

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК  
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Інтерактивний додаток з вивчення чисельного методу Ньютона»

за спеціальністю 122 «Комп'ютерні науки»,  
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студентка групи ІТ-62 Литвин Ірина Вікторівна

Кваліфікаційна робота бакалавра  
захищена на засіданні ЕК  
з оцінкою \_\_\_\_\_

«\_\_» \_\_\_\_\_ 2020 р.

Науковий керівник \_\_\_\_\_

(підпис)

к.т.н., доц., Чибіряк Я. І.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії \_\_\_\_\_

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів  
без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Суми-2020

Сумський державний університет  
 Факультет електроніки та інформаційних технологій  
 Кафедра комп'ютерних наук  
 Секція інформаційних технологій проектування  
 Спеціальність 122 «Комп'ютерні науки»  
 Освітньо-професійна програма «Інформаційні технології проектування»

**ЗАТВЕРДЖУЮ**

Зав. секцією ІТП

\_\_\_\_\_ В. В. Шендрик  
 «\_\_» \_\_\_\_\_ 2020 р.

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ**

*Литвин Ірина Вікторівна*

- 1 Тема роботи** Інтерактивний додаток з вивчення методу Ньютона
- керівник роботи** Чибіряк Яна Іванівна, к.т.н., доцент,
- затверджені наказом по університету від «14» травня 2020 р. № 0576-III
- 2 Строк подання студентом роботи** «1» червня 2020 р.
- 3 Вхідні дані до роботи** технічне завдання на розробку інтерактивного додатку з вивчення чисельного методу Ньютона, сценарій
- 4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)** аналіз предметної області, моделювання web-тренажера, проектування моделі бази даних, архітектура програмного продукту, розробка web-тренажера з вивчення чисельного методу Ньютона
- 5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** актуальність роботи, мета та задачі дипломного проекту, дослідження web-тренажерів аналогів, функціональні вимоги до web-тренажеру, засоби реалізації, розробка бази даних, практична реалізація, тестування додатку

**6. Консультанти розділів роботи:**

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7.Дата видачі завдання 01.10.2019

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	10.11.19 – 17.11.19	
2	Огляд останніх досліджень	19.11.19 – 23.12.19	
3	Постановка задачі	25.12.19 – 29.12.19	
4	Функціональні вимоги до ПП	03.01.20 – 08.01.20	
5	Розробка інтерфейсу web-тренажера	10.01.20 – 01.02.20	
6	Проектування БД програмного продукту	03.02.20 – 08.02.20	
7	Наповнення web-тренажера	10.02.20 – 29.02.20	
8	Розробка калькулятора	02.03.20 – 11.03.20	
9	Розробка всіх рівнів web-тренажеру	12.03.20 – 12.04.20	
10	Розробка зворотнього зв'язку	13.04.20 – 22.04.20	
11	Тестування додатку	23.04.20 – 10.05.20	
12	Оформлення документації	15.05.20 – 01.06.20	
13	Архівація програмного продукту	17.05.20 – 23.05.20	
14	Інтегрування web-тренажера до elearning	29.05.20 – 01.06.20	

Студент

\_\_\_\_\_

(підпис)

Литвин І.В.

Керівник роботи

\_\_\_\_\_

(підпис)

к.т.н., доц. Чибіряк Я.І.

## РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Інтерактивний додаток з вивчення методу Ньютона».

Пояснювальна записка містить вступ, 3 розділи, висновки, список використаних джерел, додатки. Вона включає 93 сторінок, 7 таблиці, 47 ілюстрацій, та 20 джерел.

Кваліфікаційну роботу бакалавра присвячено розробці та реалізації інтерактивного додатку з вивчення чисельного методу Ньютона, як засобу вирішення систем нелінійних алгебраїчних рівнянь.

В першому розділі досліджено актуальність роботи, наведено аналіз програмних продуктів – аналогів. Також цей розділ містить інформацію про мету та задачі проекту, засоби реалізації.

Другий розділ включає в себе моделювання та проектування процесу розробки. В цьому розділі зображено модель бази даних у вигляді ER-діаграми. Виконано структурно-функціональне моделювання, в результаті якого були змодельовані такі діаграми як: діаграма варіантів використання, контекстна діаграма та IDEF0.

Третій розділ описує розробку web-тренажера. Наведена архітектура програмного додатку та програмна реалізація. Зображений приклад використання програмного продукту.

Результатом проведеної роботи є розроблений інтерактивний додаток з вивчення чисельного методу Ньютона. Практичне значення роботи полягає в удосконаленні навчального процесу та підвищенні рівня засвоєння знань студентами при вивченні теми «Застосування чисельного методу Ньютона для вирішення систем нелінійних рівнянь».

Ключові слова: web-додаток, чисельний метод Ньютона, сценарій, калькулятор, спливаючі підказки, web-тренажер, дистанційне навчання elearning, архітектура web-тренажера.

## ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Дослідження актуальності роботи .....	8
1.2 Аналіз програмних продуктів – аналогів .....	9
1.3. Вимоги до тренажеру .....	16
1.4. Постановка задачі .....	18
1.5. Вибір засобів реалізації.....	19
2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІНТЕРАКТИВНОГО ДОДАТКУ.....	21
2.1. Моделювання діаграми варіантів використання .....	21
2.2. Моделювання роботи інтерактивного додатку в IDEF0.....	23
2.3. Проектування моделі бази даних web-тренажера з вивчення чисельного методу Ньютона.....	26
2.4. Концептуальна модель web-тренажера .....	30
3 РОЗРОБКА WEB-ТРЕНАЖЕРА З ВИВЧЕННЯ ЧИСЕЛЬНОГО МЕТОДУ НЬЮТОНА.....	32
3.1. Архітектура програмного продукту .....	32
3.2. Програмна реалізація.....	33
3.2.1. Розробка клієнтської частини.....	35
3.2.2. Розробка серверної частини .....	37
3.3. Використання програмного додатку.....	40
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	52
Додаток А.....	54
Технічне завдання .....	54
Додаток Б.....	62
Планування робіт.....	62
Додаток В .....	76

## ВСТУП

Інформаційні технології активно задіяні в усіх життєвих процесах сучасного суспільства. Практика показує, що традиційні засоби навчання не забезпечують необхідної варіативності, гнучкості та динамічності системи освіти, її адаптивності до соціальних умов та сучасних вимог роботодавців. Одним із способів інтенсифікації навчального процесу та розвитку якісно нової методики навчання у закладах вищої освіти є впровадження дистанційної форми навчання. У зв'язку з цим виникає потреба у створенні навчального матеріалу, необхідного для віддаленого доступу, а саме: віртуальних лабораторних робіт, практикумів, мультимедійних демонстрацій. Особлива увага приділяється розробці інтерактивних віртуальних тренажерів, призначених для розвитку необхідних навичок при вирішенні практичних задач.

Дану роботу присвячено розробці та реалізації інтерактивного web-додатку з вивчення чисельного методу Ньютона, як засобу вирішення систем нелінійних алгебраїчних рівнянь.

Об'єктом дослідження дипломного проектування є: розробка web-тренажера та інтегрування до університетської системи електронного навчання elearning.

Предметом дослідження є інтерактивний додаток з вивчення теми «Застосування чисельного методу Ньютона для вирішення систем нелінійних рівнянь».

Метою кваліфікаційної роботи бакалавра є створення інтерактивного додатку для удосконалення навчального процесу завдяки практико-орієнтованому підходу та підвищення рівня засвоєння знань студентами при вивченні теми «Застосування чисельного методу Ньютона для вирішення систем нелінійних рівнянь».

Для досягнення мети кваліфікаційної роботи потрібно вирішити такі задачі:

- аналіз предметної сфери;
- формування функціональних вимог до web-тренажера;
- розробка бази даних;

- розробка та реалізація web-додатку;
- тестування роботи тренажеру;
- інтеграція тренажеру до університетської системи elearning.

Результатом проекту буде інтерактивний додаток, впроваджений в навчальний процес, який дозволить краще засвоїти отримані теоретичні знання та підвищити якість навчального процесу завдяки практико-орієнтованому підходу і підвищенню мотивації у навчанні.

# 1 АНАЛІЗ ПРДМЕТНОЇ ОБЛАСТІ

## 1.1 Дослідження актуальності роботи

У всіх навчальних сферах, все частіше і частіше, впроваджуються ІТ-технології. Враховуючи це, web-тренажерні системи все активніше йдуть у використання.

З розвитком веб-технологій на просторах Інтернету з'явилась велика кількість веб-тренажерів та різних серверів, які були створенні для реалізації будь-яких потреб користувача. На сьогоднішній день велика кількість веб-тренажерів допомагають користувачам підвищити рівень знань, допомагають активізувати розумову діяльність. У зв'язку з впровадженням дистанційної форми навчання виникає потреба у створенні навчального матеріалу, необхідного для віддаленого доступу, а саме: віртуальних лабораторних робіт, мультимедійних демонстрацій та ін. Особлива увага приділяється розробці інтерактивних віртуальних тренажерів, призначених для розвитку необхідних навичок при вирішенні практичних задач.

Тому створення web-тренажера з вивчення методу Ньютона є найбільш ефективним способом для вдосконалення отриманих навичок та актуальним завданням сьогодення. Інтерактивність даного тренажеру забезпечується наявністю інструментів динамічного відображення алгоритму, інструментів навігації, спливаючих підказок, можливістю побудови графіків, використанню калькулятора, зворотного зв'язку з викладачем.

Таким чином впровадження розробленого web-додатку у навчальний процес дозволить краще засвоїти отримані теоретичні знання та підвищити якість навчального процесу завдяки практико-орієнтованому підходу і підвищенню мотивації у навчанні.



## 1.2 Аналіз програмних продуктів – аналогів

У наш час використання web-тренажерів набуває великої популярності, особливо в закладах вищої освіти. Однак є досить багато web-тренажерів, які не забезпечують наявністю деяких інструментів навігації, спливаючих підказок, можливістю побудови графіків, використанню калькулятора, а також не надають можливості звернутися за допомогою.

Тому перед початком розробки web-тренажера було проведено аналіз вже існуючих подібних аналогів. Аналізуючи інші програмні продукти, було визначено сильні та слабкі сторони тренажерів.

Розглянемо аналоги тренажерів Інституту хімічних технологій (м.Рубіжне) Східноукраїнського національного університету імені Володимира Даля на сайті «Диференціальні рівняння» (рис. 1.1).

Доступ до тренажерів знаходяться за посиланням:  
<http://difur.in.ua/studentam/trenazheri/>.

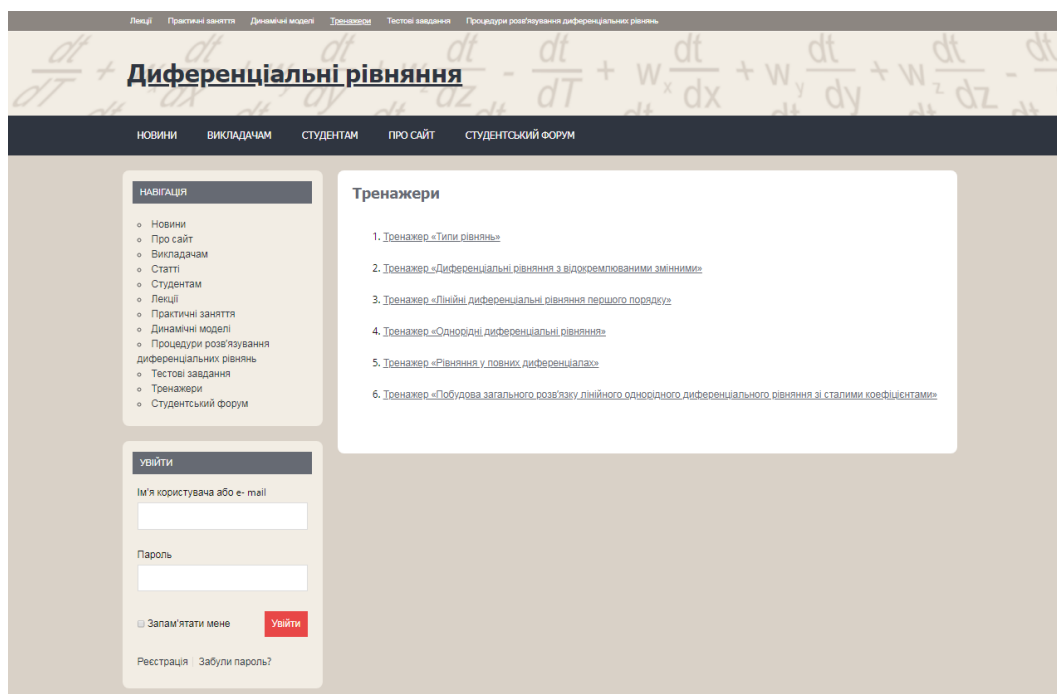


Рисунок 1.1 – Сторінка сайту «Диференціальні рівняння» розділ «Тренажери»

На сайті «Диференціальні рівняння» в панелі навігації знаходяться вбудовані тренажери. В розділі «Тренажери» користувач може обирати будь-який із запропонованих тренажерів та проходити його. В тренажері використовується онлайн-калькулятор для обчислення рівнянь (рис. 1.5). Також в калькуляторі містить теорія та опис функцій (рис. 1.6).

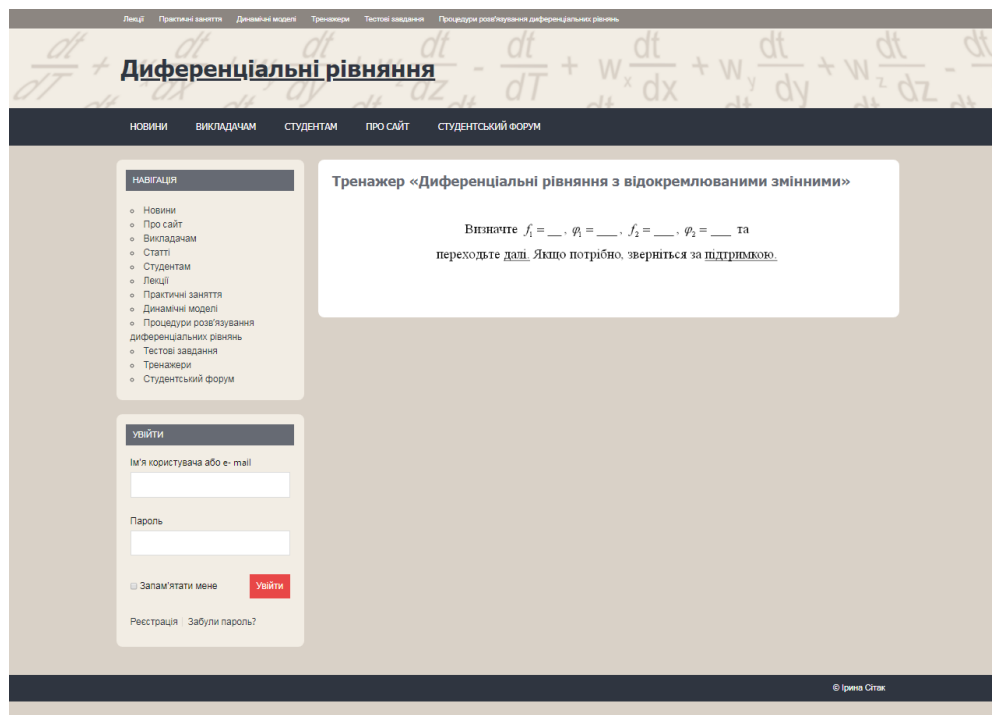


Рисунок 1.2 - Умова задачі в тренажері з теми «Диференціальні рівняння з відокремленими змінними»

Даний тренажер має спливаючі підказки (підтримка) (рис. 13), які допомагають користувачам знайти шлях до правильного рішення та зауваження (рис. 1,4) – повідомляє про хід неправильності дій. Недоліком тренажера є те що користувач не має можливості вводити свої результати та перевіряти їх.

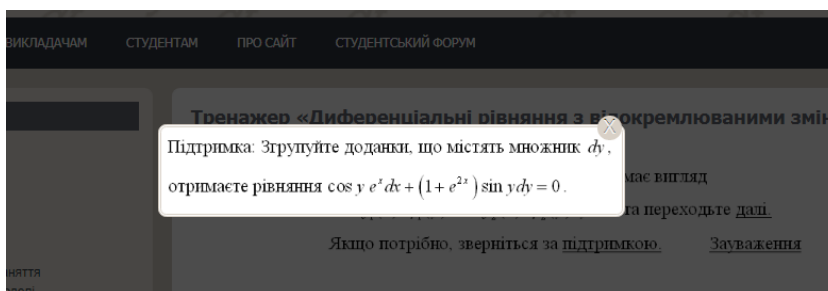


Рисунок 1.3 – Вікно підтримки

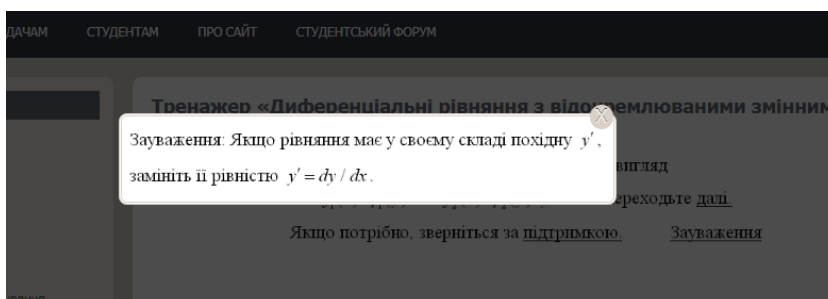


Рисунок 1.4 – Вікно зауваження

**MatCabi.net**  
Кабинет математики онлайн

Математика онлайн    Функции    Теория    **Контрольные/курсовые/рефераты**

**Решение математики онлайн**

$\frac{d}{dx}(x^a) = ax^{a-1}$        $\lim_{x \rightarrow 0} \frac{\sin(x)}{x} = 1$   
 $\sum_{n=0}^{\infty} \frac{1}{n!} = e$        $\int \cos(x) dx = \sin(x) + \text{constant}$

✓ Пошаговое решение на сайте [step.matcabi.net](http://step.matcabi.net)

[Вконтакте](#)    [Одноклассники](#)    [Facebook](#)    [Twitter](#)    [Мой мир](#)    [Google+](#)

Уравнения  
 Неравенства  
 Матрицы  
 Пределы  
 Производная  
 Сумма ряда  
 Неопределенный интеграл  
 Определенный интеграл  
 Дифференциальные уравнения

8003  
 1130  
 1100

Copyright 2010 – 2020, [www.matcabi.net](http://www.matcabi.net)      E-mail: [info@www.matcabi.net](mailto:info@www.matcabi.net)

Рисунок 1.5 – Онлайн-калькулятор

**MatCabi.net**  
Кабинет математики онлайн

Математика онлайн    **Функции**    Теория    Контрольные/курсовые/рефераты

## Константы и Функции

**e** - основание натурального логарифма с приближенным числовым значением 2.71828...

**pi** - число, имеющее значение 3.14159... и равное отношению длины окружности к ее диаметру

**i** - представляет мнимую единицу,  $\sqrt{-1}$

**Degree** - число радиан в одном градусе, которое имеет числовое значение  $\pi/180$

**EulerGamma** - постоянная Эйлера с числовым значением 0.577216...

**GoldenRatio** - константа со значением  $(1+\sqrt{5})/2$ , определяющая деление отрезка по правилу золотого сечения

Элементарные функции:

**abs(x)** - модуль значения x,  $|x|$

**sqrt(x)** - квадратный корень значения x,  $\sqrt{x}$

**x^y** - x в степени y,  $x^y$

**e^x=exp(x)** - экспонента значения x,  $e^x$

**log(a,b)** - логарифм значения b по основанию a,  $\text{Log}_a(b)$

**log(x)** - натуральный логарифм значения x,  $\text{Log}_e(x)$

**dilog(x)** - дилогарифм значения x,  $\text{Li}_2(x)$

**n!** - факториал числа n, равный  $n \times (n-1) \times \dots \times 3 \times 2 \times 1$ , причем  $0!=1$  и  $1!=1$

**n!!** - двойной факториал числа n, равный  $n \times (n-2) \times (n-4) \times \dots$

Тригонометрические функции:

**sin(x)** - синус значения x

**cos(x)** - косинус значения x

**tan(x)** - тангенс значения x

**cot(x)** - котангенс значения x

**sec(x)** - секанс значения x,  $\sec(x)=1/\cos(x)$

**csc(x)** - косеканс значения x,  $\csc(x)=1/\sin(x)$

Обратные тригонометрические функции:

Рисунок 1.6 - Опис функцій

Розглянемо другий аналог тренажера з теми «Градiєнтний метод» (рис. 1.7–1.11).

В тренажері з теми «Градiєнтний метод» немає інструментів навігації, спливаючих підказок, калькулятора, а також не надають можливості звернутися за допомогою.

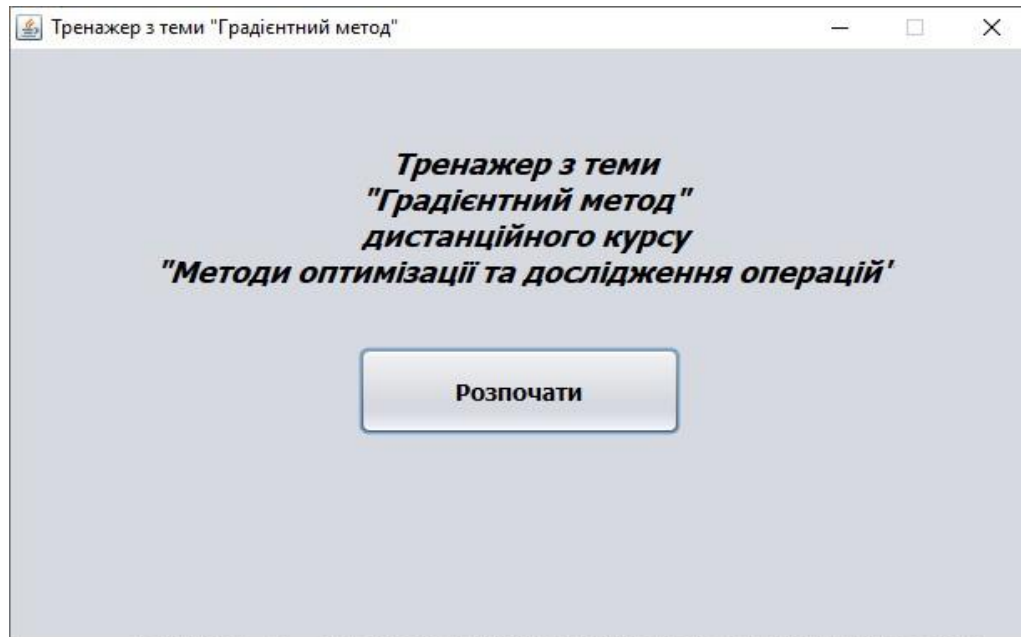


Рисунок 1.7 – Стартова сторінка тренажера з теми «Градiєнтний метод»

Спочатку виводиться умова задачі, а потім завдання (рис. 1.8).

Завдання пропонують як вибір однієї з наведених відповідей (рис. 1.9), так і заповнення комірок (рис. 1.10).

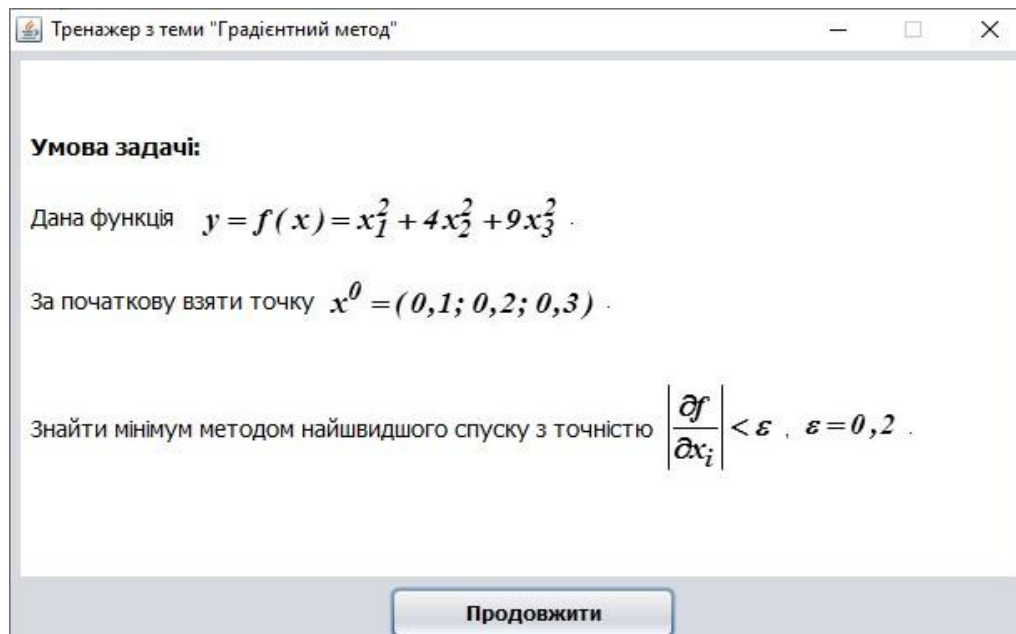


Рисунок 1.8 – Умова задачі в тренажері з теми «Переставні многогранники»

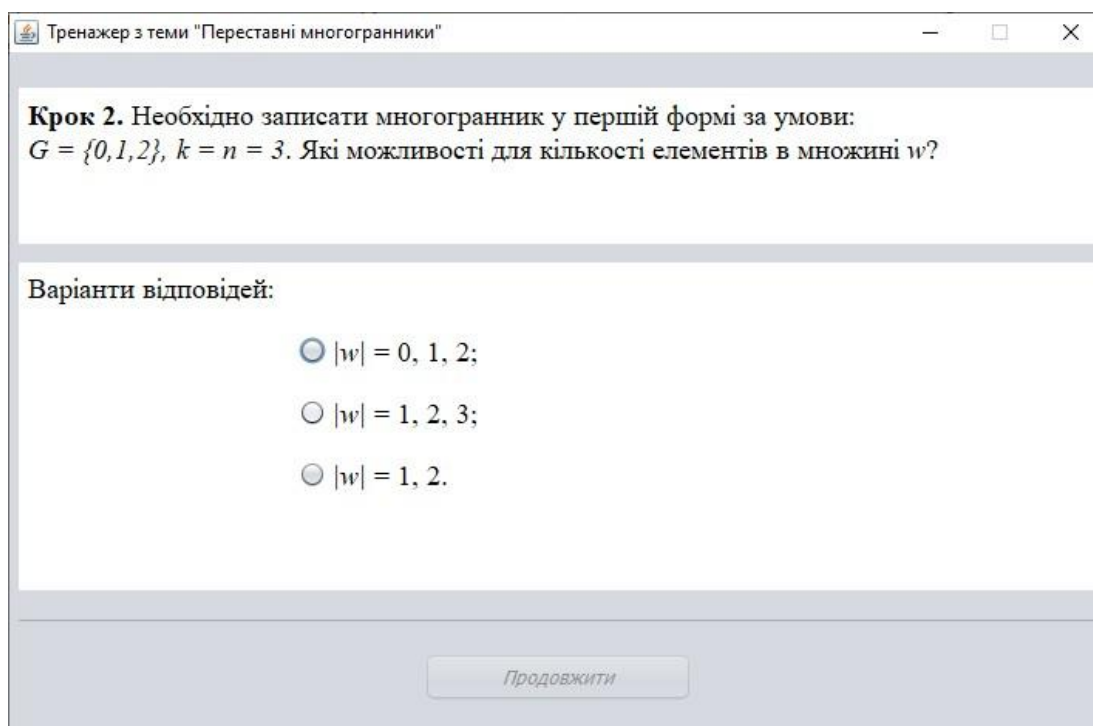


Рисунок 1.9 – Завдання з вибором відповіді в тренажері з теми «Переставні многогранники»

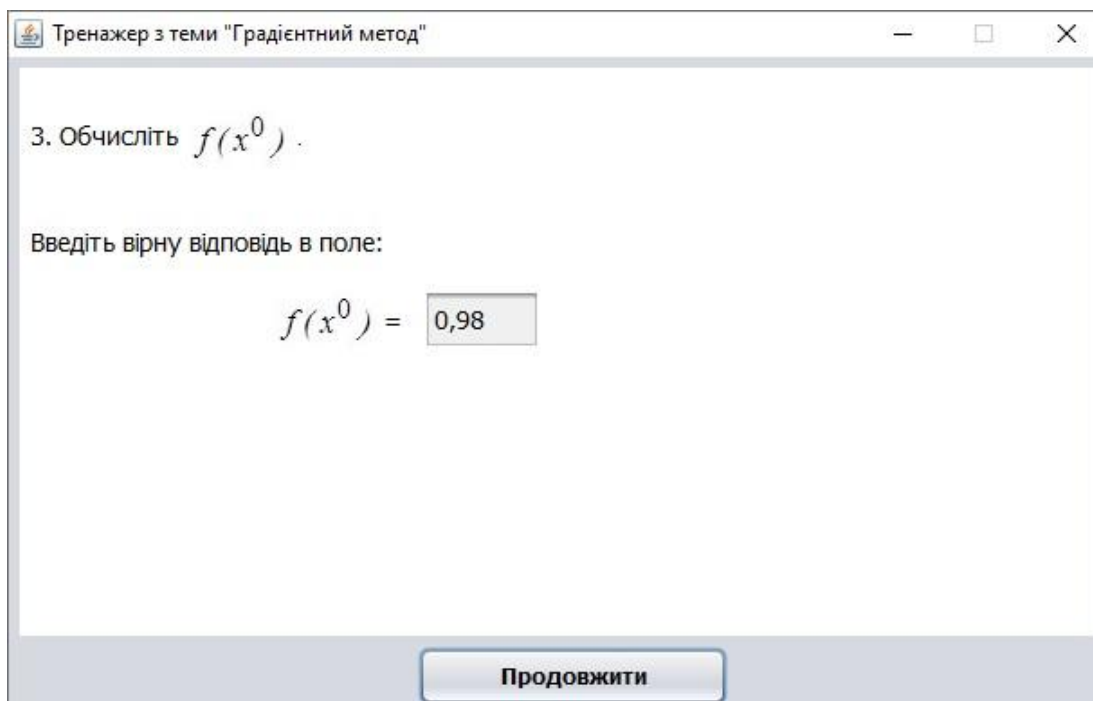


Рисунок 1.10 – Завдання з заповненням комірки в тренажері з теми «Переставні многогранники»

Як висновок відображається повідомлення про завершення роботи тренажера (рис. 1.11).

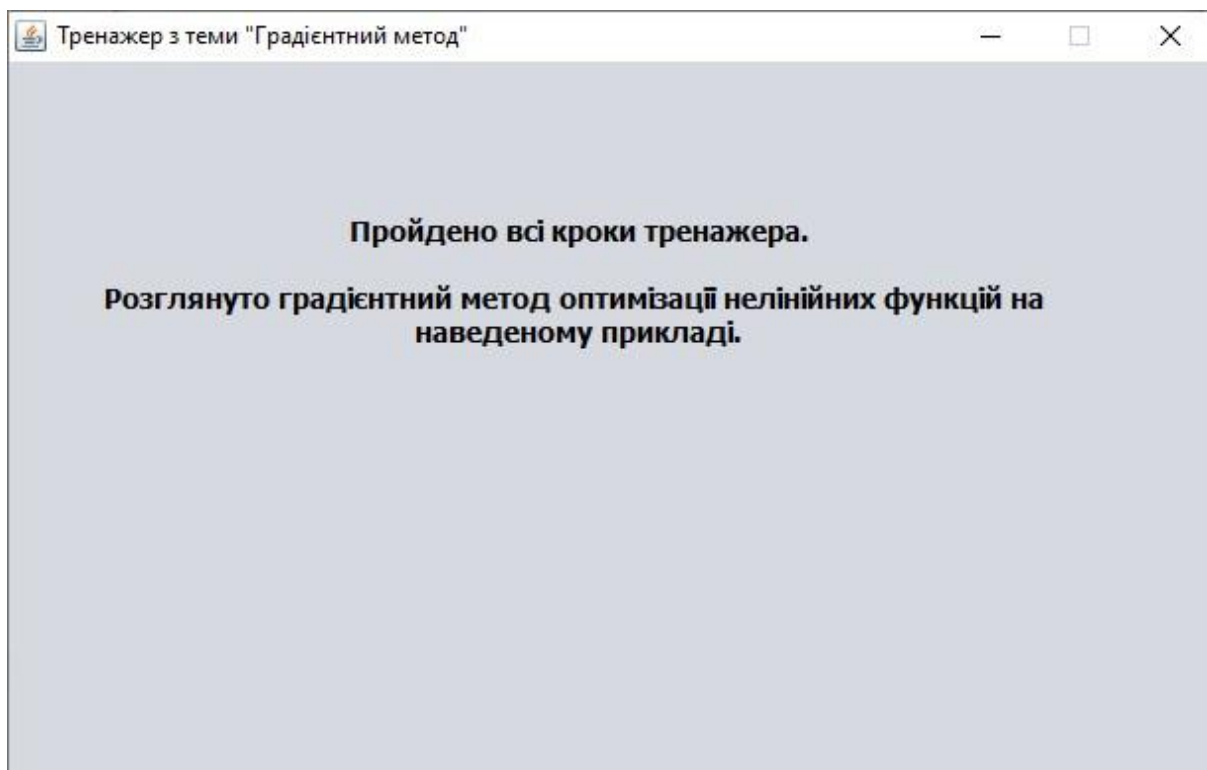


Рисунок 1.11 – Повідомлення про завершення роботи тренажера з теми «Переставні многогранники»

Проаналізувавши web-тренажери в пошуковій системі, які представляються користувачеві було створено порівняльну таблицю та проведено порівняльний аналіз (табл.1.1) аналогів та власного web-тренажера за сформованими критеріями.

Таблиця 1.1 – Порівняльний аналіз розглянутих web-тренажерів

Критерії	Тренажер «Диференціальні рівняння»	Тренажера з теми «Градiєнтний метод»	Тренажер з вивчення методу Ньютона
Наявність панелі навігації	-	-	+
Можливість зворотнього зв'язку	-	-	+

Наявність калькулятора	+	-	+
Привабливий дизайн	+	-	+
Можливість збереження результатів до БД	-	-	+
Наявність спливаючих підказок	+	-	+
Підтримка багатомовності	-	-	+
Можливість розрахунку оцінки	-	-	+
Блок реєстрації та авторизації	+	-	+

Отже виходячи з аналізу розглянутих подібних аналогів, були визначені основні функціональні вимоги до розроблюваного програмного продукту: наявність панелі навігації, можливість зворотнього зв'язку, наявність калькулятора, можливість збереження результатів до БД, наявність спливаючих підказок, підтримка багатомовності, можливість розрахунку оцінки.

### 1.3. Вимоги до тренажеру

Веб-тренажер визначається як навчальний засіб, призначений для формування та вдосконалення в студентів навичок шляхом багаторазового виконання. У веб-тренажері має бути передбачено режим роботи: режим «Почати тренажер». Цей режим є роботою за алгоритмом, який представлений як послідовність кроків. У режимі «Почати тренажер» студенту пропонується виконувати крок за кроком завдання, при цьому є автоматична система контролю, що стежить за правильністю дій студента.

Веб-тренажер як засіб навчання повинен надавати користувачам наступні функціональні можливості:



- підтримка багатомовності;
- зручність та простота у використанні;
- розрахунок оцінки після виконання завдань тренажеру;
- можливість введення та збереження до БД необхідної інформації;
- забезпечення зворотного зв'язку з викладачем;
- реалізація додаткових функцій (калькулятор).

Основним завданням web-тренажера є прищеплення навичок з чисельного методу Ньютона для розв'язування систем нелінійних рівнянь[5]. Таким чином, web-тренажер дає можливість: навчити розв'язувати системи нелінійних рівнянь ітераційним методом Ньютона, освоївши алгоритм з методу Ньютона (рис. 1.12).

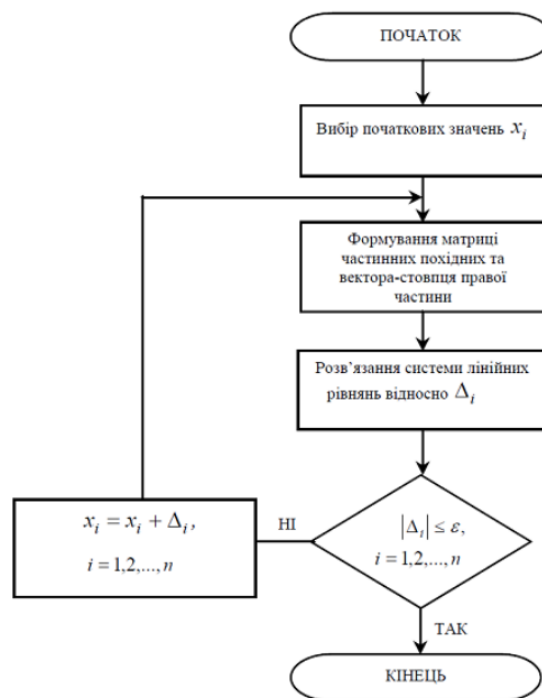


Рисунок 1.12 – Алгоритм методу Ньютона для систем нелінійних рівнянь

Розробка web-тренажера буде здійснюватися за принципом сценарію. Сценарій явно визначає ролі сторін і послідовність дій, отже, і алгоритм вирішення задачі і алгоритм роботи з web-тренажером укладаються в рамки цього поняття. Сценарій веб-

тренажера - система моделей, повно і однозначно описують його роботу. Дана система містить в собі опис логічного шляху програми і формальний опис послідовності дій при вирішенні завдань.

#### **1.4. Постановка задачі**

Мета кваліфікаційної роботи полягає в створенні інтерактивного додатку з вивчення методу Ньютона, призначенням якого є засвоїти отримані теоретичні знання та підвищити якість навчального процесу завдяки практико-орієнтованому підходу і підвищенню мотивації у навчанні.

Для досягнення мети роботи були визначені такі задачі: аналіз функціональних вимог до web-тренажера, розробка бази даних, створення інтерфейсу тренажера, побудова моделі роботи web-тренажера, створення варіативності, вибір програмних засобів для реалізації web-тренажера, створення калькулятора, інтегрування web-тренажера до університетської системи elearning, тестування додатку.

Інтерактивний додаток повинен бути реалізований у вигляді web-тренажера. Тренажер має бути інтегровано до внутрішньо університетської системи дистанційного навчання elearning та доступний в мережі Інтернет.

Розроблений додаток повинен виконувати такі функції:

- можливість введення та збереження до БД необхідної інформації;
- розрахунок оцінки після проходження всіх рівнів тренажера;
- вільний доступ до калькулятора;
- можливість обміну онлайн-повідомлення з викладачем;
- дозвіл переходу до наступного кроку;
- нарахування штрафних балів за повторні спроби виконання завдання;
- рандомне присвоювання певного варіанту завдання після натискання кнопки «Почати тренажер»;
- підтримка багатомовності: українська, російська та англійська;

- автоматична перевірка правильності виконання завдання на кожному кроці тренажеру.

Інтерфейс web-тренажеру має забезпечувати інтуїтивно зрозуміле представлення інформації розміщеної на ньому, логічний та швидкий перехід до відповідних розділів.

Навігаційні елементи повинні забезпечувати для користувачів простоту та зручність використання.

Для більш детального та чіткого розуміння про створення web-тренажера наведено в технічному завданні, який знаходиться в додатку А. В додатку Б виконано планування робіт. В додатку В наведено 10 сторінок початку і 8 сторінок кінця коду.

### **1.5. Вибір засобів реалізації**

В роботі у ході практичної реалізації інтерактивного додатку задіяно такий інструментарій:

- Visual Studio Code – для програмної реалізації тренажеру мовою програмування JavaScript;
- WEB технології HTML та CSS – для створення графічного інтерфейсу;
- СУБД MySQL – для створення бази даних;
- Мова PHP – для роботи з базою даних.

Для програмної реалізації тренажеру було обрано інтегроване середовище Visual Studio Code, яке забезпечує автодоповнення типових конструкцій, навігацію по коду. Також в редакторе Visual Studio Code є вбудований зневаджувач, який використовується для виправлення вад та для тестування. Підтримує мови та технології: JavaScript, C#, PHP, C++ та інші[17].

Для того, щоб зберігати всі дані була обрана СУБД MySQL, яка дозволяє забезпечую високий рівень безпеки даних і є зручною у використанні. База даних

міститиме, як всі персональні дані користувачів після реєстрація, так і всі результати оцінювання, запитання користувачів та завдання web-тренажера.

Панель phpMyAdmin було використано для зберігання самої бази даних. Цей веб-додаток дозволяє створювати запити, редагувати вміст таблиць, а також їх переглядати. PhpMyAdmin має відкритий код на мові PHP[6].

Створення зручного користувальницького інтерфейсу є однією з головних умов під час розробки web-тренажера. Для верстки елементів інтерфейсу тренажера використовували веб технології HTML та CSS.

### **Висновок по розділу**

Отже, у першому розділі було досліджено актуальність роботи, проведений порівняльний аналіз подібних аналогів web-тренажера, сформовано мету та задачі, функціональні вимоги до web-тренажеру та було описано засоби реалізації.

## 2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІНТЕРАКТИВНОГО ДОДАТКУ

### 2.1. Моделювання діаграми варіантів використання

На етапі моделювання web-тренажера було створено діаграму варіантів використання (Use Case), яка описує поведінку та функціональні вимоги системи, тобто як вона відповідає на зовнішні запити. Дана діаграма відіграє важливу роль у проектуванні, аналізі та тестуванні[18,14].

Діаграма варіантів використання складається з акторів, варіантів використання та зображує зв'язки між ними.

Варіанти використання описують послідовність дій, які робить система у відповідь на діяльність користувачів.

*Актори діаграми ВВ інтерактивного додатку з вивчення чисельного методу Ньютона:*

- викладач – людина, яка керує функціонуванням web-тренажера, оновлює та додає нові варіанти завдань, відповідає на питання користувачів тощо;
- незареєстрований користувач – людина, яка не зможе користуватися web-тренажером без реєстрації;
- зареєстрований користувач – людина, зайшовши до web-тренажеру, зможе проходити всі рівні тренажеру з чисельного методу Ньютона, має можливість звернутися за допомогою до викладача, написавши йому та переглядати попередні свої результати;
- база даних – база даних, яка містить усю інформацію стосовно web-тренажера та самого користувача, також зберігає нову та надає необхідну інформацію.

*Варіанти використання*

- 1) ВВ Вхід – дозволяє користувачеві увійти до web-тренажеру.

- 2) ВВ Виконання всіх рівнів web-тренажеру – дозволяє зареєстрованому користувачеві проходити завдання відповідного варіанту. Викладач може додавати, видаляти нові варіанти завдання, а також змінювати їх.
- 3) ВВ Перегляд кінцевого результату – дозволяє зареєстрованому користувачеві переглянути результат (оцінку) після виконання всіх рівнів web-тренажера.
- 4) ВВ Перегляд попередньої оцінки – дозволяє переглянути оцінки, які були отримані раніше.
- 5) ВВ Зв'язок з викладачам – ВВ дає можливість написати повідомлення викладачу.
- 6) ВВ Редагування та додавання варіантів завдань – ВВ дозволяє викладачу додавати, видаляти та змінювати завдання у web-тренажері.

Діаграма варіантів використання зображено на рис. 2.1

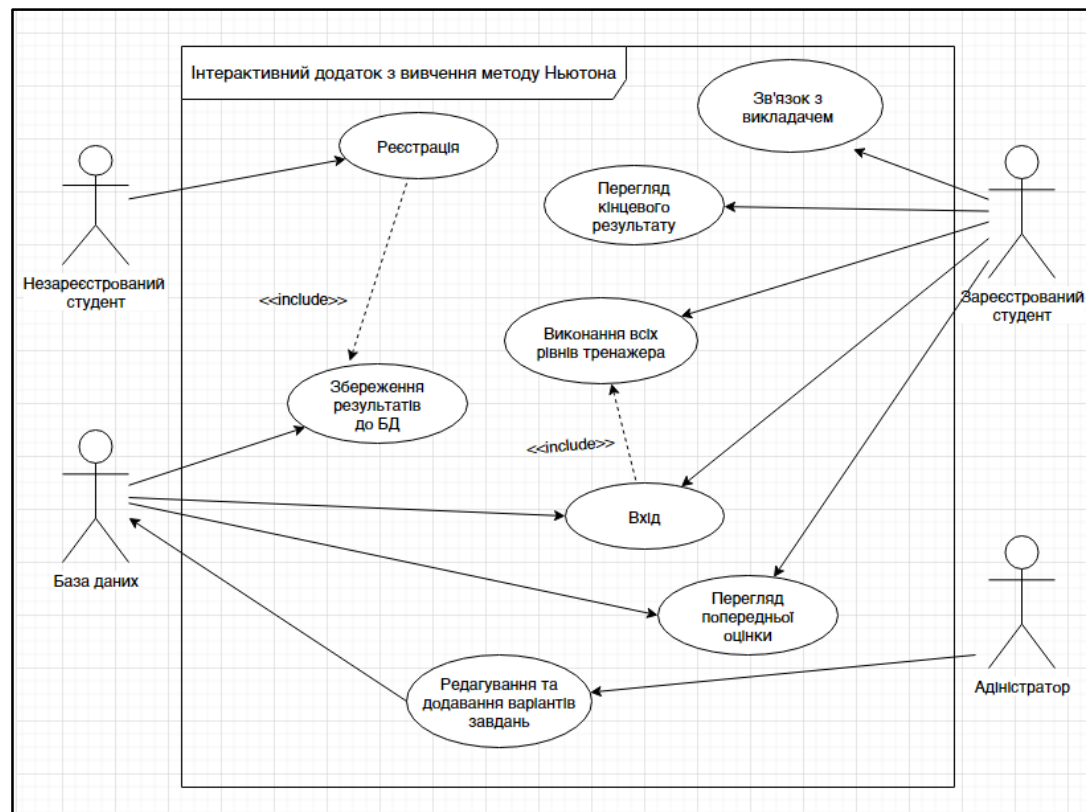


Рисунок 2.1 – Діаграма варіантів використання

## 2.2. Моделювання роботи інтерактивного додатку в IDEF0

Структурно-функціональне моделювання відбувається за рахунок побудови діаграми IDEF0 – це технологія опису системи та формування бізнес-процесів. Вона має невелику графічну нотацію (блоки та стрілки) із певними рекомендаціями. Принцип структурно – функціонального моделювання полягає в розбитті основного процесу на підпроцеси з усіма необхідними стрілками та зв'язками.

Розбиття основного процесу на менші для виправлення слабких місць є декомпозицією. Вона проводиться для ефективності системи та підвищення її продуктивності.

Для створення структурно-функціональної моделі було використано програмний продукт AllFusion Process Modeler r7.

На рис. 2.2 зображено контекстну діаграму процесу проходження Web-тренажеру в нотації IDEF0.



Рисунок 2.2 – Контекстна діаграма процесу проходження Web-тренажеру в нотації IDEF0

Дана діаграма складається з самого блоку діяльності та стрілок, які з різних сторін входять в цей блок. Блок, який розглядається, відповідає за конкретний процес і зображений у вигляді прямокутника, в середині якого зображено назва процесу. Об'єкти позначають стрілками, які впливають на процес, або є результатом його завершення [9].

Напрямок та положення стрілок на діаграмі вони описують різне значення:

- ліва сторона для стрілок входу, а саме «Потреба у вивченні методу Ньютона» вони означають матеріали або дані, що використовуються або перетворюються процесом для отримання результату;
- верхня сторона для стрілок контролю, вони означають правила, стандарти, що керують, регламентують та визначають умови. Це «Реєстраційна форма», «Інструкції до тренажеру», «Завдання тренажеру» та «Правила оцінювання»;
- нижня сторона для стрілок механізму – ресурси, які підтримують виконання функції, тобто, «Користувач», «БД», «Програмне та апаратне забезпечення»;
- права сторона для стрілок виходу, тобто, є результатом проходження процесу. В даному випадку це «Оцінка (за 5-бальною шкалою)»

Наступним кроком приведено декомпозицію контекстної діаграми, тобто розкладання основного процесу на менші складові, для детальнішого аналізу основного блоку [1].

На першому рівні декомпозиції головний блок розкладається на такі процеси:

- реєстрація;
- виконання всіх рівнів тренажеру;
- розрахунок оцінки;
- збереження результатів до БД;
- перегляд результату оцінювання.



Що і в контекстній діаграмі, що і в декомпозиції, положення стрілок має те ж саме значення, кожна стрілка автоматично дублюється в дочірню діаграму, а також створюються стрілки, які пов'язують блоки між собою, вони одночасно є виходом одного блоку та входом наступного.

Діаграма декомпозиції зображена на рис. 2.3.

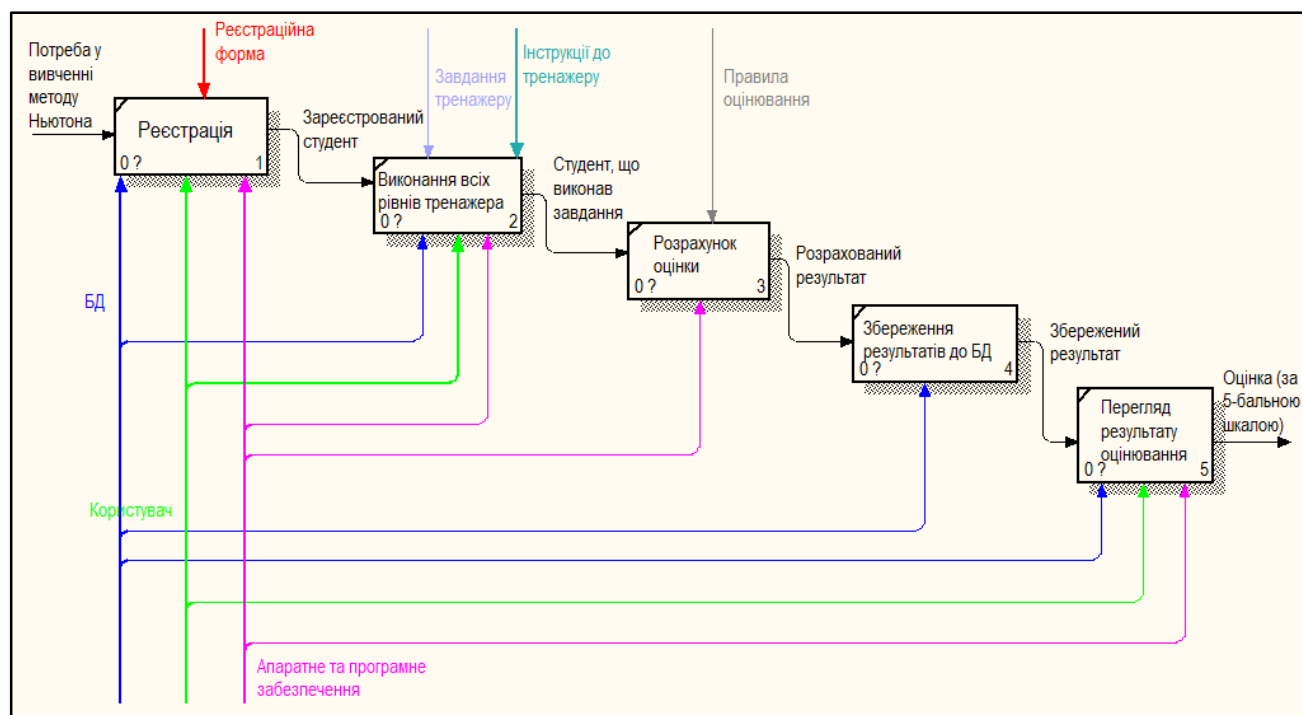


Рисунок 2.3 – Декомпозиція діаграми IDEFO

Вхідною стрілкою до блоку «Реєстрація» є «Потреба у вивченні методу Ньютона»; вихідною – «Зареєстрований студент»; стрілка контролю – «Реєстраційна форма»; стрілками механізмів – «Користувач», «БД» та «Апаратне, програмне забезпечення».

Вхідною стрілкою до блоку «Виконання всіх рівнів тренажера» є «Зареєстрований студент»; вихідною – «Студент, що виконав завдання»; стрілки контролю – «Завдання тренажера» та «Інструкції до тренажера»; стрілками механізмів – «Користувач», «БД» та «Апаратне, програмне забезпечення».

Вхідними стрілками до блоку «Розрахунок оцінки» є «Студент, що виконав завдання»; вихідною – «Розрахований результат»; стрілки контролю – «Правила оцінювання»; стрілками механізмів – «Апаратне, програмне забезпечення».

Вхідною стрілкою до блоку «Збереження результатів до БД» є «Розрахований результат»; вихідною – «Збережений результат»; стрілками механізмів – «БД».

Вхідною стрілкою до блоку «Перегляд результату оцінювання» є «Збережений результат»; вихідними – «Оцінка (за 5-бальною шкалою)»; стрілка механізмів – «Користувач», «БД» та «Апаратне, програмне забезпечення».

### **2.3. Проектування моделі бази даних web-тренажера з вивчення чисельного методу Ньютона**

Розробка серверної частини інтерактивного додатку починається зі створення та підключення бази даних з використанням Framework laravel. Створена база даних має назву міх. В ході розробки web-тренажера, Framework laravel автоматично створює необхідні таблиці в базі даних, які використовуються для авторизації, зберігання сесії користувачів та для роботи самого Framework laravel. Окрім таблиць, що створені Framework laravel, також є можливість редагувати вже існуючі та створювати свої власні[13].

Для зберігання бази даних використовується web-додаток phpMyAdmin (рис. 2.4).

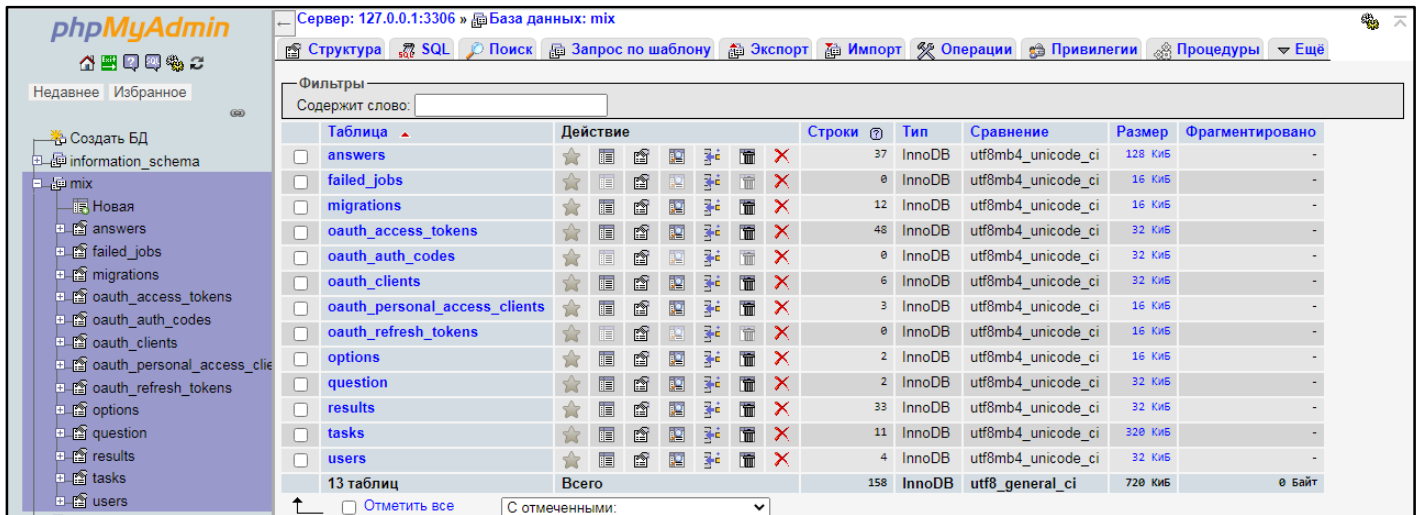


Рисунок 2.4 – База даних в phpMyAdmin

Найбільш широко використовуваним засобом розробки логічних моделей баз даних є діаграми «сутність-зв'язок» - Entity Relationship (ER-діаграми). Тому було сформовано ER-діаграму фрагменту усієї бази для відображення ролі в одній з найважливіших функцій інтерактивного додатку (рис. 2.5). ER-діаграму відображає графічно сутності, їх атрибути та взаємозв'язки між ними[8].

Даний фрагмент відповідає за роботу з даними, під час проходження web-тренажеру з чисельного методу Ньютона.

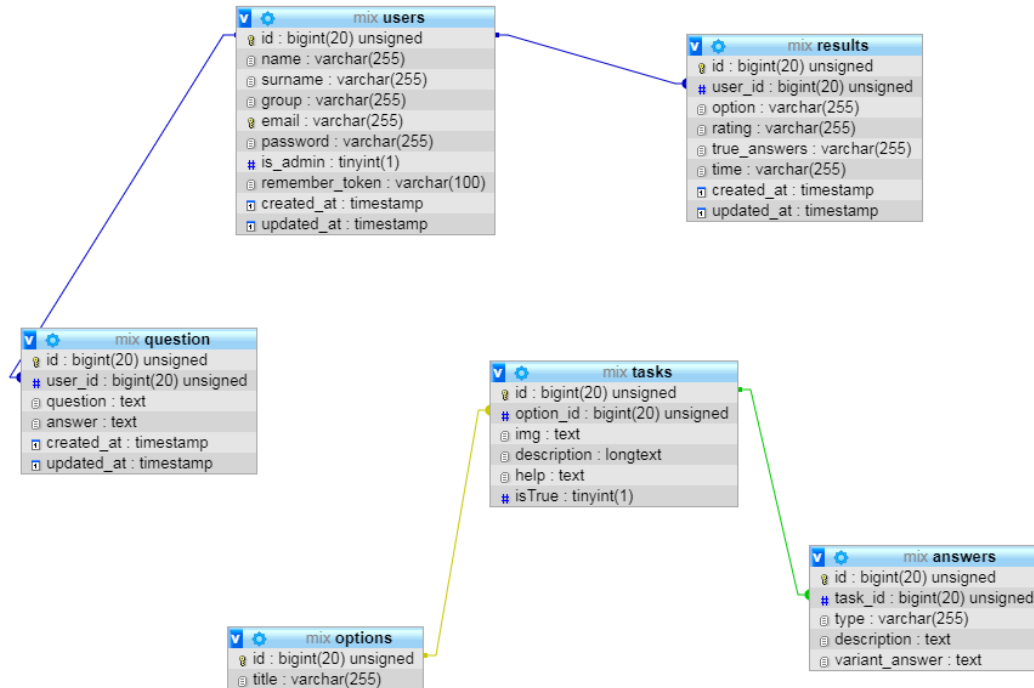


Рисунок 2.5 – ER-діаграма бази даних

Цей фрагмент містить шість сутностей:

- user (користувач). Атрибути: name (ім'я), surname (прізвище), email (електронна пошта), password (пароль), group (група);
- results (результати тестів). Атрибути: user\_id (зовнішній ключ таблиці user), option (зовнішній ключ таблиці tasks), rating (кількість балів), true\_answers (підсумковий бал), time (час закінчення виконання тренажеру), created\_at (час виконання тренажеру), updated\_at (дата);
- tasks (кроки). Атрибути: option\_id (зовнішній ключ таблиці results), img (картинки), description (опис завдання), help (підказки);
- question (питання користувачів). Атрибути: user\_id (зовнішній ключ таблиці user), question (питання), created\_at (час), updated\_at (дата);

- answers (таблиця з варіантами відповідей). Атрибути: task\_id (зовнішній ключ таблиці tasks), type (тип), description (опис завдання), variant\_answer (варіант відповіді);
- options (варіант). Атрибути: title (назва).

Web-тренажер має дві бази даних. Одна база даних відповідає за зберігання даних користувачів та їх результати оцінювання, а інша зберігає всі завдання тренажеру.

Сутності зображуються у вигляді таблиць, а атрибути у вигляді полів цих таблиць. В даній ER-діаграмі представлено один тип зв'язку з сутностями це: «один до багатьох».

В табл. 3.1 зображено концептуальну модель бази даних, в якій представлено імена необхідних таблиць, типів, атрибутів, їх обмеження.

Таблиця 3.1 Концептуальна модель бази даних

Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
users	user_id	Ідентифікатор користувача	bigint(20)	PK	Не пустий, унікальний
	name	Ім'я користувача	varchar(255)	FK	Не пустий
	surname	Прізвище користувача	varchar(255)		Не пустий
	email	Почта	varchar(255)		Не пустий
	password	Пароль	varchar(255)		Не пустий
	group	Група	varchar(255)		Не пустий
results	result_id	Ідентифікатор результату тестів	bigint(20)	PK	Не пустий, унікальний
	user_id	Ідентифікатор користувача	varchar(255)	FK	Не пустий
	option	Варіант	varchar(255)		Не пустий
	rating	Кількість балів	varchar(255)		Не пустий
	time	Час закінчення тренажера	varchar(255)		Не пустий
	true_answers	Підсумковий бал	varchar(255)		Не пустий
	created_at	Час виконання	timestamp		Не пустий
	updated_at	Дата	timestamp		Не пустий
tasks	tasks_id	Ідентифікатор кроків	bigint(20)	PK	Не пустий, унікальний
	optoin_id	Ідентифікатор варіанта	bigint(20)	FK	Не пустий

	img	Картинка	text		Не пустий
	description	Опис завдання	longtext		Не пустий
	help	Підказки	text		Не пустий
question	question_id	Ідентифікатор питання користувачів	bigint(20)	PK	Не пустий, унікальний
	user_id	Ідентифікатор користувача	varchar(255)	FK	Не пустий
	question	Запитання	text		Не пустий
	answer	Відповіді	text		Не пустий
	created_at	Час виконання	timestamp		Не пустий
	updated_at	Дата	timestamp		Не пустий
answers	answers_id	Ідентифікатор відповіді	bigint(20)	PK	Не пустий, унікальний
	task_id	Ідентифікатор завдання	bigint(20)	FK	Не пустий
	type	Тип	varchar(255)		Не пустий
	description	Опис завдання	text		Не пустий
	variant_answer	Варіант відповіді	text		Не пустий
options	options_id	Ідентифікатор варіанта	bigint(20)	PK	Не пустий, унікальний
	title	Назва	varchar(255)		Не пустий

## 2.4. Концептуальна модель web-тренажера

Для опису роботи веб-тренажера була побудована модель, тобто діаграма потоків даних (data flow diagram, DFD), яка слугує одним з інструментів аналізу структури. Модель DFD - є ієрархічною моделлю. До всіх її процесів може бути застосована декомпозиція, тобто розбиття цілого на структурні складові, відносини яких можуть бути відображені на окремій діаграмі[17].

Схема потоків даних в реалізованому веб-тренажері представлена на рис. 2.3.

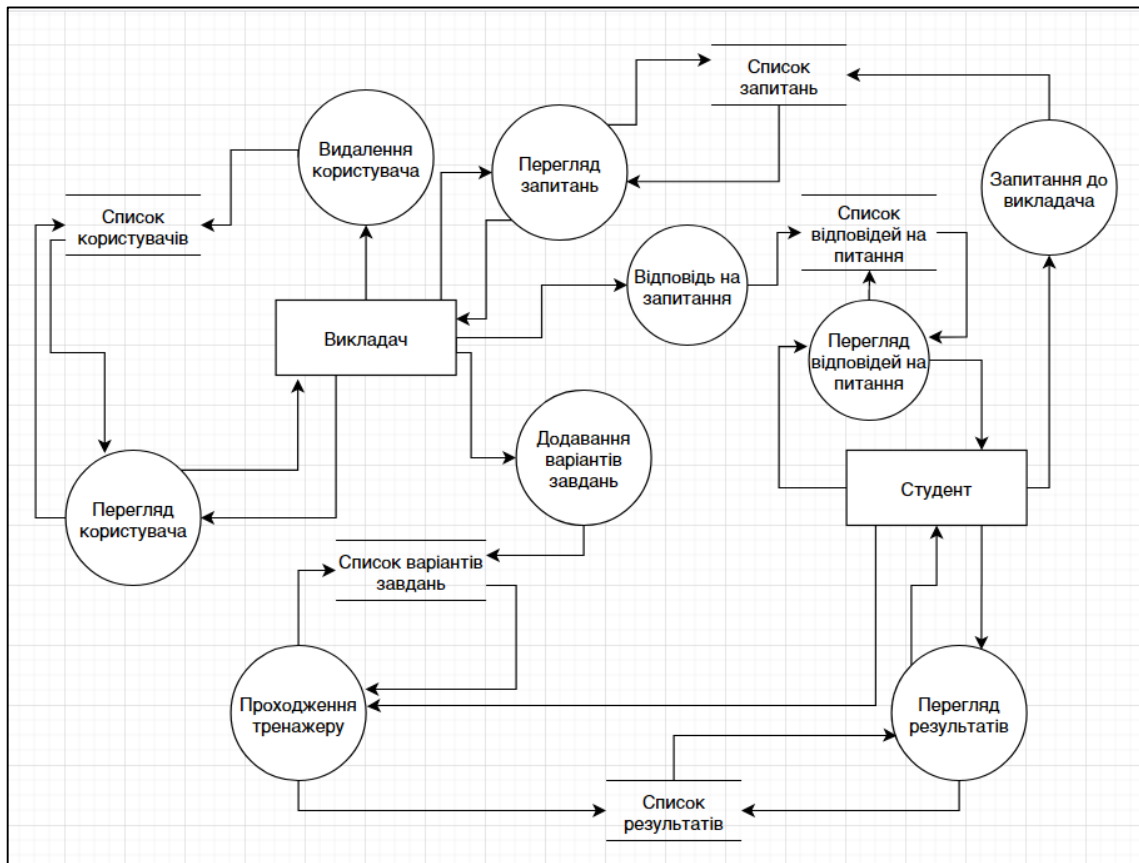


Рисунок 2.6 – Діаграма потоків даних web-тренажера

## Висновок по розділу

Отже, у другому розділі було виконано структурно-функціональне моделювання процесу проходження web-тренажера, моделювання діаграми варіантів використання, спроектовано модель бази даних та наведений опис таблиць. Також було побудовано концептуальну модель web-тренажера тобто діаграму потоків даних.

### 3 РОЗРОБКА WEB-ТРЕНАЖЕРА З ВИВЧЕННЯ ЧИСЕЛЬНОГО МЕТОДУ НЬЮТОНА

#### 3.1. Архітектура програмного продукту

При розробці комп'ютерного навчального web-тренажера з вивчення чисельного методу Ньютона береться до увагу розробка його архітектури.

На рис. 3.1 зображена трьохшарова архітектура web-додатку.

Модель трьохшарової клієнт-серверної архітектури дозволяє розділити функціонал навчального тренажера за трьома різними рівнями:

- клієнт,
- сервер інтерактивного додатку,
- сервер БД.

Всі підсистеми web-тренажера відповідають своїм функціональними вимогам.

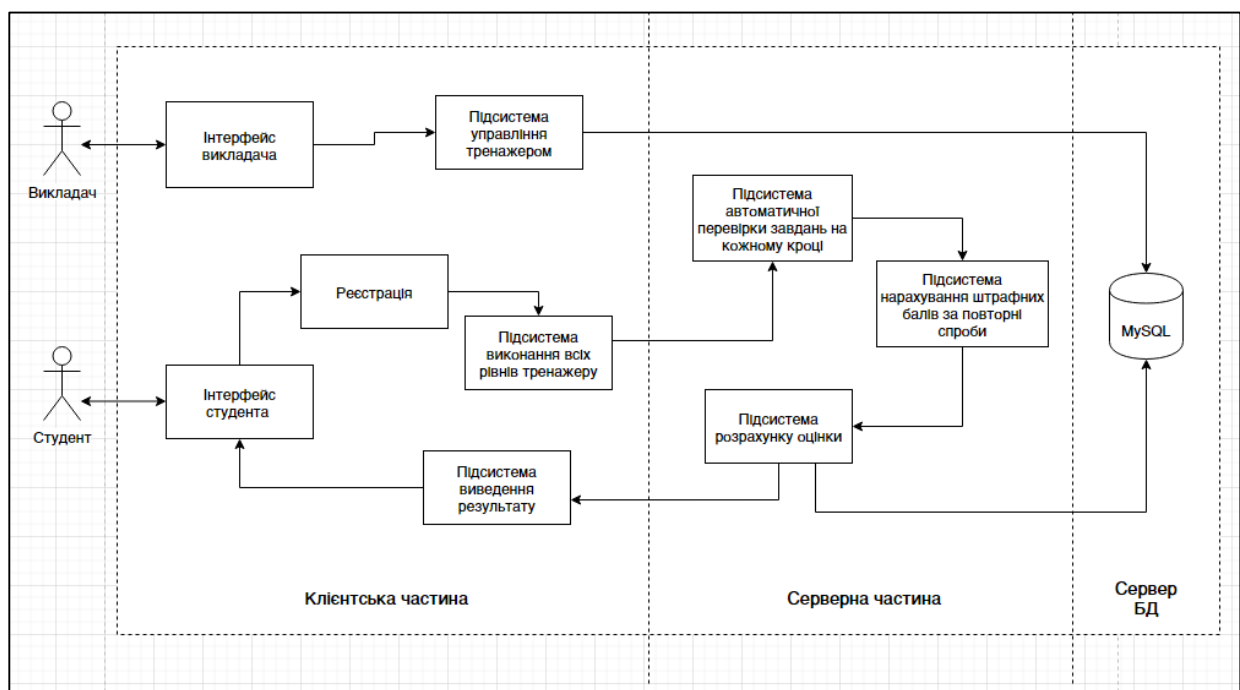


Рисунок 3.1 – Архітектура інтерактивного додатку з вивчення методу Ньютона



Основні користувачі такої системи поділяються на дві групи:

- 1) викладач;
- 2) студенти.

Для кожної групи інтерфейс унікальний і має своє функціональне завдання. Інтерфейси розташовуються на рівні клієнта. На другому рівні - рівні сервера додатку реалізується основна частина тренажера для підготовки студентів, а на третьому рівні сервера даних знаходиться СУБД MySQL [12].

Перед тим як зайти в тренажер студенти повинні зареєструватися. У базі даних зберігаються профілі всіх студентів. Робота студента в тренажері починається з натискання на кнопку «Почати тренажер». Після цього студенту присвоюється певний варіант завдань з 10 кроків, які він виконує послідовно. Виконавши крок, підсистема автоматично перевіряє на правильність завдання. Якщо студент невірно виконав завдання йому дається можливість виправити помилку, в наслідок чого йому будуть нараховуватися штрафні балі, за що відповідає підсистема нарахування штрафних балів. Після проходження тренажера студенту виводиться його результат оцінювання, після чого він зберігається в базі даних MySQL.

Підсистема управління тренажером дозволяє викладачу змінювати та додавати нові варіанти завдань, які також зберігаються в базі даних, відповідати на повідомлення студентів. За необхідності викладач може видаляти користувачів з тренажеру.

### **3.2. Програмна реалізація**

Сама розробка була здійснена у безкоштовному текстовому редакторі VS Code. Visual Studio Code – це крос-платформенний редактор скриптів, який створений корпорацією Майкрософт. Спрощуючи написання надійних скриптів PowerShell, він надає широкі інтерактивні можливості редагування цих скриптів з розширенням PowerShell.

Також для відслідковування версій web-додатку та збереження вихідного коду використовували систему контролю версій Git. Системи контролю версій дозволяють розробникам зберігати всі зміни, внесені в код. Тому в разі, помилки, вони можуть просто відкотити код до робочого стану замість того, щоб витратити години на пошуки маленької помилки.

Проект був розроблений на віртуальному веб-сервері OpenServer. Open Server Panel – це портативна серверна платформа і програмне середовище, створена спеціально для веб-розробників з урахуванням їх рекомендацій і побажань. Також використовувався менеджер модулів для PHP – Composer.

Розробка web-тренажера починається з створення нового проекту з використанням Framework laravel.

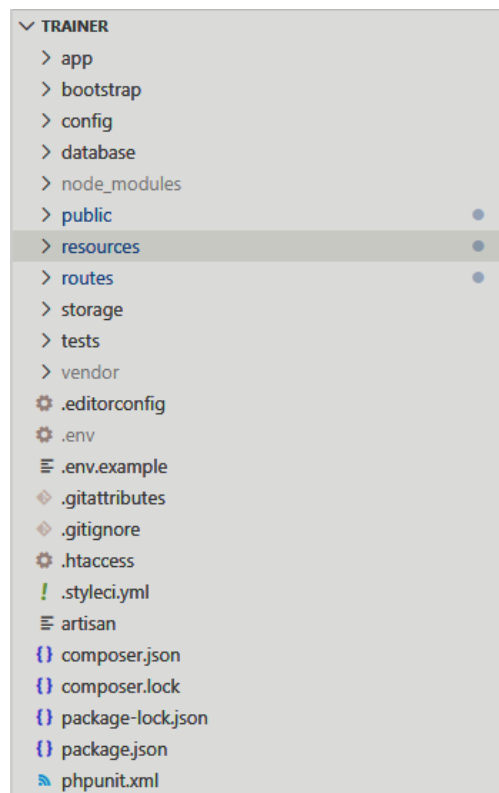


Рисунок 3.2 Кореневий каталог web-тренажера

Потім починається розробка клієнтської частини додатку, а саме створення всіх необхідних компонентів, налаштування маршрутів доступу і верстка шаблонів та

розробка модулів серверної частини додатку, створення всіх необхідних контролерів, моделей та налаштування API роутеру.

### 3.2.1. Розробка клієнтської частини

Написання додатку починається із створення «скелету» з використанням мови розмітки HTML. Після чого за допомогою мови каскадних стилів CSS формується зовнішній вигляд web-тренажеру і надається візуальна форма елементів для зручного користування. Одним з основних файлів є `style.css`, який слугує для графічного оформлення теми та відповідає за стилі оформлення HTML сторінок інтерактивного додатку з вивчення методу Ньютона (рис. 3.3).

```
43 <style lang="css">
44   h1.titleHeader {
45     font-size: 49px;
46     font-weight: 400;
47     color: #444444 !important;
48     line-height: 1;
49     margin-bottom: 20px;
50   }
51   h5.countStep {
52     font-size: 20px !important;
53     color: #444444 !important;
54   }
55   h5.countStep span {
56     padding: 3px 6px 5px;
57     font-size: 12px !important;
58     color: #ffffff;
59     background-color: #bbbbbb;
60     border-radius: 10px;
61   }
62   .buttonHome {
63     color: #fff !important;
64     font-weight: bold;
65   }
66   img.language_img {
67     float: left;
68     width: 100px;
69     margin: 5px;
70     cursor: pointer;
71   }
72 </style>
```

Рисунок 3.3 – Фрагмент файлу `style.css`

Для кожної сторінки створюється власний компонент, який підключається в файлі `router.js`. Всі компоненти знаходяться в папці `resources/js/`.

Наступним кроком створюється єдина точка входу, яка підключає всі необхідні компоненти і модулі JavaScript (рис. 3.4).

```
JS app.js  x
resources > js > JS app.js > ...
1  import router from "./router";
2  import Vuetify from "vuetify";
3  import VueSweetalert2 from 'vue-sweetalert2';
4  import 'sweetalert2/dist/sweetalert2.min.css';
5  import uk from 'vuetify/src/locale/uk.ts';
6  import AppComponent from "./components/AppComponent";
7  import * as VueWindow from '@hscmap/vue-window'
8
9  window.Vue = require('vue');
10 window.Vue.use(VueWindow);
11 window.Vue.use(Vuetify);
12 window.Vue.use(VueSweetalert2);
13
14 const app = new Vue({
15   el: '#app',
16   components: {
17     AppComponent
18   },
19   vuetify: new Vuetify({
20     icons: {
21       iconfont: 'mdi',
22     },
23     lang: {
24       locales: { uk },
25       current: 'uk',
26     },
27   }),
28   router
29 });
30 |
```

Рисунок 3.4 – Файл єдиної точки входу

Створивши всі необхідні шаблони і компоненти. Відбувається налаштування маршрутів для навігації між сторінками web-тренажеру. Далі на рис. 3.5 наведено приклад коду маршруту для головної сторінки.

```
JS router.js x
resources > js > JS router.js > ...
16 let router = new Router({
17   mode: "history",
18   routes: [
19     {
20       path: '/',
21       name: 'home',
22       component: HomeComponent,
23       meta: {
24         requiresAuth: true
25       }
26     },
27     {
28       path: '/test',
29       name: 'test',
30       component: TestComponent,
31       meta: {
32         requiresAuth: true
33       }
34     },
35     {
36       path: '/login',
37       name: 'login',
38       component: LoginComponent,
39       meta: {
40         guest: true
41       }
42     },
43   ]
44 })
```

Рисунок 3.5 – Файл router

В результаті при кліку на посилання відбувається перехід на необхідний компонент вказаний в маршрутизаторі.

### 3.2.2. Розробка серверної частини

Розробка серверної частини починається зі створення та підключення бази даних. Потім створюються міграції для автоматичного створення бази даних. Міграції – це система контролю версій для бази даних. Вони дозволяють змінювати структуру БД. Далі наведено приклад міграції для створення таблиці користувачів (рис. 3.6).

```

2014_10_12_000000_create_users_table.php X
database > migrations > 2014_10_12_000000_create_users_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateUsersTable extends Migration
8  {
9      public function up()
10     {
11         Schema::create('users', function (Blueprint $table) {
12             $table->id();
13             $table->string('name');
14             $table->string('surname');
15             $table->string('group')->nullable();
16             $table->string('email')->unique();
17             $table->string('password');
18             $table->boolean('is_admin');
19             $table->rememberToken();
20             $table->timestamps();
21         });
22     }
23     public function down()
24     {
25         Schema::dropIfExists('users');
26     }
27 }
28

```

Рисунок 3.6 – Файл міграції таблиці користувачів

Далі створюються моделі для зручної роботи з базою даних, налаштувань залежностей між таблицями і підключення додаткових модулів. На рис. 3.7 наведено приклад моделі користувачів.

```

User.php X
app > Models > User.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Contracts\Auth\MustVerifyEmail;
6  use Illuminate\Foundation\Auth\User as Authenticatable;
7  use Illuminate\Notifications\Notifiable;
8  use Laravel\Passport\HasApiTokens;
9
10 class User extends Authenticatable {
11     use HasApiTokens,Notifiable;
12     protected $fillable = [
13         'name', 'surname', 'group', 'email', 'password', 'is_admin'
14     ];
15     protected $hidden = [
16         'password', 'remember_token',
17     ];
18 }
19

```

Рисунок 3.7 – Файл моделі таблиці користувачів

Всі запити що надходять на сервер обробляються API маршрутизатором, що знаходиться в файлі `routes/api.php`. Маршрутизація визначає, як додаток відповідає на клієнтський запит до конкретної адреси (рис. 3.8).

```
api.php x
routes > api.php
1 |<?php
2
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\Route;
5
6 //options
7 Route::get('options', 'TasksController@getOptions');
8 Route::get('option/{id}', 'TasksController@getOptionId');
9 Route::post('option', 'TasksController@postOption');
10 Route::delete('option/{id}', 'TasksController@delOption');
11
12 //tasks
13 Route::post('post-task/{option_id}', 'TasksController@postTask');
14 Route::post('update-task/{task_id}', 'TasksController@updateTask');
15 Route::delete('task/{task_id}', 'TasksController@delTask');
16
17 //answer
18 Route::post('post-answer/{task_id}', 'TasksController@postAnswer');
19 Route::post('update-answer/{answer_id}', 'TasksController@updateAnswer');
20 Route::delete('answer/{answer_id}', 'TasksController@delAnswer');
21
22 //users
23 Route::delete('user/{id}', 'UserController@deleteUser');
24 Route::get('users', 'UserController@getUsers');
25 Route::post('register', 'UserController@registerUser');
26 Route::post('login', 'UserController@userLogin');
27 Route::post('user/{id}', 'UserController@updateUser');
28
```

Рисунок 3.8 – Файл запитів на сервер

Потім в залежності від типу запиту підключається необхідний контролер з необхідним методом який опрацює дані отримані з бази даних. Контролер – з'єднує моделі, види і інші компоненти необхідні для роботи системи. А також відповідає за обробку запитів користувача, а саме отримання і обробку даних та відправку готового результату клієнтській частині системи. Приклад такого контролеру для отримання списку всіх новин наведено нижче на рис. 3.9.

```

TasksController.php X
app > Http > Controllers > TasksController.php
12 class TasksController extends Controller
13 {
14     // options
15     function getOptions() {
16         $data = Options::with("tasks")->get();
17         foreach($data as $key => $value) {
18             foreach($value->tasks as $k => $v) {
19                 $v->answers = Answers::where("task_id", $v->id)->get();
20             }
21         }
22         return response()->json($data);
23     }
24     function getOptionId($id) {
25         $data = Options::with("tasks")->find($id);
26         foreach($data->tasks as $k => $v) {
27             $v->answers = Answers::where("task_id", $v->id)->get();
28         }
29         return response()->json($data);
30     }
31     function postOption(Request $request) {
32         $model = new Options();
33         $response = $model->create($request->all());
34         return response()->json($response);
35     }
36     function delOption($id) {
37         Options::find($id)->delete();
38     }
39 }

```

Рисунок 3.9 – Файл Controller завдань

### 3.3. Використання програмного додатку

Інтерактивний додаток з вивчення методу Ньютона було інтегровано до університетської системи elearning для дистанційної форми навчання.

Отже для того, щоб розпочати виконувати тренажер, студенту спочатку потрібно зареєструватися та пройти авторизацію. У першому випадку, студенту необхідно заповнити відповідну форму, ввівши свої персональні дані, після чого вони вносяться до бази даних. Форми реєстрації та авторизації наведені на рис. 3.10-3.11.



Регістрація

Ім'я  
Ирина

Прізвище  
Литвин

Е-mail  
ira.lit22@gmail.com

Група  
IT-62

Пароль  
.....

Повторіть пароль  
.....

НАЗАД ЗАРЕЄСТРУВАТИСЬ

Рисунок 3.10 – Форма Реєстрації

Авторизація

Е-mail  
ira.lit22@gmail.com

Пароль  
.....

ВХІД РЕЄСТРАЦІЯ

Рисунок 3.11 – Форма Авторизації

Після проходження авторизації студент потрапляє на головну сторінку тренажера (рис. 3.12). На головній сторінці web-тренажера зображено кнопки, а також є додаткове випадаюче меню для зручності користувача переходити між сторінками. В даному додатку підтримується багатомовність: українська, російська та англійська мови.

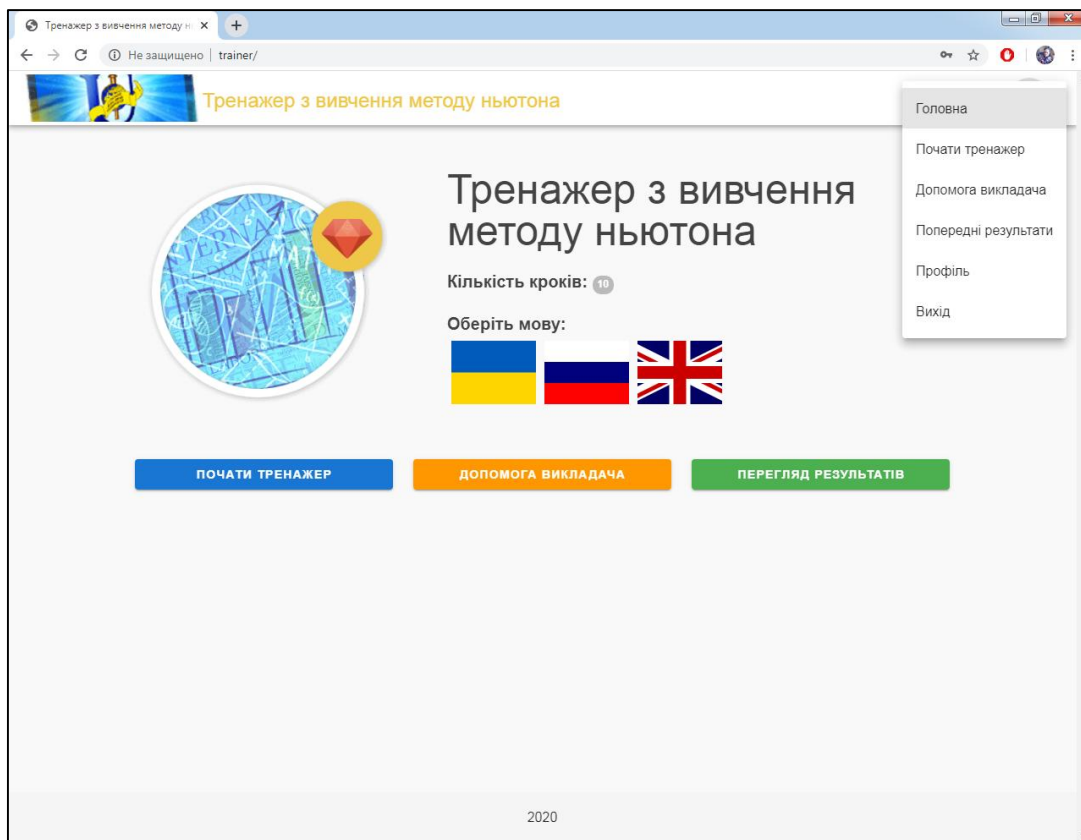


Рисунок 3.12 – Головна сторінка web-тренажера

Натиснувши у випадаючому меню на профіль студент потрапляє до своїх персональних даних, які він може змінювати (рис. 3.13). Також у студента є можливість змінювати пароль та навіть видалити свій акаунт, підтвердивши у спливаючому вікні свої дії (рис. 3.14).

Персональні дані

Ім'я: Ирина

Прізвище: Литвин

E-mail: ira.lit22@gmail.com

Група: IT-62

ЗМІНИТИ ПАРОЛЬ

Новий пароль

Повторіть пароль

ЗБЕРЕГТИ ВИДАЛИТИ АКАУНТ

2020

Рисунок 3.13 – Персональні дані користувача

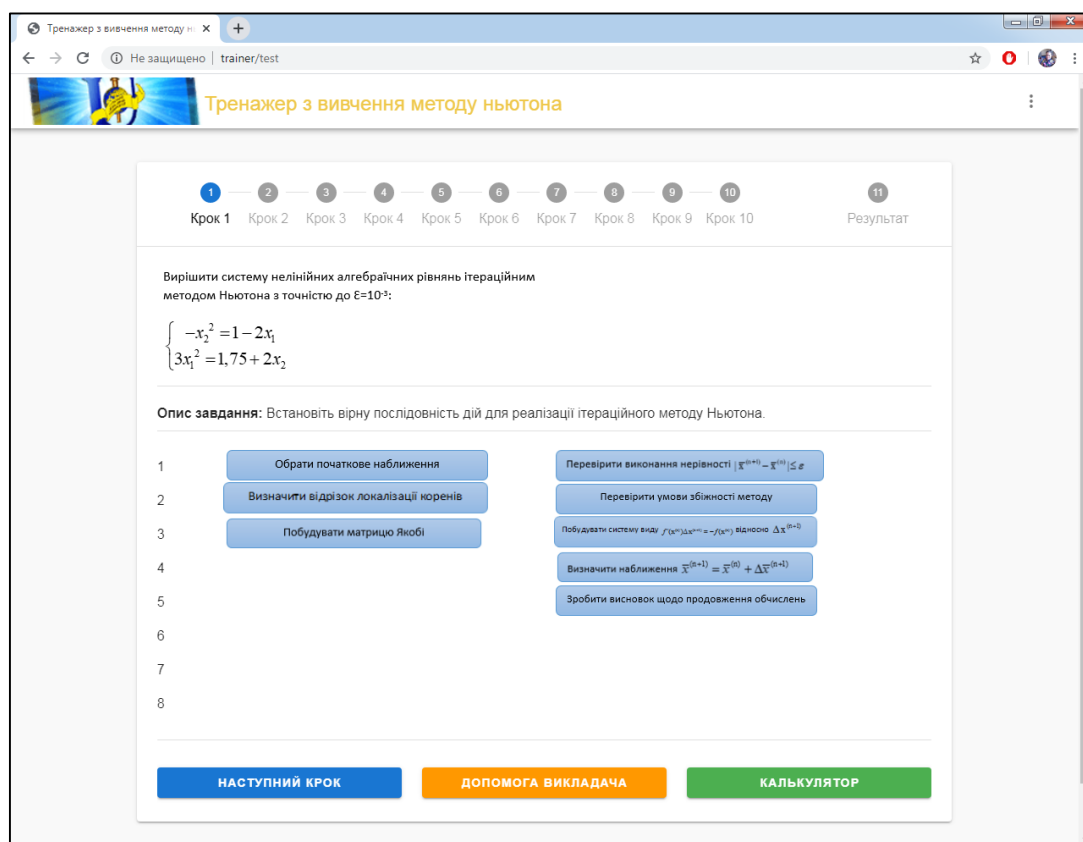
!

Бажаєте видалити аккаунт?

Так Закрити

Рисунок 3.14 – Спливаюче вікно для підтвердження дії

Щоб почати проходити тренажер студент повинен натиснути кнопку «Почати тренажер» на головній сторінці або у випадаючому меню. Після цього йому присвоюється певний варіант завдань, який складається з 10 кроків. Кроки виконуються послідовно. На рис. 3.15 зображено вікно крок 1, в якому є постановка задачі та саме завдання, яке потрібно виконувати перетягуванням.



Тренажер з вивчення методу ньютона

Крок 1 Крок 2 Крок 3 Крок 4 Крок 5 Крок 6 Крок 7 Крок 8 Крок 9 Крок 10 Результат

Вирішити систему нелінійних алгебраїчних рівнянь ітераційним методом Ньютона з точністю до  $\epsilon=10^{-3}$ :

$$\begin{cases} -x_2^2 = 1 - 2x_1 \\ 3x_1^2 = 1,75 + 2x_2 \end{cases}$$

Опис завдання: Встановіть вірну послідовність дій для реалізації ітераційного методу Ньютона.

1	Обрати початкове наближення	Перевірити виконання нерівності $ x^{(n+1)} - x^{(n)}  \leq \epsilon$
2	Визначити відрізок локалізації коренів	Перевірити умови збіжності методу
3	Побудувати матрицю Якобі	Побудувати систему виду $f(x^{(n+1)}) = -f(x^{(n)})$ відносно $\Delta x^{(n)}$
4		Визначити наближення $x^{(n+1)} = x^{(n)} + \Delta x^{(n)}$
5		Зробити висновок щодо продовження обчислень
6		
7		
8		

НАСТУПНИЙ КРОК ДОПОМОГА ВИКЛАДАЧА КАЛЬКУЛЯТОР

Рисунок 3.15 – Вікно завдань кроку 1

На кожному кроці передбачається автоматична перевірка правильності виконання завдань, де невірні відповіді підсвічуються червоним кольором, вірні – зеленим (рис. 3.16). Коли студент допустив помилки йому на екран виводиться повідомлення про помилку (рис. 3.16). Перехід до наступного кроку відбувається за умови правильного виконання попереднього кроку. У студента є три спроби повторно виконати відповідний крок, але за це йому будуть нараховуватися штрафні бали.

Тренажер з вивчення методу ... x

Не захищено | trainer/test

Крок 1 Крок 2 Крок 3 Крок 4 Крок 5 Крок 6 Крок 7 Крок 8 Крок 9 Крок 10 Результат

За побудованими графіками визначити початкове наближення рішення системи (варіанти представлені справа)

$x_1^{(0)}=1$   
 $x_2^{(0)}=0$   
 $x_1^{(0)}=0,5$   
 $x_2^{(0)}=3$   
 $x_1^{(0)}=-2$   
 $x_2^{(0)}=-1$   
 $x_1^{(0)}=-1$   
 $x_2^{(0)}=1$   
 $x_1^{(0)}=2,5$   
 $x_2^{(0)}=4$

Опис завдання: Дайте правильні відповіді.

ПІДСКАЗКА

x1 =:  
 Відповідь: 1

x2 =:  
 Відповідь: 4

Не все виконано вірно! Перевірте та виправте помилки!

НАСТУПНИЙ КРОК ДОПОМОГА ВИКЛАДАЧА КАЛЬКУЛЯТОР

Рисунок 3.16 – Вікно завдань кроку 3

Також web-тренажер містить підказки (рис. 3.17), які дають змогу студентам більш точно зрозуміти питання. Під час розрахунків студент має можливість використовувати вбудований калькулятор, який зображений на рис. 3.18.

Опис завдання: Дайте правильні відповіді.

ПІДСКАЗКА

Для забезпечення умов збіжності методу початкове наближення слід обирати у малому околі розв'язку

Рисунок 3.17 – Підказка



Рисунок 3.18 – Калькулятор

Якщо у студента під час проходження кроків виникають запитання до самого завдання або ж до даного методу Ньютона, то він може звернутися за допомогою до викладача, написавши йому повідомлення в спливаючому вікні, яке зображене на рис. 3.19

Тренажер з вивчення методу Ньютона

Не захищено | trainer/test

Будемо матрицю Якобі для функцій  
 $f_1 = -x_2^2 - 1 + 2x_1$ ;  $f_2 = 3x_1^2 - 1,75 - 2x_2$

$\vec{f}'(\vec{x}) = \begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix}$  ← Вірно розставити компоненти матриці

$\vec{f}'(\vec{x}) = \begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix}$  ← Вірно обрати та розставити компоненти

Опис завдання: Дайте правильні відповіді на запитання.

Перша матриця (вказіть порядкові номери елементів):

2	4
1	3

Друга матриця (вказіть правильні значення):

2	-2x <sup>2</sup> +2
6x <sup>1</sup>	-2

Запитання: У чому полягає суть методу Ньютона?

ЗАКРИТИ НАДІСЛАТИ

Не все виконано вірно! Перевірте та виправте помилки!

НАСТУПНИЙ КРОК ДОПОМОГА ВИКЛАДАЧА КАЛЬКУЛЯТОР

Рисунок 3.19 – Спливаюче вікно повідомлення

Після виконання всіх кроків web-тренажера студенту на екран виводить кінцевий його результат (рис. 3.20), який показує час виконання роботи, підсумковий бал у відсотковому відношенні та бал за 5-бальною шкалою.

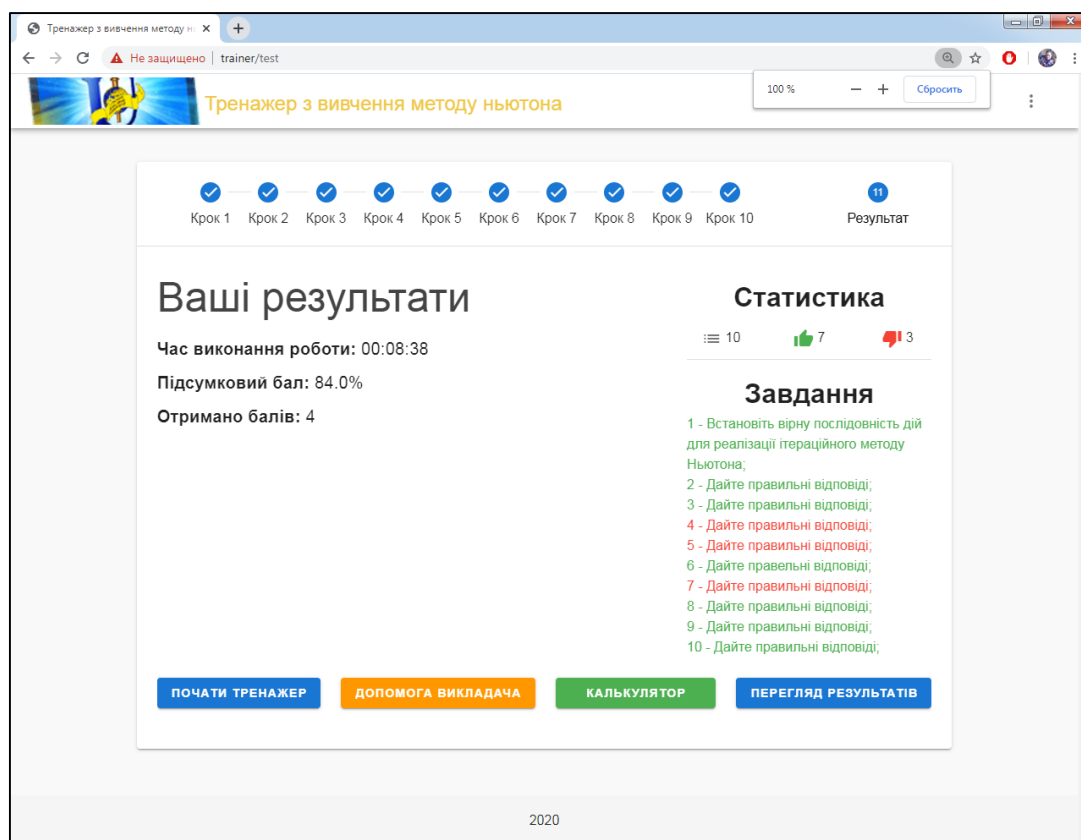


Рисунок 3.20 – Вікно кінцевого результату

Для того щоб , студент мав змогу дивитися свої попередні результати, які він проходив раніше, йому необхідно перейти до вікна попередніх результатів, натиснувши кнопку «Перегляд результатів». У цьому вікні зображено його номер варіанту, підсумковий бал, бал за 5-бальною шкалою, дату проходження тренажера та час виконання тесту (рис. 3.21).

Варіант	Підсумковий бал %	Отримано балів	ДатаЧас	Час виконання тесту
1	84.0	4	2020-06-02 - 08:31:03	00:08:38
1	100.0	5	2020-05-27 - 23:52:55	00:04:30
2	87.2	4	2020-05-27 - 23:35:37	00:07:54
1	58.4	3	2020-05-27 - 23:20:18	00:13:39

Рисунок 3.21 – Вікно попередніх результатів

В даному web-тренажері також є панель викладача. На рис. 3.22 зображено меню «Варіанти» для викладача. В якому він може змінювати завдання, видаляти їх, а також додавати нові (рис. 3.22).

Меню

- Варіанти
- Користувачі
- Питання користувачів

ДОДАТИ ВАРІАНТ

ВАРІАНТ 1    ВАРІАНТ 2    ВАРІАНТ 3    ВАРІАНТ 4

ВИДАЛИТИ ВАРІАНТ

Крок 1: ВИДАЛИТИ ЗАВДАННЯ

Опис завдання:  
Встановіть вірну послідовність дій для реалізації ітераційного методу Ньютона

Файл зображення

Вирішити систему нелінійних алгебраїчних рівнянь ітераційним методом Ньютона з точністю до  $\epsilon=10^{-3}$ :

$$\begin{cases} -x_2^2 = 1 - 2x_1 \\ 3x_1^2 = 1,75 + 2x_2 \end{cases}$$

Підказка

ЗБЕРЕГТИ

Рисунок 3.22 – Вікно викладача для редагування варіантів завдань



Також викладач має можливість слідкувати за користувачами, які зареєструвалися до тренажеру та навіть видаляти їх за потреби у меню «Користувачі» (рис. 3.23).

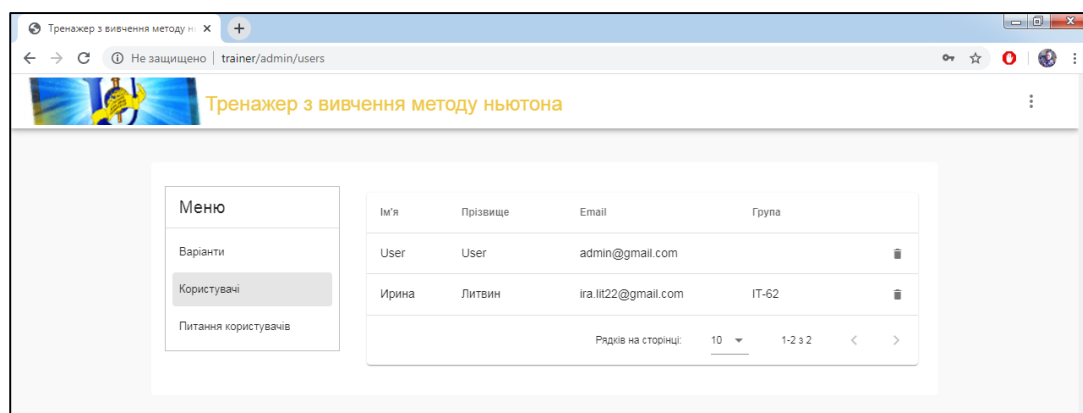


Рисунок 3.23 – Вікно зареєстрованих користувачів у панелі викладача

Отримавши повідомлення від користувачів викладач може відповідати на них, перейшовши до меню «Питання користувачів», на якому зображено саме запитання, дата та час його відправлення (рис. 3.24).

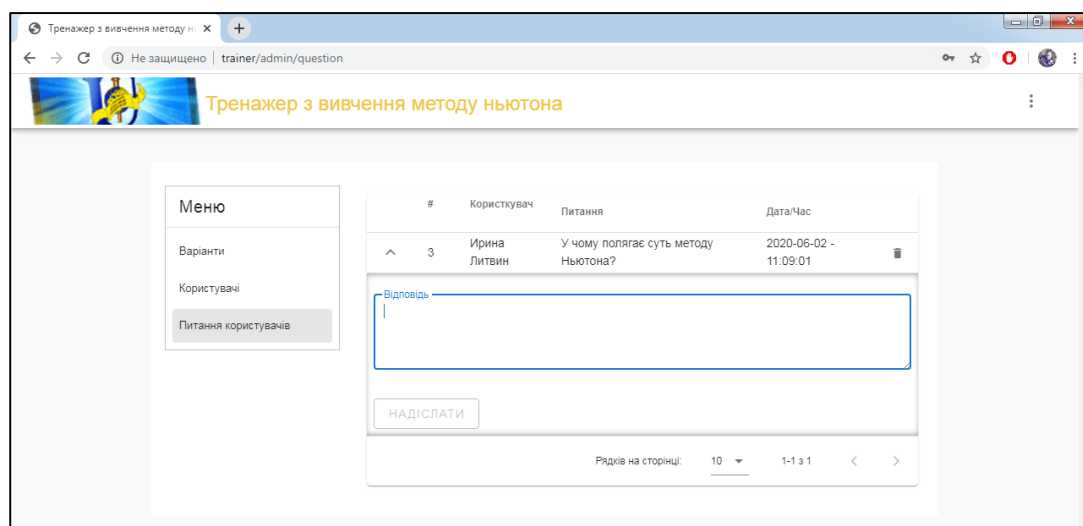


Рисунок 3.24 – Вікно для відповідей на панелі викладача

## ВИСНОВКИ

Метою кваліфікаційної роботи бакалавра було створення інтерактивного додатку для удосконалення навчального процесу завдяки практико-орієнтованому підходу та підвищення рівня засвоєння знань студентами при вивченні теми «Застосування чисельного методу Ньютона для вирішення систем нелінійних рівнянь».

Для досягнення мети, кваліфікаційну роботу було поділено на такі підзадачі:

- аналіз предметної сфери;
- формування функціональних вимог до web-тренажеру;
- розробка бази даних;
- розробка та реалізація web-додатку;
- тестування роботи тренажеру;
- інтеграція тренажеру до університетської системи elearning

На початку розробки дипломного проекту бакалавра було досліджено актуальність роботи, створено порівняльну таблицю та проведено порівняльний аналіз аналогів та власного web-тренажера за сформованими критеріями. На базі існуючих аналогів були виявлені переваги та недоліки.

В роботі практичної реалізації інтерактивного додатку було задіяно такий інструментарій:

- Visual Studio Code – для програмної реалізації тренажеру;
- WEB технології HTML та CSS – для створення графічного інтерфейсу;
- СУБД MySQL – для створення бази даних;
- мова JavaScript – для роботи з базою даних.

Для завершення проекту було проведене планування робіт, в ході якого відбулася декомпозиція мети на підзадачі. Було побудовано WBS структуру, у якій зазначені всі виконувані роботи, після цього було сформовано OBS – організаційну структуру виконавців згідно поставлених робіт. Також створена матриця

відповідальності на підставі WBS та OBS структур. Був проведений аналіз ризиків та прописані стратегії вирішення їх. Завдяки продукту MS Project, у вигляді гістограми, була побудована діаграма Ганта, яка відображає тривалість відповідного процесу. В кінці були проаналізовані можливі ризики до web-тренажера.

Проектування інтерактивного додатку можна поділити на три частини: структурно-функціональне моделювання, проектування бази даних та моделювання варіантів використання. Для реалізації структурно-функціонального моделювання було побудовано діаграму в нотації IDEF0, а саме контекстну діаграму процесу проходження навчального web-тренажера з вивчення чисельного методу Ньютона її декомпозицію, яка складається з п'яти підпроцесів. Також було побудована база даних у вигляді ER-діаграми та наведено опис їх таблиць. В результаті побудови діаграми варіантів використання було показано зв'язок між акторами та функціями web-тренажеру.

Проект був розроблений на віртуальному веб-сервері OpenServer. Також в проекті використовувався менеджер модулів для PHP – Composer. Розробка web-тренажера починалася зі створення нового проекту з використанням Framework Laravel.

Завершенням стало наведення прикладів реалізації у вигляді зображень кожної сторінки та функції готового web-тренажеру.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Використання Visual Studio Code для розробки в PowerShell // URL: <https://docs.microsoft.com/ru-ru/powershell/scripting/dev-cross-plat/vscode/using-vscode?view=powershell-7> (дата звернення: 17.04.2020).
2. Методология IDEF0 // URL: <https://itteach.ru/bpwin/metodologiya-idef0> (дата звернення: 25.04.2020).
3. Основы работы с базами данных (Laravel 5.4) — Laravel ... // URL: <https://laravel.su/docs/5.4/database> (дата звернення: 03.05.2020).
4. Основы HTML - HTML5Book // URL: <https://html5book.ru/osnovy-html/> (дата звернення: 18.04.2020).
5. Решение систем нелинейных уравнений<br>Метод Ньютона // URL: [https://old.math.tsu.ru/EEResources/cm/text/3\\_2\\_1.htm](https://old.math.tsu.ru/EEResources/cm/text/3_2_1.htm) (дата звернення: 03.04.2020).
6. Работа в phpMyAdmin – ИТ-Проффи // URL: <https://itproffi.ru/rabota-v-phpmyadmin/> (дата звернення: 28.04.2020).
7. Розв'язання систем нелінійних рівнянь // URL: [https://web.posibnyky.vntu.edu.ua/fksa/2kvetnyj\\_komp'yuterne\\_modelyuvannya\\_system\\_procesiv/t1/316..htm](https://web.posibnyky.vntu.edu.ua/fksa/2kvetnyj_komp'yuterne_modelyuvannya_system_procesiv/t1/316..htm) (дата звернення: 18.04.2020).
8. Создание ER-диаграмм (диаграмм сущность-связь) // URL: <http://inf-teh-lotos.ru/sozдание-er-diagramm> (дата звернення: 25.04.2020).
9. Create IDEF0 diagrams - Visio - Office support - Office 365 // URL: <https://support.microsoft.com/en-us/office/create-idef0-diagrams-ea7a9289-96e0-4df8-bb26-a62ea86417fc?ui=en-us&rs=en-us&ad=us> (дата звернення: 18.04.2020).
10. CSS / Методология / БЭМ // URL: <https://ru.bem.info/methodology/css/> (дата звернення: 21.04.2020).
11. JavaScript // URL: <https://getbootstrap.com/docs/3.4/javascript/> (дата звернення: 25.04.2020).

12. MySQL // URL: <https://www.mysql.com/> (дата звернення: 03.05.2020).
13. PHP фреймворк Laravel. Обзор, плюсы, минусы // URL: <http://unetway.com/blog/laravel-framework-review/> (дата звернення: 03.05.2020).
14. PHP: Что такое PHP? // URL: <https://www.php.net/manual/ru/intro-what-is.php> (дата звернення: 25.04.2020).
15. PHP Programming with Visual Studio Code // URL: <https://code.visualstudio.com/docs/languages/php> (дата звернення: 02.05.2020).
16. UML Use Case Diagram Tutorial | Lucidchart // URL: <https://www.lucidchart.com/pages/uml-use-case-diagram> (дата звернення: 24.04.2020).
17. Visual Studio Code - Code Editing. Redefined // URL: <https://code.visualstudio.com/> (дата звернення: 01.05.2020).
18. What is Data Flow Diagram? - Visual Paradigm // URL: <https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/> (дата звернення: 026.04.2020).
19. What is Use Case Diagram? // URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/> (дата звернення: 25.04.2020).
20. What is WBS (Work Breakdown Structure) in Project Management? // URL: <https://www.workamajig.com/blog/guide-to-work-breakdown-structures-wbs> (дата звернення: 25.04.2020).

## **Додаток А**

**Технічне завдання  
на розробку інтерактивного додатку з вивчення чисельного методу  
Ньютона**

**Суми 2020**

## **1 Призначення й мета створення інтерактивного додатку з вивчення чисельного методу Ньютона**

### **1.1 Призначення інтерактивного додатку**

Інтерактивний додаток повинен представляти комп'ютерний тренажер з вивчення чисельного методу Ньютона.

### **1.2 Мета створення інтерактивного додатку з вивчення чисельного методу Ньютона**

Удосконалення навчального процесу завдяки практико-орієнтованому підходу та підвищення рівня засвоєння знань студентами при вивченні теми «Застосування чисельного методу Ньютона для вирішення систем нелінійних рівнянь».

### **1.3 Цільова аудиторія**

У цільовій аудиторії інтерактивного додатку можна виділити наступні групи:

1. Абітурієнти.
2. Студенти.
3. Викладачі.
4. Співробітники.
5. Аспіранти.
6. Інші зацікавлені користувачі.

## **2 Вимоги до інтерактивного додатку**

### **2.1 Вимоги до додатку в цілому**

#### **2.1.1 Вимоги до структури й функціонування інтерактивного додатку**

Інтерактивний додаток повинен бути реалізований у вигляді web-тренажеру та інтегрований до університетської системи електронного навчання elearning. Web-тренажер повинен мати необхідний набір функціональних можливостей та складатися із взаємозалежних розділів із чітко розділеними функціями. Користувачі web-додатку мають вільний доступ до калькулятора, використовуючи його для вирішення поставлених завдань. Студент самостійно має вводити своє прізвище та ініціали, в результаті чого йому присвоюється певний варіант завдання. Також у користувача, у процесі роботи, є можливість обміну онлайн-повідомленнями з викладачем.

#### **2.1.2 Вимоги до користувачів**

Для проходження тестування у web-тренажері користувач повинен володіти базовими навичками персонального комп'ютера та стандартним веб-браузером, для цього йому не потрібні професійні вміння.

#### **2.1.3 Вимоги до збереження інформації**

Уся інформація буде зберігатися у базі даних, реалізованої засобами системи управління базами даних MySQL.

#### **2.1.4 Вимоги до розмежування доступу**

Web-тренажер інтегрований у середовище університетської системи електронного навчання elearning.

Користувачів Web-тренажер можна розділити на 2 групи відповідно до прав доступу:

1. Студенти, зареєстровані у системі elearning



Розробник web-тренажеру

**Зареєстровані студенти** мають можливість запустити та виконати всі рівні тренажеру.

**Розробник** може удосконалювати функціонал тренажеру та коригувати інформацію в базі даних.

## **2.2 Вимоги до функцій, виконуваних web-тренажером**

### **2.2.1 Основні вимоги**

#### **2.2.1.1 Структура web-тренажера**

Web-тренажер повинен складатися з таких розділів:

Головна сторінка.

Вікно даних (прізвище та ініціали).

Вікно калькулятора.

Вікно зворотного зв'язку (допомога викладача).

Вікно завдань.

Вікно результату.

#### **2.2.1.2 Навігація**

Користувацький інтерфейс web-тренажеру повинен забезпечувати наочне, інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації, швидкий і логічний перехід до розділів.

Навігаційні елементи повинні забезпечувати для користувачів простоту та зручність використання.

### **2.2.2 Вимоги до функціональних можливостей**

#### **2.2.2.1 Функціональні можливості розділів**

На головній сторінці представлені такі функціональні елементи:

- початок виконання тренажеру;
- перехід до калькулятора;
- скористатися допомогою викладача.

На початку виконання тренажеру з'являється форма для введення прізвища та ініціалів, у результаті чого користувачу присвоюється певний варіант завдання.

Вікно калькулятора містить:

- звичайний калькулятор, яким користувач може користуватися під час виконання завдань.

Тренажер містить:

- 10 кроків в одному варіанті;
- випадючі списки для вибору правильного варіанту відповіді;
- кнопку для підказок зі спливаючим текстом;
- кнопку для переходу до наступного кроку;
- кнопка для завершення тестування.

Вікно зворотного зв'язку (допомога викладача):

- чат між студентом та викладачем, у якому студент має можливість звернутися до викладача за допомогою.

Вікно результату буде виводити кінцеву оцінку у відсотковому відношенні.

## **2.3 Вимоги до видів забезпечення**

### **2.3.1 Вимоги до інформаційного забезпечення**

У якості інструменту для реалізації web-тренажеру було обрано такі web-технології:

- JavaScript
- HTML
- CSS3
- MySQL

### **2.3.2 Вимоги до лінгвістичного забезпечення**

Web-тренажер повинен підтримувати українську, російську та англійську мови. Користувач, під час виконання тренажеру повинен мати можливість перемикатися на необхідну мову.

### **2.3.3 Вимоги до програмного забезпечення**

Для забезпечення нормальної роботи тренажеру потрібно, щоб було в наявності базове програмне забезпечення:

- Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище.

### **2.3.4 Короткий опис послідовності роботи тренажеру**

При завантаженні web-тренажеру користувач отримує доступ до головної сторінки, де може переглянути інструкцію, щодо використання тренажеру. Для проходження завдань, необхідно у головному вікні натиснути на кнопку «Почати тренажер» та заповнити інформаційне поле, увівши своє прізвище та ініціали. Після заповнення необхідних даних, користувачу присвоюється варіант і він автоматично переходить до виконання відповідних завдань.

### **2.3.5 Вимоги до надійності**

Робота з тренажером не вимагає від користувача спеціальної технічної підготовки.

- Web-тренажер здійснює автоматичну перевірку правильності виконання завдання на кожному кроці (невірні відповіді виділяються червоним кольором, вірні - зеленим).
- Перехід до наступного кроку повинен відбуватись за умови правильного виконання попереднього кроку.

- Якщо на поточному кроці допущено помилку, перехід на наступний етап не здійснюється та виводиться повідомлення «Не все виконано вірно! Перевірте та виправте помилки!»
- Штрафні бали нараховуються тільки за повторні спроби виконання на кожному кроці.

### 3 Склад і зміст робіт зі створення web-тренажера

Докладний опис етапів роботи зі створення інтерактивного додатку наведено в табл. А.1.

Таблиця А.1 – Етапи створення інтерактивного додатку

№п/п	Склад і зміст робіт	Строк розробки (у робочих днях)
1.	Розробка каркасу: Проектування розмітки та наповнення web - тренажеру	2 день
2.	Введення даних(прізвище та ініціали): Реалізація блоку введення даних у web - тренажері	1 день
3.	Розробка завдань: Розроблення та реалізація блоку для проходження завдань	4 день
4.	Калькулятор: Реалізація звичайного калькулятора у окремому вікні по натисканню на кнопку правої нижньої панелі	1 день
5.	Вивід результату: Реалізація виводу кінцевого результату у відсотковому відношенні по натисканню на кнопку «Завершити тестування»	2 день

6.	Зворотній зв'язок: Розроблення та реалізація блоку Зворотнього зв'язку(чат з викладачем)	3 дні
8.	Зміна мови: Реалізація переключення мови серед трьох зазначених у вимогах варіантів	1 день
9.	Завершення роботи: Проведення стилістичних виправлень веб-тренажеру, перевірка (тестування) реалізованого функціоналу	1 день
	<b>Загальна тривалість робіт (з урахуванням налагодження й виправлення помилок) і строк закінчення проекту</b>	<b>15 днів</b>

#### **4 Вимоги до складу й змісту робіт із введення web-тренажера в експлуатацію**

Для створення умов функціонування інтерактивного додатку з вивчення методу Ньютона повинен бути проведений певний комплекс заходів.

Для інтегрування web-тренажеру до університетської системи elearning необхідно, щоб параметри web-тренажеру відповідали вимогам, зазначеним у ТЗ.

## Додаток Б

### Планування робіт

#### 1 Ідентифікація ідеї проекту

Інформаційні технології активно задіяні в усіх життєвих процесах сучасного суспільства.

З розвитком веб-технологій на просторах Інтернету з'явилась велика кількість веб-тренажерів та серверів, які були створенні для реалізації будь-яких потреб користувача. На сьогоднішній день велика кількість веб-тренажерів допомагають користувачам засвоїти отримані теоретичні знання, підвищити свій рівень знань, активізують їхню розумову діяльність.

Метою інтерактивного додатку дипломного проекту є удосконалення навчального процесу завдяки практико-орієнтованому підходу та підвищення рівня засвоєння знань студентами при вивченні теми «Застосування чисельного методу Ньютона для вирішення систем нелінійних рівнянь».

Розроблений web-додаток призначений для навчання, відпрацювання отриманих навичок, оцінювання та підвищення рівня засвоєння знань студентів.

Інтерактивний додаток повинен бути реалізований у вигляді web-тренажеру та інтегрований до університетської системи електронного навчання elearning. Web-тренажер повинен мати необхідний набір функціональних можливостей та складатися із взаємозалежних розділів із чітко розділеними функціями.

#### 2 Деталізація мети методом SMART

Технологія SMART (SMART) - сучасний підхід до постановки працюючих цілей. Система постановки SMART - цілей дозволяє на етапі визначення мети узагальнити

всю наявну інформацію, встановити прийнятні терміни роботи, визначити достатність ресурсів, надати всім учасникам процесу ясні, точні, конкретні завдання.

SMART є аббревіатурою, розшифровка якої: конкретна (Specific), вимірювана (Measurable), досяжна (Achievable), реалістична (Relevant), обмежена у часі (Time-framed). Кожна буква аббревіатури SMART означає критерій ефективності поставлених цілей.

Розглянемо кожен критерій SMART мети більш докладно.

S – конкретність.

Сформульована мета повинна бути конкретною, що збільшує ймовірність її досягнення.

M – вимірюваність.

Передбачає показників вартості які вимірюються. При відсутності фізичних способів та інструментарію виміру використовуються експерти – як інструмент для виміру.

A – узгодженість.

Встановлює, що мета повинна впливати з реальних проблем, місії, стратегічних планів, планів розвитку, а також узгоджуватись з інтересами зацікавлених сторін проекту.

R – реалістичність.

Цілі повинні бути досяжні, так як реалістичність виконання завдання впливає на мотивацію виконавця. Досяжність мети визначається на основі власного досвіду з урахуванням всіх наявних ресурсів і обмежень.

T – обмеженість в часі.

Мета повинна бути обмежена по виконанню в часі, а значить повинен бути визначений фінальний термін, перевищення якого говорить про невиконання мети. Встановлення тимчасових рамок і кордонів для виконання мети дозволяє зробити процес управління контрольованим.

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити інтерактивний додаток з вивчення чисельного метода Ньютона
Measurable (вимірювання)	Розробити та реалізувати якісний програмний продукт, використовуючи для цього мінімум ресурсів.
Achievable (досяжна, узгоджена)	Ціль даного проекту вважається досяжною, оскільки розробник володіє необхідними навичками у створенні веб-сторінок засобами мов html, css, javascript.
Relevant (реалістична)	Для реалізації продукту проекту є всі необхідні технічні та програмні засоби, доступ до мережі Інтернет. Розробник досить кваліфікований для виконання поставлених задач.
Time-framed (обмежена в часі)	Web-тренажер розроблюється з обмеженням у часі на основі сформованого календарного плану та матриці відповідальності.

### 3 Описання фази розробки ІТ—проекту

#### 3.1 Планування змісту структури робіт ІТ—проекту (WBS)

Основним інструментом для планування змісту структури робіт служить WBS(Work Break Structure) - представлення проекту, виконане у вигляді ієрархічної структури робіт, що досягається за допомогою послідовної декомпозиції. Інструмент спрямований на детальне планування, оцінку вартості, визначення та розподіл персональної відповідальності виконавців та інші - тобто, на основні роботи і результати, що визначають зміст проекту.



Під час побудови WBS відбувається послідовне розбиття проекту на підпроекти, роботи різних рівнів, детальні робочі пакети. Декомпозиція повинна бути коректною, тобто елементи будь-якого рівня WBS повинні бути необхідними та достатніми для створення відповідного елемента верхнього рівня.

Ієрархічна структура робіт в основному являє собою перелік завдань проекту. Вона може бути представлена графічно або у формі опису, який показує включення робіт. Ієрархічна структура робіт організовує і визначає весь зміст проекту. Роботи, не включені в WBS, не є роботами проекту.

Виконаємо побудову WBS структури, у якій зазначимо всі виконувані роботи в залежності від головних етапів:

1. **Формування технічного завдання** — розробка технічного завдання, що встановлює основне призначення, показники якості, техніко-економічні та спеціальні вимоги до розроблюваного інструментального засобу. Формування технічного завдання включає в себе підпункти (аналіз предметної області, постановка задачі, функціональні вимоги до ПП, формування етапів розробки продукту).

2. **Реалізація програмного продукту** - написання відповідних модулів, що забезпечують функціонування програмного продукту.

2.1. Розробка інтерфейсу програмного продукту.

2.2. Розробка стилю.

2.3. Наповнення web-тренажера.

2.4. Розробка калькулятора;

2.5. Проектування БД програмного продукту.

2.6. Розробка зворотнього зв'язку(чат з викладачем).

2.7. Програмування JavaScript та HTML.

3. **Тестування – перевірка роботи програмного продукту, виявлення помилок.**

3.1. Тестування розробником.

3.2. Тестування незалежною особою.

4. **Завершення програмного продукту.**

4.1. Оформлення документації.

4.2. Архівація програмного продукту.

4.3 Інтегрування web-тренажера до університетської системи elearning.

WBS-структура для даного проекту представлена на рис. Б.1

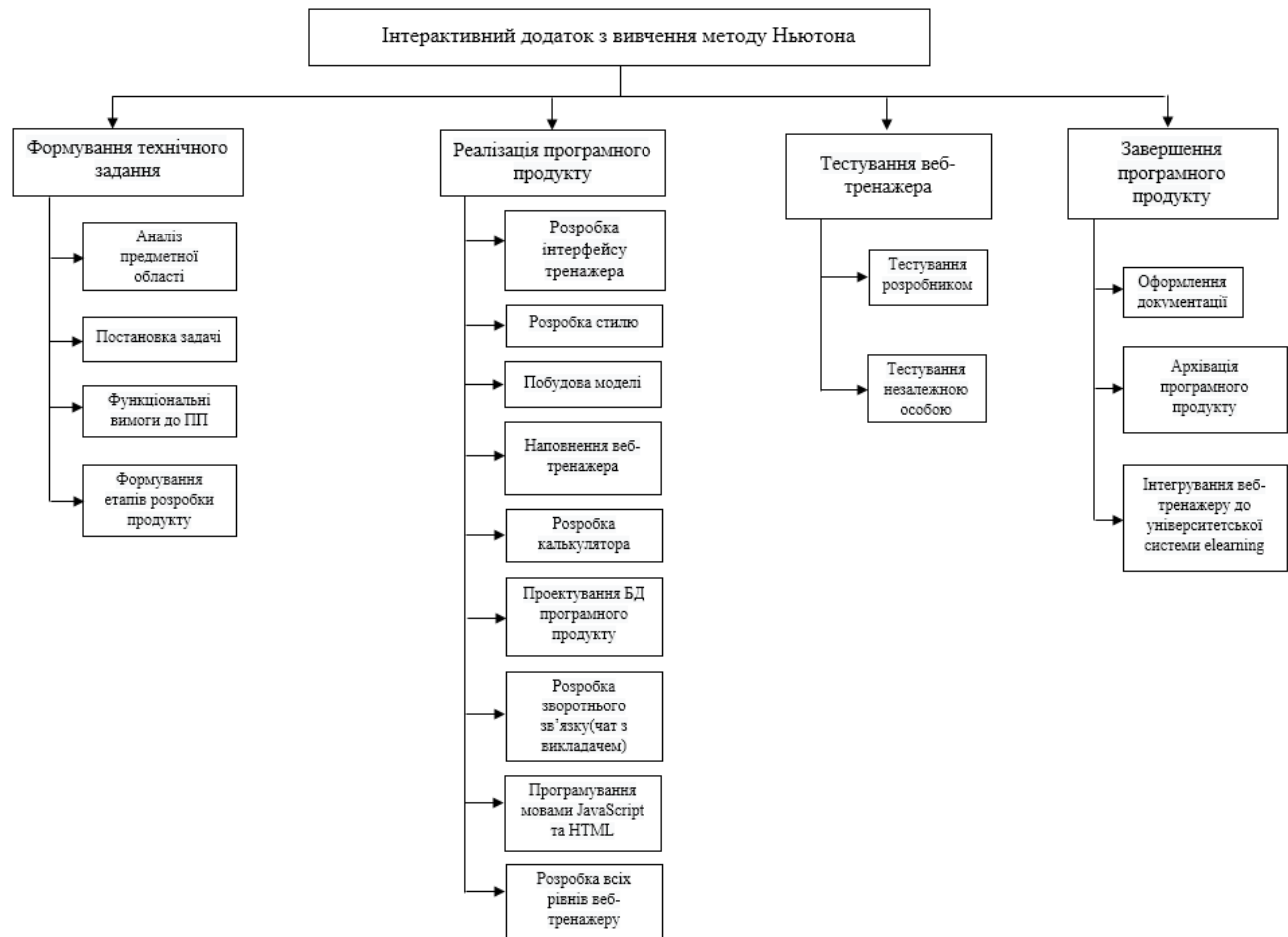


Рисунок Б.1 - WBS-структура інтерактивного додатку з вивчення методу Ньютона

### 3.2 Планування структури організації, для впровадження готового проекту (OBS)

OBS-структура проекту – організаційна структура виконавців (організацій) проекту. Визначається за переліком пакетів робіт нижнього рівня кожної гілки WBS—структури. Представляється відповідальними (відповідальні – це не обов’язково

керівники організацій (відділів), а ті люди які безпосередньо організують виконання робіт) за виконання пакетів робіт.

Організаційна структура представляє собою графічне відображення учасників проекту та їх відповідальних осіб, які задіяні в реалізації проекту. На верхньому рівні OBS розташована команда проекту.

На наступному рівні фіксуються виконавці: організації, відділи тощо. Потім, рівнем нижче, для кожного виконавця вказують прізвища конкретних осіб, які будуть відповідати за виконання елементарних робіт WBS. Потрібно пам'ятати, що відповідальні – це не обов'язково керівники, а ті співробітники, які безпосередньо організують і відповідають у виконавця за виконання елементарної роботи, зазначеної у WBS. Для них ця елементарна робота також є проектом (у порівнянні з загальним проектом). Для себе вони також можуть побудувати WBS— структуру й застосовувати інші інструменти планування.

OBS-структура для даного проекту представлена на рисунку Б.2

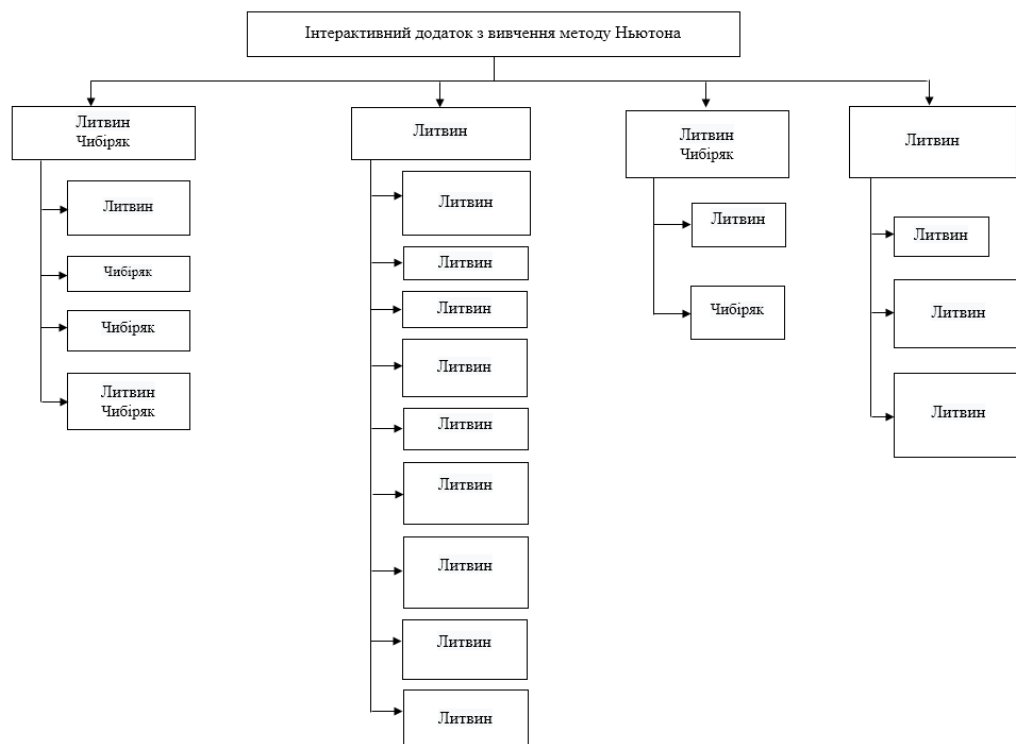


Рисунок Б.2 - Організаційна структура виконавців

### 3.3 Побудова матриці відповідальності (виконавців пакетів робіт)

На підставі OBS та WBS структур було побудовано матрицю відповідальності. Для кожного з виконавців була визначена його роль:

- відповідальний (В) – повністю відповідає за виконання задачі та має право приймати рішення щодо способу її реалізації;
- консультант (К) – наглядає за ходом виконання завдання і висловлює свої міркування стосовно способу та якості реалізації. Несе відповідальність, якщо не помітить явного недоліку.
- спостерігач (С) – те ж саме що і консультант, але відповідальності не несе.

Матриця відповідальності представлена в табл. Б.2

Таблиця Б.2 – Матриця відповідальності

WBS\OBS	Литвин І.В.	Чибіряк Я.І.	Рецензент
1 Формування технічного завдання	В	К	
1.1 Аналіз предметної області	В	К	
1.2 Постановка задачі	В	К	
1.3 Функціональні вимоги до ПП	В	К	
1.4 Формування етапів розробки продукту	В	К	
2 Реалізація програмного продукту	В	С	
2.1 Розробка інтерфейсу тренажера	В	С	
2.2 Розробка стилю	В	С	
2.3 Побудова моделі	В	С	
2.4 Наповнення web-тренажера	В	С	
2.5 Розробка калькулятора	В	С	
2.6 Проектування БД програмного	В	С	

продукту			
2.7 Розробка зворотнього зв'язку(чат з викладачем)	В	С	
2.8 Програмування мовами JavaScript та HTML	В	С	
2.9 Розробка всіх рівнів web-тренажеру	В	С	
3 Тестування web-тренажера	В	К	С
3.1 Тестування розробником	В	С	С
3.2 Тестування незалежною особою	С	В	С
4 Завершення програмного продукту	В	К	
4.1 Оформлення документації	В	К	
4.2 Архівація програмного продукту	В	К	
4.3 Інтегрування web-тренажера до університетської системи elearning	В	К	

#### **4 Побудова календарного графіку виконання ІТ—проекту (включаючи побудову часткових мережевих моделей у вигляді діаграм Ганта)**

Графіки і діаграми, спеціальні програми і додатки - це інструменти, які допомагають учасникам проекту керувати своїми процесами і робити їх краще. Мережа діаграма Ганта вважається якісним інструментом для відображення цілей та задач.

Діаграма Ганта – горизонтальна лінійна діаграма, на якій задачі проекту представляються протяжними в часі відрізками, що характеризуються датами початку та закінчення, затримками і, можливо, іншими тимчасовими параметрами.

Управління проектами з діаграмами Ганта засноване на форматі гістограм. Це допомагає відслідковувати відсоток робіт, виконаних по кожному завданню. Основна увага діаграм Ганта зосереджено на процентному завершенні кожного завдання. Для отримання реального уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, на підставі часткової мережевої моделі було побудовано календарний графік робіт.

Завдяки засобам програмного продукту MS Project була розроблена діаграма Ганта, яка у вигляді гістограми, яка відображає тривалість кожного процесу, що був визначений на етапі формування WBS.

Графік виконання дипломного проекту представлено у вигляді Діаграми Ганта на рисунку Б.3.

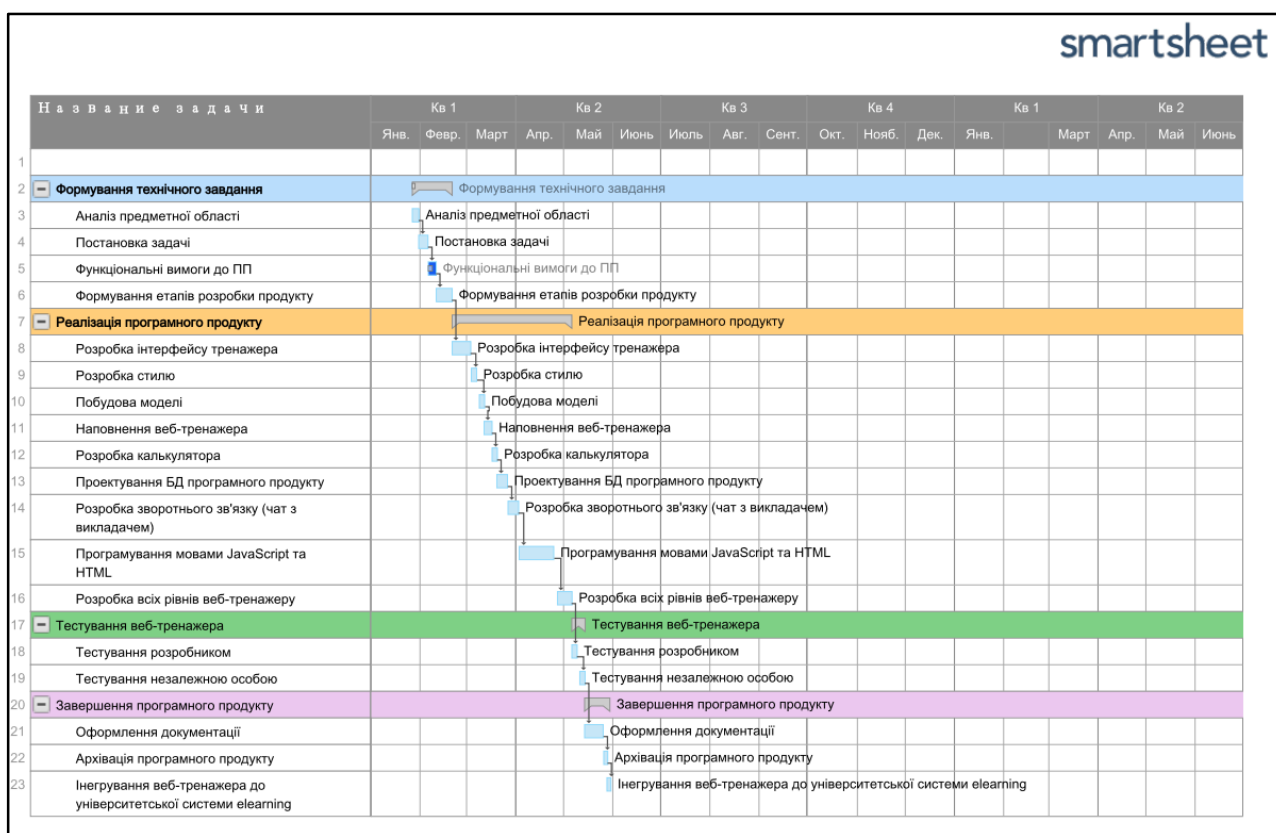


Рисунок Б.3 - Діаграма Ганта

#### 4.1 Ідентифікація ризиків

Ризик – ймовірнісна подія, яка може позитивно чи негативно вплинути на проект. Причиною виникнення ризиків є невизначеності, існуючі в кожному проекті.

Управління ризиком – це процес реагування на події та зміни ризиків у процесі виконання проекту. При цьому важливим є проведення моніторингу ризиків.

Процес управління ризиками включає в себе такі пункти:

- 1) Ідентифікація ризиків (виявлення ризиків)
- 2) Оцінювання ризиків (оцінка ймовірності та впливу)
- 3) Заходи реагування на ризики
- 4) Моніторинг ризиків

Ідентифікація ризиків визначає, які ризики здатні вплинути на проект, і документує характеристики цих ризиків. Ідентифікація ризиків не буде ефективною, якщо вона не буде проводитися регулярно протягом реалізації проекту.

В процесі аналізу для визначення числових значень ймовірності виникнення ступеня впливу, зазвичай застосовується метод експертних оцінок. На їх основі визначається ранг ризику, як потенційний вплив ризику на проект, який оцінюється як добуток ймовірності виникнення та ступеню впливу.

Ідентифікація ризиків повинна залучати якомога більше учасників: менеджерів проекту, замовників, користувачів, незалежних фахівців.

На рис. Б.3 розташовується класифікація ризиків.

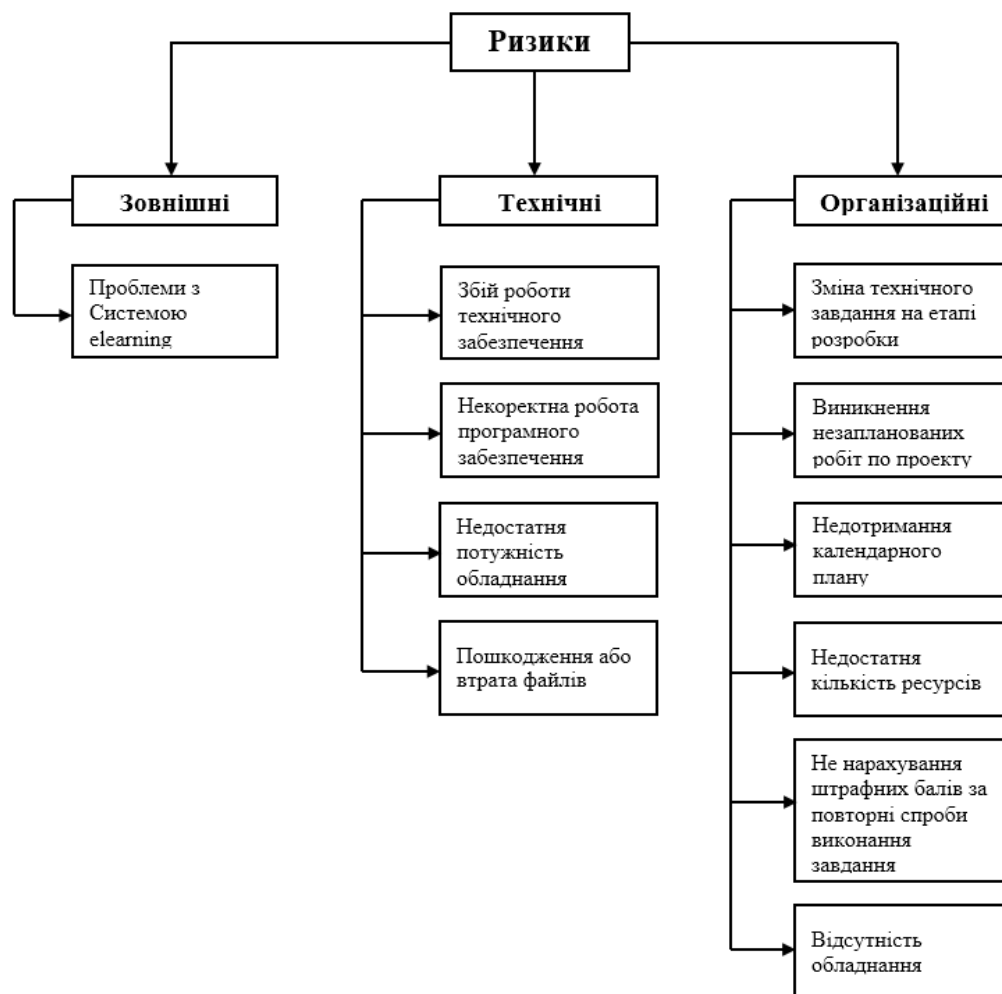


Рисунок Б.4 – Ризики

## 6.1 Матриця ризиків

Ризики представити за допомогою RBM матриці (Risk Breakdown Matrix) на рис.

Б.5.

Класифікація ризиків за імовірністю виникнення:

- слабоімовірнісні;
- малоімовірнісні;
- імовірні;
- досить імовірні;



– майже імовірні.

Класифікація ризиків за імовірністю виникнення за величиною втрат:

- мінімальна;
- низька
- середня
- висока
- максимальна

Виконаємо класифікацію ризиків даного проекту. Для цього складемо табл. Б.3.

Таблиця Б.3 – Класифікація ризиків дипломного проекту

Ризик		ймовірність виникнення	величина втрат
Проблеми з системою elearning	R1	3	4
Збій роботи технічного забезпечення	R2	3	2
Некоректна робота програмного забезпечення	R3	2	5
Недостатня потужність обладнання	R4	2	2
Пошкодження або втрата файлів	R5	3	2
Зміна технічного завдання на етапі розробки	R6	2	3
Виникнення незапланованих робіт по проекту	R7	3	3
Недотримання календарного плану	R8	2	4
Недостатня кількість ресурсів	R9	1	3
Не нарахування штрафних балів за повторні спроби	R10	1	5

виконання задання			
Відсутність обладнання	R11	5	5

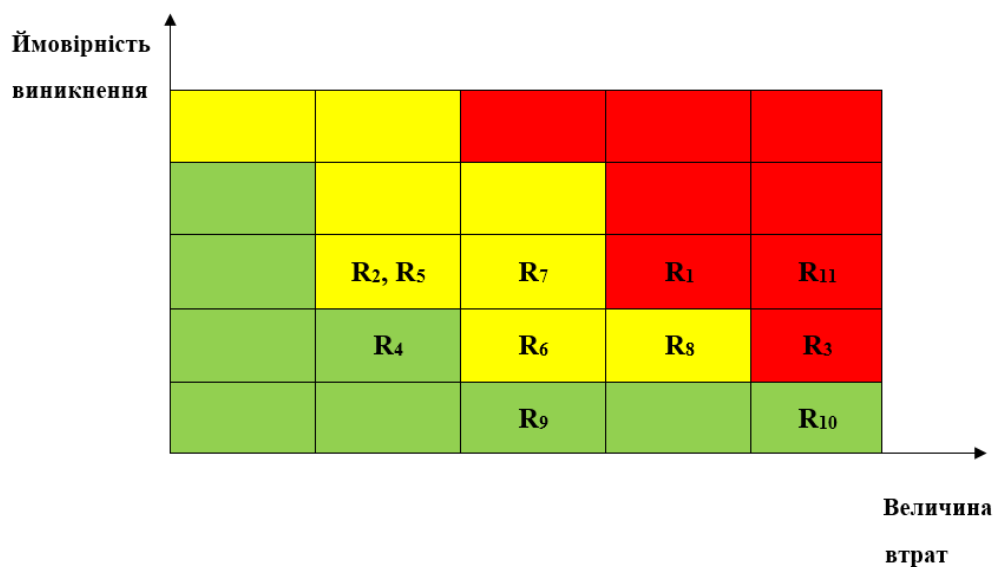


Рисунок Б.5 – Матриця імовірності втрат

## 6.2 Класифікація за ступенем впливу та за рівнем ризику (табл. Б.4)

Класифікація за ступенем впливу:

- ігноровані ( $1 \leq R \leq 4$ );
- незначні ( $5 \leq R \leq 8$ );
- помірні ( $9 \leq R \leq 11$ );
- вагомі ( $12 \leq R \leq 19$ );
- критичні ( $20 \leq R \leq 25$ ).

Класифікація за рівнем ризику:

- прийнятні ризику;
- виправданні ризику;
- недопустимі ризику;

Таблиця Б.4 – Класифікація за ступенем впливу та за рівнем ризику

Ризик		Ступінь впливу	Рівень ризику
Проблеми з системою elearning	R1	12	недопустимі ризики
Збій роботи технічного забезпечення	R2	6	виправданні ризики
Некоректна робота програмного забезпечення	R3	10	недопустимі ризики
Недостатня потужність обладнання	R4	4	прийнятні ризики
Пошкодження або втрата файлів	R5	15	недопустимі ризики
Зміна технічного завдання на етапі розробки	R6	6	виправданні ризики
Виникнення незапланованих робіт по проекту	R7	6	виправданні ризики
Недотримання календарного плану	R8	9	виправданні ризики
Недостатня кількість ресурсів	R9	8	виправданні ризики
Не нарахування штрафних балів за повторні спроби виконання завдання	R10	3	прийнятні ризики
Відсутність обладнання	R11	5	виправданні ризики

Після виконання прогнозування виникнення ризиків та їх ступінь впливу на результат реалізації проекту, були розроблені варіанти запобігання.

- вибір потужного обладнання для виконання проекту;
- резервувати час на випадок виникнення непередбачуваних помилок;
- притримування плану роботи;
- використовувати програми страхування технічних ризиків;
- ретельний вибір інструментів виконання проекту.

## Додаток В

```
App.js
import router from "./router";
import Vuetify from "vuetify";
import VueSweetalert2 from 'vue-sweetalert2';
import 'sweetalert2/dist/sweetalert2.min.css';
import uk from 'vuetify/src/locale/uk.ts';
import AppComponent from "./components/AppComponent";
import * as VueWindow from '@hscmap/vue-window'
```

```
window.Vue = require('vue');
window.Vue.use(VueWindow);
window.Vue.use(Vuetify);
window.Vue.use(VueSweetalert2);
```

```
const app = new Vue({
  el: '#app',
  components: {
    AppComponent
  },
  vuetify: new Vuetify({
    icons: {
      iconfont: 'mdi',
    },
    lang: {
      locales: { uk },
      current: 'uk',
    },
  }),
  router
});
```

```
Router.js
import Vue from "vue";
import Router from "vue-router";
import HomeComponent from "./components/HomeComponent";
import TestComponent from "./components/TestComponent";
import LoginComponent from "./components/LoginComponent";
import RegisterComponent from "./components/RegisterComponent";
import ProfileComponent from "./components/ProfileComponent";
import HelpComponent from "./components/HelpComponent";
import ResultsComponent from "./components/ResultsComponent";
import DashboardComponent from "./components/admin/DashboardComponent";
import UsersComponent from "./components/admin/UsersComponent";
import QuestionComponent from "./components/admin/QuestionComponent";
```

```
Vue.use(Router);

let router = new Router({
  mode: "history",
  routes: [
    {
      path: '/',
      name: 'home',
      component: HomeComponent,
      meta: {
        requiresAuth: true
      }
    },
    {
      path: '/test',
      name: 'test',
      component: TestComponent,
      meta: {
        requiresAuth: true
      }
    },
    {
      path: '/login',
      name: 'login',
      component: LoginComponent,
      meta: {
        guest: true
      }
    },
    {
      path: '/register',
      name: 'register',
      component: RegisterComponent,
      meta: {
        guest: true
      }
    },
    {
      path: '/profile',
      name: 'profile',
      component: ProfileComponent,
      meta: {
        requiresAuth: true
      }
    },
    {
      path: '/help',
```

```

    name: 'help',
    component: HelpComponent,
    meta: {
      requiresAuth: true
    }
  },
  {
    path: '/results',
    name: 'results',
    component: ResultsComponent,
    meta: {
      requiresAuth: true
    }
  },
  {
    path: '/admin',
    name: 'admin',
    component: DashboardComponent,
    meta: {
      requiresAuth: true,
      is_admin : true
    }
  },
  {
    path: '/admin/users',
    name: 'users',
    component: UsersComponent,
    meta: {
      requiresAuth: true,
      is_admin : true
    }
  },
  {
    path: '/admin/question',
    name: 'question',
    component: QuestionComponent,
    meta: {
      requiresAuth: true,
      is_admin : true
    }
  },
]
});

router.beforeEach((to, from, next) => {
  if(to.matched.some(record => record.meta.requiresAuth)) {
    if (localStorage.getItem('jwt') == null) {
      next({

```

```

    path: '/login',
    params: { nextUrl: to.fullPath }
  })
} else {
  let user = JSON.parse(localStorage.getItem('user'))
  if(to.matched.some(record => record.meta.is_admin)) {
    if(user.is_admin == 1) {
      next()
    }
    else {
      window.location.href = '/';
    }
  }else {
    next()
  }
}
} else if(to.matched.some(record => record.meta.guest)) {
  if(localStorage.getItem('jwt') == null){
    next()
  }
  else{
    next({ name: 'register'})
  }
} else {
  next()
}
})

```

export default router

TestComponent.vue

```

<template>
  <div>
    <div class="calculator" id="calculator"></div>
    <div v-if="preloader" class="preloader">
      
    </div>

    <v-dialog v-model="dialogHelp" max-width="400px">
      <v-card class="py-4">
        <v-card-text>
          <v-textarea
            class="mb-0"
            v-model="userQuestion"
            label="Запитання"
            rows="3"
            required
            outlined

```

```

        dense
      ></v-textarea>
    </v-card-text>
    <v-card-actions>
      <v-spacer></v-spacer>
      <v-btn color="blue darken-1" text @click="dialogHelp = false">Закрити</v-btn>
      <v-btn color="blue darken-1" :disabled="userQuestion == "" text
@click="sendQuestion()">Надіслати</v-btn>
    </v-card-actions>
  </v-card>
</v-dialog>

```

```

<v-stepper
  alt-labels
  v-model="e1"
>
  <template>
    <v-stepper-header class="elevation-0">
      <template v-for="n in steps">
        <v-stepper-step
          :key="`${n}-step`"
          :complete="e1 > n"
          :step="n"
        >
          Крок {{ n }}
        </v-stepper-step>

        <v-divider
          v-if="n !== steps"
          :key="n"
        ></v-divider>
      </template>
    <template>
      <v-stepper-step
        :key="`${steps+1}-step`"
        :complete="e1 > steps+1"
        :step="steps+1"
      >
        Результат
      </v-stepper-step>
    </template>
  </v-stepper-header>

  <v-divider></v-divider>

  <v-stepper-items>
    <v-stepper-content
      v-for="n in steps"

```



```

:key=""${n}-content`
:step="n"
>
<v-card
  class="mb-5 elevation-0"
  >
  
  <v-divider class="my-2"></v-divider>
  <v-row>
    <v-col>
      <b>Опис завдання:</b> {{tasks[n-1].description}}.
      <div v-if="tasks[n-1].help">
        <v-btn outlined class="mt-2" @click="openHelp = !openHelp">Підказка</v-btn>
        <p v-if="openHelp" class="mt-2">{{tasks[n-1].help}}</p>
      </div>
      <v-divider class="my-2"></v-divider>
      <div v-for="answer in tasks[n-1].answers" :key="answer.id">
        <div v-if="answer.type == 'inputText'">
          {{ answer.description }}:
          <v-text-field
            v-model="answer.userUnswer"
            class="errorInput"
            label="Відповідь"
            outlined
            dense
            :error="answer.valid === false"
            :success="answer.valid === true"
            style="width: 200px;margin-top: 5px;"
          ></v-text-field>
        </div>
        <div v-if="answer.type == 'inputSelect'">
          {{ answer.description }}:
          <v-select
            v-model="answer.userUnswer"
            :items="answer.variant_answer"
            item-text="value"
            value="isTrue"
            label="Варіанти відповіді"
            outlined
            dense
            :error="answer.valid === false"
            :success="answer.valid === true"
            style="width: 200px;margin-top: 5px"
          ></v-select>
        </div>
        <div v-if="answer.type == 'inputArray'">
          {{ answer.description }}:
          <v-row v-for="(row, index) in answer.userUnswer" :key="index">

```

```

<v-col v-for="(col, indexCol) in row" class="py-0" md="3" :key="indexCol">
  <v-text-field
    v-model="col.value"
    required
    outlined
    dense
    :error="col.valid === false"
    :success="col.valid === true"
  ></v-text-field>
</v-col>
</v-row>
</div>
<div v-if="answer.type == 'inputSequence'">
  {{ answer.description }}
  <v-row>
    <v-col md="1">
      <div style="height: 40px;padding-top: 10px" v-for="(variant_answer, index) in
answer.variant_answer" :key="index">
        {{ index+1 }}
      </div>
    </v-col>
    <v-col md="5">
      <draggable @change="clearValid(answer.userUnswer)" style="height: 100%" v-
model="answer.userUnswer" :options="{ group:'answer_'+answer.id}">
        <div style="height: 40px" v-for="(userUnswer, index) in answer.userUnswer"
:key="index">
          
        </div>
      </draggable>
    </v-col>
    <v-col md="5">
      <draggable style="height: 100%" v-model="answer.variant_answer_check"
:options="{ group:'answer_'+answer.id}">
        <div style="height: 40px" v-for="(variant_answer_check, index) in
answer.variant_answer_check" :key="index">
          
        </div>
      </draggable>
    </v-col>
  </v-row>

```

```

</div>

<div v-if="answer.type == 'inputTrueElement'">
  <v-row>
    <v-col md="2">
      {{ answer.description }}
    </v-col>
    <v-col md="5">
      <draggable style="height: 100%" v-model="answer.userUnswer"
:options="{group:'answer_'+answer.id}">
        
      </draggable>
    </v-col>
    <v-col md="5">
      <draggable style="height: 100%;width: 100%" v-model="answer.variant_answer"
:options="{group:'answer_'+answer.id}">
        <div v-for="(variant_answer, index) in answer.variant_answer" :key="index">
          
        </div>
      </draggable>
    </v-col>
  </v-row>
</div>

</div>
</v-col>
</v-row>
<v-divider class="mb-2"></v-divider>
</v-card>
<v-alert v-if="countErrorStep < 3 && errorAnswer" type="error">Не все виконано вірно!
Перевірте та виправте помилки!</v-alert>
<v-alert v-if="countErrorStep == 3" type="error">Погано! Крок провалений. Перейдіть до
наступного...</v-alert>
<v-row>

```

```

<v-col v-if="countErrorStep == 4">
  <v-btn
    class="buttonHome"
    block
    color="primary"
    v-if="n != steps"
    @click="nextStep(n)"
  >
    Наступний крок
  </v-btn>
</v-col>
<v-col v-if="countErrorStep < 4">
  <v-btn
    class="buttonHome"
    block
    color="primary"
    v-if="n != steps"
    @click="validationStep(n, tasks[n-1])"
  >
    Наступний крок
  </v-btn>
  <v-btn
    class="buttonHome"
    block
    color="primary"
    v-else
    @click="validationStep(n, tasks[n-1])"
  >
    Завершити тренажер
  </v-btn>
</v-col>
<v-col>
  <v-btn
    class="buttonHome"
    block
    color="orange"
    @click="dialogHelp = true"
  >
    Допомога викладача
  </v-btn>
</v-col>
<v-col>
  <v-btn
    class="buttonHome"
    block
    color="green"
    @click="openCalc"
  >

```

```

        Калькулятор
    </v-btn>
</v-col>
</v-row>
</v-stepper-content>
</v-stepper-items>
<v-stepper-content
  :key="'${steps+1}-content'"
  :step="steps+1"
>
  <v-card
    class="mb-5 elevation-0"
    color="lighten-1"
  >
    <v-row>
      <v-col md="8">
        <h1 class="titleHeader">Ваші результати</h1>
        <div class="title my-2"><b>Час виконання роботи:</b> {{ this.time }}</div>
        <div class="title my-2"><b>Підсумковий бал:</b> {{ (rating*100/5).toFixed(1)
}}%</div>
        <div class="title my-2"><b>Отримано балів:</b> {{ Math.round(rating) }}</div>
      </v-col>
      <v-col my="4">
        <h1 class="text-center">Статистика</h1>
        <v-row>
          <v-col class="text-center"><v-icon>list</v-icon> {{ tasks.length }}</v-col>
          <v-col class="text-center"><v-icon color="green">thumb_up</v-icon> {{
trueTasks.length }}</v-col>
          <v-col class="text-center"><v-icon color="red">thumb_down</v-icon> {{
falseTasks.length }}</v-col>
        </v-row>
        <v-divider></v-divider>
        <h1 class="text-center mt-4">Завдання</h1>
        <div v-for="(item, index) in tasks" :key="index">
          <span :class="item.isTrue ? 'green--text' : 'red--text'">{{ index+1 }} -
{{ item.description }}</span>
        </div>
      </v-col>
    </v-row>
    <v-row>
      <v-col>
        <v-btn
          class="buttonHome"
          block
          color="primary"
          @click="reload()"
        >

```

```

<script>
import Snackbar from './includes/Snackbar';
import Menu from './includes/Menu';
import validation from '../mixins/validation';

export default {
  mixins: [validation],
  data: () => ({
    preloader: true,
    tab: null,
    dialogAddOption: false,
    optionNumber: "",
    options: [],
    snackbar: {
      status: false,
      text: "",
    },
  }),

  components: {
    Snackbar,
    Menu
  },

  created() {
    this.getData();
  },

  methods: {
    getData() {
      axios.get('/api/options')
        .then((response) => {
          this.options = response.data;
        })
    },
    // Options
    addOption() {
      axios.post('/api/option', {
        title: "Вариант "+this.optionNumber
      })
        .then((response) => {
          this.dialogAddOption = false
          this.options.push(response.data)
        })
    },
    delOption(option) {
      axios.delete('/api/option/'+option.id)
        .then(() => {

```

```

        this.options.splice(this.options.indexOf(option), 1);
    })
},

// Tasks
addTask(option) {
    var idOption = this.options.indexOf(option);
    this.options[idOption].tasks.push({
        description: "",
        help: "",
        img: null
    })
},
previewFiles(option, task, event) {
    if (event) {
        var idOption = this.options.indexOf(option);
        var idTask = this.options[idOption].tasks.indexOf(task);
        var reader = new FileReader();
        reader.onload = (e) => {
            this.options[idOption].tasks[idTask].fileName = e.target.result.name;
            this.options[idOption].tasks[idTask].img = e.target.result;
        }
        reader.readAsDataURL(event);
    }
},
previewVariantAnswerFile(option, task, answer, index, event) {
    var input = event.target;
    var idOption = this.options.indexOf(option);
    var idTask = this.options[idOption].tasks.indexOf(task);
    var idAnswer = this.options[idOption].tasks[idTask].answers.indexOf(answer);
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        reader.onload = (e) => {
            this.options[idOption].tasks[idTask].answers[idAnswer].variant_answer[index].value =
e.target.result
        }
        reader.readAsDataURL(input.files[0]);
    }
},
saveTask(option, task) {
    var idOption = this.options.indexOf(option);
    var idTask = this.options[idOption].tasks.indexOf(task);
    if(task.id) {
        axios.post('/api/update-task/'+task.id, this.options[idOption].tasks[idTask])
        .then((response) => {
            this.snackbar.status = true;
            this.snackbar.text = "Збережено";
            this.getData();
        })
    }
}

```

```

    })
    .catch(() => {
      this.snackbar.status = true;
      this.snackbar.text = "Помилка";
    })
  } else {
    axios.post('/api/post-task/'+option.id, this.options[idOption].tasks[idTask])
    .then(() => {
      this.getData();
      this.snackbar.status = true;
      this.snackbar.text = "Додано";
    })
    .catch(() => {
      this.snackbar.status = true;
      this.snackbar.text = "Помилка";
    })
  }
},
delTask(option, task) {
  var idOption = this.options.indexOf(option);
  if(task.id) {
    axios.delete('/api/task/'+task.id)
    .then(() => {
      this.options[idOption].tasks.splice(this.options[idOption].tasks.indexOf(task), 1);
    })
  } else {
    this.options[idOption].tasks.splice(this.options[idOption].tasks.indexOf(task), 1);
  }
},
// answers
addFormAnswerText(option, task) {
  var idOption = this.options.indexOf(option);
  var idTask = this.options[idOption].tasks.indexOf(task);
  this.options[idOption].tasks[idTask].answers.push({
    type: "inputText",
    description: "",
    variant_answer: [
      {
        value: ""
      }
    ]
  })
},
addFormAnswerSelect(option, task) {
  var idOption = this.options.indexOf(option);
  var idTask = this.options[idOption].tasks.indexOf(task);

```



```

this.options[idOption].tasks[idTask].answers.push({
  type: "inputSelect",
  description: "",
  variant_answer: [
    {
      value: "",
      isTrue: false,
    }
  ]
})
},
addFormAnswerArray(option, task) {
  var idOption = this.options.indexOf(option);
  var idTask = this.options[idOption].tasks.indexOf(task);
  this.options[idOption].tasks[idTask].answers.push({
    type: "inputArray",
    description: "",
    cols: 1,
    rows: 1,
    variant_answer: []
  })
},
addFormAnswerSequence(option, task) {
  var idOption = this.options.indexOf(option);
  var idTask = this.options[idOption].tasks.indexOf(task);
  this.options[idOption].tasks[idTask].answers.push({
    type: "inputSequence",
    description: "",
    variant_answer: [
      {
        value: "",
        isTrue: false,
      }
    ]
  })
},
addFormAnswerTrueElement(option, task) {
  var idOption = this.options.indexOf(option);
  var idTask = this.options[idOption].tasks.indexOf(task);
  this.options[idOption].tasks[idTask].answers.push({
    type: "inputTrueElement",
    description: "",
    variant_answer: []
  })
},
delFormAnswer(option, task, answer) {
  var idOption = this.options.indexOf(option);
  var idTask = this.options[idOption].tasks.indexOf(task);

```

```

var idAnswer = this.options[idOption].tasks[idTask].answers.indexOf(answer);
if(answer.id) {
  axios.delete('/api/answer/'+answer.id)
    .then(() => {

this.options[idOption].tasks[idTask].answers.splice(this.options[idOption].tasks[idTask].answers.indexOf(an
swer), 1);
    })
  } else {

this.options[idOption].tasks[idTask].answers.splice(this.options[idOption].tasks[idTask].answers.indexOf(an
swer), 1);
  }
},
addTrueAnswer(option, task, answer) {
  var idOption = this.options.indexOf(option);
  var idTask = this.options[idOption].tasks.indexOf(task);
  var idAnswer = this.options[idOption].tasks[idTask].answers.indexOf(answer);
  if(answer.type == "inputText") {
    this.options[idOption].tasks[idTask].answers[idAnswer].variant_answer.push({
      value: "",
    })
  }
  if(answer.type == "inputSelect" || answer.type == "inputTrueElement") {
    this.options[idOption].tasks[idTask].answers[idAnswer].variant_answer.push({
      value: "",
      isTrue: false,
    })
  }
  if(answer.type == "inputSequence") {
    this.options[idOption].tasks[idTask].answers[idAnswer].variant_answer.push({
      description: "",
      value: null,
    })
  }
},
saveTrueAnswer(option, task, answer) {
  var idOption = this.options.indexOf(option);
  var idTask = this.options[idOption].tasks.indexOf(task);
  var idAnswer = this.options[idOption].tasks[idTask].answers.indexOf(answer);
  if(answer.id) {
    axios.post('/api/update-answer/'+answer.id,
this.options[idOption].tasks[idTask].answers[idAnswer])
    .then((response) => {
      this.getData();
      this.snackbar.status = true;
      this.snackbar.text = "Збережено";
    })
  }
}

```

```

        .catch() => {
            this.snackbar.status = true;
            this.snackbar.text = "Помилка";
        })
    } else {
        axios.post('/api/post-answer/'+task.id, this.options[idOption].tasks[idTask].answers[idAnswer])
        .then() => {
            this.getData();
            this.snackbar.status = true;
            this.snackbar.text = "Додано";
        })
        .catch() => {
            this.snackbar.status = true;
            this.snackbar.text = "Помилка";
        })
    }
},
createArray(option, task, answer) {
    var idOption = this.options.indexOf(option);
    var idTask = this.options[idOption].tasks.indexOf(task);
    var idAnswer = this.options[idOption].tasks[idTask].answers.indexOf(answer);
    var array = [];
    for (var i = 0; i < this.options[idOption].tasks[idTask].answers[idAnswer].rows; i++) {
        array[i] = [];
        for (var j = 0; j < this.options[idOption].tasks[idTask].answers[idAnswer].cols; j++) {
            array[i][j] = {value: 0};
        }
    }
    this.options[idOption].tasks[idTask].answers[idAnswer].variant_answer = array;
},
closeDialog() {
    this.dialogAddOption = false
}
},
updated() {
    this.preloader = false;
}
}
</script>

<style lang="css">
    .box {
        border: 1px solid silver;
        box-sizing: border-box;
    }
</style>

```

```

TaskController.php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Options;
use App\Models\Tasks;
use App\Models\Answers;
use App\Models\Results;

class TasksController extends Controller
{
    // options
    function getOptions() {
        $data = Options::with("tasks")->get();
        foreach($data as $key => $value) {
            foreach($value->tasks as $k => $v) {
                $v->answers = Answers::where("task_id", $v->id)->get();
            }
        }
        return response()->json($data);
    }
    function getOptionId($id) {
        $data = Options::with("tasks")->find($id);
        foreach($data->tasks as $k => $v) {
            $v->answers = Answers::where("task_id", $v->id)->get();
        }
        return response()->json($data);
    }
    function postOption(Request $request) {
        $model = new Options();
        $response = $model->create($request->all());
        return response()->json($response);
    }
    function delOption($id) {
        Options::find($id)->delete();
    }

    // tasks
    function postTask(Request $request, $option_id) {
        $model = new Tasks();
        $data = $request->all();
        $data["option_id"] = $option_id;
        $response = $model->create($data);
        return response()->json($response);
    }
}

```

```

function updateTask(Request $request, $task_id) {
    $model = Tasks::find($task_id);
    $data = $request->all();
    $response = $model->update($data);
    return response()->json($response);
}
function delTask($task_id) {
    Tasks::find($task_id)->delete();
}

// answers
function postAnswer(Request $request, $task_id) {
    $model = new Answers();
    $data = $request->all();
    $data["task_id"] = $task_id;
    $response = $model->create($data);
    return response()->json($response);
}
function updateAnswer(Request $request, $answer_id) {
    $model = Answers::find($answer_id);
    $data = $request->all();
    $response = $model->update($data);
    return response()->json($response);
}
function delAnswer($answer_id) {
    Answers::find($answer_id)->delete();
}

//results
function postResult(Request $request, $user_id) {
    $model = new Results();
    $data = $request->all();
    $data["user_id"] = $user_id;
    date_default_timezone_set('Europe/Volgograd');
    $data["created_at"] = date('d-m-Y H:i:s');
    $response = $model->create($data);
    return response()->json($response);
}
function getResult($user_id) {
    $data = Results::where("user_id", $user_id)->orderBy('created_at', 'desc')->get();
    return response()->json($data);
}
}

```