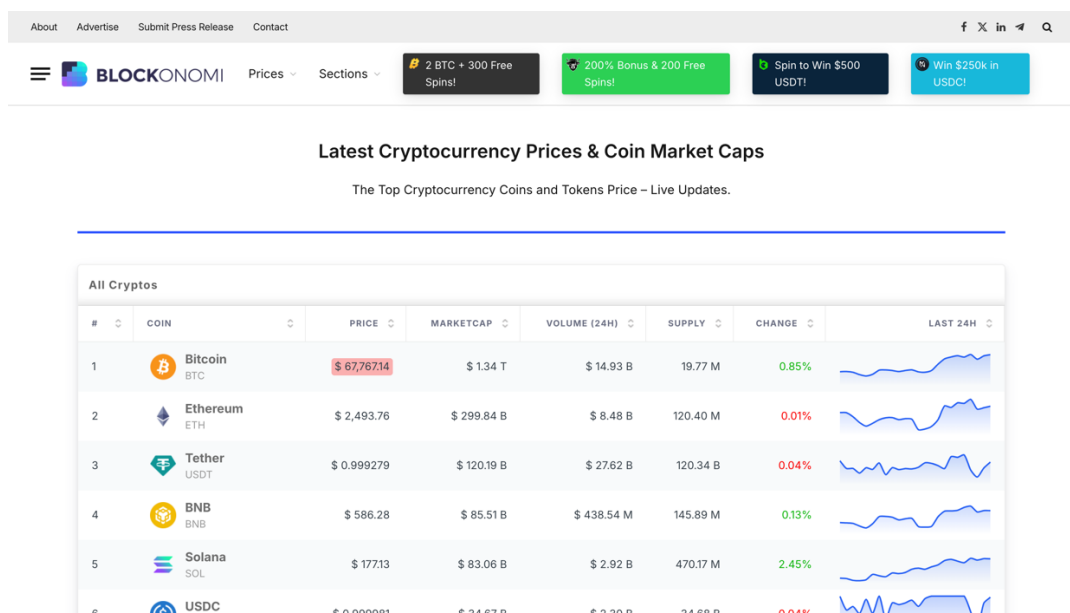


The screenshot shows the CoinGecko homepage. At the top, there's a navigation bar with links for Coins, Exchanges, NFT, Learn, and Products. Below this is a section titled 'All Cryptocurrencies' with a table of the top 8 cryptocurrencies. The table includes columns for Rank, Coin, Price, 1h, 24h, 7d, 30d, 24h Volume, Circulating Supply, Total Supply, and Market Cap.

#	Coin	Price	1h	24h	7d	30d	24h Volume	Circulating Supply	Total Supply	Market Cap
1	Bitcoin BTC	\$67,787.16	▲ 0.3%	▲ 0.8%	▼ 1.0%	▲ 3.2%	\$15,079,850,334	19,773,853	21M	\$1,340,413,396,449
2	Ethereum ETH	\$2,491.62	▲ 0.0%	▼ 0.3%	▼ 7.7%	▼ 7.7%	\$8,432,963,944	120,401,222	120.4M	\$300,059,632,435
3	Tether USDT	\$0.9991	▼ 0.2%	▼ 0.0%	▼ 0.1%	▼ 0.1%	\$28,757,779,983	120,339,320,859	120.34B	\$120,252,767,006
4	BNB BNB	\$586.49	▲ 0.1%	▲ 0.1%	▼ 1.9%	▼ 3.1%	\$442,492,903	145,887,576	145.89M	\$85,562,925,323
5	Solana SOL	\$176.72	▲ 0.3%	▲ 2.4%	▲ 10.2%	▲ 12.1%	\$3,106,608,173	470,174,928	587.25M	\$83,120,509,627
6	USDC USDC	\$1.00	▼ 0.0%	▼ 0.0%	▼ 0.0%	▲ 0.0%	\$2,386,054,350	34,676,415,855	34.68B	\$34,690,479,339
7	XRP XRP	\$0.5148	▲ 0.4%	▲ 0.1%	▼ 5.1%	▼ 12.2%	\$463,696,575	56,811,862,950	99.99B	\$29,249,688,419
8	Lido Staked Ether STETH	\$2,490.91	▲ 0.3%	▼ 0.2%	▼ 7.7%	▼ 7.6%	\$20,282,641	9,750,297	9.75M	\$24,290,815,289

Рисунок 1.2 – Головна сторінка сайту CoinGecko

Проте, одним із основних недоліків Bloconomi є те, що надані дані часто відстають від більш сучасних платформ, таких як CoinMarketCap чи CoinGecko. Це може ввести користувачів в оману, якщо вони спиратимуться на застарілу інформацію при прийнятті рішень.



The screenshot shows the Bloconomi homepage. At the top, there's a navigation bar with links for About, Advertise, Submit Press Release, and Contact. Below this is a section titled 'Latest Cryptocurrency Prices & Coin Market Caps' with a table of the top 6 cryptocurrencies. The table includes columns for Rank, Coin, Price, MarketCap, Volume (24h), Supply, Change, and Last 24h.

#	COIN	PRICE	MARKETCAP	VOLUME (24H)	SUPPLY	CHANGE	LAST 24H
1	Bitcoin BTC	\$ 67,767.14	\$ 1.34 T	\$ 14.93 B	19.77 M	0.85%	
2	Ethereum ETH	\$ 2,493.76	\$ 299.84 B	\$ 8.48 B	120.40 M	0.01%	
3	Tether USDT	\$ 0.999279	\$ 120.19 B	\$ 27.62 B	120.34 B	0.04%	
4	BNB BNB	\$ 586.28	\$ 85.51 B	\$ 438.54 M	145.89 M	0.13%	
5	Solana SOL	\$ 177.13	\$ 83.06 B	\$ 2.92 B	470.17 M	2.45%	
6	USDC	\$ 0.999981	\$ 34.67 B	\$ 2.30 B	34.68 B	0.04%	

Рисунок 1.3 – Головна сторінка сайту Bloconomi

Крім того, як і у випадку з іншими згаданими платформами, на Blockonomi також відсутні прогнози цін і віджети для аналізу динаміки ринку в реальному часі.

На відміну від згаданих платформ, розроблюваний веб-сервіс “CryptoAdvisor” пропонує користувачам не лише актуальні ринкові дані, але й інтегровані аналітичні інструменти для прогнозування цін криптовалют. Це дозволяє трейдерам та інвесторам отримувати не лише поточну інформацію, але й передбачати можливі тенденції на ринку, що є ключовим елементом для ефективної торгівлі. Крім того, інтерфейс додатку спеціально розроблений для того, щоб бути максимально зручним та інтуїтивно зрозумілим, мінімізуючи надлишок інформації, але зберігаючи всі необхідні інструменти для аналізу ринку.

Отже, проведемо порівняльний аналіз описаних засобів та розробленого веб-сервісу "CryptoAdvisor" (таблиця 1.1).

Таблиця 1.1 – Порівняльний аналіз аналогів

Характеристика / Засіб	CoinMarket Cap	CoinGecko	Blockonomi	Crypto Advisor
Надання даних про ціни криптовалют у реальному часі	+	+	-	+
Наявність віджетів з аналітичною інформацією	+	-	-	+
Можливість перегляду прогнозованих цін криптовалют	-	-	-	+
Наявність навчального контенту	+	-	+	-
Простота взаємодії з інтерфейсом	-	+	+	+
Сума	3	2	2	4

Отже, на основі проведеного аналізу зазначених програмних продуктів можна стверджувати, що розроблений веб-сервіс має більше переваг порівняно з розглянутими аналогами.

1.3 Аналіз способів реалізації функціоналу прогнозування

Для створення функціоналу прогнозування криптовалют існує декілька варіантів, кожен з яких використовує нейронні мережі. Найгнучкішим і найдетальнішим підходом є написання власної нейронної мережі з нуля. Це рішення дозволить врахувати всі особливості й нюанси модуля прогнозування, що створюється для специфічної задачі. Однак, розробка нейромережі з нуля є складним процесом, що вимагає професійних знань у галузі програмування, математики, та статистики, а також глибокого розуміння принципів роботи нейронних мереж.

Нейронні мережі – це складні математичні моделі, які складаються з великої кількості "нейронів", що взаємодіють один з одним через вагові коефіцієнти. Завданням розробника є підбір оптимальних значень цих коефіцієнтів, що досягається у процесі навчання нейромережі. Процес навчання є ресурсоємним і може зайняти значний час, особливо якщо тренування моделі проводиться на великих обсягах даних. Крім того, потрібно вирішувати такі завдання, як вибір архітектури нейромережі, підбір алгоритмів оптимізації, контроль за навчанням та оцінка результатів. Хоча цей підхід забезпечує максимальну гнучкість і можливість контролювати кожен етап, він також є надзвичайно трудомістким і може бути неефективним з точки зору часу та ресурсів.

Іншим підходом є використання готових рішень, таких як сервіси машинного навчання AWS SageMaker [11] або Azure Machine Learning [12]. Ці платформи надають потужні інструменти для побудови нейромереж без необхідності розробляти їх з нуля. Використання таких сервісів дозволяє розробникам зосередитись на створенні потрібного модуля прогнозування, а не на технічних деталях побудови нейромереж. Важливою перевагою цих рішень є можливість швидкого старту, оскільки більшість таких сервісів пропонують попередньо налаштовані моделі, які

можна використовувати для конкретних завдань без написання власного коду. Проте, використання таких платформ має свої недоліки, зокрема високу вартість та залежність від обраного провайдера, що може призвести до додаткових витрат і обмежень при виборі налаштувань або гнучкості моделей.

Найбільш збалансованим рішенням є використання бібліотек машинного навчання, таких як `deeplearning4j` [13], `TensorFlow` або `Keras`, які дозволяють створювати власні нейромережі на базі вже існуючих архітектур. У цьому випадку розробник може обрати найбільш відповідний тип нейромережі та налаштувати її параметри для виконання конкретного завдання, такого як прогнозування фінансових показників. Цей підхід позбавляє необхідності створювати мережу з нуля, одночасно надаючи достатньо гнучкості для налаштування моделей під власні потреби. Бібліотеки машинного навчання зазвичай містять безліч готових компонентів, що прискорює процес розробки та дозволяє створити ефективну модель за коротший час.

Однак, використання таких бібліотек також вимагає певних технічних знань і навичок у налаштуванні нейромереж, а також вміння створювати програми на мові `Java` (у випадку `deeplearning4j`) чи `Python` (для інших згаданих бібліотек). Крім цього, процес навчання моделі може зайняти чимало часу залежно від обсягу даних та складності завдання.

Проаналізувавши переваги та недоліки кожного з цих підходів, було вирішено використати метод розробки модуля прогнозування ціни криптовалюти на основі бібліотеки машинного навчання `deeplearning4j`. Цей варіант надає гнучкість у налаштуванні нейромережі, дозволяючи уникнути зайвих витрат часу на розробку з нуля, одночасно забезпечуючи достатню функціональність для створення ефективного та точного прогнозу.

1.4 Аналіз типів моделей нейронних мереж

Нейронні мережі є основою сучасного машинного навчання, забезпечуючи широкий спектр можливостей для аналізу та прогнозування. Різні типи мереж розроблені для вирішення специфічних завдань, таких як аналіз часових рядів або

виявлення складних закономірностей у даних. У контексті прогнозування фінансових показників, зокрема цін криптовалют, особливу увагу приділяють таким типам, як звичайні нейронні мережі, рекурентні нейронні мережі, мережі довготривалої короткочасної пам'яті, згорткові нейронні мережі та гібридні моделі.

Звичайні нейронні мережі (Feedforward Neural Networks) працюють за принципом передачі даних від вхідного до вихідного шару без зворотного зв'язку [14]. Цей процес складається з декількох етапів:

- Вхідні дані представляються у вигляді векторів, де кожен елемент відображає певну характеристику об'єкта.
- Кожен вектор передається через приховані шари мережі. У прихованих шарах обчислюється лінійна комбінація вхідних сигналів, після чого застосовується активаційна функція для додання нелінійності.
- Результати обробки прихованих шарів передаються до вихідного шару, де формується підсумковий прогноз.
- Під час навчання мережі вагові коефіцієнти оптимізуються за допомогою методу зворотного поширення помилки, що дозволяє мінімізувати різницю між передбаченнями та реальними значеннями.

Цей тип мереж ефективний у задачах класифікації або регресії, проте він має обмежене застосування у задачах прогнозування часових рядів через неможливість враховувати часову залежність між даними.

Рекурентні нейронні мережі (RNN) усувають це обмеження, дозволяючи зберігати інформацію про попередні стани через механізм зворотного зв'язку. Їх механізм роботи виглядає наступним чином:

- Дані подаються до мережі у вигляді послідовності, де кожен елемент відповідає певному моменту часу.
- На кожному кроці обчислюється поточний стан, який є функцією вхідних даних і стану з попереднього кроку.
- Поточний стан використовується для формування прогнозу для відповідного моменту часу.

- Для оптимізації вагових коефіцієнтів використовується алгоритм зворотного поширення через час (BPTT), що дозволяє враховувати всі послідовності даних.

Хоча RNN добре працюють із послідовними даними, вони стикаються з проблемою зникнення градієнтів, що ускладнює навчання на довгих послідовностях.

Мережі довготривалої короткочасної пам'яті (LSTM) вирішують цю проблему завдяки унікальній структурі, яка дозволяє зберігати важливу інформацію протягом тривалих проміжків часу (рис. 1.4).

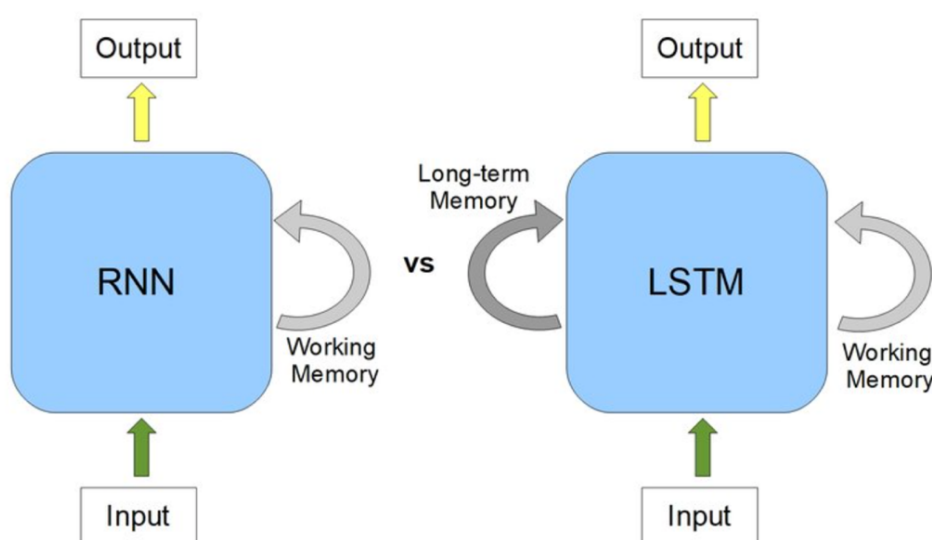


Рисунок 1.4 – Порівняння RNN та LSTM

Можна виділити наступні етапи роботи LSTM мереж:

- Вхідні дані передаються до мережі у форматі послідовності векторів.
- Кожен вектор обробляється через чотири основних шари: забувальний шар, вхідний шар, шар пам'яті та вихідний шар.
- Забувальний шар визначає, яку частину інформації з попереднього стану слід видалити, використовуючи функцію активації сигмоїд.
- Вхідний шар визначає, яку нову інформацію потрібно додати до поточного стану, шляхом поєднання входу і стану попереднього часу.

- Шар пам'яті оновлює довготривалий стан, інтегруючи збережену інформацію та поточні зміни.
- Вихідний шар генерує підсумковий сигнал, який використовується для формування прогнозу.

Ця архітектура дозволяє LSTM працювати з довгими часовими рядами, враховуючи як короткострокові зміни, так і довготривалі тенденції [15]. Це робить їх надійним інструментом для фінансових прогнозів.

Згорткові нейронні мережі (CNN) часто застосовуються для виявлення локальних закономірностей у даних, таких як патерни у часових рядах. Їхній процес роботи включає наступні етапи:

- Дані передаються через згорткові шари, кожен з яких застосовує фільтри для виявлення локальних закономірностей.
- Результати згортки проходять через шар підвибірки (pooling), що зменшує розмірність, зберігаючи ключову інформацію.
- Після цього дані передаються до повнозв'язних шарів, які інтегрують виявлені закономірності для формування підсумкового прогнозу.

У контексті фінансових даних, CNN можуть бути корисними для виділення особливих закономірностей у цінових тенденціях, які можуть бути приховані в великих обсягах даних [16]. На рисунку 1.5 наведено візуалізацію роботи CNN мережі.

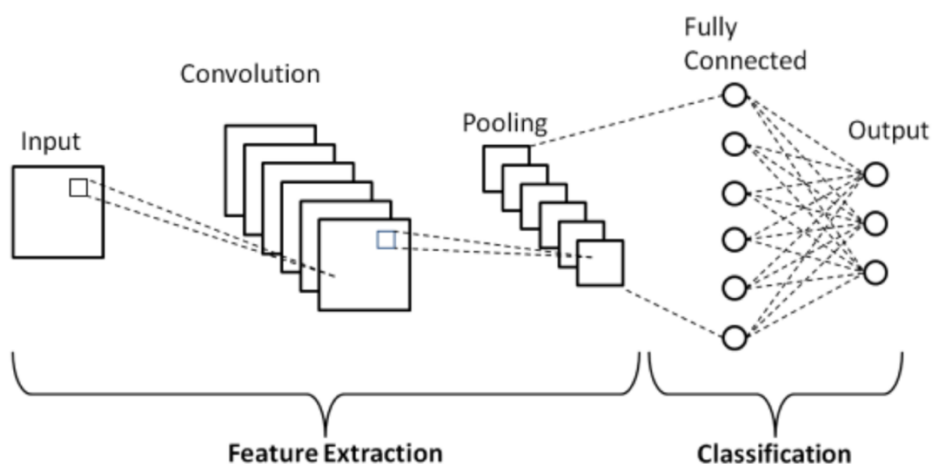


Рисунок 1.5 – Візуалізація роботи CNN мереж

Гібридні моделі, які поєднують CNN та LSTM, забезпечують ефективну інтеграцію локальних закономірностей і часових залежностей. У цих моделях CNN використовуються для виявлення локальних патернів у даних, після чого результати передаються на вхід LSTM, яка аналізує часові взаємозв'язки між виявленими особливостями. Такий підхід є надзвичайно ефективним для прогнозування фінансових показників, оскільки ціни залежать не лише від історичних даних, але й від прихованих шаблонів у поведінці ринку.

Хоча нейронні мережі є потужним інструментом для прогнозування, вони також мають низку обмежень, які можуть ускладнювати їхнє використання у реальних умовах. По-перше, для ефективного навчання моделей потрібен великий обсяг якісних даних. У випадку криптовалют, дані часто містять значний рівень шуму через непередбачувані коливання на ринку, що спричиняються різними факторами, такими як новини, регуляторні зміни або навіть психологія трейдерів. Такі умови підвищують ризик перенавчання моделі, коли вона починає запам'ятовувати шум замість виявлення загальних закономірностей.

По-друге, налаштування нейронної мережі є складним і багатоетапним завданням. Воно потребує ретельного вибору гіперпараметрів, таких як кількість шарів і нейронів у кожному шарі, типи оптимізаторів, функції активації, а також параметри регуляризації для запобігання перенавчанню. Наприклад, вибір занадто малої кількості шарів може обмежити здатність моделі до вивчення складних патернів, тоді як занадто велика кількість може призвести до надмірної складності, що знижує її узагальнюючі можливості. Крім того, налаштування швидкості навчання є критично важливим: занадто високі значення можуть спричинити нестабільність процесу оптимізації, а надто низькі – надмірно тривалий процес тренування.

Тренування нейромереж вимагає значних обчислювальних ресурсів, особливо при роботі з великими наборами даних або складними архітектурами, такими як гібридні моделі. Навіть із сучасними технологіями час тренування може варіюватися від декількох годин до кількох днів, залежно від розміру моделі та кількості ітерацій. Крім того, на етапі розгортання моделей виникають виклики, пов'язані з необхідністю

забезпечення швидкого виконання прогнозів у режимі реального часу, що потребує оптимізації архітектури для роботи на менш потужних пристроях.

Незважаючи на ці виклики, нейронні мережі, зокрема їх гібридні моделі, демонструють високу ефективність у вирішенні задач прогнозування фінансових показників, таких як ціни криптовалют. Поєднання CNN і LSTM дозволяє створити модель, яка не лише виділяє локальні патерни, але й враховує часову динаміку змін. CNN забезпечує обробку вхідних даних для виявлення ключових патернів, таких як короткострокові коливання або різкі зміни цін, які можуть свідчити про важливі події на ринку. Водночас LSTM зберігає інформацію про довгострокові тенденції, інтегруючи отримані локальні патерни у цілісну картину.

Проте, використання цих методів у практичних умовах вимагає певної підготовки. Гібридні моделі потребують узгодженості між частинами: параметри між різними типами шарів повинні бути налаштовані таким чином, щоб ефективно працювати разом. Крім того, результати моделі потребують інтерпретації, оскільки складність архітектури може ускладнювати пояснення її рішень для кінцевих користувачів.

Підсумовуючи, нейронні мережі забезпечують високий потенціал у прогнозуванні фінансових показників, але їхнє впровадження супроводжується різноманітними викликами, що включають в себе налаштування параметрів, врахування обчислювальних витрат, підготовку тренувальних даних та інтеграцію моделі в існуючі бізнес-процеси.

1.5 Постановка задач розробки

Після аналізу переваг і недоліків існуючих веб-сервісів, що мають функціонал надання рекомендацій щодо торгівлі криптовалютою, а також дослідження підходів до використання нейронних мереж для задач прогнозування, було визначено перелік функціональних вимог для розроблюваного програмного модуля:

- можливість перегляду актуальної інформації про криптовалюту;
- можливість перегляду історичних даних про ціни криптовалют;

- мати функціонал відображення віджетів з корисними статистичними даними;
- мати функціонал прогнозування цін криптовалют, що здійснюється за допомогою гібридної нейромережі;
- можливість зв'язатися з розробниками та надати зворотній зв'язок.

Також перед створюваним програмним забезпеченням було виставлено наступні нефункціональні вимоги:

- інтуїтивність користувацького інтерфейсу;
- легкий доступ до інформації;
- відсутність довгих затримок при виконанні запитів;
- відсутність помилок під час взаємодії;
- відображення прогнозованих цін у форматі лінійного графіка.

1.6 Висновки

У першому розділі було проведено аналіз сучасного стану веб-сервісів рекомендацій щодо торгівлі криптовалютами. Було порівняно відомі платформи, такі як CoinMarketCap, CoinGecko та Blockonomi. В результаті аналізу стало очевидним, що ці веб-сайти здобули велику популярність серед користувачів завдяки своїй зручності та широкому спектру функцій. Проте, було виявлено значний недолік – відсутність інструментів для прогнозування цін криптовалют. Це свідчить про наявність незаповненої ніші на ринку, що підтверджує доцільність розробки власного програмного рішення з функціоналом прогнозування. На основі цього аналізу було визначено перелік задач, необхідних для створення нового веб-сервісу, який би включав прогнозування цін криптовалют.

2 МЕТОДИ КОНФІГУРУВАННЯ НЕЙРОННОЇ МЕРЕЖІ ТА СТВОРЕННЯ ВЕБ-SERVICES

2.1 Конфігурування гібридної моделі нейронної мережі

Процес конфігурування гібридної нейронної мережі є основним етапом, що забезпечує її здатність ефективно розпізнавати складні закономірності у часових даних. Модель складається з поєднання згорткових шарів (CNN) для обробки ознак та рекурентних шарів (LSTM), що спеціалізуються на обробці часових рядів. Далі розглянуто конфігураційні параметри, які було обрано для даної моделі, їх ролі та можливий вплив на ефективність.

1. Початкове значення для генерації випадкових чисел (SEED). Цей параметр гарантує відтворюваність результатів у нейронних мережах, використовуючи фіксоване значення для генерації випадкових чисел. Це важливий елемент стабільності для навчання, що знижує ймовірність випадкових змін у результатах при повторних експериментах.

2. Кількість ітерацій. Цей параметр визначає кількість разів, які модель проходить через навчальні дані в процесі одного циклу навчання. Вибір цього значення залежить від обсягу даних і складності мережі, щоб забезпечити оптимальне оновлення ваг і уникнути перенавчання.

3. Швидкість навчання. Ключовий параметр, що визначає, наскільки швидко змінюються ваги під час оновлення. Баланс між надто високою швидкістю, що може призвести до нестабільного навчання, і надто низькою, що може значно подовжити процес навчання, є вирішальним для успішного навчання моделі.

4. Оптимізаційний алгоритм. Параметр, що визначає алгоритм, який керує процесом оновлення ваг у моделі. Серед популярних варіантів – стохастичний градієнтний спуск (SGD), RMSprop, Adam та інші. Кожен алгоритм має свої особливості та підходить для певних типів задач і структур моделі.

5. Ініціалізація ваг. Ваги на початку навчання ініціалізуються методом, що забезпечує стабільність активаційних значень у мережі. Наприклад, метод “Xavier”

або “KHe Initialization” сприяють стабільному навчальному процесу, запобігаючи надто великим або надто малим значенням активацій, що може зашкодити навчанню.

6. Архітектура моделі. Модель організована з декількох шарів, кожен з яких має свої функції у загальному процесі обробки даних:

- Згорткові шари (Convolutional Layers) обробляють вхідні дані, виділяючи основні ознаки на базі фільтрів. Вибір кількості фільтрів і розмірів ядер має значний вплив на ефективність виділення ознак та подальшу обробку даних.

- Шар глобального усереднення (Global Pooling Layer), що використовується після згорткових шарів, знижує розмірність даних, обчислюючи середнє значення активацій по всіх фільтрах. Цей процес сприяє узагальненню моделі та запобігає перенавчанню.

- Рекурентні шари (LSTM Layers) обробляють часові ряди, зберігаючи послідовну інформацію та виявляючи довгострокові залежності. LSTM дозволяють ефективно аналізувати серії даних, що є особливо важливим для завдань, де зміна з часом відіграє критичну роль.

7. Функції активації (Activation Functions). Вони вносять нелінійність у модель, дозволяючи обробляти складні патерни. Серед популярних функцій – ReLU (Rectified Linear Unit), що найчастіше використовується для прихованих шарів, та "Identity" для вихідного шару, коли потрібно повернути неперетворене значення.

8. Шар регуляризації (Regularization Layer). Для зниження ймовірності перенавчання використовується L2-регуляризація, яка додає штраф до великих значень ваг у мережі, тим самим забезпечуючи стабільність та більш точне узагальнення моделі.

9. Механізм Dropout. Випадково вимикає частину нейронів під час навчання, зменшуючи залежність між окремими нейронами та сприяючи покращенню здатності до узагальнення.

10. Метод усіченого зворотного поширення помилок через час (Truncated BPTT). Обробка довгих часових рядів може бути неефективною, тому використання усіченого BPTT дозволяє враховувати лише частину послідовності, забезпечуючи при цьому ефективну обробку даних та стабільність навчання.

Підсумовуючи, процес конфігурування гібридної моделі нейронної мережі включає ретельний вибір параметрів, що мають безпосередній вплив на точність та стабільність роботи моделі. Поєднання згорткових та рекурентних шарів дозволяє моделі ефективно виділяти та аналізувати часові закономірності, що є ключовими для прогнозування у сфері криптовалют. Різноманітні параметри, включно з ініціалізацією ваг, швидкістю навчання, регуляризацією та структурою шарів, було визначено так, щоб забезпечити гнучкість та збалансованість моделі. Ця конфігурація дозволяє вирішувати завдання прогнозування з урахуванням складних патернів у даних, зберігаючи при цьому продуктивність і стабільність моделі, що є критичними аспектами для подальшого етапу оптимізації.

2.2 Оптимізація моделі нейромережі

На рисунку 2.1 зображено візуалізацію першої частини конфігурації гібридної моделі нейронної мережі, вхідні параметри якої потребують проведення оптимізації.

```
MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
    .seed(SEED)
    .iterations(ITERATIONS)
    .learningRate(LEARNING_RATE)
    .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
    .weightInit(WeightInit.XAVIER)
    .updater(Updater.RMSPROP)
    .regularization(true)
    .l2(1e-4)
    .list()
    .layer(0, new Convolution1DLayer.Builder(1)
        .nIn(nIn)
        .nOut(CNN_LAYER_SIZE)
        .stride(1)
        .activation(Activation.RELU)
        .build())
    .layer(1, new GlobalPoolingLayer.Builder(PoolingType.AVG)
        .build())
    .layer(2, new GravesLSTM.Builder()
        .nIn(CNN_LAYER_SIZE)
        .nOut(LSTM_LAYER_SIZE)
        .activation(Activation.RELU)
        .gateActivationFunction(Activation.HARDSIGMOID)
        .dropOut(DROPOUT_RATIO)
        .build())
```

Рисунок 2.1 – Візуалізація першої частини конфігурації моделі

Другу частину конфігурації наведено на рисунку 2.2.

```

.layer(3, new GravesLSTM.Builder()
    .nIn(LSTM_LAYER_SIZE)
    .nOut(LSTM_LAYER_SIZE)
    .activation(Activation.RELU)
    .gateActivationFunction(Activation.HARDSIGMOID)
    .dropOut(DROPOUT_RATIO)
    .build())
.layer(4, new DenseLayer.Builder()
    .nIn(LSTM_LAYER_SIZE)
    .nOut(DENSE_LAYER_SIZE)
    .activation(Activation.RELU)
    .build())
.layer(5, new RnnOutputLayer.Builder()
    .nIn(DENSE_LAYER_SIZE)
    .nOut(nOut)
    .activation(Activation.IDENTITY)
    .lossFunction(LossFunctions.LossFunction.MSE)
    .build())
.inputPreProcessor(4, new RnnToFeedForwardPreProcessor())
.inputPreProcessor(5, new FeedForwardToRnnPreProcessor())
.backpropType(BackpropType.TruncatedBPTT)
.tBPTTForwardLength(TRUNCATED_BPTT_LENGTH)
.tBPTTBackwardLength(TRUNCATED_BPTT_LENGTH)
.pretrain(false)
.backprop(true)
.build();

```

Рисунок 2.2 – Візуалізація другої частини конфігурації моделі

На етапі оптимізації головний фокус знаходиться на виборі конкретних значень параметрів, які були визначені та проаналізовані в процесі конфігурування моделі.

1. Для забезпечення стабільності та відтворюваності, значенням SEED було обрано 12345. Це гарантує, що результати моделі не змінюватимуться при повторному запуску, що є критичним для порівняння варіантів налаштувань та відбору оптимальних параметрів.

2. Кількість ітерацій встановлена на 1, що дозволяє оновлювати ваги на кожному циклі, підвищуючи швидкість навчання. Було протестовано більше значень, проте одна ітерація виявилася оптимальною для підтримання балансу між швидкістю і точністю.

3. Швидкість навчання була налаштована на 0.01 після тестування з вищими та нижчими значеннями. Це оптимальне значення забезпечило швидку конвергенцію без ризику значного коливання при оновленні ваг.

4. Алгоритм стохастичного градієнтного спуску (SGD) було обрано як оптимізаційний алгоритм після порівняння з іншими алгоритмами, такими як Adam і

RMSprop. З огляду на обсяг і складність даних, SGD продемонстрував стабільні результати з найменшою похибкою при навчанні.

5. Метод ініціалізації ваг Xavier показав кращі результати при налаштуванні інших параметрів. Це дозволило ефективно починати процес навчання з оптимальними значеннями активацій, знижуючи ризик надто великих або надто малих активацій на початкових етапах.

6. Архітектура шарів та кількість фільтрів у згорткових шарах була налаштована з використанням 25 фільтрів у першому шарі. Це значення дозволило досягти необхідного компромісу між складністю та ефективністю обробки ознак. У рекурентних шарах встановлено 512 нейронів, що забезпечило достатню потужність для прогнозування цінових патернів.

7. Функції активації ReLU для згорткових та рекурентних шарів показали оптимальні результати, прискорюючи навчання завдяки властивостям збереження градієнта для великих мереж.

8. L2-регуляризація зі значенням $1e-4$ виявилася достатньою для уникнення перенавчання, зберігаючи при цьому високу точність прогнозування.

9. Коефіцієнт Dropout 0.5 допоміг значно покращити узагальнення, знижуючи чутливість до окремих нейронів і дозволяючи моделі краще узагальнювати нові дані.

10. Truncated BPTT налаштовано на 22 тимчасових кроки, що стало оптимальним компромісом між обсягом оброблюваних даних і ефективністю навчання. Це значення дозволило обробляти досить довгі послідовності для аналізу часових закономірностей, не перевантажуючи модель і зберігаючи стабільну швидкість навчання.

Таким чином, в рамках налаштування та оптимізації моделі були обрані параметри, що забезпечують стабільну і точну роботу гібридної нейронної мережі для прогнозування цін криптовалют. Кожен елемент конфігурації відіграє ключову роль у підвищенні загальної ефективності, а правильний баланс між параметрами допомагає уникнути перенавчання та забезпечити високу здатність до узагальнення.

2.3 Метод створення функціоналу прогнозування цін криптовалют

Запропонований метод створення функціоналу прогнозування цін криптовалют базується на використанні оптимізованої гібридної LSTM + CNN моделі нейронної мережі, та візуалізації результатів у форматі лінійного горизонтального графіка за допомогою бібліотеки Google Charts [17]. Даний підхід забезпечує ефективний та зручний спосіб отримання орієнтовних майбутніх цін криптовалют, підвищуючи доступність і наочність прогнозів.

Основними етапами реалізації методу є підключення до AlphaVantage API [18], отримання та реформатування історичних даних, їх передача у нейромережу, що безпосередньо виконує генерацію масиву ймовірних майбутніх цін, повторна конвертація даних в JSON відповідно до правил бібліотеки Google Charts, яка надає можливість побудови графіків.

Першим етапом реалізації методу слугує процес отримання історичних даних, що починається з ініціалізації списку параметрів та створення об'єкта запиту на основі цих параметрів. Формат запиту продемонстровано на рисунку 2.3.

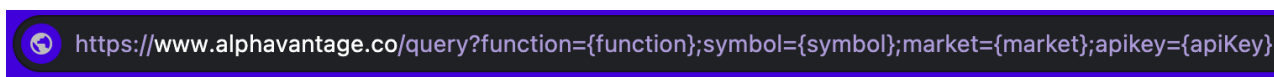


Рисунок 2.3 – Формат URL запиту до AlphaVantage API

Пояснення параметрів:

- **function:** назва функції, що визначає тип даних, який ми хочемо отримати у відповідь, в нашому випадку – “DIGITAL_CURRENCY_DAILY”;
- **symbol:** символ криптовалюти, для якої буде отримано колекцію історичних даних;
- **market:** назва ринку, відповідно до якого аналізуватиметься інформація, в поточному сценарії – EUR (Європа);
- **apikey:** секретний ключ, що видається після реєстрації на порталі, та надає доступ до API відповідно до рівня доступу. Більш комплексні типи даних можуть

потребувати вищих рівнів доступу, які зазвичай передбачають створення платної підписки.

Для відправки запиту до API створюється HTTP клієнт, що і отримує відповідь у форматі JSON. Цю відповідь необхідно перетворити у JSON об'єкти обраної мови програмування задля зручності взаємодії з даними, після чого дістати з об'єктів значення поля “2. high”, яке відповідає за найбільшу ціну вказаної криптовалюти, що була зафіксована в той день. Ціна заокруглюється до максимум п'яти знаків після коми, щоб уникнути відображення величезних десятичних частин користувачу. В результаті обробки історичних даних формується колекція у форматі “дата - ціна”, відсортована по даті, яка передається до логіки, що безпосередньо комунікує з нейромережею.

Наступним етапом реалізації методу є використання засобів бібліотек `deeplearning4j` та `nd4j` для конфігурації та оптимізації моделі гібридної нейронної мережі, що детально було описано у попередніх підрозділах. При отриманні запиту на створення прогнозу потенційних майбутніх цін певної криптовалюти, бізнес-логіка взаємодії з нейромережею отримує колекцію вхідних даних, для якої створюється власний ітератор. За допомогою ітератора, кожна пара значень із отриманою колекцією даних обробляється та аналізується, відповідно до чого в результаті заповнюється масив прогнозованих цін.

Фінальним етапом виступає перетворення масиву прогнозованих значень у JSON формат, який має певні правила. Він повинен містити список з елементів метаданих під назвою “cols” (рисунок 2.4). Перші два елементи призначені для визначення назв та типів даних для вісі X та Y відповідно. Третій елемент має тип даних `boolean` й інструктує бібліотеку, якою малювати лінію графіка – пунктирною, чи суцільною.

```
"cols": [
    {"id": "dt", "label": "Date", "type": "string"},
    {"id": "prc", "label": "Price", "type": "number"},
    {"id": "crt", "role": "certainty", "type": "boolean"}
],
```

Рисунок 2.4 – Список метаданих

JSON файл також повинен мати список самих даних “rows”, структура яких повинна відповідати метаданим. Це означає, що кожен елемент матиме вигляд, зображений на рисунку 2.5.

```
{
  "c": [
    {
      "v": "${date}"
    },
    {
      "v": "${price}"
    },
    {
      "v": "${certainty}"
    }
  ]
}
```

Рисунок 2.5 – Вигляд часової точки

Коли файл згенеровано, він потрапляє у функцію побудови, яка завдяки інструментарію бібліотки Google Chart створює лінійний горизонтальний графік, на якому присутні усі пари “дата - ціна” у вигляді часових точок – що значно зручніше та компактніше, ніж інформація, надана в звичайному текстовому форматі.

2.4 Розробка архітектури програмного модуля веб-сервісу рекомендацій по торгівлі криптовалютою

Ефективність програмного забезпечення для прогнозування, яке використовує нейронні мережі, на пряму залежить від ступеня навченості цих мереж, а отже і від якості та кількості даних, на яких здійснюється їх навчання. Нейронні мережі, щоб навчитися ефективно та точно виконувати завдання, що перед ними поставлені, потребують великого обсягу тренувальних даних. Ці дані повинні бути не лише численними, але й високоякісними, адже недостатність даних або їх низька якість можуть суттєво знизити точність прогнозів та навіть призвести до помилкових

результатів. Тренувальні дані необхідно підготувати у правильному форматі, який повинен відповідати конкретним завданням, які вирішує нейронна мережа. Наприклад, якщо нейронна мережа призначена для прогнозування вартості криптовалют, то її тренувальні дані повинні містити історичні записи змін цін криптовалют, включаючи час і вартість на кожен окремий момент.

Після успішного навчання нейронної мережі формується її модель, яка в майбутньому може бути використана для вирішення різних завдань. У нашому випадку, навчена модель є основним елементом модуля прогнозування криптовалют, від якого залежить точність результатів. Процес роботи модуля розпочинається з того, що користувач обирає необхідну йому криптовалюту. Далі система отримує історичні дані для цієї криптовалюти у форматі, що містить пари значень "дата - ціна". Після цього дані передаються на обробку нейронній мережі, яка на основі своєї навченої моделі та отриманих даних генерує нові пари прогнозованих значень у тому ж форматі.

Наступним кроком результати прогнозування разом з історичними даними передаються до модуля графічного інтерфейсу. Завданням цього модуля є візуалізація даних у форматі горизонтального лінійного графіка, де на осі X розташовані дати, а на осі Y – відповідні ціни криптовалют. Для того, щоб користувач міг легко відрізнити історичні дані від прогнозованих, графік будується із застосуванням різних стилів ліній: для історичних даних використовується суцільна лінія, тоді як для прогнозованих – пунктирна. Таким чином, візуальне відображення даних стає зрозумілим і зручним для користувача, дозволяючи йому легко розрізнити минулі й прогнозовані значення.

Для представлення актуальних даних про криптовалюту було розроблено клас "CryptoStats", який включає в себе низку атрибутів. Ці атрибути містять унікальний ідентифікаційний номер криптовалюти, її поточний рейтинг, назву, символ, актуальну ціну, дату останнього оновлення ціни, ринкову капіталізацію та інші показники, такі як зміна ціни за різні часові інтервали у відсотках. Для представлення історичних даних створено клас "CryptoData", атрибутами якого є дата, ціна та символ

криптовалюти. Це дозволяє чітко структурувати інформацію та використовувати її для подальшої обробки нейронною мережею.

Ключову роль у процесі прогнозування відіграє клас “CryptoPricePrediction”, який містить основний метод “predict”. Цей метод відповідає за обчислення прогнозів і працює з об’єктом класу “CryptoDataSetIterator”, що дозволяє ітерувати по вхідним даним, зокрема історичним. У цьому ж методі використовується клас “NeuralNetworkBuilder”, що надає можливість створення нової нейронної мережі для прогнозування. Метод “predict” викликається з класу “CryptoService”, який відповідає за отримання та обробку як історичних, так і актуальних даних про криптовалюти. У цьому класі також використовується об’єкт “CmcApiClient”, що забезпечує доступ до актуальних даних, та об’єкт “AlphaVantageClient”, відповідальний за отримання історичних даних.

Заключним етапом у процесі є передача отриманих актуальних, історичних та прогнозованих даних на front-end частину для відображення. Цим займається клас “CryptoController”, який взаємодіє з об’єктом класу “CryptoService”. Клас “CryptoController” передає дані до об’єкта класу “Model”, додаючи до нього відповідні атрибути. Після цього дані обробляються за допомогою JavaScript функції “drawChart”, що знаходиться у файлі “crypto-stats.html”. Функція перетворює дані з JSON-формату у рядковий формат і використовує їх для побудови лінійного графіка за допомогою бібліотеки Google Charts, мови розмітки HTML та каскадних стилів CSS. Результатом є графік, який бачить користувач, і який дозволяє йому аналізувати як історичні, так і прогнозовані дані цін криптовалют.

Додатково до основного функціоналу, було взято до уваги проблему безпечності взаємодії зі створений веб-сайтом та безпеку його існування в цілому. Для цього впроваджено механізми реєстрації нового користувача та авторизації, з розподілом на дві ролі: адміністратор та звичайний користувач. У звичайного користувача є доступ до більшості функціоналу. Тим часом, у адміністратора є можливість скористатися сторінкою, де наведена інформація про зареєстрованих користувачів, що включає їх імена та ролі. Крім перегляду звичайних статистичних

даних, з'являється функція блокування користувачів, якщо на те є певні причини. За потреби, користувача можна розблокувати.

Крім того, система також передбачає механізм обмеження трафіку від користувачів, що дозволяє запобігти перевантаженню серверів через велику кількість запитів за короткий час. Хоча цей механізм не стосується безпосередньо логіки модуля прогнозування криптовалют, його наявність суттєво покращує загальну стабільність і надійність системи.

На рисунку 2.6 продемонстровано діаграму класів системи.

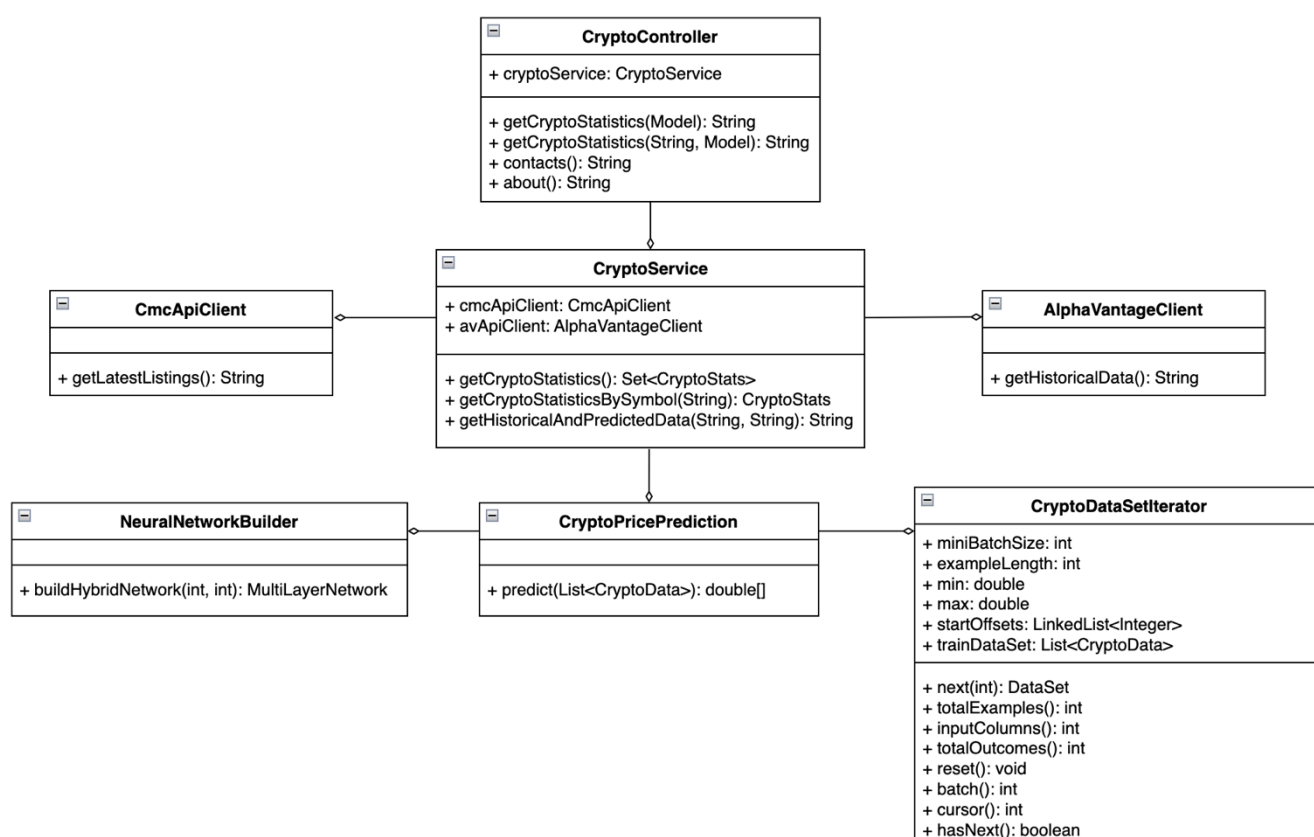


Рисунок 2.6 – Діаграма класів

Розроблюваний модуль для збереження та подальшого використання моделі нейронної мережі зберігає її у файлі з назвою “crypto-price-model.zip”. Вся логіка, пов’язана безпосередньо з роботою нейронної мережі, реалізована із застосуванням бібліотек `deeplearning4j` та `nd4j`. Для виклику API сервісів `CoinMarketCap` та `AlphaVintage` використовуються інструменти бібліотеки `Apache HttpClient`. Після

отримання даних з API, вони обробляються відповідно до вимог системи за допомогою бібліотеки gson. Крім цього, модуль інтегровано з фреймворком Spring, що залучає різноманітні його компоненти для забезпечення функціонування.

На рисунку 2.7 наведено діаграму компонентів програми.

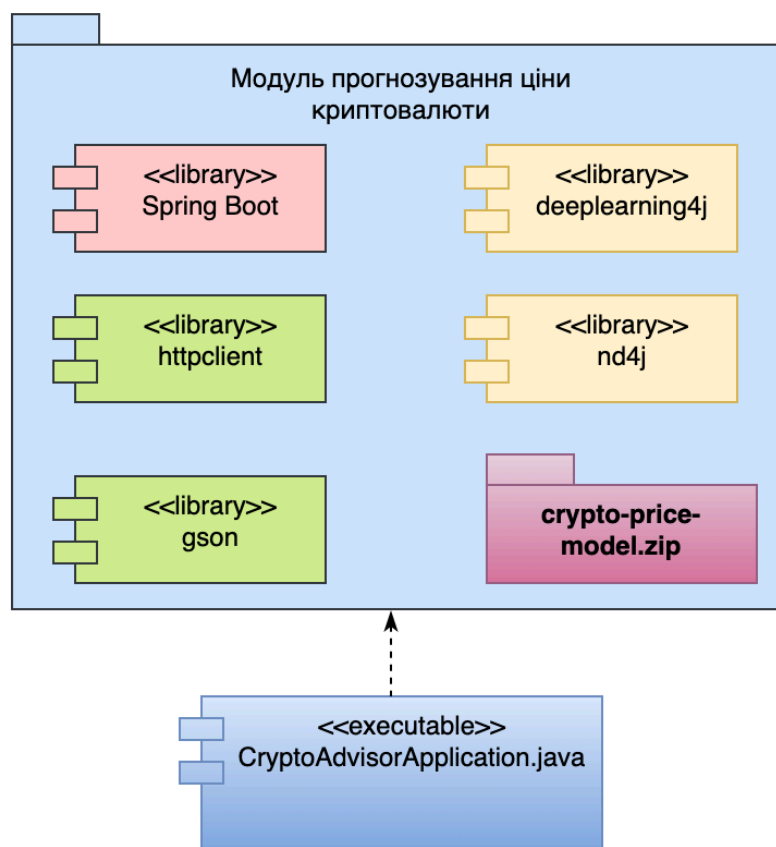


Рисунок 2.7 – Діаграма компонентів

Розроблений програмний продукт функціонує за двома основними сценаріями: перший сценарій передбачає отримання даних для всіх доступних криптовалют, а другий зосереджується на отриманні інформації про конкретну криптовалюту. В обох випадках процес починається з отримання актуальних даних. У другому сценарії додатково виконуються кроки для отримання історичних та прогнозованих даних. Алгоритм прогнозування включає збір історичних даних, їхню обробку, завантаження моделі, запуск нейронної мережі на основі цієї моделі, та подальше форматування результатів для відображення.

Діаграму варіантів використання програми продемонстровано на рисунку 2.8.



Рисунок 2.8 – Діаграма використання роботи програми

Використовуючи засоби мови UML побудовано діаграму діяльності розроблюваного веб-сервісу (рисунок 2.9).

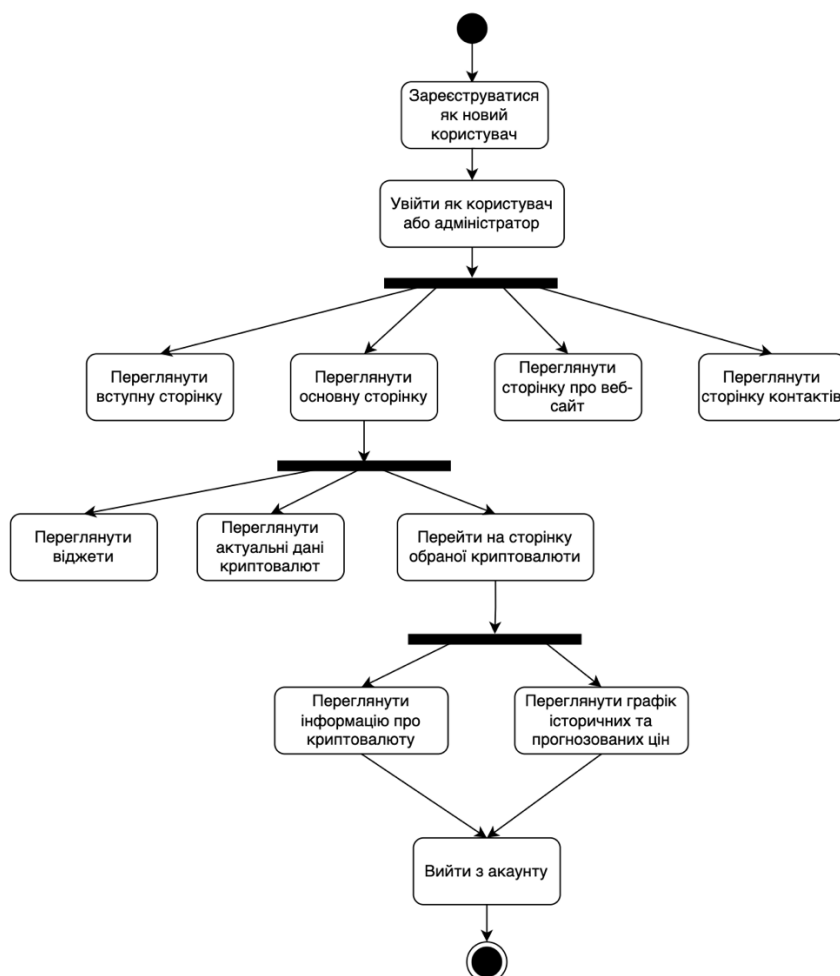


Рисунок 2.9 – Діаграма діяльності

Оскільки додаток реалізований у вигляді веб-сервісу, взаємодія з ним здійснюється через прості послідовності дій та зручну навігацію, що значно покращує загальний досвід використання.

2.5 Розробка алгоритмів функціонування системи

Основними алгоритмами, необхідними для створення програми, є: загальний алгоритм, алгоритм отримання історичних даних, алгоритм прогнозування та алгоритм отримання прогнозованих даних.

Загальний алгоритм (рисунок 2.10) складається з декількох етапів. Спершу відбувається отримання історичних даних, потім їх обробка та форматування, далі виконується отримання прогнозованих даних, після чого історичні та прогнозовані дані об'єднуються у відповідному форматі для подальшого повернення результатів користувачу.

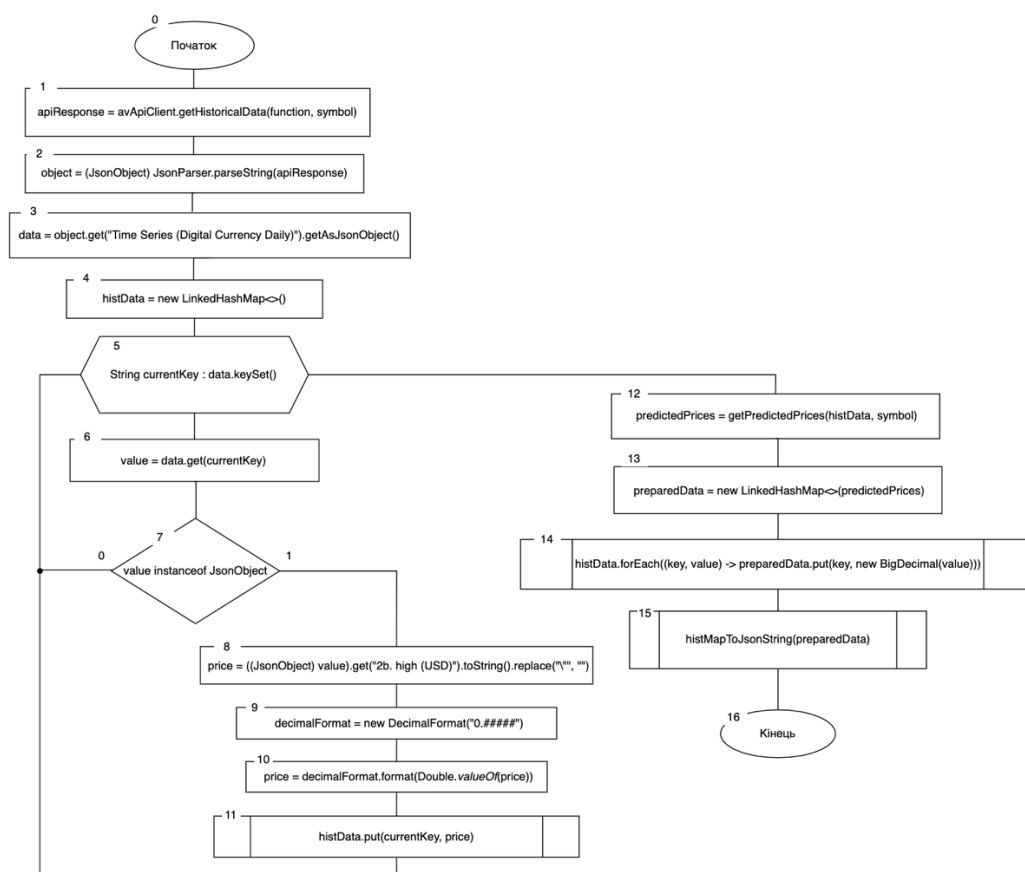


Рисунок 2.10 – Загальний алгоритм додатку

На рисунку 2.11 зображено алгоритм отримання історичних даних. Процес починається з ініціалізації списку параметрів та створення об'єкта запиту на основі цих параметрів. Після цього створюється HTTP клієнт, який відправляє запит до сервера та отримує відповідь. Отримана відповідь обробляється та передається до відповідного методу сервісу для подальшого використання.

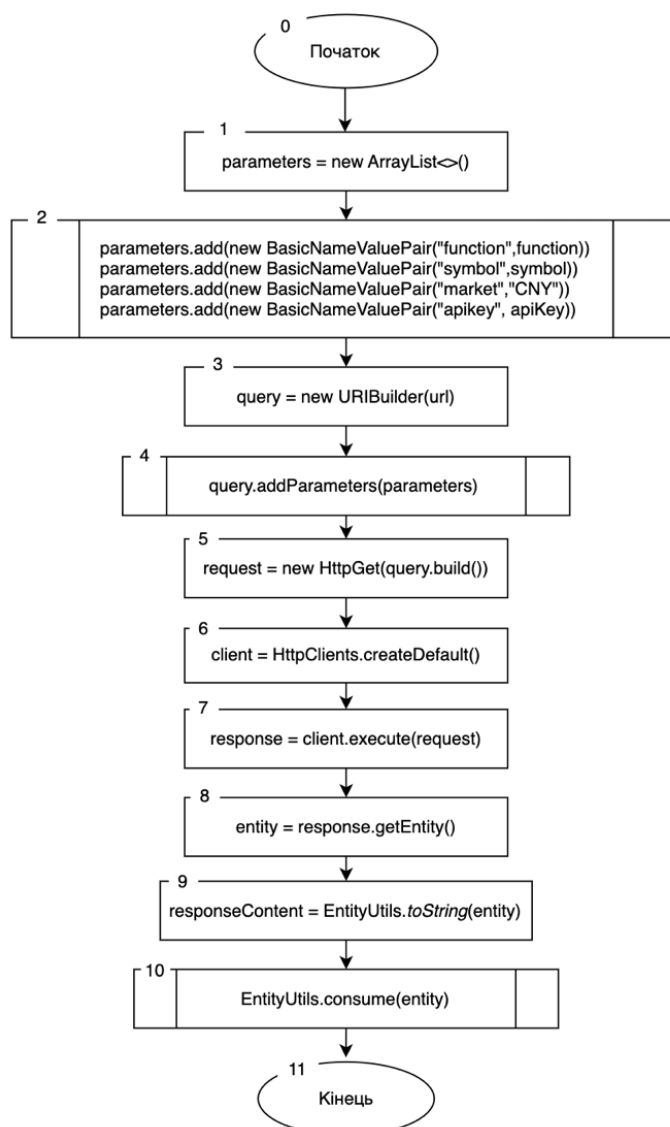


Рисунок 2.11 – Алгоритм отримання історичних даних

Алгоритм прогнозування ціни криптовалюти (рисунок 2.12), побудований з використанням бібліотек `deeplearning4j` та `nd4j`, включає кілька кроків: спершу ініціалізуються необхідні компоненти, потім проводиться тренування моделі

нейронної мережі та її збереження, після чого відбувається ітерація по вхідним даним, під час якої заповнюється масив прогнозованих цін.

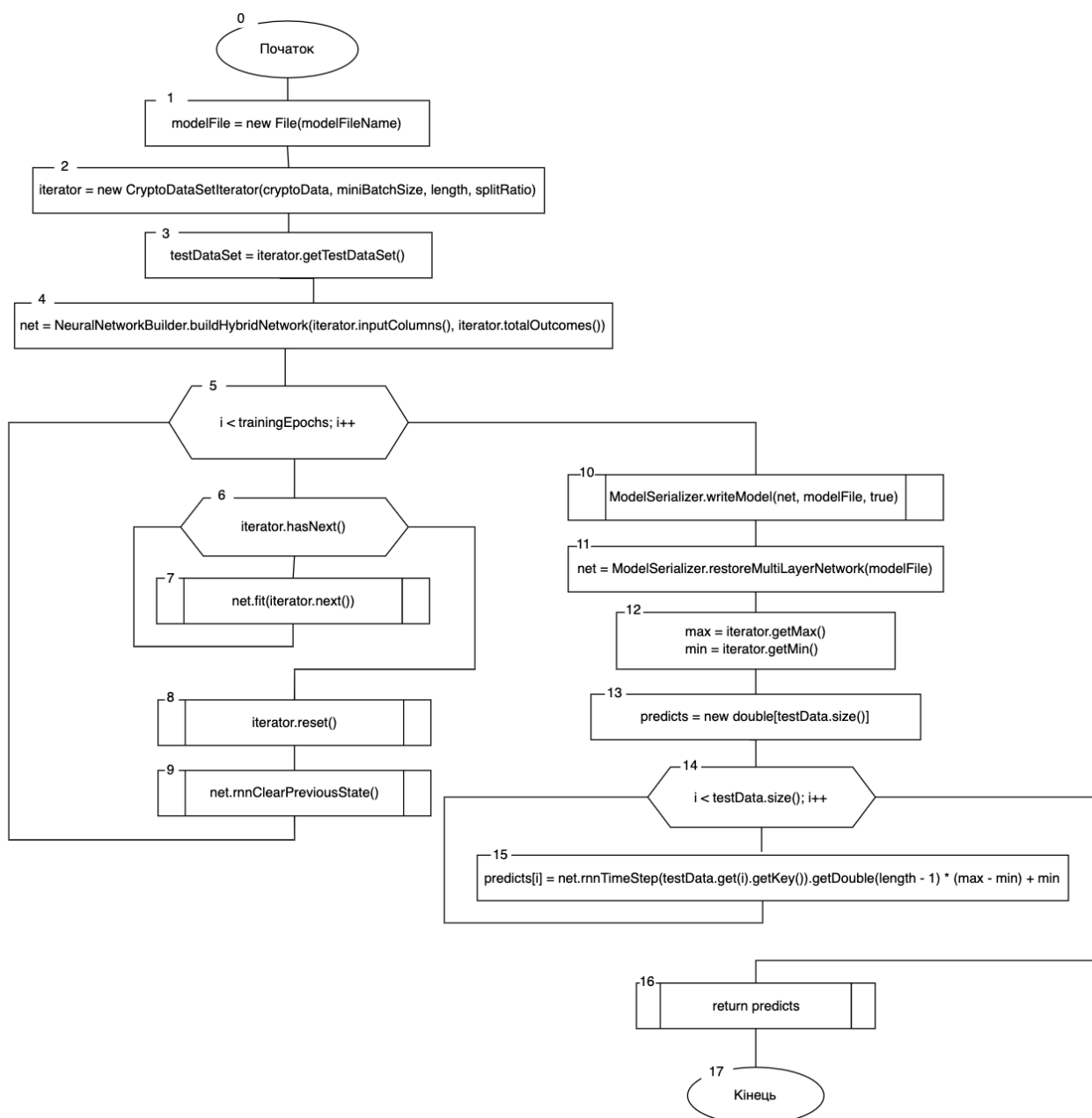


Рисунок 2.12 – Алгоритм прогнозування

Алгоритм отримання прогнозованих даних (рисунок 2.13) починається з сортування та форматування історичних даних, яке передбачає пакування цих даних в окремі об'єкти. Далі створені об'єкти передаються до функції прогнозування, яка повертає прогнозовані ціни криптовалют.

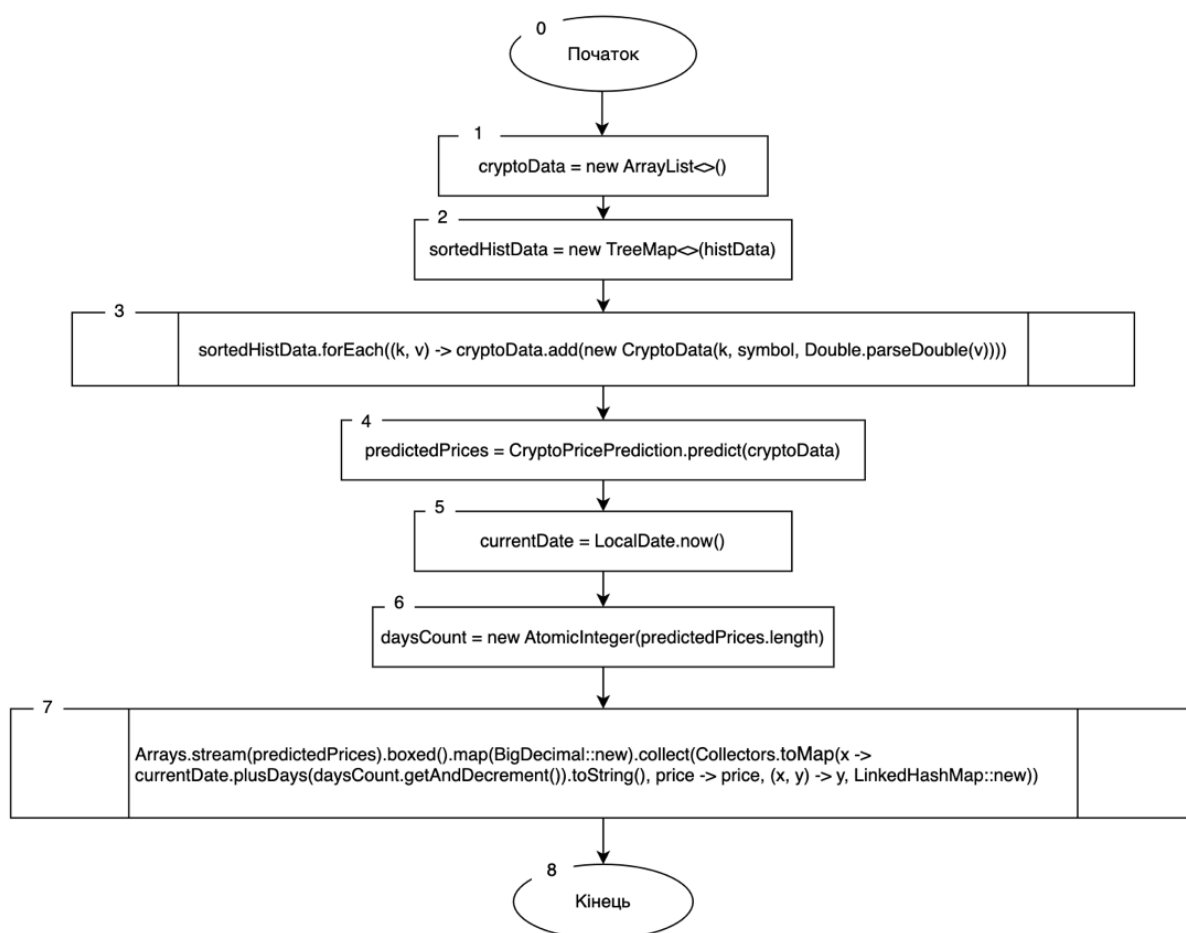


Рисунок 2.13 – Алгоритм отримання прогнзованих даних

На основі масиву прогнзованих значень створюються пари у форматі “дата, ціна”, які потім використовуються для подальшого аналізу та відображення у форматі лінійного горизонтального графіку.

2.6 Розробка структури інтерфейсу

Для веб-сервісів, що функціонують у сфері криптовалют, одним із ключових факторів є забезпечення користувачів приємним, інтуїтивно зрозумілим, а водночас функціональним графічним інтерфейсом. Враховуючи специфіку криптовалютного ринку, де інформація постійно оновлюється та може бути складною для сприйняття, зручний інтерфейс стає невід’ємною частиною успішного програмного продукту. Тому при розробці даного веб-сервісу було приділено особливу увагу дизайну та структурі інтерфейсу для максимальної зручності користувачів.

Інтуїтивний інтерфейс забезпечує легкість освоєння системи користувачем та виключає потребу в додатковому навчанні чи інструкціях. Основною характеристикою такого інтерфейсу є його здатність відповідати очікуванням і природним інстинктам користувача, дозволяючи йому легко знаходити потрібні функції та розуміти структуру взаємодії. За словами дослідників, інтуїтивність інтерфейсу значною мірою залежить від дотримання принципів когнітивного дизайну, таких як мінімізація когнітивного навантаження, забезпечення консистентності елементів і використання зрозумілих візуальних підказок [19]. Інтерфейси можна вважати інтуїтивно зрозумілими, якщо вони відповідають культурним та технічним уявленням користувачів. Крім того, ефективне використання простих і зрозумілих елементів управління, таких як кнопки та поля введення з чітким маркуванням, є критично важливим.

Приємність інтерфейсу для сприйняття користувачем базується не лише на його функціональності, але й на естетичних аспектах. Візуально приємний інтерфейс забезпечує гармонію між кольорами, шрифтами та розташуванням елементів. Крім того, згідно з дослідженнями, суб'єктивне сприйняття користувачем може покращуватися завдяки використанню плавних анімацій, гармонійного дизайну і відповідності сучасним трендам у сфері UI/UX-дизайну [20]. Таким чином, поєднання візуальної привабливості та зручності значно підвищує ефективність роботи користувача із системою.

Інтерфейс, створений для веб-сервісу рекомендацій по торгівлі криптовалютою, є прикладом інтуїтивного та приємного дизайну. Він побудований з урахуванням принципів доступності та когнітивного дизайну, що забезпечує мінімальну криву навчання для нових користувачів. В основі його структури лежить чітка ієрархія елементів, з якими легко взаємодіяти. Крім того, використання гармонійної палітри кольорів, сучасних шрифтів і візуально зрозумілих кнопок створює приємний естетичний ефект. Усі ключові функції розташовані так, щоб забезпечити швидкий доступ, що робить роботу з системою ефективною та комфортною. Таким чином, створений інтерфейс відповідає як потребам

користувачів, так і вимогам сучасного дизайну. Далі розглянемо структуру інтерфейсу більш детально.

Верхня навігаційна панель створеного веб-додатку містить кілька основних пунктів меню. Пункт "Crypto Advisor" веде на вступну сторінку, де користувачі можуть ознайомитися з основною інформацією про можливості веб-сервісу. Розділ "Home" відправляє користувачів на сторінку, яка відображає свіжі дані про різноманітні криптовалюти, включаючи ціни та іншу корисну інформацію. Пункт "About" містить інформацію про сам веб-сайт і його мету, а розділ "Contacts" пропонує користувачам можливість зв'язатися з командою розробників або отримати підтримку. Така чітка і лаконічна структура меню робить навігацію по сервісу зручною та інтуїтивною, навіть для користувачів, які не мають глибоких технічних знань.

Під навігаційною панеллю розміщується основне робоче поле програми. На головній сторінці це поле складається з трьох віджетів та великої таблиці, що містить актуальну інформацію про криптовалюти. Таблиця включає такі дані, як порядковий номер криптовалюти в рейтингу, її назву, символ, поточну ціну, ринкову капіталізацію та відсоткову зміну ціни за різні періоди часу. Завдяки такому відображенню користувачі можуть швидко оцінити стан ринку і отримати необхідну інформацію для ухвалення рішень. На рисунку 2.14 представлено схему інтерфейсу головної сторінки програми.

На схемі зображено наступні елементи домашньої сторінки:

1. Пункт меню "Crypto Advisor" – посилання на вступну сторінку;
2. Пункт меню "Home" – перехід на сторінку з актуальними даними;
3. Пункт меню "About" – інформація про веб-сервіс;
4. Пункт меню "Contacts" – контактна інформація розробників;
5. Кнопка "Log out" – вихід з акаунту та перехід на сторінку логіну;
6. Робоча область – основне поле для відображення контенту;
7. Віджет "Trending" – показує популярні криптовалюти, що швидко набирають популярність;
8. Віджет "Stable" – інформація про стабільні криптовалюти;

9. Віджет "Deviating" – дані про криптовалюти з найбільш значними коливаннями цін;

10. Таблиця з актуальними даними криптовалют – відображає основну інформацію про різні криптовалюти.

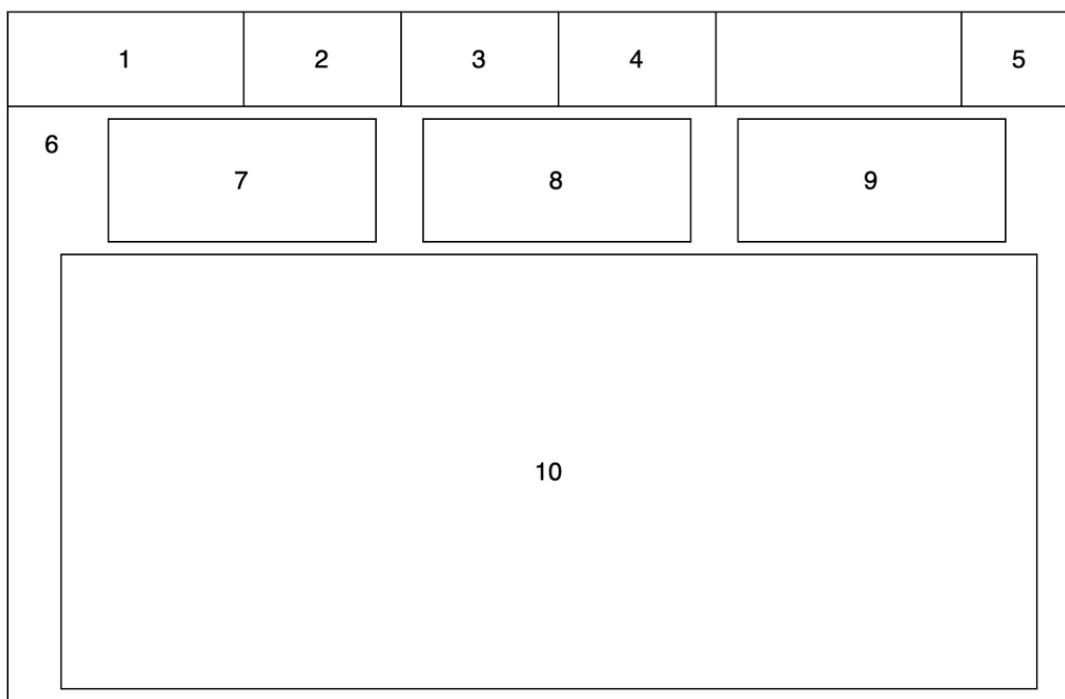


Рисунок 2.14 – Схема інтерфейсу домашньої сторінки програми “Crypto Advisor”

Кожен рядок таблиці має гіперпосилання на окрему сторінку, де можна переглянути детальні дані по конкретній криптовалюті. При переході на сторінку індивідуальної криптовалюти на робочому полі відображається її актуальна інформація, включаючи назву, символ, ціну та інші важливі показники. Нижче розташований горизонтальний лінійний графік, який відображає історичні дані та прогнозовані ціни для обраної криптовалюти. Такий підхід дозволяє користувачам легко переглядати минулі зміни ціни та оцінювати можливі майбутні тенденції зростання чи падіння. На рисунку 2.15 наведено схему інтерфейсу сторінки індивідуальної криптовалюти.

Елементи інтерфейсу цієї сторінки включають:

- Пункти меню (від 1 до 5) аналогічні до тих, що знаходяться на головній сторінці;
- Пункт №7 – актуальна інформація про криптовалюту, показує поточні дані для обраної монети;
- Пункт №8 – лінійний горизонтальний графік, що візуалізує історичні та прогнозовані дані цін.

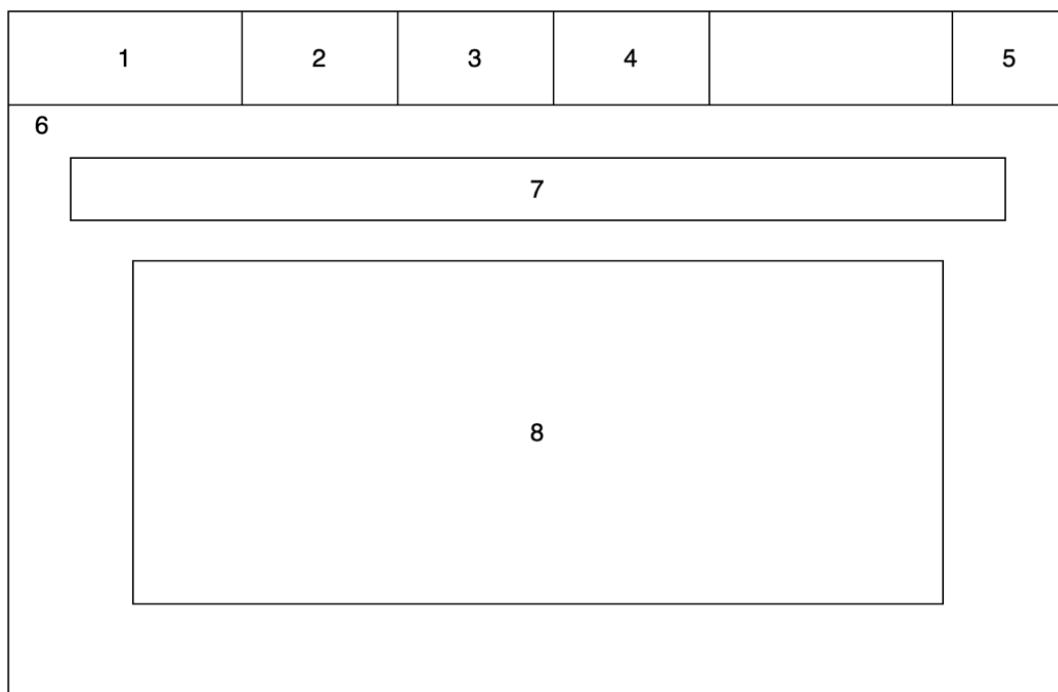


Рисунок 2.15 – Схема інтерфейсу сторінки індивідуальної криптовалюти

У рамках розробки інтерфейсу було використано кольорову гаму, що поєднує блакитні, білі та сині відтінки, створюючи привабливий та сучасний вигляд. Вибір цих кольорів зумовлений необхідністю виділити сервіс серед інших платформ у криптовалютній сфері, адже, окрім функціональності, важливим є візуальне сприйняття інтерфейсу користувачами.

2.7 Висновки

У даному розділі проведено та описано процес конфігурування гібридної моделі нейронної мережі та здійснено оптимізацію параметрів. Також проаналізовано

особливості вхідних та вихідних даних для прогнозування ціни криптовалюти. Описано запропонований метод розробки функціоналу прогнозування цін криптовалют. Крім того, розроблено схеми інтерфейсу головної сторінки веб-сайту та індивідуальної сторінки криптовалюти. Розроблено UML-моделі системи. Також наведено блок-схеми основних алгоритмів програмного модуля прогнозування цін криптовалют.

3 РОЗРОБКА ПРОГРАМИ ДЛЯ ВЕБ-СЕРВІСУ РЕКОМЕНДАЦІЙ ПО ТОРГІВЛІ КРИПТОВАЛЮТОЮ

3.1 Варіантний аналіз і обґрунтування вибору інструментів та технологій розробки

3.1.1 Варіантний аналіз і обґрунтування вибору мови програмування

Ефективність програмного продукту значною мірою залежить від вибору технологій, зокрема мови програмування, бібліотек, фреймворків та середовища розробки. Розглянемо три популярні мови програмування: Java, C#, і C++.

Java – це об'єктно-орієнтована мова, створена компанією Sun Microsystems, яка пізніше була придбана компанією Oracle. З моменту свого випуску в 1995 році, вона стала однією з найбільш поширених мов програмування у світі. Основна особливість Java полягає у її платформенній незалежності – програми, написані на Java, можуть виконуватися на різних операційних системах (Windows, macOS, Linux) без перекомпіляції завдяки Java Virtual Machine (JVM), яка перетворює байт-код Java у машинний код, зрозумілий для конкретної ОС [21].

Java також спрощує роботу з пам'яттю завдяки автоматичному збиранню сміття, який вивільняє невикористану пам'ять без участі програміста. Це робить розробку на Java зручнішою, оскільки не потрібно вручну керувати пам'яттю або піклуватися про її очищення [22].

Серед інших переваг Java можна виділити:

1. Об'єктно-орієнтований підхід. Принципи об'єктно-орієнтованого програмування дозволяють створювати структурований, модульний і повторно використовуваний код, що полегшує розробку, тестування та підтримку програм.

2. Безпека. Java забезпечує високий рівень безпеки завдяки системі контролю доступу та цифрових сертифікатів, що дозволяє створювати надійні рішення для різних середовищ.

3. Багатопотоковість. Java підтримує багатопотоковість, що дозволяє виконувати кілька процесів паралельно в межах однієї програми. Це особливо корисно для мережових додатків або програм, що працюють з великими обсягами даних.

4. Широкий вибір бібліотек та фреймворків. Java має багатий набір бібліотек і фреймворків для вирішення різних завдань, таких як робота з базами даних, створення графічного інтерфейсу або обробка мережових запитів, що прискорює розробку та підвищує якість кінцевого продукту.

5. Велика спільнота розробників. Широка спільнота Java-програмістів надає доступ до величезної кількості навчальних матеріалів, форумів та блогів, де можна знайти допомогу й підтримку.

6. Розширюваність. Java дозволяє створювати власні класи, бібліотеки та фреймворки, що дає змогу розширювати можливості мови під конкретні потреби проекту.

7. Бібліотеки для машинного навчання. Для Java існує багато бібліотек з відкритим кодом для машинного навчання, що активно підтримуються спільнотою, сприяючи впровадженню сучасних алгоритмів у різні проекти.

Таким чином, Java є потужним інструментом для створення кросплатформених, безпечних і масштабованих програмних рішень, зокрема в галузі машинного навчання.

C# – це об'єктно-орієнтована мова програмування, створена Microsoft і вперше представлена в 2000 році як частина платформи .NET. Основною метою розробки було розширення можливостей C++ і спрощення процесу створення програмного забезпечення.

C# є потужною і універсальною мовою, що використовується для розробки різноманітних типів додатків, включаючи веб-додатки, настільні програми, мобільні додатки та ігри. Вона підтримує ключові концепції об'єктно-орієнтованого програмування, такі як спадкування, поліморфізм, абстракція та інтерфейси, що робить код більш зрозумілим, зручним у підтримці та легким для розширення. Також

C# відрізняється наявністю сильної системи типів, яка підвищує надійність і безпеку програмного забезпечення.

Однією з важливих характеристик C# є автоматичне управління пам'яттю завдяки вбудованому збирачу сміття, що допомагає мінімізувати ризики витоків пам'яті та пов'язаних з ними помилок. Крім того, C# активно підтримує асинхронне програмування, що дозволяє створювати високоефективні додатки, які можуть взаємодіяти з мережею або іншими процесами без блокування основного потоку виконання.

C# також пропонує розширені можливості роботи з базами даних. Завдяки таким інструментам, як ADO.NET, Entity Framework та іншим ORM (Object-Relational Mapping) бібліотекам, програмісти можуть легко працювати з реляційними та іншими типами баз даних.

Ще одна ключова сфера застосування C# – це розробка веб-додатків через ASP.NET. Використовуючи цей фреймворк, розробники можуть створювати веб-сторінки, веб-сервіси та інші типи веб-застосунків, що розширює можливості мови.

Окрім того, C# широко використовується в індустрії відеоігор, особливо завдяки платформі Unity, де C# виступає основною мовою для створення 2D та 3D ігор для різних платформ, таких як ПК, консолі та мобільні пристрої [23].

C++ – це універсальна мова програмування, створена на основі мови C в 1980-х роках, яка набула значної популярності серед розробників. Вона використовується для створення широкого спектра програмних рішень, включаючи настільні додатки, вбудовані системи та відеоігри.

Однією з основних переваг C++ є можливість безпосереднього доступу до пам'яті та низькорівневого програмування, що дозволяє розробникам мати повний контроль над управлінням ресурсами та оптимізацією продуктивності додатків. Це особливо корисно для розробки високопродуктивних систем.

C++ також відіграє важливу роль в ігровій індустрії, де такі популярні ігрові двигуни, як Unreal Engine та Unity, використовують її як одну з основних мов. Завдяки цьому розробники можуть створювати потужні ігри з високими вимогами до продуктивності для різних платформ.

Крім того, C++ широко застосовується у системному програмуванні та вбудованих системах. Її здатність працювати безпосередньо з апаратним забезпеченням робить її ідеальною для розробки операційних систем, драйверів пристроїв, мікроконтролерів та інших вбудованих систем, де необхідний тісний зв'язок з апаратною частиною [24].

Однак C++ вважається складнішою мовою для освоєння, оскільки вимагає глибшого розуміння низькорівневих аспектів програмування, таких як управління пам'яттю та оптимізація використання ресурсів. Через це розробники повинні приділяти більше уваги деталям, щоб уникнути потенційних помилок та проблем з продуктивністю.

Переваги та недоліки всіх розглянутих мов підсумовано в таблиці 3.1.

Таблиця 3.1 – Порівняння мов програмування

№	Критерій	C++	C#	Java
1.	Мультиплатформенність	-	+	+
2.	Багатопотоковість	+	+	+
3.	Присутність збірника сміття	-	+	+
4.	Легкість створення веб-додатків	-	-	+
5.	Наявність бібліотек для машинного навчання	+	+	+
6.	Високе різноманіття бібліотек та фреймворків	-	-	+
	Підсумок:	2/6	4/6	6/6

Аналіз мов програмування здійснювався на основі шести ключових критеріїв, які мають важливе значення при виборі мови для реалізації програмного продукту. Відповідно до результатів, наведених у таблиці 3.1, Java перевершує C++ на 66,6%, а

C# – на 33,3%. З огляду на всі переваги, саме Java була обрана як основна мова програмування для цього проекту.

Окрім того, для створення графіків на front-end частині було використано JavaScript, а для розмітки та оформлення веб-сторінок застосовано HTML та CSS.

3.1.2 Варіантний аналіз і обґрунтування вибору середовища розробки

Під час вибору інтегрованого середовища розробки (IDE) програмісти зазвичай стоять перед вибором між IntelliJ IDEA та Eclipse. Обидва середовища є потужними інструментами для розробки програмного забезпечення, але мають свої унікальні особливості, що впливають на їх вибір для конкретних задач.

IntelliJ IDEA – це високоефективне інтегроване середовище розробки, яке підтримує широкий спектр мов програмування, серед яких Java, Kotlin, Scala, Groovy та інші. Ця IDE, розроблена компанією JetBrains, стала однією з найбільш популярних серед професійних розробників завдяки інноваційним функціям і можливостям, що значно підвищують продуктивність розробки [25]. На рисунку 3.1 зображено приклад інтерфейсу IntelliJ IDEA.

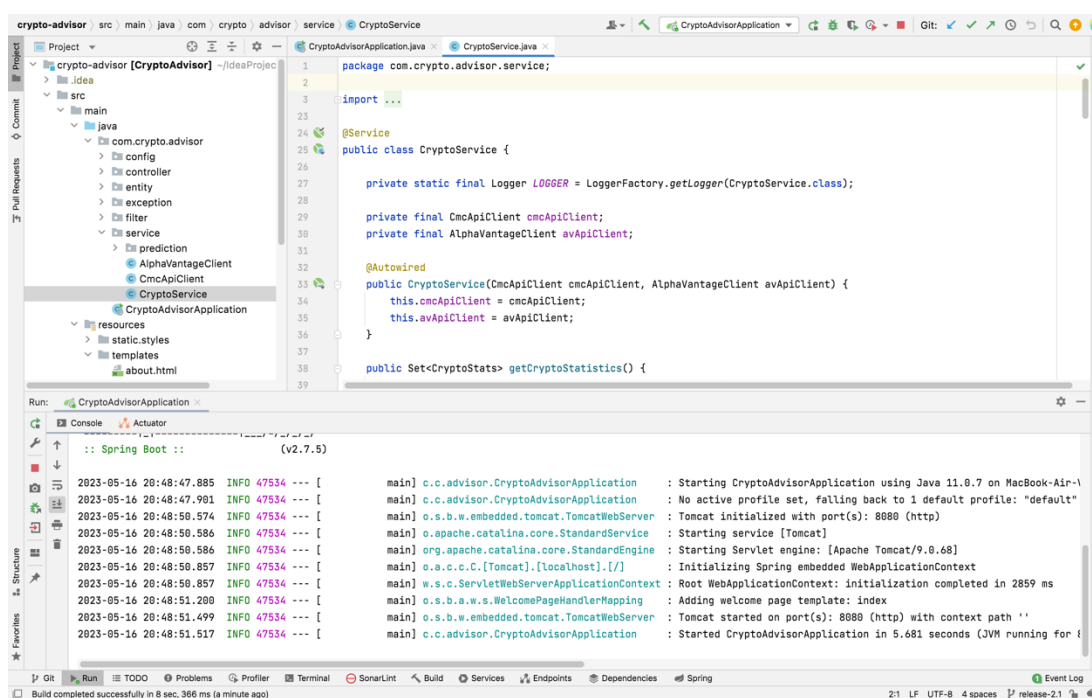


Рисунок 3.1 – Приклад інтерфейсу IntelliJ IDEA

Основні переваги IntelliJ IDEA:

1. Висока продуктивність та оптимізація ресурсів. IntelliJ IDEA відзначається швидкою обробкою великих проєктів і мінімальним використанням ресурсів комп'ютера, що дозволяє швидко перемикатися між файлами, компілювати код і виконувати інші задачі без втрати продуктивності. Це особливо важливо для розробників великих систем або корпоративних проєктів, де важлива ефективність робочого процесу.

2. Відладка та профілювання. IntelliJ IDEA має вбудовані інструменти для відладки, що дозволяють легко знаходити та виправляти помилки в коді. Крім стандартних функцій відладки, таких як крокування по коду та точкові зупинки, IDE також підтримує аналіз продуктивності програм за допомогою профілювання. Це дозволяє виявити вузькі місця в додатках та підвищити їх продуктивність.

3. Інтелектуальний редактор коду. IntelliJ IDEA має вбудований інтелектуальний редактор коду, який автоматично підказує варіанти автодоповнення, перевіряє синтаксис та пропонує швидке виправлення помилок. Це дозволяє скоротити час на написання коду та знижує кількість синтаксичних помилок. Такі інструменти як рефакторинг, кодові аналізатори та автоматичні рекомендації дозволяють поліпшити якість коду, що позитивно впливає на підтримуваність проєкту.

4. Широка функціональність та підтримка різних технологій. IntelliJ IDEA підтримує безліч мов програмування та фреймворків, включаючи Java, Kotlin, Scala, Groovy, JavaScript, TypeScript, HTML, CSS, Python, PHP, SQL та інші. Це робить її універсальним інструментом, придатним для розробки різних типів додатків, від веб-додатків до складних корпоративних рішень.

5. Розширена підтримка роботи з базами даних. IntelliJ IDEA забезпечує інтеграцію з різними базами даних, дозволяючи працювати з ними безпосередньо зі середовища розробки. Інструменти для редагування SQL-запитів, перегляду даних і налагодження інтегровані в середовище, що значно полегшує процес роботи з базами даних.

6. Екосистема плагінів. IntelliJ IDEA володіє величезним репозиторієм плагінів, які дозволяють значно розширити функціональність середовища та налаштувати його під конкретні потреби розробників. Репозиторій містить тисячі розширень для інтеграції з різними технологіями, інструментами управління проектами, базами даних тощо. Це робить IntelliJ IDEA дуже гнучким інструментом, який можна адаптувати для роботи з будь-якою технологією або платформою.

7. Підтримка популярних фреймворків та платформ. IntelliJ IDEA надає вбудовану підтримку для таких популярних фреймворків, як Spring, Hibernate, JavaFX, Android, JavaScript та інші. Це значно полегшує розробку на базі цих технологій, оскільки середовище автоматизує багато рутинних завдань, включаючи конфігурацію проектів, роботу з базами даних, написання тестів тощо.

Ці переваги роблять IntelliJ IDEA ідеальним вибором для професійних розробників, які шукають надійне та функціональне середовище для написання якісного коду, незалежно від складності або типу проекту.

Eclipse – це інше популярне інтегроване середовище розробки програмного забезпечення, яке зокрема підтримує розробку на Java. Воно було створене фондом Eclipse Foundation і доступне для вільного використання як відкритий проект. Цю IDE часто обирають завдяки її модульній архітектурі та багатій підтримці інструментів для розробки [26]. На рисунку 3.2 наведено приклад інтерфейсу Eclipse.

Основні риси Eclipse:

1. Кодовий редактор. Редактор коду в Eclipse підтримує автодоповнення, перевірку синтаксису, рефакторинг та інші функції, що допомагають підвищити продуктивність під час написання коду. Крім того, середовище надає інструменти для налагодження коду, включаючи крокування по коду, використання точок зупинки та інші можливості відладки.

2. Модульна архітектура. Eclipse має архітектуру на основі плагінів, що дозволяє легко розширювати її функціональність. Завдяки можливості додавати плагіни для різних мов, фреймворків та інструментів, Eclipse адаптується під конкретні потреби користувача. Однак, кількість доступних плагінів у репозиторії

Eclipse Marketplace менша порівняно з IntelliJ IDEA, що може обмежити можливості налаштування.

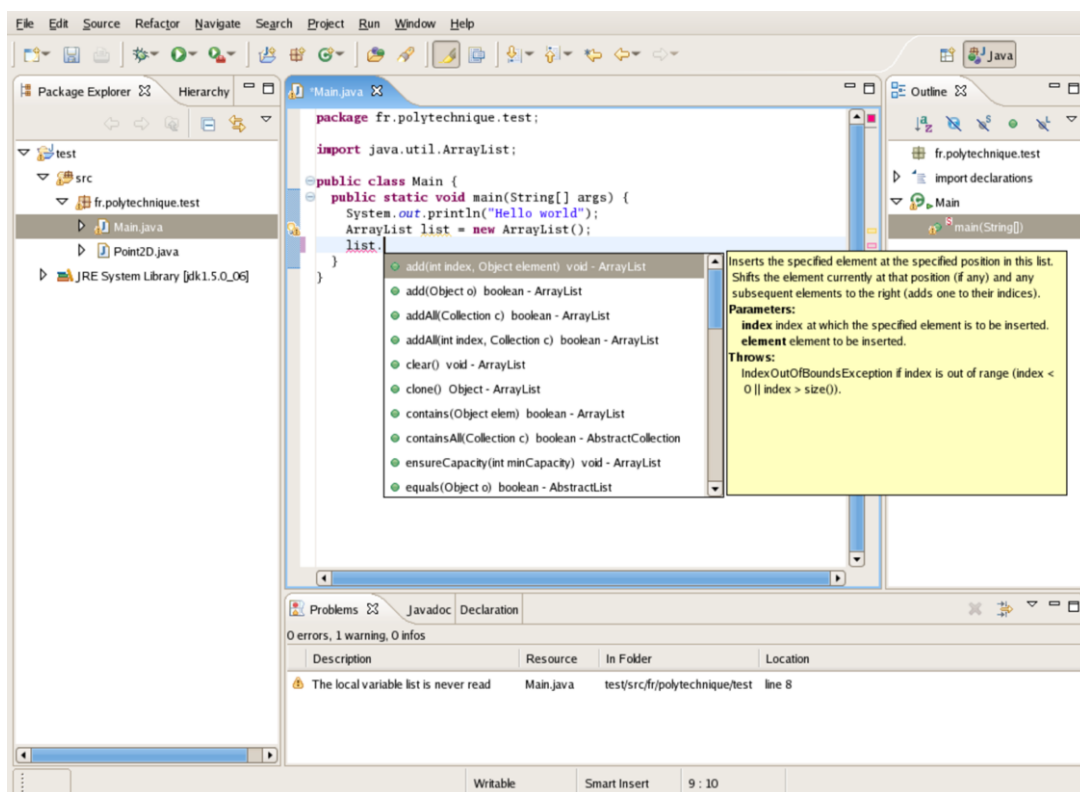


Рисунок 3.2 – Приклад інтерфейсу Eclipse IDE

3. Інтеграція зі сторонніми інструментами. Eclipse забезпечує інтеграцію з такими інструментами, як Apache Maven, Ant, JUnit, що дозволяє автоматизувати процес складання проєкту, тестування та виконання інших завдань. Це робить середовище придатним для розробки складних багатокомпонентних проєктів.

4. Підтримка систем контролю версій. Eclipse інтегрується з такими системами контролю версій, як Git, SVN, CVS та іншими. Це дозволяє легко керувати версіями проєкту та працювати в команді, ефективно співпрацюючи над кодом.

5. Підтримка різних операційних систем. Eclipse доступний для Windows, macOS і Linux, що дозволяє розробникам використовувати його на будь-якій платформі без обмежень.

Хоча Eclipse залишається популярним середовищем, IntelliJ IDEA все ж користується більшою популярністю серед розробників Java завдяки своїм передовим

інструментам та можливостям аналізу проєктів. IntelliJ індексує весь код проєкту, створює синтаксичне дерево і глибоко аналізує код, що дозволяє швидко знайти помилки, рефакторити код та працювати з великими проєктами без втрат продуктивності.

Згідно зі звітом The State of Developer Ecosystem 2021 [27], 75% розробників Java обрали IntelliJ IDEA як основне середовище розробки, тоді як Eclipse використовується лише 11% розробників. На рисунку 3.3 зображена діаграма, що ілюструє ці дані.

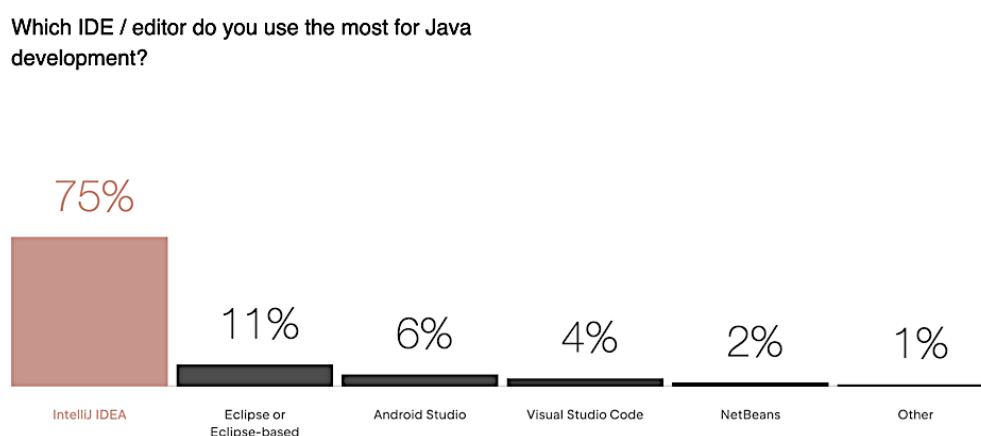


Рисунок 3.3 – Статистичні дані використання різних IDE розробниками

Таким чином, беручи до уваги власний досвід, статистичні дані та аналіз інших альтернатив, у якості інтегрованого середовища розробки було обрано IntelliJ IDEA Ultimate Edition, оскільки вона є найбільш функціональною та зручною у використанні IDE, з доступних на ринку.

3.2 Розробка програмного модуля

Розробка програмного модуля передбачає виконання декількох стадій, кожна з яких є не менш важливою за іншу. До них входить:

- Усвідомлення проблеми та формування вимог для рішення;
- Проектування архітектури та процесів;
- Програмна реалізація;

- Проведення тестування;
- Реліз у production;
- Експлуатація та підтримка.

Перші два етапи можна вважати успішно виконаними, оскільки вимоги було сформовано у першому розділі, а проектування архітектури та процесів було здійснено у другому розділі. Далі можна переходити до стадії програмної реалізації.

У першу чергу було створено новий проект за допомогою середовища IntelliJ IDEA. Далі було прописано усі необхідні залежності на зовнішні бібліотеки у файлі `pom.xml`, так як для побудови проекту використовується Maven. Наступним кроком стало створення всіх потрібних Java класів (рис. 3.4). Кожен клас в програмному модулі належить певному пакету, яких на найвищому рівні налічується 9, кожен з яких семантично відрізняється від інших. До згаданих пакетів входять:

- `config` – містить класи, що відповідають за конфігурацію програми;
- `controller` – відповідає за усі класи-контролери, що містять ендпоїнти, які викликаються через веб-браузер;
- `entity` – класи в які конвертуються таблиці з бази даних за допомогою object-relational mapping (ORM);
- `exception` – класи специфічних помилок, які можуть виникати під час роботи програми;
- `interceptor` – містить класи-перехоплювачі запитів, які виконують певну логіку перед тим, як виконається основна логіка застосунку;
- `model` – моделі, що зберігають стан та слугують для трансферу даних в межах програми;
- `repository` – класи-репозиторії, що виконують функцію діставання даних з бази даних;
- `security` – відповідає за логіку, що забезпечує функціонал авторизації/реєстрації;
- `service` – основний пакет, в якому знаходиться основна логіка програмного модуля.

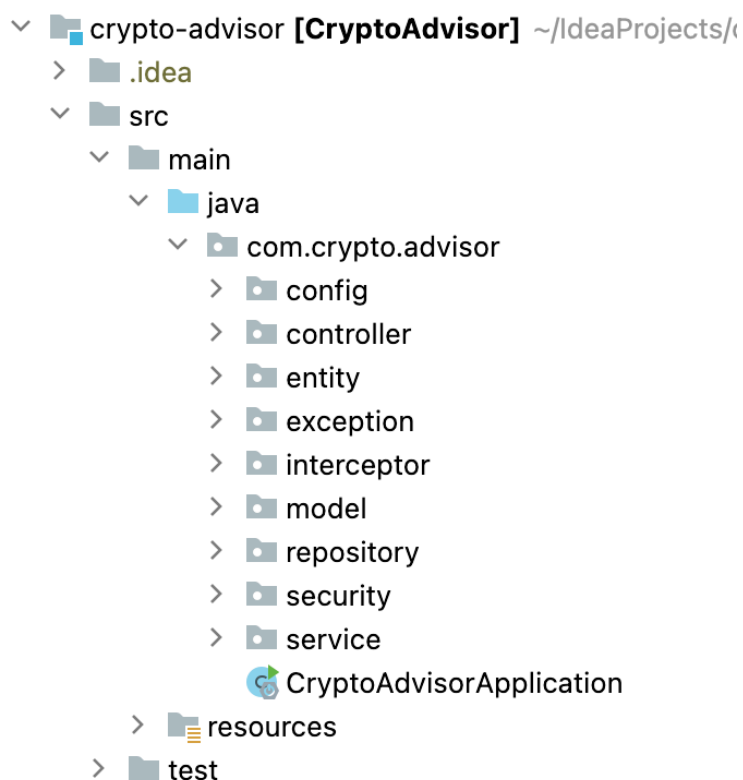


Рисунок 3.4 – Структура пакетів проекту

Вхідною точкою запуску програми є клас `CryptoAdvisorApplication`, який відмічений анотацією `@SpringBootApplication` та містить єдиний і найголовніший метод під назвою “main”. Після успішного старту веб-сервера стають доступними ендпоїнти контролерів, які можна викликати з веб-браузера.

Основним контролером є клас `CryptoController`. Саме він відповідальний за доступ до основного функціоналу веб-сервісу. Фактично в ньому відсутня бізнес-логіка, адже його ціль полягає в отриманні вхідних даних від користувача, їх обробці, передачі повноважень класу `CryptoService` для виконання мети запиту, та надсилання кінцевих результатів назад до користувача.

`CryptoController` має такі публічні методи:

- `getAllCryptoStatistics(Model model)` – цей метод дістає актуальну інформацію про найпопулярніші криптовалюти та формує наповнення для віджетів “trending”, “deviating” та “stable” криптовалют;
- `getCryptoStatistics(String symbol, Model model)` – знаходить актуальну інформацію для обраної криптовалюти, а також дістає масив історичних цін, за допомогою яких прогнозуються можливі майбутні ціни;

- `home()` – відповідає за головну сторінку веб-сайту;
- `contacts()` – відповідає за сторінку контактів розробників веб-сайту;
- `about()` – відповідає за сторінку з інформацією про веб-сайт.

Пакет `controller` містить ще три класи: `SecurityController`, `UserController` та `ErrorController`. Перший відповідає за доступ до логіки реєстрації та авторизації користувачів, другий за перегляд зареєстрованих користувачів (доступно лише для адміністраторів), а третій за сторінки помилок, які можуть виникнути у ході взаємодії з веб-сайтом рекомендацій по торгівлі криптовалютою. Оскільки ці контролери не мають прямого відношення до функціоналу прогнозування, детально описувати їх реалізацію не є доцільним.

Розглянемо шлях виконання користувачем запиту на отримання історичних та прогнозованих даних для обраної криптовалюти у програмному коді. Цей запит надходить до методу `getHistoricalAndPredictedData`.

Наступним кроком викликається метод `getHistoricalData` на об'єкті класу `AlphaVantageClient`, повертаючи історичні дані у форматі JSON через API порталу Alpha Vantage. Далі ці дані обробляються за допомогою бібліотеки `gson`, яка дозволяє перетворити JSON-елементи на Java-об'єкти і представити їх у вигляді пар "ключ - значення", де ключем виступає дата, а значенням – ціна криптовалюти.

Після отримання даних про історичні ціни криптовалюти у потрібному форматі, вони передаються як параметр до методу `getPredictedPrices`. У цьому методі дані сортуються і додатково форматуються для потреб прогнозування. Потім викликається статичний метод `predict` класу `CryptoPricePrediction`, куди й надходять відформатовані дані.

У методі `predict` реалізована основна логіка модуля прогнозування цін криптовалюти. Процес починається зі створення об'єкта класу `CryptoDataSetIterator` на основі вхідних даних, що буде використовуватися нейромережею. Далі створюється гібридна нейронна мережа за допомогою статичного методу `buildHybridNetworks` класу `NeuralNetworkBuilder`, після чого відбувається процес навчання моделі мережі, яка потім серіалізується у файл архіву для подальшого використання. Метод `predict` повертає масив прогнозованих цін, отриманих від

виклику методу `predictPriceOneAhead`, де міститься безпосередня логіка роботи з нейронною мережею.

Масив прогнозованих цін передається до методу `getPredictedPrices`, де шляхом додавання дат він трансформується у набір пар "дата - ціна". Наступним етапом є об'єднання історичних та прогнозованих даних в одну колекцію в методі `getHistoricalAndPredictedData`, після чого вони конвертуються у формат JSON за допомогою методу `mapToJsonString`.

3.3 Розробка інтерфейсу користувача

Директорія під назвою "resources" містить два пакети: `static styles` та `templates`. У `templates` знаходяться HTML файли для кожної з сторінок веб-сайту: `all-crypto-stats.html`, `crypto-stats.html`, `home.html`, `about.html`, `contacts.html`, `registration.html`, `login.html`, `error.html` та `users.html`. Всі сторінки мають навігаційну панель, закріплену у верхній частині, що реалізована з використанням структур `nav`, `ul` та `li`. Гіперосилання на інші сторінки стали можливими завдяку атрибуту `href`, значенням якого у форматі "`@{/page}`" задається адреса сторінки.

Пакет статичних стилів слугує лише для одного файлу – `styles.css`. За допомогою мови `Cascading Style Sheets (CSS)` було розроблено стиль візуальної частини, що включає в себе специфікацію масштабу елементів, їх координати на сторінці, кольорову гаму та інші конфігурації, як от підсвічення іншим кольором кнопки, на яку наведено курсор.

Як було згадано раніше, на сторінці обраної криптовалюти, після завершення бізнес-логіки прогнозування, історичні та прогнозовані ціни відображаються у вигляді лінійного горизонтального графіку. Це стало можливим завдяки конвертації списку класів-моделей у відповідний формат за допомогою функції-маппера та послідовному надсиланню переформатованих даних на `front-end` частину, а саме у функцію `drawChart` (рисунк 3.5), що написана мовою `JavaScript`. Конфігурація `options` містить налаштування ширини, висоти, координат та кольорової гами графіка, а

також стилі для вісі X та вісі Y, що включають в себе шрифт заголовків та їх розмір, колір тексту та напрямки.

```
var cryptoGraphJson = [{historicalData}]];
var data = new google.visualization.DataTable(JSON.parse(cryptoGraphJson));

var options = {
  width: 1950,
  height: 500,
  chartArea: {left:"5%", top:"12%", width:"90%", height:"75%"},
  hAxis: {
    direction: -1,
    titleTextStyle: {color: '#FFF', fontSize: 12},
    textStyle: {color: '#FFF', fontSize: 12}
  },
  vAxis: {
    titleTextStyle: {color: '#FFF', fontSize: 14},
    textStyle: {color: '#FFF', fontSize: 14}
  },
  series: {
    0: { targetAxisIndex: 1 }
  },
  backgroundColor: {
    fill: 'transparent'
  },
  legend: 'none',
  tooltip: {
    textStyle: {color: '#3BF4FB', fontSize: 14}
  },
  titleTextStyle: {color: '#3BF4FB', fontSize: 12},
  colors:['#3BF4FB', '#004411']
};

var chart = new google.visualization.LineChart(document.getElementById('curve_chart'));
chart.draw(data, options);
```

Рисунок 3.5 – Конфігурація та побудова графіка

Дана функція використовує інструменти бібліотеки Google Charts для побудови лінійного горизонтального графіку, віссю Y якого є ціна криптовалюти, а віссю X – дата. Додатково лінія графіку має візуальне розподілення на дві частини – суцільну та пунктирну. Метою цього є покращення користувацького досвіду при перегляді графіку цін, адже таке рішення значно полегшує сприйняття та дає змогу відрізнити історичні та прогнозовані дані.

3.4 Висновки

Отже, у третьому розділі було проведено варіантний аналіз мов програмування Java, C# та C++ і здійснено їх порівняння за шістьма ключовими критеріями. За підсумками цього порівняння було обрано мову Java.

Окрім цього, було проведено варіантний аналіз інтегрованих середовищ розробки та обґрунтовано вибір IntelliJ IDEA для здійснення розробки програмного модуля.

З використанням обраних інструментів було визначено етапи процесу розробки програмного модуля та виконано кожен з них, що було задокументовано. Також було описано процес створення інтерфейсу користувача.

4 ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

4.1 Аналіз методів тестування

Процес тестування є критично важливим етапом розробки програмного забезпечення, оскільки тестування забезпечує виявлення помилок, забезпечення якості програмного продукту і задоволення користувацьких потреб. Завдяки ретельному тестуванню можна зменшити ризик негативного досвіду користувачів і покращити репутацію програмного продукту на ринку, що підвищує рівень задоволеності клієнтів і загальну лояльність. Тестування також сприяє ефективнішій взаємодії між розробниками та користувачами, що в свою чергу допомагає у визначенні пріоритетів розвитку продукту.

Тестування програмного забезпечення відбувається у різних формах і включає численні типи тестів, кожен з яких призначений для досягнення певної мети. На рисунку 4.1 представлено ієрархічну структуру основних типів тестування, яка дозволяє чітко бачити взаємозв'язок між різними типами і формами тестів.

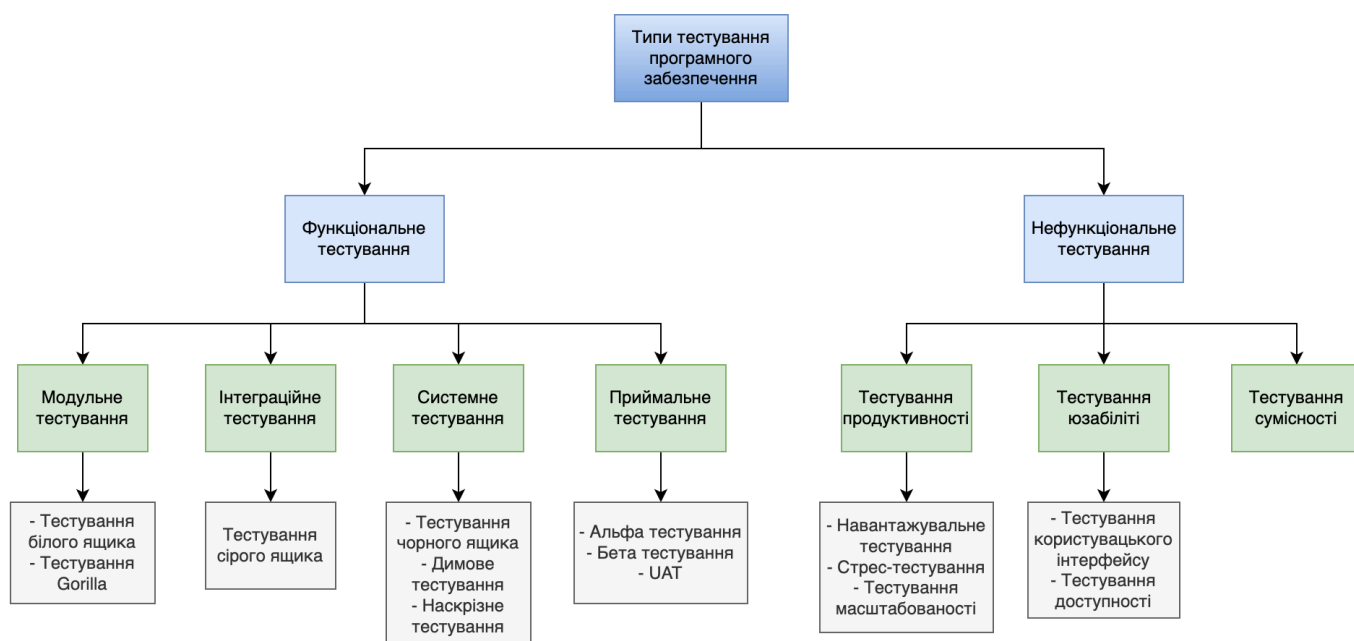


Рисунок 4.1 – Ієрархія типів тестування

Тестування поділяється на дві основні категорії: функціональне та нефункціональне тестування. Функціональне тестування спрямоване на перевірку того, як програмне забезпечення виконує свої основні функції, чи відповідає воно технічним та бізнес-вимогам. Нефункціональне тестування зосереджене на таких параметрах, як продуктивність, безпека та зручність використання, що впливають на загальний досвід користувачів. Розглянемо більш детально кожен із наведених методів тестування [28].

1. Тестування білого ящика. Це підхід, що дозволяє тестувальнику мати повний доступ до внутрішньої структури та вихідного коду програми. Завдяки цьому методу тестувальник може оцінювати функції, оптимізувати ефективність алгоритмів та виявляти помилки на рівні коду, що дозволяє виконувати більш глибоке тестування. Такий метод особливо підходить для виявлення прихованих дефектів, що можуть виникати в умовах специфічного кодування або розгалуження логіки.

2. Тестування чорного ящика. На відміну від білого ящика, цей метод не передбачає доступу до внутрішньої структури коду, а зосереджений на поведінці програми з точки зору кінцевого користувача. Тестувальники взаємодіють із програмою, базуючись лише на вихідних і вхідних даних, і можуть виявляти помилки, пов'язані з неправильними виведеннями, функціональністю або інтерфейсом програми. Це дозволяє оцінювати продукт з перспективи користувача, ідентифікуючи дефекти, які можуть вплинути на зручність і якість досвіду.

3. Тестування сірого ящика. Цей метод поєднує елементи тестування білого та чорного ящика, надаючи тестувальнику частковий доступ до внутрішньої інформації про структуру або специфікації програми. Це дозволяє проводити тести на основі знань про специфікації системи, але без повного доступу до вихідного коду. Такий метод застосовується для аналізу інтеграції підсистем і дозволяє краще розуміти роботу продукту, зокрема щодо надійності функцій і сумісності компонентів.

4. Димове тестування. Проводиться для швидкої перевірки базової працездатності програми після змін, таких як оновлення чи корекція помилок. Це первинний етап тестування, який виявляє грубі дефекти та підтверджує готовність до

більш глибокого тестування. Він дозволяє визначити, чи працюють основні функції, перш ніж переходити до більш складних тестів. Часто використовується для скорочення часу тестування, так як дозволяє швидко виявити критичні проблеми.

5. Приймальне користувацьке тестування (UAT). Цей тип тестування виконується кінцевими користувачами або клієнтами для перевірки того, чи відповідає продукт бізнес-вимогам і потребам користувачів. UAT зазвичай є завершальним етапом тестування перед впровадженням продукту на ринок. Воно зосереджене на оцінці зручності використання, а також ефективності виконання ключових функцій, що особливо важливо для забезпечення відповідності очікуванням кінцевих користувачів.

6. Навантажувальне тестування. Це тестування дозволяє визначити, як система функціонує під час підвищеного навантаження, наприклад, при одночасному великому обсязі запитів або значній кількості користувачів. Воно дозволяє ідентифікувати вузькі місця в продуктивності програми і виявити проблеми, що можуть виникати під навантаженням. Завдяки цьому тестуванню розробники можуть оптимізувати додаток для забезпечення його надійної роботи в умовах реального використання.

7. Тестування користувацького інтерфейсу (UI). Охоплює оцінку візуальних та інтерактивних аспектів програми, таких як розміщення елементів інтерфейсу, реакція на введення, узгодженість дизайну і зручність користування. Це тестування гарантує, що програмний інтерфейс відповідає вимогам щодо ергономіки і є зручним для користувачів. UI-тестування дозволяє забезпечити інтуїтивну взаємодію з продуктом і усунути можливі проблеми, що можуть знизити користувацьке задоволення.

8. Тестування доступності. Цей вид тестування зосереджений на забезпеченні доступності програмного забезпечення для людей з обмеженими можливостями. Перевіряються такі аспекти, як зручність читання для користувачів з вадами зору, можливість використання без миші для людей із фізичними обмеженнями, а також правильний підбір кольорів для користувачів із порушенням

кольоросприйняття. Це тестування допомагає зробити програму більш інклюзивною та підвищити її доступність для різних категорій користувачів.

9. Тестування сумісності. Це метод тестування, що оцінює працездатність програми в різних середовищах, на різних пристроях, операційних системах і браузерях. Його мета – забезпечити стабільну роботу програми незалежно від платформи чи конфігурації. Сумісність є важливим фактором, особливо для програмного забезпечення, яке має бути доступним на різних платформах, і її забезпечення дозволяє розширити аудиторію користувачів.

Таким чином, наведені види тестування програмного забезпечення охоплюють ключові аспекти його якості та функціональності, дозволяючи виявляти можливі дефекти, підвищувати зручність використання та забезпечувати надійність у різних умовах експлуатації. На основі проведеного аналізу, для тестування розробленого програмного продукту було вирішено використовувати тестування чорного ящика, димове тестування, приймальне тестування користувача, UI-тестування та тестування сумісності для всебічної оцінки його ефективності та зручності.

4.2 Тестування програми

Почнемо з перевірки основних сторінок веб-сайту для оцінки їх функціональних можливостей. На рисунку 4.2 представлена домашня сторінка створеного веб-сайту "Crypto Advisor".

У верхній частині сторінки розміщується навігаційна панель. За її допомогою можна перейти на головну сторінку, на домашню сторінку, на сторінку з інформацією про веб-сайт та на сторінку з контактами розробників.

Домашня сторінка містить кілька важливих елементів: таблицю з актуальними даними про найбільш популярні криптовалюти, з можливістю переходу на сторінку кожної з них, а також віджети "Trending", "Stable" і "Deviating". Ці віджети показують трійки найбільш зростаючих, стабільних та падаючих криптовалют відповідно. Вся інформація відображається коректно.

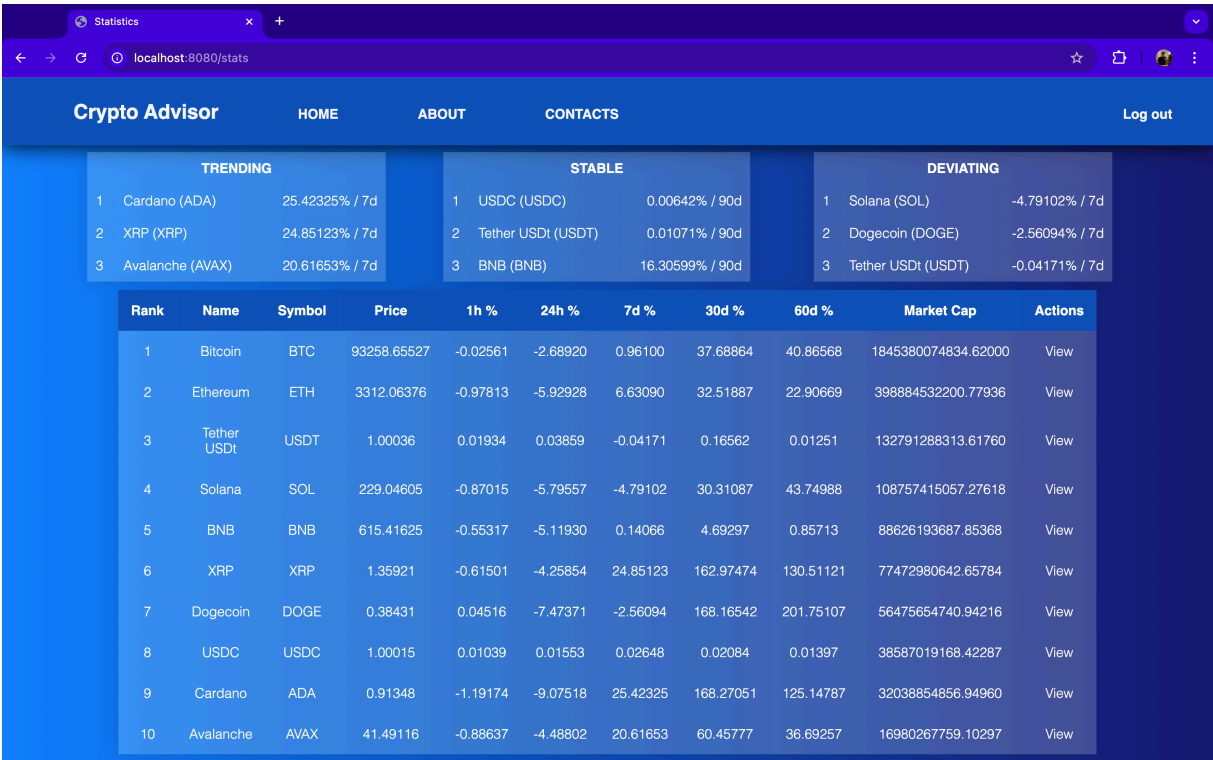


Рисунок 4.2 – Домашня сторінка

Наступним кроком розглянемо сторінку окремої криптовалюти, зображену на рисунку 4.3, де демонструється функціонал прогнозування і побудови графіків.



Рисунок 4.3 – Сторінка індивідуальної криптовалюти

Графік, побудований на основі історичних та прогнозованих даних обраної криптовалюти, відображається належним чином. Тип лінії дозволяє розрізнити дані: суцільна лінія представляє історичні дані, а пунктирна — прогнозовані.

Додатково, користувач може навести курсор на будь-яку точку графіка для перегляду точної дати та ціни криптовалюти. Приклад перегляду історичних даних наведено на рисунку 4.4.



Рисунок 4.4 – Історичні дані в конкретній точці

Натомість рисунок 4.5 демонструє приклад перегляду прогнозованих даних обраної криптовалюти.

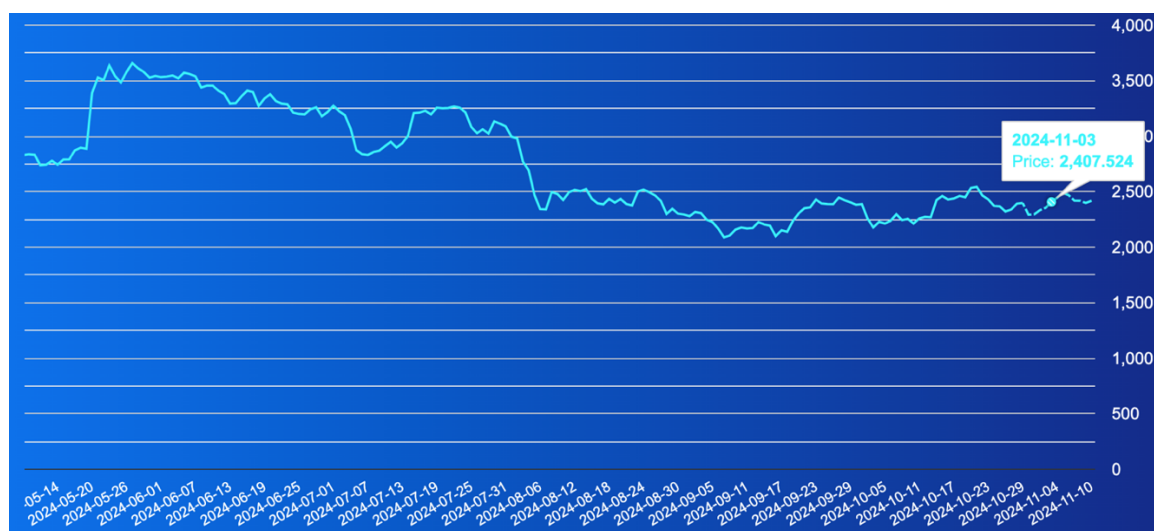
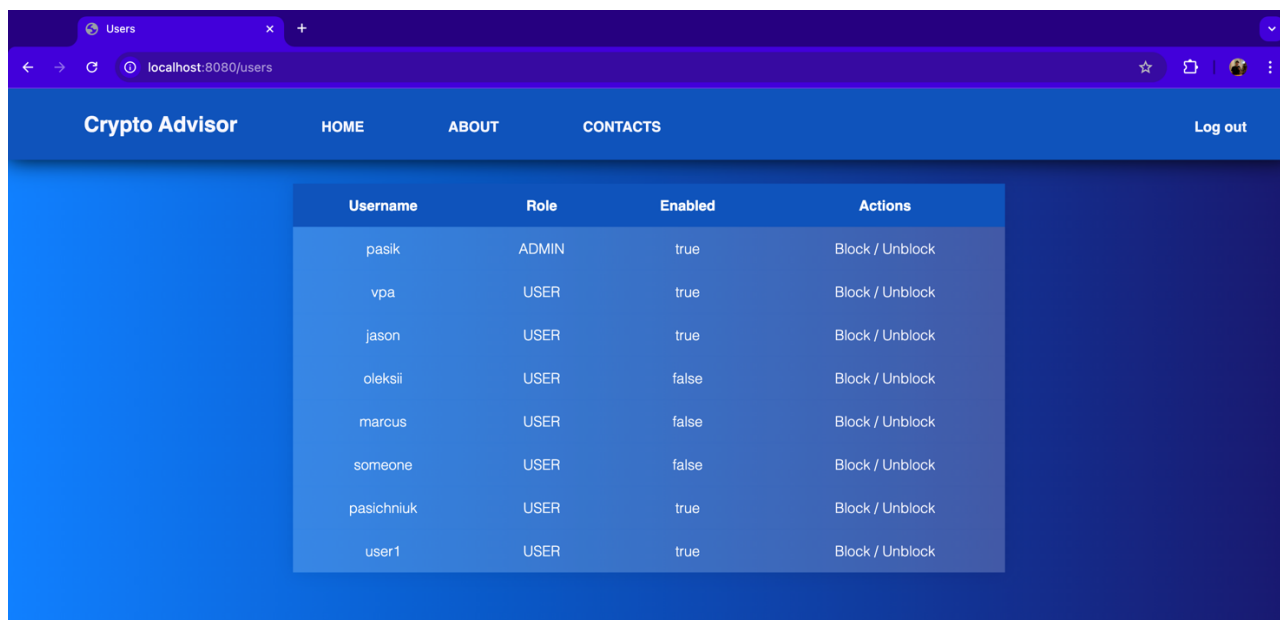


Рисунок 4.5 – Прогнозовані дані в конкретній точці

Якщо користувач має права адміністратора, у нього з'являється доступ до сторінки з інформацією про користувачів. Навпроти кожного користувача у списку є дві кнопки, які дають змогу заблокувати або розблокувати певного користувача. Поточний стан блокування користувача відображується в колонці “Enabled” (рис. 4.6).



Username	Role	Enabled	Actions
pasik	ADMIN	true	Block / Unblock
vpa	USER	true	Block / Unblock
jason	USER	true	Block / Unblock
oleksii	USER	false	Block / Unblock
marcus	USER	false	Block / Unblock
someone	USER	false	Block / Unblock
pasichniuk	USER	true	Block / Unblock
user1	USER	true	Block / Unblock

Рисунок 4.6 – Сторінка з інформацією про користувачів

У разі переходу на сторінку неіснуючої криптовалюти користувач отримує відповідне повідомлення про неможливість її знайти (рис. 4.7).

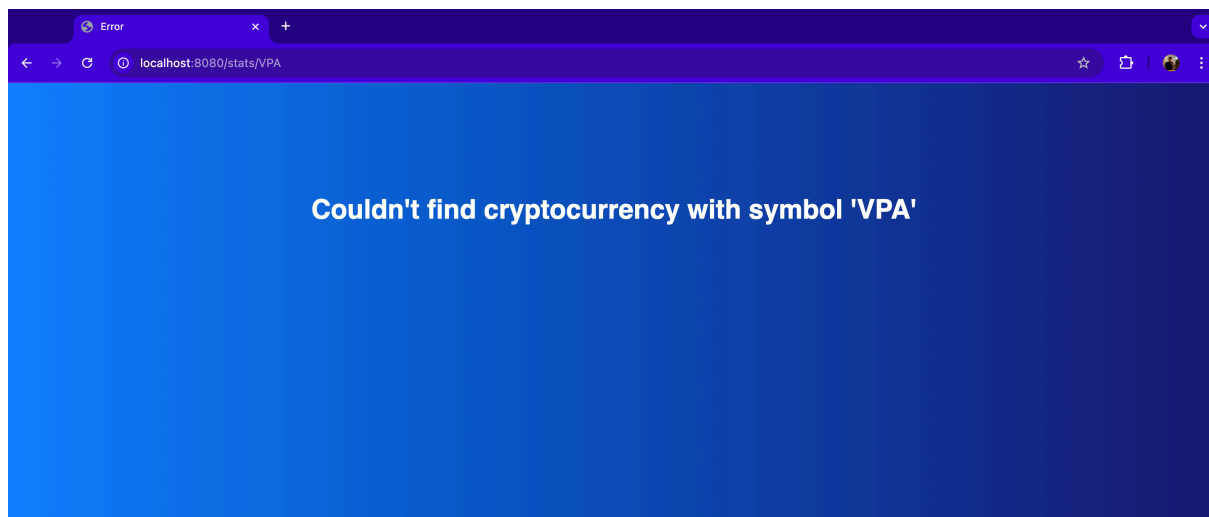


Рисунок 4.7 – Сторінка неіснуючої криптовалюти

Якщо ж здійснити перехід на сторінку, яка не існує в межах програми, або виникне непередбачена помилка, користувача буде перенаправлено на сторінку з повідомленням про помилку (рис. 4.8).

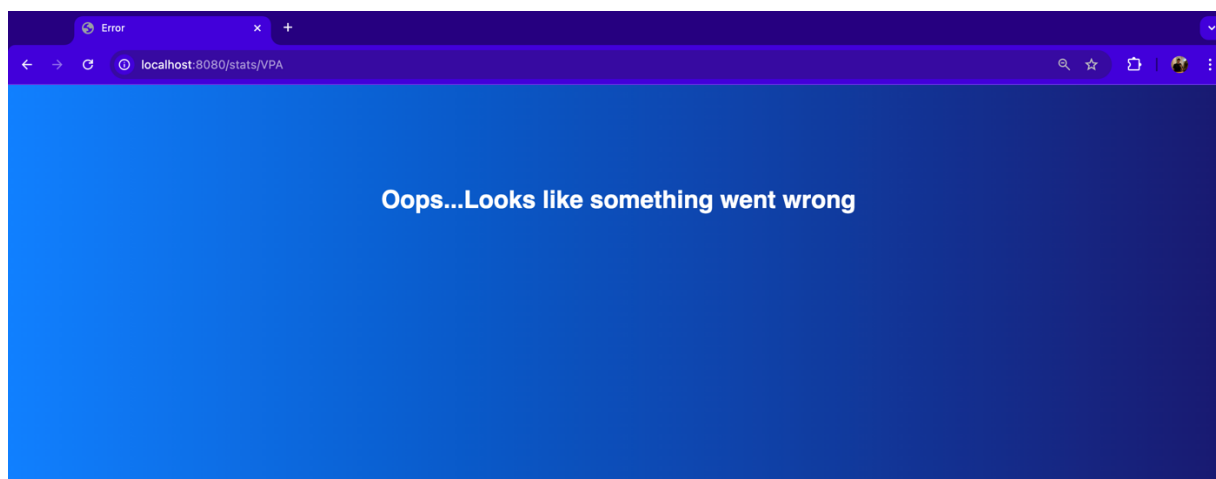


Рисунок 4.8 – Сторінка помилки

Далі проведемо перевірку сумісності створеного веб-сайту з найвідомішими сучасними браузерами: Google Chrome, Safari та Mozilla Firefox. Для перевірки роботи в Safari переходимо на сторінку з актуальними даними криптовалют (рис. 4.9).

Rank	Name	Symbol	Price	1h %	24h %	7d %	30d %	60d %	Market Cap	Actions
1	Bitcoin	BTC	93317.74811	0.22079	-2.88278	1.01018	37.82166	41.23017	1846549389822.27780	View
2	Ethereum	ETH	3314.22414	-0.60426	-5.74415	6.42715	32.67302	23.20987	399144715593.81130	View
3	Tether USDt	USDT	1.00030	-0.06673	0.00528	-0.05047	0.14109	0.00473	132782592480.59076	View
4	Solana	SOL	229.00865	-0.87443	-5.86900	-5.17402	30.58235	44.13294	108739657941.54814	View
5	BNB	BNB	614.91020	-0.54419	-5.55056	-0.02038	4.64795	0.87715	88553317107.51587	View
6	XRP	XRP	1.36005	-0.08873	-5.04561	23.81751	163.34110	130.79582	77520859165.67090	View
7	Dogecoin	DOGE	0.38481	0.66906	-7.86896	-2.61531	169.15524	203.31894	56548228431.09141	View
8	USDC	USDC	1.00007	-0.05375	-0.01547	0.00800	-0.00539	0.00433	38583967846.98464	View
9	Cardano	ADA	0.91517	-0.50887	-8.56463	25.44718	168.31408	125.83669	32098305633.38226	View
10	Avalanche	AVAX	41.51970	-0.71738	-4.14659	20.42412	60.62659	37.32925	16991949185.37386	View

Рисунок 4.9 – Домашня сторінка, відкрита у браузері Safari

У браузері Mozilla Firefox таким самим чином перевіримо коректність роботи створеного веб-сайту, переглянувши сторінку обраної криптовалюти (рисунок 4.10).

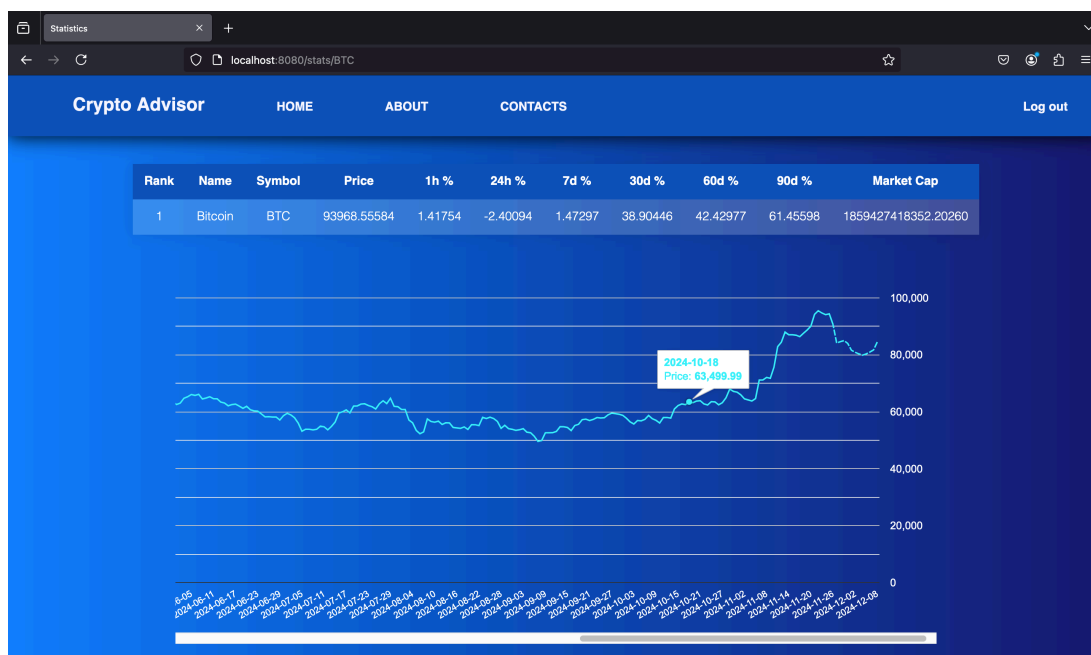


Рисунок 4.10 – Сторінка індивідуальної криптовалюти, відкрита у Mozilla Firefox

Оскільки все тестування функціоналу попередньо проводилося в Google Chrome, можна стверджувати, що розроблений веб-сайт повністю сумісний і з ним.

Таким чином, тестування підтвердило, що додаток однаково стабільно функціонує в Chrome, Safari та Firefox.

Також сервіс має захист від надмірної кількості запитів від одного користувача. При великій кількості запитів за короткий проміжок часу сервер захищає себе від перевантаження, повертаючи помилку "HTTP 429 Too Many Requests" (рис. 4.11).

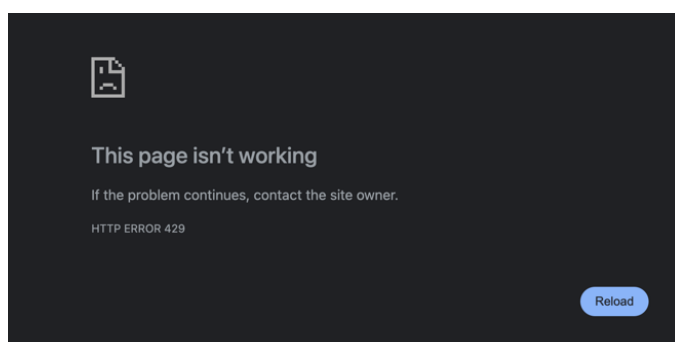
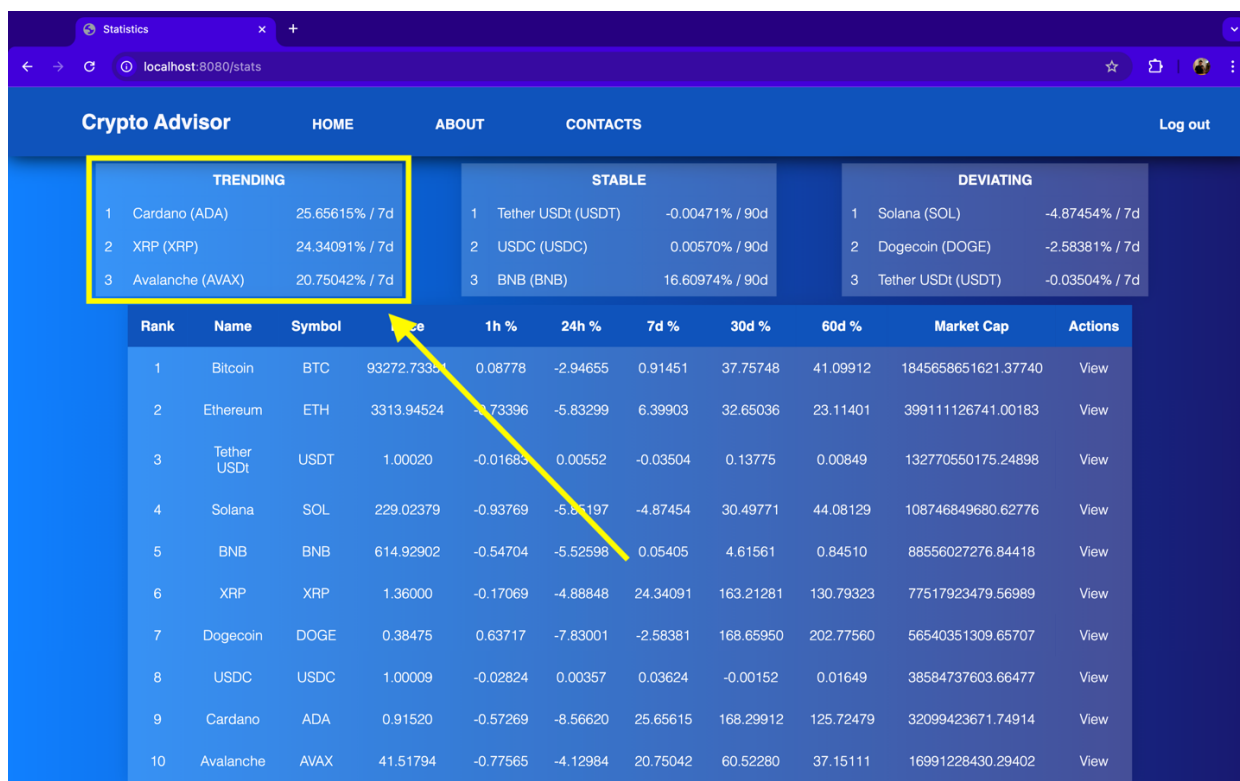


Рисунок 4.11 – Сторінка помилки з кодом 429

На домашній сторінці представлено три віджети: "Trending", "Stable" та "Deviating". Перевіримо правильність роботи кожного з них.

На рисунку 4.12 продемонстровано віджет "Trending", який містить список з трьох криптовалют із найкращими показниками росту за останній тиждень.



Crypto Advisor										
HOME ABOUT CONTACTS Log out										
TRENDING										
1	Cardano (ADA)	25.65615% / 7d								
2	XRP (XRP)	24.34091% / 7d								
3	Avalanche (AVAX)	20.75042% / 7d								
STABLE										
1	Tether USDt (USDT)	-0.00471% / 90d								
2	USDC (USDC)	0.00570% / 90d								
3	BNB (BNB)	16.60974% / 90d								
DEVIATING										
1	Solana (SOL)	-4.87454% / 7d								
2	Dogecoin (DOGE)	-2.58381% / 7d								
3	Tether USDt (USDT)	-0.03504% / 7d								
Rank	Name	Symbol	Price	1h %	24h %	7d %	30d %	60d %	Market Cap	Actions
1	Bitcoin	BTC	93272.7335	0.08778	-2.94655	0.91451	37.75748	41.09912	1845658651621.37740	View
2	Ethereum	ETH	3313.94524	-0.73396	-5.83299	6.39903	32.65036	23.11401	399111126741.00183	View
3	Tether USDt	USDT	1.00020	-0.01683	0.00552	-0.03504	0.13775	0.00849	132770550175.24898	View
4	Solana	SOL	229.02379	-0.93769	-5.85197	-4.87454	30.49771	44.08129	108746849680.62776	View
5	BNB	BNB	614.92902	-0.54704	-5.52598	0.05405	4.61561	0.84510	88556027276.84418	View
6	XRP	XRP	1.36000	-0.17069	-4.88848	24.34091	163.21281	130.79323	77517923479.56989	View
7	Dogecoin	DOGE	0.38475	0.63717	-7.83001	-2.58381	168.65950	202.77560	56540351309.65707	View
8	USDC	USDC	1.00009	-0.02824	0.00357	0.03624	-0.00152	0.01649	38584737603.66477	View
9	Cardano	ADA	0.91520	-0.57269	-8.56620	25.65615	168.29912	125.72479	32099423671.74914	View
10	Avalanche	AVAX	41.51794	-0.77565	-4.12984	20.75042	60.52280	37.15111	16991228430.29402	View

Рисунок 4.12 – Віджет "Trending"

Згідно з даними колонки "7d %" таблиці актуальних криптовалют, до трійки найбільш зростаючих криптовалют потрапили DOGE, BTC і SOL, що співпадає з інформацією, відображеною у віджеті.

Наступний віджет, під назвою "Stable", зображений на рисунку 4.13. Він демонструє три найстабільніші криптовалюти, відповідно до статистики за попередні три місяці.

Як видно з рисунку, перші дві лідерські позиції у віджеті зайняли так звані "stablecoins": USDT та USDC. Активи цього типу мають стабільну вартість і, як правило, грають роль цифрової валюти, за допомогою якої купують інші активи. Це свідчить про коректність роботи віджета.

Statistics

+

localhost:8080/stats

Log out

TRENDING

1

Cardano (ADA)

25.65615% / 7d

2

XRP (XRP)

24.34091% / 7d

3

Avalanche (AVAX)

20.75042% / 7d

STABLE

1

Tether USDt (USDT)

-0.00471% / 90d

2

USDC (USDC)

0.00570% / 90d

3

BNB (BNB)

16.60974% / 90d

DEVIATING

1

Solana (SOL)

-4.87454% / 7d

2

Dogecoin (DOGE)

-2.58381% / 7d

3

Tether USDt (USDT)

-0.03504% / 7d

Rank	Name	Symbol	Price	1h %	24h %	7d %	30d %	60d %	Market Cap	Actions
1	Bitcoin	BTC	93272.73351	0.08778	-2.94655	0.91451	37.75748	41.09912	1845658651621.37740	View
2	Ethereum	ETH	3313.94524	-0.73396	-5.83299	6.39903	32.65036	23.11401	399111126741.00183	View
3	Tether USDt	USDT	1.00020	-0.01683	0.00552	-0.03504	0.13775	0.00849	132770550175.24898	View
4	Solana	SOL	229.02379	-0.93769	-5.85197	-4.87454	30.49771	44.08129	108746849680.62776	View
5	BNB	BNB	614.92902	-0.54704	-5.52598	0.05405	4.61561	0.84510	88556027276.84418	View
6	XRP	XRP	1.36000	-0.17069	-4.88848	24.34091	163.21281	130.79323	77517923479.56989	View
7	Dogecoin	DOGE	0.38475	0.63717	-7.83001	-2.58381	168.65950	202.77560	56540351309.65707	View
8	USDC	USDC	1.00009	-0.02824	0.00357	0.03624	-0.00152	0.01649	38584737603.66477	View
9	Cardano	ADA	0.91520	-0.57269	-8.56620	25.65615	168.29912	125.72479	32099423671.74914	View
10	Avalanche	AVAX	41.51794	-0.77565	-4.12984	20.75042	60.52280	37.15111	16991228430.29402	View

Рисунок 4.13 – Віджет “Stable”

Віджет "Deviating", зображений на рисунку 4.14, відповідає за відображення трьох криптовалют, що продемонстрували найгірші показники за останні 7 днів.

Statistics

localhost:8080/stats

Crypto Advisor

HOMEABOUTCONTACTS

Log out

TRENDING

1Cardano (ADA)25.65615% / 7d

2XRP (XRP)24.34091% / 7d

3Avalanche (AVAX)20.75042% / 7d

STABLE

1Tether USDt (USDT)-0.00471% / 90d

2USDC (USDC)0.00570% / 90d

3BNB (BNB)16.60974% / 90d

DEVIATING

1Solana (SOL)-4.87454% / 7d

2Dogecoin (DOGE)-2.58381% / 7d

3Tether USDt (USDT)-0.03504% / 7d

Rank	Name	Symbol	Price	1h %	24h %	7d %	30d %	60d %	Market Cap	Actions
1	Bitcoin	BTC	93272.73351	0.08778	-2.94655	0.91451	37.75748	41.09912	1845658651621.37740	View
2	Ethereum	ETH	3313.94524	-0.73396	-5.83299	6.39903	32.65036	23.11401	399111126741.00183	View
3	Tether USDt	USDT	1.00020	-0.01683	0.00552	-0.03504	0.13775	0.00849	132770550175.24898	View
4	Solana	SOL	229.02379	-0.93769	-5.85197	-4.87454	30.49771	44.08129	108746849680.62776	View
5	BNB	BNB	614.92902	-0.54704	-5.52598	0.05405	4.61561	0.84510	88556027276.84418	View
6	XRP	XRP	1.36000	-0.17069	-4.88848	24.34091	163.21281	130.79323	77517923479.56989	View
7	Dogecoin	DOGE	0.38475	0.63717	-7.83001	-2.58381	168.65950	202.77560	56540351309.65707	View
8	USDC	USDC	1.00009	-0.02824	0.00357	0.03624	-0.00152	0.01649	38584737603.66477	View
9	Cardano	ADA	0.91520	-0.57269	-8.56620	25.65615	168.29912	125.72479	32099423671.74914	View
10	Avalanche	AVAX	41.51794	-0.77565	-4.12984	20.75042	60.52280	37.15111	16991228430.29402	View

Рисунок 4.14 – Віджет “Deviating”

Здійснивши додатковий аналіз даних з таблиці актуальних даних криптовалют, було отримано список з трьох криптовалют, що мали найменш втішні показники росту ціни за останній тиждень: TON, XRP та USDC. Цей список повністю співпадає з даними віджета, підтверджуючи його правильну роботу.

Наступним кроком перевіримо наявність можливості зв'язатися з розробниками веб-сервісу та надати зворотній зв'язок. Для цього перейдемо на сторінку “Contacts”, зображену на рисунку 4.27, та скопіюємо надану email-адресу. Перейдемо у додаток Gmail і створимо новий електронний лист (рис. 4.15).

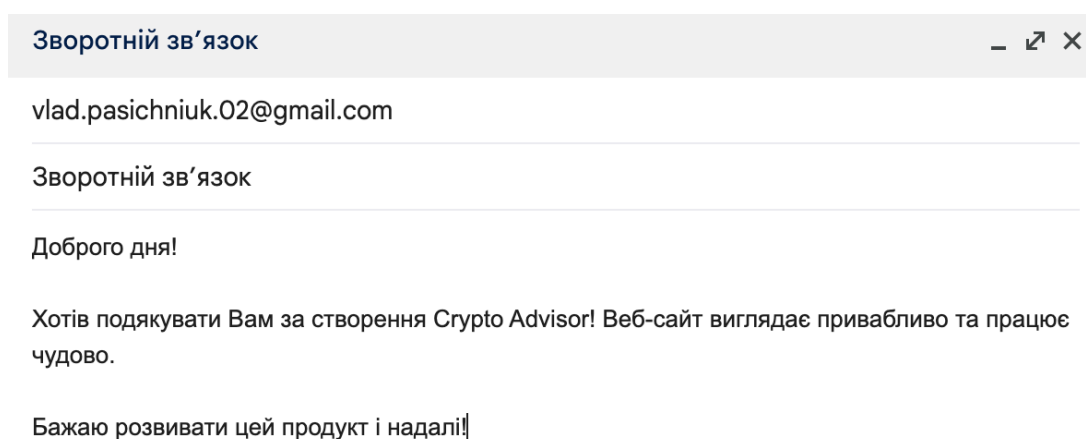


Рисунок 4.15 – Лист до розробників

Через невеликий проміжок часу після надсилання листа, отримуємо відповідь (рис. 4.16)

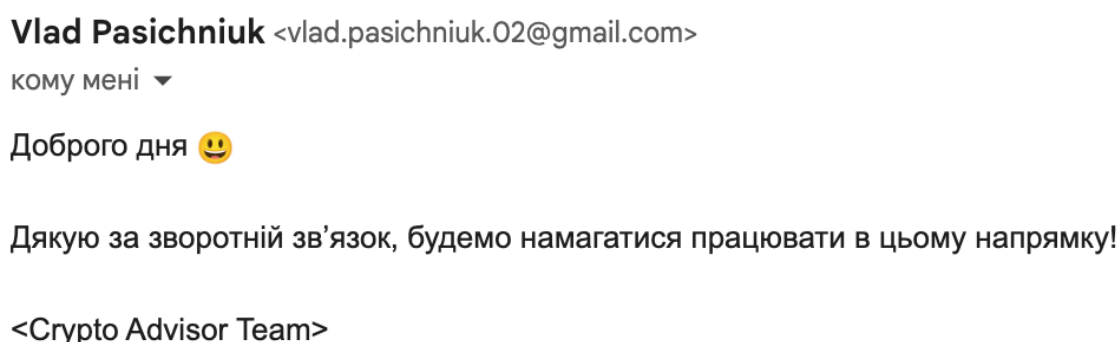


Рисунок 4.16 – Відповідь від розробників

Крім поштової скриньки, зв'язатися із командою розробників можна за допомогою відомого месенджера Telegram, де потрібно у пошуковій стрічці ввести ім'я користувача, наданого на сторінці контактів, та обрати його зі списку, після чого можна розпочати особисте листування.

Далі проведемо тестування швидкості виконання запитів до веб-сервісу, надісланих користувачем через інтерфейс за допомогою веб-браузера. Для цього скористаємося інструментом під назвою “Developer Tools”.

Developer Tools – це інструмент, що є вбудованим у кожен із сучасних веб-браузерів, призначений для аналізу, діагностики та налагодження веб-додатків. Він забезпечує доступ до широкого спектра функцій, які допомагають розробникам оцінювати продуктивність, тестувати функціональність і відстежувати проблеми у веб-інтерфейсах. Однією з ключових вкладок цього інструмента є вкладка Network, яка спеціалізується на моніторингу та аналізі мережевої активності веб-сторінки.

Вкладка Network дозволяє в реальному часі спостерігати за всіма запитами, що відправляються до сервера з веб-браузера, і отримувати детальну інформацію про кожен із них. Це включає тип запиту (GET, POST тощо), адресу кінцевої точки, статус відповіді, обсяг переданих даних і, що найважливіше у нашій задачі, час, витрачений на виконання запиту. Час виконання детально розбивається на кілька складових:

1. DNS Lookup – час, потрібний для перетворення доменного імені в IP-адресу.
2. Initial Connection – час встановлення TCP-з'єднання.
3. Waiting (TTFB) – час до отримання першого байта відповіді від сервера.
4. Content Download – час завантаження отриманих даних.

Для тестування швидкості виконання запитів до створеного веб-сервісу з використанням інструмента Developer Tools необхідно:

1. Відкрити веб-сторінку Crypto Advisor у браузері.
2. Відкрити Developer Tools (за клавіші F12 або комбінацій Ctrl+Shift+I / Cmd+Option+I).
3. Перейти на вкладку Network.

4. Виконати потрібні дії в інтерфейсі, які генерують запити до сервера (наприклад, заповнення форм, натискання кнопок).

5. Аналізувати кожен запит у списку, переглядаючи час його виконання та інші деталі.

Спершу визначимо швидкість виконання запиту, що надсилається при логіні користувача (рис. 4.17).

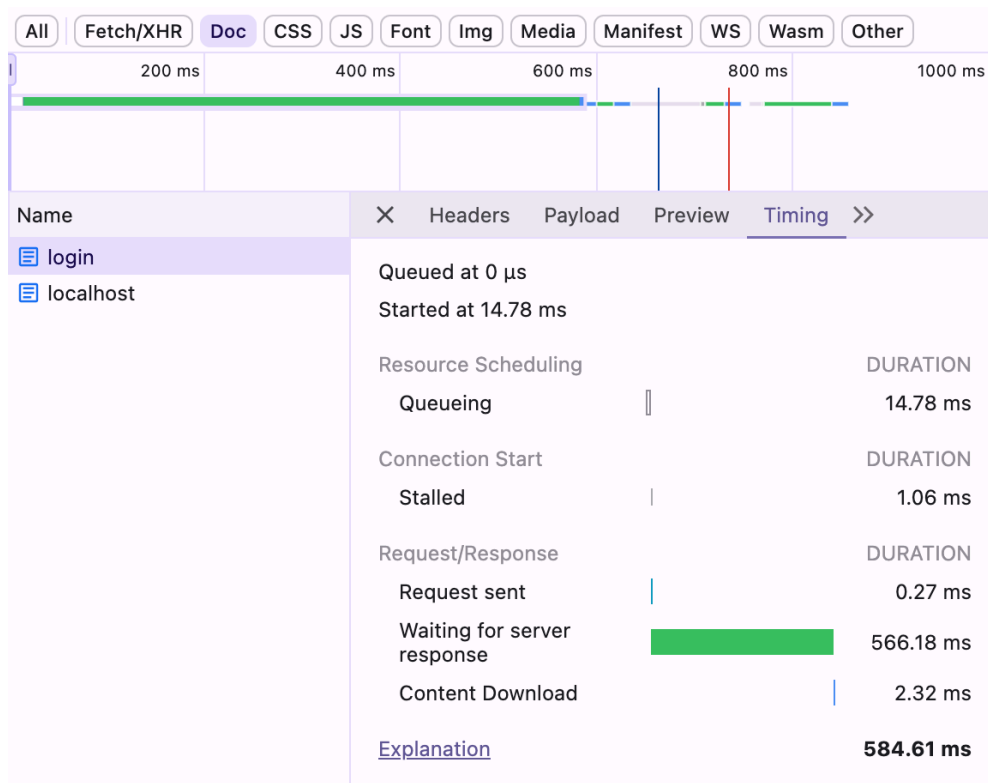


Рисунок 4.17 – Час виконання запиту /login

Як видно зі вкладки “Timing”, час виконання запиту на логін користувача склав 584.6 мс. Наступний запит надсилаємо на головну сторінку веб-сайту, де відображаються актуальні дані криптовалют у вигляді таблиці, а також три віджети з інформацією про найбільш зростаючі, найбільш стабільні та найбільш спадаючі в ціні криптовалюти (рис. 4.18).

Середнім показником часу виконання цього запиту є 1,5 секунди, що можна вважати чудовим результатом, особливо враховуючи кількість даних, що отримуються.

Resource Scheduling		DURATION
Queueing		7.34 ms
Connection Start		DURATION
Stalled		17.50 ms
Request/Response		DURATION
Request sent		12.55 ms
Waiting for server response		1.50 s
Content Download		3.26 ms
Explanation		1.54 s

Рисунок 4.18 – Час виконання запиту /stats

Третім запитом, що було використано для перевірки швидкості виконання, обрано запит на отримання даних обраної криптовалюти, тобто перехід на її окрему сторінку. Даний запит є найбільш навантаженим, оскільки крім діставання актуальних даних про криптовалюту потребує ще й історичні дані, які у ході виконання запиту обробляються та передаються у нейромережу, що прогнозує масив потенційних майбутніх цін. Результат перевірки наведено на рисунку 4.19.

Resource Scheduling		DURATION
Queueing		18.83 ms
Connection Start		DURATION
Stalled		2.41 ms
Request/Response		DURATION
Request sent		0.26 ms
Waiting for server response		3.36 s
Content Download		109.79 ms
Explanation		3.49 s

Рисунок 4.19 – Час виконання запиту /stats/ETH

Як бачимо, загальний час виконання запиту склав 3,5 секунди. Для такої великої кількості прихованих розрахунків та розміру контенту, цей результат можна вважати більш ніж прийнятним [29].

У результаті проведення тестування швидкості роботи створеного веб-сервісу було доведено, що час виконання запитів є відмінним у більшості випадків оскільки не перевершує 2 секунд. Найбільш повільним є запит на отримання актуальних, історичних та прогнозованих даних криптовалют, який в середньому займає близько 3,5 секунд – проте, це не можна вважати критичним недоліком.

4.3 Порівняльний аналіз точності прогнозування

Для оцінки точності роботи модуля прогнозування було проведено тестування, що передбачає розподіл історичних даних на дві частини: 90% використано для навчання моделі нейронної мережі, а 10% залишено для тестування. Такий підхід забезпечив можливість навчання моделі на значному обсязі даних та створив умови для перевірки точності прогнозування на основі відокремленої тестової частини.

Для порівняльного аналізу точності прогнозування використано два підходи: класичну LSTM модель і гібридну модель LSTM + CNN. Класична LSTM модель була обрана через здатність обробляти послідовні дані та враховувати довготривалі залежності, що є важливим у прогнозуванні цін криптовалюти. Гібридна модель LSTM + CNN, в свою чергу, оптимізована для виявлення не тільки довготривалих, але й короткотермінових шаблонів у даних, що дає змогу підвищити передбачувальні можливості модуля та досягти більшої точності.

З метою покращення сприйняття тестових даних під час проведення аналізу було вирішено використати засоби бібліотеки JavaFX для побудови графіків, які міститимуть історичні та прогнозовані дані у вигляді окремих ліній. Вісь Y відповідає за ціну криптовалюти, а вісь X – за порядковий номер ціни в масиві тестових даних.

На першому графіку (рисунок 4.20), відображено прогнозовані значення класичної LSTM моделі. Синя лінія відповідає реальним історичним даним, а червона – прогнозованим. Аналіз показав, що лінія прогнозованих значень LSTM має суттєві відхилення від реальних значень, а патерн її поведінки відрізняється від історичних даних. Ці відмінності свідчать про обмеження звичайної LSTM моделі у точному прогнозуванні складних коливань ціни криптовалюти.

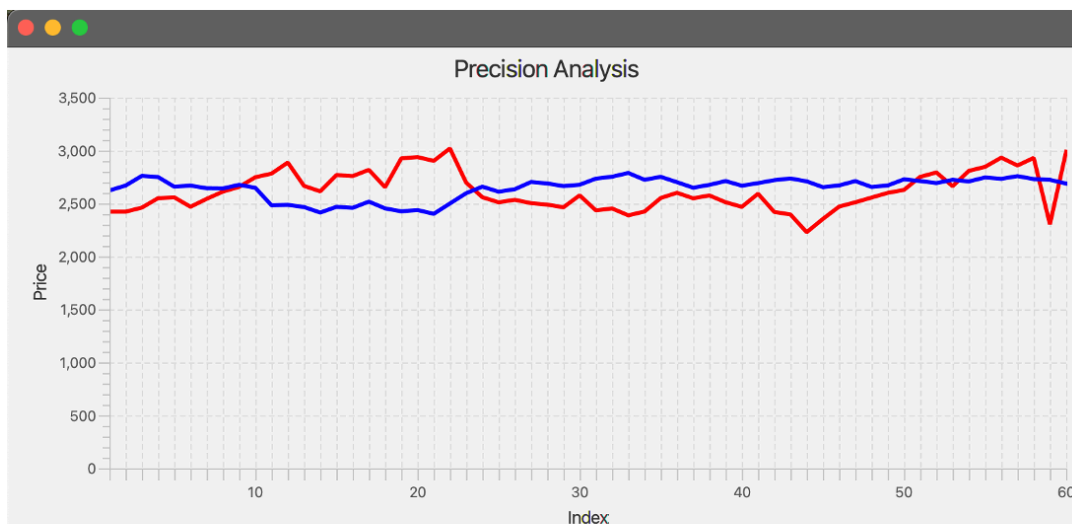


Рисунок 4.20 – Порівняння реальних даних та даних, прогнозованих звичайною LSTM моделлю

На наступному графіку (рисунок 4.21) відображено результати прогнозування за допомогою гібридної моделі LSTM + CNN. Помітно, що прогнозовані значення майже повністю повторюють динаміку синьої лінії. Червона лінія прогнозованих значень тут значно ближче до реальних даних, мінімізуючи відхилення та більш точно слідуючи шаблонам, присутнім у часовому ряді. Це свідчить про високу здатність гібридної моделі до захоплення як довготривалих, так і короткотермінових залежностей, що покращує якість прогнозування.

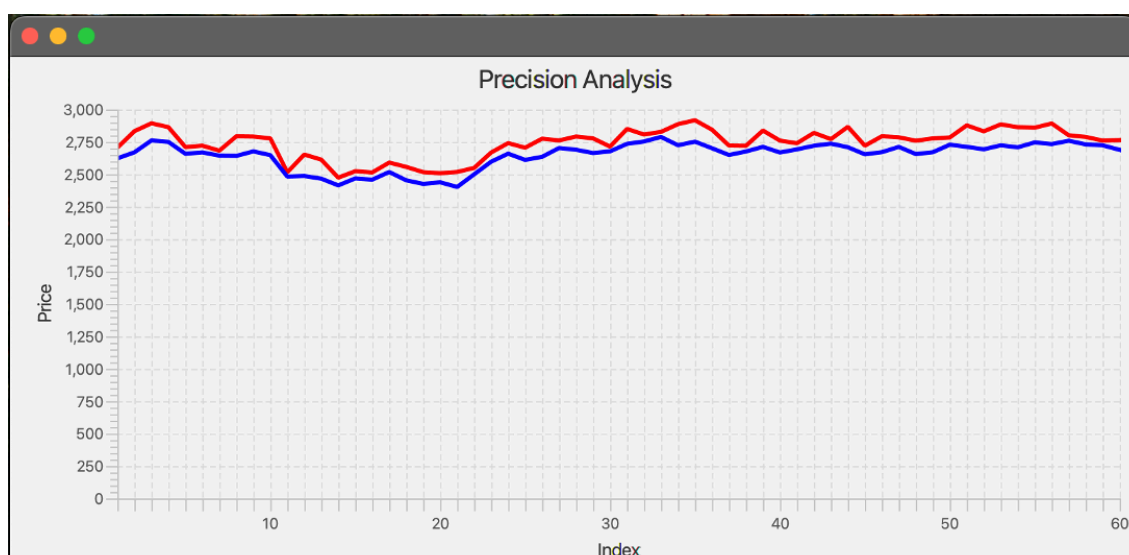


Рисунок 4.21 – Порівняння реальних даних та даних, прогнозованих гібридною моделлю

Таким чином, результати порівняльного аналізу підтверджують, що гібридна модель LSTM + CNN забезпечує значно вищу точність прогнозування, уникаючи суттєвих відхилень від реальних значень, які спостерігалися у класичній LSTM моделі. Це підтверджує, що використання гібридної моделі не лише покращує точність, а й підвищує стабільність та надійність прогнозування, що відповідає технічним вимогам додатку та поставленому завданню.

4.4 Опис основних функціональних можливостей користувача

Розглянемо основні можливості користувача створеного веб-сервісу для забезпечення правильного використання програмного продукту та надання детального опису існуючого функціоналу.

Для отримання доступу до веб-сервісу необхідно використовувати сучасний веб-браузер, такий як Google Chrome, Mozilla Firefox, Microsoft Edge, Safari або Opera. Після цього потрібно перейти на головну сторінку веб-сайту “Crypto Advisor” (рисунок 4.22).

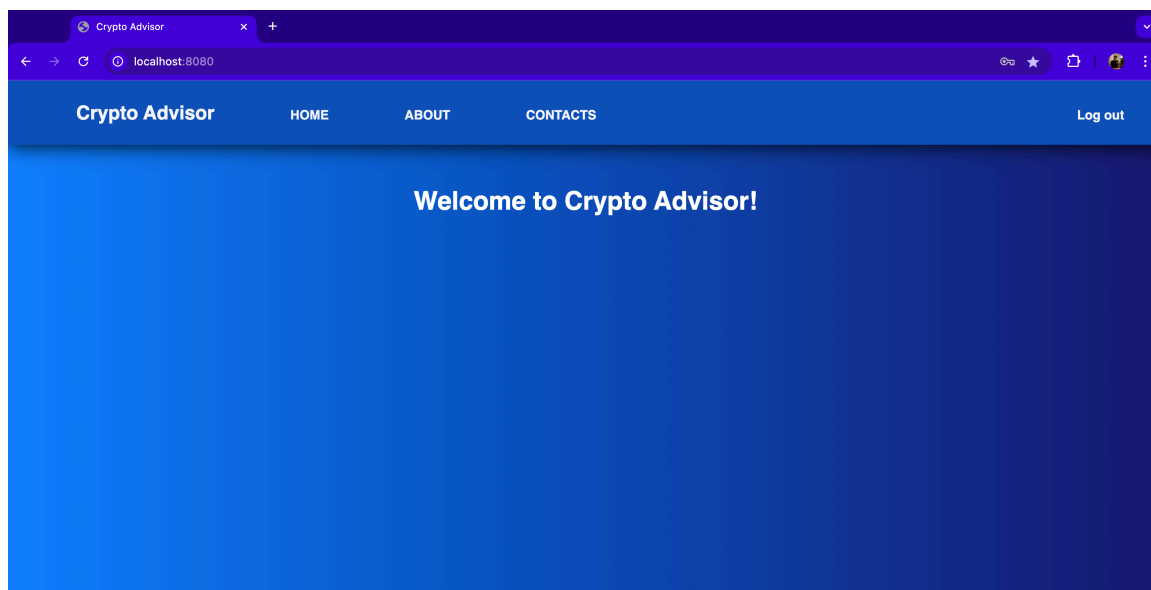


Рисунок 4.22 – Головна сторінка веб-сайту “Crypto Advisor”

У верхній частині сторінки розташована горизонтальна навігаційна панель, яка містить пункти “Crypto Advisor”, “Home”, “About” та “Contacts”.

Проте, перед початком користування сервісом новий користувач повинен зареєструватися. Для цього передбачена сторінка реєстрації (рис. 4.23).

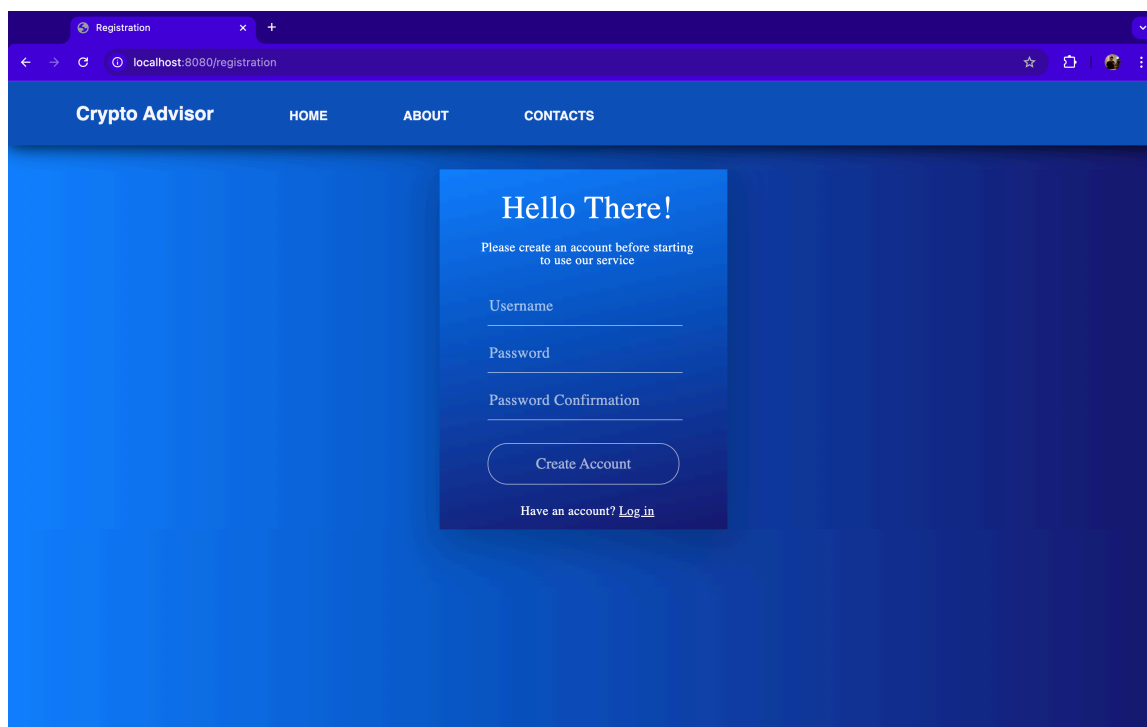


Рисунок 4.23 – Сторінка реєстрації

У формі реєстрації потрібно заповнити два головні поля: "username" та "password", і одне додаткове поле "password confirmation", вміст якого повинен співпадати з полем "password" – такий механізм дозволяє виявити випадкові помилки при написанні паролю користувачем на ранній стадії. Поле "username" повинно містити унікальне ім'я користувача, довжина якого варіюється від 5 до 15 символів, що повинно допомогти в запобіганні дублювання імен користувачів у системі. Поле "password" повинно містити пароль, який має бути довшим за 5 символів для забезпечення мінімального рівня безпеки.

Якщо користувач уже має обліковий запис, він може перейти на сторінку авторизації (рис. 4.24), де потрібно лише ввести ім'я свого користувача та відповідний пароль для входу в систему.

Для перегляду актуальних даних криптовалют і віджетів слід перейти на домашню сторінку за допомогою пункту меню "Home" (рисунок 4.2).

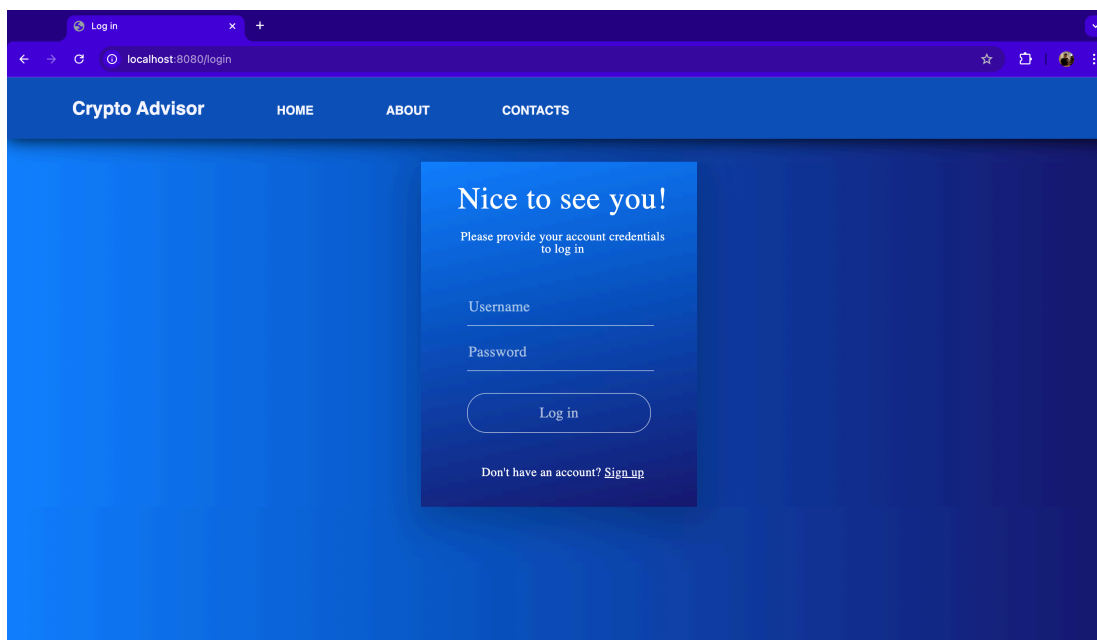


Рисунок 4.24 – Сторінка авторизації

На домашній сторінці користувач може вибрати криптовалюту зі списку і натиснути на кнопку “View” (рисунок 4.25), що спрямовує його на індивідуальну сторінку обраної криптовалюти. Ця сторінка містить найсвіжішу інформацію про вибрану криптовалюту та надає можливість переглянути лінійний графік, що відображає історичні та прогнозовані ціни активу (рисунок 4.3).

Statistics

localhost:8080/stats

Log out

Crypto Advisor

HOMEABOUTCONTACTS

TRENDING

1Cardano (ADA)25.65615% / 7d

2XRP (XRP)24.34091% / 7d

3Avalanche (AVAX)20.75042% / 7d

STABLE

1Tether USDt (USDT)-0.00471% / 90d

2USDC (USDC)0.00570% / 90d

3BNB (BNB)16.60974% / 90d

DEVIATING

1Solana (SOL)-4.87454% / 7d

2Dogecoin (DOGE)-2.58381% / 7d

3Tether USDt (USDT)-0.03504% / 7d

Rank	Name	Symbol	Price	1h %	24h %	7d %	30d %	60d %	Market Cap	Actions
1	Bitcoin	BTC	93272.73351	0.08778	-2.94655	0.91451	37.75748	41.09912	1845658651621.37740	View
2	Ethereum	ETH	3313.94524	-0.73396	-5.83299	6.39903	32.65036	23.11401	399111126741.00183	View
3	Tether USDt	USDT	1.00020	-0.01683	0.00552	-0.03504	0.13775	0.00849	132770550175.24898	View
4	Solana	SOL	229.02379	-0.93769	-5.85197	-4.87454	30.49771	44.08129	108746849680.62776	View
5	BNB	BNB	614.92902	-0.54704	-5.52598	0.05405	4.61561	0.84510	88556027270.84473	View
6	XRP	XRP	1.36000	-0.17069	-4.88848	24.34091	163.21261	130.79323	77517923479.56989	View
7	Dogecoin	DOGE	0.38475	0.63717	-7.83001	-2.58381	168.65950	202.77560	56540351309.65707	View
8	USDC	USDC	1.00009	-0.02824	0.00357	0.03624	-0.00152	0.01649	38584737603.66477	View
9	Cardano	ADA	0.91520	-0.57269	-8.56620	25.65615	168.29912	125.72479	32099423671.74914	View
10	Avalanche	AVAX	41.51794	-0.77565	-4.12984	20.75042	60.52280	37.15111	16991228430.29402	View

Рисунок 4.25 – Місцезнаходження кнопок “View”

Для ознайомлення з додатковою інформацією про програму можна скористатися пунктом “About” у верхній навігаційній панелі, який переведе користувача на відповідну сторінку з інформацією (рисунк 4.26).

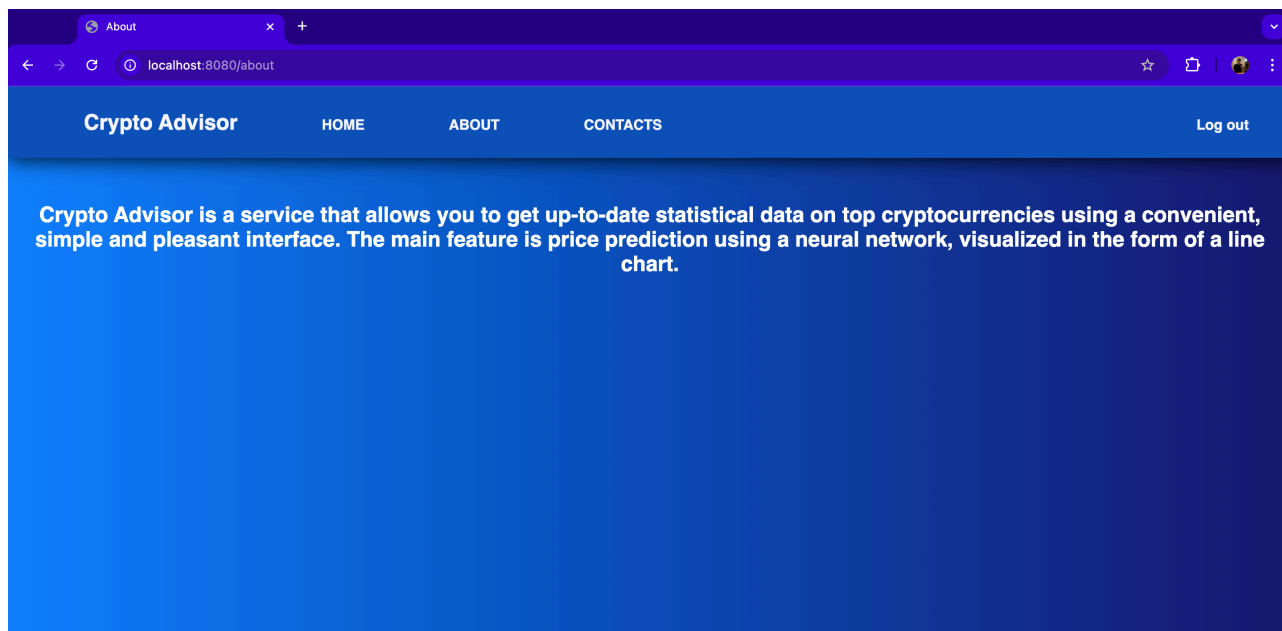


Рисунок 4.26 – Сторінка “About”

Для перегляду контактної інформації розробників можна обрати пункт “Contacts”. (рисунк 4.27).

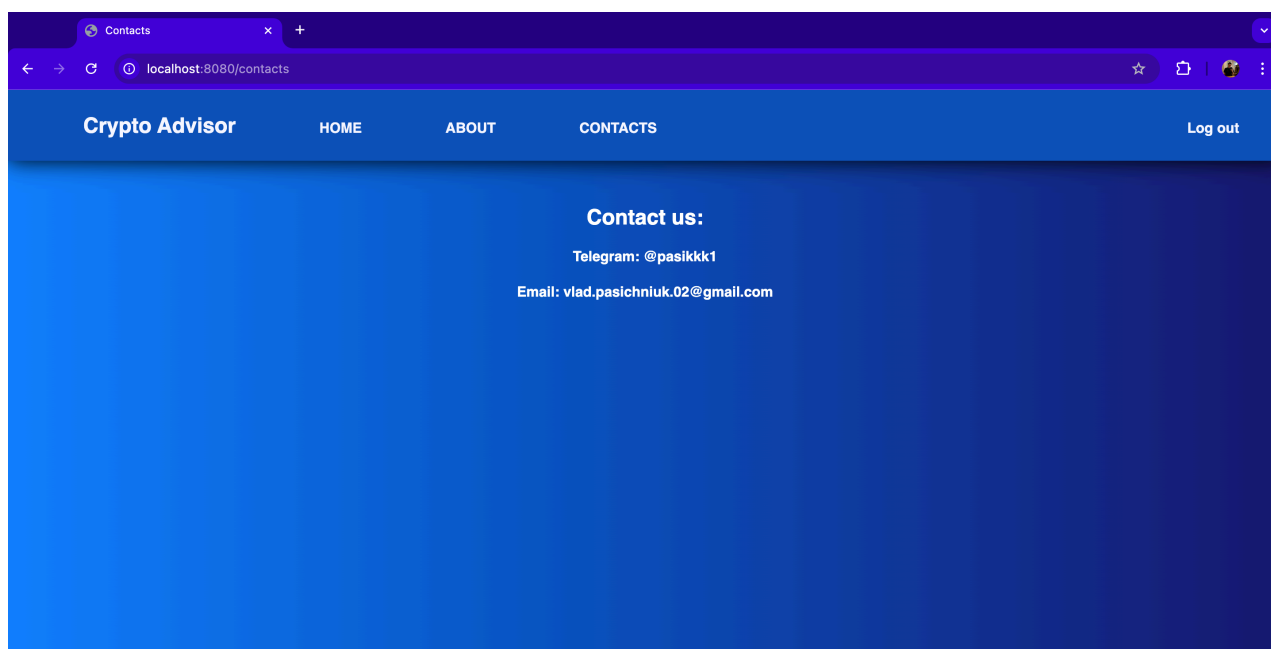


Рисунок 4.27 – Сторінка “Contacts”

Ця сторінка містить контакти команди, що розробляла програмний продукт та здійснює його підтримку.

4.5 Висновки

У четвертому розділі детально описано основні типи тестування, зокрема функціональне, інтеграційне, модульне та системне, і обрано найбільш оптимальні методи для перевірки розробленого програмного продукту. Виконано повне тестування системи, під час якого ретельно перевірено весь функціонал. Виявлені недоліки було виправлено, що дозволило підвищити стабільність і надійність роботи програми. Результати тестування підтвердили відповідність продукту всім поставленим задачам. Також було проведено порівняльний аналіз точності прогнозування.

Додатково було детально розглянуто основні функціональні можливості користувача створеного веб-сервісу та описано покрокові вказівки для роботи з наявним функціоналом та вирішення можливих питань.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Оцінювання комерційного потенціалу розробки

Метою проведення комерційного та технологічного аудиту є оцінювання комерційного потенціалу впровадження програмних засобів для веб-сервісу рекомендацій по торгівлі криптовалютою, який допомагатиме трейдерам та інвесторам приймати обґрунтовані рішення щодо купівлі або продажу активів, мінімізуючи ризики.

Для проведення технологічного аудиту було залучено трьох незалежних експертів Вінницького національного технічного університету кафедри програмного забезпечення. Для ідентифікації експертів використано нумерацію від 1 до 3: Експерт 1, Експерт 2 та Експерт 3. У таблиці 5.1 наведено 12 критеріїв оцінки комерційного потенціалу за п'ятибальною шкалою [30].

Таблиця 5.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Крт.	0	1	2	3	4
1	2	3	4	5	6
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження табл. 5.1.

1	2	3	4	5	6
4	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї

Продовження табл. 5.1.

1	2	3	4	5	6
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Таблиця 5.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

В таблиці 5.3 наведено результати оцінювання експертами комерційного потенціалу розробки.

Таблиця 5.3 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами:		
1	3	3	4
2	3	3	2
3	4	3	3
4	3	2	3
5	2	2	3
6	3	3	2
7	3	3	3
8	2	1	3
9	4	2	3
10	3	3	4
11	3	4	3
12	3	3	2
Сума балів	СБ ₁ =36	СБ ₂ =32	СБ ₃ =35
Середньоарифметична сума балів <u>СБ</u>	$\underline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = \frac{36 + 32 + 35}{3} = 34$		

Середньоарифметична сума балів, що було розрахована базуючись на висновках експертів, склала 34 бали. Згідно таблиці 5.2 можна стверджувати, що рівень комерційного потенціалу розробки є вищим за середній.

Новий веб-сервіс можна зробити доступним на весь світ використавши засоби платформ-хостингів, які також надають можливість отримати свій власний домен. У якості аналога розробки обрано веб-сайт Blockonomi.

Головними недоліками цього веб-сайту є відсутність корисних віджетів з аналітичною інформацією про криптовалюти, відображення неактуальних цін криптовалют, та відсутність функціоналу перегляду прогнозованих цін криптовалют, що може бути важливим для тих, хто шукає допоміжні інструменти для аналізу ринку та обґрунтованого прийняття рішень при інвестуванні.

Основною метою розробки веб-сервісу є реалізація функціоналу, що надасть перевагу над конкурентами в обраній сфері.

Далі проведемо оцінку якості та конкурентоспроможності нового веб-сервісу в порівнянні з аналогом. Основні техніко-економічні показники аналога і створеного програмного рішення наведені в таблиці 5.4.

Таблиця 5.4 – Основні параметри нового програмного рішення та аналога-конкурента

Показник	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (товар-конкурент)	Новий (інноваційне рішення)		
1	2	3	4	5
Кількість віджетів, шт	1	3	3	30%
Частота оновлення даних, с	25	15	1,67	15%
Затримка на виконанні запиту для отримання історичних даних, с	>60	5-15	4	30%
Кількість знаків після коми при відображенні цін	2	10	5	25%

Наступним кроком проведемо оцінку якості продукції, що є найбільш ефективним засобом задовільнення вимог користувачів та виконаємо порівняння з аналогом.

Розрахуємо відносні одиничні показники якості кожного параметра за формулами (5.1) і (5.2) та занесемо результати у відповідні колонки таблиці 5.5.

$$q_i = \frac{P_{Hi}}{P_{Bi}} \quad (5.1)$$

або

$$q_i = \frac{P_{Bi}}{P_{Hi}} \quad (5.2)$$

де P_{Hi} , P_{Bi} – числові значення і-го параметру відповідно нової та базової розробки.

$$q_1 = \frac{3}{1} = 3;$$

$$q_2 = \frac{25}{15} = 1,67;$$

$$q_3 = \frac{60}{15} = 4;$$

$$q_4 = \frac{10}{2} = 5.$$

Відносний рівень якості нової розробки розраховуємо за формулою:

$$K_{\text{я.в.}} = \sum_{i=1}^n q_i \cdot \alpha_i, \quad (5.3)$$

$$K_{\text{я.в.}} = 3 \cdot 0,3 + 1,67 \cdot 0,15 + 4 \cdot 0,3 + 5 \cdot 0,25 = 3,6.$$

Як бачимо, відносний коефіцієнт показника якості нової розробки є більшим за одиницю. Таким чином, можна стверджувати що нова розробка якісніша базового товару-конкурента.

Далі проведемо обрахунки конкурентоспроможності власної розробки. Конкурентоспроможність підприємства залежить від конкурентноспроможності товарів та є ключовим фактором прибутковості його діяльності.

Головною умовою обрання товару споживачем є збіг основних ринкових характеристик виробу з умовними характеристиками специфічної потреби клієнта. Одними з таких характеристик є нормативні та технічні параметри, а також ціна придбання та вартість споживання товару.

Нова розробка та аналог є безкоштовними для використання та не мають платного контенту, так як обидва орієнтуються на заробіток від рекламодавців.

Таблиця 5.5 містить технічні та економічні показники для розрахунку конкурентоспроможності нової розробки відносно розробки-аналога, технічні дані взяті з попередніх розрахунків.

Таблиця 5.5 – Нормативні, технічні та економічні параметри нової розробки і аналогу-конкурента

Показники	Варіанти	
	Базовий (товар-конкурент)	Новий (інноваційне рішення)
1	2	3
1. Нормативно-технічні показники		
Кількість віджетів, шт	1	3
Частота оновлення даних, с	25	15
Затримка на виконанні запиту для отримання історичних даних, с	>60	5-15
Кількість знаків після коми при відображенні цін	2	10
2. Економічні показники		
Ціна реклами, грн/рік	~360490	196560
Кількість рекламодавців, шт/рік	~6	1

Загальний показник конкурентоспроможності інноваційного рішення (К) з урахуванням вищезазначених груп показників можна визначити за формулою:

$$K = \frac{I_{m.n.}}{I_{e.n.}}, \quad (5.4)$$

де $I_{m.n.}$ – індекс технічних параметрів; $I_{e.n.}$ – індекс економічних параметрів.

Індекс технічних параметрів є відносним рівнем якості інноваційного рішення. Індекс економічних параметрів визначається за формулою (5.5)

$$I_{e.n.} = \frac{\sum_{i=1}^n P_{Hei}}{\sum_{i=1}^n P_{Bei}}, \quad (5.5)$$

де P_{Hei} , P_{Bei} – економічні параметри (ціна придбання та споживання товару) відповідно нового та базового товарів.

$$I_{e.n.} = \frac{196560}{360490} = 0,545;$$

$$K = \frac{3,6}{0,545} = 6,6.$$

Зважаючи на розрахунки, можна зробити висновок, що нова розробка буде конкурентоспроможніше, ніж конкурентний товар.

5.2 Прогнозування витрат на виконання науково-дослідної роботи

Витрати на проведення науково-дослідної роботи зазвичай групують за наступними статтями: витрати на оплату праці, витрати на соціальні заходи, матеріали, паливо та енергію для науково-виробничих цілей, витрати на службові відрядження, програмне забезпечення для наукових робіт, накладні витрати, та інші види.

Основна заробітна плата кожного із дослідників $З_0$, за умови, що вони працюють в наукових установах бюджетної сфери, розраховується за формулою:

$$З_0 = \frac{M}{T_p} * t \text{ (грн)} \quad (5.6)$$

де M – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

T_p – число робочих днів в місяці; приблизно $T_p \approx 21...23$ дні;

t – число робочих днів роботи дослідника.

Зведемо сумарні розрахунки до табл. 5.6.

Таблиця 5.6 – Заробітні плати дослідників в науковій установі бюджетної сфери

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	22000	1000	23	23000
Розробник	13000	590,9	18	10636,7
Всього				33636,7

Для здійснення розробки програмних засобів для веб-сервісу рекомендацій по торгівлі криптовалютою з використанням оптимізованих гібридних моделей нейронних мереж необхідно залучити одного розробника з посадовим окладом 13 тисяч гривень. Кількість робочих днів у місяці дорівнює 22, кожен з яких є робочим днем керівника, а у розробника на 5 робочих днів менше.

Далі проведемо розрахунки додаткової заробітної плати робітників.

Додаткова заробітна плата Z_d всіх розробників та робітників, які приймали участь у створенні нового технічного рішення, розраховується як 10-12 % від основної заробітної плати робітників.

На даному підприємстві додаткова заробітна плата начисляється в розмірі 12% від основної заробітної плати.

$$Z_d = (Z_o + Z_p) * \frac{H_{\text{дод}}}{100\%} \quad (5.7)$$

$$Z_d = 0,12 * 33636,7 = 4036,4 \text{ (грн)}$$

Нарахування на заробітну плату $H_{3П}$ дослідників та робітників, що приймали участь у виконанні цього етапу роботи, розраховуються за формулою:

$$H_{3П} = (Z_o + Z_d) * \frac{B}{100} \text{ (грн)}, \quad (5.8)$$

де Z_o – основна заробітна плата розробників, грн.;

Z_d – додаткова заробітна плата всіх розробників та робітників, грн.;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % .

Ставка єдиного внеску на загальнообов'язкове державне соціальне страхування складатиме 22%.

$$H_{3П} = (33636,7 + 4036,4) * \frac{22}{100} = 8288 \text{ (грн)}$$

Витрати на матеріали (М) та комплектуючі (К), які було використано у процесі виконання даного етапу роботи, розраховуються по кожному виду матеріалів за допомогою формули:

$$M = \sum_1^n H_i \cdot \Pi_i \cdot K_i - \sum_1^n B_i \cdot \Pi_b \quad (5.9)$$

де H_i – витрати матеріалу і-го найменування, кг;

Π_i – вартість матеріалу і-го найменування, грн./кг.;

K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$;

B_i – маса відходів матеріалу і-го найменування, кг;

Π_b – ціна відходів матеріалу і-го найменування, грн/кг;

n – кількість видів матеріалів.

Таблиця 5.7 – Матеріали, використані у ході розробки

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено	Вартість витраченого матеріалу, грн.
Набір паперу А4	120	1	120
Папка	20	1	20
Гелева ручка	32	1	32
Флешка	345	1	345
Всього			517
З врахуванням коефіцієнта транспортування			568,7

Програмні засоби для наукової роботи включають витрати на розробку та придбання спеціалізованого програмного забезпечення, що є обов'язковим для проведення дослідження.

Виходячи з того, що для нової розробки було використано безкоштовне середовище розробки IntelliJ IDEA, дані витрати враховуватись не будуть.

Наступним кроком визначимо амортизацію обладнання, комп'ютерів та приміщень, що були використані у ході виконання даного етапу роботи.

Ці розрахунки виконуються по кожному виду обладнання, приміщення, тощо.

$$A = \frac{Ц \cdot T}{T_{кор} \cdot 12} \quad [грн], \quad (5.10)$$

де Ц – балансова вартість даного виду обладнання (приміщень), грн.;

$T_{кор}$ – час користування;

T – термін використання обладнання (приміщень), цілі місяці.

Амортизаційні відрахування по кожному виду обладнання наведено у таблиці 5.8.

Таблиця 5.8 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Ноутбук	39000	2	1	1625
Обладнання робочого місця дослідника	20500	5	1	342
Всього				1967

Розрахунки амортизаційних витрат не включають витрати на використання програмного забезпечення, так як при розробці було використано ПЗ, що є безкоштовним. Оскільки у виконанні роботи приймало участь двоє спеціалістів

(керівник та розробник), кінцеву суму амортизаційних відрахувань треба збільшити вдвічі:

$$A = 2 \cdot 1967 = 3934 \text{ (грн)}$$

До статті «Паливо та енергія для науково-виробничих цілей» відносять усі витрати на різновиди палива та енергії, які були напряму використані з технологічною метою для проведення дослідження.

$$B_e = \sum_{i=1}^n \frac{W_{yt} \cdot t_i \cdot C_e \cdot K_{впi}}{\eta_i} \quad (5.11)$$

де W_{yt} – встановлена потужність обладнання на певному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн;

$K_{впi}$ – коефіцієнт, що враховує використання потужності, $K_{впi} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

Розрахуємо витрати на електроенергію для ноутбуків, які були використані у процесі написання магістерської кваліфікаційної роботи.

Тариф на електроенергію розраховується за наступною формулою:

$$C_e = (C_{опт} + C_{розп} + C_{пост}) \cdot \left(1 + \frac{\text{ПДВ}}{100\%}\right) \quad (5.12)$$

де $C_{опт}$ – оптова ціна електроенергії без ПДВ, грн. за кВт/год;

$C_{розп}$ – вартість розподілу електроенергії окремою розподільною компанією (без ПДВ), грн. за кВт/год;

$C_{пост}$ – вартість постачання електроенергії від розподільчої компанії (без ПДВ), грн за кВт/год.

Виходячи з того, що спеціалісти розташовані у Вінницькій області, для розрахунків було обрано тарифи від ПАТ "Вінницяобленерго":

$$C_e = (8,258 + 0,34 + 0,16) \cdot \left(1 + \frac{20}{100\%}\right) = 10,51 \text{ грн.}$$

Таким чином, загальна вартість витраченої електроенергії становитиме:

$$B_e = \frac{0,2 \cdot 170 \cdot 10,51 \cdot 0,6}{0,9} \cdot 2 = 238,2 \text{ грн.}$$

Оскільки у проведеному дослідженні відсутні службові відрядження та роботи, які виконуються сторонніми підприємствами, установими, організаціями, витрати на них враховуватися не будуть.

Загальновиробничі накладні витрати включають у себе витрати на управління організацією, оплату службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці, та інше. Ці витрати можна визначити як 100-150 % від суми основної заробітної плати розробників та робітників, які виконували дану роботу, тобто:

$$B_{\text{нзв}} = (Z_o + Z_p) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (5.12)$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Інші витрати».

$$B_{\text{нзв}} = 33636,7 \cdot \frac{100}{100\%} = 33636,7 \text{ грн}$$

Сума всіх попередніх статей витрат дає витрати, які безпосередньо стосуються даного розділу магістерської кваліфікаційної роботи:

$$B = 33636,7 + 3934 + 8288 + 568,7 + 238,2 + 33636,7 = 80302,3$$

Прогнозування загальних втрат ЗВ на виконання та впровадження результатів виконаної МКР розраховується за формулою:

$$ЗВ = \frac{B}{\eta}, \quad (5.13)$$

де η – коефіцієнт, який характеризує стадію виконання даної НДР.

Так як робота знаходиться на стадії науково-дослідних робіт, то коефіцієнт $\beta = 0,5$, звідси:

$$ЗВ = \frac{80302,3}{0,5} = 160604,6 \text{ грн.}$$

5.3 Оцінка економічної ефективності науково-дослідної роботи

У цьому підрозділі зробимо кількісне прогнозування вигоди, яку можна отримати в майбутньому від впровадження результатів виконаної наукової роботи. Проведемо розрахунки збільшення чистого прибутку підприємства $\Delta\Pi_i$, для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження нової розробки, за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{v}{100}\right), \quad (5.14)$$

де $\Delta\Pi_o$ – покращення основного оціночного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

Π_o – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

ρ – коефіцієнт, який враховує рентабельність продукту. $\rho = 0,25$;

x – ставка податку на прибуток. У 2024 році – 18%.

Зробимо припущення, що при впровадженні результатів наукової розробки покращується якість програмного продукту для рекомендацій по торгівлі криптовалютою. Також припустимо, що ціна надання рекламних послуг на рік на сайті зросте на 28400 грн протягом першого року. Кількість рекламодавців також збільшується: протягом першого року на трьох, протягом другого року – на п'ятьох,

протягом третього року на шість клієнтів. Кількість рекламодавців до впровадження розробки складалась з одного клієнта, а ціна надання рекламних послуг на рік до складає 196560 грн. Розрахуємо прибуток, який отримає підприємство протягом трьох років.

$$\Delta\Pi_1 = [28400 \cdot 1 + (196560 + 28400) \cdot 3] \cdot 0,833 \cdot 0,25 \cdot \left(1 + \frac{18}{100}\right) = 172820,5 \text{ грн.}$$

$$\Delta\Pi_2 = [28400 \cdot 1 + (196560 + 28400) \cdot (3 + 5)] \cdot 0,833 \cdot 0,25 \cdot \left(1 + \frac{18}{100}\right) = 449223,2 \text{ грн.}$$

$$\begin{aligned} \Delta\Pi_3 &= [28400 \cdot 1 + (196560 + 28400) \cdot (3 + 5 + 6)] \cdot 0,833 \cdot 0,25 \cdot \left(1 + \frac{18}{100}\right) \\ &= 780906,5 \text{ грн.} \end{aligned}$$

5.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Проведемо розрахунки основних показників, що визначають доцільність фінансування наукової розробки певним інвестором. Існують абсолютна і відносна ефективність вкладених інвестицій, а також термін їх окупності.

Обчислимо величину початкових інвестицій PV , які потенційний інвестор повинен вкласти для впровадження і комерціалізації науково-технічної розробки.

$$PV = k_{\text{інв}} \cdot 3B, \quad (5.15)$$

$k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. До них можуть входити витрати на створення технологій, підготовку приміщень, навчання робітників, маркетинг та інше ($k_{\text{інв}} = 2 \dots 5$).

$$PV = 2 \cdot 160604,6 = 321209,2$$

Далі розрахуємо абсолютну ефективність вкладених інвестицій $E_{\text{абс}}$, згідно нижченаведеної формули:

$$E_{\text{абс}} = (III - PV) \quad (5.16)$$

ПП – приведена вартість усіх чистих прибутків, які отримає підприємство від реалізації результатів наукової розробки, грн.;

$$ПП = \sum_1^T \frac{\Delta P_i}{(1 + \tau)^t}, \quad (5.17)$$

де ΔP_i – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати виконаної та впровадженої НДЦКР, грн.;

T – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,2;

t – період часу, роки.

$$ПП = \frac{172820,5}{(1 + 0,2)^1} + \frac{449223,2}{(1 + 0,2)^2} + \frac{780906,5}{(1 + 0,2)^3} = 907891,14 \text{ грн.}$$

$$E_{abc} = (907891,14 - 321209,2) = 586681,94 \text{ грн.}$$

Оскільки $E_{abc} > 0$, вкладання коштів на виконання та впровадження результатів НДДКР може бути доцільним.

Наступним кроком розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_e . Для цього користуються формулою:

$$E_e = \sqrt[T_{ж}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.18)$$

$T_{ж}$ – життєвий цикл наукової розробки, роки.

$$E_e = \sqrt[3]{1 + \frac{586681,94}{321209,2}} - 1 = 0,414 = 41,4\%$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.19)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = (0,14 \dots 0,2)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05 \dots 0,1)$.

$$\tau_{\min} = 0,18 + 0,05 = 0,23$$

Так як $E_b > \tau_{\min}$ то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_e} \quad (5.20)$$

$$T_{ок} = \frac{1}{0,414} = 2,4 \text{ роки}$$

Виходячи з того, що $T_{ок} \leq 3$ -х років, можна стверджувати, що фінансування даної наукової розробки є доцільним [26].

5.5 Висновки

В даному розділі було виконано оцінювання комерційного потенціалу програмних засобів для веб-сервісу рекомендацій по торгівлі криптовалютою з використанням гібридних оптимізованих моделей нейронних мереж, який знаходиться на рівні вище середнього. Під час порівняння нової розробки з аналогом було виявлено, що нова розробка є більш якісною й конкурентоспроможнішою відносно розробки-аналога, а також перевершує його у технічних та економічних показниках.

Прогнозування витрат на виконання науково-дослідної роботи по кожній з статей витрат складе 80302,3 грн. Загальна величина витрат на виконання та впровадження результатів даної роботи складатиме 160604,6 грн.

Інвестиції, вкладені в дану розробку, окупляться через 2,4 роки при прогнозованому прибутку 907891,14 грн. за три роки.

ВИСНОВКИ

У магістерській кваліфікаційній роботі було розглянуто стан предметної галузі веб-сервісів рекомендацій по торгівлі криптовалютою. Було проведено порівняльний аналіз існуючих аналогічних рішень та методів, що використовуються для задачі прогнозування ринкових тенденцій. Особлива увага приділена огляду різних типів моделей нейронних мереж, що дало змогу оцінити їх переваги та недоліки. У результаті аналізу доведено доцільність створення програмних засобів для веб-сервісу рекомендацій по торгівлі криптовалютою із використанням гібридних оптимізованих моделей нейронних мереж.

У роботі запропоновано та обґрунтовано удосконалений метод конфігурації моделей гібридних нейронних мереж, що базується на використанні рекурентних та згорткових шарів, включаючи підбір параметрів для оптимізації її роботи. Описано запропонований метод створення функціоналу прогнозування цін криптовалют, що забезпечив ефективний спосіб отримання орієнтовних майбутніх цін криптовалют та зручну візуалізацію результатів у форматі лінійного горизонтального графіка, завдяки застосуванню засобів бібліотеки Google Charts. Крім того, висвітлено принципи функціонування веб-сервісу, а також наведено схеми основних алгоритмів, що забезпечують його роботу. Серед них представлені діаграми класів, компонентів, діяльності та використання, які наочно демонструють структуру та логіку роботи системи. Увагу приділено також розробці структури інтерфейсу користувача.

Обґрунтовано вибір мови програмування та середовища розробки, що забезпечили ефективне виконання поставлених завдань. Детально описано етапи розробки програмного модуля та графічного інтерфейсу. Проведено кілька етапів тестування програмного продукту, що дало змогу оцінити його надійність і функціональність. Також здійснено порівняльний аналіз точності прогнозування, який підтвердив ефективність використання гібридної моделі нейромережі.

На завершальному етапі роботи проведено економічний аналіз для підтвердження фінансової доцільності розробки. У ньому виконано розрахунок витрат, пов'язаних із розробкою, та визначено потенційний прибуток, що може бути

отриманий від впровадження даного веб-сервісу. Це дозволило оцінити економічну вигідність проекту та перспективи його подальшого розвитку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Машинне навчання. URL: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning> (date of access: 20.09.2024).
2. Нейронна мережа. URL: <https://livingfo.com/shcho-take-nejronni-merezhi-ta-iaak-vony-pratsiuiut/> (date of access: 20.09.2024).
3. Романюк О. Н., Пасічнюк В. А. Аналіз бібліотек машинного навчання для мови Java // Матеріали XXIII Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 20-21 квітня 2023 р. - Одеса, Видавництво ОНТУ, 2023 р. – 449 с.– С. 362-363.
4. Пасічнюк В. А., Романюк О. Н. Основні типи нейронних мереж // Молодь в науці: дослідження, проблеми, перспективи (2023). URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2023/paper/view/18204> (date of access: 21.09.2024).
5. Пасічнюк В. А., Коваленко О. О. Аналіз типів моделей нейронних мереж для задачі прогнозування фінансових показників // Матеріали XVII Міжнародної науково-практичної конференції “Інформаційні технології і автоматизація (2024)”. Одеса, 31 жовтня – 1 листопада 2024 р. – Одеса, Видавництво ОНТУ, 2024 р. – 847 с. – С. 89-90.
6. Crypto data and research sites. URL: <https://www.analyticsinsight.net/top-10-crypto-data-and-research-sites-investors-should-know/> (date of access: 22.09.2024).
7. What makes crypto volatile? URL: <https://www.paypal.com/us/cshelp/article/what-makes-crypto-volatile-help559> (date of access: 24.09.2024).
8. What is CoinMarketCap? URL: https://en.wikipedia.org/wiki/Coin_Market_Cap (date of access: 24.09.2024).
9. What is CoinGecko?. URL: <https://golden.com/wiki/CoinGecko-6398V3M> (date of access: 24.09.2024).
10. Blockonomi. URL: <https://blockonomi.com/crypto-market-cap/> (date of access: 24.09.2024).

11. AWS SageMaker. URL: <https://docs.aws.amazon.com/sagemaker/latest/dg/whatis.html> (date of access 02.10.2024).
12. Azure Machine Learning. URL: <https://learn.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-machine-learning?view=azureml-api-2> (date of access 02.10.2024).
13. Guide to deeplearning4j. URL: <https://codenamewei.medium.com/ultimate-guide-to-getting-started-with-deeplearning4j-d497603cbe0b> (date of access 04.10.2024).
14. Feedforward neural networks. URL: <https://learnopencv.com/understanding-feedforward-neural-networks/> (date of access 11.10.2024).
15. LSTM neural networks. URL: <https://www.turing.com/kb/comprehensive-guide-to-lstm-rnn> (date of access 11.10.2024).
16. Convolutional neural networks. URL: <https://www.researchgate.net/publication/336805909> (date of access 11.10.2024).
17. Google Charts. URL: https://www.w3schools.com/js/js_graphics_google_chart.asp (date of access: 13.10.2024).
18. Alpha Vantage API guide. URL: <https://www.alphavantage.co/documentation/> (date of access: 14.10.2024).
19. Usability characteristics for user interface design. URL: <https://www.scirp.org/reference/referencespapers?referenceid=1682244> (date of access 15.10.2024).
20. User experience towards a perspective on product quality. URL: https://www.researchgate.net/publication/238472807_User_experience_UX_Towards_an_experiential_perspective_on_product_quality (date of access 15.10.2024).
21. Java Virtual Machine. URL: <https://gocoding.org/uk/java-virtual-machine-jvm-in-java/> (date of access: 21.10.2024).
22. Garbage Collection in Java. URL: <https://www.geeksforgeeks.org/garbage-collection-java> (date of access: 21.10.2024).
23. Илго take C#? URL: <https://robotdreams.cc/uk/blog/284-s-cto-eto-za-yazyk-i-gde-ego-ispolzuyut> (date of access: 22.10.2024).

24. What is the purpose of C++? URL: <https://uk.myservername.com/what-is-c-used> (date of access: 22.10.2024).
25. Intro to IntelliJ IDEA. URL: https://www.tutorialspoint.com/intellij_idea/intellij_idea_introduction.htm (date of access: 25.10.2024).
26. Eclipse IDE. URL: [https://en.wikipedia.org/wiki/Eclipse_\(software\)](https://en.wikipedia.org/wiki/Eclipse_(software)) (date of access: 25.10.2024).
27. The State of Developer Ecosystem in 2021. URL: <https://www.jetbrains.com/lp/devecosystem-2021/java/> (date of access: 01.11.2024).
28. Types of software testing. URL: <https://www.softwaretestinghelp.com/types-of-software-testing/> (date of access: 12.11.2024).
29. Швидкість завантаження сайту. URL: <https://wezom.com.ua/ua/blog/kak-uvelichit-skorost-zagruzki-sajta> (date of access: 13.11.2024).
30. Козловський В.О., Лесько О.Й., Кавецький В.В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. Вінниця: ВНТУ, 2021. 42 с.

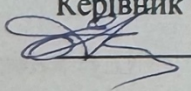
ДОДАТКИ

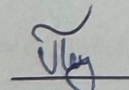
ДОДАТОК А. Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
д.т.н., професор Романюк О.Н.
«19» вересня 2024 року

Технічне завдання
на магістерську кваліфікаційну роботу «Розробка програмних засобів для веб-сервісу рекомендацій по торгівлі криптовалютою з використанням оптимізованих гібридних моделей нейронних мереж»
за спеціальністю
121 – Інженерія програмного забезпечення

Керівник магістерської кваліфікаційної роботи:
 к.т.н., доц. каф. ПЗ Коваленко О. О.
" 19 " вересня 2024 р.

Виконав:
 студент гр. 2ПІ-23м Пасічнюк В. А.
" 19 " вересня 2024 р.

Вінниця – 2024 рік

1. Найменування та галузь застосування

Магістерська кваліфікаційна робота: «Розробка програмних засобів для веб-сервісу рекомендацій по торгівлі криптовалютою з використанням оптимізованих гібридних моделей нейронних мереж».

Галузь застосування – програмне забезпечення для попереднього аналізу при торгівлі та інвестуванні.

2. Підстава для розробки

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ № 310 від 17.09.2024 ректора по ВНТУ про затвердження тем МКР.

3. Мета та призначення розробки

Метою магістерської кваліфікаційної роботи є покращення процесів купівлі та продажу криптовалют за рахунок використання оптимізованих гібридних моделей нейронних мереж у програмному модулі, який здатен прогнозувати ціни криптовалют, базуючись на історичних даних зміни цін.

Призначення роботи – розробка програмних засобів для веб-сервісу рекомендацій по торгівлі криптовалютою з використанням оптимізованих гібридних моделей нейронних мереж.

4. Вихідні дані для проведення НДР

Актуальні дані криптовалют з CoinMarketCap API; історичні дані криптовалют з AlphaVantage API; конфігурація оптимізованої гібридної моделі нейронної мережі; мова програмування Java; фреймворки та бібліотеки: Spring Framework, Thymeleaf, Deeplearning4j, Gson, Google Charts; розмір екрану 2560-1600px, кольоровий режим, тип графіку – лінійний горизонтальний.

5. Вимоги до програмного засобу

Параметри розроблюваного програмного засобу:

- операційна система – будь-яка;
- мова програмування – Java;
- середовище розробки – IntelliJ IDEA.

Програмний засіб повинен виконувати такі дії:

- надавати можливість перегляду актуальних даних про криптовалюту;
- надавати можливість перегляду історичних даних цін криптовалют;
- мати віджети з корисними статистичними даними;
- надавати можливість перегляду прогнозованих цін криптовалют.

6. Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до МКР;
- технічне завдання;
- лістинги програми.

7. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

8. Стадії та етапи розробки

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз стану веб-сервісів рекомендацій по торгівлі криптовалютою	20.09.2024 – 10.10.2024
2	Методи конфігурування нейронної мережі та створення веб-сервісу	10.10.2024 – 21.10.2024
3	Розробка програми для веб-сервісу рекомендацій по торгівлі криптовалютою	21.10.2024 – 15.11.2024
4	Тестування програмного продукту	15.11.2024 – 22.11.2024
5	Економічна частина	22.11.2024 – 30.11.2024

9. Порядок контролю та прийняття

Виконання етапів магістерської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи. Прийняття магістерської кваліфікаційної роботи здійснюється екземенаційною комісією, затвердженою зав. кафедрою згідно з графіком.

ДОДАТОК Б. Протокол перевірки

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: Розробка програмних засобів для веб-сервісу рекомендацій по торгівлі криптовалютою з використанням оптимізованих гібридних моделей нейронних мереж.

Тип роботи: кваліфікаційна робота

Підрозділ : кафедра програмного забезпечення, ФІТКІ, 2ПІ-23м

Науковий керівник: доц. Коваленко О. О.

Turnitin	
Оригінальність	95 %
Схожість	5 %

Аналіз звіту подібності

■ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлена з повним звітом подібності, який був згенерований системою щодо роботи «Розробка програмних засобів для веб-сервісу рекомендацій по торгівлі криптовалютою з використанням оптимізованих гібридних моделей нейронних мереж».

Автор

Пасічнюк Владислав Андрійович

Опис прийнятого рішення: допустити до захисту

Особа, відповідальна за перевірку
(підпис) (прізвище, ініціали)

Черноволик Г. О.

Експерт

(за потреби)

(підпис)

(прізвище, ініціали, посада)

ДОДАТОК В. Лістинг коду

CryptoAdvisorApplication.java

```
package com.crypto.advisor;

import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class CryptoAdvisorApplication {

    public static void main(String[] args) {
        SpringApplication.run(CryptoAdvisorApplication.class, args);
    }
}
```

ApplicationConfig.java

```
package com.crypto.advisor.config;

import org.springframework.context.annotation.Configuration;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.servlet.config.annotation.InterceptorRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

import com.crypto.advisor.interceptor.RateLimitInterceptor;

@Configuration
public class ApplicationConfig implements WebMvcConfigurer {

    private final RateLimitInterceptor interceptor;

    @Autowired
    public ApplicationConfig(RateLimitInterceptor interceptor) {
        this.interceptor = interceptor;
    }

    @Override
    public void addInterceptors(InterceptorRegistry registry) {
        registry.addInterceptor(interceptor).addPathPatterns("/*");
    }
}
```

SecurityConfig.java

```
package com.crypto.advisor.config;

import com.crypto.advisor.model.Role;
import org.springframework.context.annotation.Configuration;
```

```

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import
org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
import org.springframework.security.authentication.dao.DaoAuthenticationProvider;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import lombok.RequiredArgsConstructor;

```

```
@Configuration
```

```
@EnableWebSecurity
```

```
@RequiredArgsConstructor
```

```
public class SecurityConfig extends WebSecurityConfigurerAdapter {
```

```
    private final UserDetailsService service;
```

```
    @Override
```

```
    protected void configure(HttpSecurity http) throws Exception {
```

```
        http.csrf()
            .disable()
            .authorizeRequests()
            .antMatchers("/users/**")
            .hasRole(Role.ADMIN.name())
            .antMatchers("/registration", "/error", "/static/**", "/styles/**")
            .permitAll()
            .anyRequest()
            .authenticated()
            .and()
            .formLogin()
            .loginPage("/login")
            .defaultSuccessUrl("/")
            .permitAll()
            .and()
            .logout()
            .invalidateHttpSession(true)
            .permitAll();
    }

```

```
    @Override
```

```
    protected void configure(AuthenticationManagerBuilder builder) {
```

```
        builder.authenticationProvider(authenticationProviderBean());
```

```
    }
```

```
    @Bean
```

```
    public DaoAuthenticationProvider authenticationProviderBean() {
```

```
        var authenticationProvider = new DaoAuthenticationProvider();
```

```
        authenticationProvider.setUserDetailsService(service);
```

```
        authenticationProvider.setPasswordEncoder(passwordEncoderBean());
```

```

        return authenticationProvider;
    }

    @Bean
    public PasswordEncoder passwordEncoderBean() {
        return new BCryptPasswordEncoder();
    }
}

```

CryptoController.java

```

package com.crypto.advisor.controller;

import com.crypto.advisor.model.Constants;
import com.crypto.advisor.model.CryptoStats;
import lombok.RequiredArgsConstructor;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.lang.NonNull;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;

import com.crypto.advisor.service.CryptoService;

import java.util.ArrayList;
import java.util.Comparator;
import java.util.List;
import java.util.Set;
import java.util.stream.Collectors;

@Controller
@RequiredArgsConstructor
public class CryptoController {

    private final CryptoService cryptoService;

    @GetMapping("/stats")
    public String getCryptoStatistics(Model model) {
        var cryptoStats = cryptoService.getCryptoStatistics();
        model.addAttribute("cryptoStatsSet", cryptoStats);

        addTrendingCryptos(model, cryptoStats);
        addDeviatingCryptos(model, cryptoStats);
        addStableCryptos(model, cryptoStats);

        return Constants.ALL_CRYPTOSTATS_PATH;
    }

    private void addTrendingCryptos(Model model, Set<CryptoStats> cryptoStats) {
        var cryptoStatsCopy = getCryptoStatsCopy(cryptoStats);
        var trendingCryptos = List.copyOf(cryptoStatsCopy).stream()

```

```

        .sorted(Comparator.comparingDouble(CryptoStats::getPercentChangeWeek).reversed())
        .limit(3)
        .collect(Collectors.toList());

    trendingCryptos.forEach(crypto -> crypto.setRank((long) trendingCryptos.indexOf(crypto) + 1));

    model.addAttribute("trendingCryptos", trendingCryptos);
}

private void addDeviatingCryptos(Model model, Set<CryptoStats> cryptoStats) {
    var cryptoStatsCopy = getCryptoStatsCopy(cryptoStats);
    var deviatingCryptos = List.copyOf(cryptoStatsCopy).stream()
        .sorted(Comparator.comparingDouble(CryptoStats::getPercentChangeWeek))
        .limit(3)
        .collect(Collectors.toList());

    deviatingCryptos.forEach(crypto -> crypto.setRank((long) deviatingCryptos.indexOf(crypto) + 1));

    model.addAttribute("deviatingCryptos", deviatingCryptos);
}

private void addStableCryptos(Model model, Set<CryptoStats> cryptoStats) {
    var cryptoStatsCopy = getCryptoStatsCopy(cryptoStats);
    var stableCryptos = List.copyOf(cryptoStatsCopy).stream()
        .sorted(Comparator.comparingDouble(s -> Math.abs(s.getPercentChangeThreeMonths())))
        .limit(3)
        .collect(Collectors.toList());

    stableCryptos.forEach(crypto -> crypto.setRank((long) stableCryptos.indexOf(crypto) + 1));

    model.addAttribute("stableCryptos", stableCryptos);
}

private List<CryptoStats> getCryptoStatsCopy(Set<CryptoStats> cryptoStats) {
    List<CryptoStats> cryptoStatsCopy = new ArrayList<>();
    cryptoStats.forEach(stats -> cryptoStatsCopy.add(new CryptoStats(stats)));
    return cryptoStatsCopy;
}

@GetMapping("/stats/{symbol}")
public String getCryptoStatistics(@PathVariable @NonNull String symbol, Model model) {
    model.addAttribute("cryptoStats", cryptoService.getCryptoStatisticsBySymbol(symbol));
    model.addAttribute("historicalData",
cryptoService.getHistoricalAndPredictedData("DIGITAL_CURRENCY_DAILY", symbol));
    return Constants.CRYPTO_STATS_PATH;
}

@GetMapping("/{", Constants.HOME_PATH})
public String home() {
    return Constants.HOME_PATH;
}

```

```

    @GetMapping("/contacts")
    public String contacts() {
        return Constants.CONTACTS_PATH;
    }

    @GetMapping("/about")
    public String about() {
        return Constants.ABOUT_PATH;
    }
}

```

CustomErrorController.java

```

package com.crypto.advisor.controller;

import com.crypto.advisor.model.Constants;
import org.springframework.boot.web.servlet.error.ErrorController;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

import javax.servlet.RequestDispatcher;
import javax.servlet.http.HttpServletRequest;

@Controller
public class CustomErrorController implements ErrorController {

    @RequestMapping(Constants.ERROR_PATH)
    public ModelAndView handleError(HttpServletRequest request) {
        String message = (String) request.getAttribute(RequestDispatcher.ERROR_MESSAGE);
        if (message != null && message.isEmpty()) {
            message = null;
        }

        return new ModelAndView(Constants.ERROR_PATH)
            .addObject("message", message);
    }
}

```

SecurityController.java

```

package com.crypto.advisor.controller;

import com.crypto.advisor.entity.User;
import com.crypto.advisor.model.Constants;
import com.crypto.advisor.service.dao.UserDao;
import com.crypto.advisor.service.validator.UserValidator;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Controller;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.WebDataBinder;
import org.springframework.web.bind.annotation.GetMapping;

```

```

import org.springframework.web.bind.annotation.InitBinder;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.servlet.ModelAndView;

import javax.validation.Valid;
import java.security.Principal;

@Controller
@RequiredArgsConstructor
public class SecurityController {

    private final UserDao userDao;
    private final UserValidator userValidator;

    @GetMapping(Constants.LOGIN_PATH)
    public String login(Principal principal) {
        return principal != null ? Constants.HOME_PATH : Constants.LOGIN_PATH;
    }

    @GetMapping(Constants.REGISTRATION_PATH)
    public ModelAndView registration() {
        return new ModelAndView(Constants.REGISTRATION_PATH)
            .addObject("user", new User());
    }

    @PostMapping(Constants.REGISTRATION_PATH)
    public ModelAndView registration(@Valid User user, BindingResult result) {
        if (result.hasErrors()) {
            return new ModelAndView(Constants.REGISTRATION_PATH);
        }

        userDao.save(user);

        return new ModelAndView(Constants.LOGIN_PATH)
            .addObject("message", "User has been registered successfully");
    }

    @InitBinder
    private void bindValidator(WebDataBinder binder) {
        binder.setValidator(userValidator);
    }
}

```

UserController.java

```

package com.crypto.advisor.controller;

import com.crypto.advisor.model.Constants;
import com.crypto.advisor.service.dao.UserDao;
import lombok.NonNull;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Controller;

```

```

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

@Controller
@RequestMapping(Constants.USERS_PATH)
@RequiredArgsConstructor
public class UserController {

    private final UserDao userDao;

    @GetMapping
    public ModelAndView findAll() {
        var users = userDao.findAll();
        return new ModelAndView(Constants.USERS_PATH)
            .addObject("users", users);
    }

    @GetMapping("/{username}/enable")
    public ModelAndView block(@PathVariable @NonNull String username,
                              @RequestParam(defaultValue = "false") boolean value) {
        userDao.enable(username, value);
        var users = userDao.findAll();
        return new ModelAndView(Constants.USERS_PATH)
            .addObject("users", users);
    }
}

```

User.java

```

package com.crypto.advisor.entity;

import java.io.Serializable;

import javax.persistence.*;
import javax.validation.constraints.NotBlank;

import com.crypto.advisor.model.Role;
import org.hibernate.validator.constraints.Length;
import com.fasterxml.jackson.annotation.JsonIgnore;
import lombok.Data;
import lombok.Builder;
import lombok.NoArgsConstructor;
import lombok.AllArgsConstructor;

@Data
@Entity
@Builder
@NoArgsConstructor

```



```

@AllArgsConstructor
@Table(name = "user")
public class User implements Serializable {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", unique = true, nullable = false)
    private Long id;

    @Column(name = "username", unique = true, nullable = false)
    @NotBlank(message = "Username is required")
    @Length(min = 5, max = 15, message = "Username must be between 5 and 15 characters")
    private String username;

    @JsonIgnore
    @Column(name = "password", nullable = false)
    @NotBlank(message = "Password is required")
    @Length(min = 5, message = "Password must be greater than 5 characters")
    private String password;

    @Transient
    @JsonIgnore
    private String passwordConfirmation;

    @Column(name = "is_enabled", nullable = false)
    private Boolean isEnabled = Boolean.TRUE;

    @Column(name = "role", nullable = false)
    @Enumerated(EnumType.STRING)
    private Role role = Role.USER;
}

```

CryptoNotFoundException.java

```

package com.crypto.advisor.exception;

public class CryptoNotFoundException extends IllegalArgumentException {

    public CryptoNotFoundException(String symbol) {
        super(String.format("Couldn't find cryptocurrency with symbol '%s'", symbol));
    }
}

```

RateLimitInterceptor.java

```

package com.crypto.advisor.interceptor;

import java.time.Duration;
import java.io.IOException;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```

```

import org.springframework.stereotype.Component;
import org.springframework.http.HttpStatus;
import org.springframework.web.servlet.HandlerInterceptor;
import io.github.bucket4j.Bucket;
import io.github.bucket4j.Refill;
import io.github.bucket4j.Bandwidth;
import lombok.NonNull;

@Component
public class RateLimitInterceptor implements HandlerInterceptor {

    private static final long CAPACITY = 50;

    private final Bucket tokenBucket;

    public RateLimitInterceptor() {
        var limit = Bandwidth.classic(CAPACITY, Refill.greedy(CAPACITY, Duration.ofMinutes(5)));
        this.tokenBucket = Bucket.builder()
            .addLimit(limit)
            .build();
    }

    @Override
    public boolean preHandle(@NonNull HttpServletRequest request,
        @NonNull HttpServletResponse response,
        @NonNull Object handler) throws IOException {

        var probe = tokenBucket.tryConsumeAndReturnRemaining(1);

        if (probe.isConsumed()) {
            response.addHeader("X-Rate-Limit-Remaining", String.valueOf(probe.getRemainingTokens()));
            return true;
        } else {
            var waitForRefill = probe.getNanosToWaitForRefill() / 1_000_000_000;
            response.addHeader("X-Rate-Limit-Retry-After-Seconds", String.valueOf(waitForRefill));
            response.sendError(HttpStatus.TOO_MANY_REQUESTS.value());
            return false;
        }
    }
}

```

Constants.java

```

package com.crypto.advisor.model;

import lombok.AccessLevel;
import lombok.NoArgsConstructor;

@NoArgsConstructor(access = AccessLevel.PRIVATE)
public final class Constants {

```

```

public static final String REGISTRATION_PATH = "/registration";
public static final String LOGIN_PATH = "/login";
public static final String USERS_PATH = "/users";
public static final String HOME_PATH = "/home";
public static final String ALL_CRYPTOSTATS_PATH = "/all-crypto-stats";
public static final String CRYPTO_STATS_PATH = "/crypto-stats";
public static final String CONTACTS_PATH = "/contacts";
public static final String ABOUT_PATH = "/about";
public static final String ERROR_PATH = "/error";
}

```

CryptoData.java

```

package com.crypto.advisor.model;

import lombok.AllArgsConstructor;
import lombok.Data;

@Data
@AllArgsConstructor
public class CryptoData {
    private String date;
    private String symbol;
    private double price;
}

```

CryptoStats.java

```

package com.crypto.advisor.model;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.util.Map;

@Data
@NoArgsConstructor
@JsonIgnoreProperties(ignoreUnknown = true)
public class CryptoStats {

    private Long id;

    @JsonProperty("cmc_rank")
    private Long rank;

    private String name;
    private String symbol;
    private String lastUpdated;

    private Double price;
}

```

```

private Double marketCap;
private Double percentChangeHour;
private Double percentChangeDay;
private Double percentChangeWeek;
private Double percentChangeMonth;
private Double percentChangeTwoMonths;
private Double percentChangeThreeMonths;

public CryptoStats(CryptoStats statsToClone) {
    this.id = statsToClone.getId();
    this.rank = statsToClone.getRank();
    this.name = statsToClone.getName();
    this.symbol = statsToClone.getSymbol();
    this.lastUpdated = statsToClone.getLastUpdated();
    this.price = statsToClone.getPrice();
    this.marketCap = statsToClone.getMarketCap();
    this.percentChangeHour = statsToClone.getPercentChangeHour();
    this.percentChangeDay = statsToClone.getPercentChangeDay();
    this.percentChangeWeek = statsToClone.getPercentChangeWeek();
    this.percentChangeMonth = statsToClone.getPercentChangeMonth();
    this.percentChangeTwoMonths = statsToClone.getPercentChangeTwoMonths();
    this.percentChangeThreeMonths = statsToClone.getPercentChangeThreeMonths();
}

@SuppressWarnings("unchecked")
@JsonProperty("quote")
private void unpackFieldsFromQuote(Map<String, Object> quote) {
    Map<String, Object> node = (Map<String, Object>) quote.get("USD");

    lastUpdated = (String) node.get("last_updated");

    price = (Double) node.get("price");
    marketCap = (Double) node.get("market_cap");
    percentChangeHour = (Double) node.get("percent_change_1h");
    percentChangeDay = (Double) node.get("percent_change_24h");
    percentChangeWeek = (Double) node.get("percent_change_7d");
    percentChangeMonth = (Double) node.get("percent_change_30d");
    percentChangeTwoMonths = (Double) node.get("percent_change_60d");
    percentChangeThreeMonths = (Double) node.get("percent_change_90d");
}
}

```

Role.java

```

package com.crypto.advisor.model;

import lombok.Getter;
import lombok.RequiredArgsConstructor;

@Getter
@RequiredArgsConstructor
public enum Role {

```

```

    ADMIN("ROLE_ADMIN"),
    USER("ROLE_USER");

    private final String code;
}

```

UserRepository.java

```

package com.crypto.advisor.repository;

import com.crypto.advisor.entity.User;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import java.util.Optional;

@Repository
public interface UserRepository extends JpaRepository<User, Long> {

    Optional<User> findByUsername(String username);
}

```

UserDetailsImpl.java

```

package com.crypto.advisor.security;

import com.crypto.advisor.entity.User;
import lombok.RequiredArgsConstructor;
import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.authority.SimpleGrantedAuthority;
import org.springframework.security.core.userdetails.UserDetails;

import java.util.Collection;
import java.util.List;

@RequiredArgsConstructor
public class UserDetailsImpl implements UserDetails {

    private final User user;

    @Override
    public Collection<? extends GrantedAuthority> getAuthorities() {
        return List.of(new SimpleGrantedAuthority(user.getRole().getCode()));
    }

    @Override
    public String getPassword() {
        return user.getPassword();
    }

    @Override
    public String getUsername() {

```

```

    return user.getUsername();
}

@Override
public boolean isAccountNonExpired() {
    return true;
}

@Override
public boolean isAccountNonLocked() {
    return true;
}

@Override
public boolean isCredentialsNonExpired() {
    return true;
}

@Override
public boolean isEnabled() {
    return user.getIsEnabled();
}
}

```

UserDetailsServiceImpl.java

```

package com.crypto.advisor.security;

import com.crypto.advisor.repository.UserRepository;
import lombok.RequiredArgsConstructor;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Service;

@Service
@RequiredArgsConstructor
public class UserDetailsServiceImpl implements UserDetailsService {

    private final UserRepository repository;

    @Override
    public UserDetails loadUserByUsername(final String username) throws UsernameNotFoundException
    {
        final var user = repository.findByUsername(username)
            .orElseThrow(() -> new UsernameNotFoundException(
                String.format("User with username '%s' not found", username)
            ));
        return new UserDetailsImpl(user);
    }
}

```

UserDao.java

```
package com.crypto.advisor.service.dao;

import com.crypto.advisor.entity.User;

import java.util.Optional;

public interface UserDao {

    void save(User user);

    Iterable<User> findAll();

    Optional<User> findByUsername(String username);

    void enable(String username, boolean value);
}
```

UserDaoImpl.java

```
package com.crypto.advisor.service.dao;

import com.crypto.advisor.entity.User;
import com.crypto.advisor.repository.UserRepository;
import lombok.RequiredArgsConstructor;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import java.util.Optional;

@Service
@Transactional
@RequiredArgsConstructor
public class UserDaoImpl implements UserDao {

    private final UserRepository userRepository;
    private final PasswordEncoder passwordEncoder;

    @Override
    public void save(User user) {
        var encodedPassword = passwordEncoder.encode(user.getPassword());
        user.setPassword(encodedPassword);
        userRepository.save(user);
    }

    @Override
    public Iterable<User> findAll() {
        return userRepository.findAll();
    }
}
```

```

@Override
public Optional<User> findByUsername(String username) {
    return userRepository.findByUsername(username);
}

@Transactional
@Override
public void enable(String username, boolean value) {
    var user = userRepository.findByUsername(username).orElseThrow();
    user.setIsEnabled(value);
    userRepository.saveAndFlush(user);
}
}

```

NeuralNetworkBuilder.java

```

package com.crypto.advisor.service.prediction.model;

import lombok.AccessLevel;
import lombok.NoArgsConstructor;
import org.deeplearning4j.nn.api.OptimizationAlgorithm;
import org.deeplearning4j.nn.conf.*;
import org.deeplearning4j.nn.conf.layers.*;
import org.deeplearning4j.nn.conf.preprocessor.FeedForwardToRnnPreProcessor;
import org.deeplearning4j.nn.conf.preprocessor.RnnToFeedForwardPreProcessor;
import org.deeplearning4j.nn.multilayer.MultiLayerNetwork;
import org.deeplearning4j.nn.weights.WeightInit;
import org.deeplearning4j.optimize.listeners.ScoreIterationListener;
import org.nd4j.linalg.activations.Activation;
import org.nd4j.linalg.lossfunctions.LossFunctions;

@NoArgsConstructor(access = AccessLevel.PRIVATE)
public final class NeuralNetworkBuilder {

    private static final int SEED = 12345;
    private static final int ITERATIONS = 1;
    private static final double LEARNING_RATE = 0.01;
    private static final double DROPOUT_RATIO = 0.5;
    private static final int CNN_LAYER_SIZE = 25;
    private static final int LSTM_LAYER_SIZE = 512;
    private static final int DENSE_LAYER_SIZE = 32;
    private static final int TRUNCATED_BPTT_LENGTH = 22;

    public static MultiLayerNetwork buildHybridNetwork(int nIn, int nOut) {
        MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
            .seed(SEED)
            .iterations(ITERATIONS)
            .learningRate(LEARNING_RATE)
            .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
            .weightInit(WeightInit.XAVIER)
            .updater(Updater.RMSPROP)
            .regularization(true)

```



```

.l2(1e-4)
.list()
.layer(0, new Convolution1DLayer.Builder(1)
    .nIn(nIn)
    .nOut(CNN_LAYER_SIZE)
    .stride(1)
    .activation(Activation.RELU)
    .build())
.layer(1, new GlobalPoolingLayer.Builder(PoolingType.AVG)
    .build())
.layer(2, new GravesLSTM.Builder()
    .nIn(CNN_LAYER_SIZE)
    .nOut(LSTM_LAYER_SIZE)
    .activation(Activation.RELU)
    .gateActivationFunction(Activation.HARDSIGMOID)
    .dropOut(DROPOUT_RATIO)
    .build())
.layer(3, new GravesLSTM.Builder()
    .nIn(LSTM_LAYER_SIZE)
    .nOut(LSTM_LAYER_SIZE)
    .activation(Activation.RELU)
    .gateActivationFunction(Activation.HARDSIGMOID)
    .dropOut(DROPOUT_RATIO)
    .build())
.layer(4, new DenseLayer.Builder()
    .nIn(LSTM_LAYER_SIZE)
    .nOut(DENSE_LAYER_SIZE)
    .activation(Activation.RELU)
    .build())
.layer(5, new RnnOutputLayer.Builder()
    .nIn(DENSE_LAYER_SIZE)
    .nOut(nOut)
    .activation(Activation.IDENTITY)
    .lossFunction(LossFunctions.LossFunction.MSE)
    .build())
.inputPreProcessor(4, new RnnToFeedForwardPreProcessor())
.inputPreProcessor(5, new FeedForwardToRnnPreProcessor())
.backpropType(BackpropType.TruncatedBPTT)
.tBPTTForwardLength(TRUNCATED_BPTT_LENGTH)
.tBPTTBackwardLength(TRUNCATED_BPTT_LENGTH)
.pretrain(false)
.backprop(true)
.build();

```

```

MultiLayerNetwork net = new MultiLayerNetwork(conf);
net.init();
net.setListeners(new ScoreIterationListener(100));

```

```

return net;

```

```

    }
}

```

CryptoDataSetIterator.java

```

package com.crypto.advisor.service.prediction.predict;

import com.crypto.advisor.model.CryptoData;
import lombok.Getter;
import org.nd4j.linalg.api.ndarray.INDArray;
import org.nd4j.linalg.dataset.DataSet;
import org.nd4j.linalg.dataset.api.DataSetPreProcessor;
import org.nd4j.linalg.dataset.api.iterator.DataSetIterator;
import org.nd4j.linalg.factory.Nd4j;
import org.nd4j.linalg.primitives.Pair;

import java.util.*;

public class CryptoDataSetIterator implements DataSetIterator {

    private static final String NOT_IMPLEMENTED = "Not Implemented";

    private static final int PREDICT_LENGTH = 1;
    private static final int VECTOR_SIZE = 1;

    private final int miniBatchSize;
    private final int exampleLength;

    @Getter
    private final double min;
    @Getter
    private final double max;

    private final LinkedList<Integer> exampleStartOffsets = new LinkedList<>();
    @Getter
    private final List<Pair<INDArray, INDArray>> testDataSet;
    private final transient List<CryptoData> trainDataSet;

    public CryptoDataSetIterator(List<CryptoData> cryptoData, int miniBatchSize, int exampleLength,
double splitRatio) {
        this.min = calculateMin(cryptoData);
        this.max = calculateMax(cryptoData);
        this.miniBatchSize = miniBatchSize;
        this.exampleLength = exampleLength;
        int split = (int) Math.round(cryptoData.size() * splitRatio);
        trainDataSet = cryptoData.subList(0, split);
        testDataSet = generateTestDataSet(cryptoData.subList(split, cryptoData.size()));
        initializeOffsets();
    }

    private double calculateMin(List<CryptoData> cryptoData) {
        return cryptoData.stream()
            .min(Comparator.comparing(CryptoData::getPrice))
            .orElseThrow(NoSuchElementException::new)
            .getPrice();
    }
}

```

```

private double calculateMax(List<CryptoData> cryptoData) {
    return cryptoData.stream()
        .max(Comparator.comparing(CryptoData::getPrice))
        .orElseThrow(NoSuchElementException::new)
        .getPrice();
}

private List<Pair<INDArray, INDArray>> generateTestDataSet(List<CryptoData> cryptoData) {
    List<Pair<INDArray, INDArray>> testDs = new ArrayList<>();
    int window = exampleLength + PREDICT_LENGTH;

    for (int i = 0; i < cryptoData.size() - window; i++) {
        var input = Nd4j.create(new int[] {exampleLength, VECTOR_SIZE}, 'f');

        for (int j = i; j < i + exampleLength; j++) {
            var crypto = cryptoData.get(j);
            input.putScalar(new int[] {j - i, 0}, (crypto.getPrice() - min) / (max - min));
        }

        var crypto = cryptoData.get(i + exampleLength);
        var label = Nd4j.create(new int[] {1}, 'f');

        label.putScalar(new int[] {0}, crypto.getPrice());
        testDs.add(new Pair<>(input, label));
    }

    return testDs;
}

private void initializeOffsets() {
    exampleStartOffsets.clear();
    int window = exampleLength + PREDICT_LENGTH;
    for (int i = 0; i < trainDataSet.size() - window; i++) {
        exampleStartOffsets.add(i);
    }
}

@Override
public DataSet next(int num) {
    if (exampleStartOffsets.isEmpty()) {
        throw new NoSuchElementException();
    }

    int actualMiniBatchSize = Math.min(num, exampleStartOffsets.size());
    var input = Nd4j.create(new int[] {actualMiniBatchSize, VECTOR_SIZE, exampleLength}, 'f');
    var label = Nd4j.create(new int[] {actualMiniBatchSize, PREDICT_LENGTH}, 'f');

    for (int index = 0; index < actualMiniBatchSize; index++) {
        int startIdx = exampleStartOffsets.removeFirst();
        int endIdx = startIdx + exampleLength;
    }
}

```

```

CryptoData curData = trainDataSet.get(startIdx);
CryptoData nextData;

for (int i = startIdx; i < endIdx; i++) {
    int c = i - startIdx;

    input.putScalar(new int[] {index, 0, c}, (curData.getPrice() - min) / (max - min));
    nextData = trainDataSet.get(i + 1);

    if (i == endIdx - 1) {
        label.putScalar(new int[] {index, 0}, (nextData.getPrice() - min) / (max - min));
    }

    curData = nextData;
}

if (exampleStartOffsets.isEmpty()) {
    break;
}
}

return new DataSet(input, label);
}

@Override
public int totalExamples() {
    return trainDataSet.size() - exampleLength - PREDICT_LENGTH;
}

@Override
public int inputColumns() {
    return VECTOR_SIZE;
}

@Override
public int totalOutcomes() {
    return PREDICT_LENGTH;
}

@Override
public boolean resetSupported() {
    return false;
}

@Override
public boolean asyncSupported() {
    return false;
}

@Override
public void reset() {
    initializeOffsets();
}

```

```

    }

    @Override
    public int batch() {
        return miniBatchSize;
    }

    @Override
    public int cursor() {
        return totalExamples() - exampleStartOffsets.size();
    }

    @Override
    public int numExamples() {
        return totalExamples();
    }

    @Override
    public void setPreProcessor(DataSetPreProcessor dataSetPreProcessor) {
        throw new UnsupportedOperationException(NOT_IMPLEMENTED);
    }

    @Override
    public DataSetPreProcessor getPreProcessor() {
        throw new UnsupportedOperationException(NOT_IMPLEMENTED);
    }

    @Override
    public List<String> getLabels() {
        throw new UnsupportedOperationException(NOT_IMPLEMENTED);
    }

    @Override
    public boolean hasNext() {
        return !exampleStartOffsets.isEmpty();
    }

    @Override
    public DataSet next() {
        return next(miniBatchSize);
    }
}

```

CryptoPricePrediction.java

```

package com.crypto.advisor.service.prediction.predict;

import com.crypto.advisor.service.prediction.model.NeuralNetworkBuilder;
import com.crypto.advisor.model.CryptoData;
import lombok.AccessLevel;
import lombok.NoArgsConstructor;
import org.deeplearning4j.nn.multilayer.MultiLayerNetwork;

```

```

import org.deeplearning4j.util.ModelSerializer;
import org.nd4j.linalg.api.ndarray.INDArray;
import org.nd4j.linalg.primitives.Pair;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.io.File;
import java.io.IOException;
import java.util.List;

@NoArgsConstructor(access = AccessLevel.PRIVATE)
public final class CryptoPricePrediction {

    private static final Logger LOGGER = LoggerFactory.getLogger(CryptoPricePrediction.class);

    private static final int TS_LENGTH = 22; // time series length, 22 days/mo
    private static final int TRAINING_EPOCHS = 30;
    private static final int MINI_BATCH_SIZE = 64;
    private static final double SPLIT_RATIO = 0.9;

    public static double[] predict(List<CryptoData> cryptoData) throws IOException {
        var modelFile = new File("src/main/resources/crypto-price-model.zip");

        LOGGER.info("Create data set iterator...");
        var iterator = new CryptoDataSetIterator(cryptoData, MINI_BATCH_SIZE, TS_LENGTH,
SPLIT_RATIO);

        LOGGER.info("Load test dataset...");
        var testDataSet = iterator.getTestDataSet();

        MultiLayerNetwork net;

        boolean trainModel = false; // true -> train & use the model; false -> simply use the model
        if (trainModel) {
            LOGGER.info("Build hybrid CNN-LSTM network...");
            net = NeuralNetworkBuilder.buildHybridNetwork(iterator.inputColumns(),
iterator.totalOutcomes());

            LOGGER.info("Training...");
            for (int i = 0; i < TRAINING_EPOCHS; i++) {
                while (iterator.hasNext()) net.fit(iterator.next());
                iterator.reset();
                net.rnnClearPreviousState();
            }

            LOGGER.info("Saving model...");
            ModelSerializer.writeModel(net, modelFile, true);
        }

        LOGGER.info("Load model...");
        net = ModelSerializer.restoreMultiLayerNetwork(modelFile);
    }
}

```

```

    LOGGER.info("Testing...");
    double max = iterator.getMax();
    double min = iterator.getMin();

    return predictPriceOneAhead(net, testDataSet, max, min);
}

private static double[] predictPriceOneAhead(MultiLayerNetwork net, List<Pair<INDArray,
INDArray>> testData, double max, double min) {
    double[] predicts = new double[testData.size()];
    double[] actuals = new double[testData.size()];

    for (int i = 0; i < testData.size(); i++) {
        INDArray key = testData.get(i).getKey();
        INDArray input = key.reshape(key.shape()[0], 1, key.shape()[1]);

        predicts[i] = net.rnnTimeStep(input).getDouble(TS_LENGTH - 1) * (max - min) + min;
        actuals[i] = testData.get(i).getValue().getDouble(0);
    }

    LOGGER.info("Predicted, Actual");
    for (int i = 0; i < predicts.length; i++) {
        LOGGER.info("{} , {}", predicts[i], actuals[i]);
    }

    LOGGER.info("Done!");

    return predicts;
}
}

```

UserValidator.java

```

package com.crypto.advisor.service.validator;

import com.crypto.advisor.entity.User;
import com.crypto.advisor.service.dao.UserDao;
import lombok.NonNull;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Component;
import org.springframework.validation.Errors;
import org.springframework.validation.Validator;

@Component
@RequiredArgsConstructor
public class UserValidator implements Validator {

    private static final String ERROR_CODE = "error.user";

    private final UserDao userDao;

    @Override

```

```

public boolean supports(@NonNull Class<?> type) {
    return User.class.equals(type);
}

@Override
public void validate(@NonNull Object target, @NonNull Errors errors) {
    var user = (User) target;

    if (userDao.findByUsername(user.getUsername()).isPresent()) {
        errors.rejectValue("username", ERROR_CODE, "This username is already taken");
    }

    if (!user.getPassword().equals(user.getPasswordConfirmation())) {
        errors.rejectValue("passwordConfirmation", ERROR_CODE, "Password mismatch");
    }
}
}

```

AlphaVantageClient.java

```

package com.crypto.advisor.service;

import org.apache.http.NameValuePair;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.utils.URIBuilder;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Service;

import java.io.IOException;
import java.net.URISyntaxException;
import java.util.List;

@Service
public class AlphaVantageClient {

    private static final Logger LOGGER = LoggerFactory.getLogger(AlphaVantageClient.class);

    private final String url;
    private final String apiKey;

    public AlphaVantageClient(@Value("${alpha-vantage-api-url}") String url,
                              @Value("${alpha-vantage-api-key}") String apiKey) {
        this.url = url;
        this.apiKey = apiKey;
    }

    public String getHistoricalData(String function, String symbol) {

```



```

final List<NameValuePair> parameters = List.of(
    new BasicNameValuePair("function", function),
    new BasicNameValuePair("symbol", symbol),
    new BasicNameValuePair("market", "EUR"),
    new BasicNameValuePair("apikey", apiKey)
);

try {
    return makeApiCall(parameters);
} catch (Exception e) {
    LOGGER.error("Caught exception during API call: " + e.getMessage());
    return "";
}
}

private String makeApiCall(final List<NameValuePair> parameters) throws URISyntaxException,
IOException {
    String responseContent;

    var query = new URIBuilder(url);
    query.addParameters(parameters);

    var request = new HttpGet(query.build());

    try (var client = HttpClients.createDefault();
        var response = client.execute(request)) {

        var entity = response.getEntity();
        responseContent = EntityUtils.toString(entity);
        EntityUtils.consume(entity);
    }

    responseContent = FileUtils.readFileToString(
        new File("src/main/resources/mock-btc-av-response.json"),
        StandardCharsets.UTF_8
    );

    return responseContent;
}
}

```

CmcApiClient.java

```

package com.crypto.advisor.service;

import org.apache.http.HttpHeaders;
import org.apache.http.NameValuePair;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.utils.URIBuilder;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;

```

```

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Service;

import java.io.IOException;
import java.net.URISyntaxException;
import java.util.List;

@Service
public class CmcApiClient {

    private static final Logger LOGGER = LoggerFactory.getLogger(CmcApiClient.class);

    private static final String LISTINGS_LATEST_ENDPOINT = "/v1/cryptocurrency/listings/latest";

    private final String url;
    private final String apiKey;

    public CmcApiClient(@Value("${cmc-api-url}") String url,
                        @Value("${cmc-api-key}") String apiKey) {
        this.url = url;
        this.apiKey = apiKey;
    }

    public String getLatestListings() {
        final List<NameValuePair> parameters = List.of(
            new BasicNameValuePair("start", "1"),
            new BasicNameValuePair("limit", "10"),
            new BasicNameValuePair("convert", "USD")
        );

        try {
            return makeApiCall(url + LISTINGS_LATEST_ENDPOINT, parameters);
        } catch (Exception e) {
            LOGGER.error("Caught exception during API call: " + e.getMessage());
            return "";
        }
    }

    private String makeApiCall(final String uri, final List<NameValuePair> parameters) throws
    URISyntaxException, IOException {
        String responseContent;

        var query = new URIBuilder(uri);
        query.addParameters(parameters);

        var request = new HttpGet(query.build());
        request.setHeader(HttpHeaders.ACCEPT, "application/json");
        request.addHeader("X-CMC_PRO_API_KEY", apiKey);

        try (var client = HttpClient.createDefault());

```

```

        var response = client.execute(request)) {

        var entity = response.getEntity();
        responseContent = EntityUtils.toString(entity);
        EntityUtils.consume(entity);
    }

    return responseContent;
}
}

```

CryptoService.java

```

package com.crypto.advisor.service;

import com.crypto.advisor.model.CryptoStats;
import com.crypto.advisor.exception.CryptoNotFoundException;
import com.crypto.advisor.service.prediction.predict.CryptoPricePrediction;
import com.crypto.advisor.model.CryptoData;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.google.gson.*;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.io.IOException;
import java.math.BigDecimal;
import java.text.DecimalFormat;
import java.time.LocalDate;
import java.util.*;
import java.util.concurrent.atomic.AtomicBoolean;
import java.util.concurrent.atomic.AtomicInteger;
import java.util.stream.Collectors;
import java.util.stream.Stream;

@Service
@RequiredArgsConstructor
public class CryptoService {

    private static final Logger LOGGER = LoggerFactory.getLogger(CryptoService.class);

    private final CmcApiClient cmcApiClient;
    private final AlphaVantageClient avApiClient;

    public Set<CryptoStats> getCryptoStatistics() {
        var apiResponse = cmcApiClient.getLatestListings();
        var object = (JsonObject) JsonParser.parseString(apiResponse);
        var data = object.get("data");

        CryptoStats[] cryptoStats = new CryptoStats[0];
        try {

```

```

        cryptoStats = new ObjectMapper().readValue(data.toString(), CryptoStats[].class);
    } catch (Exception e) {
        LOGGER.error("Failed to process json. Reason: " + e.getMessage());
    }

    return Stream.of(cryptoStats)
        .sorted(Comparator.comparing(CryptoStats::getRank))
        .collect(Collectors.toCollection(LinkedHashSet::new));
}

public CryptoStats getCryptoStatisticsBySymbol(final String symbol) {
    return getCryptoStatistics().stream()
        .filter(c -> c.getSymbol().equals(symbol))
        .findFirst()
        .orElseThrow(() -> new CryptoNotFoundException(symbol));
}

public String getHistoricalAndPredictedData(final String function, final String symbol) {
    var apiResponse = avApiClient.getHistoricalData(function, symbol);
    var object = (JsonObject) JsonParser.parseString(apiResponse);

    JsonObject data;
    try {
        data = object.get("Time Series (Digital Currency Daily)").getAsJsonObject();
    } catch (Exception e) {
        LOGGER.error("Failed to get json object. Reason: " + e.getMessage());
        throw new IllegalArgumentException("Unexpected error occurred. Please wait and try again");
    }

    Map<String, String> histData = new LinkedHashMap<>();

    for (String currentKey : data.keySet()) {
        Object value = data.get(currentKey);

        if (value instanceof JsonObject) {
            var price = ((JsonObject) value).get("2. high").toString().replace("\"", "");
            var decimalFormat = new DecimalFormat("0.#####");
            price = decimalFormat.format(Double.valueOf(price));
            histData.put(currentKey, price);
        }
    }

    var predictedPrices = getPredictedPrices(histData, symbol);
    Map<String, BigDecimal> preparedData = new LinkedHashMap<>(predictedPrices);
    histData.forEach((key, value) -> preparedData.put(key, new BigDecimal(value)));

    return mapToJsonString(preparedData, histData.size());
}

private Map<String, BigDecimal> getPredictedPrices(Map<String, String> histData, String symbol) {
    List<CryptoData> cryptoData = new ArrayList<>();
    TreeMap<String, String> sortedHistData = new TreeMap<>(histData);

```

```

    sortedHistData.forEach((k, v) -> cryptoData.add(new CryptoData(k, symbol,
Double.parseDouble(v))));

    double[] predictedPrices;
    try {
        predictedPrices = CryptoPricePrediction.predict(cryptoData);
    } catch (IOException e) {
        throw new IllegalStateException("Failed to predict prices", e);
    }

    var currentDate = LocalDate.now();
    var daysCount = new AtomicInteger(predictedPrices.length);

    return Arrays.stream(predictedPrices)
        .boxed()
        .map(BigDecimal::new)
        .collect(Collectors.toMap(
            x -> currentDate.plusDays(daysCount.getAndDecrement()).toString(),
            price -> price,
            (x, y) -> y,
            LinkedHashMap::new)
        );
}

private String mapToJsonString(Map<String, BigDecimal> data, int histDataSize) {
    var sb = new StringBuilder();
    var counter = new AtomicInteger(data.size() - histDataSize);
    var certainty = new AtomicBoolean(false);
    var rowFormat = "{\n\"c\":[" +
        "{\n\"v\": \"%s\", \"f\": null}, {\n\"v\": %s, \"f\": null}, {\n\"v\": %s}]]",
        "\n\"rows\": ["
    );

    data.forEach((date, price) -> {
        if (counter.getAndDecrement() == 0) {
            certainty.set(true);
        }
        sb.append(String.format(rowFormat, date, price, certainty));
    });

    sb.deleteCharAt(sb.length() - 1);
    sb.append("]}");

    return sb.toString();
}
}

```

ДОДАТОК Г. Ілюстративна частина

**РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ВЕБ-СЕРВІСУ РЕКОМЕНДАЦІЙ ПО
ТОРГІВЛІ КРИПТОВАЛЮТОЮ З ВИКОРИСТАННЯМ ОПТИМІЗОВАНИХ
ГІБРИДНИХ МОДЕЛЕЙ НЕЙРОННИХ МЕРЕЖ**

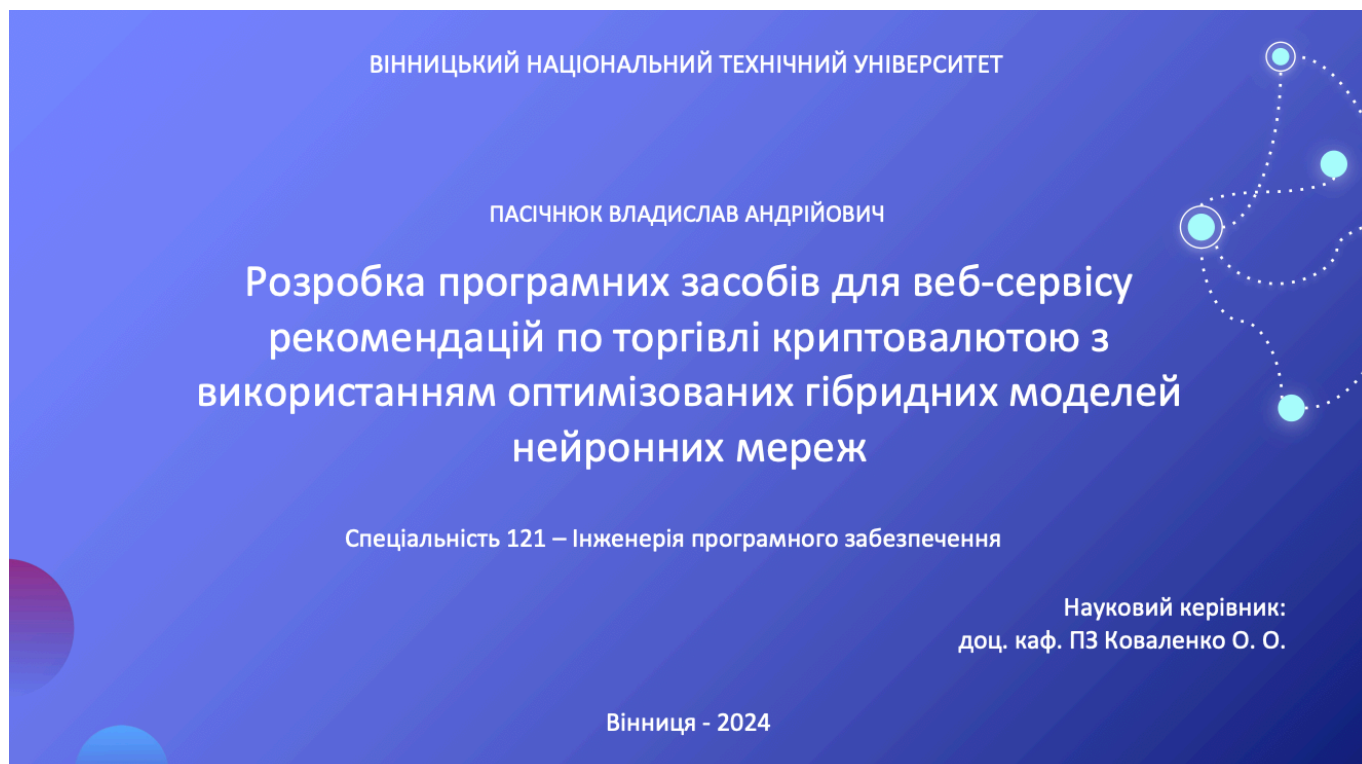


Рисунок Г.1 – Назва роботи

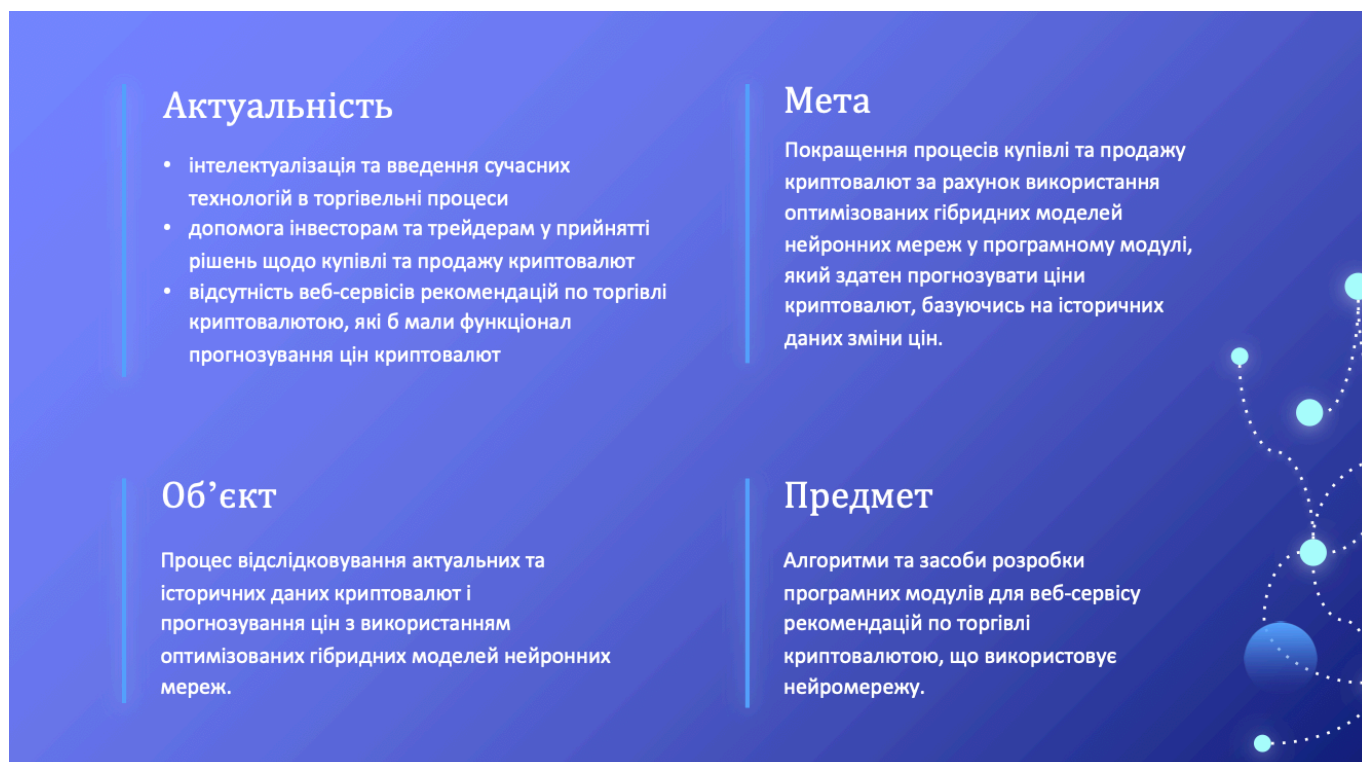


Рисунок Г.2 – Актуальність, мета, об'єкт і предмет дослідження

Задачі дослідження

- проаналізувати стан веб-сервісів рекомендацій по торгівлі криптовалютою
- визначити найефективніший підхід для прогнозування цін криптовалют
- Розробити методи створення функціоналу прогнозування цін криптовалют
- розробити алгоритм отримання та процесингу історичних даних зміни цін криптовалют
- створити алгоритм прогнозування ціни криптовалюти
- провести конфігурування гібридної моделі нейронної мережі
- здійснити оптимізацію вхідних параметрів в конфігурації гібридної моделі нейронної мережі
- розробити інтерфейс користувача
- розробити програмний модуль веб-сервісу рекомендацій по торгівлі криптовалютою з використанням оптимізованих гібридних моделей нейронних мереж
- провести варіантний аналіз засобів та середовищ розробки
- проаналізувати методи тестування програмного забезпечення
- протестувати програму для виявлення потенційних проблем

Рисунок Г.3 – Задачі дослідження

Основні етапи роботи програми



Рисунок Г.4 – Основні етапи роботи програми



Рисунок Г.5 – Основні етапи процесу прогнозування



Рисунок Г.6 – Конфігурація гібридної моделі нейромережі

Загальний алгоритм роботи

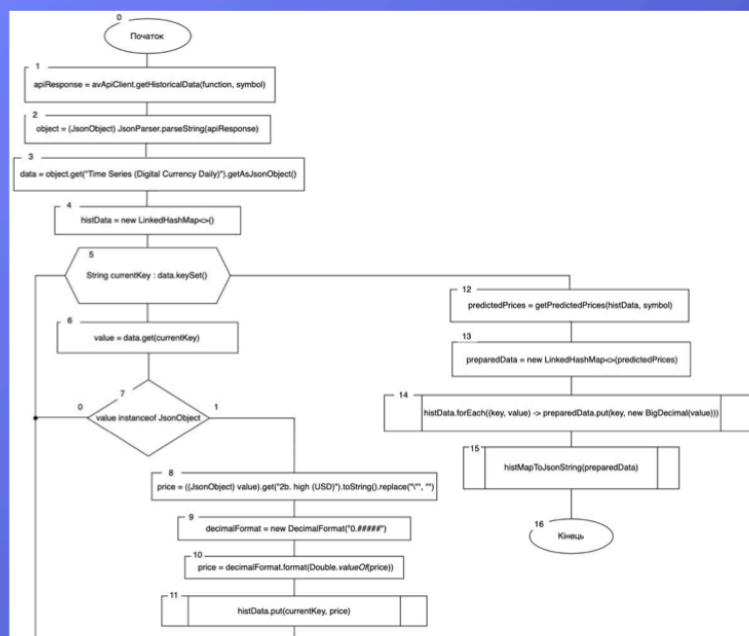


Рисунок Г.7 – Блок-схема загального алгоритму роботи

Алгоритм прогнозування

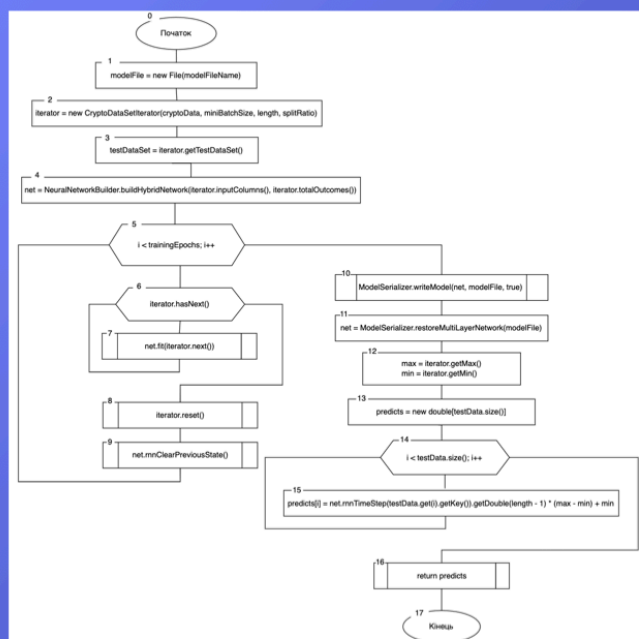


Рисунок Г.8 – Блок-схема алгоритму прогнозування цін криптовалюти

Вигляд інтерфейсу домашньої сторінки



Рисунок Г.9 – Вигляд інтерфейсу домашньої сторінки

Вигляд інтерфейсу сторінки криптовалюти



Рисунок Г.10 – Вигляд інтерфейсу сторінки криптовалюти

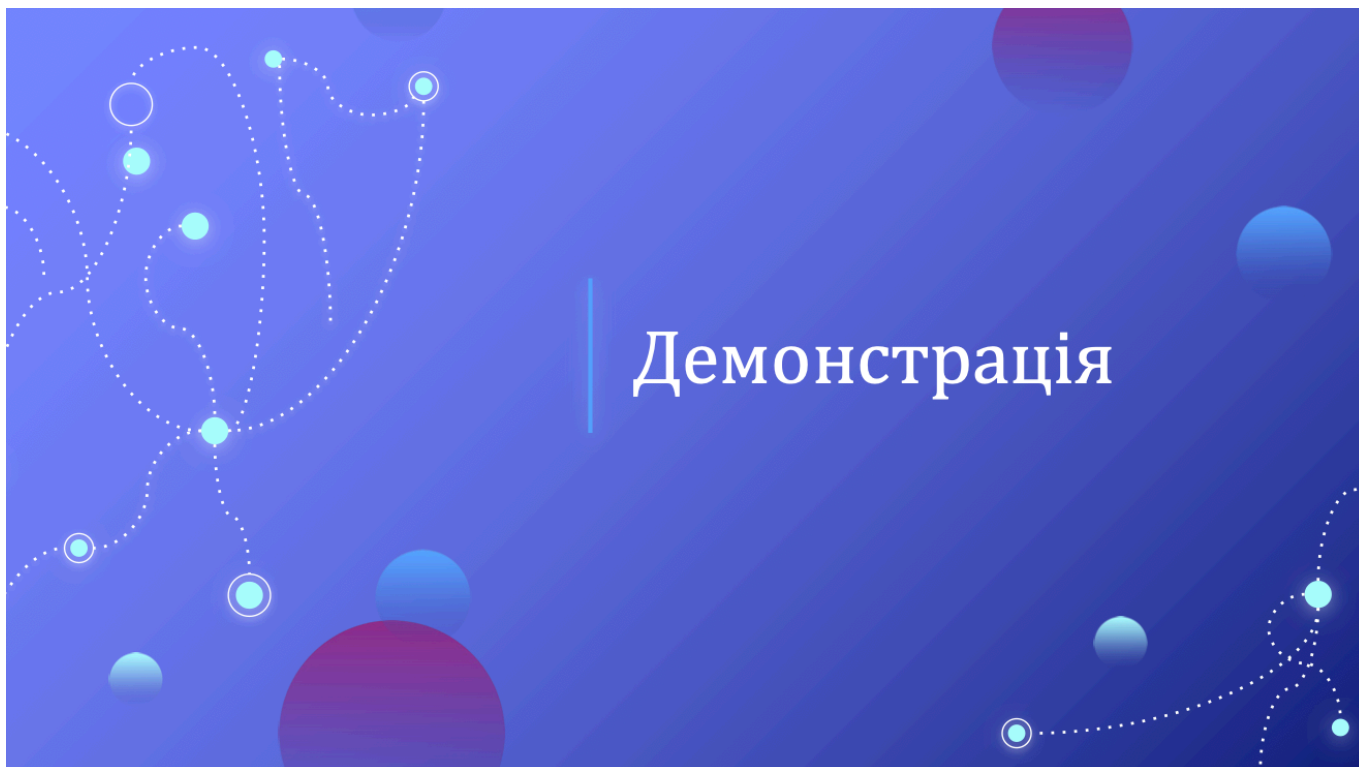


Рисунок Г.11 – Демонстрація

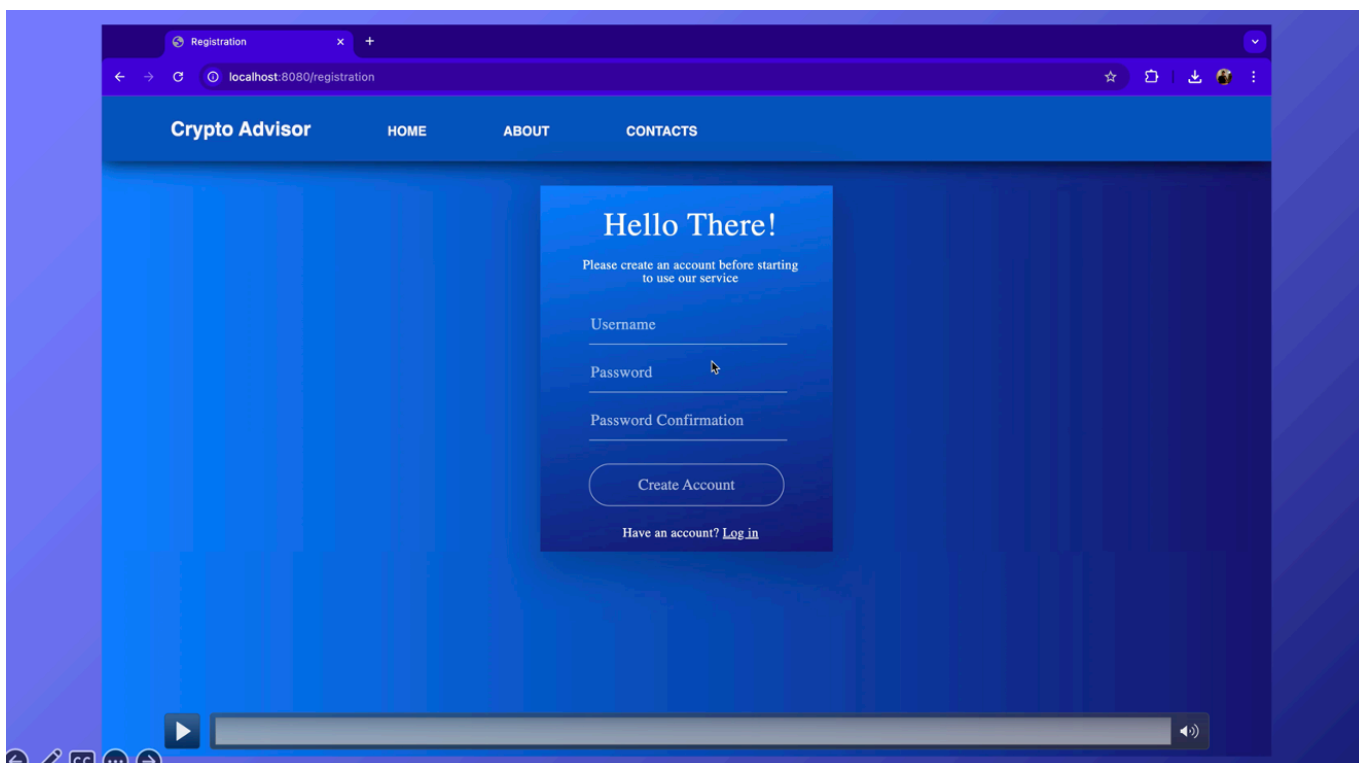


Рисунок Г.12 – Відео демонстрації

Наукова новизна

- Удосконалено метод створення функціоналу прогнозування цін криптовалют шляхом розробки підходу, який, на відміну від існуючих, передбачає використання нейронної мережі та візуалізацію результатів у форматі лінійного горизонтального графіка, що забезпечило зручний спосіб отримання орієнтовних майбутніх цін криптовалют, підвищуючи доступність і наочність прогнозів.
- Покращено метод конфігурації моделей гібридних нейронних мереж, що вирізняється інтеграцією рекурентних (LSTM) і згорткових (CNN) шарів із використанням шарів активації (ReLU) та глобального усереднення. Додатково оптимізовано вхідні параметри моделі, що дозволило ефективно застосовувати її для прогнозування фінансових показників.

Рисунок Г.13 – Наукова новизна

Практична цінність

Практична цінність одержаних результатів полягає в тому, що на основі отриманих в магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби для прогнозування цін криптовалют з використанням оптимізованих гібридних моделей нейронних мереж.

За темою магістерської кваліфікаційної роботи надруковано 3 наукові праці.

Рисунок Г.14 – Практична цінність

Апробації

- XVII Міжнародна науково-практична конференція «Інформаційні технології і автоматизація – 2024» (2024)
- XXIII Всеукраїнська науково-технічна конференція молодих вчених, аспірантів та студентів (2023)
- Молодь в науці: дослідження, проблеми, перспективи (2023)

Рисунок Г.15 – Апробації

Дякую за увагу!

Рисунок Г.16 – Завершення презентації