

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Бакалаврська дипломна робота

на тему: : «Розробка програмних засобів навчальної веб-системи для вивчення математики»

Виконав: студент 4 курсу
групи 4ПІ-206
спеціальності

121 – Інженерія програмного забезпечення
(шифр і назва напрямку підготовки, спеціальності)

Закасовський Б. І.

(прізвище та ініціали)

Керівник: к.т.н. доц. каф. ПЗ Черноволик Г.О.

(прізвище та ініціали)

Рецензент: к.т.н., доц. каф. ЗІ Маліновський В.І.

(прізвище та ініціали)

Допущено до захисту

Зав. кафедри ПЗ Романюк О. Н.
« 10 » червня 2024 р.

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

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
О. Н. Романюк
« 12 » березня 2024 р.

ЗАВДАННЯ НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Закасовському Богдану Івановичу

1. Тема роботи – Розробка програмних засобів навчальної веб-системи для вивчення математики

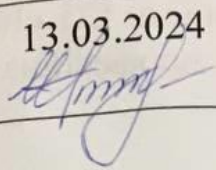
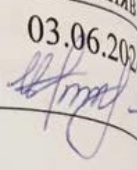
Керівник роботи: Черноволик Галина Олександрівна, доцент кафедри програмного забезпечення, затверджені наказом вищого навчального закладу від 11 березня 2024 р. № 80.

2. Строк подання студентом роботи 3.червня 2024.

3. Вихідні дані до роботи: середовище розробки – Visual Studio Code, мова розробки – HTML, CSS, JavaScript/TypeScript; фреймворк React; операційна система – Windows.

4. Зміст текстової частини: вступ; обґрунтування вибору методу розробки та постановка задач дослідження; розробка моделі та алгоритмів програмного застосунку; розробка програмних модулів застосунку; тестування програми; висновки; література; додатки; графічна частина.

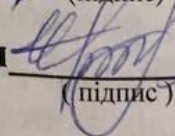
5. Перелік ілюстративного матеріалу: титульний слайд; актуальність теми; мета, об'єкт і предмет дослідження; новизна отриманих результатів; порівняльний аналіз аналогів; розробка алгоритмів роботи системи; тестування програми; висновки.

6. Консультанти розділів роботи		Підпис, дата	
Розділ	Прізвище, ініціали та посада консультанта	Завдання видав	Завдання прийняв
1-4	Черноволик Г.О. к.т.н., доцент. каф. ПЗ	13.03.2024 	03.06.2024 

7. Дата видачі завдання 13.03.2024

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз стану питання та постановка задач дослідження	14.03.2024 – 29.03.2024	вс.
2	Аналіз вхідних та вихідних даних системи	30.03.2024 – 07.04.2024	вс.
3	Розробка моделі та алгоритмів системи	08.04.2024 – 26.04.2024	вс.
4	Розробка інтерфейсу	27.04.2024 – 04.05.2024	вс.
5	Розробка програмних модулів системи	05.05.2024 – 14.05.2024	вс.
6	Тестування системи	15.05.2024 – 20.05.2024	вс.
7	Оформлення матеріалів до захисту БДР	21.05.2024 – 30.05.2024	вс.

Студент ЗКС Б.І. Закасовський
 (підпис) (ініціали та прізвище)
 Керівник бакалаврської дипломної роботи  Г.О. Черноволик
 (підпис) (ініціали та прізвище)

АНОТАЦІЯ

Бакалаврська дипломна робота складається зі 54 сторінок формату А4, на яких є 34 рисунки, 3 таблиці, список використаних джерел містить 25 найменувань.

У бакалаврській дипломній роботі було проведено аналіз сфери навчальних веб-систем для вивчення математики. Проведено аналіз аналогів, визначено їх переваги і недоліки та доведено актуальність розробки власного веб-сервісу.

Розроблена веб-система дозволяє отримувати знання з математики шляхом перегляду теоретичного матеріалу та закріплювати знання завдяки проміжному контролю у вигляді тестів до вивченого матеріалу. Розроблено метод графічного відображення результатів користувача.

Веб-систему було написано на мові програмування JavaScript/TypeScript, каскадних стилях CSS та мові гіпертекстової розмітки HTML. Середовищем розробки було обрано Visual Studio Code, а базою даних платформи Firebase.

В результаті виконання бакалаврської дипломної роботи отримано навчальну веб-систему, що підвищить ефективність процесу вивчення математики та сприятиме зацікавленості учнів вивченням математики.

Отримані в бакалаврській дипломній роботі результати можна використати для покращення процесу вивчення математики.

Ключові слова: веб-системи, вивчення математики, онлайн-навчання.

ABSTRACT

The bachelor's thesis consists of 54 A4 pages, which have 34 figures, 3 tables, the list of sources used contains 25 items.

In the bachelor's thesis, an analysis of the field of educational web systems for the study of mathematics was performed. An analysis of analogues was carried out, their advantages and disadvantages were determined, and the relevance of developing one's own web service was proven.

The developed web system allows to gain knowledge of mathematics by lectures and theoretical material, and consolidate knowledge with the help of intermediate control in the form of tests for the studied material. A method of graphical display of user results has been developed.

As a result of completing the bachelor's thesis, an educational web system was obtained, which increases the effectiveness of the mathematics learning process and the favorable interest of students in studying mathematics.

The results obtained in the bachelor thesis can be used to improve the process of studying physics.

Keywords: web systems, learning mathematics, online learning.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ СТАНУ РОЗВИТКУ НАВЧАЛЬНИХ СИСТЕМ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ	7
1.1 Аналіз стану навчальних веб-систем для вивчення математики	7
1.2 Порівняльний аналіз аналогів.....	8
1.3 Аналіз методів розв’язання задачі	12
1.4 Постановка задач для розробки навчальної веб-системи для вивчення математики	15
1.5 Висновки	15
2 РОЗРОБКА МЕТОДУ, МОДЕЛІ ТА АЛГОРИТМІВ РОБОТИ ВЕБ-СИСТЕМИ.....	16
2.1 Аналіз даних та методів розробки.....	16
2.2 Розробка моделі системи	17
2.3 Розробка методу формування і відображення прогресу навчання	21
2.4 Розробка загального алгоритму роботи додатку	23
2.5 Висновки	25
3 РОЗРОБКА МОДУЛІВ ПРОГРАМНОГО ПРОДУКТУ	26
3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації веб-системи	26
3.2 Розробка модуля керування обліковими записами	30
3.3 Розробка модуля навчальних матеріалів	32
3.4 Розробка модуля проміжного тестування	33
3.5 Розробка модуля формування і відображення прогресу навчання	35
3.6 Розробка структури та інтерфейсу веб-системи	38
3.7 Розробка дизайну інтерфейсу веб-системи	42
3.8 Висновки	43
4 ТЕСТУВАННЯ ПРОГРАМИ.....	44
4.1 Аналіз методів тестування	44
4.2 Тестування онлайн платформи	47

	3
4.3 Розробка інструкції користувача.....	51
4.4 Висновки	52
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
ДОДАТКИ.....	57
Додаток А. Технічне завдання.....	58
Додаток Б. Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	62
Додаток В. Лістинг програми	63
Додаток Г. Ілюстративна частина	81

ВСТУП

Обґрунтування вибору теми дослідження. З розвитком інформаційних технологій сфера освіти зазнала вагомих і революційних змін, авдяки чому різноманітна інформація, знання та навички стали набагато доступнішими. У сучасному світі майже кожна людина має смартфон із доступом в інтернет – всесвітньої системи поєднаних між собою мереж де знаходиться чимала кількість інформаційних ресурсів, послуг, тощо [1, 2].

Одними з таких є освітні веб-ресурси, різноманітність яких може сягати від наукових статей і публікацій до інтерактивних навчальних систем. Онлайн навчання вражає своєю доступністю і зручністю, надаючи навчальні матеріали, віртуальні лабораторні та практичні заняття, завдяки чому учні можуть отримати доступ до цих ресурсів будь-де та в будь-який зручний для них час. Проте здебільшого поширеними є веб-курси, які потребують оплати за навчання, а їхні безплатні аналоги зазвичай дають лише поверхневі ознайомчі знання, які до того ж викладені переважно англійською мовою.

Зараз в світі спостерігається зростання зацікавленості вивченням математики серед студентів та учнів шкіл. Однак, вивчення математики може бути викликом для багатьох учнів через складність та абстрактність науки, а також через перехід на дистанційне навчання, до якого багато закладів освіти виявились не готовими. Через що якість освіти та засвоєння учнями навчального матеріалу дещо погіршилися [3, 4].

Враховуючи широке застосування і значимість математики у розвитку логічного та критичного мислення, розробка навчальної веб-системи має великий потенціал для поліпшення якості навчання, в тому числі й дистанційного. Це сприятиме зростанню рівня математичної освіти серед молоді. Також важливою умовою є забезпечення підтримки української мови навчальною системою, оскільки складні терміни та їх пояснення іноземними мовами можуть бути зрозумілі не всім користувачам.

Тому розробка навчальної веб-системи для вивчення математики є

актуальною задачею на сьогоднішній день.

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

Мета та завдання дослідження. Метою дослідження є підвищення ефективності процесу вивчення математики за рахунок використання розробленої веб-системи, яка дозволяє систематизувати навчальний матеріал, забезпечує проведення тестового контролю, візуалізує ілюстративний матеріал, що підвищує інтерактивність навчального процесу та сприяє зацікавленості учнів вивченням математики.

Задачі дослідження:

- розробити модель навчальної веб-системи для вивчення математики, яка включатиме основний функціонал застосунку;
- програмно реалізувати функціонал веб-системи для вивчення математики;
- розробити зручний інтерфейс користувача, який буде легким у використанні;
- забезпечити підтримку української мови веб-системою та доступність на різних платформах і пристроях, включаючи мобільні пристрої;
- провести тестування розробленої веб-системи.

Об'єкт дослідження – процес розробки навчальної веб-системи для вивчення математики.

Предмет дослідження – методи та засоби реалізації навчальної веб-системи для вивчення математики.

Методи дослідження. У дослідженні було використано методи:

- методи і технології модульної розробки веб-систем для програмної реалізації модулів системи;
- методи захисту даних у Firebase для забезпечення захисту веб-системи та даних користувачів;
- методи UI/UX для розробки та тестування інтерфейсу веб-системи;

– методи тестування для перевірки працездатності роботи програми.

Новизна отриманих результатів:

1. Подальшого розвитку набув метод формування і відображення прогресу навчання, який, на відміну від відомих, надає можливість сформувати звіт з наведеною успішністю у вигляді діаграм, що підвищує інформативність візуалізованих даних та підвищує мотиваційну складову навчального процесу.

2. Подальшого розвитку набув метод інтерактивного вивчення матеріалу, який, на відміну від існуючих, базується на використанні розширеного функціоналу і реалізації візуальних методів графічного подання навчальних матеріалів, підвищує інформативність навчальних матеріалів та сприяє підвищенню ефективності навчального процесу в цілому.

Практична цінність отриманих результатів. Практична цінність одержаних результатів полягає в тому, що розроблена у бакалаврській дипломній роботі веб-система може використовуватись у навчальному процесі для вивчення математики та забезпечує систематизацію наведення освітнього матеріалу, проведення тестового контролю, відстеження успішності та сприятиме зацікавленості учнів вивченням математики.

Особистий внесок здобувача. Усі наукові результати, викладені у бакалаврській дипломній роботі, отримані автором особисто. У друкованій роботі [5], опублікованій у співавторстві, автору належать такі результати: алгоритм проходження навчання й алгоритм роботи персоналізованої навчальної-системи.

Апробація матеріалів бакалаврської дипломної роботи. Матеріали бакалаврської дипломної роботи доповідалися на конференції Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)»

Публікації. Результати дослідження були опубліковані в матеріалах конференції [5].

1. АНАЛІЗ СТАНУ РОЗВИТКУ НАВЧАЛЬНИХ СИСТЕМ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз стану навчальних веб-систем для вивчення математики

На сьогоднішній день аналіз стану навчальних веб-систем показує, що вони стають все більш популярними та затребуваними серед учнів та студентів.

Зараз на ринку існує багато навчальних веб-систем для вивчення математики, що дозволяють ефективніше засвоювати навчальний матеріал та закріплювати знання. Більшість з них містять велику кількість відеоуроків, теоретичних матеріалів, тестів та завдань, які користувач може виконувати для закріплення знань [6].

Деякі з них пропонують інтерактивний підхід до навчання, що дозволяє учням здійснювати експерименти та взаємодіяти з математичними об'єктами через віртуальну реальність. Інші системи базуються на ігровому підході, де користувачі можуть вивчати математику, граючись та розважаючись [7].

Проте більшість існуючих навчальних веб-систем мають певні недоліки, такі як недостатня інтерактивність, низька адаптованість до індивідуальних потреб користувачів та малоефективні засоби закріплення знань.

Тому, створення нової навчальної веб-системи з використанням сучасних веб-технологій та з фокусом на інтерактивному та зручному користуванні може стати вагомим внеском у розвиток освіти та покращення якості вивчення математики.

Отже, можна зробити висновок, що на сьогоднішній день існує потреба в створенні ефективної та доступної веб-системи для вивчення математики.

1.2. Порівняльний аналіз аналогів

При розробці навчальної веб-системи для вивчення математики необхідно враховувати наявність на ринку аналогів, які можуть задовольнити потреби користувачів.

На сьогоднішній день існує чимало аналогів веб-систем вивчення математики. Нижче наведений порівняльний аналіз трьох найбільш популярних аналогів детальні описи кожного з аналогів, а також їх переваги та недоліки:

Matific (рисунок 1.1) – є одним з найбільш відомих та поширених серед навчальних веб-систем для вивчення математики [8]. Він надає користувачам можливість отримувати детальну аналітику, відстежувати роботу учнів за допомогою звітів в реальному часі, виявляти прогалини в навчанні класу та окремих осіб і вживати необхідних дій.

Основними перевагами Matific є:

- Matific забезпечує адаптивний досвід для учнів, учнів заохочують бути відповідальними щодо своєї освіти.
- Matific доступний більш ніж у 50 країнах і більш ніж 40 мовами. Це означає, що ним охоплені особливості навчальних програм більш ніж 50 країн. Кожен учень може навчатися будь-якою мовою, навіть якщо вона відрізняється від тієї, якою проводиться викладання. Завдяки цьому Matific чудово підходить для багатомовних класів, бо існує навіть можливість для кількох учнів в одному класі використовувати різні мови.

Проте Matific не містить навчальних матеріалів, тестів та завдань, які допомагали б закріплювати отримані знання. Крім того, відсутня можливість персоналізації навчального процесу для конкретного користувача.

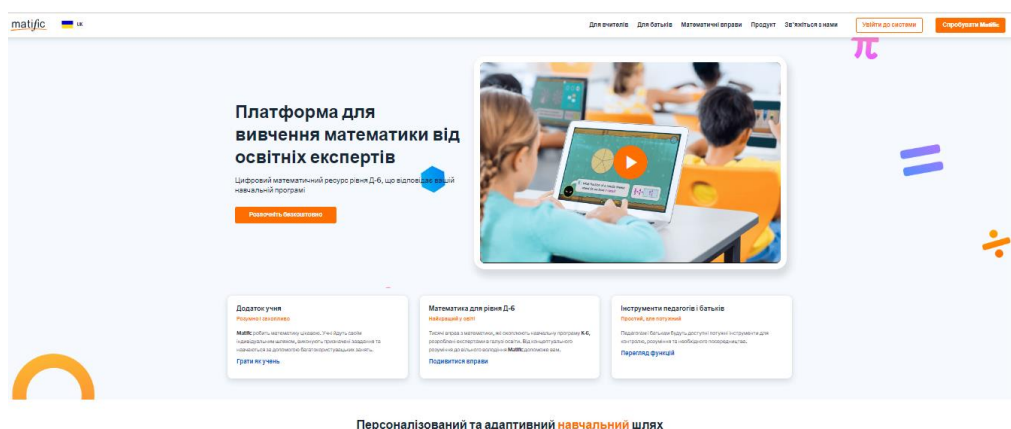


Рисунок 1.1 – Інтерфейс ресурсу Matific

Khan Academy (рисунок 1.2) – одна з найкращих програм для вивчення математики для будь-якого віку. Додаток легко поєднується з інтерактивною панеллю, щоб заохочувати учнів вивчати все, від простих арифметичних дій до складних розрахунків геометрії та алгебри. Його простий дизайн полегшує сприйняття та взаємодію з ним. Немає нічого зайвого, що б відволікало учнів від суті завдань. Має понад 40 000 інтерактивних завдань, які відповідають різним навчальним планам з математики. Додатково зацікавити учнів у навчанні допоможуть математичні відео та статті доступні у програмі. Недоліком є обмежена тематична охопленість.

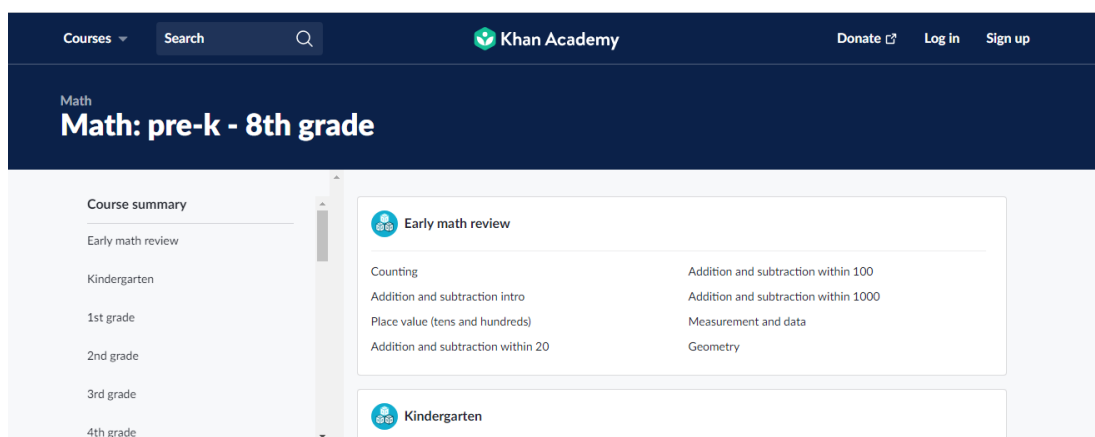


Рисунок 1.2 – Інтерфейс Khan Academy

Mathema (рисунок 1.3) є ще однією популярною навчальною веб-системою для вивчення математики, яка спрямована на старші класи та університетські курси [9]. Система містить велику кількість навчальних матеріалів, включаючи уроки, тести та практичні завдання. Крім того, вона має розширену підтримку вчителів, які можуть створювати свої класи та назначати завдання своїм учням.

Основними перевагами Mathema є:

- безкоштовний доступ до веб-сайту та навчального матеріалу;
- велика кількість матеріалу, розподілена на різні теми з математики;
- наявність тестів та завдань для перевірки знань;
- можливість персоналізації навчального процесу для конкретного користувача.

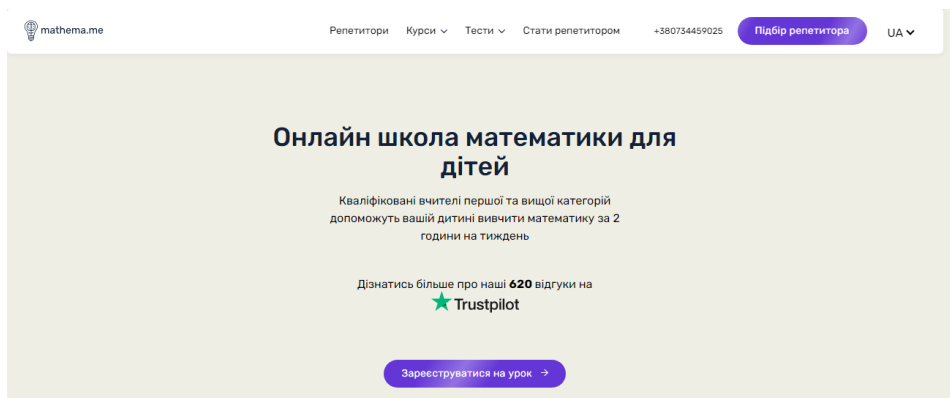


Рисунок 1.3 – Інтерфейс Mathema

OpenStax (рисунок 1.4) є однією з найбільш популярних навчальних веб-систем для вивчення математики, яка надає безкоштовний доступ до великої кількості підручників та матеріалів з різних галузей математики[10].

Це безкоштовна навчальна веб-система для вивчення математики, створена відкритою спільнотою під керівництвом університету Райсу. Вона містить велику кількість безкоштовних підручників та навчальних матеріалів з математики для різних рівнів вивчення.

Серед переваг OpenStax можна відзначити:

- доступна безкоштовно;
- містить велику кількість навчальних матеріалів;
- може бути використана на різних рівнях навчання. Однак, серед недоліків можна виділити:
- відсутність інтерактивних симуляцій та практичних завдань;
- може бути складно для самостійного вивчення без підтримки вчителя.

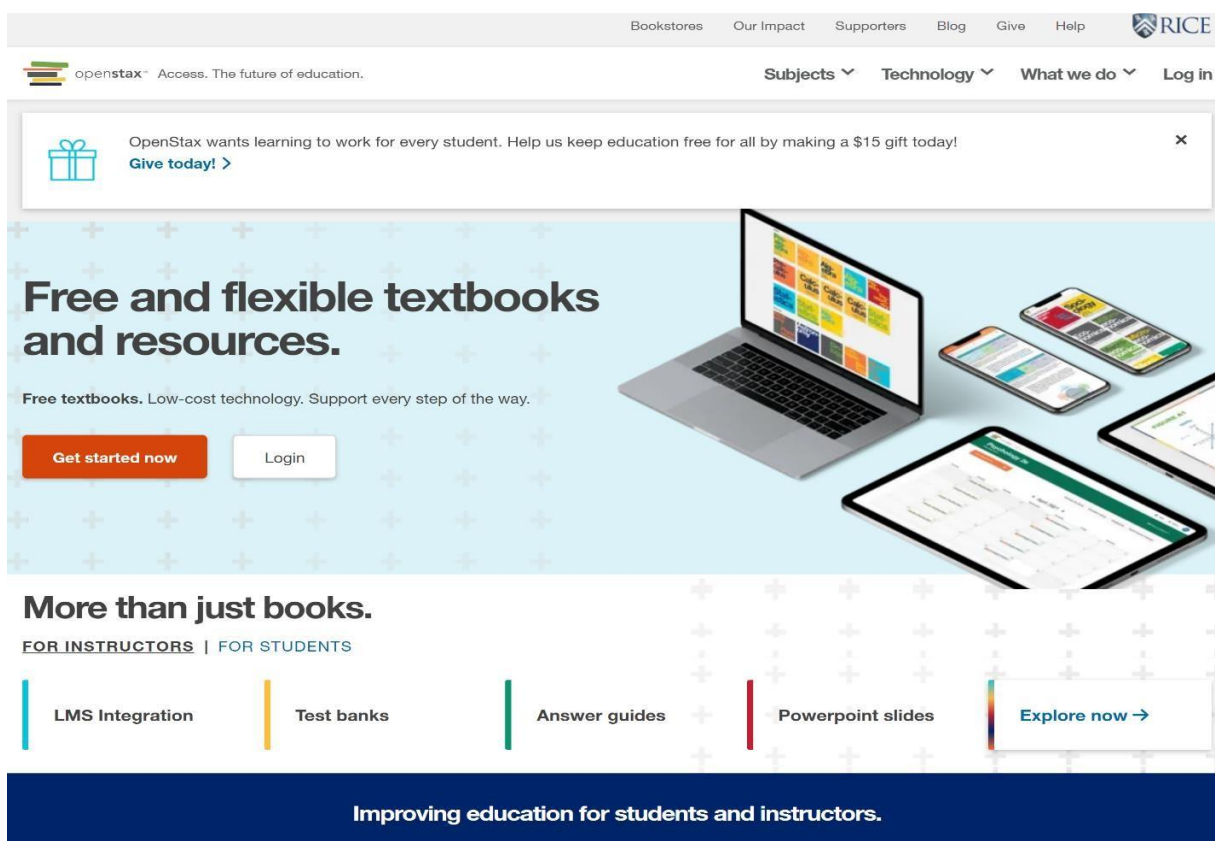


Рисунок 1.4 – Інтерфейс ресурсу OpenStax

Отже, здійснимо порівняльний аналіз даних засобів та розробленої веб-системи (табл.1.1).

Таблиця 1.1 – Порівняльний аналіз аналогів

Характеристики / Функції	Matific	Khan Academy	Mathema	OpenStax	MathExp
Інтерактивність	-	+	—	—	+
Підтримується українська мова	+	—	+	—	+
Тестування	—	—	+	—	+
Безкоштовність	+	+	+	+	+
Сума		2	3	1	4

Отже, внаслідок проведеного аналізу встановлюємо, що розроблений програмний засіб переважає у функціональності розглянуті аналоги.

1.3 Аналіз методів розв'язання задачі

Основним завданням розробки програмних модулів навчальної веб-системи для вивчення математики є забезпечення можливості вивчення математики через веб-сайт, що дозволяє учням вивчати матеріал в будь-якому місці та в будь-який час. Для досягнення цієї мети можна використовувати такі методи.

1. Метод побудови інтерактивних відеоуроків з викладом основних теоретичних матеріалів, демонстрацією прикладів та вирішенням задач [11]. Для цього можна використовувати програмне забезпечення для зняття відео та редагування, наприклад, Camtasia, Adobe Premiere та інші.

2. Метод інтерактивного вивчення, що передбачає використання різних інтерактивних елементів, таких як тести, віртуальні лабораторії, вправи, інтерактивні демонстрації [12]. Для цього можна використовувати спеціалізовані платформи, такі як Moodle, Blackboard, Canvas, або розробляти власний веб-сайт з використанням мов програмування, таких як JavaScript, PHP, HTML, CSS.

3. Метод проєктного навчання, що полягає в розробці учнями власних проєктів, які дозволяють застосувати набуті знання та навички на практиці [13]. Для цього можна використовувати спеціалізовані сервіси, такі як Tinkercad, Scratch, або розробляти власні додатки з використанням мов програмування, таких як Python, Java, C++.

4. Метод гейміфікації, що передбачає побудову освітнього процесу з використанням ігрових елементів для мотивації учнів, забезпечення їх активної участі в навчальному процесі та збереження зацікавленості вивченням матеріалу [14].

5. Метод особистісно-орієнтованого навчання, що передбачає індивідуальний підхід до кожного користувача, з урахуванням його потреб та особливостей [15]. Для цього можна використовувати систему аналізу та обробки даних, що дозволяє підібрати матеріал та завдання, що найбільше відповідають індивідуальним потребам кожного учня.

6. Метод змішаного навчання, що передбачає поєднання традиційного

очного навчання з використанням навчальної веб-системи та онлайн-ресурсів, що застосовуються як додатковий інструмент для доступу до матеріалів, завдань та ресурсів, які мож вивчати у власному темпі в будь-який зручний для них час [16].

Оцінюючи різні методи розробки, було вирішено використовувати в поєднанні декілька методів, а саме метод побудови інтерактивних уроків, метод інтерактивного вивчення та метод особистісно-орієнтованого навчання.

Важливим кроком до проектування навчальної веб-системи є підготовка дидактичних матеріалів. Далі ці матеріали формалізуються і передаються в комплексну базу знань навчального контенту (рівень управління знаннями), що представляється на двох взаємопов'язаних рівнях у відповідних моделях: понятійно-тезисній моделі і ієрархічно-мережевій моделі (Tree-Net) організації навчального контенту. Крім того, експертами готується система професійних компетенцій, і встановлюються зв'язки відповідності між компетенціями і елементами бази знань навчального контенту. Користувач взаємодіє з освітньою системою і за допомогою системи професійних компетенцій формується запит на формування освітнього процесу.

Особливу роль відіграють моделі рівня організації навчання, які «вміють» використовувати закладені в систему дидактично-орієнтовані знання предметної області і описання професій для формування адекватного навчального процесу. Навчальний процес відбувається за допомогою інтелектуального навчаючого середовища на основі гіпертексту і гіпермедіа, що автоматично генеруються у процесі взаємодії користувача з системою. У процесі навчання і проходження контролю знань система акумулює знання про учня і адаптується до його поточних потреб за допомогою усіх складових бази знань. Організаційні і адміністративні аспекти, а також форма навчання налаштовуються у відповідній підсистемі. Соціальна складова навчального процесу підтримується за допомогою організації освітньої Інтернет спільноти на основі єдиної системи взаємодії учасників навчального процесу. Архітектуру програмного забезпечення навчальної веб-системи безперервного навчання, що ґрунтується на такій комплексній моделі зображено на рисунку 1.5.

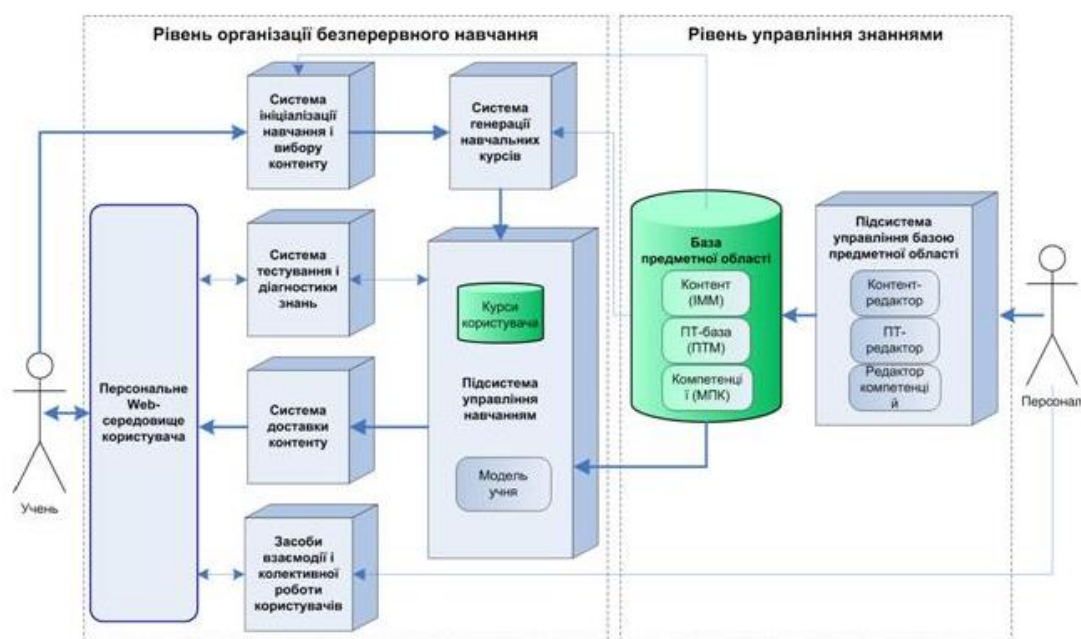


Рисунок 1.4. Архітектура навчальної веб-системи

База знань предметної області складається з трьох інформаційних взаємопов'язаних сховищ: база контенту на основі Tree-Net, понятійно-тезисна база і компетенції. Відповідно персоналу системи надаються спеціальні інструментальні засоби формування і актуалізації такої бази знань. З іншого боку студент починає свою роботу з системи ініціалізації навчання і вибору цільового контенту, які функціонує на основі знань закладених в базі знань предметної області. За результатами цього етапу підсистема управління навчанням генерує персональні навчальні курси студента і зберігає їх за допомогою спеціальних структур.

Система доставки контенту служить для постачання персонального освітнього контенту курсів студенту. Система тестування і діагностики відповідає за механізми контролю знань. Між учасниками освітнього процесу відбувається взаємодія завдяки засобам колективної роботи.

1.4 Постановка задач для розробки навчальної веб-системи для вивчення математики

Здійснивши аналіз особливостей розробки навчальної веб-системи для вивчення математики, було визначено набір задач для розробки модулів програми:

- розробити модель навчальної веб-системи для вивчення математики, яка включатиме основні розділи математики, тести, завдання та інші корисні матеріали;
- розробити функціонал для розміщення різноманітних навчальних матеріалів, оцінювання знань учнів, зокрема, можливість проходження тестів розв’язування завдань та графічного відображення прогресу;
- розробити інтерфейс користувача, який буде зручним та легким у використанні;
- забезпечити підтримку веб-системи на різних платформах та пристроях, включаючи мобільні пристрої;
- провести тестування розробленої веб-системи.

1.5 Висновки

У першому розділі було проаналізовано стан сучасних веб-системи для вивчення математики. Здійснено аналіз існуючих аналогів, а саме: Matific, Khan Academy, Mathema, OpenStax. Визначено переваги та недоліки в порівнянні з навчальною веб-системою для вивчення математики.

Було встановлено, що розроблений програмний засіб переважає у функціональності розглянуті аналоги. Здійснено постановку задач та обрано такі методи для моделювання та інтегрування навчального матеріалу:

- метод інтерактивних уроків;
- метод інтерактивного вивчення.

2 РОЗРОБКА МЕТОДУ, МОДЕЛІ ТА АЛГОРИТМІВ РОБОТИ ВЕБ- СИСТЕМИ

2.1 Аналіз даних та методів розробки

Для досягнення успішного функціонування системи, необхідно вивчити та обробити вхідні дані, що включають текстовий матеріал, задачі, тести. Початковий етап аналізу даних полягає в ознайомленні з вхідними даними, їх форматами та структурою.

Вхідні дані:

- текстовий матеріал з ілюстративними прикладами, що буде використовуватися як структурований конспект;
- списки теоретичних питань, практичних завдань та відповідей до них, що будуть використовуватись для проведення тестувань;
- посилання на додаткову літературу або сервіси.

Далі проводиться попередня обробка вхідних даних, включаючи перевірку на наявність помилок, відсутніх або пошкоджених файлів, та валідацію форматів. Також текстовий матеріал та тести мають бути перевірені на правильність.

Ще одним важливим етапом аналізу даних є визначення потреб користувачів та встановлення вимог до функціональності системи. Це включає опитування та дослідження цільової аудиторії, визначення їхніх очікувань та вимог до системи, а також встановлення пріоритетів функцій, які мають бути реалізовані у системі для вивчення математики.

На підставі проведеного аналізу даних, можна розробити концептуальну модель веб-системи для вивчення математики. Концептуальна модель може включати схематичне відображення компонентів системи, їхніх взаємодій та взаємозв'язків. Вона може бути представлена у вигляді діаграми, схеми або іншого візуального зображення, що допомагає зрозуміти структуру та організацію системи.

Для розробки ефективної навчальної веб-системи для вивчення математики було вирішено використовувати такі методи:

1. Метод інтерактивних уроків: цей метод дозволяє використовувати уроки, що створені викладачем для веб-системи та містять у собі інтерактивні елементи, такі як вправи для самоперевірки, зображення поясненнями, тестові завдання тощо. Цей метод є ефективним, оскільки дає можливість учням вивчати матеріал в зручному для них темпі, повертатися до певних моментів уроку, які потрібно повторити, та самостійно перевіряти свої знання.

2. Метод інтерактивного вивчення: це один з найефективніших підходів до навчання, який передбачає активну взаємодію між учнями та навчальним матеріалом. Цей метод дозволяє учням бути активними учасниками навчального процесу, збільшує їхню мотивацію та сприяє більш ефективному засвоєнню навчального матеріалу. Для використання методу інтерактивного вивчення в навчальній веб-системі для вивчення математики можна використовувати різні інтерактивні елементи. Одним з таких елементів є тести, які можна створювати з різними рівнями складності та обсягом матеріалу. Тести дозволяють учням перевірити свої знання та закріпити матеріал, який вони вивчали.

2.2 Розробка моделі системи

Для кращого розуміння роботи онлайн-платформи для вивчення математики було вирішено розробити модель роботи системи на прикладі UML-діаграм.

На рисунку 2.1 подано діаграму діяльності, що демонструє дії, які користувач може та має здійснити на веб-сайті для успішного вивчення матеріалу та містить блоки дій користувача й взаємодію з елементами веб-інтерфейсу (кнопки, форми, посилання).

Процес навчання має лінійну структуру, тому у початковому стані користувач переходить до веб-сторінки з каталогом уроків, де йому доступний тільки перший урок.

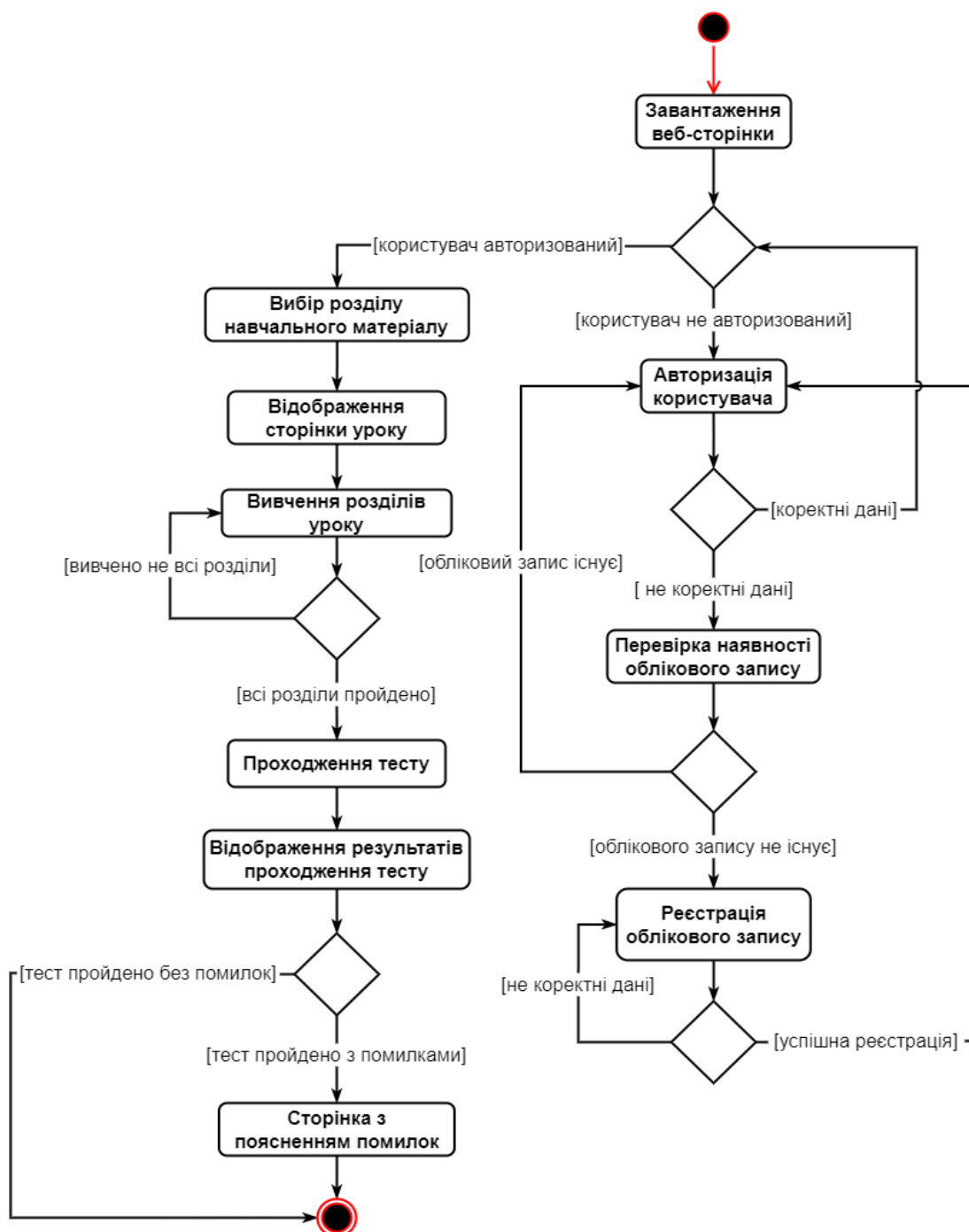


Рисунок 2.1 – Діаграма діяльності

Після натискання на іконку або назву уроку, система перевіряє стан авторизації користувача, оскільки це є обов'язковою умовою для збереження прогресу. Якщо користувач авторизований, то система відображає сторінку з обраним уроком, в іншому випадку – пропонує авторизуватись. У випадку, коли обліковий запис відсутній, система пропонує зареєструватись.

Сторінка уроку містить наступні секції:

1. Змістовна: містить розділи уроку, що відносяться до певної теми, а також посилання на тест, скласти який надається можливість після вивчення усіх розділів уроку.

2. Інформативна: містить навчальний матеріал, який може складатись з теоретичних конспектів, ілюстрацій, таблиць та практичних завдань.

Після освоєння матеріалів уроку, користувач може приступити до проходження тесту, натиснувши на кнопку «Далі» в останньому розділі уроку або натиснувши на посилання у змістовній секції. Система відображає тест, який містить теоретичні питання з варіантами відповідей, а також може містити питання з відкритою відповіддю, наприклад, задачею, яку користувачу потрібно розв'язати та ввести відповідь з клавіатури.

У разі проходження тесту без помилок, користувач завершує вивчення обраної теми, інакше йому показуються його помилки та наводиться пояснення правильного варіанту відповіді. Після цього, користувач знову опиняється на сторінці з каталогом уроків, де йому надається доступ до наступного уроку, а на вивченому уроці з'являється відповідна позначка.

Структура класів веб-системи передбачає наступну взаємодію:

Клас User має зв'язок один до багатьох з класами Lesson, Test та Report оскільки один користувач може вивчати багато уроків, проходити багато тестів та сформувати багато звітів.

Клас Lesson має зв'язок один до одного з класом Test, оскільки для одного уроку математики може бути тільки один контрольний тест.

Клас Test має зв'язок один до багатьох з класом Results, оскільки для одного тесту може бути багато результатів.

Для кращого розуміння описаної взаємодії було розроблено діаграму класів наведену на рисунку 2.2.

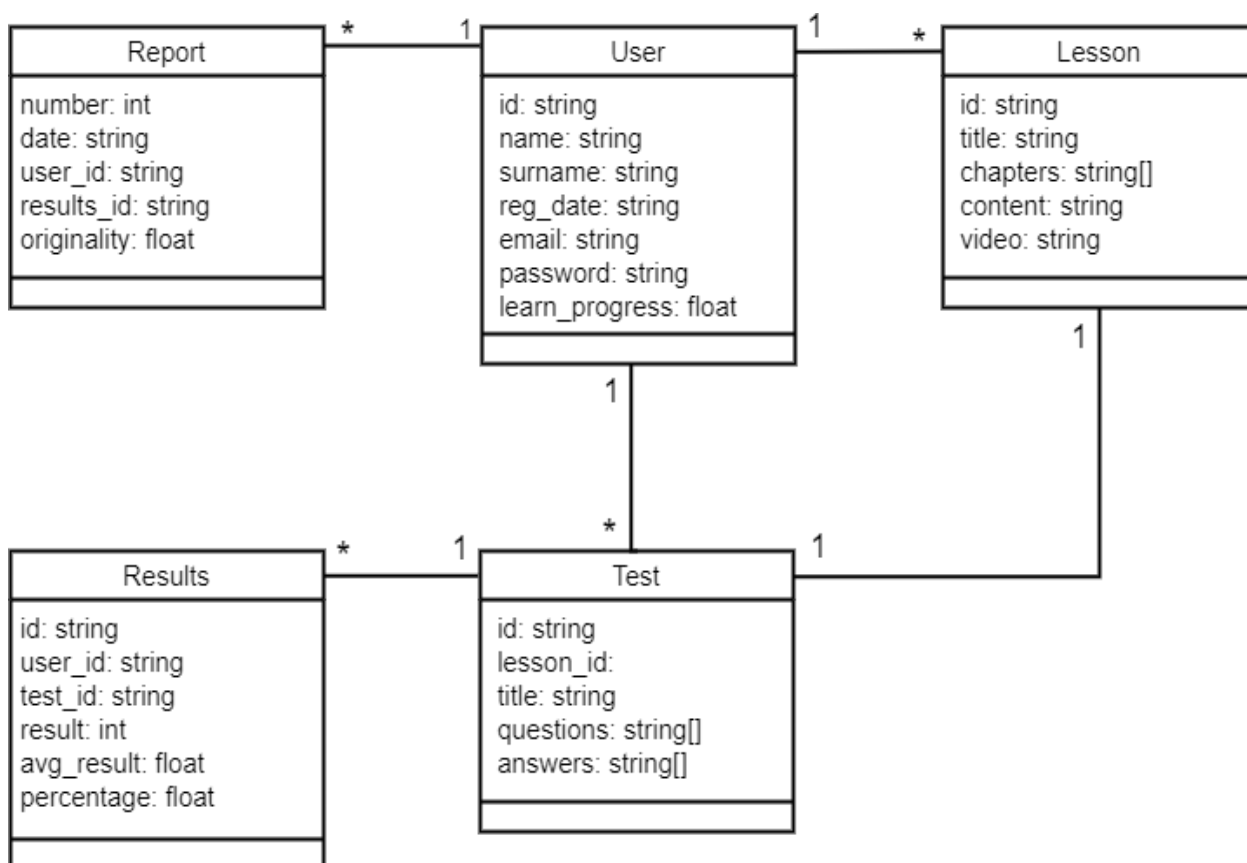


Рисунок 2.2 – Модель роботи системи з використанням діаграми класів

Клас User має поля: id, name, surname, reg_date, email, password та learn_progress, які зберігають інформацію про користувача системи.

Клас Lesson: має поля id, title, chapters, content, video, які зберігають інформацію про урок математики, який може бути вивчений.

Клас Test має поля: id, title, result, avg_result, user_id (зовнішній ключ, що посиляється на поле id класу User) та lesson_id (зовнішній ключ, що посиляється на поле id класу Lesson) і зберігає інформацію про тест, який може пройти користувач з кожного уроку математики.

Клас Results має поля: result, avg_result, percentage, user_id (зовнішній ключ, що посиляється на поле id класу User) та test_id (зовнішній ключ, що посиляється на поле id класу Test) і зберігає результати тестів, які пройшов користувач з кожного уроку математики.

Клас Report має поля: number, date, originality, user_id (зовнішній ключ, що посиляється на поле id класу User) та results_id (зовнішній ключ, що

посилається на поле id класу Results) та отримує дані необхідні для формування файлу-звіту.

2.3 Розробка методу формування і відображення прогресу навчання

У будь-який момент навчання, до повного завершення проходження навчального курсу, користувач має можливість за бажанням або необхідністю сформувати файл-звіт, що дозволяє детально відслідковувати навчальний прогрес.

Метод формування і відображення прогресу навчання дозволяє згрупувати результати користувача та отримати їх у порівнянні з результатами інших у зручному графічному вигляді.

Метод включає такий функціонал:

1. Авторизований користувач переходить на сторінку з уроками.
2. Користувач натискає кнопку «Сформувати звіт» у секції профілю для ініціювання процесу формування звіту.
3. При натисканні кнопки, запускається функція, яка відповідає за отримання даних з Firebase Firestore та їх підготовку для подальшого використання.
4. Метод `getDataFromFirestore` виконує з'єднання з базою даних Firebase Firestore.
5. За допомогою Firebase SDK до бази даних виконується запит для отримання `user_id`, який використовується як критерій для додаткових запитів до бази даних.
6. За протоколом HTTP з бази даних отримується `user_id`.
 - 6.1. За критерієм `user_id` здійснюються запити до бази даних з метою отримання інформації, яка необхідна для заповнення решти полей звіту.
 - 6.2. Відповідно до запитів, з бази даних отримуються дані у форматі JSON, які записуються у змінну `user_data`.
 - 6.3. Метод `getDataFromFirestore` повертає об'єкт, що містить усю інформацію, яку запитувала веб-система, зокрема: прізвище та ім'я користувача,

номери тестів та результати користувача, середній результат користувачів по окремих тестах.

7. Отримані дані передаються в метод `formatDataForChart`, який форматує дані для подальшого використання та повертає підготовлений масив `formattedData`.

7.1. Після отримання підготовленого масиву даних метод заповнює поля ідентифікації користувача: `surname` та `name`.

7.2. Після заповнення полів ідентифікації метод встановлює решту даних масиву у стан компонента `data`, який далі буде використано для графічного представлення даних.

7.2. Сформатовані дані передаються в компонент фреймворку `ReactJS`, який використовує бібліотеки `react-charts` та `react-chartjs-2`.

7.3. За допомогою бібліотек генеруються діаграми та гістограми, що відображають прогрес навчання користувача.

8. Після цього завершується формування звіту:

8.1. Отримується актуальна дата і встановлюється у поле звіту «Дата формування».

8.2. Встановлюється значення поля «`report_number`» типу `int`.

9. Користувач переадресовується на сторінку із сформованим файлом, де може його переглянути та/або скачати.

Описаний метод містить кроки авторизації користувача, отримання даних з `Firebase Firestore`, форматування цих даних для графічного представлення, завершення формування звіту і відображення результату користувачу. Алгоритм роботи цього методу наведено на рисунку 2.3.

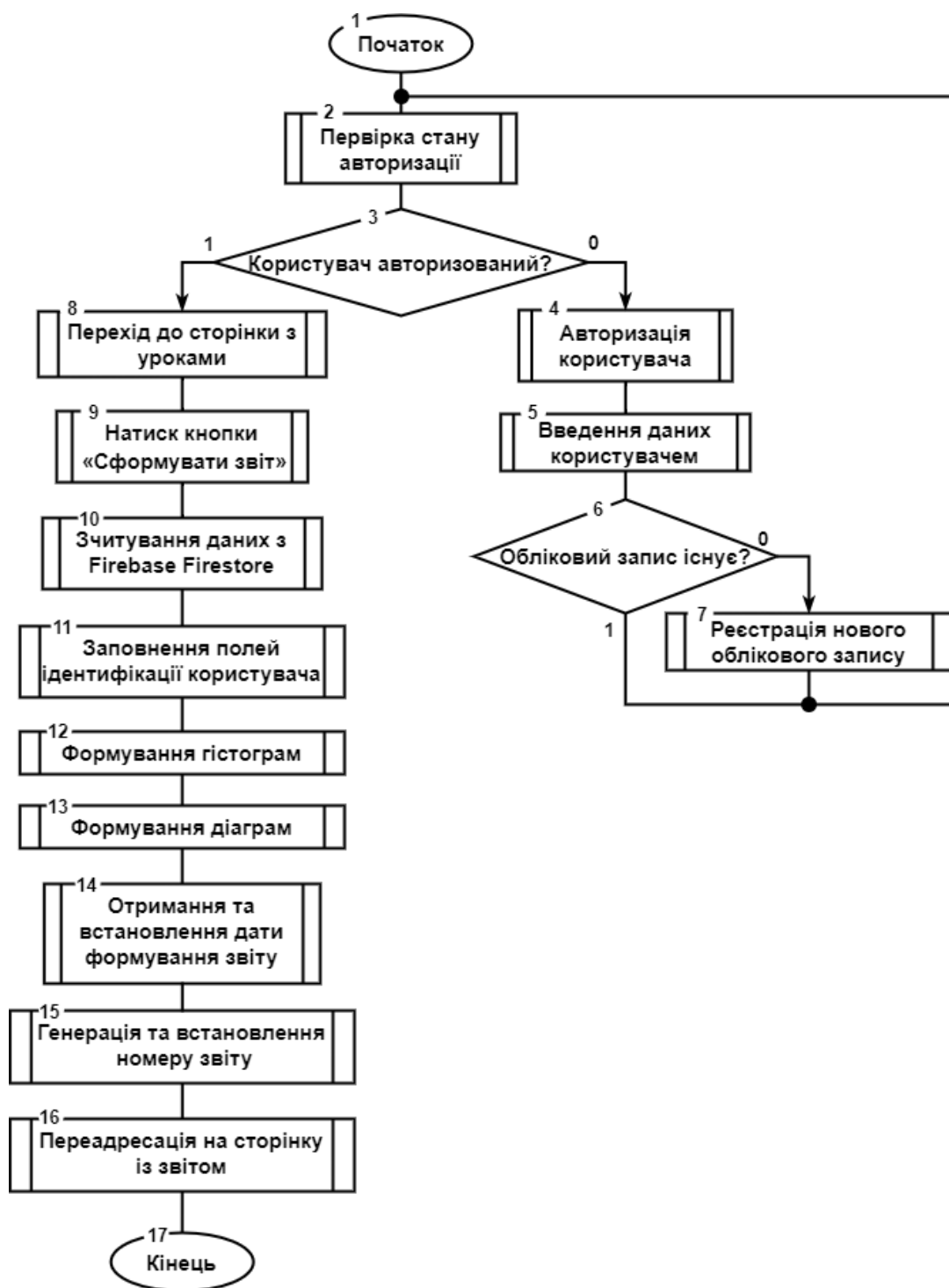


Рисунок 2.3 – Блок-схема алгоритму роботи методу формування і відображення прогресу навчання

2.4 Розробка загального алгоритму роботи додатку

Для розробки навчальної веб-системи для вивчення математики необхідно розробити загальний алгоритм, згідно з яким користувач відкриває веб-сайт для вивчення математики в браузері або на мобільному пристрої, після чого

авторизується або реєструється.

Авторизований користувач обирає урок, який бажає вивчити зі списку доступних варіантів.

Після вибору уроку користувач може вивчити матеріал, перейшовши до відповідної сторінки з уроком, де він має змогу, прочитати теоретичний конспект. Далі користувач проходить тестування: у випадку успішного проходження тестування, надається доступ до наступних тем, або сповіщення про необхідність перепройти тестування. Загальний алгоритм роботи веб-системи наведено на рисунку 2.4.

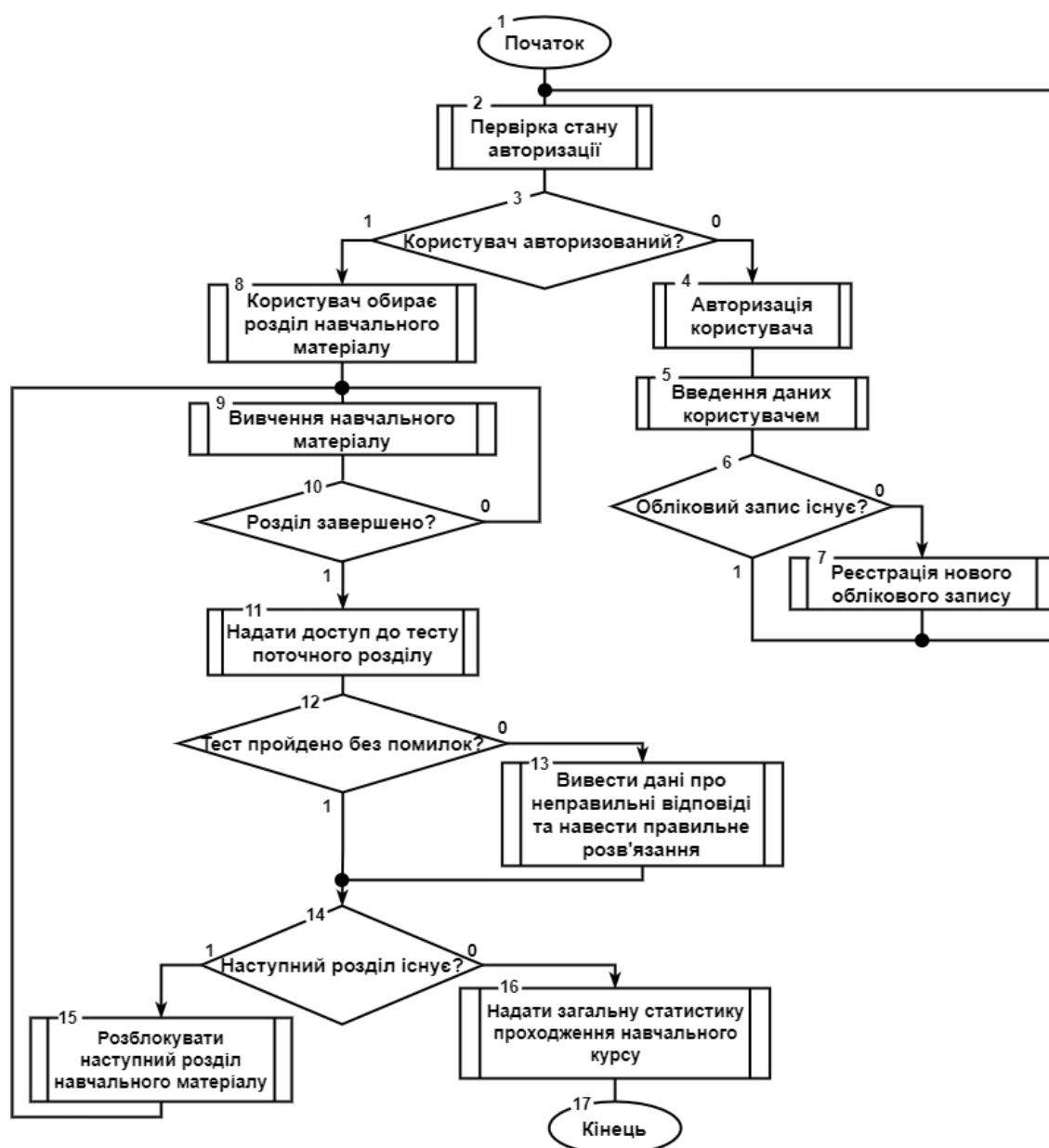


Рисунок 2.4 – Блок-схема загального алгоритму роботи веб-системи

2.5 Висновки

У даному розділі було проаналізовано особливості вхідних даних навчальної веб-системи для вивчення математики:

- текстовий матеріал з ілюстративними прикладами, що буде використовуватися як структурований конспект;
- списки теоретичних питань, практичних завдань та відповідей до них, що будуть використовуватись для проведення тестувань;
- посилання на додаткову літературу або сервіси.

На прикладі UML-діаграм класів та діаграми діяльності розроблено модель роботи системи, яка описує основний функціонал онлайн-платформи та взаємодію між основними класами. Також розроблено метод формування і відображення прогресу навчання. Наведено блок-схеми алгоритму розробленого методу та загального алгоритму роботи веб-системи.

3. РОЗРОБКА МОДУЛІВ ПРОГРАМНОГО ПРОДУКТУ

3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації веб-системи

Ефективність розробки запланованої веб-системи залежить від вибору мови програмування та середовища розробки. Можливими мовами програмування для реалізації системи є Python, JavaScript та PHP.

PHP [17] – це мова програмування, яка використовується для розробки динамічних веб-сторінок та веб-додатків.

Переваги:

- легкість вивчення та використання;
- розширюваність;
- висока швидкість виконання.

Недоліки:

- проблеми з масштабуванням;
- залежність від бази даних;
- перевантаження сервера;

JavaScript [18] – це мова програмування, яка використовується для створення веб-додатків та розширення функціональності веб-сторінок.

Переваги:

- широке застосування;
- легкість вивчення та використання;
- велика кількість бібліотек та фреймворків;
- швидкість виконання. Недоліки:
- залежність від браузера;
- складність масштабування.

Python [19] – це високорівнева, інтерпретована мова програмування, яка здатна виконувати широкий спектр завдань, від простих скриптів до веб-розробки та наукових досліджень.

Переваги:

- простота та легкість вивчення;
- швидкість розробки;
- велика кількість сторонніх бібліотек;
- крос-платформеність. Недоліки:
- швидкість виконання;
- обмеження масштабування;
- проблеми з безпекою.

JavaScript є найкращим вибором для розробки веб-системи для вивчення математики з наступних причин:

Клієнтська валідація: JavaScript можна використовувати для валідації даних введених користувачем на стороні клієнта. Це забезпечує більш швидку та ефективну роботу з валідацією даних.

Реактивність: JavaScript дозволяє реалізувати реактивну взаємодію між елементами веб-сторінки, що забезпечує більш зручне та ефективне використання веб-системи для вивчення математики.

TypeScript: TypeScript є розширенням JavaScript, яке додає до мови сильну типізацію та інші функціональні можливості. Це дозволяє знизити кількість помилок під час розробки та полегшити підтримку коду в майбутньому.

JSX: JSX є розширенням синтаксису JavaScript, яке дозволяє використовувати HTML-подібний код для створення веб-елементів. Це забезпечує більш зручну та ефективну розробку веб-інтерфейсу для веб-системи для вивчення математики.

ReactJS: один з найпопулярніших фреймворків JavaScript, який використовується для розробки інтерактивних веб-інтерфейсів. Він дозволяє легко створювати інтерфейс у вигляді невеликих компонентів які можна перевикористовувати та модифікувати. А також забезпечує миттєву реакцію на дії користувача або зміни стану, ефективну роботу з DOM-деревом та знижує навантаження на сервер, що є дуже корисним при розробці інтерактивних систем [20].

Використання JavaScript для розробки веб-системи для вивчення

математики, а також його розширення TypeScript та JSX і фреймворк ReactJS дозволять забезпечити більш швидку та ефективну розробку веб-інтерфейсу, покращити безпеку та забезпечити більш швидку відповідь на дії користувачів.

Отже, як основну мову було обрано JavaScript, тому що це дозволить забезпечити швидку та ефективну розробку, покращити безпеку, забезпечити більш зручний інтерфейс та надійну і стабільну роботу системи.

Розглянемо деякі з популярних середовищ розробки, такі як Visual Studio Code, WebStorm та Atom.

Visual Studio Code [21] – це безкоштовне кросплатформне середовище розробки, розроблене компанією Microsoft. Воно підтримує ряд мов програмування, включаючи JavaScript, TypeScript, Python, Java, інші. В Visual Studio Code доступна підтримка розширень, що дозволяє розширювати його функціональність в залежності від потреб розробника. Він також має вбудовану підтримку Git, що дозволяє легко працювати з контролем версій.

Visual Studio Code має зручний та простий інтерфейс, що робить його зручним для використання як для початківців, так і для досвідчених розробників. Воно також підтримує розширення для роботи з різними фреймворками та бібліотеками, що дозволяє забезпечити більш ефективний та швидкий процес розробки.

Крім того, Visual Studio Code є легким та швидким, що робить його популярним серед розробників, які вимагають високої продуктивності та ефективності.

WebStorm [22] – це інтегроване середовище розробки для веб-розробки від компанії JetBrains. Воно підтримує більшість популярних мов програмування для веб-розробки, таких як HTML, CSS, JavaScript, TypeScript, а також ряд фреймворків та бібліотек, включаючи React, Angular, Vue та інші. WebStorm має велику кількість корисних функцій та інструментів, які полегшують та прискорюють роботу розробників. Воно має вбудований дебагер, підтримку автодоповнення коду, попередній перегляд, підтримку мініфікації та компіляції коду та інше. WebStorm також має дружній та простий інтерфейс, що

дозволяє розробникам працювати ефективніше та продуктивніше.

Крім того, WebStorm забезпечує різні інструменти для збірки, тестування та розгортання веб-додатків, що дозволяє розробникам зосередитися на розробці коду та покращенні якості веб-додатків. WebStorm є платним продуктом, але доступна пробна версія на 30 днів.

Atom [23] – це безкоштовний текстовий редактор та інтегроване середовище розробки, розроблене компанією GitHub. Atom був розроблений для підтримки різних мов програмування, включаючи HTML, CSS, JavaScript, PHP, Python, Ruby та інші.

Однією з головних переваг Atom є його відкритість та можливість налаштування, що дозволяє розробникам налаштовувати редактор згідно своїх потреб. Atom також має велику кількість додатків та плагінів, які дозволяють додавати різноманітні функції та інструменти.

У Atom також є ряд корисних функцій, таких як автодоповнення, відступи та вирівнювання коду, попередній перегляд, вбудований дебагер, можливість роботи з репозиторіями Git та інші. Однак, порівняно з іншими інтегрованими середовищами розробки, Atom може працювати повільніше на менш потужних комп'ютерах та не має вбудованої підтримки TypeScript та JSX.

Тому середовищем для розробки веб-системи для вивчення математики було обрано Visual Studio Code з кількох причин. По-перше, це середовище підтримується компанією Microsoft, що гарантує стабільність та безпеку. По-друге, має багато корисних функцій, таких як автодоповнення, відступи, відладка, підтримка систем контролю версій та інші, що значно полегшують процес розробки та роблять його більш ефективним.

Також серед переваг Visual Studio Code можна виділити безкоштовну версію та можливість налаштування під потреби розробника. Крім того, Visual Studio Code має велику кількість додатків та плагінів, що дозволяють доповнювати розробку новими функціями та інструментами. Нарешті, Visual Studio Code підтримує усі необхідні розширення та фреймворки, що дозволить ефективно створювати веб-додатки з використанням цих технологій.

Отже, вибраними засобами реалізації завдання є мова JavaScript, його розширення TypeScript та JSX. Фреймворк ReactJS та середовище розробки Visual Studio Code.

3.2 Розробка модуля керування обліковими записами

Функціональний модуль, що реалізовуватиме процес керування обліковими записами у веб-системі, складається з блоків реєстрації та авторизації користувача.

Блок реєстрації необхідний для того, щоб відвідувачі могли створити облікові записи, оскільки можливості не авторизованих відвідувачів будуть обмежені переглядом головної сторінки та сторінки з уроками, а при спробі відкрити якийсь урок система відкриватиме форму авторизації з повідомленням про необхідність увійти в систему для перегляду навчального матеріалу.

Блок авторизації потрібен для того, щоб зареєстровані користувачі могли увійти в систему та отримати доступ до перегляду вмісту уроків і тестування або іншого функціоналу веб-системи.

Основні цілі блоку реєстрації:

1. Створення облікового запису: блок реєстрації модуля керування обліковими записами дозволяє користувачам ввести дані, такі як ім'я, прізвище, електронна пошта та пароль. Потім ці дані будуть використовуватись для ідентифікації та авторизації користувача в системі.

2. Перевірка та валідація даних: надана користувачем інформація буде перевірятись на наступні умови: наявність електронної пошти у базі даних (користувач вже існує), правильність формату пошти, довжину пароля та наявність заповнення обов'язкових полів.

3. Зберігання даних: після перевірки та валідації даних, модуль реєстрації зберігає інформацію користувача у базі-даних для подальшого її використання.

4. Безпека: блок реєстрації повинен включати заходи безпеки, такі як шифрування паролів, захист від атак переповнення, обмеження спроб авторизації та інші механізми, щоб забезпечити безпеку облікових записів



користувачів.

Рисунок 3.1 – Діаграма послідовності блоку реєстрації

Основні цілі блоку авторизації:

1. Введення та валідація облікових даних: на сторінці авторизації користувач заповнює форму вказуючи пошту та пароль, які було використано при реєстрації. Після чого користувач натискає кнопку “Увійти” і веб-сайт на його стороні перевіряє валідність даних.

2. Перевірка даних у базі: якщо дані проходять валідацію на стороні користувача далі вони надсилаються на сервер Firebase та перевіряються на належність до вже зареєстрованих облікових записів, а також на відповідність комбінації паролю та пошти існуючого облікового запису.

3. Аутентифікація та отримання токена: якщо облікові дані користувача виявляються правильними, сервер Firebase видає токен аутентифікації, що підтверджує валідність користувача. Цей токен за згоди користувача зберігається на клієнтській стороні, для подальшого використання при доступі до сторінок з

теоретичними матеріалами уроків та тестування.

4. Авторизація та управління сесією: після успішної авторизації токен аутентифікації використовується для доступу до захищених ресурсів або сторінок на веб-сайті без необхідності повторного введення облікових даних на кожній сторінці. Сесія авторизації користувача може бути збережена, якщо було обрано функцію «Запам'ятати мене». Та при бажанні вийти з облікового запису користувач може обрати функцію «Вийти», що призведе до закінчення сесії авторизації і знищення токена.

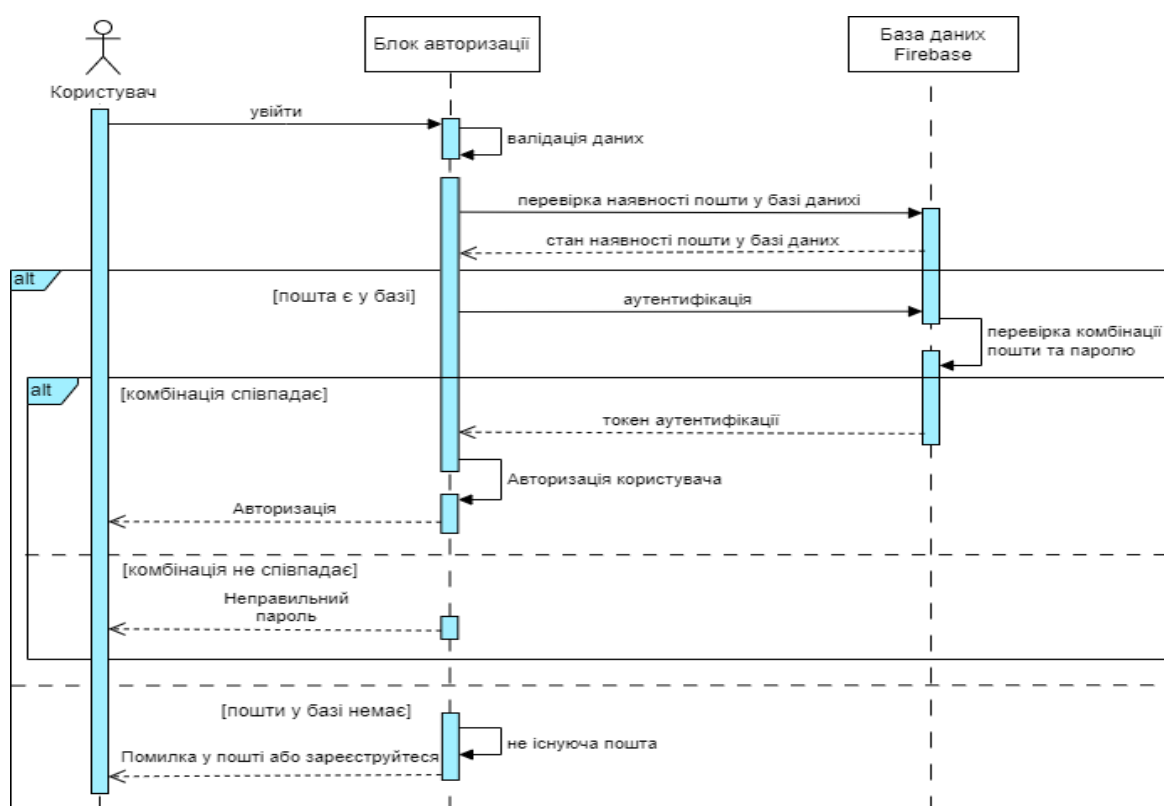


Рисунок 3.2 – Діаграма послідовності блоку авторизації

3.3 Розробка модуля навчальних матеріалів

Основним програмним модулем веб-системи для вивчення математики є модуль, що відповідає за навчальний матеріал, з яким буде взаємодіяти користувач.

У цьому програмному модулі буде використано компонент Lessons, що приймає масив об'єктів Lesson і відображає кожен урок за допомогою

ідентифікатора Lesson_ID – значення типу string, що містять інформацію про ідентифікатор та зміст уроку, включаючи заголовок, теорію та масив питань і відповідей для тестування. Він відображає цю інформацію у вигляді заголовку, тексту теорії та списку з питаннями тесту.

Застосування такої структури дозволить легко управляти навчальними матеріалами та додавати нові уроки і їх підрозділи без необхідності змінювати код компонентів, що відповідають за відображення. Такий компонент можна використовувати для відображення навчальних матеріалів в кожному уроці.

3.4 Розробка модуля проміжного тестування

Модуль проміжного тестування є важливою складовою веб-системи навчання, і його розробка має на меті оцінити знання користувача після проходження певного проміжку матеріалу. Основна суть розробки модуля проміжного тестування полягає в тому, щоб дозволити користувачеві пройти тест, який містить питання та варіанти відповідей, щоб перевірити його розуміння та запам'ятовування навчального матеріалу.

Цей модуль приймає наступні вхідні параметри:

- questions: масив об'єктів, що містять запитання та відповіді на них.

Кожен об'єкт містить наступні поля:

- question: рядок, що містить запитання;
- answers: масив рядків, що містять варіанти відповідей на запитання;
- correctAnswer: рядок, що містить правильну відповідь на запитання.
- onSubmit: функція, що викликається при поданні форми тестування.

Функція повинна приймати один параметр – масив об'єктів, що містять відповіді на запитання. Кожен об'єкт містить наступні поля:

- question: рядок, що містить запитання;
- answer: рядок, що містить відповідь на запитання. Основні функції

модуля проміжного тестування включають:

1. Завантаження тесту: Модуль дозволяє завантажити тест для кожного уроку, який містить питання та варіанти відповідей або можливість введення

відповіді у поле. Це дозволяє користувачу перевірити свої знання, відповідаючи на питання, пов'язані з пройденим матеріалом.

2. Перевірка тесту: Після заповнення користувачем тестувальної форми, модуль перевіряє відповіді та обчислює кількість правильних відповідей. Якщо тест пройдено без помилок, користувач переходить до наступного уроку. В іншому випадку, якщо є помилки, користувач переглядає сторінку з поясненням помилок та можливістю переглянути правильні відповіді.

3. Збереження результатів тестування: Результати тестування зберігаються у базі даних веб-системи для подальшого використання. Це дозволяє вести статистику успішності користувачів, аналізувати їх прогрес та надавати детальне відображення успішності у вигляді діаграм та гістограм.

Алгоритм роботи модуля наведено на рисунку 3.4.

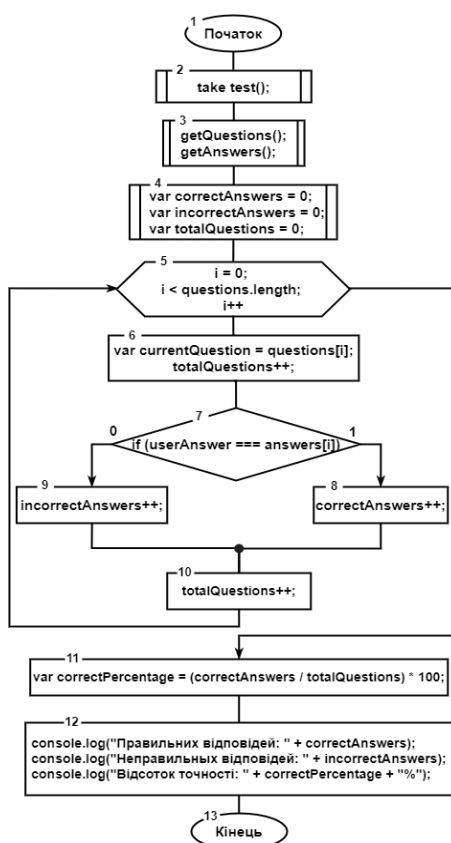


Рисунок 3.4 – Блок-схема алгоритму модуля проміжного тестування

3.5 Розробка модуля формування і відображення прогресу навчання

Відповідно до методу описаному у розділі 2.3. реалізовано модуль, що передбачає наведення результатів проходження тестів користувачем у вигляді

відсоткових гістограм, а також двох діаграм, що показують середній результат користувача за усі тести, та унікальність такого результату.

Кнопка «Сформувати звіт», знаходиться на сторінці з уроками та містить власний id, для якого є обробник подій. Коли користувач натискає цю кнопку, скрипт спрацьовує і викликає функцію formReport.

Функція formReport містить у собі інші функції які викликаються для отримання та форматування даних, а також представлення їх у графічному вигляді:

1. Запускається функція useEffect, яка відповідає за отримання даних з Firebase Firestore та їх підготовку для використання. У цій функції використовується метод getDataFromFirestore який виконує з'єднання з базою даних Firebase Firestore і отримує посилання на колекції «users» та «results», спроектовані у базі-даних відповідно до моделі системи та її основних класів.

2. Потім цей ж метод отримує документ користувача за його ідентифікатором “user_id” і отримує з цього документа дані користувача, такі як: прізвище, ім'я, відсоток вивченого матеріалу. Далі виконується запит до колекції «results», де за запитом “where(“user_id”, “==”, user_id)” отримуються результати, пов'язаних з конкретним користувачем. Результати запиту зберігаються у змінній «results_data». Після чого метод повертає об'єкт, що містить дані користувача (“user_data”) та результати тестів (“results_data”).

3. Далі дані користувача та результати тестів передаються до методу formatDataForChart, який форматує їх для відображення в графічному інтерфейсі. Створюється масив “formattedData”, до якого додаються дані про користувача: прізвище, ім'я, відсоток вивченого матеріалу та результати кожного тесту. Кожен елемент масиву “formattedData” містить об'єкт з властивостями: “primary” – номер тесту; “secondary” – результатом користувача; та “tertiary” – середній результат за тест серед усіх користувачів навчальної веб-системи, які використовуються для відображення відповідних даних в графічному інтерфейсі. Після чого функція повертає сформатований масив даних .

4. Результати тестів з «results_data» отримуються функцією

calculateAverageResult, яка обчислює середнє значення результатів користувача. Цей процес відбувається шляхом обчислення суми усіх результатів, перебираючи елементи масиву «results_data», та ділячи їх на загальну кількість результатів “results_data.length”. Результат обчислення є середнім значенням результатів користувача. Після чого ця функція повертає значення змінної “avgResult” для подальшого використання.

5. Середнє значення результатів користувача “avgResult” має два призначення: формування діаграми «Середній результат користувача», та у розрахунку наступної діаграми «Такий ж результат має користувачів». Для цього функція checkOriginality отримує значення змінної “avgResult” і перевіряє кількість результатів у базі даних, які співпадають з цим середнім значенням, а саме створюється з’єднання з базою даних Firestore, отримується посилання на колекцію «results» та виконується запит з умовою “where(“result”, “==”, avgResult)”. Результати запиту отримуються у вигляді “resultsQuerySnapshot”, і їх кількість отримується за допомогою властивості “size”. Після чого отримується загальна кількість користувачів та обчислюється відсоток результатів, що співпадають з середнім значенням: $percentage = (resultsCount / totalUsers) * 100$. В результаті чого отримується число, що представляє кількість результатів, які співпадають з середнім значенням.

6. Усі отримані дані формуються у графічне відображення у гістограмах та кругових діаграмах функцією chartData.

7. Останнім кроком функція getCurrentDate отримує поточну дату без годин та встановлює її у форматі “\${day}-\${month}-\${year}”. А функція generateReportFilename отримує з бази даних значення «report_number» та оновлює його на інкрементоване значення, яке щойно було присвоєно створеному файлу-звіту.

Після чого користувач переадресується на сторінку із сформованим звітом про свою результативність. Де він може переглядати його, або скачати: функція яку за замовчуванням надає браузер.

Блок-схема алгоритму роботи модуля наведена на рисунку 3.5

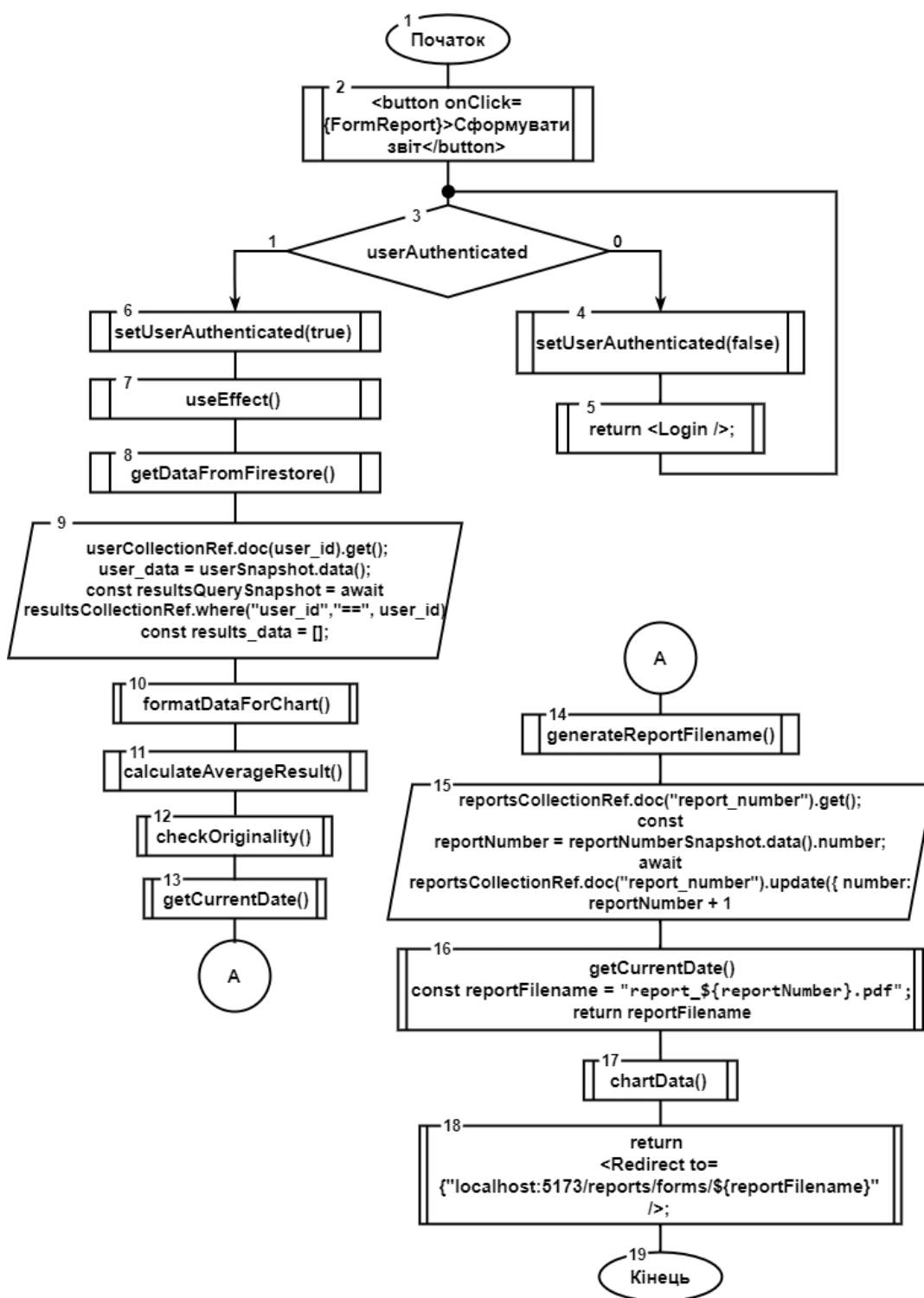


Рисунок 3.5 – Блок-схема алгоритму роботи модуля

3.6 Розробка структури та інтерфейсу веб-системи

Розробка структури інтерфейсу є важливим етапом у процесі розробки цифрових продуктів, таких як веб-сайти, мобільні додатки та програмне забезпечення. Вона визначає організацію та взаємозв'язок елементів і

компонентів інтерфейсу, що забезпечують зручну та ефективну взаємодію користувача з продуктом.

Для розробки структури інтерфейсу використовуються різні методи. Один з найпоширеніших – це використання карти сайту або дерева навігації. Карта сайту відображає ієрархію сторінок та взаємозв'язки між ними, що допомагає організувати інформацію та полегшує навігацію для користувачів. Карта сайту може бути представлена у вигляді текстового документа або візуальної схеми зі зв'язками між сторінками.

При розробці структури інтерфейсу необхідно враховувати потреби користувачів, інформаційного упорядкування, візуального дизайну. Елементи та компоненти інтерфейсу повинні бути логічно організовані та легко доступні, щоб користувачі могли швидко знайти необхідну інформацію або виконати необхідну дію. Розроблена карта сайту для навчальної веб-системи з математики зображена на рисунку 3.7.

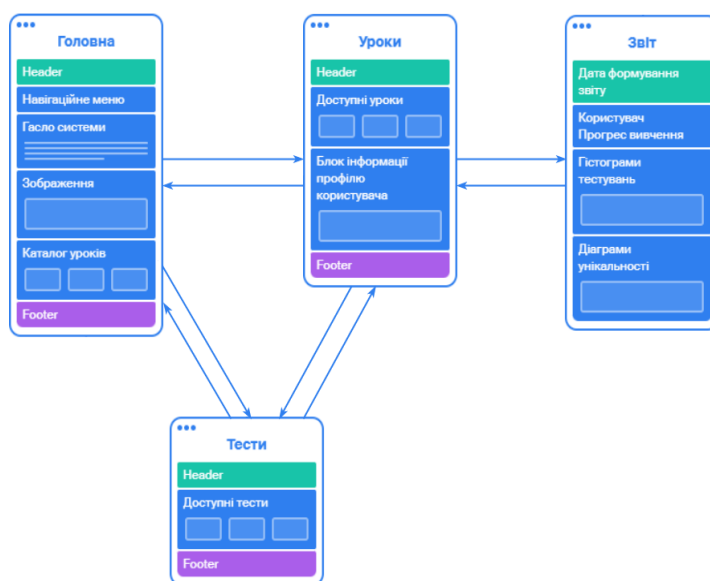


Рисунок 3.7 – Карта сайту для навчальної веб-системи

Для структури інтерфейсу було обрано стандартний тип коли основна веб-сторінка містить посилання на інші сторінки веб-сайту, а документи містять посилання, відповідно, на основну веб-сторінку.

Ще одним методом розробки структури інтерфейсу є створення прототипів. Прототипи дозволяють візуалізувати та перевірити взаємодію

елементів та компонентів інтерфейсу перед реалізацією. Вони можуть бути статичними (наприклад, зображеннями) або інтерактивними (з можливістю взаємодії з елементами). Прототипування дозволяє виявити потенційні проблеми та внести необхідні зміни ще на ранніх етапах розробки.

Статичний прототип інтерфейсу розроблений для навчальної веб- системи зображено на рисунку 3.8.

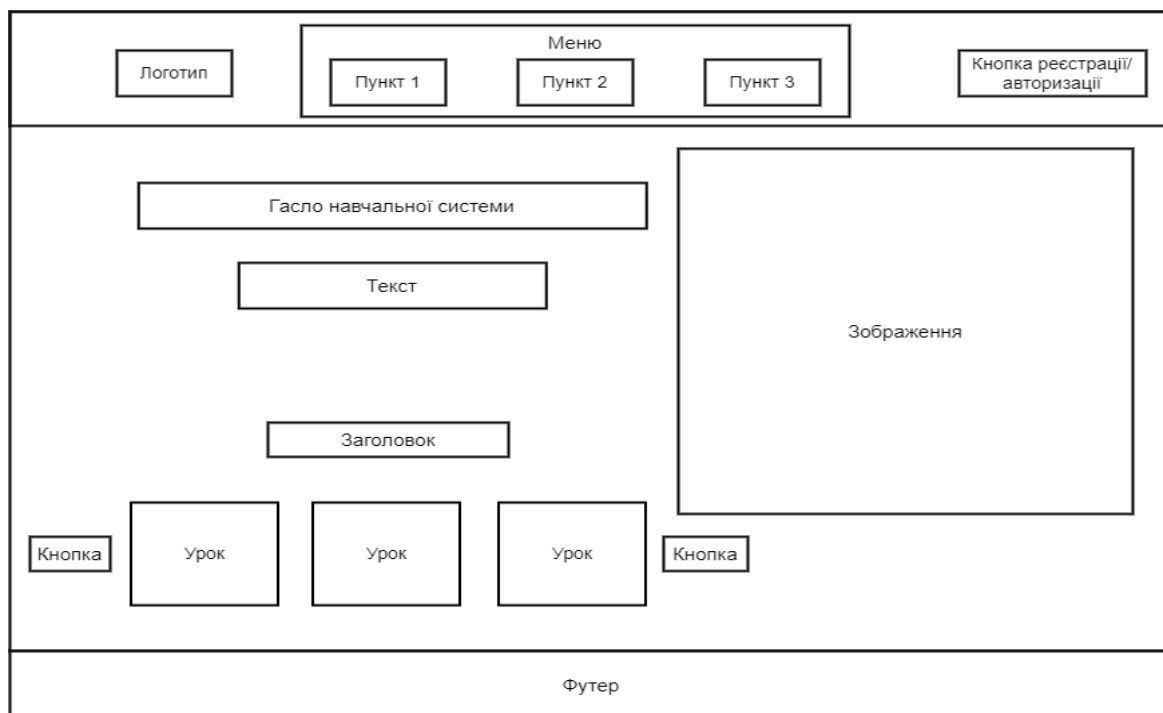


Рисунок 3.8 – Статичний прототип інтерфейсу головної сторінки

3.7 Розробка дизайну інтерфейсу веб-системи

Для комфортної взаємодії користувачами із сайтом його інтерфейс повинен бути “дружнім” [24].

Дружній веб-інтерфейс відображає простоту та зручність взаємодії з ресурсом. Головним критерієм хорошого інтерфейсу є ергономічність – необхідність якомога меншої кількості дій зі сторони користувача, для досягнення його мети. Ніщо не повинно відволікати увагу відвідувача, перешкоджати або ускладнювати використання системи.

Запорукою зручного інтерфейсу є декілька критеріїв:

1. Логічність дизайну. Звичне розташування елементів, зручні кнопки

навігація, та інтуїтивна зрозумілість. Відвідувачу повинно бути очевидним куди йому потрібно натискати, щоб досягнути бажаного результату.

2. Передбачуваність. Інтерфейс повинен бути асоціативним порівняно із схожими типами веб-ресурсів. Тоді користувачі будуть орієнтуватись покладаючись на досвід взаємодії з іншими схожими платформами.

3. Швидкість завантаження. Перевантаження сайту різноманітними елементами та анімаціями може сповільнити його роботу, що негативно вплине на враження відвідувачів сайту.

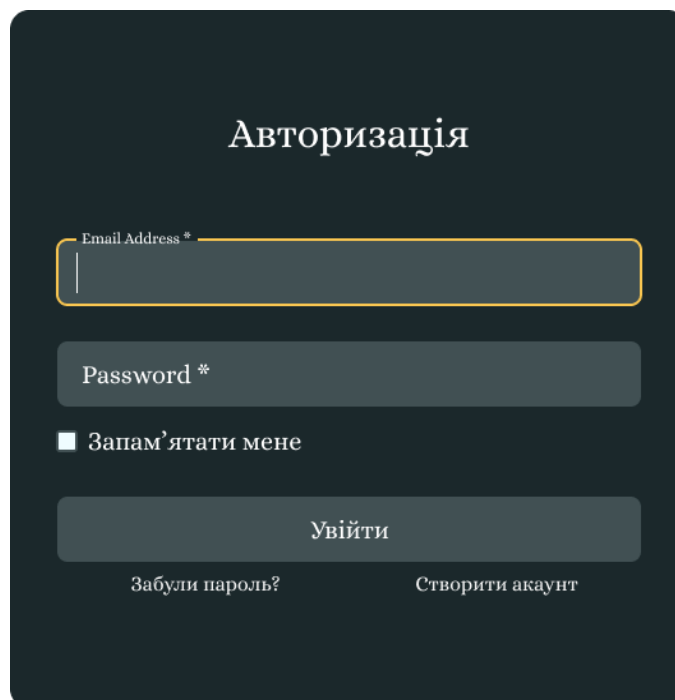
Орієнтуючись на вищеописані принципи було розроблено наступні елементи веб-сервісу:

Головна сторінка: головна сторінка веб-системи містить посилання на каталог уроків, загальну інформацію про систему, посилання на реєстрацію або вхід для зареєстрованих користувачів.

Каталог доступних уроків: каталог містить список доступних уроків з тематикою уроку та відповідним зображенням. Також на сторінці наведено інформацію про користувача та його загальну статистику. Користувач може переглядати каталог, обрати урок або за потреби сформулювати більш детальне наведення прогресу його навчання – звіт.

Кожен урок має свою власну сторінку з необхідною для вивчення матеріалу інформацією у вигляді тексту й ілюстрацій або уроків. Також сторінка містить блок з навігацією по підрозділам уроку, відміткою про завершення того чи іншого підрозділу, а також посиланням на підсумковий тест.

Сторінка реєстрації та авторизації надає можливість авторизуватись або ж зареєструватись, щоб зберігати прогрес. Макет інтерфейсів реєстрації та авторизації зображено на рисунках 3.9 та 3.10.



Авторизація

Email Address *

Password *

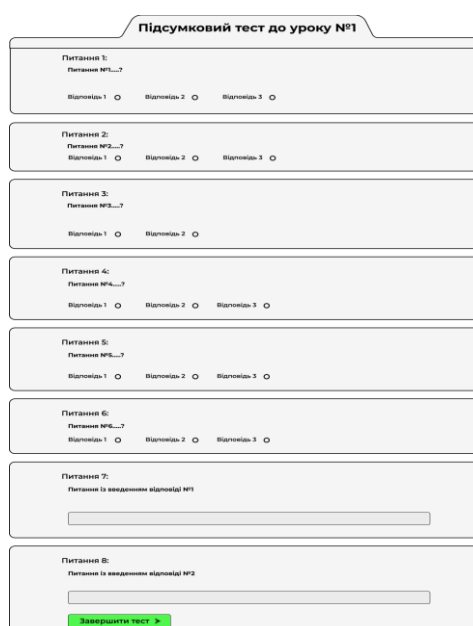
☐ Запам'ятати мене

Увійти

Забули пароль? Створити акаунт

Рисунок 3.10 – Макет дизайну інтерфейсу авторизації

Сторінка тестування: містить тест з автоматичною перевіркою відповідей для закріплення отриманих знань. Макет сторінки тестування зображено на рисунку 3.11.



Підсумковий тест до уроку №1

Питання 1:
Питання №1...?

Відповідь 1 ☐ Відповідь 2 ☐ Відповідь 3 ☐

Питання 2:
Питання №2...?

Відповідь 1 ☐ Відповідь 2 ☐ Відповідь 3 ☐

Питання 3:
Питання №3...?

Відповідь 1 ☐ Відповідь 2 ☐

Питання 4:
Питання №4...?

Відповідь 1 ☐ Відповідь 2 ☐ Відповідь 3 ☐

Питання 5:
Питання №5...?

Відповідь 1 ☐ Відповідь 2 ☐ Відповідь 3 ☐

Питання 6:
Питання №6...?

Відповідь 1 ☐ Відповідь 2 ☐ Відповідь 3 ☐

Питання 7:
Питання із введеними відповідями №1

Питання 8:
Питання із введеними відповідями №2

Завершити тест ➤

Рисунок 3.11 – Макет дизайну інтерфейсу сторінки тестування

Сторінка файлу-звіту: також було розроблено макет інтерфейсу елементів файлу-звіту, який користувач може сформувати з свого профілю на сторінці з уроками. Файл містить інформацію про дату формування, навчальну веб систему, дату формування та статистику успішності користувача. Розроблений макет наведено на рисунку 3.12.

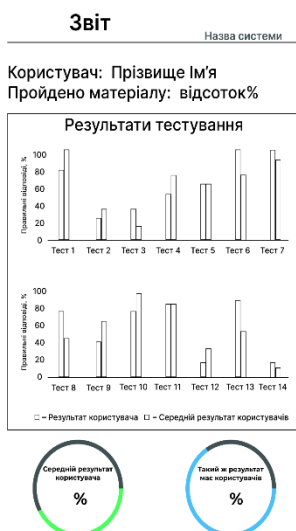


Рисунок 3.12 – Макет інтерфейсу елементів файлу-звіту

Розробка всього графічного інтерфейсу веб-системи була проведена у графічному редакторі Figma.

3.8 Висновки

Отже, у третьому розділі було проведено варіантний аналіз та обґрунтовано вибір середовища розробки MS Visual Studio Code та мови програмування Java Script разом з JSX, TypeScript та фреймворком React для розробки функціоналу веб-системи. За допомогою вибраних засобів було здійснено розробку основних програмних модулів, а саме:

- модуль керування обліковими записами, для реєстрації, авторизації та використання облікового запису користувачем;
- модуль навчальних матеріалів, що дозволяє розміщувати та відображати уроки;
- модуль проміжного тестування, що дозволяє перевірити розуміння та запам'ятовування навчального матеріалу користувачем;

– модуль формування і відображення прогресу навчання, що дозволяє користувачеві отримати зручне представлення результатів проходження тестів у вигляді відсоткових гістограм, а також двох діаграм, що показують середній результат користувача за усі тести, та унікальність такого результату.

Наведено діаграми послідовності та блок-схеми роботи алгоритмів розроблених модулів. Також було обрано стандартну структуру сайту, розроблено карту сайту та статичний прототип інтерфейсу головної сторінки. Змодельовано макети веб-сторінок: головна, каталог уроків, урок, реєстрація, авторизація, тест та макет структури елементів звіту.

4. ТЕСТУВАННЯ ПРОГРАМИ

4.1 Аналіз методів тестування

Програмні системи стали невід’ємною частиною нашого повсякденного життя. Більшість людей стикалися з програмним забезпеченням, яке не працювало так, як очікувалося. Неналежна робота програм може призвести до значних проблем, включаючи втрату грошей, часу, ділової репутації, а в крайніх випадках навіть до травм або смерті.

Тестування програмного забезпечення оцінює його якість і допомагає знизити ризик збоїв під час експлуатації. Це комплекс заходів для виявлення дефектів і оцінки якості програмних компонентів, відомих під час тестування як тестові об’єкти. Поширена помилка полягає в тому, що тестування складається лише з виконання тестів (тобто запуску програмного забезпечення та перевірки результатів). Однак тестування також включає інші види діяльності і має бути узгоджене з життєвим циклом розробки програмного забезпечення.

Інше поширене непорозуміння щодо тестування полягає в тому, що воно повністю зосереджене на перевірці тестового об’єкта. Тестування передбачає не лише верифікацію, тобто перевірку відповідності системи заданим вимогам, але й валідацію, тобто перевірку відповідності системи потребам користувачів та інших зацікавлених сторін у реальному робочому середовищі.

Тестування може бути динамічним і статичним. Динамічне тестування передбачає виконання програмного забезпечення, тоді як статичне тестування – ні. Статичне тестування включає огляди і статичний аналіз. Динамічне тестування використовує різні методики та підходи для створення тестових випадків. Тестування – це не лише технічна діяльність. Воно потребує ретельного планування, управління, оцінки, контролю та моніторингу [17].

Методи тестування допомагають тестувальнику аналізувати, що слід тестувати, і розробляти, як це тестувати. Вони дозволяють систематично створювати відносно невеликий, але достатній набір тестів. Крім того, методи тестування допомагають визначати умови тестування, ідентифікувати елементи

покриття та обирати тестові дані під час аналізу та розробки тестів.

Методи тестування чорної скриньки (також відомі як методи на основі специфікацій) ґрунтуються на аналізі заданої поведінки тестового об'єкта без врахування його внутрішньої структури. Тестові випадки, таким чином, не залежать від реалізації програмного забезпечення, тому залишаються корисними навіть при зміні реалізації, якщо поведінка системи залишається незмінною.

Методи тестування білої скриньки (також відомі як методи на основі структури) базуються на аналізі внутрішньої структури та обробки тестового об'єкта. Оскільки тестові випадки залежать від реалізації програмного забезпечення, вони можуть бути створені лише після проектування або реалізації об'єкта тестування.

Методи тестування на основі досвіду використовують знання та досвід тестувальників для розробки та виконання тестових випадків. Ефективність цих методів значною мірою залежить від навичок тестувальника. Вони можуть виявити дефекти, які залишаються непоміченими за допомогою методів чорної та білої скриньки, тому методи на основі досвіду є важливим доповненням до інших методів тестування [17].

На основі вищезазначеної інформації можна зробити висновок, що для повного покриття тестами розробленого додатку необхідно застосувати різноманітні техніки тест дизайну. Це дозволить забезпечити всебічне тестування програмного забезпечення та виявити потенційні дефекти на різних рівнях його функціонування. Техніки тест дизайну – це методи, що допомагають у створенні тестових випадків для перевірки програмного забезпечення на відповідність вимогам. Вони дозволяють тестувальникам систематично й ефективно підходити до тестування, забезпечуючи високу якість продукту [18].

Техніки тестування чорної скриньки базуються на аналізі специфікацій і вимог без врахування внутрішньої структури програми. До них належать [19]:

- Еквівалентне розбиття: розподіл вхідних даних на рівнозначні класи, для кожного з яких передбачається однаковий результат.

- Аналіз граничних значень: тестування меж еквівалентних класів для виявлення помилок, які можуть виникнути на границях діапазонів.
- Таблиця прийняття рішень: використання таблиць для опису логіки прийняття рішень і відповідних результатів на основі комбінацій вхідних умов.
- Тестування стану переходів: перевірка системи на основі можливих станів і переходів між ними відповідно до діаграми станів.
- Тестування на основі сценаріїв використання (Use Case Testing): тестування шляхом моделювання реальних сценаріїв використання програмного забезпечення користувачами.
- Техніки тестування на основі досвіду використовують знання і досвід тестувальників для створення тестових випадків [19]:
- Аналіз помилок (Error Guessing): створення тестових випадків на основі передбачуваних помилок, які можуть виникнути в системі.
- Тестування на основі чек-листів (Checklist-based Testing): використання попередньо підготовлених списків можливих дефектів і типових помилок.
- Адаптивне тестування (Exploratory Testing): динамічне тестування, де тестувальник досліджує систему без чітко визначених тест-кейсів, щоб виявити нові дефекти.

Використовуючи описані техніки тест дизайну, було прийнято рішення створити тестову документацію, включаючи тест-кейси та чек-листи, щоб повністю покрити додаток тестами. Це дозволить мінімізувати ризик виникнення дефектів під час подальшого використання додатку користувачами, що в свою чергу позитивно вплине на їхнє враження про додаток.

Тестова документація – це набір документів, які описують процес тестування програмного забезпечення та його результати. Вона є критично важливою для забезпечення якості програмного продукту і включає різні типи документів, що допомагають планувати, проводити, контролювати та оцінювати тестування [19].

Тест-кейс – це окремий документ, який описує конкретний тестовий сценарій для перевірки певного аспекту функціональності програмного

забезпечення. Він містить детальну інформацію про умови, кроки, вхідні дані та очікувані результати, необхідні для проведення тестування. Тест-кейси є основним інструментом, що використовується тестувальниками для систематичної перевірки та верифікації роботи програмного продукту. Тест-кейси є важливим інструментом у процесі тестування, забезпечуючи ретельну перевірку програмного забезпечення, мінімізуючи ризик помилок і підвищуючи загальну якість продукту.

Чек-лист (або список перевірок) – це інструмент, який використовується для систематизації і фіксації перевірок під час тестування програмного забезпечення. Він являє собою перелік елементів, критеріїв або дій, які необхідно виконати для забезпечення повного і якісного тестування продукту. Чек-листи є важливим інструментом для організації і контролю тестування програмного забезпечення, забезпечуючи послідовність і повноту перевірок, що сприяє підвищенню загальної якості продукту [19].

4.2 Тестування онлайн платформи

Тестування веб-платформи є важливою частиною розробки веб-системи для вивчення математики, оскільки воно дозволяє перевірити функціональність, надійність, та коректність роботи веб-додатку перед його випуском у виробництво. Тестування допомагає виявляти можливі дефекти, помилки та проблеми в роботі веб-системи та забезпечує високу якість кінцевого продукту.

Для початку протестуємо швидкість завантаження веб-сторінки для вивчення математики. Для цього необхідно зайти на головну сторінку та відкрити інструменти розробника. Загальний час завантаження головної сторінки становив 1.13 с., що є достатньо гарним показником.

Під час тестування сторінки реєстрації користувача було виконано спробу зареєструвати користувача з порожнім полем імені, поштою чи паролем. В результаті, система відобразила сповіщення, що користувач не може бути зареєстрований без введеного імені, пошти чи паролю.

Така поведінка системи правильна, оскільки ці поля є обов'язковими для подальшої роботи користувача. Результати цього тестування наведені на рисунку 4.1.

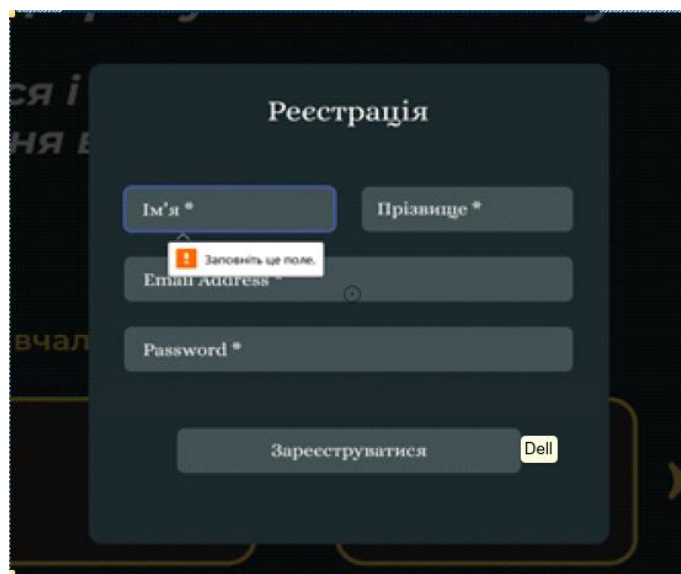


Рисунок 4.1 – Спроба реєстрації без заповнення обов'язкових полів

Переконавшись, що система не дозволяє реєструвати користувача без обов'язкових полів, спробуємо ввести невалідне значення в поле електронної адреси. В результаті тестування, система відобразить сповіщення про неправильну електронну адресу, яке можна побачити на рисунку 4.2.

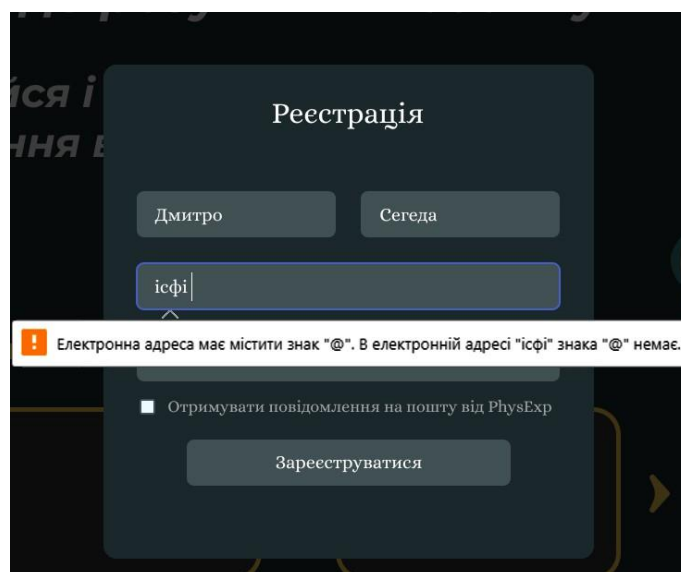


Рисунок 4.2 – Спроба реєстрації з невалідною поштою

Після авторизації було протестовано швидкість завантаження сторінки з

уроками. Загальний час завантаження сторінки був гіршим ніж на головній сторінці, проте знаходився у межах оптимально допустимих значень.

Найголовнішим аспектом тестування веб-системи для вивчення математики є відображення сторінки уроку, який включає, теоретичний матеріал та тестування.

Після успішної реєстрації було проведено перевірку авторизації.

Спочатку було введено невалідну пошту, потім в поле пароллю не було введено жодних значень і останньою перевіркою було введено неправильний пароль. Результати тестування зображено на рисунках 4.3, 4.4 та 4.5.

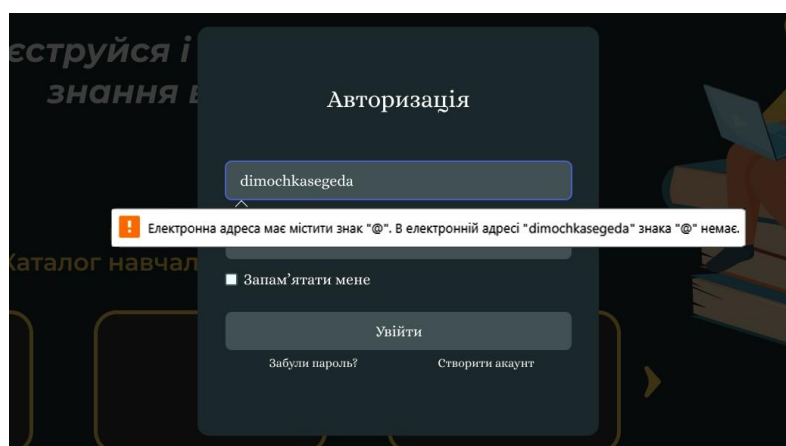


Рисунок 4.3 – Спроба авторизації з невалідною поштою

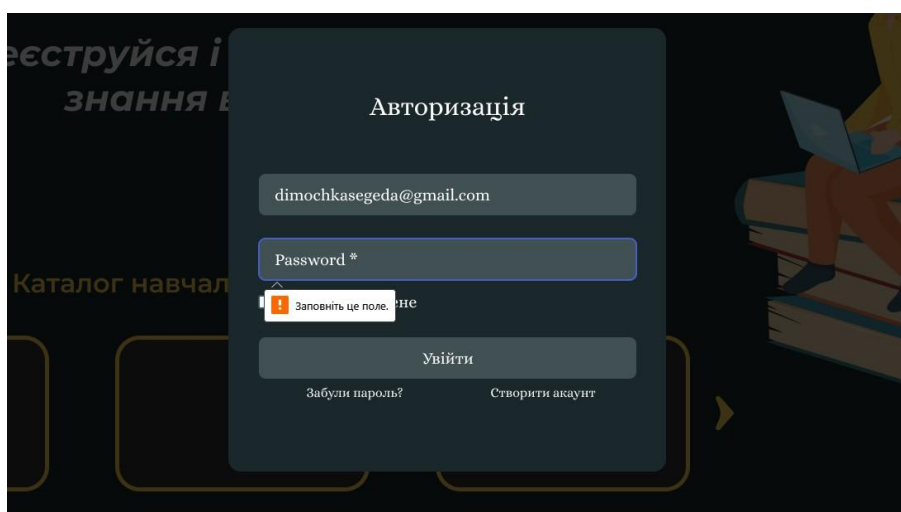


Рисунок 4.4 – Спроба авторизації без введення пароллю

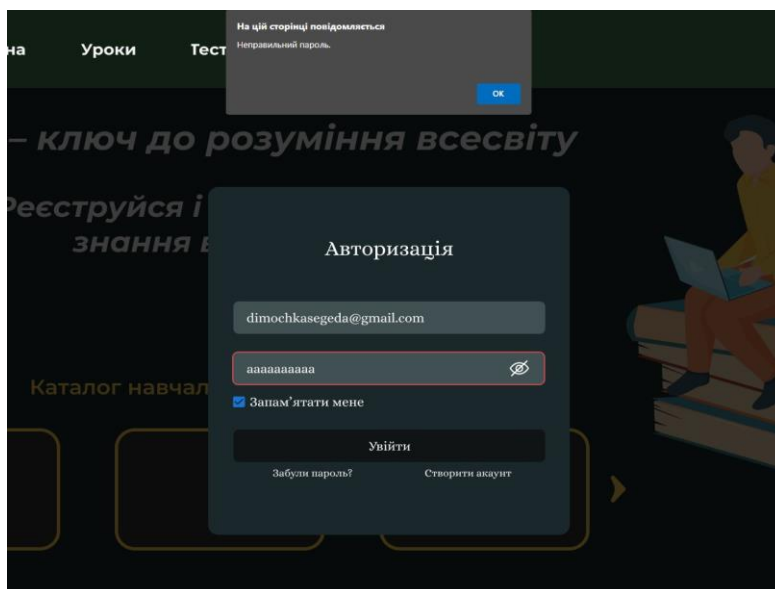


Рисунок 4.5 – Спроба авторизації з неправильним паролем

Після авторизації було протестовано швидкість завантаження сторінки з уроками. Загальний час завантаження сторінки був гіршим ніж на головній сторінці, проте знаходився у межах оптимально допустимих значень.

Найголовнішим аспектом тестування веб-системи для вивчення математики є відображення сторінки уроку, який включає, теоретичний матеріал та тестування.

Передбачається, що користувачі зможуть переглянути прочитати теоретичний конспект, зрозуміти матеріал та перевірити своє розуміння за допомогою тесту.

Сторінка тестування має бути спеціально розроблена для того, щоб дозволити користувачеві пройти тестування після кожного уроку і перевірити свої знання з поточних тем.

Основними елементами сторінки тестування повинні бути:

- заголовок сторінки, який повинен вказувати назву тесту;
- список запитань з варіантами відповідей.

Важливо врахувати, що кожен тест повинен бути добре збалансованим та вимірним, і не повинен бути надто легким або складним для цільової аудиторії.

Для проведення тестування веб-системи було створено тестового

користувача, якого також було використано для перевірки формування файлу-звіту. Для цього у базі даних було встановлено результати проходження тестів цим користувачем, а також заповнено значення середнього відсотку результатів за той чи інший тест, щоб перевірити відображення гістограм та діаграм у звіті.

4.3 Розробка інструкції користувача

Інструкція користувача передбачає визначення технічних вимог для запуску програмного продукту. Деталі щодо мінімальної та рекомендованої конфігурації серверного комп'ютера можна знайти в таблицях 4.1 та 4.2.

Таблиця 4.1 – Мінімальна конфігурація:

Тип процесора	32-розрядний (x86) або <u>64-розрядний (x64)</u> процесор з тактовою частотою 1 ГГц
Об'єм оперативної пам'яті	2 ГБ
Місце на жорсткому диску	1 ГБ
Операційна система	Windows 7

Таблиця 4.2 – Рекомендована конфігурація:

Тип процесора	64-розрядний (x64) процесор з тактовою частотою 2 ГГц
Об'єм оперативної пам'яті	4 ГБ
Розмір жорсткого диску	128 ГБ
Операційна система	Windows 10

Для встановлення програмного продукту потрібно використовувати Docker – це платформа, яка дозволяє розробляти, тестувати та розгортати додатки швидко та ефективно. Docker упаковує програмне забезпечення в контейнери, які містять усі необхідні компоненти для роботи програми, включаючи бібліотеки, системні інструменти, код та середу виконання [25].

Використання Docker дозволяє користувачам швидко розгортати та масштабувати програмний продукт в будь-якому середовищі, забезпечуючи стабільну роботу програми. Використання Docker також зменшує необхідність вручну налаштовувати середовище, оскільки користувачам достатньо виконати кілька команд для запуску контейнерів на онлайн-платформі.

Після входу в систему користувач повинен мати доступ до головної сторінки, на якій він зможе вибрати тему для вивчення математики. Після вибору теми користувач отримає список доступних уроків. Для отримання більш детальної інформації щодо певного уроку, користувач повинен перейти на сторінку уроку. На даній сторінці він зможе ознайомитися з теоретичним матеріалом та пройти тестування для закріплення своїх знань.

Після завершення уроку користувач може переглянути свій прогрес та зберегти результати вивчення.

4.4 Висновки

У четвертому розділі було проведено тестування веб-системи для вивчення математики. Було перевірено швидкість відображення сторінок веб-системи, валідацію полів введення інформації у формах реєстрації та авторизації.

Перевірено відображення елементів уроку, та сформовано тестовий файл-звіт. Також було розроблено інструкцію користувача по встановленню та початковій експлуатації програмного продукту.

В результаті тестування розроблена веб-система показала цілковиту працездатність програмного продукту та відповідність поставленому технічному завданню.

ВИСНОВКИ

У бакалаврській дипломній роботі було розроблено навчальну веб-систему «MathExp» для підвищення ефективності та інтерактивності процесу вивчення математики за рахунок систематизації навчального матеріалу, проведення тестового контролю та наведення ілюстративного матеріалу.

Проведено аналіз стану навчальних веб-систем, розглянуто існуючі аналог та визначено їхні переваги та недоліки в порівнянні з навчальною веб- системою для вивчення математики та встановлено, що розроблюваний програмний засіб переважає у функціональності розглянуті аналоги.

Базуючись на результатах аналізу та порівняння було поставлено останні задачі бакалаврської дипломної роботи та обрано методи розробки веб-системи. Проведено варіантний аналіз та обґрунтування вибору засобів реалізації програмного продукту. В якості мови програмування було обрано мову програмування: JavaScript, її розширення TypeScript та JSX. Фреймворк ReactJS та середовище розробки Visual Studio Code.

Відповідно до поставлених задач було:

- на прикладі UML-діаграм діяльності та класів розроблено модель навчальної веб-системи для вивчення математики, яка описує основний функціонал онлайн-платформи та взаємодію між основними класами.
- розроблено алгоритми основних програмних компонентів застосунку;
- розроблено функціонал навчальної веб-системи;
- розроблено зручний та легкий інтерфейс користувача для веб-сервісу;
- описано програмну реалізацію основних модулів;
- забезпечено підтримку української мови та доступність веб-системи на різних платформах і пристроях, включаючи мобільні пристрої;
- проведено тестування програми, яке показало цілковиту працездатність програмного продукту та відповідність поставленому технічному завданню.

Крім того було розроблено метод формування і відображення прогресу навчання, який дозволяє згрупувати результати користувача та отримати їх у

порівнянні з результатами інших у зручному графічному вигляді. Що дозволяє користувачам детально відслідковувати навчальний прогрес, за рахунок можливості у будь-який момент навчання сформувати файл-звіт.

Також було розроблено інструкцію користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інтернет [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82>
2. Інформаційно-комунікаційні технології (ІКТ) та їх роль в освітньому процесі [Електронний ресурс] – Режим доступу до ресурсу: <https://osvita.ua/school/method/technol/6804/>
3. M. Belloni, W. Christian, and A. Cox, "Open Source: A User's Guide with Examples," The MathTeacher – 2006. – 480 с.
4. Результати дослідження якості організації дистанційного навчання у закладах загальної середньої освіти України [Електронний ресурс] – Режим доступу до ресурсу: https://znayshov.com/News/Details/rezultaty_doslidzhennia_Yakosti_orhanizatsii_dystantsiinoho_navchannia
5. Закасовський Б. І. Розробка навчальної веб-системи для вивчення математики / Б. І. Закасовський, Г. О. Черноволик // Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)» , Вінниця, 2024. [Електронний ресурс] – Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/view/21163>.
6. E. I. Ezenwoke and R. O. Nwokorie, "Design and Implementation of a Web-Based MathTutor," International Journal of Emerging Science and Engineering – 2019. – 140 с.
7. S. K. Halder, S. Roy, S. Saha, and S. Sarkar, "An Augmented Reality-Based Educational Game for Learning," in Proceedings of the 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India – 2016. – 139 с.
8. MathET Interactive Simulations [Електронний ресурс] – Режим доступу до ресурсу: <https://phet.colorado.edu/uk/>.
9. OpenStax [Електронний ресурс] – Режим доступу до ресурсу: <https://openstax.org/>.
10. 7 Ways To Build Interactive Video Learning [Електронний ресурс] –

Режим доступу до ресурсу: <https://elearningindustry.com/7-ways-build-interactive-video-learning>

11. How to personalize content for your learning audience [Електронний ресурс] – Режим доступу до ресурсу: <https://www.allencomm.com/blog/custom-elearning-development-topics-how-to-personalize-content-for-your-learning-audience/>

12. Project-Based Learning: Definition, Benefits & Strategies [Електронний ресурс] – Режим доступу до ресурсу: edutopia.org/project-based-learning.

13. Gamification in Education: 4 Key Benefits [Електронний ресурс] – Режим доступу до ресурсу: tophat.com/blog/education-technology/.

14. Personalized Learning: What It Really Is and Why It Really Matters [Електронний ресурс] – Режим доступу до ресурсу: <https://er.educause.edu/articles/2016/3/personalized-learning-what-it-really-is-and-why-it-really-matters>

15. Michael B. Horn, Heather Staker, Clayton M. Christensen. Blended: Using Disruptive Innovation to Improve Schools – 2017. – 336 с.

16. Alan Forbes. The Joy of PHP – 2015. – 128 с.

17. Douglas Crockford. JavaScript: The Good Parts – 2013. – 176 с.

18. Paul Barry. Head First Python: A Brain-Friendly Guide – 2016. – 622 с.

19. Adam Boduch, Roy Derks. React and React Native: A complete hands-on guide to modern web and mobile development with React.js – 2020. – 526 с.

20. Visual Studio Code [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com/>.

21. WebStorm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jetbrains.com/webstorm/>.

22. Atom [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/atom/>.

23. Дружній інтерфейс [Електронний ресурс] – Режим доступу: <https://pbb.lviv.ua/statti-i-novyny/statti-shchodo-stvorennia-saitu/jak-zrobyty-dyzayn-saytu-zruchnym/>

ДОДАТКИ

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

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

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
О. Н. Романюк
« 12» березня 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на бакалаврську дипломну роботу «: «Розробка програмних засобів навчальної
веб-системи для вивчення математики»
» за спеціальністю 121 – Інженерія програмного забезпечення

Керівник бакалаврської дипломної роботи:

_____ доц. каф. ПЗ Черноволик Г.О.

«12» березня 2024р.

Виконав:

_____ студент гр. 4ПІ-20Б Закасовський Б. І.

«12» березня 2024р.

Вінниця – 2024 року

1. Найменування та галузь застосування

Бакалаврська дипломна робота: «Розробка програмних засобів навчальної веб-системи для вивчення математики».

Галузь застосування – навчальні інтернет ресурси.

2. Підстава для розробки.

Підставою для виконання бакалаврської дипломної роботи (БДР) є індивідуальне завдання на БДР та наказ №80 від «11» березня 2024 р. ректора ВНТУ про закріплення тем БДР.

3. Мета та призначення розробки.

Метою дослідження є підвищення ефективності процесу вивчення математики за рахунок використання розробленої веб-системи, яка дозволяє систематизувати навчальний матеріал, забезпечує проведення тестового контролю, візуалізує ілюстративний матеріал, що підвищує інтерактивність навчального процесу та сприяє зацікавленості учнів вивченням математики.

Призначення роботи – розробка програмного продукту для вивчення математики.

4. Вихідні дані для проведення НДР

1. E. I. Ezenwoke and R. O. Nwokorie, "Design and Implementation of a Web-Based Math Tutor," International Journal of Emerging Science and Engineering – 2019. – 140 с.
2. How to personalize content for your learning audience [Електронний ресурс] – Режим доступу: <https://www.allencomm.com/blog/custom-elearning-development-topics-how-to-personalize-content-for-your-learning-audience>
3. Adam Boduch, Roy Derks. React and React Native: A complete hands-on guide to modern web and mobile development with React.js – 2020. – 526 с.

5. Технічні вимоги

Серверний комп'ютер з 64-розрядним (x64) процесором та тактовою частотою 2 ГГц. Об'єм оперативної пам'яті 4 ГБ, розмір жорсткого диску 128 ГБ. Операційна система Windows 10.

З клієнтського боку: ОС Windows, Android, IOS та будь-який браузер.

6. Конструктивні вимоги.

Веб-додаток має відповідати ергономічним та технічним вимогам. Текстова та графічна документація повинна відповідати стандартам України.

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

- пояснювальна записка до бакалаврської дипломної роботи;
- технічне завдання;
- лістинги програми.

7. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

8. Стадії та етапи розробки

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз стану питання та постановка задач дослідження	14.03.2024 – 29.03.2024
2	Аналіз вхідних та вихідних даних системи	30.03.2024 – 07.04.2024
3	Розробка моделі та алгоритмів системи	08.04.2024 – 26.04.2024
4	Розробка інтерфейсу	27.04.2024 – 04.05.2024
5	Розробка програмних модулів системи	05.05.2024 – 14.05.2024
6	Тестування системи	15.05.2024 – 20.05.2024
7	Оформлення матеріалів до захисту БДР	21.05.2024 – 30.05.2024

10 Порядок контролю та прийняття.

Всі етапи бакалаврської роботи контролюються науковим керівником згідно плану по виконанню роботи. Прийняття бакалаврської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком захисту. Дозволяється корегування бакалаврської дипломної роботи.

ДОДАТОК Б – ПЕРЕВІРКА НА ПЛАГІАТ
ПРОТОКОЛ
ПЕРЕВІРКИ БАКАЛАВРСЬКОЇ ДИПЛОМНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: : «Розробка програмних засобів навчальної веб-системи для вивчення математики»

Тип роботи: БДР

Підрозділ: кафедра програмного забезпечення, ФІТКІ

Науковий керівник: Черноволик Г.О.

Оригінальність	89,4 %
Схожість	10,6 %

Аналіз звіту подібності

▪ **Запозичення, виявлені у роботі, оформленні коректно і не містять ознак плагіату.**

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

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

Особа, відповідальна за перевірку _____ Черноволик Г.О.

Ознайомлені з повним звітом подібності, який був згенерований системою Unichек

Автор роботи _____

Закасовський Б. І.

Керівник роботи _____

Черноволик Г.О.

Додаток В. Лістинг програми

Лістинг файлу index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<link rel="icon" type="image/svg+xml" href="/vite.svg" />
<link
rel="stylesheet"

href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"
/>
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,600;0,700;1,700&display=swap" rel="stylesheet">
<title>PhysExp</title>
<script type="module" crossorigin src="/assets/index-8a0357b9.js"></script>
<link rel="stylesheet" href="/assets/index-0cd8fcec.css">
</head>
<body>
<div id="root"></div>

</body>
</html>

```

Лістинг файлу main.jsx

```

import React from 'react'
import ReactDOM from 'react-dom/client'
import { RouterProvider } from 'react-router-dom'; import router from

'./router/Router.jsx';

import './index.css'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
  <RouterProvider router={router} />
</React.StrictMode>,
)

```

Лістинг файлу index.css

```

@import
url('https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,600;0,700;1,700&
display=swap');

body {
margin: 0;
padding: 0;
}

```

Лістинг файлу app.jsx

```

import './App.css'

```

```
function App() { return (
  <div>
    App
  </div>
)
}

export default App
```

Лістинг файлу routes.js

```
const ROUTES = { APP: '/',
  LOGIN: '/login', REGISTER: '/register', DASHBOARD: '/dashboard', PROFILE: '/profile',
  LESSONS: '/lessons', LESSON: '/lessons/:id', TESTS: '/tests',
  TEST: '/tests/:id', NOT_FOUND: '*',
};

export default ROUTES;
```

Лістинг файлу Router.jsx

```
import { createBrowserRouter, createRoutesFromElements, Route
} from 'react-router-dom';

// Pages
import Login from '../pages/Login/Login';
import Register from '../pages/Register/Register'; import Dashboard from
'../pages/Dashboard/Dashboard'; import Lessons from '../pages/Lessons/Lessons'; import
Tests from '../pages/Tests/Tests';

import ROUTES from './routes'; const router = createBrowserRouter(
createRoutesFromElements(
  <>
    <Route path={ROUTES.LOGIN} element={<Login/>}/>
    <Route path={ROUTES.REGISTER} element={<Register/>}/>
    <Route path={ROUTES.DASHBOARD} element={<Dashboard/>}/>
    <Route path={ROUTES.LESSONS} element={<Lessons/>}/>
    <Route path={ROUTES.TESTS} element={<Tests/>}/>

    <Route path={ROUTES.NOT_FOUND} element={<div>Not Found</div>}/>
  </>
)
);

export default router;
```

Лістинг файлу Dashboard.jsx

```
import Wrapper from '../components/Wrapper/Wrapper';
import LessonsCarousel from '../components/LessonsCarousel/LessonsCarousel'; import

{ lessons } from '../mocks/lessons';

import './dashboard.scss';

const Dashboard = () => { return (
  <Wrapper>
    <div className='dashboard'>
      <div className='dashboard__content'>
        <div className='dashboard__content-header'>
<h1 className='dashboard__title'> Математика – ключ до розуміння всесвіту
</h1>
<h2 className='dashboard__subtitle'> Реєструйся і покращуй свої знання вже зараз!
```

```

</h2>
</div>
<LessonsCarousel lessons={lessons}/>
</div>
<div className='dashboard__hero'>
  
</div>
</div>
</Wrapper>
);
};

```

```
export default Dashboard;
```

Лістинг файлу Dashboard.scss

```

@use '/src/static/styles/variables.scss' as var;

.dashboard { height: 100%; display: grid;
  grid-template-columns: 3fr 1fr;

  @media screen and (max-width: 980px) { display: flex;
    flex-direction: column; align-items: center;
  }

  &__content { display: flex;

    flex-direction: column; align-content: space-between;
  }

  &__title {
    color: var.$text-primary; font-style: italic;
    font-size: 52px; text-align: center;

    @media screen and (max-width: 720px) { font-size: 32px;
    }
  }

  &__subtitle {
    color: var.$text-primary; font-size: 44px;
    text-align: center;

    @media screen and (max-width: 720px) { font-size: 24px;
    }
  }
}

```

Лістинг файлу LessonCard.jsx

```

import PropTypes from 'prop-types'; import './lessonCard.scss';

const LessonCard = (props) => { const { lesson } = props;
  const { index: lessonPosition, title } = lesson;

  return (
    <div className='lesson-card'>
      <div className='lesson-card__image'>
      </div>
      <p className="lesson-card__title"> Урок {lessonPosition} - {title}
      </p>
    </div>
  );
}

```

```

};

LessonCard.propTypes = { lesson: PropTypes.shape({
  id: PropTypes.number.isRequired, index: PropTypes.number.isRequired, title:
  PropTypes.string.isRequired,
}),
};

LessonCard.defaultProps = { lesson: {}},

};

export default LessonCard;

```

Лістинг файлу lessonCard.scss

```

@use '/src/static/styles/variables.scss' as var;

.lesson-card {
  max-height: 170px;

  &_image { height: 100%;
    border: 2px solid #415053; border-radius: 30px;
    transition: .4s border-color ease-in-out;

    display: flex;
    justify-content: center; align-items: center;

    &:hover {
      border-color: #F9C651;
    }
  }

  &_title {
    text-align: center; color: var.$text-primary;
  }
}

```

Лістинг файлу lessons.js

```

Const App = () => {
  export const lessons = [
    {
      lessonid: 1,
      title: 'Урок 1. Вступ до математики',
      videoUrl: 'https://www.example.com/video1.mp4' content: 'Теорія до першого заняття.',
      questions: [
        {
          text: 'Питання 1',
          choices: ['Відповідь 1', 'Відповідь 2', 'Відповідь 3']
        },
        {
          text: 'Питання 2',
          choices: ['Відповідь 1', 'Відповідь 2', 'Відповідь 3']
        },
        // Таких питань може бути скільки завгодно від вимог викладача
      ],
    }
    // Додаткові об'єкти з інформацією про інші уроки
  ];
}

```



```
return <Lessons lessons={lessonsData}/>
};
```

Лістинг файлу Lessons.jsx

```
import { lessons } from '../../mocks/lessons';

import Wrapper from '../../components/Wrapper/Wrapper'; import LessonCard from
'./components/LessonCard/LessonCard';

import UserCard from './components/UserCard/UserCard'; import './lessons.scss';

const Lessons = () => { return (
  <Wrapper>
    <div className='lessons'>
      <div className="lessons__grid">
        {lessons.map((lesson, index) => (
          <LessonCard key={lesson.id} lesson={{...lesson, index: index + 1 }}/>
        ))}
      </div>
      <div className="lessons__profile">
        <UserCard/>
      </div>
    </div>
  </Wrapper>
);
};
```

Лістинг файлу lessons.scss

```
export default Lessons;
```

```
.lessons { display: flex;
  justify-content: space-between; gap: 48px;
  height: 90%;

  &__grid { width: 100%; display: grid;
    grid-template-columns: repeat(3, 1fr); grid-auto-rows: 200px;
    grid-gap: 1rem; row-gap: 4rem;
  }

  &__profile {
    min-width: 500px;
  }
}
```

Лістинг файлу Login.jsx

```
import { useState } from 'react';

import AccountCircleIcon from '@mui/icons-material/AccountCircle'; import Avatar from
'@mui/material/Avatar';
import Button from '@mui/material/Button';
import CssBaseline from '@mui/material/CssBaseline'; import TextField from
'@mui/material/TextField';
import FormControlLabel from '@mui/material/FormControlLabel'; import Checkbox from
'@mui/material/Checkbox';
import Link from '@mui/material/Link'; import Grid from '@mui/material/Grid'; import
Box from '@mui/material/Box'; import Modal from '@mui/material/Modal';
import LockOutlinedIcon from '@mui/icons-material/LockOutlined'; import Typography from
'@mui/material/Typography';
import Container from '@mui/material/Container'; import ROUTES from
```

```

'../../router/routes'; import './login.scss';

const Login = () => {
  const [open, setOpen] = useState(false);

const handleOpen = () => { setOpen(true);
  };

const handleClose = () => { setOpen(false);
  };

const handleSubmit = (event) => { event.preventDefault();
  const data = new FormData(event.currentTarget); console.log({
    email: data.get('email'), password: data.get('password'),
  });
  };

if (open) { return (
  <Modal
    open={open} onClose={handleClose}
    aria-labelledby="modal-modal-title"
    aria-describedby="modal-modal-description"
  >
    <Container component="main" maxWidth="xs">
      <CssBaseline />
      <Box
        sx={{
          marginTop: 8, display: 'flex',
          flexDirection: 'column',

          alignItems: 'center', bgcolor: '#1B282B', padding: '32px', borderRadius: '15px',
        }}
      >
        <Avatar sx={{ m: 1, bgcolor: 'secondary.main' }}>
          <LockOutlinedIcon />
        </Avatar>
        <Typography component="h1" variant="h5"> Увійти
        </Typography>
        <Box component="form" onSubmit={handleSubmit} noValidate sx={{ mt: 1 }}>
        <TextField margin="normal" required fullWidth id="email"
          label="Email Address" name="email" autoComplete="email" autoFocus
        />
        <TextField margin="normal" required fullWidth name="password" label="Password"
          type="password" id="password"
          autoComplete="current-password"
        />
        <FormControllabel
          control={<Checkbox value="remember" color="primary" />} label="Remember me"
        />
        <Button type="submit" fullWidth
          variant="contained" sx={{ mt: 3, mb: 2 }}
        >
          Sign In
        </Button>
        <Grid container>
          <Grid item xs>
            <Link href="#" variant="body2"> Забули пароль?
            </Link>
          </Grid>

```

```

<Grid item>
<Link href={ROUTES.REGISTER} variant="body2">
{"Створити акаунт"}
</Link>
</Grid>
</Grid>

</Box>
</Box>
</Container>
</Modal>
);
}

return (
<button className='login_button' onClick={handleOpen}> Увійти
  <AccountCircleIcon/>
</button>
);
}

export default Login;

```

Лістинг файлу login.scss

```

@use '/src/static/styles/variables.scss' as var;

.login {
&_button { display: flex;
  align-items: center; gap: 8px;
  color: var.$text-primary; cursor: pointer; background: none; border: none;
  font-size: 16px;
  font-family: 'Montserrat', serif;
}
}

```

Лістинг файлу variables.scss

```

$accent-color: #111E14;
$primary-color: #1B282B;

$text-primary: #F5F5F5;

```

Лістинг файлу Register.jsx

```

import Avatar from '@mui/material/Avatar'; import Button from '@mui/material/Button';
import CssBaseline from '@mui/material/CssBaseline'; import TextField from
 '@mui/material/TextField';
import FormControlLabel from '@mui/material/FormControlLabel'; import Checkbox from
 '@mui/material/Checkbox';
import Link from '@mui/material/Link'; import Grid from '@mui/material/Grid'; import Box
 from '@mui/material/Box';
import LockOutlinedIcon from '@mui/icons-material/LockOutlined'; import Typography from
 '@mui/material/Typography';
import Container from '@mui/material/Container';
import { createTheme, ThemeProvider } from '@mui/material/styles';

import ROUTES from '../../router/routes';

import Copyright from '../../components/Copyright/Copyright'; const defaultTheme =

```

```

createTheme();

const Register = () => {
const handleSubmit = (event) => { event.preventDefault();
  const data = new FormData(event.currentTarget); console.log({
    email: data.get('email'), password: data.get('password'),
  });
};

  return (
    <ThemeProvider theme={defaultTheme}>
    <Container component="main" maxWidth="xs">
    <CssBaseline />
    <Box
      sx={{
        marginTop: 8, display: 'flex',
        flexDirection: 'column', alignItems: 'center',
      }}
    >
    <Avatar sx={{ m: 1, bgcolor: 'secondary.main' }}>
    <LockOutlinedIcon />
    </Avatar>
    <Typography component="h1" variant="h5"> Sign up
    </Typography>
    <Box component="form" noValidate onSubmit={handleSubmit} sx={{ mt: 3 }}>
      <Grid container spacing={2}>
        <Grid item xs={12} sm={6}>
          <TextField autoComplete="given-name" name="Ім'я"
            required fullWidth id="name" label="Ім'я" autoFocus
          />
        </Grid>
        <Grid item xs={12} sm={6}>
          <TextField required fullWidth id="surname" label="Прізвище" name="Прізвище"
            autoComplete="family-name"
          />
        </Grid>
        <Grid item xs={12}>
          <TextField required fullWidth id="email"
            label="Email Address" name="email" autoComplete="email"
          />
        </Grid>
        <Grid item xs={12}>
          <TextField required fullWidth name="password" label="Password" type="password"
            id="password"
            autoComplete="new-password"
          />
        </Grid>
        <Grid item xs={12}>
          <FormControllabel
            control={<Checkbox value="allowExtraEmails" color="primary" />} label="I want to receive
            inspiration, marketing promotions and
            updates via email."
          />
        </Grid>
      </Grid>
    <Button type="submit" fullWidth variant="contained"
      sx={{ mt: 3, mb: 2 }}
    >
    Увійти
  )
}

```

```

</Button>
<Grid container justifyContent="flex-end">
  <Grid item>
<Link href={ROUTES.LOGIN} variant="body2"> Увійти
  </Link>
</Grid>
</Grid>
</Box>
</Box>
<Copyright sx={{ mt: 5 }} />
</Container>
</ThemeProvider>
);
}

```

Лістинг файлу TestCard.jsx

```
export default Register;
```

```
import PropTypes from 'prop-types'; import './testCard.scss';
```

```
const TestCard = (props) => { const { test } = props;
  const { index: testPosition, title } = test;
```

```

  return (
    <div className='test-card'>
      <div className='test-card__image'>
      </div>
    <p className="test-card_title"> Тест {testPosition} - {title}
    </p>
    </div>
  );
};

```

```

TestCard.propTypes = { test: PropTypes.shape({
  id: PropTypes.number.isRequired, index: PropTypes.number.isRequired, title:
  PropTypes.string.isRequired,
}),
};

```

```

TestCard.defaultProps = { test: {}},
};

```

```
export default TestCard;
```

Лістинг файлу testCard.scss

```
@use '/src/static/styles/variables.scss' as var;
```

```

.test-card {
  max-height: 170px;

```

```

&__image { height: 100%;
  border: 2px solid #415053; border-radius: 30px;
  transition: .4s border-color ease-in-out;

```

```

display: flex;
justify-content: center; align-items: center;

```

```

&:hover {
  border-color: #F9C651;
}
}

```

```

&__title {
text-align: center;

color: var.$text-primary;
}
}

```

Лістинг файлу Tests.jsx

```

import { tests } from '../../mocks/tests';

import Wrapper from '../../components/Wrapper/Wrapper'; import TestCard from
'./components/TestCard/TestCard';

import UserCard from '../Lessons/components/UserCard/UserCard'; import './tests.scss';

const Tests = () => { return (
  <Wrapper>
    <div className='tests'>
      <div className="tests__grid">
        {tests.map((test, index) => (
          <TestCard key={test.id} test={{...test, index: index + 1 }}/>
        ))}
      </div>
      <div className="tests__profile">
        <UserCard/>
      </div>
    </div>
  </Wrapper>
);
};

```

Лістинг файлу tests.scss

```
export default Tests;

.tests { display: flex;
  justify-content: space-between; gap: 48px;
  height: 90%;

  &__grid { width: 100%; display: grid;
    grid-template-columns: repeat(3, 1fr); grid-auto-rows: 200px;
    grid-gap: 1rem; row-gap: 4rem;
  }

  &__profile {
    min-width: 500px;
  }
}
```

Лістинг файлу Header.jsx

```
import { NavLink } from 'react-router-dom';

import Login from '../pages/Login/Login'; import ROUTES from '../router/routes';

import './header.scss'; const Header = () => {
  return (
    <header className='header'>
      <nav className='header__nav nav'>
        <ul className="nav-list">
          <div className='header__logo'>
            <img src='/src/assets/Logo.svg' alt='logo' />
          </div>
          <li className="nav-list__item">
            <NavLink to={ROUTES.DASHBOARD}>Головна</NavLink>
          </li>
          <li className="nav-list__item">
            <NavLink to={ROUTES.LESSONS}>Уроки</NavLink>
          </li>
          <li className="nav-list__item">
            <NavLink to={ROUTES.TESTS}>Тести</NavLink>
          </li>
        </ul>
      </nav>
      <div className='header__user'>
        <Login />
      </div>
    </header>
  );
}

export default Header;
```

Лістинг файлу header.scss

```
@use '/src/static/styles/variables.scss' as var;

.header { position: sticky; top: 0;
  padding: 8px 16px;

  background-color: var.$accent-color; display: flex;
  align-items: center;
  justify-content: space-between;
```

```

&_user { display: flex;
  align-items: center; gap: 16px;
  a {
    text-decoration: none;

    color: var.$text-primary;
  }
}

.nav-list { display: flex;
  align-items: center; gap: 32px;

  &__item {
    list-style: none;

    a {
      text-decoration: none; color: var.$text-primary;
    }
  }
}

```

Лістинг файлу Report.js

```

import ResizableBox from "../ResizableBox";
import React, { useEffect, useState } from "react"; import Login from './Login';
import "firebase/storage";
import { Chart } from "react-charts"; import firebase from "firebase/app"; import
"firebase/firestore";
import { Pie } from "react-chartjs-2";

export default function FormReport() {
  const [userAuthenticated, setUserAuthenticated] = useState(false); const storage =
  firebase.storage();
  const [data, setData] = useState([]);
  const [averageResult, setAverageResult] = useState(0); const [originality,
  setOriginality] = useState(0); const [learnProgress, setLearnProgress] = useState(0);
  const [redirect, setRedirect] = useState(false);

  useEffect(() => {
const unsubscribe = firebase.auth().onAuthStateChanged((user) => { if (user) {
  setUserAuthenticated(true);

  getDataFromFirestore().then((resultData) => {
    const formattedData = formatDataForChart(resultData); setData(formattedData);

    const avgResult = calculateAverageResult(resultData); setAverageResult(avgResult);

    const originalityPercentage = checkOriginality(avgResult);
    setOriginality(originalityPercentage);

    const learnProgress = resultData.user_data.learn_progress;
    setLearnProgress(learnProgress);

    generateReportFile().then((reportFilename) => {
      const [reportFileUrl, setReportFileUrl] = useState(null);

      useEffect(() => {
        if (reportFileUrl) { window.location.href = reportFileUrl;

```



```

    }
    }, [reportFileUrl]);

    const generateReportFile = async () => {
    const reportFilename = await generateReportFilename();

    const reportFileRef = storage.ref().child(reportFilename); await
    reportFileRef.putString("file content");

    const reportFileUrl = await reportFileRef.getDownloadURL();
    setReportFileUrl(reportFileUrl);
    };    });
    });
    } else {
    setUserAuthenticated(false);
    }
    });

const getDataFromFirestore = async () => { const firestore = firebase.firestore();
    const userCollectionRef = firestore.collection("users"); const resultsCollectionRef =
    firestore.collection("results");

    const user_id = "1";

    const userSnapshot = await userCollectionRef.doc(user_id).get(); const user_data =
    userSnapshot.data();

    const resultsQuerySnapshot = await resultsCollectionRef
    .where("user_id", "==", user_id)
    .get();
    const results_data = []; resultsQuerySnapshot.forEach((doc) => {
    const result = doc.data(); results_data.push(result);
    });

    return { user_data, results_data };
    };

const formatDataForChart = (resultData) => { const { user_data, results_data } =
    resultData;

    const formattedData = [];

    const user_name = `${user_data.surname} ${user_data.name}`; formattedData.push({
    primary: "User Name", secondary: user_name });

    results_data.forEach((result, index) => { const test_number = index + 1;

    const result_value = result.result;
    const avg_result_value = result.avg_result; formattedData.push({
    primary: "Test ${test_number}", secondary: result_value, tertiary: avg_result_value,
    });
    });

    return formattedData;
    };

const calculateAverageResult = (resultData) => { const { results_data } = resultData;
    let totalResult = 0;

    results_data.forEach((result) => { totalResult += result.result;

```

```

});
const avgResult = totalResult / results_data.length;

return avgResult;
};

const checkOriginality = async (avgResult) => { const firestore = firebase.firestore();
const resultsCollectionRef = firestore.collection("results"); const usersCollectionRef =
firestore.collection("users");

const resultsQuerySnapshot = await resultsCollectionRef
.where("result", "==", avgResult)
.get();

const resultsCount = resultsQuerySnapshot.size;

const usersQuerySnapshot = await usersCollectionRef.get(); const totalUsers =
usersQuerySnapshot.size;

const percentage = (resultsCount / totalUsers) * 100; return percentage;
};

const getCurrentDate = () => { const currentDate = new Date();
const year = currentDate.getFullYear();
const month = (currentDate.getMonth() + 1).toString().padStart(2, "0"); const day =
currentDate.getDate().toString().padStart(2, "0");
return `${day}-${month}-${year}`;
};

const generateReportFilename = async () => { const firestore = firebase.firestore();
const reportsCollectionRef = firestore.collection("reports");

const reportNumberSnapshot = await reportsCollectionRef.doc("report_number").get();

const reportNumber = reportNumberSnapshot.data().number;

await reportsCollectionRef.doc("report_number").update({ number: reportNumber + 1
});

const currentDate = getCurrentDate();
const reportFilename = `report_${reportNumber}.pdf`;

return reportFilename;
const generateReportFile = async () => {
const reportFilename = await generateReportFilename(); const reportFileRef =
storage.ref().child(reportFilename); const reportFileUrl = await
reportFileRef.getDownloadURL();

return reportFileUrl;
};
};

const chartData = {
labels: ["Середній результат користувача ", "Такий ж результат має користувачів
"],
datasets: [
{
data: [averageResult, originality], backgroundColor: ["#1FDB1B", "#2494AA"],
},
],
};
};

```

```

if (!userAuthenticated) { return <Login />;
}

if (redirect) {
return <Redirect to={"localhost:5173/reports/forms/${reportFilename}" />;
}

return (
<div>
<div style={{ display: "flex", justifyContent: "center" }}>
<Pie data={chartData} options={chartOptions} />
</div>
<ResizableBox>
    <Chart data={data} primaryAxis={primaryAxis} secondaryAxes={secondaryAxes}
series={{ type: "bar" }} />
</ResizableBox>
</div>
);
}

```

Лістинг файлу LessonsCarousel.jsx

```
import PropTypes from 'prop-types'; import './lessonsCarousel.scss';
```

```
const LessonsCarousel = (props) => { const { lessons } = props;
```

```

return (
  <div className='lessons-carousel'>
<h3 className='lessons-carousel__title'> Каталог навчального матеріалу
</h3>
<button className='lessons-carousel__left-arrow'>

</button>
<ul className='lessons-carousel__list'>
  {lessons.map((lesson) => (
    <li className='lessons-carousel__item' key={lesson.id}>
      {lesson.title}
    </li>
  ))}
</ul>

  <button className='lessons-carousel__right-arrow'>
    
  </button>
</div>
);
}

```

```

LessonsCarousel.propTypes = { lessons: PropTypes.arrayOf(
  PropTypes.shape({
    id: PropTypes.number.isRequired, title: PropTypes.string.isRequired,
  })
),
};

```

```

LessonsCarousel.defaultProps = { lessons: [],
};

```

```
export default LessonsCarousel;
```

Лістинг файлу lessonsCarousel.scss

```

@use '/src/static/styles/variables.scss' as var;

$primary-color: #F9C651;
$secondary-color: #363030;

.lessons-carousel { position: relative; max-width: 820px; overflow: hidden; margin: 0
  auto;

  display: flex;
  flex-direction: column; align-items: center;

  &__title {
    color: $primary-color;

    font-size: 32px;
  }

  &__left-arrow, &__right-arrow {
    position: absolute; top: 50%;
    transform: translateY(50%);

    background: none; border: none; cursor: pointer;
  }

  &__left-arrow { left: 0;
  }

  &__right-arrow { right: -5px;
    transform: translateY(30%) rotate(180deg);
  }

  &__list { display: flex; gap: 80px;
  }

  &__item {
    list-style: none;

    min-width: 300px; height: 200px;

    border: 5px solid $primary-color; border-radius: 30px;
    background-color: $secondary-color;

    display: flex;
    align-items: center; justify-content: center; color: var.$text-primary;
  }
}

```

Лістинг файлу Footer.jsx

```

import Copyright from '../Copyright/Copyright'; import './footer.scss';

const Footer = () => { return (
  <footer className="footer">
    <div className="container">
      <Copyright/>
    </div>

    </footer>
  );
}

```

```
export default Footer;
```

Лістинг файлу footer.scss

```
@use '/src/static/styles/variables.scss' as var;
```

```
.footer {
  min-height: 40px;

  display: flex;
  justify-content: center; align-items: center;

  background-color: var.$accent-color; color: #fff;

  position: fixed; bottom: 0;
  width: 100%;
}
```

Лістинг файлу Copyright.js

```
import { Link } from 'react-router-dom'; import ROUTES from '../router/routes';
```

```
import './copyright.scss';
```

```
const Copyright = () => { return (
  <p className='copyright'>
    {'Copyright © '}
  </p>
);
```

```
<Link color="inherit" href={ROUTES.DASHBOARD}> PhysExp
</Link>{' '}
```

```
{new Date().getFullYear()}
{'.'}
</p>
);
};
```

Лістинг файлу copyright.scss

```
export default Copyright;
```

```
.copyright {
  font-size: 20px; text-align: center;
```

```
a {
  color: white;
  text-decoration: none;
}
}
```

Лістинг файлу Wrapper.jsx

```
import PropTypes from 'prop-types';
```

```
import Header from '../Header/Header'; import Footer from '../Footer/Footer';
```

```
import './wrapper.scss'; const Wrapper = (props) => {
  const { children } = props;
```

```
  return (
    <div className="wrapper">
      <Header/>
      <main className='wrapper__content'>
```

```
{children}
```

```
</main>  
<Footer/>
```

```
</div>  
);  
};
```

```
Wrapper.propTypes = {  
  children: PropTypes.node.isRequired  
};
```

```
export default Wrapper;
```

Лістинг файлу wrapper.scss

```
@use '/src/static/styles/variables.scss' as var;
```

```
.wrapper { height: 100vh;  
  position: relative; display: flex;  
  flex-direction: column;  
  font-family: 'Montserrat', serif;
```

```
&__content { flex: 1 1 auto; padding: 32px;  
  background-color: var.$primary-color;  
}
```

Додаток Г. Ілюстративна частина

ІЛЮСТРАТИВНА ЧАСТИНА
РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ НАВЧАЛЬНОЇ ВЕБ-СИСТЕМИ ДЛЯ
ВИВЧЕННЯ МАТЕМАТИКИ

Розробка програмних засобів веб-системи для вивчення математики

Студент гр. 4ПІ-206:

Закасовський Б. І.

Керівник: к.т.н, доцент

Черноволик Г. О.

Активация Windows
Перейдіть до розділу

Рисунок Г.1 – Слайд «Тема роботи»

Актуальність

Математика має широке застосування і значимість у розвитку логічного та критичного мислення тому розробка навчальної веб-системи є актуальною для поліпшення якості навчання, в тому числі й дистанційного. Це сприятиме зростанню рівня математичної освіти серед молоді.

Рисунок Г.2 – Слайд «Актуальність роботи»

Мета

Метою дослідження є підвищення ефективності процесу вивчення математики за рахунок використання розробленої веб-системи, яка дозволяє систематизувати навчальний матеріал, забезпечує проведення тестового контролю, візуалізує ілюстративний матеріал, що підвищує інтерактивність навчального процесу та сприяє зацікавленості учнів вивченням математики.

Рисунок Г.3 – Слайд «Мета роботи»

Об'єкт

Процес розробки навчальної веб-системи для вивчення математики.

Предмет

Методи та засоби реалізації навчальної веб-системи для вивчення математики

Новизна отриманих результатів

1. Подальшого розвитку набув метод формування і відображення прогресу навчання, який, на відміну від відомих, надає можливість сформулювати звіт з наведеною успішністю у вигляді діаграм, що підвищує інформативність візуалізованих даних та підвищує мотиваційну складову навчального процесу.

2. Подальшого розвитку набув метод інтерактивного вивчення матеріалу, який, на відміну від існуючих, базується на використанні розширеного функціоналу і реалізації візуальних методів графічного подання навчальних матеріалів, підвищує інформативність навчальних матеріалів та сприяє підвищенню ефективності навчального процесу в цілому.

Рисунок Г.4 – Слайд «Об'єкт та предмет роботи»
Рисунок Г.5 – Слайд «Новизна отриманих результатів»

Практичне значення

Практична цінність одержаних результатів полягає в тому, що розроблена у бакалаврській дипломній роботі веб-система може використовуватись у навчальному процесі для вивчення математики та забезпечує систематизацію наведення освітнього матеріалу, проведення тестового контролю, відстеження успішності та сприятиме зацікавленості учнів вивченням математики.

Рисунок Г.6 – Слайд «Практичне значення»

Порівняльний аналіз аналогів

Характеристики / Функції	Matific	Khan Academy	Mathema	OpenStax	MathExp
Інтерактивність	-	+	-	-	+
Підтримується українська мова	+	-	+	-	+
Тестування	-	-	+	-	+
Безкоштовність	+	+	+	+	+
Сума	2	2	3	1	4

Рисунок Г.7 – Слайд «Порівняльний аналіз аналогів»

Алгоритм роботи системи

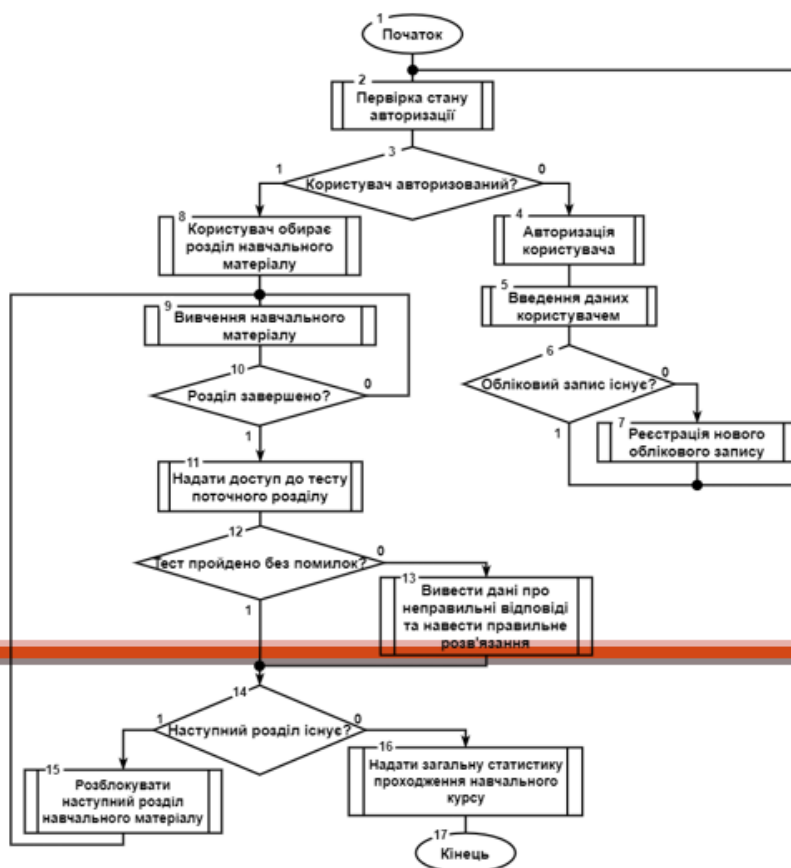
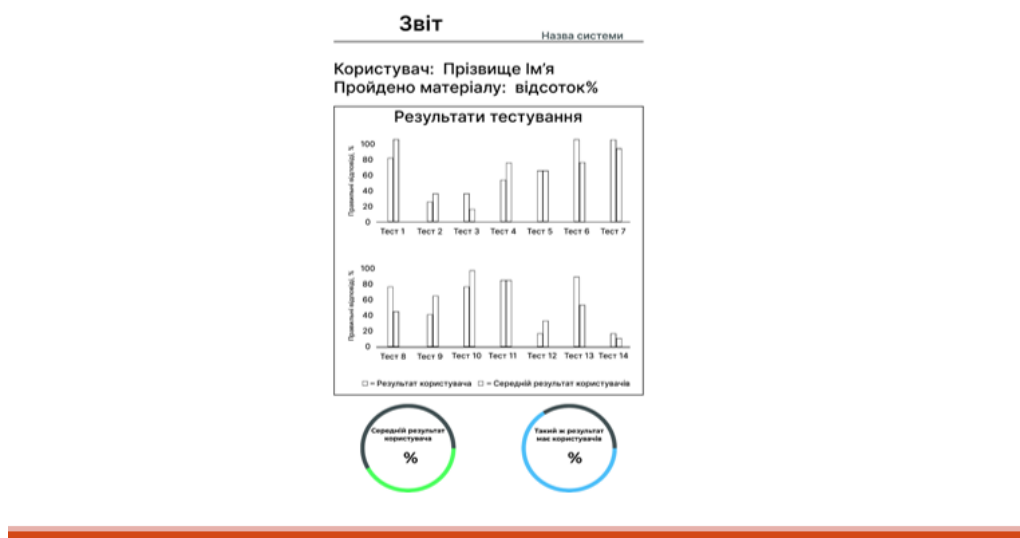


Рисунок Г.8 – Слайд «Алгоритм роботи системи»



Інтерфейс звіту

Рисунок Г.9 – Слайд «Інтерфейс звіту проходження тесту»

Висновки

- розроблено алгоритми основних програмних компонентів застосунку;
- розроблено функціонал навчальної веб-системи;
- розроблено інтерфейс користувача для веб-сервісу;
- описано програмну реалізацію основних модулів;
- забезпечено підтримку української мови та доступність веб-системи на різних платформах і пристроях, включаючи мобільні пристрої;
- проведено тестування програми.

Рисунок Г.10 – Слайд «Висновки»