

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційно-аналітична система адміністрування веб-додатку з
візуалізацією даних»

Виконала: студент 2-го курсу, групи ІКІ-21м
спеціальності 123 — Комп'ютерна інженерія

Шкуренко А.В.
(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ОТ

Войцеховська О. В.
(прізвище та ініціали)

«16» 12 2022 р.

Опонент: к.т.н., доц. каф. ПЗ

Рейда О. М.
(прізвище та ініціали)

«19» 12 2022 р.

Допущено до захисту

Завідувач кафедри ОТ

Азаров О.Д.
(прізвище та ініціали)

«19» 12 2022 р

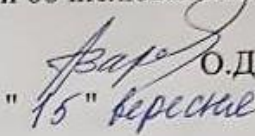
Вінниця 2022

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітній рівень — магістр
Спеціальність — 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри обчислювальної техніки

 О.Д. Азаров
" 15 " вересня 2022 р.

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Студентці Шкуренко Аліні Віталіївні

1 Тема роботи «Інформаційно-аналітична система адміністрування веб-додатку з візуалізацією даних» керівник роботи Войцеховська О. В. к.т.н., доц. каф. ОТ, затверджені наказом вищого навчального закладу від 15.09.2022 року № 205-А.

2 Строк подання студентом 09.09.22

3 Вихідні дані до роботи: контент для наповнення веб-додатку; сучасні методи розробки веб-додатків; методи веб-аналітики і візуалізації даних; види інформаційно-аналітичних систем.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): огляд інформаційно-аналітичних систем та технологій для їх побудови; аналіз методів розробки інформаційно-аналітичної системи; проектування та тестування роботи інформаційно-аналітичної системи адміністрування веб-додатку; економічна частина.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): структурна схема інтернет-магазину; вигляд сторінок інтернет-

магазину; UML-діаграма варіантів використання інтернет-магазину; структурна схема інформаційно-аналітичної системи; вигляд інформаційно-аналітичної системи адміністрування веб-додатку; схема бази даних.

6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3,4	Войцеховська О.В., к.т.н., доцент каф. ОТ	<i>[Signature]</i>	<i>[Signature]</i>
5	Небава М.І., к.е.н., професор каф. ЕПВМ	<i>[Signature]</i>	<i>[Signature]</i>

7 Дата видачі завдання

8 Календарний план виконання МКР приведений в таблиці 2

Таблиця 2 — Календарний план

№ з/п	Назва етапів МКР	Строк виконання	Підпис
1	Постановка задач роботи	09.09.22	<i>[Signature]</i>
2	Огляд та аналіз сучасних технологій побудови інформаційних систем	10.09.22-15.09.22	<i>[Signature]</i>
3	Проведення порівняльного аналізу аналогів інформаційно-аналітичних систем	16.09.22-18.09.22	<i>[Signature]</i>
4	Огляд технологій для розробки інформаційно-аналітичної системи	19.09.22-21.09.22	<i>[Signature]</i>
5	Розробка структури, дизайну для веб-додатку та інформаційно-аналітичної системи	22.09.22-10.10.22	<i>[Signature]</i>
6	Програмна реалізація окремих сторінок веб-додатку	11.10.22-15.11.22	<i>[Signature]</i>
7	Програмна реалізація інформаційно-аналітичної системи	16.11.22-20.11.22	<i>[Signature]</i>
8	Підготовка матеріалів для опису проектування інформаційно-аналітичної системи	20.11.22-24.11.22	<i>[Signature]</i>
9	Оформлення пояснювальної записки	25.11.22-12.12.22	<i>[Signature]</i>
10	Перевірка якості виконання магістерської роботи	21.12.22	<i>[Signature]</i>

Студент

[Signature]

Шкуренко Аліна Віталіївна

Керівник

[Signature]

к.т.н., доц. Войцеховська Олена Валеріївна

Консультант з економічної частини

[Signature]

к.е.н., професор Небава Микола

Іванович

АНОТАЦІЯ

УДК 004.4

Шкуренко А.В. Інформаційно-аналітична система адміністрування веб-додатку з візуалізацією даних. Магістерська кваліфікаційна робота зі спеціальності 123 — Комп'ютерна інженерія, освітня програма — комп'ютерна інженерія. Вінниця: ВНТУ, 2022. 112 с.

Укр. мовою. Бібліогр.: 35 назв; рис. 32; табл.: 16

Магістерська дипломна робота присвячена проектуванню інформаційно-аналітичної системи адміністрування веб-додатку за допомогою таких технологій як Angular, Bootstrap, SQL, C#.

Під час виконання даної роботи було проведено аналіз поняття інформаційно-аналітичних систем, проведено порівняльний аналіз аналогів та розглянуто сучасні технології для проектування системи адміністрування. Створено структуру інформаційно-аналітичної системи адміністрування веб-додатку, проаналізовано методи створення статистики на основі отриманих даних з веб-додатку. Також розглянуто засоби для візуального представлення статистики.

Виконано програмну реалізацію інформаційно-аналітичної системи. Проведено тестування роботи інформаційно-аналітичної системи, наведено результати візуалізації даних, отриманих на основі розглянутих методів.

Ключові слова: веб-аналітика, інформаційно-аналітична система, система адміністрування, візуалізація даних.

ABSTRACT

UDC 004.4

A. V. Shkurenko Software tool for secure containerization of a web application. Master's thesis in specialty 123 — computer engineering, educational program — computer engineering. Vinnytsia: VNTU, 2022.

The master's thesis is devoted to the design of an information and analytical system for the administration of a web application using such technologies as Angular, Bootstrap, SQL, C#.

During the execution of this work, an analysis of the concept of information and analytical systems was carried out, a comparative analysis of analogues was carried out, and modern technologies for the design of the administration system were considered. The structure of the information and analytical system of web application administration was created, the methods of creating statistics based on the data obtained from the web application were analyzed. Means for visual presentation of statistics are also considered.

The software implementation of the information and analytical system has been completed. The information and analytical system was tested, the data visualization results obtained on the basis of the considered methods are given.

Keywords: web analytics, information and analytical system, administration system, data visualization.

ЗМІСТ

ВСТУП		8
1 ОГЛЯД ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ ДЛЯ ЇХ ПОБУДОВИ		10
1.1 Поняття інформаційно-аналітичної системи		10
1.2 Інформаційно-аналітичні системи адміністрування та web-аналітики.....		12
1.3 Аналіз технологій для побудови інформаційно-аналітичної системи		15
1.3.1 Огляд технологій для реалізації серверної частини.....		15
1.3.2 Огляд технологій для створення бази даних.....		17
1.3.3 Огляд технологій для реалізації клієнтської частини		19
1.4 Порівняльний аналіз аналогів інформаційно-аналітичних систем з можливістю візуалізації даних		22
2 АНАЛІЗ МЕТОДІВ РОЗРОБКИ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ		26
2.1 Аналіз методів обробки отриманих даних з веб-додатку та розрахунок статистики.....		26
2.1.1 Метод сегментації клієнтів		27
2.1.2 Метод сегментації товарів.....		29
2.1.3 Метод для побудови прогнозів майбутніх продаж		30
2.2 Засоби для візуального представлення даних.....		35
3 ПРОЕКТУВАННЯ ТА ТЕСТУВАННЯ РОБОТИ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ АДМІНІСТРУВАННЯ ВЕБ-ДОДАТКУ		39
3.1 Вибір технологій для реалізації інформаційно-аналітичної системи.....		39
3.2 Розробка структури веб-додатку для отримання даних.....		40
3.3 Розробка структури інформаційно-аналітичної системи для адміністрування веб-додатку		43

						08-23.МДР.016.00.000 ПЗ		
Змн.	Лист	№ докум.	Підпис	Дата				
Розроб.		Шкуренко А.В.			Інформаційно-аналітична система адміністрування веб- додатку з візуалізацією даних. Пояснювальна записка	Лім.	Арк.	Акрушів
Перевір.		Войцеховська О.В.					6	112
Реценз.		Рейда О.М.				ВНТУ, гр. ІКІ-21м		
Н. Контр.		Швець С.І.						
Затверд.		Азаров О.Д.						

3.4 Реалізація інформаційно-аналітичної системи адміністрування веб-додатку	45
3.4.1 Реалізація клієнтської частини	45
3.4.2 Налаштування маршрутизації засобами Angular.....	46
3.4.3 Інтеграція бази даних в інформаційно-аналітичну систему.....	47
3.4.4 Реалізація авторизації в інформаційно-аналітичній системі.....	52
3.5 Програмна реалізація окремих модулів інформаційно-аналітичної системи адміністрування веб-додатку	54
3.5.2 Реалізація сторінки «Dashboard»	54
3.5.3 Реалізація сторінки «Statistics&Analysis».....	58
3.6 Тестування інформаційно-аналітичної системи	60
4 ЕКОНОМІЧНА ЧАСТИНА	67
4.1 Комерційний та технологічний аудит науково-технічної розробки	67
4.2 Прогнозування витрат на виконання науково-дослідної роботи.....	70
4.2.1 Розрахунок основної заробітної плати розробників	70
4.2.2 Розрахунок додаткової заробітної плати розробників	71
4.2.3 Нарахування на заробітну плату розробників	71
4.2.4 Розрахунок витрат на матеріали.....	71
4.2.5 Амортизація обладнання	71
4.2.6 Витрати на електроенергію	73
4.2.7 Інші витрати та загальновиробничі витрати	73
4.2.8 Витрати на проведення науково-дослідної роботи	74
4.2.9 Розрахунок загальних витрат на науково-дослідну роботу та оформлення її результатів	74
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	75
4.3.1 Розробка програмного засобу для використання масовим споживачем.....	75
4.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності	76

ВИСНОВКИ	80
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	82
ДОДАТОК А Технічне завдання	86
ДОДАТОК Б Структурна схема інтернет-магазину	89
ДОДАТОК В Вигляд сторінок інтернет-магазину	90
ДОДАТОК Г UML-діаграма використання інтернет-магазину.....	94
ДОДАТОК Д Структурна схема інформаційно-аналітичної системи.....	95
ДОДАТОК Е Вигляд інформаційно-аналітичної системи адміністрування веб- додатку	96
ДОДАТОК Ж Схема бази даних	98
ДОДАТОК И Лістинг коду маршрутизації Angular в інформаційно-аналітичній системі.....	99
ДОДАТОК К Лістинг коду розмітки сторінок інформаційно-аналітичної системи	101
ДОДАТОК Л Лістинг коду методів аналізу отриманих даних	105
ДОДАТОК М Лістинг коду сторінок адміністративної панелі.....	110
ДОДАТОК Н Протокол перевірки навчальної (кваліфікаційної) роботи.....	112

ВСТУП

Загострення конкуренції в умовах глобалізації та відкритості економіки зумовлює необхідність ефективно використовувати інноваційний потенціал підприємства, для чого потрібні нові управлінські технології, реалізація яких можлива на основі застосування автоматизованих інформаційних систем. Діяльність високорозвинених економічних суб'єктів у регіонах світу з сформованою розвиненою ринковою економікою, досягнення помітного темпу зростання прибутку дедалі більше пов'язується зі складною аналітичною роботою, використанням нових досягнень у сфері інформаційних технологій.

У сучасному діловому світі в будь-яких сферах діяльності обсяги інформації, з якими доводиться стикатися організаціям, просто колосальні. І від того, якою мірою організація здатна витягти максимум із наявної в її розпорядженні інформації, залежить успіх. Запорука успіху — у побудові ефективної інформаційно-аналітичної системи (ІАС).

Призначенням будь-якої сучасної інформаційно-аналітичної системи (ІАС) є забезпечення керівників, аналітиків та менеджерів інформацією у всіх аспектах діяльності компанії для її подальшої оцінки та аналізу.

Використання в інфраструктурі підприємства інформаційно-аналітичної системи пояснюється низкою причин: прагненням загальної реорганізації бізнес-процесів, бажанням підвищити якість ділової інформації, необхідністю підтримки стратегічного планування та досягнення високоефективних рішень.

На сьогоднішній день існує безліч спеціалізованих комп'ютерних систем для керування даними. Існуючі інформаційні системи для управління даними різних веб-додатків охоплюють різновид всіх поставлених перед ними завдань для окремої інфраструктури або організації та мають велику кількість недоліків і плюсів для кожної організації відповідно.

Тому **метою** даної магістерської роботи є створення інформаційно-аналітичної системи для адміністрування веб-додатку.

Для досягнення цієї мети були поставлені такі завдання:

- провести огляд та аналіз існуючих ІАС для адміністрування веб-додатків;
- проаналізувати методи та засоби обробки інформаційних даних;
- розробити структуру інформаційно-аналітичної системи адміністрування веб-додатку;
- розробити структуру інтернет-магазину для отримання даних для ІАС;
- обґрунтувати вибір технологій реалізації ІАС;
- виконати програмну реалізацію веб-додатку та інформаційно-аналітичної системи для адміністрування веб-сайту;
- провести тестування розробленої системи.

Об’єктом дослідження є процес адміністрування веб-сайту кондитерської продукції з можливістю візуалізації даних.

Предметом дослідження є методи та програмні засоби візуалізації даних на основі інформаційних даних, отриманих з розробленого веб-сайту кондитерської продукції.

Наукова новизна полягає у вдосконаленні системи візуалізації даних в інтернет ресурсі за рахунок застосування сегментації товарів та користувачів, що додасть більше функціональних можливостей для аналізу даних адміністратору веб-додатку та дозволить прогнозувати майбутні обсяги виробництва.

Практичне значення отриманих результатів магістерської роботи полягає в створенні програмного засобу для адміністрування веб-додатку з можливістю візуалізації даних.

Апробація результатів бакалаврської роботи: опубліковано доповіді на І науково-технічній конференції підрозділів Вінницького національного технічного університету та Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2023)».

1 ОГЛЯД ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ ДЛЯ ЇХ ПОБУДОВИ

1.1 Поняття інформаційно-аналітичної системи

Інформаційно-аналітичні системи визначаються як особливий клас інформаційних систем, призначених для аналітичної обробки даних. Інформаційно-аналітичні системи об'єднують, аналізують і зберігають як єдине ціле інформацію, яка вилучена як з баз даних, так і із зовнішніх джерел. Сховища даних, що входять до складу інформаційно-аналітичних систем, забезпечують перетворення великих об'ємів деталізованих даних в узагальнену вивірену інформацію, яка придатна для прийняття обґрунтованих рішень. На відміну від звичайних баз даних сховища містять оброблене, впорядковане і зрозуміле керівникам представлення даних [1].

Технологія і методика створення інформаційно-аналітичних систем охоплює такі види діяльності [2]:

— збір, аналіз і деталізацію вимог до інформаційно-аналітичної системи, визначення пріоритетів реалізації цих вимог і постановка завдань з їх реалізації, визначення вимог щодо архітектури, надійності і захисту від несанкціонованого доступу і визначення складу даних;

— розробка проектних рішень за всіма аспектами побудови інформаційно-аналітичної системи, визначення складу джерел інформації, способів передачі даних, складу додатків організації доступу до даних, проектування архітектури, проектування баз даних;

— розробка аналітичних додатків, вибір і настроювання інструментальних засобів збору, перетворення і очищення даних і організації доступу користувачів до даним, розробка метаданих, тестування, розробка документації для користувачів.

У теорії та практиці створення інформаційних систем [3, 4] виокремлюють три підходи до розробки ІС: локальний, глобальний та системний.

Особливістю використання локального підходу є поступове розширення можливостей інформаційної системи на вирішення завдань, що виникають перед системою управління. До переваг локального підходу можна віднести простоту організації та управління створенням системи, досить швидке виконання поставлених завдань, участь у вирішенні яких беруть невелика кількість розробників. Однак він має особливо істотний недолік. Інформаційні потоки та функціональна надмірність, їх дублювання та конфліктність не дозволяють правильно організувати та забезпечити раціональне вирішення складних завдань, що потребують постійної перебудови системи. Це призводить до дискредитації ідеї створення інформаційної системи. За такого підходу неможливо науково обґрунтувати вибір напрямів подальшого розвитку інформаційних систем, ні змодельовати їх, ні оцінити наслідки їх використання.

За умови використання глобального підходу спочатку розробляється проект всієї системи, потім сама система і потім впроваджується. Зазвичай, особливо в сучасній ситуації стрімкого розвитку технологій, такий підхід призводить до оновлення проекту ще до його реалізації, оскільки час його розробки перевищує період оновлення використовуваних для цього технологій та засобів.

Особливостями створення інформаційних систем з використанням системного підходу є комплексне вивчення об'єктів як єдиного цілого, примусове виділення їх основних частин у вигляді цілісних систем, обов'язковий поділ цих систем на їх взаємодії. Відповідно до системного підходу об'єкт є сукупність взаємозалежних елементів однієї складної динамічної системи, що у стані постійного зміни, викликаного низкою чинників, переважно пов'язані з узгодженням процесів перетворення вхідних ресурсів до вихідних.

Інформаційно-аналітичні системи можна розподілити на такі класи [5]:

- комплексні системи управління підприємствами (ERP-системи);
- табличні процесори;
- пакети для рішення задач фундаментального аналізу;
- пакети для рішення задач технічного аналізу;

- статистичні і математичні пакети програм;
- системи штучного інтелекту;
- веб-додатки для відстеження веб-аналітики.

Інформаційно-аналітичні системи сьогодні, це надійний та універсальний помічник у галузі управління інформацією та підтримки прийняття управлінських рішень, розраховані на користувачів з будь-яким рівнем підготовки на будь-якому рівні управління та для будь-якого інформаційно-комп'ютерного середовища.

Основним завданням інформаційно-аналітичних систем залишається багатокритеріальна оцінка для можливості повноцінного порівняння та вибору оптимального варіанти із сукупності можливих варіантів рішень. Інформаційно-аналітична система представляється як інтегрована система, призначена для допомоги керівникам різних рівнів управління, з метою оптимізації каналів збору інформації, що склалися, та забезпечення більш повного задоволення інформаційних потреб керівників. Застосування системи дозволяє ефективно розподілити складові елементи процесу підготовки та ухвалення управлінського рішення між усіма учасниками управлінської діяльності. Широке застосування на практиці пов'язано з функціональними можливостями інформаційно-аналітичної системи, які дозволяють приймати рішення не лише у разі їх підготовки самим керівником за відсутності об'єктивних даних про характеристики порівнюваних варіантів, але і у випадках підготовки рішень, як керівниками, так та спеціалістами, з обробкою великих обсягів інформації. Таким чином, застосування інформаційно-аналітичних систем — це умова, без якої неможливо ефективний розвиток будь-якої організації [6].

1.2 Інформаційно-аналітичні системи адміністрування та web-аналітики

Метою аналізу даних є знання про об'єкт дослідження — виявлення корисної інформації, знайдення висновків, врешті, розумне прийняття рішень. Аналіз даних може мати багато аспектів та підходів, реалізовуватися за допомогою різних інструментів — в тому числі математичних, статистичних, за

допомогою різноманітних способів візуалізації, але головне завдання його — стиснення інформації.

Існування сайту та його повноцінна робота неможливі без ряду регулярних заходів з його підтримки, а саме наповнення сайту новим контентом, редагування існуючих матеріалів та своєчасного оновлення необхідних компонентів і модулів, захисту від мережових атак, резервного копіювання даних, відстеження аналітики. Весь комплекс подібних робіт — це ніщо інше, як адміністрування сайту.

Адміністрування сайту має здійснюватися на постійній основі, адже відвідувачам цікавий «живий» ресурс, що містить актуальну інформацію та виконує всі заявлені функції. До завдань адміністратора сайту може входити його інформаційна і технічна підтримка, а також роботи з модернізації, які не потребують кардинальних змін дизайну або функціональності сайту. Також адміністратор має відслідковувати відвідуваність сайту, визначати поведінку користувачів в ньому. Тому в багатофункціональних інформаційно-аналітичних системах є web-аналітика.

Web-аналітика — це вимірювання, збір, аналіз, подання та інтерпретація інформації про відвідувачів сайту з метою покращення його наповнення, вирішення нових завдань та виявлення нових ринків збуту.

Основним завданням web-аналітики є моніторинг відвідуваності сайту, на підставі даних, за допомогою яких визначається аудиторія і вивчається поведінка відвідувачів для прийняття рішень з розвитку і розширення функціональних можливостей інтернет-ресурсу. Завдяки даним, отриманим під час веб-аналітики, можна визначити аудиторію сайту, її особливості та переваги, поведінку відвідувачів. Ця ж інформація, в свою чергу, є основним критерієм для підбору методів розвитку ресурсу, зміни його функціональних можливостей і навіть зовнішнього вигляду. Результати веб-аналітики можуть кардинально вплинути на напрямки сайту, його контент і навіть тематику [7].

Розрізняють такі методи web-аналізу [8]:

Аналіз відвідуваності. Вивчення активності користувачів на сайті, число унікальних та повторних візитів, визначення можливих причин спадання чи підвищення активності відвідуваності.

Аналіз цільової аудиторії. Це географія, пристрої та операційні системи, які використовуються, інтереси та демографічні дані.

Аналіз джерел трафіку. Звідки на сайт прийшли користувачі, які канали залучення продемонстрували найбільшу ефективність, яка ціна залучення одного відвідувача, завдяки яким оголошенням чи ключовим фразам прийшли відвідувачі, на які сторінки потрапили.

Аналіз юзабіліті. Вивчення карт прокрутки сторінки, кліків, найбільш популярних сторінок та маршрутів на сайті.

Аналіз технічних недоліків. Пошук помилок, низька швидкість завантаження, наявність дубльованих сторінок тощо.

Аналіз даних e-commerce. Найбільш популярні товари, кількість транзакцій, поведінка користувачів на сайті, цикл продажів, дзвінки, середній чек, дохід.

Етапи проведення web-аналітики зображені на рисунку 1.1.

Етапи web-аналітики

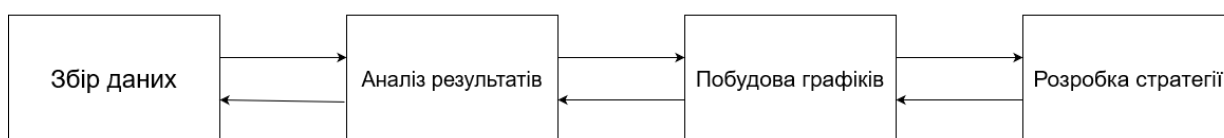


Рисунок 1.1 — Етапи web-аналітики

На основі такого аналізу можна створювати статистику і вивантажувати відповідні звіти.

Статистика являє собою строгий науковий метод, що дозволяє зрозуміти дані, дійти до їх суті. Статистика може дати миттєву загальну картину даних — на основі доступної для сприйняття візуалізації або числового узагальнення — незалежно від кількості спостережень чи одиниць даних. Окрім узагальнення

даних, важливим завданням статистики є формулювання логічних висновків та передбачення залежностей між змінними [9].

1.3 Аналіз технологій для побудови інформаційно-аналітичної системи адміністрування веб-додатку

1.3.1 Огляд технологій для реалізації серверної частини

Одним з найважливіших етапів створення веб-додатку є розробка функціональності, логіки веб-сторінки та маніпулювання даними.

C# — це об'єктно-орієнтована мова програмування загального призначення, розроблена Microsoft. Випущений у 2000 році C# залишається основною технологією для створення програм у Windows.

Програми, написані на C#, використовують середовище виконання .NET, бібліотеки класів і фактичну структуру .NET, тому обидві технології часто вважаються нероздільними. Зараз .NET є платформою та системою програмування для кросплатформної розробки [10].

За допомогою мови програмування C# можна розробляти різні типи захищених і надійних програм:

- програми windows;
- веб-додатки
- розподілені програми
- програми веб-служб
- програми баз даних тощо.

C# схвалено як стандарт ECMA та ISO. C# розроблено для CLI (Common Language Infrastructure). CLI — це специфікація, яка описує виконуваний код і середовище виконання.

WEB API ASP.NET — це поширена структура для створення служб на основі HTTP, до яких можна отримати доступ у різних програмах на різних платформах, таких як Інтернет, Windows, мобільні пристрої тощо (рисунок 1.2). Він працює більш-менш так само, як веб-програма ASP.NET MVC, за винятком

що він надсилає дані як відповідь замість перегляду html. Це як веб-служба або служба WCF, але винятком є те, що вона підтримує лише протокол HTTP. [11]

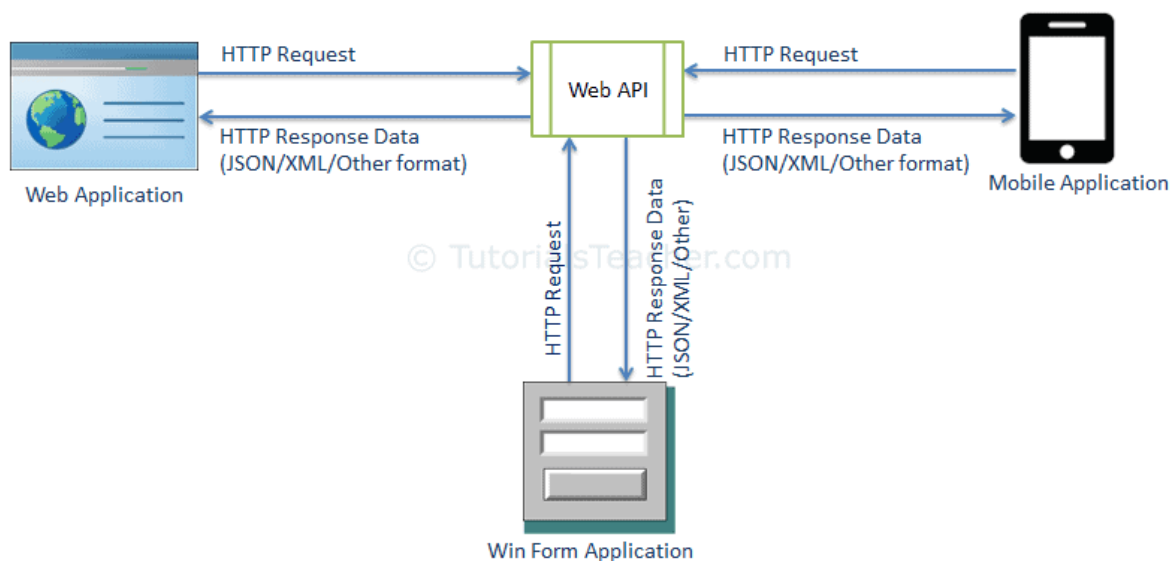


Рисунок 1.2 — Структурна схема Web API

Model-View-Controller (MVC) — це архітектурний шаблон, який розділяє програму на три основні логічні компоненти: модель, представлення та контролер (рисунок 1.3). Кожен із цих компонентів розроблено для обробки конкретних аспектів розробки програми. MVC є однією з найбільш часто використовуваних систем веб-розробки галузевих стандартів для створення масштабованих і розширюваних проєктів.

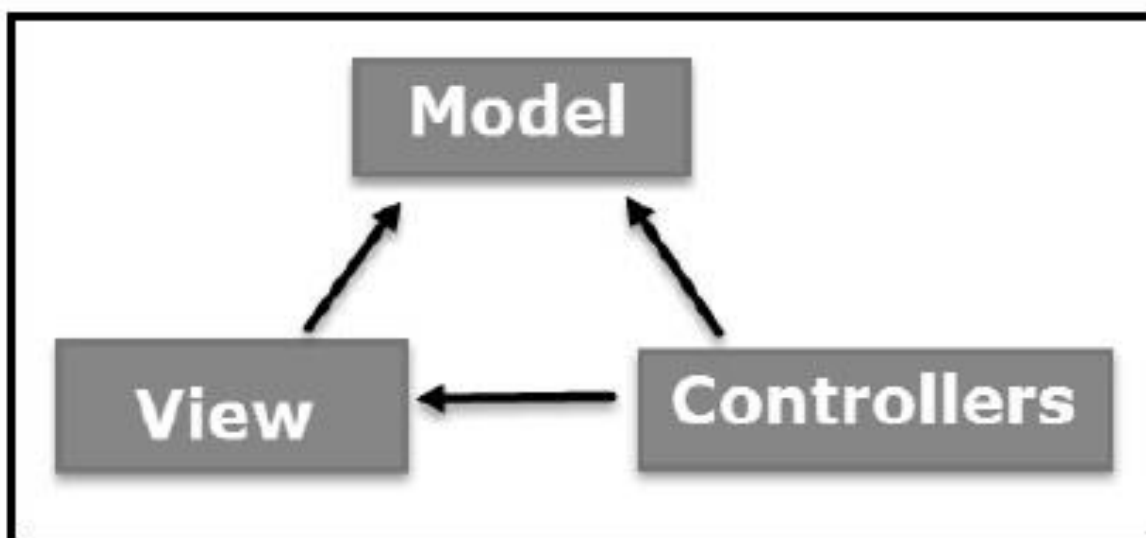


Рисунок 1.3 — Model-View-Controller

ASP.NET підтримує три основні моделі розробки: веб-сторінки, веб-форми та MVC (Model View Controller). Фреймворк ASP.NET MVC — це легкий фреймворк, який можна перевірити, інтегрований із наявними функціями ASP.NET, такими як головні сторінки, автентифікація тощо. У .NET цей фреймворк визначено в збірці System.Web.Mvc. Остання версія MVC Framework — 5.0. Visual Studio використовується для створення програм ASP.NET MVC, які можна додати як шаблон у Visual Studio [12].

Entity framework (далі — EF) — це платформа ORM (об'єктно-реляційне відображення), яку Microsoft надає як частину розробки .NET (версія 3.5 SP1 і пізніші). Її мета полягає в тому, щоб абстрагувати зв'язки з реляційною базою даних таким чином, щоб розробник міг ставитися до сутності бази даних як до набору об'єктів, а потім до класів на додаток до їхніх властивостей. По суті, ми говоримо про відокремлення між нашими програмами та логікою доступу до даних, що є головним плюсом [13].

1.3.2 Огляд технологій для створення бази даних

Бази даних необхідні в багатьох різних випадках зберігання даних. Базу даних зазвичай використовують, коли різні набори даних потрібно зв'язати разом, наприклад:

- учні школи та їхні оцінки;
- записи клієнтів і інформація про продажі;
- історії пацієнтів і лікарів;
- операції між різними банківськими рахунками;
- платники податків і платежі з податку на прибуток.

Трьома основними перевагами баз даних перед іншими, простішими системами зберігання даних (такими як текстові файли та електронні таблиці) є доступ, цілісність і безпека [14].

SQL означає мову структурованих запитів. SQL використовується для зв'язку з базою даних. Згідно з ANSI (Американський національний інститут стандартів), це стандартна мова для систем управління реляційними базами

даних. Інструкції SQL використовуються для виконання таких завдань, як оновлення даних у базі даних або отримання даних із бази даних. Деякі поширені системи керування реляційними базами даних, які використовують SQL: Oracle, Sybase, Microsoft SQL Server, Access, Ingres тощо.

Хоча більшість систем баз даних використовують SQL, більшість із них також мають власні додаткові приватні розширення, які зазвичай використовуються лише в їхній системі. Однак стандартні команди SQL, такі як «Select», «Insert», «Update», «Delete», «Create» та «Drop» можна використовувати для виконання майже всього, що потрібно зробити з базою даних [15].

MySQL — це швидка, проста у використанні система управління реляційними базами даних, яка використовується багатьма малими та великими підприємствами. MySQL розробляється, продається та підтримується шведською компанією MySQL AB. MySQL такий популярний через багато вагомих причин:

- MySQL випускається за ліцензією з відкритим кодом;
- MySQL сама по собі є дуже потужною програмою, вона обробляє велику частину функціональних можливостей найдорожчих і потужних пакетів баз даних;
- MySQL використовує стандартну форму відомої мови даних SQL;
- MySQL працює в багатьох операційних системах і з багатьма мовами, включаючи PHP, PERL, C, C++, JAVA тощо;
- MySQL працює дуже швидко і добре, працює навіть з великими наборами даних;
- MySQL дуже дружній до PHP, найпопулярнішої мови для веб-розробки;
- MySQL підтримує великі бази даних, до 50 мільйонів рядків або більше в таблиці. Обмеження розміру файлу для таблиці за замовчуванням становить 4 ГБ, але можна збільшити його (якщо ваша операційна система може це впоратися) до теоретичного обмеження в 8 мільйонів терабайт (ТБ);

— MySQL можна налаштувати. Ліцензія GPL з відкритим вихідним кодом дозволяє програмістам модифікувати програмне забезпечення MySQL відповідно до власного середовища [16].

SQL Server — це версія Microsoft для системи керування реляційними базами даних. Він має багато функцій і масу досвіду роботи з багатьма різними програмами. Microsoft пропонує різні версії SQL Server, включаючи спеціалізовані версії, такі як Azure SQL Database, яка є хмарною версією SQL Server. Це програмний продукт, основною функцією якого є зберігання та отримання даних за запитом іншого програмного забезпечення, яке може працювати на тому самому комп'ютері або на іншому комп'ютері в мережі (включаючи Інтернет). Корпорація Майкрософт продає принаймні десяток різних версій Microsoft SQL Server, орієнтованих на різні аудиторії та для робочих навантажень.

1.3.3 Огляд технологій для реалізації клієнтської частини

Ключовим моментом у розробці веб-сайту є створення графічного інтерфейсу, який дозволяє користувачам легко взаємодіяти з сайтом. Трьома найважливішими мовами для створення інтерфейсу є HTML, CSS і JavaScript. Усі вони мають специфічне використання, і всі три використовуються для розробки веб-сайтів. Вони допомагають визначити форму, функції та стиль.

HTML — це один із основних будівельних блоків веб-сайту. HTML, або «мова гіпертекстової розмітки», існує в певній формі приблизно з 1993 року, коли її створив фізик Тім Бернерс-Лі. Зараз це вже п'яте покоління, HTML5. HTML — це не мова програмування, а «мова розмітки». Вона використовує синтаксис тегів, щоб, наприклад, змінити спосіб відображення тексту. Вона також може визначити, де на сторінці розміщувати зображення. Чудово те, що HTML відносно проста у вивченні [17].

CSS розшифровується як каскадні таблиці стилів. Ця мова програмування визначає, як HTML-елементи веб-сайту мають відображатися на інтерфейсі сторінки. У той час як HTML є базовою структурою веб-сайту, CSS — це те, що

надає стилю всьому веб-сайту. Яскраві кольори, цікаві шрифти та фонові зображення — все завдяки CSS. Ця мова впливає на весь настрій і тон веб-сторінки, що робить її неймовірно потужним інструментом і важливою навичкою для вивчення веб-розробниками. CSS також дозволяє веб-сайтам адаптуватися до різних розмірів екрана та типів пристроїв [18].

JavaScript є ще однією мовою програмування, яка є однією з основних технологій Всесвітньої павутини, поряд з HTML і CSS. Станом на 2022 рік 98% веб-сайтів використовують JavaScript на стороні клієнта для поведінки веб-сторінок. Тоді як HTML і CSS використовуються для керування презентацією, форматуванням і макетом, JavaScript використовується для керування поведінкою різних веб-елементів [17].

Angular — це фреймворк з відкритим кодом. Це мова розробки TypeScript на основі JavaScript, яка усуває непотрібні функції та код, щоб забезпечити легші та швидші програми. Angular дозволяє створювати цікаві та динамічні односторінкові програми (SPA), використовуючи низку привабливих функцій, таких як:

- шаблонування;
- двостороння прив'язка;
- модульність;
- доступ RESTful API;
- ін'єкція залежності;
- підтримка AJAX.

В Angular можна використовувати HTML як мову шаблонів, а його простий синтаксис можна легко розширити для передачі компонентів програми. Крім того, з Angular не потрібно покладатися на сторонні бібліотеки для створення динамічних програм [19].

Angular більше підходить для великих проектів із жорсткою структурою. У ньому є досить багато готових рішень, більш продумана система збирання та зберігання інформації. Це спрощує конструювання великих веб-сайтів і забезпечує надійне функціонування.

Фреймворк React.js — це фреймворк і бібліотека JavaScript із відкритим кодом, розроблена Facebook. Він використовується для швидкого й ефективного створення інтерактивних інтерфейсів користувача та веб-додатків із значно меншою кількістю коду, ніж JavaScript. Основна роль React полягає в тому, щоб обробляти рівень перегляду цього додатка так само, як у шаблоні «model-view-controller» (MVC), забезпечуючи найкраще та найефективніше виконання візуалізації. Замість того, щоб мати справу з усім інтерфейсом користувача як єдиним блоком, React.js заохочує розділити ці складні інтерфейси користувача на окремі багаторазово використовувані компоненти, які утворюють будівельні блоки цілого інтерфейсу користувача. При цьому фреймворк ReactJS поєднує швидкість і ефективність JavaScript з більш ефективним методом маніпулювання DOM для швидшого відтворення веб-сторінок і створення високодинамічних і адаптивних веб-додатків [20].

Vue.js — це прогресивний фреймворк JavaScript, який використовується для створення UI (інтерфейсів користувача) і SPA (односторінкових програм). Цей фреймворк славиться своїм швидким процесом навчання. Це настільки проста в освоєнні та доступна бібліотека, що зі знанням HTML, CSS і JavaScript можна вже створювати веб-додатки на Vue.js. Еван Ю створив цю структуру. Ідея Евана Ю, який стоїть за цим фреймворком, полягає в тому, щоб створити найкращий фреймворк, об'єднавши найкращі функції вже існуючих Angular і React Frameworks. Він вибрав найкращі частини Angular, як-от синтаксис шаблону, простий у використанні, а також найкращі частини React, як-от двостороннє зв'язування даних, концепцію реквізитів, компонентний підхід, і об'єднав їх, щоб створити новий фреймворк Vue.js.

Vue.js — це проста, швидкозростаюча та адаптована структура для впровадження в розробку додатків, з якою може почати будь-хто з базовими знаннями веб-розробки завдяки її невидимій кривій навчання та легкій для розуміння документації [21].

1.4 Порівняльний аналіз аналогів інформаційно-аналітичних систем з можливістю візуалізації даних

На сьогоднішній день існує велика кількість реалізацій інформаційно-аналітичних систем для адміністрування веб-додатків. Кожна з них індивідуальна та призначена для різних типів веб-сайтів.

Проаналізуємо чотири аналоги інформаційно-аналітичних систем:

- «Sprout Social»;
- «Fathom»;
- «Plausible»;
- «Heap».

Sprout Social пропонує інструменти маркетингу та управління соціальними мережами, програмне забезпечення для обслуговування клієнтів, інструменти аналітики та продукти для збору даних для великих та малих компаній. Sprout Social приділяє особливу увагу чотирьом компонентам ефективності: моніторингу, залученню, виміру та зростанню. Додаткові можливості включають панель приладів, візуальне представлення ваших даних, оновлення в реальному часі. Мінусами даної платформи є обмежена функціональність та відсутність обслуговування клієнтів. Домашня сторінка інформаційно-аналітичної системи Sprout Social зображено на рисунку 1.4.

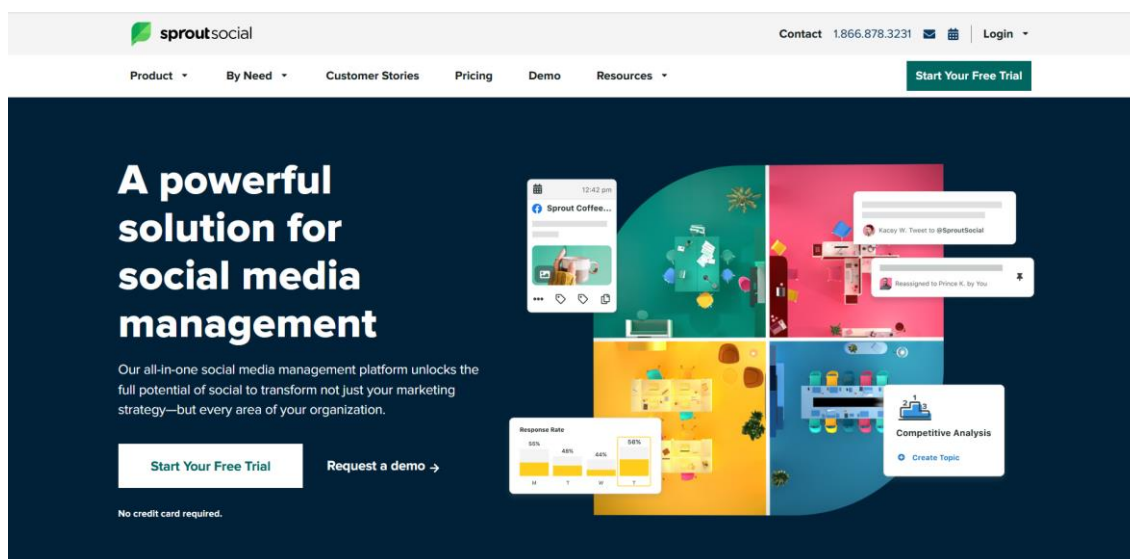


Рисунок 1.4 — Вигляд головної сторінки Sprout Social

Fathom — це незалежна, самофінансована компанія, яка займається тим, що допомагає підприємствам отримувати корисну інформацію про свої веб-сайти. Fathom використовує простий підхід, коли ви не будете отримувати складні аналітичні звіти на веб-сайті, які важко читати. Натомість ви отримаєте ефективні звіти, які легко використовувати та розуміти. Fathom Analytics надає ключові дані про відвідуваність вашого сайту. Таким чином, ви знатимете, на чому зосередитися, щоб поліпшити свій бізнес. Мінусом даної платформи є те, що пропонуються лише платні послуги. Домашня сторінка інформаційно-аналітичної системи Fathom зображено на рисунку 1.5.

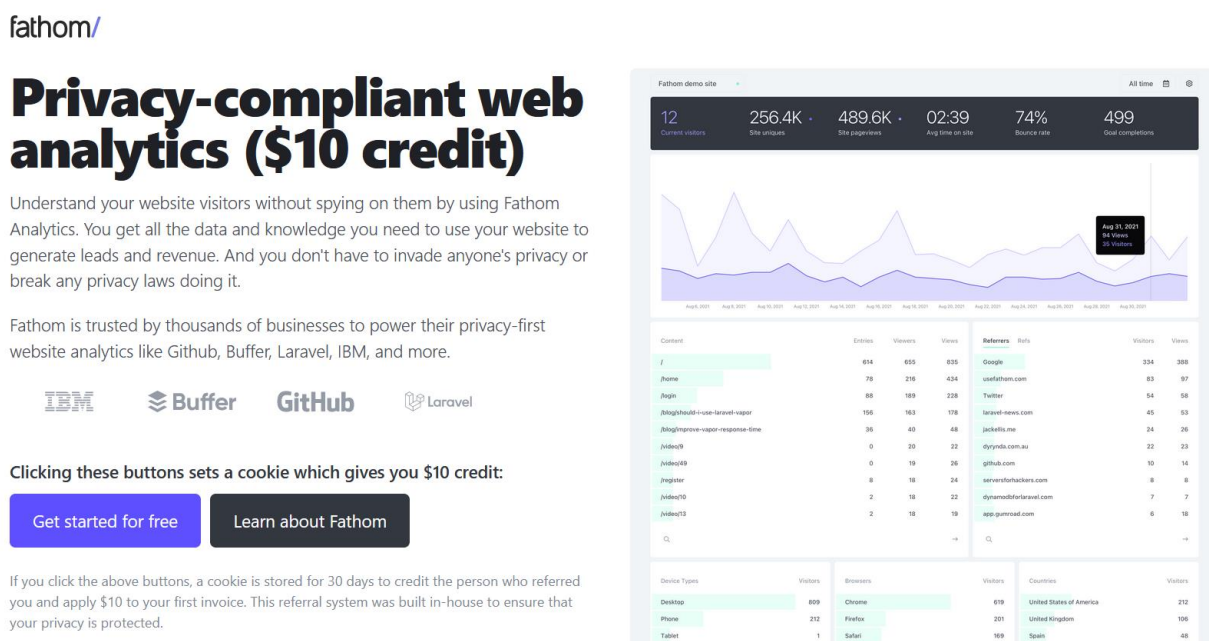


Рисунок 1.5 — Вигляд головної сторінки Fathom

Plausible Analytics є простим у використанні та розумінні без будь-яких попередніх знань. Усі важливі показники відвідуваності сайту та продукту представлені у спрощеному вигляді на одній сторінці. Все зібрано в одному місці, без рівнів навігації і необхідності створення звітів користувача. Мета Plausible — допомогти веб-сайтам покращити свою продуктивність, надаючи їм усю необхідну аналітику сайту без уповільнення часу завантаження сайту. Мінус використання Plausible Analytics є те, що він оновлюється кожні 24 години, тому

ідентифікує користувачів, що повертаються, як нових. Домашня сторінка інформаційно-аналітичної системи Plausible Analytics зображено на рисунку 1.6.

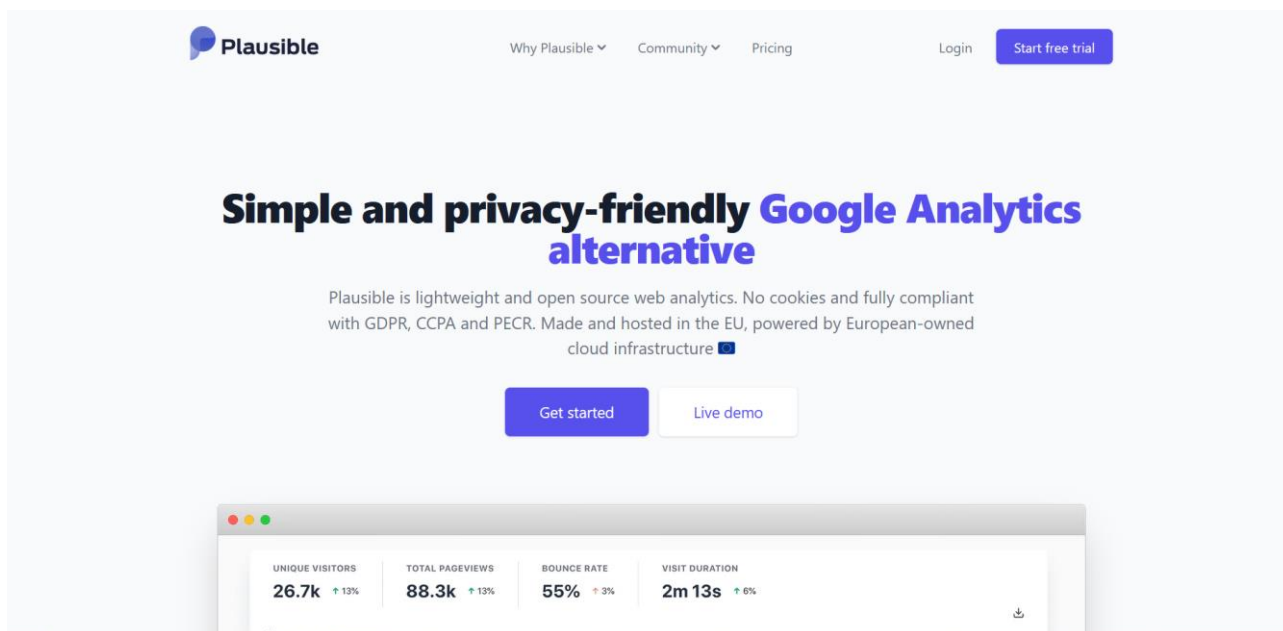


Рисунок 1.6 — Вигляд головної сторінки Plausible Analytics

Неар створений для того, щоб допомогти маркетологам зрозуміти подорож користувача через кожен крок воронки, щоб вони могли оптимізувати конверсії та отримати глибоке розуміння кожної окремої взаємодії між відвідувачами сайту та брендом. Неар має низку особливостей, які відрізняють його від інших рішень. Він включає автоматичне захоплення кожної взаємодії користувача; автоматичне відстеження подій; визначення активних користувачів та груп користувачів; простий та інтуїтивно зрозумілий інтерфейс користувача для маркування веб-сторінок та мобільних додатків; не вимагає кодування. Мінусом використання даної платформи є те, що потрібно навчання для користувачів з налаштування та формування звітів. Також потрібне базове знання тегів HTML щоб користуватися платформою. Домашня сторінка інформаційно-аналітичної системи Неар зображено на рисунку 1.7.

Порівняльний аналіз інформаційно-аналітичних систем адміністрування веб-додатків Sprout Social, Fathom, Plausible та Неар з системою, що розроблюється зображено в таблиці 1.1.

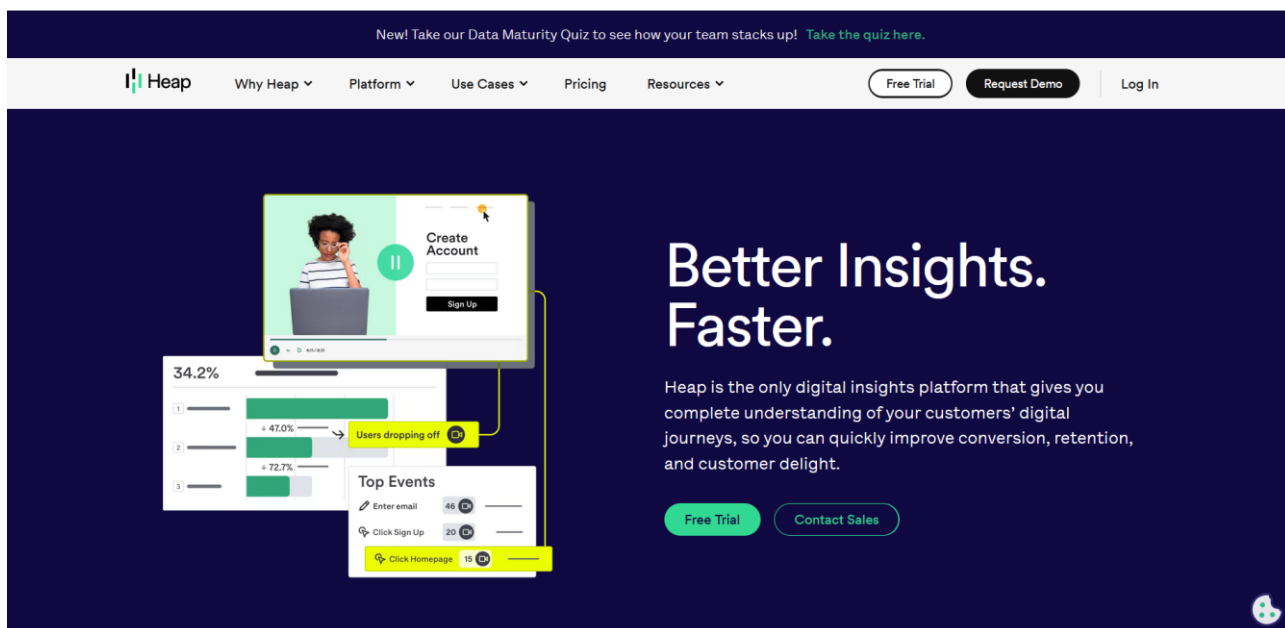


Рисунок 1.7 — Вигляд головної сторінки Heaps

Таблиця 1.1 — Порівняльний аналіз інформаційно-аналітичних систем

Характеристика	Sprout Social	Fathom	Plausible	Heaps	Система, що розроблюється
Присутність реклами	+	+	+	+	-
Необхідність оплати	+	+	+	+	-
Інтуїтивно зрозумілий інтерфейс	-	-	+	+	+
Необхідність знань в програмуванні	-	-	-	+	-
Необмежена функціональність	-	+	+	+	-
Побудова графіків	-	+	+	+	+
Оновлення в реальному часі	+	+	-	+	+
Швидкий аналіз даних	-	-	-	-	+
Адміністративна панель	-	-	-	-	+

Отже, аналіз показав що перевагами інформаційно-аналітичної системи, що розробляється є: безкоштовне користування, відсутність реклами, швидкий аналіз даних, можливість адміністрування сайту.

2 АНАЛІЗ МЕТОДІВ РОЗРОБКИ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ

2.1 Аналіз методи обробки отриманих даних з веб-додатку та розрахунок статистики

Перш ніж приступити до аналітики та статистики, потрібно зрозуміти, які дані необхідно збирати. Оскільки веб-додаток є комерційним проектом, то потрібно розуміти, з чого складається його користь для бізнесу.

Успіх веб-додатку визначає:

- великий потік якісних відвідувачів — чим він більший, чим більша ймовірність заробити;
- середній чек — скільки і на яку суму купують;
- повторні покупки — чи можливі, з якою періодичністю, на яку суму і т.п.

Необхідно провести процес обробки даних, склеювання та їх інтерпретації на основі чого робляться висновки і приймається рішення [26]. Структурну схему циклу завдань в веб-аналітиці подано на рисунку 2.1.

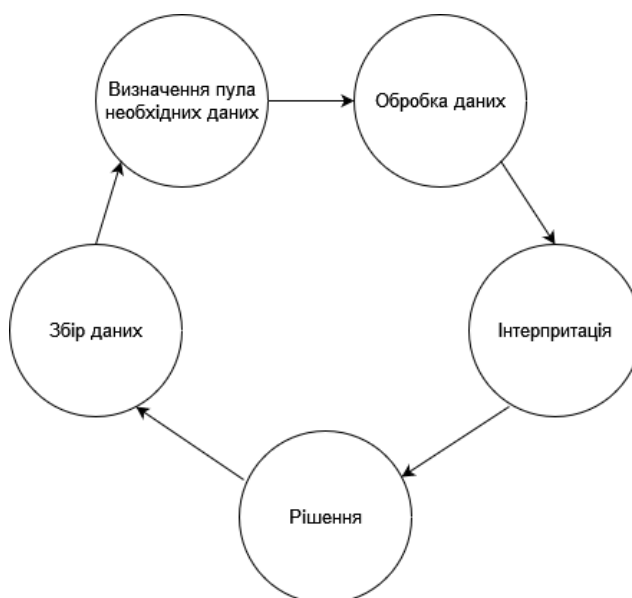


Рисунок 2.1 — Структурна схема циклу завдань в веб-аналітиці

В рамках роботи з інтернет-статистикою потрібно діяти на основі методів. До таких методів відносяться:

- метод сегментації клієнтів (RFM-аналіз);
- метод сегментації товарів (ABC-аналіз);
- метод для побудови прогнозів майбутніх продаж (лінійна регресія);

Розглянемо детальніше кожен метод і можливості застосування в магістерській дипломній роботі.

2.1.1 Метод сегментації клієнтів

Сегментація клієнтів — це процес поділу клієнтської бази на окремі частини (сегменти), що будуються завдяки окремим ознакам. Даний поділ дозволить зрозуміти, з якою аудиторією працюєте та які її потреби. Ціль сегментування клієнтів — це концентрація маркетингових зусиль на певні групи покупців, щоб більш ефективно взаємодіяти з ними.

Завдання сегментування клієнтів:

- визначення основних сегментів користувачів;
- виявлення бізнес-пріоритетів у роботі;
- побудова певних стратегій під час роботи з базою покупців.

Сегментація клієнтів популярна, оскільки вона допомагає визначити нові продукти та послуги для створення наступних, а також більш ефективно рекламувати та продавати існуючі пропозиції [27].

Сегментація діючих клієнтів відрізняється від сегментації аудиторії, якій компанія збирається вперше запропонувати продукт.

При сегментації потенційних покупців виділяються групи і з ними йде робота: будуються гіпотези, прогнози, які сегменти принесуть більше заявок і продажів.

При сегментації бази клієнтів є більше точних даних. Наприклад, вже відомо, на які продукти користувачі витрачають гроші, що їм цікаво, чи готові вони до платної доставки і чи їм сервіс.

Аналіз RFM — це маркетинговий метод, який використовується для кількісного ранжування та групування клієнтів на основі давності останньої покупки (R - Recency), сумарної частоти покупок (F - Frequency) та загальної

грошової суми, витраченої на покупки (M - Monetary), щоб визначити найкращих клієнтів і провести цілеспрямовані маркетингові кампанії. Система призначає кожному клієнту числові бали щоб на основі цих факторів забезпечити об'єктивний аналіз [28].

Класифікуємо клієнтів за балами від 1 до 5, використовуючи значення RFM. Розподілимо оцінки за такими критеріями:

- бал 1 — давність покупки більше 3-х місяців, сумарна частота покупок менше 3, сума, витрачена на покупки менше 500 грн;
- бал 2 — давність покупки від 1 до 3 місяців, сумарна частота покупок від 3 до 6, сума, витрачена на покупки від 500 до 1000 грн;
- бал 3 — давність покупки від 3-х тижнів до 1 місяця, сумарна частота покупок від 6 до 12, сума, витрачена на покупки від 1000 до 2000 грн;
- бал 4 — давність покупки від 1 до 3 тижнів, сумарна частота покупок від 12 до 17, сума, витрачена на покупки від 2000 до 5000 грн;
- бал 5 — давність покупки менше тижня, сумарна частота покупок більше 17, сума, витрачена на покупки 5000 грн.

Отримавши бали, за кожним критерієм розраховується середній бал.

Використовуючи таку систему оцінювання RFM створюються сегменти клієнта.

Лояльні клієнти — це клієнти, які отримують найкращі бали в кожній категорії (середній бал бал від 4 до 5). Вони лояльні, готові витратити гроші та, швидше за все, незабаром зроблять ще одну покупку. Такі клієнти готові реагувати на програми лояльності. Вони, швидше за все, зацікавляться новими продуктами. І оскільки вони віддані бренду та його продуктам, мабуть, має менший бізнес-сенс пропонувати їм знижки.

Клієнти групи ризику — клієнти, які були у вищому рівні в минулому (найкращі, великі витрати та/або лояльні), але які зараз мають низькі оцінки за останнім часом і частотою (середній бал від 1 до 3). Необхідно розглянути можливість націлювання на них повідомлень, спрямованих на утримання, таких як знижки, ексклюзивні пропозиції та запуск нових продуктів.

2.1.2 Метод сегментації товарів

Методи сегментації не втрачають своєї актуальності і прекрасно виконують завдання веб-аналітики для бізнесу. Прикладом таких методів може бути АВС-аналіз.

У основі АВС-аналізу лежить принцип Парето: 20% зусиль забезпечують 80% результату. Що стосується товарного асортименту його можна сформулювати так: 20% товарів роблять 80% обороту підприємства.

Суть методу у тому, щоб відранжувати позиції з погляду вкладу у загальну прибутковість. Всі товари в результаті будуть поділені на три групи:

- група А — найцінніші позиції, ці товари становлять 20% від загального асортименту та забезпечують 80% продажів;
- група В — проміжні позиції, складають 30% асортименту та роблять 15% продажів;
- група С — найменш цінні, ці товари становлять 50-60% асортименту та дають 5% продажів [29].

АВС-аналіз можна проводити для оцінювання:

- окремих позицій товарів чи всього асортименту - щоб зрозуміти, які товари є найбільш рентабельними;
- складських запасів — чого треба позбавлятися насамперед;
- постачальників — з ким вигідніше працювати;
- клієнтської бази - яка група клієнтів приносить найбільше прибутку;
- ефективності роботи підрозділів - від якого відділу велика віддача;
- витрат, інвестицій, бюджету загалом — які вкладення краще відбиваються.

Для того, щоб зрозуміти, які групи товарів продавати вигідно, а які ні обираються групи товарів та визначаються суми продажів за певний період.

Наступним кроком обчислюється відсоток від загального обсягу продажів по кожній групі і визначається належність товару до відповідної групи за такими критеріями:

- А — доля в обороті більше 50%;
- В — доля в обороті від 15% до 49%;
- С — доля в обороті від 0% до 14%.

За результатами обчислень позначається належність товару до відповідної групи. Відповідно, товари групи А продавати вигідно і на їх реалізації варто сконцентруватися, товари групи В — це товари, асортимент яких варто розширити та підтримувати, а товари з групи С приносять мінімум прибутку — їх кількість можна сміливо зменшувати або виводити з асортименту.

2.1.3 Метод для побудови прогнозів майбутніх продаж

Інтелектуальний аналіз даних включає використання вдосконалених інструментів аналізу даних для пошуку раніше невідомих дійсних закономірностей і зв'язків у величезних наборах даних. Ці інструменти можуть включати статистичні моделі, методи машинного навчання та математичні алгоритми, такі як нейронні мережі або дерева рішень. Таким чином, інтелектуальний аналіз даних включає аналіз і прогнозування.

Залежно від різних методів і технологій на перетині машинного навчання, керування базами даних і статистики, професіонали в галузі інтелектуального аналізу даних намагаються обробляти величезну кількість даних і робити висновки з них [30].

В останніх проектах інтелектуального аналізу даних були розроблені та використані різні основні методи інтелектуального аналізу даних, включаючи асоціацію, класифікацію, кластеризацію, прогнозування, послідовні шаблони та регресію.

Розглянемо методи інтелектуального аналізу даних.

Першим методом є класифікація. Цей метод використовується для отримання важливої та актуальної інформації про дані та метадані. Цей метод аналізу даних допомагає класифікувати дані в різних класах.

Методи інтелектуального аналізу даних можна класифікувати за різними критеріями.

Класифікація фреймворків інтелектуального аналізу даних за типом видобутих джерел даних. Ця класифікація залежить від типу даних, що обробляються. Наприклад, мультимедіа, просторові дані, текстові дані, дані часових рядів, Всесвітня павутина тощо.

Класифікація фреймворків інтелектуального аналізу даних відповідно до бази даних. Ця класифікація базується на задіяній моделі даних. Наприклад, об'єктно-орієнтована база даних, транзакційна база даних, реляційна база даних тощо.

Класифікація фреймворків інтелектуального аналізу даних відповідно до виду виявлених знань. Ця класифікація залежить від типів виявлених знань або функцій аналізу даних. Наприклад, дискримінація, класифікація, кластеризація, характеристика тощо. Деякі фреймворки, як правило, є розширеними фреймворками, які пропонують разом декілька функціональних можливостей інтелектуального аналізу даних.

Класифікація фреймворків інтелектуального аналізу даних відповідно до використовуваних методів інтелектуального аналізу даних. Ця класифікація відповідає використаному підходу до аналізу даних, наприклад нейронним мережам, машинному навчанню, генетичним алгоритмам, візуалізації, статистиці, орієнтованості на сховища даних або бази даних тощо.

Класифікація також може брати до уваги рівень взаємодії користувача, який бере участь у процедурі інтелектуального аналізу даних, наприклад системи, керовані запитом, автономні системи або інтерактивні пошукові системи.

Другим методом є кластеризація. Кластеризація — це метод поділу інформації на групи пов'язаних об'єктів. Опис даних кількома кластерами в основному втрачає певні обмежені деталі, але забезпечує покращення. Він моделює дані за своїми кластерами. З історичної точки зору кластеризація ґрунтується на статистиці, математиці та числовому аналізі. З точки зору машинного навчання, кластери пов'язані з прихованими шаблонами, пошук кластерів є неконтрольованим навчанням, а подальша структура представляє

концепцію даних. З практичної точки зору, кластеризація відіграє надзвичайну роль у програмах аналізу даних. Наприклад, дослідження наукових даних, аналіз тексту, пошук інформації, програми просторової бази даних, CRM, веб-аналіз, обчислювальна біологія, медична діагностика та багато іншого.

Кластерний аналіз — це техніка інтелектуального аналізу даних для ідентифікації подібних даних. Ця техніка допомагає розпізнавати відмінності та подібності між даними. Кластеризація дуже схожа на класифікацію, але передбачає групування фрагментів даних разом на основі їх подібності.

Третім методом є правила асоціації. Цей метод інтелектуального аналізу даних допомагає виявити зв'язок між двома чи більше елементами.

Правила асоціації — це оператори «якщо-тоді», які допомагають показати ймовірність взаємодії між елементами даних у великих наборах даних у різних типах баз даних. Інтелектуальний аналіз правил асоціації має кілька застосувань і зазвичай використовується для кореляції продажів у наборах даних або медичних даних.

Четвертим методом є виявлення аномалій. Цей тип методу інтелектуального аналізу даних пов'язаний зі спостереженням за елементами даних у наборі даних, які не відповідають очікуваній моделі чи очікуваній поведінці. Цю техніку можна використовувати в різних сферах, таких як вторгнення, виявлення шахрайства тощо. Викид — це точка даних, яка надто сильно відрізняється від решти набору даних. Більшість наборів реальних даних мають викид. Виявлення викидів відіграє важливу роль у сфері інтелектуального аналізу даних. Виявлення викидів є цінним у багатьох сферах, наприклад як ідентифікація перебоїв у мережі, виявлення шахрайства з кредитними чи дебетовими картками тощо.

П'ятим методом є послідовні шаблони. Послідовний шаблон — це метод аналізу даних, яка спеціалізується на оцінюванні послідовних даних для виявлення послідовних шаблонів. Він включає пошук підпослідовностей у наборі послідовностей, де послідовність можна виміряти за різними критеріями, такими як довжина, частота появи тощо.

Іншими словами, ця техніка інтелектуального аналізу даних допомагає виявити або розпізнати подібні закономірності в даних транзакцій протягом деякого часу.

Шостим методом є регресія. Метод регресійного аналізу — це процес інтелектуального аналізу даних, який використовується для визначення та аналізу зв'язку між змінними через наявність іншого фактора. Він використовується для визначення ймовірності конкретної змінної. Регресія, перш за все, форма планування та моделювання. Наприклад, ми можемо використовувати його для прогнозування певних витрат залежно від інших факторів, таких як доступність, споживчий попит і конкуренція. У першу чергу він дає точне співвідношення між двома чи більше змінними в даному наборі даних.

Сьомим методом є прогноз. Прогноз використовує комбінацію інших методів інтелектуального аналізу даних, таких як тенденції, кластеризація, класифікація тощо. Він аналізує минулі події або випадки в правильній послідовності, щоб передбачити майбутню подію.

Метод лінійної регресії є одним з методів прогнозування. Даний метод один із найпростіших, і найчастіше використовується для прогнозування більш-менш стабільних та регулярних продажів.

Переваги лінійної регресії:

- швидкість та простота отримання моделі;
- інтерпретованість моделі — лінійна модель є прозорою та зрозумілою для аналітика, за отриманими коефіцієнтами регресії можна будувати висновки про те, як той чи інший чинник впливає на результат, зробити на цій основі корисні висновки;
- широке застосування — багато реальних процесів економіки та бізнесу можна з достатньою точністю описати лінійними моделями;
- вивченість цього підходу — для лінійної регресії відомі типові проблеми та їх вирішення, розроблено та реалізовано тести оцінювання статичної значущості одержуваних моделей.

Метод базується на аналізі взаємозв'язку двох змінних (метод парної кореляції) - вплив варіації факторного показника X на результативний показник Y :

$$y_n = a + bx, \quad (2.1)$$

де y_n — результативний показник;

a і b — коефіцієнти лінійної регресії;

x — факторний показник.

Для визначення параметрів a та b на основі вимог методу найменших квадратів за допомогою диференціальних обчислень складається система нормальних рівнянь:

$$\begin{cases} \sum y = na + b * \sum x; \\ \sum xy = a * \sum x + b * \sum x^2; \end{cases} \quad (2.2)$$

де n — кількість факторів регресії;

$\sum y$ — сума результативних показників;

$\sum x$ — сума факторних показників.

Для вирішення цієї системи рівнянь застосовується спосіб визначників, що дозволяє зводити до мінімуму неточності округлень у розрахунках параметрів рівнянь регресії:

$$a = \frac{\sum y * \sum x^2 - \sum xy * \sum x}{n * \sum x^2 - \sum x^2}. \quad (2.3)$$

$$b = \frac{n * \sum xy - \sum x * \sum y}{n * \sum x^2 - \sum x^2}. \quad (2.4)$$

Для знаходження a можна скористатися спрощеною формулою:

$$a = y - bx, \quad (2.5)$$

$$\text{де } x = \frac{\sum x}{n};$$

$$y = \frac{\sum y}{n}.$$

Обраховуючи дані по формулі отримуємо очікуваний обсяг продажів у майбутньому періоді.

2.2 Засоби для візуального представлення даних

Сьогодні, особливо в діловому світі, люди одержимі даними, але не без вагомої причини. Дані дозволяють досліджувати та розуміти приховані тенденції. Через це підприємства, уряди, дослідники та навіть окремі особи щодня використовують величезні обсяги даних. Маючи так багато даних, необхідно мати можливість ділитися ними легко та інтуїтивно.

Існують потужні інструменти візуалізації даних, які допоможуть краще їх зрозуміти. Оскільки більшість даних передаються через Інтернет, багато популярних інтерфейсних фреймворків, як Angular, мають потужні бібліотеки візуалізації даних. HighChart Angular Wrapper — один із найкращих інструментів візуалізації даних (рисунок 2.2).

HighChart Angular Wrapper — це компонент на базі Angular з відкритим вихідним кодом, який забезпечує елегантну та багатофункціональну візуалізацію Highcharts у додатку Angular і може легко використовуватися разом із компонентами Angular. Highcharts — це чиста бібліотека діаграм на основі JavaScript, призначена для вдосконалення веб-додатків шляхом додавання можливостей інтерактивних діаграм. Highcharts пропонує широкий вибір діаграм. Наприклад, лінійні діаграми, сплайн-діаграми, площинні діаграми, стовпчасті діаграми, секторні діаграми тощо [31].

Особливості Highcharts:

- сумісність – підтримується всіма сучасними браузерами, а також браузерами iPhone/iPad і Internet Explorer 6 і вище;
- чистий TypeScript — JavaScript не потрібен, оскільки повний API Highcharts доступний у TypeScript;
- відсутній Flash — немає потреби в плагінах на стороні клієнта, таких як Flash player або Java, оскільки Highcharts використовує власні технології браузера, а діаграми можуть працювати без змін на сучасних мобільних пристроях;

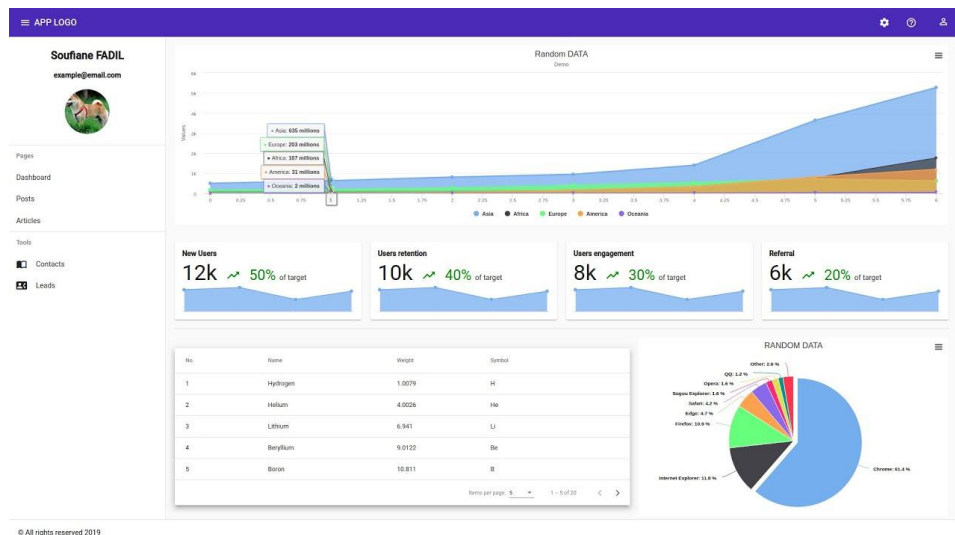


Рисунок 2.2 — Вигляд інформаційної панелі Angular Highchart

— чистий синтаксис — більшістю параметрами конфігурації діаграми можна керувати за допомогою такого синтаксису JSON;

— динамічний — точки можна додавати динамічно навіть після створення діаграми, підтримуються хуки подій, підтримується взаємодія з сервером;

— задокументований — API Highcharts ретельно задокументовано з численними прикладами коду та синтаксису.

Інший засіб для візуалізації — amCharts — це вдосконалена графічна бібліотека, яка задовольняє будь-які потреби у візуалізації даних (рисунок 2.3).

Основні можливості amCharts:

— багато різноманітних типів графіків, включаючи інтерактивні карти та діаграми Ганта;

— можливість аналізу даних «вглиб» та інші добре опрацьовані інтерактивні опції;

— документація, що описує всі потрібні методи, добре написана, щоправда, на мій смак, не надто зручна у використанні;

— інтеграція з React, Angular, Vue, Ember та ін;

— експорт графіків у зображення або PDF-файл;

— повна техпідтримка для всіх користувачів та пріоритетна для клієнтів з ліцензією.

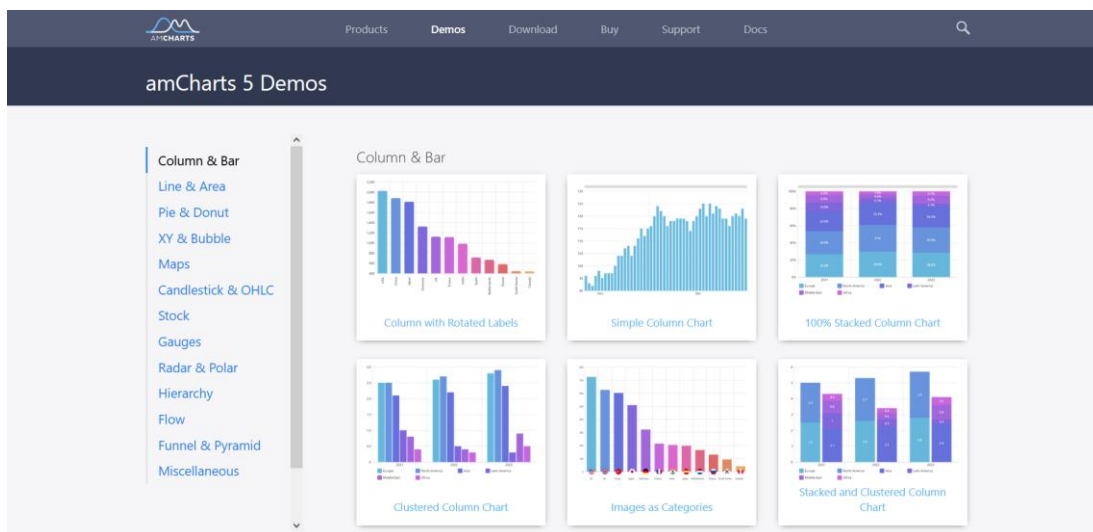


Рисунок 2.3 — Вигляд сторінки з графіками в amCharts

Ще одним засобом для візуалізації даних є AnyChart — добре опрацьована, багатофункціональна бібліотека для візуалізації даних з рендерингом в SVG/VML. Дозволяє створювати різноманітні графіки та діаграми для подальшого прийняття рішень на їх основі (рисунок 2.4).

Основні можливості:

- більше 80 типів графіків, включаючи звичайні графіки, біржові графіки, карти, а також діаграми Ганта та мережеві графіки PERT;
- багато варіантів, як завантажувати дані: XML, JSON, CSV, JS API, Google Sheets, HTML-таблиці;
- технічні індикатори для технічного аналізу та анотації;
- вичерпна документація та опис API;
- інтеграція з Angular, Oracle APEX, React, Elasticsearch, Vue.js, Android, iOS та ін.;
- підтримка старих версій браузерів.



Рисунок 2.4 — Вигляд сторінки з графіками в Anychart

3 ПРОЕКТУВАННЯ ТА ТЕСТУВАННЯ РОБОТИ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ АДМІНІСТРУВАННЯ ВЕБ-ДОДАТКУ

3.1 Вибір технологій для реалізації інформаційно-аналітичної системи

Для реалізації поставленої задачі було вирішено використовувати триланкову архітектуру, яка складається з таких компонентів: клієнт, сервер і база даних. Структурна схема даної архітектури зображена на рисунку 3.1.

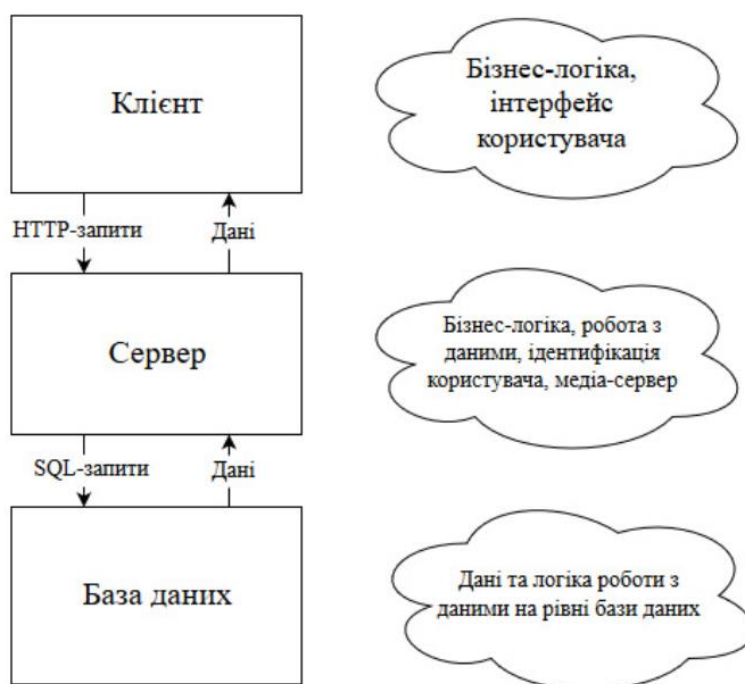


Рисунок 3.1 — Структурна схема триланкової архітектур реалізації веб-додатку

Інформаційно-аналітична система для адміністрування веб-додатків побудована за принципами триланкової архітектури побудови веб-додатків. Кожен рівень цієї архітектури реалізовано з використанням різних технологій і головною ціллю було створення мультиплатформного рішення з використанням відкритих технологій.

На основі аналізу, проведеного в 1 розділі, для реалізації клієнтського рівня використано такий набір технологій:

Фреймворк для побудови веб-застосунків Angular. Angular найбільше підходить для розробки веб-додаткі. Він повністю розширюваний і добре працює

з іншими бібліотеками. Кожну функцію можна змінити або замінити відповідно до унікального робочого процесу розробки та потреб функцій;

Bootstrap має вільний набір інструментів для створення сайтів та веб-додатків. Включає HTML і CSS шаблони оформлення для типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсу, включаючи JavaScript розширення. Bootstrap використовує сучасні напрацювання в області CSS та HTML, тому необхідно бути уважним за підтримки старих браузерів.

Для серверного рівня використано мову програмування C# так як її багата бібліотека класів полегшує реалізацію багатьох функцій та є крос-платформною. Для рівня бази даних обрано MSSQL. Сервер MS SQL вважається одним із найбезпечніших серверів баз даних зі складними алгоритмами шифрування, що робить практично неможливим зламати рівні безпеки, встановлені користувачем. Сервер MS SQL не є сервером бази даних з відкритим вихідним кодом, що знижує ризик атак на сервер бази даних.

Для візуалізації даних було застосовано бібліотеку Angular Highchart так як вона більш гнучка, функціональна і не потребує плати коштів за використання.

3.2 Розробка структури веб-додатку для отримання даних

В цілях отримання даних для аналізу в інформаційно аналітичній системі було розроблено веб-сайт. Даний сайт є інтернет-магазином кондитерської продукції. Інтернет магазин призначений для продажу кондитерських виробів. Споживачі можуть оформити замовлення на сайті, додати товар до списку бажань або відслідковувати наявність товару в магазині. Також є можливість зареєструватись на сайті або авторизуватись, щоб переглядати зроблені замовлення чи список бажань.

Розроблено загальну структуру інтернет-магазину, показану на рисунку 3.2.

Цей сайт містить всі сторінки, необхідні для представлення інформації про кондитерські вироби, вибір продукції, замовлення та зворотний зв'язок.

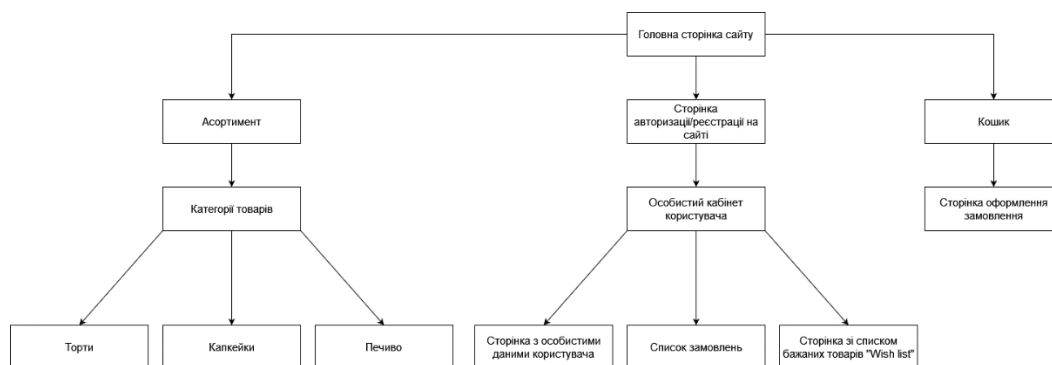


Рисунок 3.2 — Розроблена структурна схема інтернет-магазину

Головна сторінка інтернет-магазину складається з таких розділів (рисунок 3.3):

- розділ «Про нас» містить загальну інформацію про кондитерську та її продукцію;
- розділ «Коментарі» містить відгуки клієнтів;
- розділ «Форма зворотного зв'язку» містить форму, в якій клієнти можуть залишати відгуки та запитання;
- розділ «Контакти» знаходиться в футері головної сторінки.

Шапка головної сторінки містить логотип, навігаційне меню з гіперпосиланнями на головну сторінку, каталог продукції, кошик та авторизацію або реєстрацію в веб-додатку.

Нижній колонтитул головної сторінки містить логотип, гіперпосилання на розділи, контактну інформацію та посилання на соціальні мережі.

Сторінка «Асортимент» складається з трьох категорій: «Торти», «Кекси» та «Печиво». Кожна з цих категорій веде до відповідного каталогу кондитерських виробів.

Кнопка «Кошик» веде на сторінку переліку обраного товару до покупки. Можливість перейти на сторінку оформлення замовлення можливе лише для авторизованих користувачів.

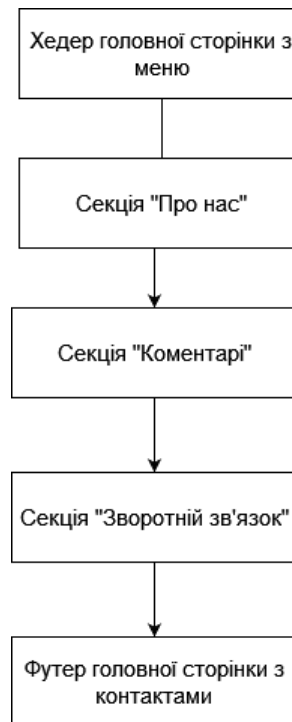


Рисунок 3.3 — Структурна схема головної сторінки інтернет-магазину

Перед користувачем відкриється сторінка з полями для введення особистої інформації та інформації про доставку.

Кнопка «Авторизація» веде на сторінку авторизації чи реєстрації користувача. Якщо користувач зареєстрований він потрапить в особистий кабінет. Там для нього буде можливість переглянути особисті дані, змінити їх, переглянути список минулих замовлень, переглянути «Wishlist».

Сторінка «Wishlist» призначена для збереження товарів, які користувач в майбутньому хотів би придбати.

Візуальне представлення сторінок наведено у Додатку В.

Розглянемо діаграму активності клієнта, показану на рисунку 3.4.

Домашня сторінка — це перше, що бачать клієнти, коли заходять на ваш веб-сайт. Звідти вони можуть переглянути інформацію про кондитерські вироби, прочитати відгуки покупців або залишити свій відгук.

Також клієнт може перейти з головної сторінки на сторінку категорії товарів та вибрати категорію. Перед користувачем відкриється сторінка з переліком товарів.

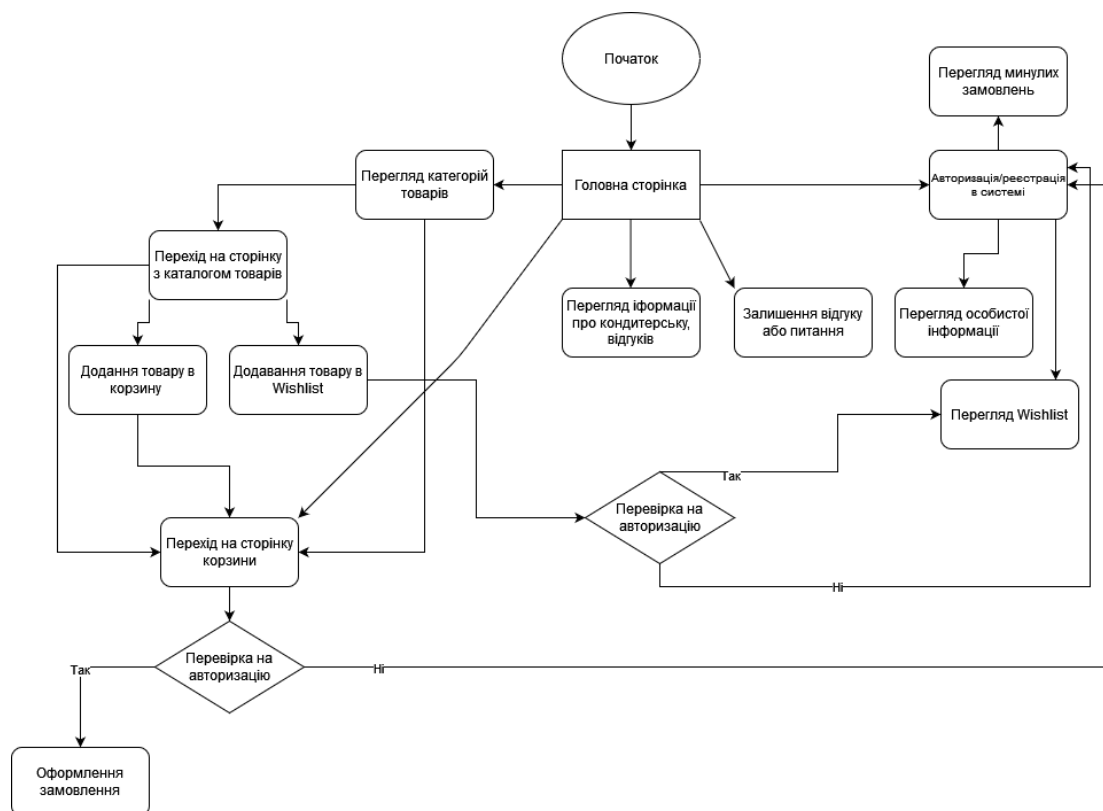


Рисунок 3.4 — Діаграма активностей користувача в веб-додатку

Клієнт може переглянути інформацію про товар, додати товар в корзину або додати в Wishlist (якщо він авторизований в системі). Потім, обравши необхідні позиції, клієнт може перейти в кошик, переглянути вибрані товари та перейти на сторінку замовлення. Сторінка оформлення замовлення також доступна лише для авторизованих користувачів.

Клієнти мають можливість зареєструватись на сайті або авторизуватись, якщо в них вже є існуючий профіль.

3.3 Розробка структури інформаційно-аналітичної системи адміністрування веб-додатку

Розглянемо структуру інформаційно-аналітичної системи, яка буде обробляти дані, отримані з інтернет-магазину (рисунок 3.5).

Інформаційна панель (сторінка «Dashboard») містить основну інформацію по веб-додатку:

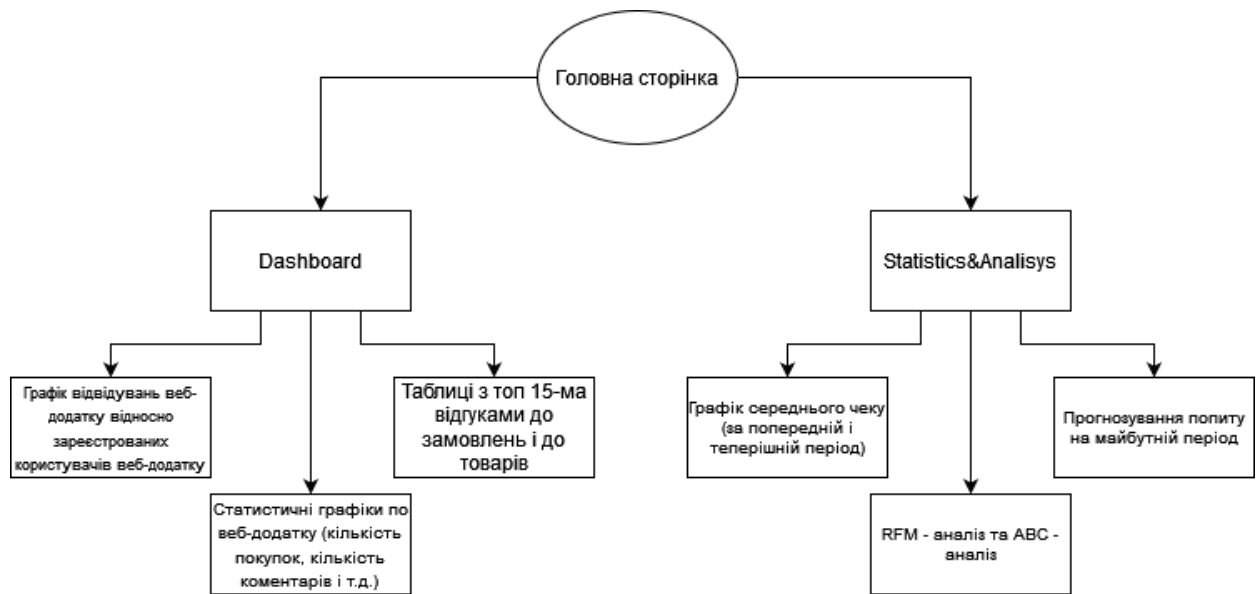


Рисунок 3.5 — Розроблена структурна схема інформаційно-аналітичної системи адміністрування веб-додатку

— графік відвідувачів та зареєстрованих за однаковий період — можна відслідковувати кількість користувачів, які просто відвідали додаток відносно кількості користувачів, які відвідали і зареєструвались;

— графіки активності в веб-додатку — статистика за переглядом категорій, статистика за переглядом товарів, статистика за кількістю замовлень, сума чеку за період, статистика за кількістю залишених коментарів, статистика за кількістю користувачів, які чекають на наявність товару в магазині, статистика за кількістю користувачів, які додали товар в список бажань;

— таблиця з коментарями, які користувачі залишили до свого замовлення;

— таблиця з коментарями, які користувачі залишили до товару.

Наступна сторінка — Statistics&Analysis — відображає для адміністратора аналіз веб-додатку згідно методів, описаних в другому розділі.

На цій сторінці відображено:

— графіки середнього чеку покупців за теперішній та минулий період — дані графіки показують як змінились продажі покупців порівняно з минулим періодом, що дає адміністратору інформацію про спад чи підвищення продажів;

— таблицю та діаграму, сформовану на основі RFM-аналізу — на основі даної таблиці можна визначитись з споживачами, які потрапляють у групу ризику і робити спеціальні пропозиції;

— таблицю та діаграму, сформовану на основі ABC — аналізу — на основі даної таблиці можна отримати дані щодо товарів, які треба вивести з продажів, оскільки вони не приносять прибутку;

— графік, який прогнозує продажі на майбутній період.

3.4 Реалізація інформаційно-аналітичної системи адміністрування веб-додатку

3.4.1 Реалізація клієнтської частини

Першим кроком у створенні інформаційно-аналітичної системи є налаштування середовища для створення, редагування та налаштування веб-додатків. Вибір припав на редактор вихідного коду Visual Studio Code, оскільки він поширюється безкоштовно і має версії для платформ Windows, Linux і OS X. Після інсталяції редактора необхідно створити папку з файлами проекту (рисунок 3.6).

Для створення дизайнів і стилів необхідно використовувати каскадні таблиці стилів CSS. Найпопулярнішим фреймворком CSS для розробки адаптивних веб-сайтів, орієнтованих на мобільні пристрої, є Bootstrap.

Bootstrap — це безкоштовна інтерфейсна платформа розробки з відкритим кодом для створення веб-сайтів і веб-додатків. Розроблений для адаптивної розробки мобільних веб-сайтів, Bootstrap надає набір синтаксису для дизайну шаблонів.

Bootstrap робить адаптивний веб-дизайн. Це дає змогу веб-сторінці чи додатку визначати розмір і орієнтацію екрана відвідувача та автоматично адаптувати відображення відповідно до цього. Підхід, орієнтований на мобільні пристрої, передбачає, що смартфони, планшети та мобільні програми для конкретних завдань є основними інструментами для виконання роботи.

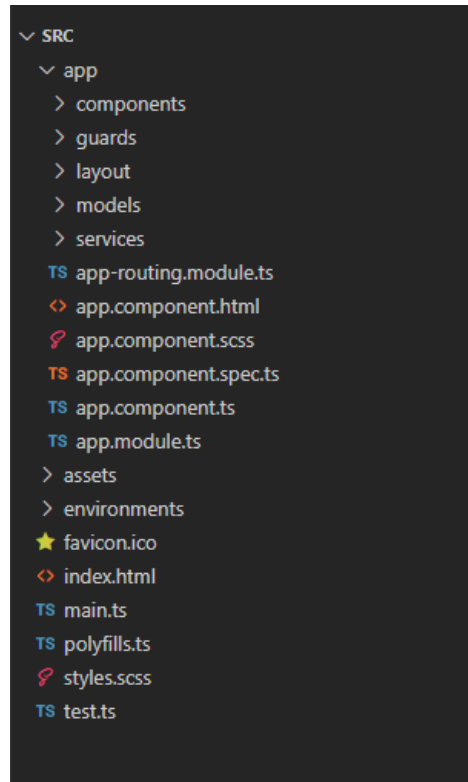


Рисунок 3.6 — Файлова структура проекту

Bootstrap відповідає вимогам цих технологій у дизайні та включає компоненти інтерфейсу користувача, макети, інструменти JavaScript і структуру для реалізації. Програмне забезпечення доступне попередньо скомпільоване або у вигляді вихідного коду [32].

Повний лістинг коду розмітки наведено в додатку К.

3.4.2 Налаштування маршрутизації засобами Angular

Після реалізації розмітки та стилів необхідно налаштувати маршрутизацію. У Angular маршрутизація являє собою перехід від одного представлення до іншого в залежності від заданого URL. Причому навігація може здійснюватись і всередині представлення.

Нижче наведено уривок коду для реалізації маршрутизації.

```
const routes: Routes = [
  {
    path: "", component: AppLayoutComponent,
    children: [
```

```

    {path: '', component: AppComponent},
    {path: 'dashboard', component: DashboardComponent},
    {path: 'analysis', component: AnalysisComponent},
  ],
  canActivate: [AuthGuard]
},
{
  path: 'auth',
  loadChildren: () =>
import('./components/auth/auth.module').then(m => m.AuthModule) },
{ path: 'notfound', component: NotfoundComponent },
{ path: '**', redirectTo: '/notfound' },
];

```

Набір маршрутів збирається в масив `Routes` для визначення конфігурації маршрутизатора. Параметр `path` відповідає URL-адресі в адресному рядку браузера. Він повідомляє маршрутизатору зіставити цю URL-адресу зі шляхом: `'dashboard'` і відобразити `DashboardComponent`.

Обробник `CanActivate` використовується щоб визначити, чи дозволено поточному користувачеві активувати дочірній компонент. За замовчуванням будь-який користувач може це зробити.

Повний лістинг коду маршрутизації наведено в додатку II.

3.4.3 Інтеграція бази даних в інформаційно-аналітичну систему

База даних для інформаційно-аналітичної системи називається «`yummy_cakes`» та містить такі таблиці: `CLIENTS`, `PRODUCTS`, `ORDERS`, `ORDERSPROUCTS`, `WISHLIST`, `WAITLIST`, `ORDERCOMMENTS`, `PRODUCTCOMMENTS`, `ADVERTISINGCOSTS`.

Таблиця `CLIENTS` призначена для зберігання даних про клієнтів. Нижче наведено код для створення таблиці.

```

CREATE TABLE Clients
(
  Id INT PRIMARY KEY IDENTITY,

```



```

Age INT DEFAULT 18,
FirstName NVARCHAR(20) NOT NULL,
LastName NVARCHAR(20) NOT NULL,
Email VARCHAR(30) UNIQUE,
Address NVARCHAR(50),
Gender NVARCHAR(10)

```

```
);
```

Таблиця PRODUCTS призначена для зберігання даних про товари.

Лістинг коду PRODUCTS має таку структуру:

```

CREATE TABLE Products
(
    Id INT PRIMARY KEY IDENTITY,
    Name NVARCHAR(50) NOT NULL,
    Description NVARCHAR(250) NOT NULL,
    ImgUrl NVARCHAR(250),

```

```
);
```

Таблиця ORDERS містить такі поля:

```

CREATE TABLE Orders
(
    Id INT PRIMARY KEY IDENTITY,
    ClientId INT,
    CreatedAt DATE,
    FOREIGN KEY (ClientId) REFERENCES Clients (Id) ON DELETE

```

```
CASCADE
```

```
);
```

Таблиця призначена для зберігання даних про замовлення. Зв'язок між таблицями CLIENTS і ORDERS визначає відповідність замовлення до людини, яка його зробила.

Структура таблиці ORDERPRODUCTS:

```
CREATE TABLE OrdersProducts
```

```
(
    ProductId INT,
    OrderId INT,
    FOREIGN KEY (ProductId) REFERENCES Products (Id) ON DELETE
    CASCADE,
    FOREIGN KEY (OrderId) REFERENCES Orders (Id) ON DELETE
    CASCADE
);
```

В таблиці відображено id замовлення та id товару з цього замовлення. Таблиця була створена щоб були відсутні неключові стовпці, які залежать від інших неключових стовпців. Таким чином, застосувавши зв'язок «багато до багатьох», нормалізували базу даних до третьої нормальної форми.

Структура таблиці WISHLIST:

```
CREATE TABLE WishList
(
    ClientId INT,
    ProductId INT,
    CreatedAt DATE,
    FOREIGN KEY (ClientId) REFERENCES Clients (Id) ON DELETE
    CASCADE,
    FOREIGN KEY (ProductId) REFERENCES Products (Id) ON DELETE
    CASCADE
);
```

Таблиця призначена для зберігання даних про товари, додані в список бажаних. В цій таблиці також застосовано зв'язок «багато до багатьох», щоб нормалізувати базу до третьої нормальної форми.

Структура таблиці WAITLIST:

```
CREATE TABLE WaitList
(
    ClientId INT,
```

```

        ProductId INT,
        CreatedAt DATE,
        FOREIGN KEY (ClientId) REFERENCES Clients (Id) ON DELETE
CASCADE,
        FOREIGN KEY (ProductId) REFERENCES Products (Id) ON DELETE
CASCADE
    );

```

Таблиця призначена для зберігання даних про товари, які користувачі додали в список для очікування в наявності.

Структура таблиці ORDERCOMMENTS:

```
CREATE TABLE OrderComments
```

```

(
    Id INT PRIMARY KEY IDENTITY,
    OrderId INT,
    Comment NVARCHAR(250),
    CreatedAt DATE,
    FOREIGN KEY (OrderId) REFERENCES Orders (Id) ON DELETE
CASCADE
);

```

Таблиця призначена для зберігання даних про коментарі, які користувачі залишили до замовлень.

Структура таблиці PRODUCTCOMMENTS:

```
CREATE TABLE ProductComments
```

```

(
    Id INT PRIMARY KEY IDENTITY,
    ClientId INT,
    ProductId INT,
    Comment NVARCHAR(250),
    CreatedAt DATE,

```

FOREIGN KEY (ClientId) REFERENCES Clients (Id) ON DELETE CASCADE,

FOREIGN KEY (ProductId) REFERENCES Products (Id) ON DELETE CASCADE,

);

Таблиця призначена для зберігання даних про коментарі, які користувачі залишили до товарів.

Структура таблиці ADVERTISINGCOSTS:

CREATE TABLE AdvertisingCosts

(

Id INT PRIMARY KEY IDENTITY,

ProductId INT,

Price DECIMAL,

FOREIGN KEY (ProductId) REFERENCES Products (Id) ON DELETE

CASCADE,

);

Таблиця призначена для зберігання даних про суми витрат на рекламу.

Зв'язки між таблицями зображено на рисунку 3.7.

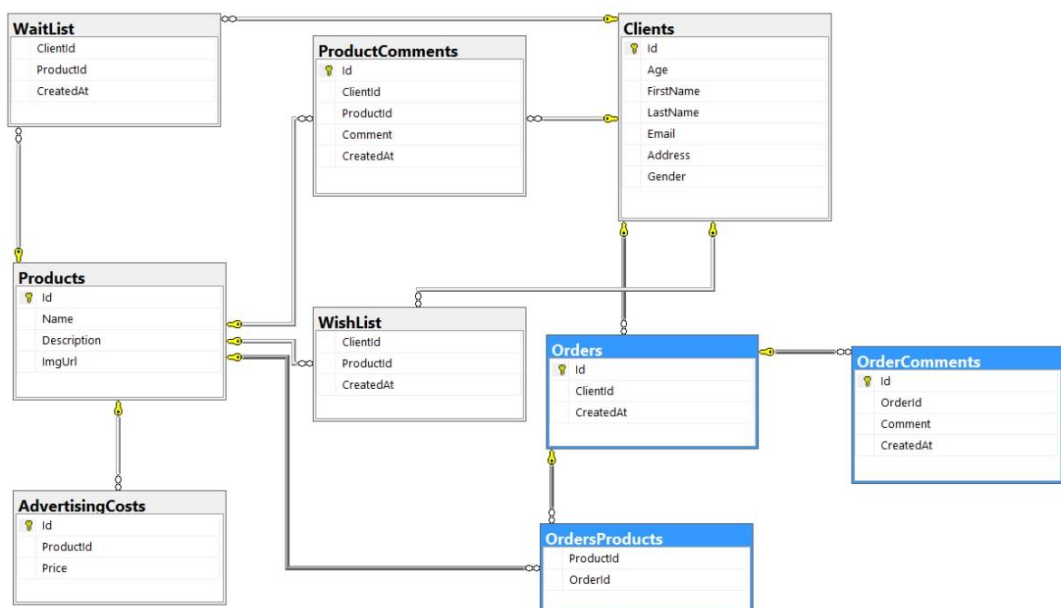


Рисунок 3.7 — Схема зв'язків між таблицями бази даних

3.4.4 Реалізація авторизації в інформаційно-аналітичній системі

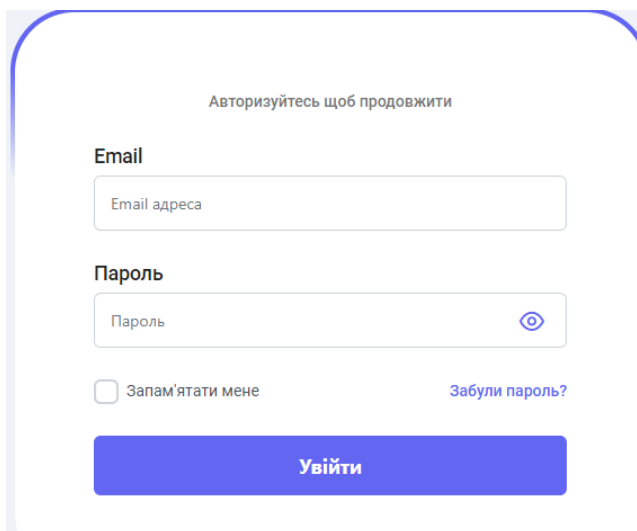
Перше, що відкривається для адміністратора при вході на адміністративну панель — це сторінка авторизації (рисунок 3.8). Без авторизації адміністратор не матиме доступу до перегляду веб-аналітики.

Модуль маршрутизації для авторизації забезпечує різні шляхи (рисунок 3.9 та рисунок 3.10), в залежності від того чи є у користувача доступ до системи чи виникла помилка. Лістинг коду наведений нижче.

```
@NgModule({
  imports: [RouterModule.forChild([
    { path: 'error', loadChildren: () => import('./error/error.module').then(m =>
m.ErrorModule) },
    { path: 'access', loadChildren: () =>
import('./access/access.module').then(m => m.AccessModule) },
    { path: 'login', loadChildren: () => import('./login/login.module').then(m =>
m.LoginModule) },
    { path: '**', redirectTo: '/notfound' }
  ])],
  exports: [RouterModule]
})
export class AuthRoutingModule { }
```

NgModule — це клас, позначений декоратором @NgModule. @NgModule приймає об'єкт метаданих, який описує, як скомпілювати шаблон компонента та як створити інжектор під час виконання. Він визначає власні компоненти модуля, директиви та канали, роблячи деякі з них загальнодоступними через властивість експорту, щоб зовнішні компоненти могли їх використовувати. @NgModule також може додавати постачальників послуг до інжекторів залежностей програм.


Після успішного входу адміністратор може перейти до сторінок адміністративної панелі «Dashboard» та «Statistics&Analysis».



Авторизуйтеся щоб продовжити

Email

Пароль

 
 Запам'ятати мене [Забули пароль?](#)

Увійти

Рисунок 3.8 — Вигляд вікна форми авторизації

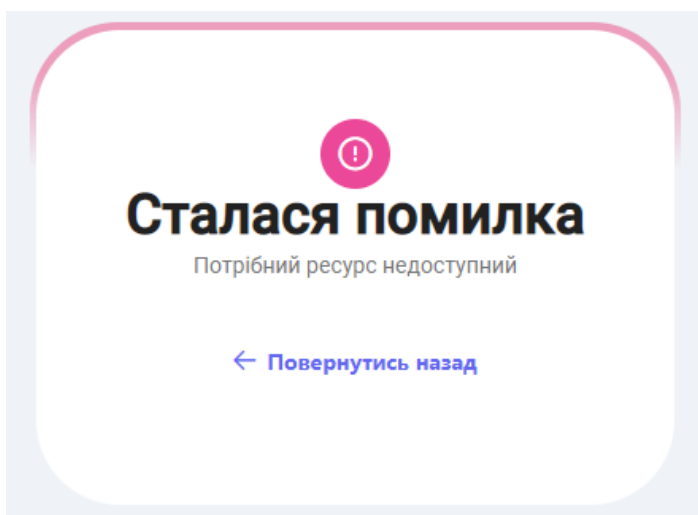


Рисунок 3.9 — Вигляд вікна повідомлення про помилку входу

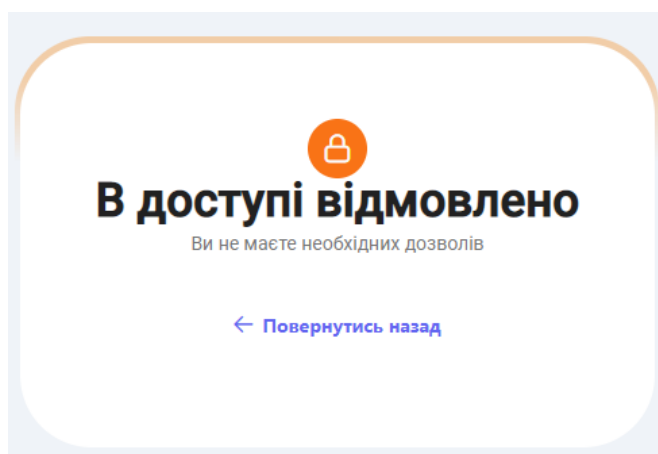


Рисунок 3.10 — Вигляд вікна повідомлення про відмову в доступі

3.5 Програмна реалізація окремих модулів інформаційно-аналітичної системи адміністрування веб-додатку

3.5.1 Реалізація сторінки «Dashboard»

Розглянемо детальніше компоненти сторінки «Dashboard».

Усі дані, які виводяться на екран у вигляді графіків, таблиць та ін., забираються з бази даних за допомогою методів `Accessor` і фільтруються відповідно заданим параметрам, наприклад тиждень, місяць, рік.

Методи `Accessor` забезпечують засоби, за допомогою яких можна отримати стан об'єкта з інших частин програми. Це кращий метод в об'єктно-орієнтованих парадигмах, оскільки забезпечує рівень абстракції, який приховує деталі реалізації наборів функціональних можливостей.

Графік «Статистика за користувачами» (рисунок 3.11) відображає кількість користувачів, які відвідали веб-додаток та кількість користувачів, які були зареєстровані в цей же період. Нижче наведено уривок коду:

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { LinearChartComponent } from './linear-chart.component';
describe('LinearChartComponent', () => {
  let component: LinearChartComponent;
  let fixture: ComponentFixture<LinearChartComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ LinearChartComponent ]
    })
    .compileComponents();
    fixture = TestBed.createComponent(LinearChartComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it('should create', () => {
```

```

    expect(component).toBeTruthy();
  });
});

```

Функція `beforeEach()` є асинхронною і викликає `TestBed.compileComponents`, оскільки `HeroDetailComponent` має зовнішній шаблон і файл `css`.

Метод `TestBed.configureTestingModule()` повертає клас `TestBed`, щоб була можливість ланцюжком викликати інші статичні методи `TestBed`, наприклад `compileComponents()`.

Функція `detectChanges()` повідомляє `Angular` запустити виявлення змін. Щоразу, коли він викликається, він оновлює прив'язки даних, і повторно відмальовує компонент на основі оновлених даних. Виклик цієї функції спричинить запуск `ngOnInit` лише під час першого виклику.

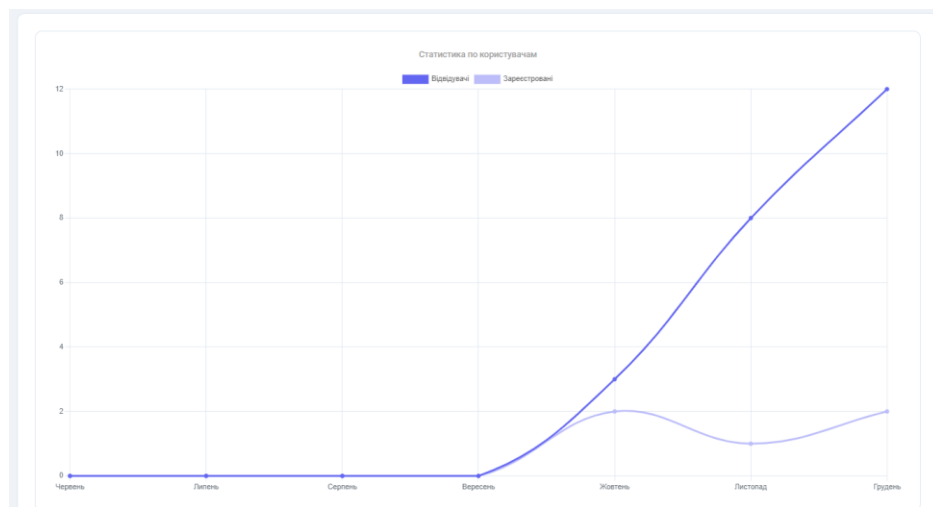


Рисунок 3.11 — Графік статистики відвідувачів та зареєстрованих користувачів

Наступний блок відображає статистику веб-додатку відповідно обраному періоду (рисунок 3.12). Фрагмент коду наведено нижче:

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { CardChartComponent } from './card-chart.component';
describe('CardChartComponent', () => {

```



```

let component: CardChartComponent;
let fixture: ComponentFixture<CardChartComponent>;
beforeEach(async () => {
  await TestBed.configureTestingModule({
    declarations: [ CardChartComponent ]
  })
  .compileComponents();
  fixture = TestBed.createComponent(CardChartComponent);
  component = fixture.componentInstance;
  fixture.detectChanges();
});
it('should create', () => {
  expect(component).toBeTruthy();
});
});

```

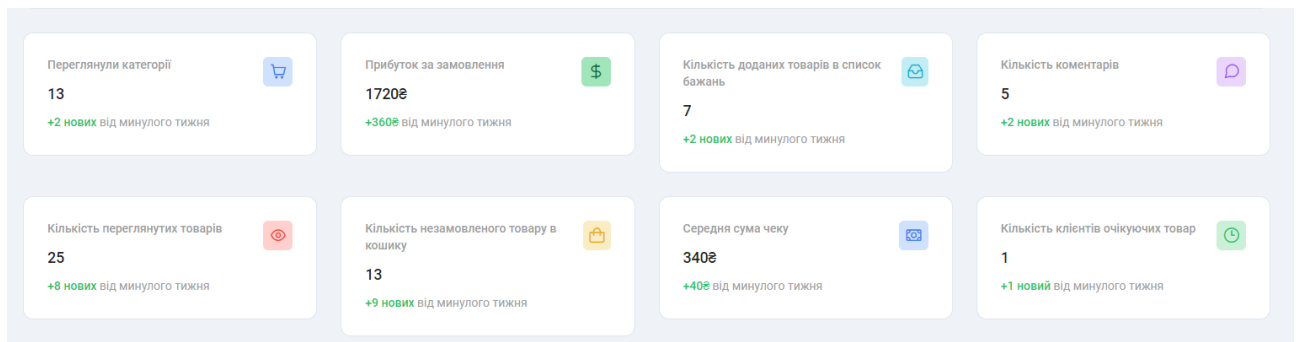


Рисунок 3.12 — Вигляд вікна з виведенням інформації по різним показникам веб-додатку

Даний блок містить такі показники:

- кількість користувачів, які переглянули категорії за певний період
- кількість користувачів, які переглянули товари за певний період
- прибуток за замовлення за певний період
- кількість доданих товарів у список бажань за певний період
- середня сума чеку за певний період;

- кількість користувачів, які чекають на наявність товару в магазині за певний період;
- кількість користувачів, які товар в корзину, але не замовили за певний період;
- кількість відгуків за певний період.

Далі відображено блок з найпопулярнішими позиціями на сайті (рисунок 3.13).

Представлені позиції мають попит серед клієнтів. Це визначається по кількості покупок по кожній позиції. Чим більша кількість — тим більший попит.

Наступний блок відображає останні 15 відгуків до замовлення та до товарів окремо (рисунок 3.14).

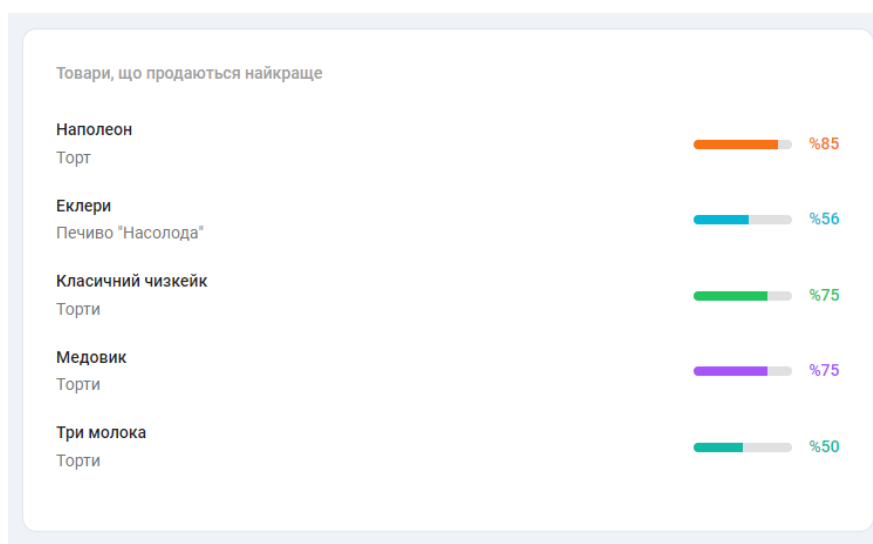


Рисунок 3.13 — Вигляд блоку найпопулярніших товарів

Відгуки до замовлень			
Id	ПІБ	Пошта	Відгук
1	Осадча Катерина	katyatorti@gmail.com	Чудесний тортік, раджу
2	Зеленський Володимир Олександрович	ZelenskyUA@gmail.com	Дуже смачно і швидко
3	Рудь Людмила	LudaRud@ukr.net	Не то шоб по вкусу вкусно но по суті вкусно
4	Олег Вінник	Vovchica@gmail.com	Сподобався молодій вовчці
5	Igor Коломойський	raveL_usik@gmail.com	Гарно, смачно, швидко!!!!

Items per page: 5 1 - 5 of 15 < >

Коментарі до товарів				
Id	ПІБ	Пошта	Товар	Відгук
1	Шкуренко Анастасія	nshkur@gmail.com	Наполеон	Дуже смачний тортік, замовлю ще!
2	Рудь Людмила	LudaRud@ukr.net	Медовик	Найкращий медовик!
3	Бугай Олекса	bugola@gmail.com	Печиво "Насолода"	Дитина в захваті!
4	Осадчий Ромка	sdchorr@mail.ru	Десерт "Три молока"	Трішки підгорів десертік, але файно!
5	Дженіфер Лопес	dlopez@gmail.com	Молочна дівчинка	Amazing! Beautiful to watch!

Items per page: 5 1 - 5 of 15 < >

Рисунок 3.14 — Вигляд таблиць відгуків до замовлень та окремих товарів

Залишаючи відгук на сайті, таблиця оновлюється і новий коментар додається в кінець списку таблиці. Перший відгук автоматично видаляється.

Повний лістинг коду для сторінки «Dashboard» наведено в додатку М.

3.5.2 Реалізація сторінки «Statistics&Analysis»

Сторінка містить графіки, які розраховуються відповідно до методів, що були проаналізовані в розділі 2.

Першим на сторінці розміщено графік, що показує як змінюється середній чек покупців. Тобто, на графіку відрисовуються дві лінії: перша відповідає за поточний період, інша — за минулий (рисунок 3.15). Порівнюючи графіки можна побачити чи збільшились продажі чи зменшились.

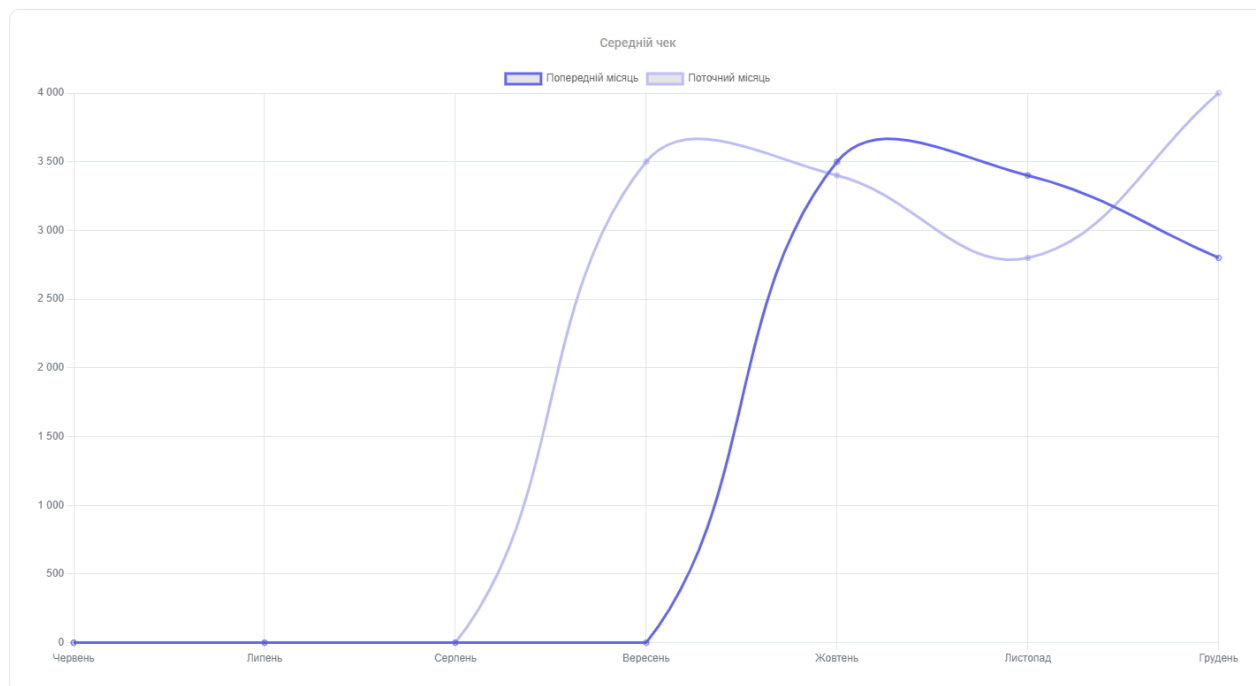


Рисунок 3.15 — Вигляд графіку статистики середнього чеку споживачів

Наступним зображено блок з RFM — аналізом. Система приймає дані клієнта і опрацьовує відповідно даному методу Accessor.

Метод accessor getRFMScoreForClient (рисунок 3.16) відправляє запит на отримання даних клієнта на сервері для подальшого опрацювання в контролері AnalysisController(). Після опрацювання метод повертає результат і встановлює

оцінку клієнту. Відповідно цій оцінці визначається приналежність користувача до сегменту.

Наступний блок — це блок розрахунку ABC аналізу. Функція `getABCAnalysisOnPeriod()` (рисунок 3.17) отримує дані про суму товару у відповідний період. Після цього відбувається розрахунок долі в обороті даного товару і записується в змінну `abcPartPercentage`. За допомогою функції `push` визначаємо відповідність до групи та записуємо в масив `result`.

```
import { Injectable } from '@angular/core';
import { HttpClient, HttpParams } from '@angular/common/http';

@Injectable()
export class AnalysisAccessor {
  constructor(
    protected httpClient: HttpClient
  ) { }

  public getRFMScoreForClient(clientId: number) {
    const params = new HttpParams()
      .set('clientId', clientId);
    return this.httpClient.get<any>(`https://localhost:4200/api/Analysis/rfmscore`, { params });
  }
}
```

Рисунок 3.16 — Вигляд функції `getRFMScoreForClient`

```
public async getABCAnalysisOnPeriod(period: number) {
  const soldProductsOnPeriod = await this.accessor.getSoldProductsOnPeriod(period);
  const totalIncome = soldProductsOnPeriod.reduce((sum, product) => sum + product.income, 0);
  let result = [];

  soldProductsOnPeriod.forEach(product => {
    const abcPartPercentage = product.income * 100 / totalIncome;
    result.push(
      {
        product,
        abcPartPercentage,
        abcRank: abcPartPercentage > 50 ? 'A' : abcPartPercentage > 15 ? 'B' : 'C'
      }
    );
  });
}
```

Рисунок 3.17 — Вигляд методу `getABCAnalysisOnPeriod()`

Наступним відображено графік лінійної регресії. Функція `getLinearRegression()` (рисунок 3.18) приймає дані про суму, витрачену на рекламу з accessor `getAdvertisementCostOfPeriod()`. Далі виконуються математичні функції розрахунку формул для отримання прогнозованого результату на наступний період.

Повний лістинг коду наведений в додатку Ж.

```
public async getLinearRegression(period: number, adSpentForNextMonth: number) {
  const advertisementCostWithSellsOfPeriod = await this.accessor.getAdvertisementCostOfPeriod(period);

  const b = (period * advertisementCostWithSellsOfPeriod.reduce((sum, entity) => entity.adCost * entity.sells, 0) -
    advertisementCostWithSellsOfPeriod.reduce((sum, entity) => sum + entity.adCost, 0) *
    advertisementCostWithSellsOfPeriod.reduce((sum, entity) => sum + entity.sells, 0)) /
    ((period - 1) * advertisementCostWithSellsOfPeriod.reduce((sum, entity) => sum + entity.adCost, 0) ** 2);
  const xAverage = advertisementCostWithSellsOfPeriod.reduce((sum, entity) => sum + entity.adCost, 0) / period;
  const yAverage = advertisementCostWithSellsOfPeriod.reduce((sum, entity) => sum + entity.sells, 0) / period;
  const a = yAverage - b * xAverage;

  return a + b * adSpentForNextMonth;
}
```

Рисунок 3.18 — Вигляд методу `getLinearRegression()`

3.6 Тестування інформаційно-аналітичної системи

Розглянемо як працює інформаційно-аналітична система на основі отриманих даних з інтернет-магазину.

Отримавши дані транзакцій клієнтів (таблиця 3.1) відбувається RFM-аналіз.

Таблиця 3.1 — Показники RFM

ID Клієнта	Давність останньої покупки (днів)	Сумарна частота покупок (кількість)	Сума, потрачена на покупки (всього)
1	25	5	1560 грн
2	4	19	5300 грн
3	6	3	890 грн
4	37	6	1420 грн
5	2	12	1300 грн
6	10	9	2700 грн

Наступним кроком клієнти отримують бали по критеріям: давність останньої покупки, сумарна частота покупок, сума, потрачена на покупки.

Ранжування клієнтів показано в таблиці 3.2.

Таблиця 3.2 — Ранжування клієнтів за давністю останньої покупки

ID Клієнта	Давність останньої покупки	R - бал
2	4	5
3	6	5
5	2	5
6	10	4
1	25	3
4	37	2

Як видно з наведеної вище таблиці, було відсортовано клієнтів за датою давності, а найновіші покупці знаходяться у верхній частині. Оскільки клієнтам призначаються бали від 1 до 5, клієнти 2, 3, 5 оцінюються в 5 балів, наступні клієнти 6 та 1 оцінюються в 4 бали, і так далі. Подібним чином можна відсортувати клієнтів за частотою від «найчастіші покупки» до «найменш часті покупки». Для грошового фактору клієнти з великими витратами оцінюються в 5 балів і з найменшими витратами оцінюються в 1 бал. Оцінювання F і M підсумовані нижче (таблиця 3.3 і 3.4):

Таблиця 3.3 — Ранжування клієнтів за сумарною частотою покупок

ID Клієнта	Сумарна частота покупок	F - бал
2	19	5
5	12	4
4	6	3
6	9	3
1	5	2
3	3	1

Таблиця 3.4 — Ранжування клієнтів за сумою, витраченою на покупки

ID Клієнта	Сума, потрачена на покупки	М - бал
2	5300 грн	5
6	2700 грн	4
1	1560 грн	3
4	1420 грн	3
5	1300 грн	3
3	890 грн	2

Тепер клієнти ранжуються, об'єднавши їхні індивідуальні рейтинги R, F та M, щоб отримати загальний бал RFM. Бал RFM, наведений в таблиці 3.5, є просто середнім значенням окремих оцінок R, F і M, отриманих шляхом надання однакових ваг кожному атрибуту RFM.

Таблиця 3.5 — Загальний бал RFM

ID Клієнта	RFM ваги	RFM оцінка
1	3, 2, 3	2.7
2	5, 5, 5	5.0
3	5, 1, 2	2.7
4	2, 3, 3	2.7
5	5, 4, 3	4
6	4, 3, 4	3.7

Вигляд блоку RFM-аналізу зображено на рисунку 3.19

Отримавши дані про суми продаж по окремим групам товарів (таблиця 3.6), система проводить ABC-аналіз.

Таблиця 3.6 — Визначення групи товарів та суми їх продажів

Товар	Жовтень	Листопад	Грудень	Підсумкова сума продаж
	Сума продаж	Сума продаж	Сума продаж	
Торти	30400 грн	28900 грн	24000 грн	83 300 грн
Тістечка	8890 грн	10400 грн	10780 грн	30 070 грн

Продовження таблиці 3.6

Печиво	2670 грн	4150 грн	3300 грн	10 120 грн
Цукерки	5750 грн	4800 грн	6000 грн	16 550 грн
Зефір	7340 грн	7270 грн	8500 грн	22 110 грн

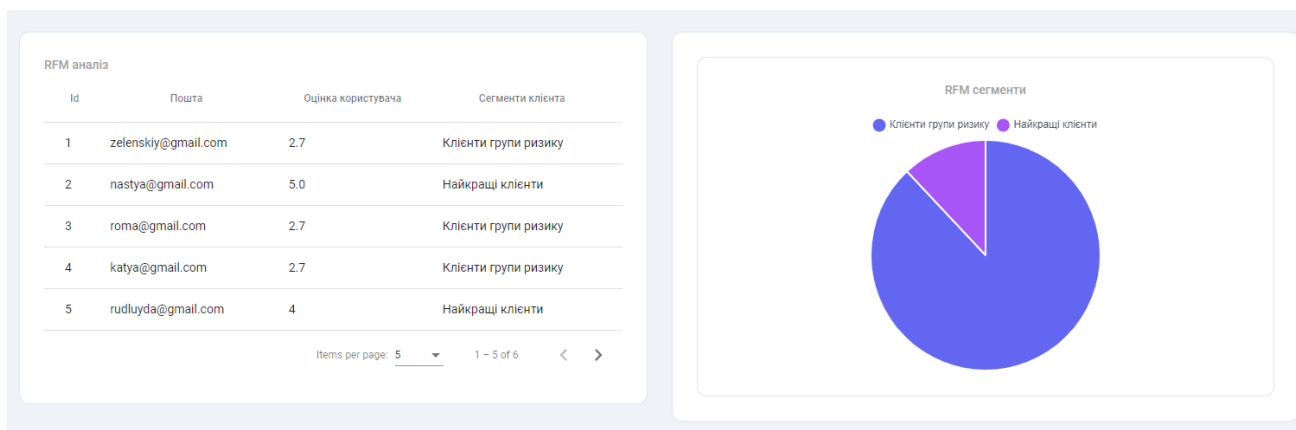


Рисунок 3.19 — Вигляд блоку RFM - аналізу

Обчислюється відсоток від загального обсягу продажів по кожній групі (таблиця 3.7) та сортується за зменшенням відсоткового значення. Наприклад, останні три місяці було продано печива на 10120 грн, а загальний дохід за цей період 162150 грн. Розрахуємо долю продажів печива в загальному обороті.

$$\text{Доля в обороті} = \frac{\text{Підсумкова сума продажі} \cdot 100}{\text{Загальний дохід}} = \frac{10120 \cdot 100}{162150} = 7,38\%$$

Таблиця 3.7 — Визначення долі товару у % в обороті та позначення належності до відповідної групи

Товар	Жовтень	Листопад	Грудень	Підсумкова сума продаж	Доля в обороті	ABC
	Сума продаж	Сума продаж	Сума продаж			
Торти	30400 грн	28900 грн	24000 грн	83 300 грн	51,37%	A
Тістечка	8890 грн	10400 грн	10780 грн	30 070 грн	21,92%	B
Зефір	7340 грн	7270 грн	8500 грн	22 110 грн	16,12%	B
Цукерки	5750 грн	4800 грн	6000 грн	16 550 грн	12,07%	C
Печиво	2670 грн	4150 грн	3300 грн	10 120 грн	7,38%	C

Отже, отримавши дані результати можна сказати, що вигідно продавати торти. Тістечками та зефіром можна підтримувати асортимент і, можливо, є сенс його розширити. А ось з печиво та цукерки тих видів, які рідко купляють, необхідно вивести з асортименту.

Вигляд блоку ABC-аналізу в адміністративній панелі зображено на рисунку 3.20.

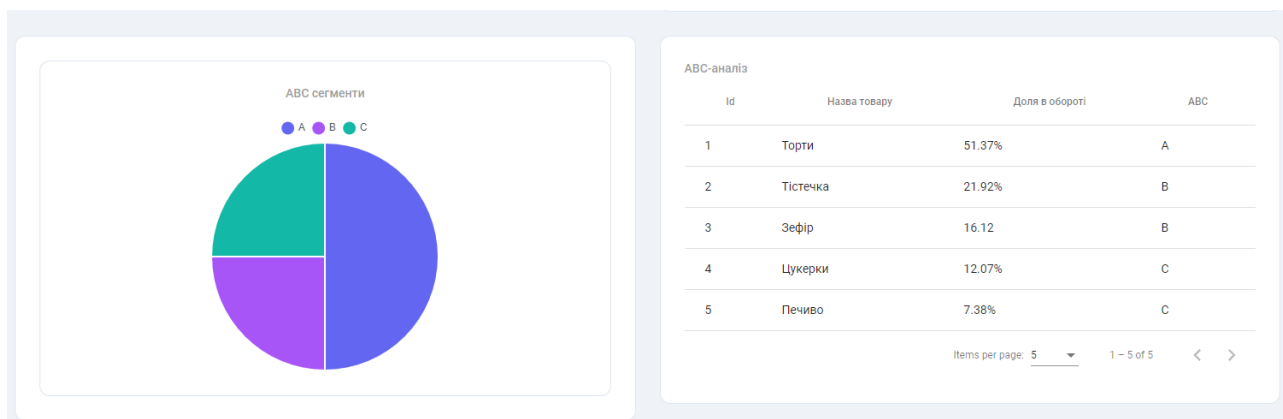


Рисунок 3.20 — Вигляд блоку ABC-аналізу

Отримавши дані про витрати на рекламу та продажі за кожен місяць (таблиця 3.8), система проводить аналіз лінійної регресії.

Таблиця 3.8 — Вихідні дані з інтернет-магазину

Місяць	Витрати на рекламу, грн, x	Продажі, грн, y
Вересень	5000	54900
Жовтень	3050	55050
Листопад	4500	55520
Грудень	2700	55580
Σ	15250	221050

Розраховуються значення b за формулою 2.4:

$$b = \frac{4 \cdot 842308500 - (15250 \cdot 221050)}{4 \cdot 61842500 - 15250^2} = -0,12011$$

Визначаються середні значення x та y :

$$x = \frac{15250}{4} = 3812,5, y = \frac{221050}{4} = 55262,5$$

Розраховується значення a за формулою 2.3:

$$a = 55262,5 - (-0,12011) * 3812,5 = 55720,42 \text{ грн}$$

Підраховується очікуваний обсяг продажів у майбутньому періоді за формулою 2.1, припускаючи, що в наступному місяці оплата за рекламу буде 3000 грн:

$$y_5 = 55720,42 + (-0,12011) * 3000 = 55360,09 \text{ грн}$$

Отже, при виділенні на рекламу наступного місяця 3000 грн прогнозний обсяг продажу становитиме 55360,09 грн

Вигляд блоку лінійної регресії для прогнозування продажів в адміністративній панелі зображено на рисунку 3.21.

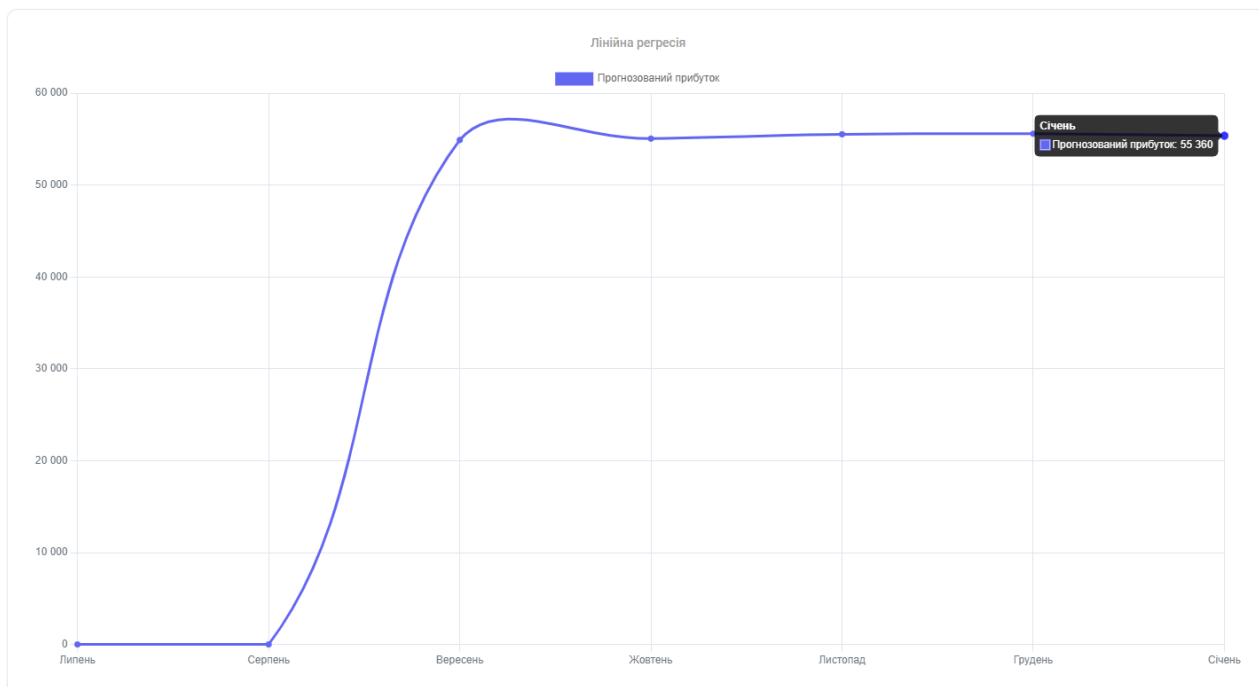


Рисунок 3.21 — Вигляд блоку з прогнозуванням прибутку

В результаті можна побачити, що в наступному місяці може бути незначний спад прибутку, але це не настільки критично для бізнесу, тому можна не змінювати стратегію продажів.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного продукту інформаційно-аналітичної системи адміністрування веб-додатку з візуалізацією даних.

Особливістю програми є те, що дана технологія є сучасною інформаційно-аналітичною системою (ІАС) для забезпечення керівників, аналітиків та менеджерів інформацією у всіх аспектах діяльності бізнесу для його подальшої оцінки та аналізу.

Аналогом може бути Fathom, ціна за базовий тариф 26400 грн/рік.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із таблиця 4.1.

Таблиця 4.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
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					
2	Багато аналогів на малому ринку	Ринкові п 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 аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно 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	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на 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 із комерційної реалізації ідеї
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	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х

Продовження 4.1

12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію	Процедура отримання дозвільних документів для виробництва та реалізації	Необхідно тільки повідомлення відповідним органам про виро-	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту
----	---	--	---	---	---

Усі дані по кожному параметру занесено в таблиці 4.2

Таблиця 4.2 — Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	3	4
Наявність аналогів на ринку	3	3	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	3	3	2
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	4	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	3	4	4
Сума	42	42	42
Середньоарифметична сума балів	$(42+42+42) / 3 = 42$		

За даними таблиці 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 4.3.

Таблиця 4.3 - 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

Продовження таблиці 4.3

31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що програмний продукт є інформаційно-аналітичною системою адміністрування веб-додатку з візуалізацією даних. Особливістю програми є те, що дана технологія є сучасною інформаційно-аналітичною системою (ІАС) для забезпечення керівників, аналітиків та менеджерів інформацією у всіх аспектах діяльності бізнесу для його подальшої оцінки та аналізу.

4.2 Прогнозування витрат на виконання науково-дослідної роботи

4.2.1 Розрахунок основної заробітної плати розробників

Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M — місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p — число робочих днів в місяці, 20 днів;

t — число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 4.4

Таблиця 4.4 — Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	48000	2400,00	32	76800,000

Продовження таблиці 4.4

Програміст	47000	2350,00	32	75200,000
Всього				152000,00

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

4.2.2 Розрахунок додаткової заробітної плати розробників

Додаткова заробітна плата прийнято розраховувати як 11 % від основної заробітної плати розробників та робітників:

$$З_д = З_о \cdot 11 \% / 100 \% \quad (4.2)$$

$$З_д = (152000,00 \cdot 11 \% / 100 \%) = 16720,00 \text{ (грн.)}$$

4.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22 \% / 100\% \quad (4.3)$$

$$Н_з = (152000,00 + 16720,00) \cdot 22 \% / 100 \% = 37118,40 \text{ (грн.)}$$

4.2.4. Розрахунок витрат на матеріали

Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

4.2.5 Амортизація обладнання

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для

розробки розраховується за формулою:

$$A = \frac{Ц}{T\vartheta} \cdot \frac{t_{\text{вик}}}{12} \text{ [грн.]}, \quad (4.4)$$

де Ц — балансова вартість обладнання, грн.;

T — термін корисного використання обладнання згідно податкового законодавства, років

$t_{\text{вик}}$ — термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 21944 грн., термін його корисного використання згідно податкового законодавства — 2 роки, а термін його фактичного використання — 1,60 міс.

$$A_{\text{обл}} = \frac{21944}{2} \times \frac{1,6}{12} = 1462,93 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.5.

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн. (Windows 10 Home), то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $B_{\text{нем.ак.}} = 5399$ грн.

Таблиця 4.5 — Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк використання, років	Термін використання, місяців	Амортизаційні відрахування, грн.
Комп'ютерна периферія (HP Латор 15s-eq1003ua)	21944	2	1,60	1462,933

Продовження таблиці 4.5

Офісне обладнання (меблі)	21500	4	1,60	716,667
Приміщення	900000	20	1,60	6000,000
Всього				8179,60

4.2.6 Витрати на електроенергію

Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi}, \quad (4.5)$$

де V — вартість 1 кВт-години електроенергії для 1 класу підприємства, $V = 6,2$ грн./кВт;

P — встановлена потужність обладнання, кВт. $P = 0,5$ кВт;

Φ — фактична кількість годин роботи обладнання, годин.

K_{Π} — коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$V_e = 0,9 \cdot 0,5 \cdot 8 \cdot 32 \cdot 6,2 = 714,24 \text{ (грн.)}$$

4.2.7 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші

витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_e = (3_o + 3_p) \cdot \frac{H_{ib}}{100\%}, \quad (4.6)$$

де H_{ib} — норма нарахування за статтею «Інші витрати».

$$I_e = 152000,00 * 63\% / 100\% = 95760 \text{ (грн.)}$$

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (3_o + 3_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.7)$$

де $H_{нзв}$ — норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 152000,00 * 127\% / 100\% = 193040 \text{ (грн.)}$$

4.2.8 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$\begin{aligned} B_{заг} &= 152000,00 + 16720,00 + 37118,40 + 8179,60 + 5399 + 714,24 + 95760 + 193040 \\ &= 508931,24 \text{ грн.} \end{aligned}$$

4.2.9 Розрахунок загальних витрат на науково-дослідну роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються ZB , визначається за формулою:

$$ЗВ = \frac{B_{заг}}{\eta} \quad (\text{грн}), \quad (4.8)$$

де η — коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 508931,24 / 0,5 = 1017862 \text{ грн.}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

4.3.1 Розробка програмного засобу для використання масовим споживачем

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.9)$$

де $\pm\Delta\Pi_0$ — зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N — кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_0 — основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_0 = \Pi_6 \pm \Delta\Pi_0$;

Π_6 — вартість програмного продукту у році до впровадження результатів розробки;

ΔN — збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

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

p — коефіцієнт, який враховує рентабельність продукту;

ϑ — ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 15500 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 500 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року — на 1200 шт., протягом другого року — на 1100 шт., протягом третього року на 1000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*500 + (15500 + 500) * 1200) * 0,8333 * 0,33 * (1 - 0,18) = 4194299,832 \text{ грн.}$$

$$\Delta\Pi_2 = (0*500 + (15500 + 500) * (1200+1100)) * 0,8333 * 0,33 * (1 - 0,18) = 8298399,668 \text{ грн.}$$

$$\Delta\Pi_3 = (0*500 + (15500 + 500) * (1200+1100+1000)) * 0,8333 * 0,33 * (1 - 0,18) = 11906399,524 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 24399099,02 грн.

4.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Розраховуємо приведену вартість збільшення всіх чистих прибутків *ПП*.

Їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (4.10)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t — період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\begin{aligned} ПП &= (4194299,832/(1+0,1)^1) + (8298399,668/(1+0,1)^2) + (11906399,524/(1+0,1)^3) \\ &= 3812999,85 + 6858181,544 + 8945454,188 = 19616635,58 \text{ грн.} \end{aligned}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} * ЗВ, \quad (4.11)$$

де $k_{инв}$ — коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{инв} = 2 \dots 5$, але може бути і більшим;

$ЗВ$ — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 1017862 = 2035724,96 \text{ грн.}$$

Тоді абсолютний економічний ефект E_{abc} або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = ПП - PV. \quad (4.12)$$

$$E_{abc} = 19616635,58 - 2035724,96 = 17580910,62 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_e . Для цього використаємо формулу:

$$E_e = \sqrt[T_{ж}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.13)$$

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

$$E_e = \sqrt[3]{(1 + 17580910,62/2035724,96)} - 1 = 1,128$$

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

$$\tau = d + f, \quad (4.14)$$

де d — середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = (0,09 \dots 0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05 \dots 0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

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:

$$T_{ок} = \frac{1}{E_b}, \quad (4.15)$$

$$T_{ок} = 1 / 1,128 = 0,89 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,89 роки, то фінансування даної наукової розробки є доцільним.

ВИСНОВКИ

Магістерська кваліфікаційна робота присвячена розробці інформаційно-аналітичної системи для адміністрування веб-додатку з можливістю візуалізації даних.

В магістерській роботі проведено аналітичний аналіз сучасних технологій для побудови веб-додатків, таких як Angular, WEB API, Bootstrap, SQL і та ін. Також було проаналізовано особливості інформаційно-аналітичних систем, розглянуто їх види. На основі порівняльного аналізу аналогів інформаційно-аналітичних систем та виявлених недоліків було вирішено розробити більш функціональну інформаційно-аналітичну систему адміністрування веб-додатку, яка буде зручною для користування та візуалізації даних.

В другому розділі було проаналізовано та обрано методи для побудови інформаційно-аналітичної системи. Було розглянуто та описано методи для проведення статистичного RFM-аналізу, ABC-аналізу, лінійної регресії, використання яких дало можливість вдосконалити систему візуалізації даних розробленого веб-додатку. Також було проаналізовано сучасні засоби для візуалізації даних.

В третьому розділі розроблено інформаційно-аналітичну систему адміністрування веб-додатку. Для цього було обрано середовище Visual Studio Code. Представлено структуру інтернет-магазину, структуру інформаційно-аналітичної системи, спроектовано структуру бази даних для збереження каталогу товарів, інформації про клієнтів, їх замовлення. Систему було розроблено за допомогою таких технологій як Bootstrap, Angular, продумано дизайн, надано інтерактивності. Для зручної взаємодії з сервером та базою даних було застосовано мову програмування С#. Розроблено адміністративну панель, яка дозволяє відслідковувати різні параметри веб-сайту, отримувати дані про попит на товар, сегментувати товар та клієнтів на категорії, прогнозувати майбутній прибуток. Також проведено тестування розробленої інформаційно-

аналітичної системи для адміністрування веб-додатку, проаналізовано роботу адміністративної панелі.

У четвертому розділі проведені дослідження та розрахунки доцільності розробки та використання розробленої веб-системи. Згідно проведених досліджень рівень комерційного потенціалу розробки становить 42 бала, що свідчить про рівень розробки вище середнього. Спрогнозовано орієнтовну величину витрат по кожній з статей витрат, чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків видно, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,89 роки.

Отже, всі поставлені в магістерській кваліфікаційній роботі задачі виконані в повному обсязі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Інтегрована інформаційно-аналітична система: проект та концепція [Електронний ресурс]. — режим доступу: URL: https://pidru4niki.com/1031062453915/informatika/integrovana_informatsiyno-analitichna_sistema_proekt_kontsepsiya
2. Основи інформаційно-аналітичної діяльності: навч. посібник / Захарова В. І., Філіпова В. Я., — К. : «Центр учбової літератури», 2013. — 336 с.
3. Береза А. М. Основи створення інформаційних систем: навч. посібник / А. М. Береза. — 2-е видання, перероблене і доповнене — К.: КНЕУ, 2001. — 214 с.
4. Золотов С. Ю. Основи проектування інформаційних систем: [Навчальний посібник] / С. Ю. Золотов., 2007. — 96 с. 10.
5. Косова Т.Д. Організація і методика економічного аналізу. Навч. посіб. / Т.Д. Косова, П.М. Сухарев, Л.О. Ващенко — К.: Центр учбової літератури, 2012. — 528 с.
6. Репкіна, О. Б. Використання інформаційно-аналітичних систем для підвищення ефективності управління підприємницькими структурами / О. Б. Репкіна. - Текст: безпосередній // Молодий учений. - 2011. - № 1 (24). - С. 98-100. - URL: <https://moluch.ru/archive/24/2548/>
7. Аналітика веб-сайту Рівне, статистика та аналіз [Електронний ресурс] /Веб-місто. — Режим доступу: <http://webmisto.rv.ua/2016/09/01/analityka-web-saitu/>
8. Web-аналітика: що це і як її використовувати? [Електронний ресурс] /Веб-місто. — Режим доступу: <https://blog.a>
9. Основи статистики та аналізу даних [Електронний ресурс]. — режим доступу: URL: <https://socialdata.org.ua/manual/manual4/#ch3>
10. The Good and the Bad of C# Programming [Електронний ресурс]. — режим доступу: URL: <https://www.altexsoft.com/blog/c-sharp-pros-and-cons/>

11. What is Web API? [Электронный ресурс]. — режим доступа: URL: <https://www.tutorialsteacher.com/webapi/what-is-web-api>

12. MVC Framework — Introduction [Электронный ресурс]. — режим доступа: URL: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

13. Entity Framework using C# [Электронный ресурс]. — режим доступа: URL: <https://www.c-sharpcorner.com/article/entity-framework-introduction-using-c-sharp-part-one/>

14. Why use a database? [Электронный ресурс]. — режим доступа: URL: <https://www.futurelearn.com/info/c>

15. What is SQL? [Электронный ресурс]. — режим доступа: URL: <https://www.sqlcourse.com/beginner-course/what-is-sql/>

16. MySQL — Introduction [Электронный ресурс]. — режим доступа: URL: <https://www.tutorialspoint.com/mysql/mysql-introduction.htm>

17. What are HTML, CSS, & Javascript? [Электронный ресурс]. — режим доступа: URL: <https://www.ciat.edu/blog/html-css-javascript/>

18. Web Design 101: How HTML, CSS, and JavaScript Work [Электронный ресурс]. — режим доступа: URL: <https://blog.hubspot.com/marketing/web-design-html-css-javascript>

19. What is Angular Used For and When Should You Use it Instead of Other Frontend Technology? [Электронный ресурс]. — режим доступа: URL: <https://neoteric.eu/blog/what-is-angular-used-for-and-when-should-you-use-it/>

20. What is React.js? (Uses, Examples, & More) [Электронный ресурс]. — режим доступа: URL: <https://blog.hubspot.com/website/react-js>

21. What is Vue.js, and Why is it Cool? [Электронный ресурс]. — режим доступа: URL: https://linuxhint.com/about_vue_js/

22. Sprout Social [Электронный ресурс]. — режим доступа: URL: <https://sproutsocial.com>

23. Fathom [Электронный ресурс]. — режим доступа: URL: <https://usefathom.com/ref/TMPY0A>

24. Plausible [Електронний ресурс]. — режим доступу: URL: <https://plausible.io/>

25. Heap [Електронний ресурс]. — режим доступу: URL: <https://heap.io/>

26. Що таке web аналітика. Введення в web-аналітику. Сервіси веб аналітики [Електронний ресурс]. — режим доступу: URL: <https://sukachoff.ru/uk/virusy/chto-takoe-web-analitika-vvedenie-v-web-analitiku-servisy-veb/>

27. Веб-аналітика: Сегментація аудиторії [Електронний ресурс]. — режим доступу: URL: <https://aweb.ua/blog/segmentaciya-auditorii/>

28. RFM analysis (recency, frequency, monetary) [Електронний ресурс]. — режим доступу: URL: <https://www.techtarget.com/searchdatamanagement/definition/RFM-analysis>

29. ABC-аналіз [Електронний ресурс]. — режим доступу: URL: <https://www.unisender.com/ru/glossary/chto-takoe-abc-analiz-i-kak-ego-provesti/>

30. Data Mining Techniques [Електронний ресурс]. — режим доступу: URL: <https://www.javatpoint.com/data-mining-techniques>

31. Highcharts [Електронний ресурс]. — режим доступу: URL: <https://www.highcharts.com/blog/tutorials/highcharts-with-angular-v14/>

32. What is Bootstrap? [Електронний ресурс]. — режим доступу: URL: <https://www.techtarget.com/whatis/definition/bootstrap>

33. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. — Вінниця : ВНТУ, 2021. — 42 с.

34. Особливості створення інформаційно-аналітичної системи для адміністрування веб-додатку / А. В. Шкуренко, О. В. Войцеховська // Матеріали Всеукраїнської науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2023): Тез. доп. — Вінниця, 2023. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2023/paper/viewFile/16898/14091>

35. Шкуренко А. В. Аналіз сучасних технологій розробки інтерактивних веб-інтерфейсів / Шкуренко А. В. // Матеріали І науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії — ВНТУ, 2021. [Електронний ресурс]. — Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2021/paper/view/12006/10007>

ДОДАТОК А

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

ЗАТВЕРДЖУЮ
Завідувач кафедри
обчислювальної техніки
_____ проф., д.т.н. О. Д. Азаров
“ ___ ” _____ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи
«Інформаційно-аналітична система адміністрування веб-додатку з візуалізацією
даних»

08-23.МКР.016.00.000 ТЗ

Керівник к.т.н., доц. каф. ОТ.
_____ Войцеховська О. В.
Студентка групи 1КІ-21м
_____ Шкуренко А.В.

Вінниця 2022

1 Підстави для виконання дипломної кваліфікаційної роботи (МКР) включають в себе:

— актуальність даної теми полягає в створенні інформаційно-аналітичної системи для адміністрування веб-додатку, що дозволить відслідковувати статистику користування веб-додатком, аналізувати попит на продукцію;

— наказ про затвердження теми дипломної роботи.

2 Метою та призначенням МКР являються:

— мета — розробка інформаційно-аналітичної системи для адміністрування веб-сайту кондитерської продукції за рахунок розробки системи веб-аналітики, застосування знань та вмінь, отриманих у процесі навчання, на практиці;

— призначення розробки — за допомогою даної розробки адміністратор зможе отримувати необхідну інформацію про попит на продукцію, відслідковувати зроблені замовлення та отримувати аналітичні дані та прогнози на наступні періоди.

3 Джерела розробки МКР наступні:

Середовище розробки Visual Studio Code та засоби для реалізації візуалізації даних Angular HighCharts.

4 Технічні вимоги до виконання МКР включають в себе:

— наявність інформаційно-аналітичної системи для адміністрування веб-додатку;

— наявність інтернет-магазину для отримання даних;

5 Етапи роботи та очікувані результати приведено в таблиці А.1.

Таблиця А.1 — Етапи виконання роботи

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	2	3	4	5
1	Аналітичний огляд сучасних технологій розробки інформаційно-аналітичних систем	09.09.22	21.09.22	1 розділ

Продовження таблиці А.1

1	2	3	4	5
2	Аналіз методів для розробки інформаційно-аналітичної системи адміністрування веб-додатку	22.09.22	01.10.22	2 розділ, аналіз методів
3	Розробка структури інформаційно-аналітичної системи та веб-додатку	02.10.22	10.10.22	3 розділ, розробка структури
4	Програмна реалізація інформаційно-аналітичної системи адміністрування веб-додатку	11.10.22	20.11.22	3 розділ
5	Підготовка економічної частини	21.11.22	01.12.22	4 розділ
6	Оформлення пояснювальної записки, графічного матеріалу і/або презентації	02.12.22	21.12.22	ПЗ, презентація

6 До захисту МКР подаються: пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, рецензія опонента, протоколи складання державних екзаменів, анотації до МКР українською та англійською мовами, нормоконтроль про відповідність оформлення МКР діючим вимогам.

7 Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8 Вимоги викладені в МЕТОДИЧНИХ ВКАЗІВКАХ до дипломного проектування, ДСТУ 3008-2015, ДСТУ 3974-2000 «Правила виконання дослідно-конструкторських робіт. Загальні положення» та діючого ГОСТ 2.114-95 ЄСКД.

9 Вимоги щодо технічного захисту інформації в МКР з обмеженим доступом відсутні.

Технічне завдання до виконання отримала _____ Шкуренко А. В.

ДОДАТОК Б

Структурна схема інтернет-магазину



Рисунок Б.1 — Структурна схема інтернет-магазину

ДОДАТОК В

Вигляд сторінок інтернет-магазину

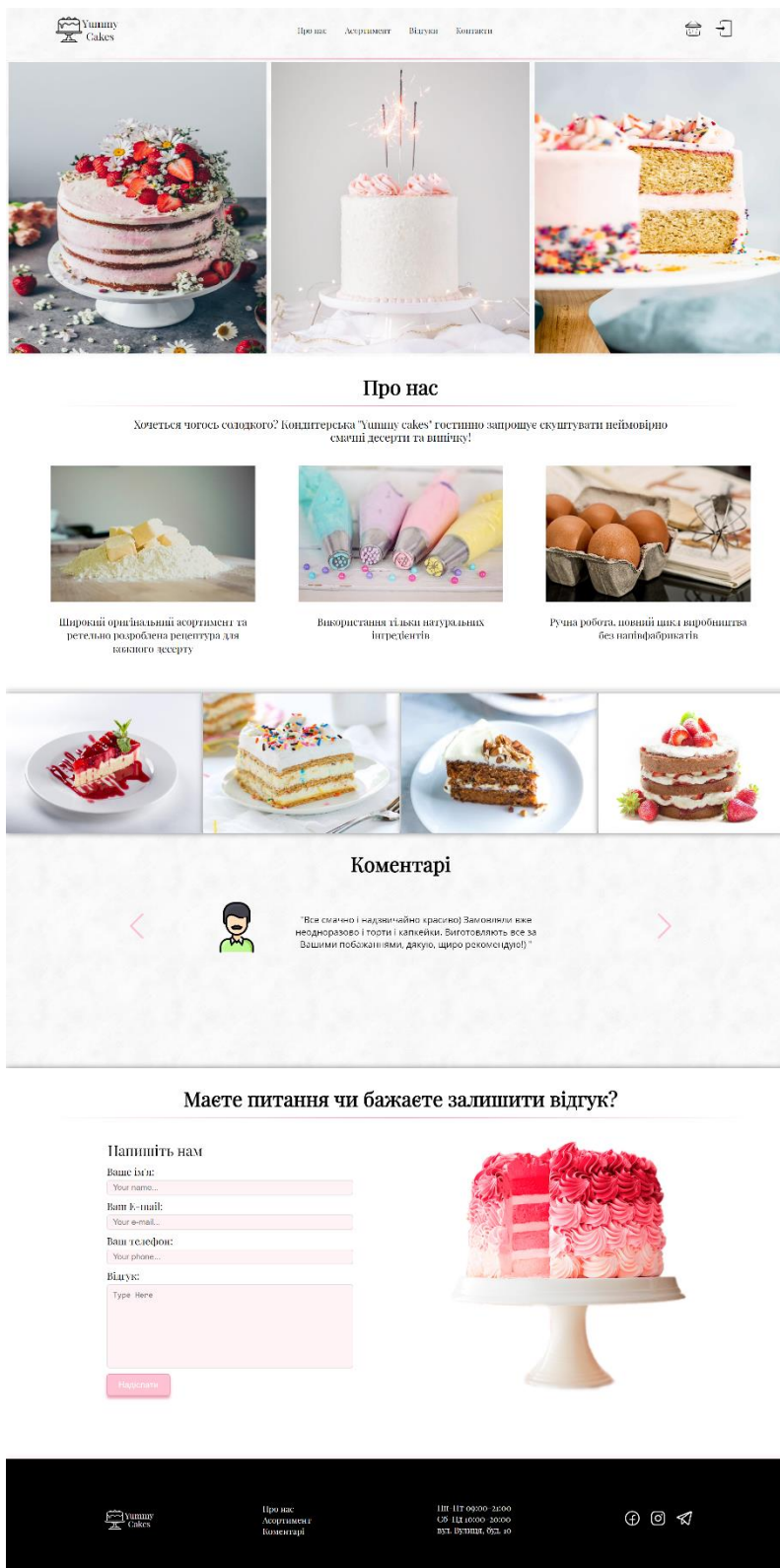


Рисунок В.1 — Вигляд головної сторінки веб-додатку

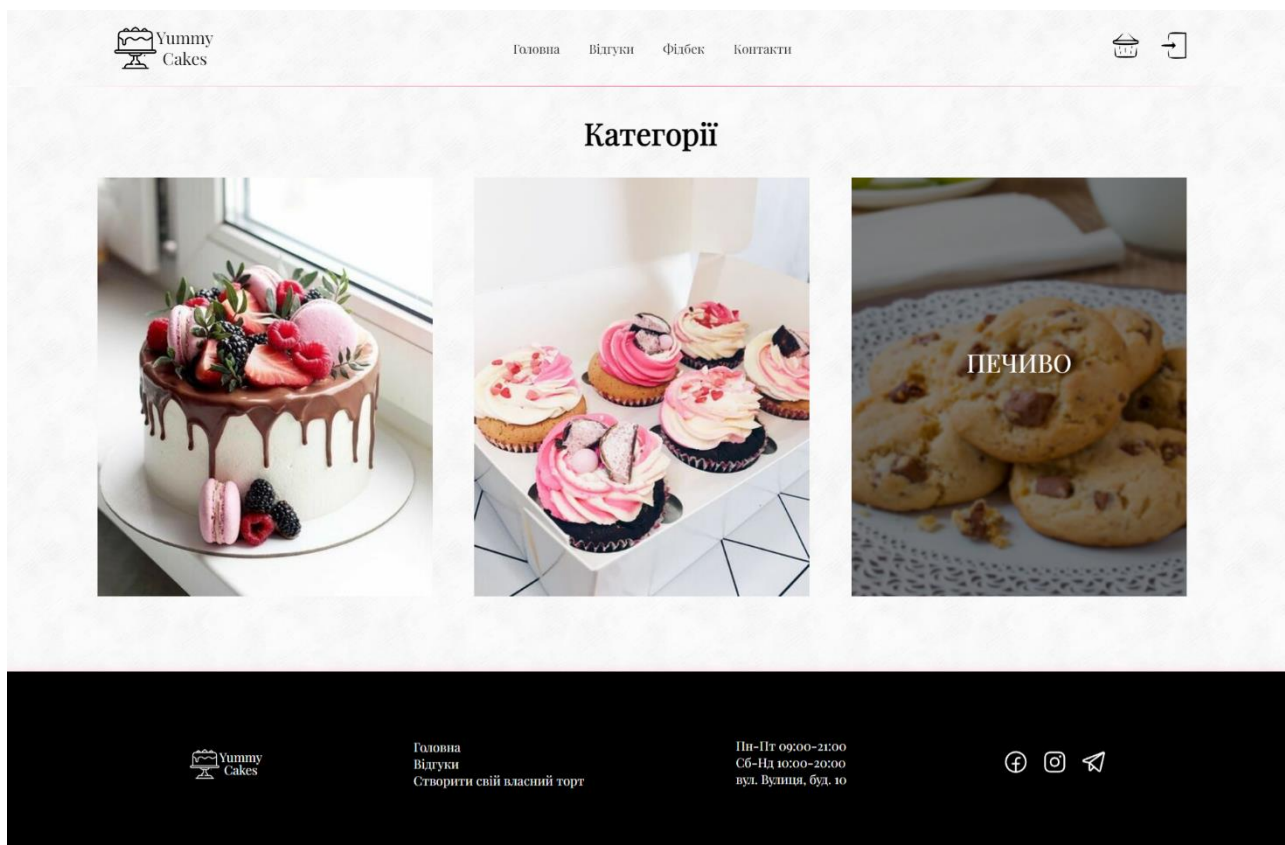





Рисунок В.2 — Вигляд сторінки категорій товарів




Yummy
Cakes

Головна Відгуки Фідбек Контакти

Торти




Наполеон

Класичні листкові коржі з прошарком заварного крему, в якій додано вершкове масло та збиті вершки.

2.5 кг - 301 €

В корзину




Медовик

Класичні медові коржі з прошарком заварного крему з додаванням сметани.

1.6 кг - 255 €

В корзину




Шоколадний чізкейк

В основі торту корж з шоколадного пісочного тіста, в середині ніжна сирна маса з вершкового сиру з додаванням шоколаду.

1.4 кг - 140 €

В корзину




Ванільний чізкейк

В основі торту корж з пісочного тіста, в середині ніжна сирна маса з вершкового сиру з додаванням ванільного дукру.

1.2 кг - 400 €

В корзину




Спартак

Медові коржі з додаванням какао з прошарком заварного крему з додаванням сметани.

2 кг - 310 €

В корзину




Молочна дівчинка

Бісквітні коржі на основі згущеного молока з прошарком заварного крему з додаванням збитих вершків.

2.5 кг - 240 €

В корзину



Yummy
Cakes

Головна
Асортимент
Відгуки
Створити свій власний торт

Пн-Пт 09:00-21:00
Сб-Нд 10:00-20:00
вул. Вулиця, буд. 10








Рисунок В.3 — Вигляд сторінки каталогу тортів

Yummy Cakes Назад

Оформити замовлення

Ваше ім'я:
Your name...

Ваш E-mail:
Your e-mail...

Ваш телефон:
Your phone...

Спосіб доставки:

Дата отримання замовлення:
 dd. mm. yyyy

Коментар до замовлення:
 Type Here

Ваш товар № 1:
Полуничні капкейки

Кількість:
1

Ціна:
150

Ваш товар № 2:
Наполеон

Кількість:
2

Ціна:
301

Сума:
752

Рисунок В.4 — Вигляд форми оформлення замовлення

Yummy Cakes Назад

Авторизуватись

Логін або номер мобільного
Login

Пароль
Password

[Забули пароль?](#) [Зареєструватись](#)

Рисунок В.5 — Вигляд сторінки авторизації

ДОДАТОК Г

UML-діаграма варіантів використання інтернет-магазину

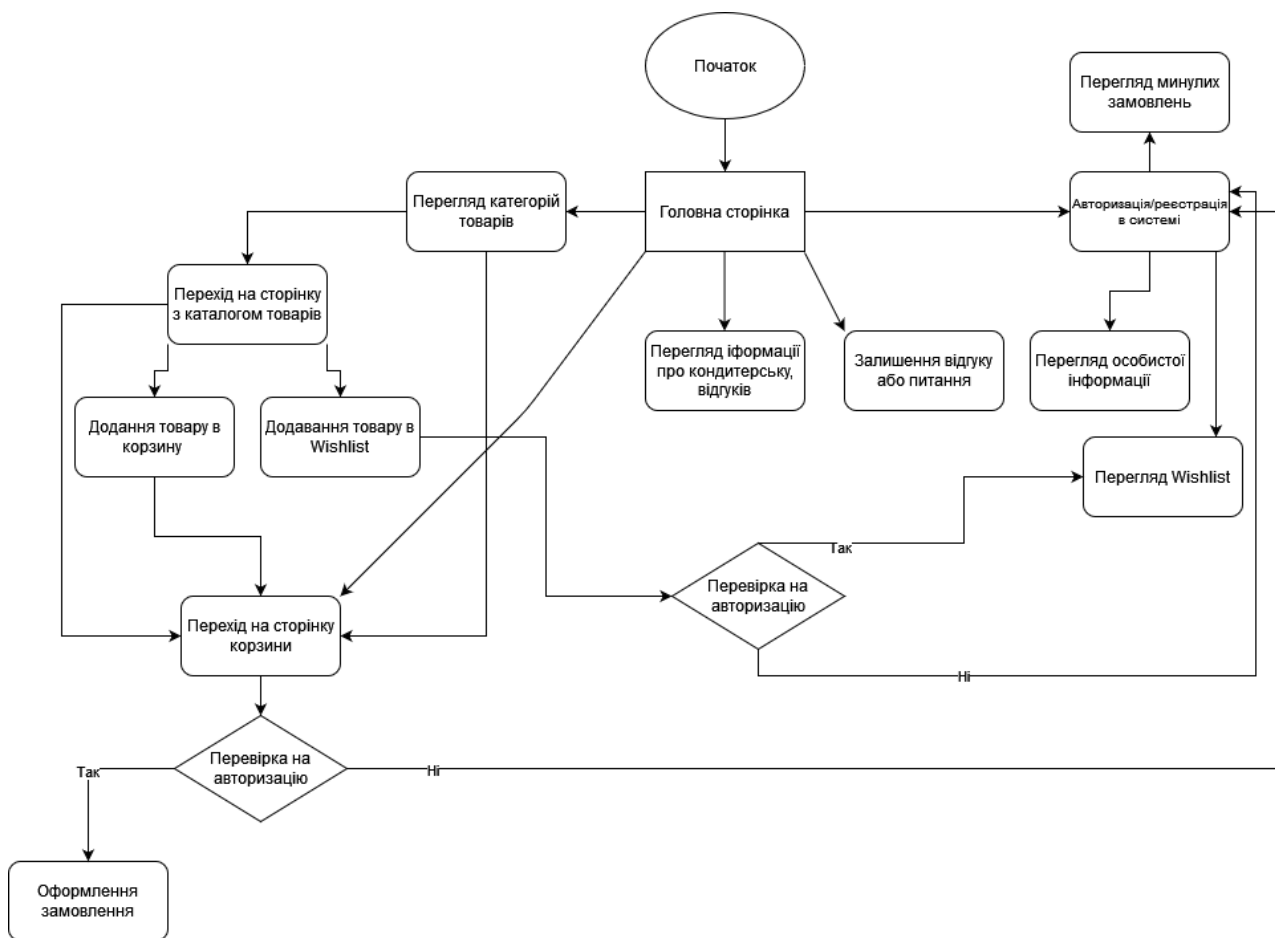


Рисунок Г.1 — UML-діаграма варіантів використання веб-додатку

ДОДАТОК Д

Структурна схема інформаційно-аналітичної системи

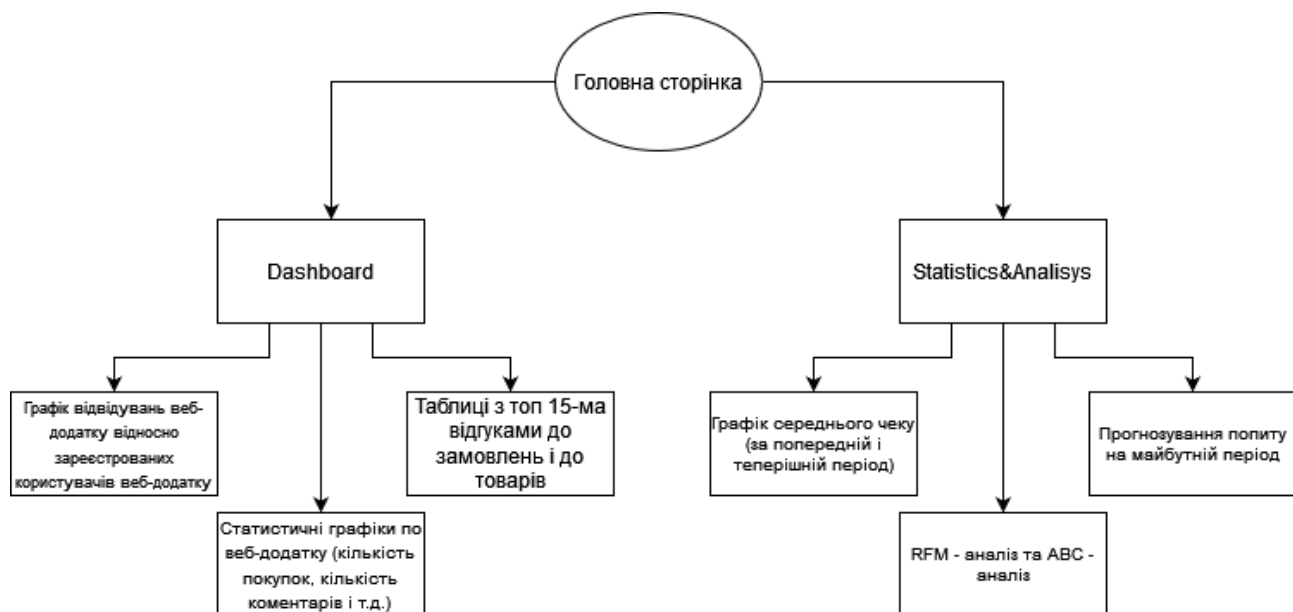


Рисунок Д.1 — Структурна схема інформаційно-аналітичної системи

ДОДАТОК Е

Вигляд інформаційно-аналітичної системи адміністрування веб-додатку

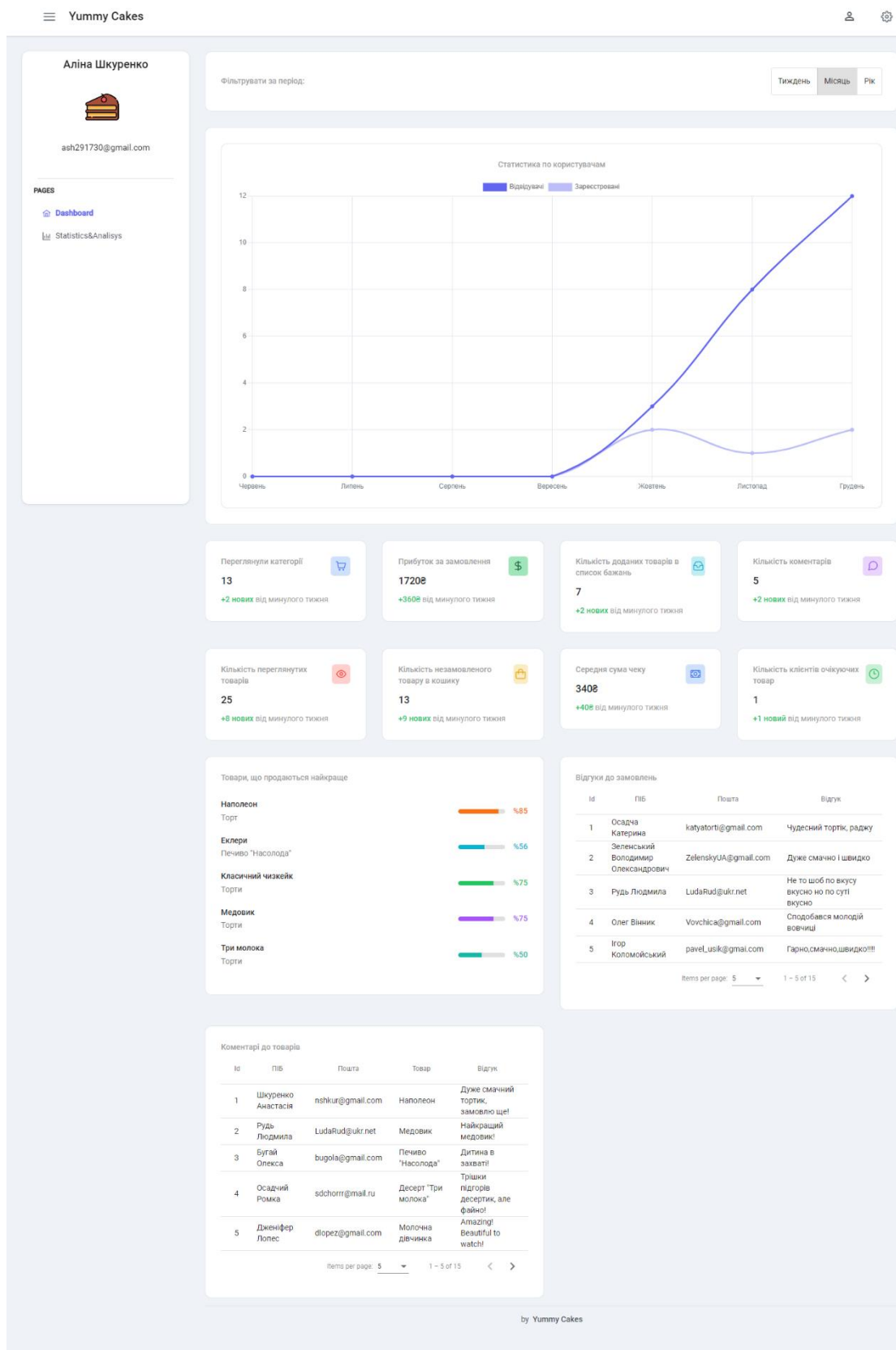


Рисунок Е.1 — Вигляд сторінки «Dashboard»

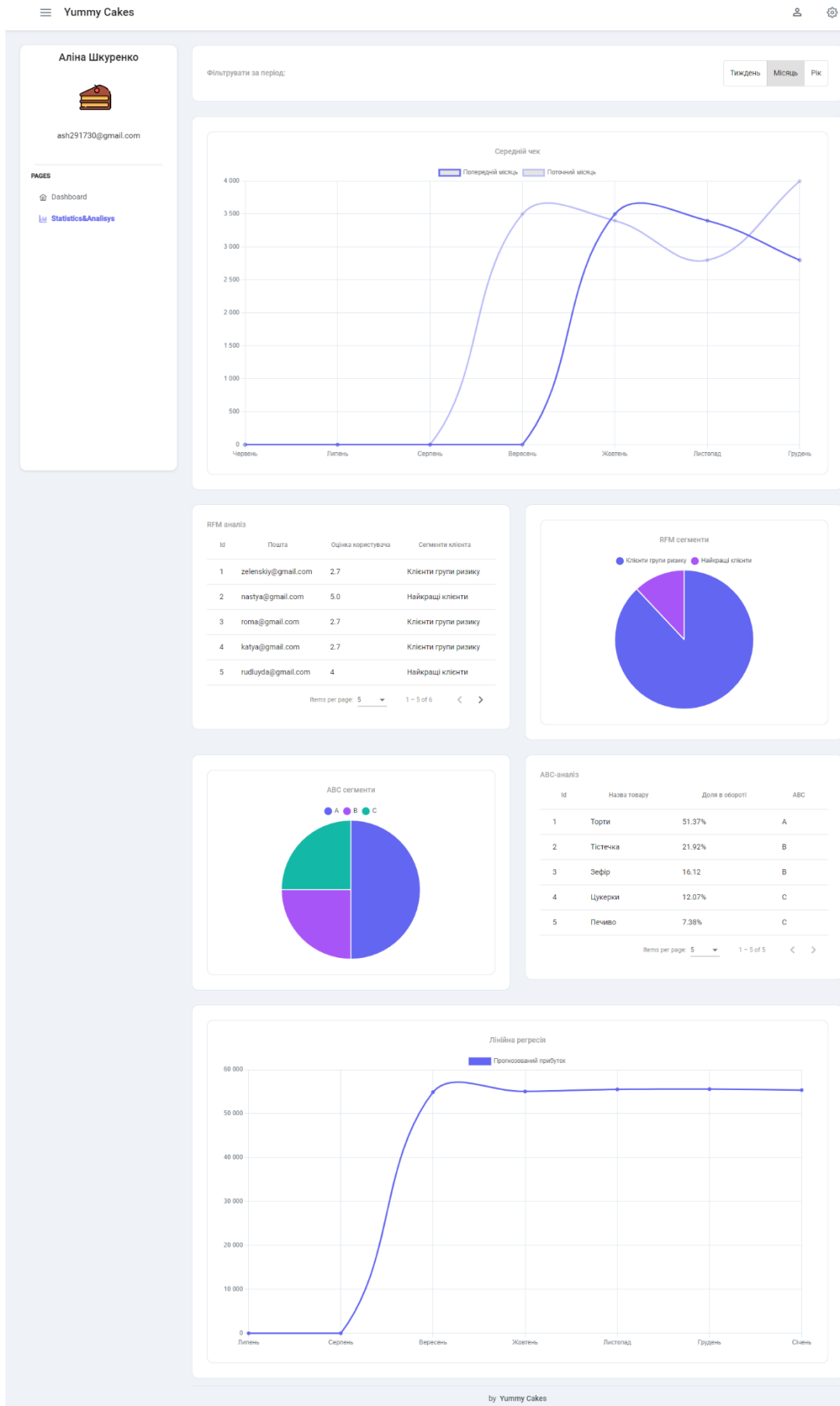


Рисунок Е.2 — Видяг сторінки «Statistics&Analysis»

ДОДАТОК Ж

Схема бази даних

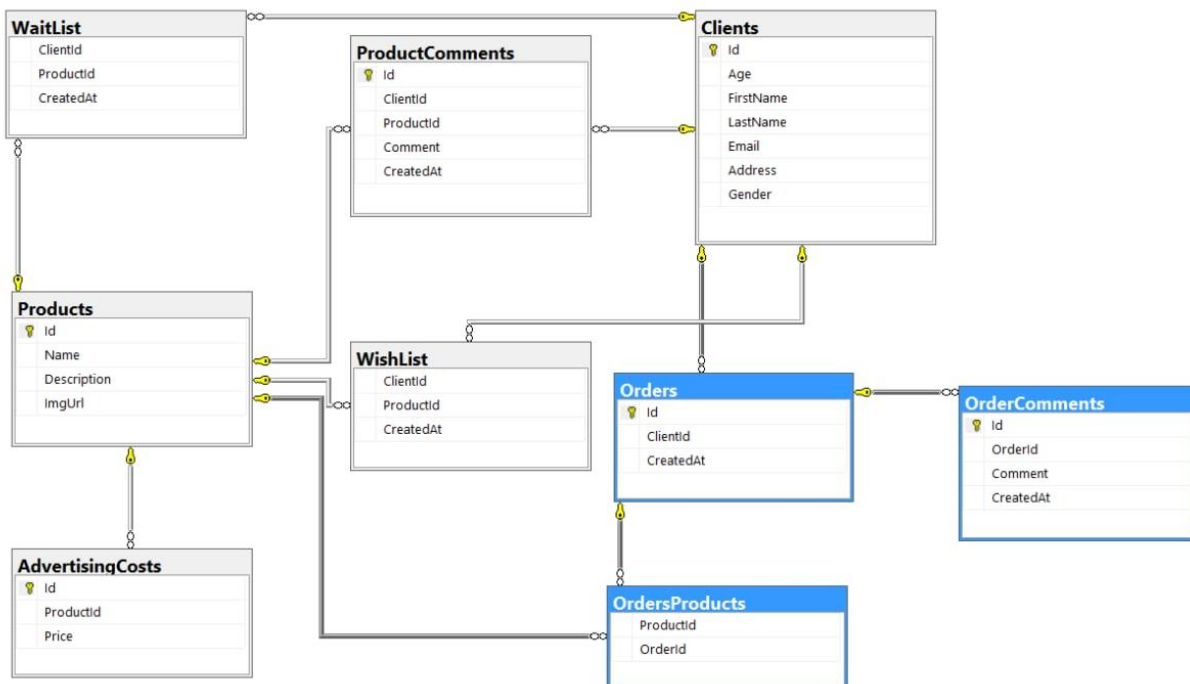


Рисунок Ж.1 — Схема бази даних

ДОДАТОК И

Лістинг коду маршрутизації в інформаційно-аналітичній системі

```
app-routing.module.ts
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AppComponent } from './app.component';
import { AnalisisComponent } from './components/analisis/analisis.component';
import { DashboardComponent } from './components/dashboard/dashboard.component';
import { NotFoundComponent } from './components/notfound/notfound.component';
import { AuthGuard } from './guards/auth.guard';
import { AppLayoutComponent } from './layout/app.layout.component';
const routes: Routes = [
  {
    path: '', component: AppLayoutComponent,
    children: [
      { path: '', component: AppComponent },
      { path: 'dashboard', component: DashboardComponent },
      { path: 'analisis', component: AnalisisComponent },
    ],
    canActivate: [AuthGuard]
  },
  { path: 'auth', loadChildren: () => import('./components/auth/auth.module').then(m
=> m.AuthModule) },
  { path: 'notfound', component: NotFoundComponent },
  { path: '**', redirectTo: '/notfound' },
];
```

```
@NgModule({
  imports: [RouterModule.forRoot(routes, { scrollPositionRestoration: 'enabled',
anchorScrolling: 'enabled', onSameUrlNavigation: 'reload' })],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

ДОДАТОК К

Лістинг коду розмітки сторінок інформаційно-аналітичної системи

Файл app.menu.component.ts

```
import { OnInit } from '@angular/core';
import { Component } from '@angular/core';
import { AuthService } from '../services/auth.service';
import { LayoutService } from '../service/app.layout.service';
@Component({
  selector: 'app-menu',
  templateUrl: './app.menu.component.html'
})
export class AppMenuComponent implements OnInit {
  model: any[] = [];
  login: string;
  constructor(public layoutService: LayoutService, private authService: AuthService)
  {
    this.login = authService.sessionUser.login;
  }
  ngOnInit() {
    this.model = [
      {
        label: 'Pages',
        items: [
          { label: 'Dashboard', icon: 'pi pi-fw pi-home', routerLink: ['/dashboard'] },
          { label: 'Statistics&Analisys', icon: 'pi pi-fw pi-chart-bar', routerLink:
[/analisys'] }
        ]
      },
    ];
  }
};
```

```

    }
}

```

Файл app.menuitem.component.ts

```

import { ChangeDetectorRef, Component, Host, HostBinding, Input, OnDestroy,
OnInit } from '@angular/core';
import { NavigationEnd, Router } from '@angular/router';
import { animate, state, style, transition, trigger } from '@angular/animations';
import { Subscription } from 'rxjs';
import { filter } from 'rxjs/operators';
import { MenuService } from './app.menu.service';
import { LayoutService } from './service/app.layout.service';

@Component({
  // eslint-disable-next-line @angular-eslint/component-selector
  selector: '[app-menuitem]',
  template: `
<ng-container>
  <div *ngIf="root && item.visible !== false" class="layout-menuitem-root-
text">{{ item.label }}</div>
  <a *ngIf="(!item.routerLink || item.items) && item.visible !== false"
[attr.href]="item.url" (click)="itemClick($event)"
  [ngClass]="item.class" [attr.target]="item.target" tabindex="0" pRipple>
    <i [ngClass]="item.icon" class="layout-menuitem-icon"></i>
    <span class="layout-menuitem-text">{{ item.label }}</span>
    <i class="pi pi-fw pi-angle-down layout-submenu-toggler"
*ngIf="item.items"></i>
  </a>
  <a *ngIf="(item.routerLink && !item.items) && item.visible !== false"
(click)="itemClick($event)" [ngClass]="item.class"

```

```

[routerLink]="item.routerLink"                routerLinkActive="active-route"
[routerLinkActiveOptions]="item.routerLinkActiveOptions||{ paths: 'exact',
queryParams: 'ignored', matrixParams: 'ignored', fragment: 'ignored' }"
    [fragment]="item.fragment"
[queryParamsHandling]="item.queryParamsHandling"
[preserveFragment]="item.preserveFragment"
    [skipLocationChange]="item.skipLocationChange"
[replaceUrl]="item.replaceUrl"                [state]="item.state"
[queryParams]="item.queryParams"
    [attr.target]="item.target" tabindex="0" pRipple>
    <i [ngClass]="item.icon" class="layout-menuitem-icon"></i>
    <span class="layout-menuitem-text">{{ item.label }}</span>
    <i class="pi pi-fw pi-angle-down layout-submenu-toggler"
*ngIf="item.items"></i>
    </a>
    <ul *ngIf="item.items && item.visible !== false"
[@children]="submenuAnimation">
    <ng-template ngFor let-child let-i="index" [ngForOf]="item.items">
    <li app-menuitem [item]="child" [index]="i" [parentKey]="key"
[class]="child.badgeClass"></li>
    </ng-template>
    </ul>
</ng-container>
`
,
animations: [
    trigger('children', [
        state('collapsed', style({
            height: '0'
        })),
        state('expanded', style({

```



```

        height: '*'
      })),
      transition('collapsed <=> expanded', animate('400ms cubic-bezier(0.86, 0,
0.07, 1)'))
    ])
  ]
})
export class AppMenuItemComponent implements OnInit, OnDestroy {
  @Input() item: any;
  @Input() index!: number;
  @Input() @HostBinding('class.layout-root-menuitem') root!: boolean;
  @Input() parentKey!: string;
  active = false;
  menuSourceSubscription: Subscription;
  menuResetSubscription: Subscription;
  key: string = "";

```

ДОДАТОК Л

Лістинг коду методів аналізу отриманих даних

```
import { Injectable } from '@angular/core';
import { HttpClient, HttpParams } from '@angular/common/http';

@Injectable()
export class AnalisisAccessor {
  constructor(
    protected httpClient: HttpClient
  ) { }

  public getRFMScoreForClient(clientId: number) {
    const params = new HttpParams()
      .set('clientId', clientId);

    return this.httpClient.get<any>(`https://localhost:4200/api/Analisis/rfmscore`, {
      params });
  }
}

using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Threading.Tasks;
using System.Linq;

namespace YummyCakes.Controllers
{
  [ApiController]
```

```

[Route("api/[Controller]")]
[ApiVersion("1.0")]
public class AnalysisController
{
    public AnalysisController()
    {
    }
    [Route("rfmscore")]
    [HttpGet]
    [ProducesResponseType(StatusCodes.Status200OK)]
    [ProducesResponseType(typeof(string), StatusCodes.Status400BadRequest)]
    [MapToApiVersion("1.0")]
    public async Task<ActionResult<int>> GetRFMScoreForClient([FromQuery]
int clientId)
    {
        var client = _db.Clients.FirstOrDefault(x => x.Id == clientId);
        var orders = _db.Orders.Where(x => x.ClientId == clientId);
        var ordersCount = orders.Count();
        var today = new DateTime();
        var lastOrderDay = orders.OrderByDescending(x =>
x.CreatedAt).FirstOrDefault().CreatedAt;
        var moneySpent = orders.Sum(x => x.Price);
        int result = 0;
        if (lastOrderDay.Month - today.Month > 3 && ordersCount < 3 &&
moneySpent < 500)
        {
            result = 1;
        }
    }
}

```

```

    else if (lastOrderDay.Month - today.Month > 1 && lastOrderDay.Month -
today.Month < 3 && ordersCount < 6 && ordersCount > 3 && moneySpent < 1000
&& moneySpent > 500)
    {
        result = 2;
    }
    else if (lastOrderDay.Week - today.Week > 3 && lastOrderDay.Week -
today.Week < 4 && ordersCount < 12 && ordersCount > 6 && moneySpent < 2000
&& moneySpent > 1000)
    {
        result = 3;
    }
    else if (lastOrderDay.Week - today.Week > 1 && lastOrderDay.Week -
today.Week < 3 && ordersCount < 17 && ordersCount > 12 && moneySpent < 5000
&& moneySpent > 2000)
    {
        result = 4;
    }
    else if (lastOrderDay.Week - today.Week < 1 && ordersCount > 17 &&
moneySpent > 5000)
    {
        result = 5;
    }
    return Ok(result);
}
}
}

public async getABCAnalisysOnPeriod(period: number) {
    const          soldProductsOnPeriod          =          await
this.accessor.getSoldProductsOnPeriod(period);

```

```

const totalIncome = soldProductsOnPeriod.reduce((sum, product) => sum +
product.income, 0);
let result = [];
soldProductsOnPeriod.forEach(product => {
  const abcPartPercentage = product.income * 100 / totalIncome;
  result.push(
    {
      product,
      abcPartPercentage,
      abcRank: abcPartPercentage > 50 ? 'A' : abcPartPercentage > 15 ? 'B' : 'C'
    }
  );
});

public async getLinearRegression(period: number, adSpentForNextMonth: number) {
  const advertisementCostWithSellsOfPeriod = await
this.accessor.getAdvertisementCostOfPeriod(period);

  const b = (period * advertisementCostWithSellsOfPeriod.reduce((sum, entity) =>
entity.adCost * entity.sells, 0) -
  advertisementCostWithSellsOfPeriod.reduce((sum, entity) => sum + entity.adCost,
0) *
  advertisementCostWithSellsOfPeriod.reduce((sum, entity) => sum + entity.sells,
0)) /
  ((period - 1) * advertisementCostWithSellsOfPeriod.reduce((sum, entity) => sum +
entity.adCost, 0) ** 2);

  const xAvarage = advertisementCostWithSellsOfPeriod.reduce((sum, entity) => sum
+ entity.adCost, 0) / period;

  const yAvarage = advertisementCostWithSellsOfPeriod.reduce((sum, entity) => sum
+ entity.sells, 0) / period;

  const a = yAvarage - b * xAvarage;

```

```
return a + b * adSpentForNextMonth;  
}
```

ДОДАТОК М

Лістинг коду сторінок адміністративної панелі

Файл `analysys.component.spec.ts`

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { AnalisysComponent } from './analysys.component';
describe('AnalisysComponent', () => {
  let component: AnalisysComponent;
  let fixture: ComponentFixture<AnalisysComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ AnalisysComponent ]
    })
    .compileComponents();
    fixture = TestBed.createComponent(AnalisysComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

Файл `dashboard.component.spec.ts`

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { DashboardComponent } from './dashboard.component';
describe('DashboardComponent', () => {
  let component: DashboardComponent;
  let fixture: ComponentFixture<DashboardComponent>;
  beforeEach(async () => {
```

```
await TestBed.configureTestingModule({
  declarations: [ DashboardComponent ]
})
.compileComponents();
fixture = TestBed.createComponent(DashboardComponent);
component = fixture.componentInstance;
fixture.detectChanges();
});
it('should create', () => {
  expect(component).toBeTruthy();
});
});
```


ДОДАТОК Н

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційно-аналітична система адміністрування веб-додатку з візуалізацією даних

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра обчислювальної техніки
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 90.1% Схожість 9.9%

Аналіз звіту подібності (відмітити потрібне):

- 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 виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- 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.

Особа, відповідальна за перевірку _____ Захарченко С.М.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Шкуренко А.В.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Войцеховська О. В.
(підпис) (прізвище, ініціали)