

**Полтавський університет економіки і торгівлі**  
Навчально-науковий інститут денної освіти  
Форма навчання денна  
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту  
Завідувач кафедри

\_\_\_\_\_ Олена ОЛЬХОВСЬКА  
(підпис)

«\_\_\_» \_\_\_\_\_ 2026 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
на тему  
**«РОЗРОБКА НАВЧАЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТРЕНІНГУ  
“КОЛЕДЖ: СТАРТ УСПІХУ”»**

зі спеціальності 122 «Комп'ютерні науки»  
освітня програма «Комп'ютерні науки»  
ступеня бакалавра

**Виконавець роботи** Циганков Дмитро Ігорович

\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 2026 р.  
(підпис)

**Науковий керівник** Чілікіна Тетяна Володимирівна

\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 2026 р.  
(підпис)

**Рецензент**

\_\_\_\_\_

# ЗАТВЕРДЖУЮ

Завідувач кафедри \_\_\_\_\_ Олена ОЛЬХОВСЬКА

«\_\_\_\_\_» \_\_\_\_\_ 202\_\_р.

## ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему «Розробка навчального програмного забезпечення тренінгу “Коледж - старт успіху”»

зі спеціальності 122 Комп’ютерні науки»

освітня програма «Комп’ютерні науки»

ступеня бакалавр

Прізвище, ім’я, по батькові Циганков Дмитро Ігорович

Затверджена наказом ректора № \_\_\_\_\_-Н від «\_\_\_\_\_» \_\_\_\_\_ 202\_\_р.

Термін подання студентом роботи «\_\_\_\_\_» \_\_\_\_\_ 202\_\_р.

Вихідні дані до кваліфікаційної роботи: публікації з теми, веб-сайти сайтів кафедр.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовна постановка задачі

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Аналіз аналогічних систем навчальних тренінгів

2.2 Переваги та недоліки існуючих рішень

2.3 Огляд веб-технологій HTML, CSS, JavaScript

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Проектування архітектури програмного забезпечення

3.2 Обґрунтування вибору технологій

3.3 Розробка алгоритмів роботи системи

3.4 Покроковий алгоритм роботи програмного забезпечення

3.5 UML-моделювання системи

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1 Реалізація інтерфейсу користувача

4.2 Реалізація функцій системи

4.3 Розробка системи тестування

4.3.1 Структура тренінгу

4.3.2 Опис навчальних модулів

#### 4.4 Реалізація збереження даних LocalStorage

### РОЗДІЛ 5. ТЕСТУВАННЯ ТА ОЦІНКА СИСТЕМИ

#### 5.1 Перевірка працездатності системи

#### 5.2 Аналіз результатів тестування

#### 5.3 Оцінка ефективності програмного продукту

### ВИСНОВКИ

### РЕКОМЕНДАЦІЇ

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

#### ДОДАТОК А. UML-діаграми системи

#### ДОДАТОК Б. Скріншоти програмного забезпечення

#### ДОДАТОК В. Фрагменти програмного коду

#### ДОДАТОК Г. Тестові матеріали

### Консультанти розділів кваліфікаційної роботи

Розділ	ПІБ, посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Постанова задачі	Чілікіна Т.В.		
Інформаційний огляд	Чілікіна Т.В.		
Теоретична частина	Чілікіна Т.В.		
Практична частина	Чілікіна Т.В.		

### Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику		
3. Постановка задачі		
4. Інформаційний огляд джерел бібліотек та інтернету		
5. Теоретична частина		
6. Практична частина		
7. Закінчення оформлення		
8. Доповідь студента на кафедрі		

9. Доопрацювання (за необхідністю), рецензування		
---	--	--

Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 202\_\_ р.

Здобувач вищої освіти Циганков Дмитро Ігорович

Науковий керівник Чілікіна Т.В.

**Результати захисту кваліфікаційної роботи**

Кваліфікаційна робота оцінена на \_\_\_\_\_  
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № \_\_\_\_ від « \_\_\_\_ » \_\_\_\_\_ 202\_\_ р.

Секретар ЕК \_\_\_\_\_  
(підпис) (ініціал та прізвище)

**Затверджую**

Зав. кафедрою \_\_\_\_\_

к.ф.-м.н. Олена Ольховська

« \_\_\_\_\_ » \_\_\_\_\_ 202\_ р.

**Погоджено**

Науковий керівник \_\_\_\_\_

к.ф.-м.н. Чілікіна Т.В.

« \_\_\_\_\_ » \_\_\_\_\_ 202\_ р.

## **План**

**ВСТУП**

**РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ**

1.1 Змістовна постановка задачі

**РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД**

2.1 Аналіз аналогічних систем навчальних тренінгів

2.2 Переваги та недоліки існуючих рішень

2.3 Огляд веб-технологій HTML, CSS, JavaScript

**РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА**

3.1 Проектування архітектури програмного забезпечення

3.2 Обґрунтування вибору технологій

3.3 Розробка алгоритмів роботи системи

3.4 Покроковий алгоритм роботи програмного забезпечення

3.5 UML-моделювання системи

**РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА**

4.1 Реалізація інтерфейсу користувача

4.2 Реалізація функцій системи

4.3 Розробка системи тестування

4.3.1 Структура тренінгу

4.3.2 Опис навчальних модулів

4.4 Реалізація збереження даних LocalStorage

**РОЗДІЛ 5. ТЕСТУВАННЯ ТА ОЦІНКА СИСТЕМИ**

5.1 Перевірка працездатності системи

5.2 Аналіз результатів тестування

5.3 Оцінка ефективності програмного продукту

**ВИСНОВКИ**

**РЕКОМЕНДАЦІЇ**

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

ДОДАТОК А. UML-діаграми системи

ДОДАТОК Б. Скріншоти програмного забезпечення

ДОДАТОК В. Фрагменти програмного коду

ДОДАТОК Г. Тестові матеріали

## РЕФЕРАТ

НАВЧАЛЬНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ТРЕНІНГ,  
ВЕБДОДАТОК, HTML, CSS, JAVASCRIPT, LOCALSTORAGE,  
ТЕСТУВАННЯ, ОСВІТНІ ТЕХНОЛОГІЇ, ІНФОРМАЦІЙНА СИСТЕМА.

Пояснювальна записка присвячена розробці навчального програмного забезпечення тренінгу «Коледж: Старт успіху». У роботі розглянуто питання проектування, розробки та тестування веборієнтованої інформаційної системи, призначеної для адаптації студентів до навчання в закладі вищої освіти. Розроблений програмний продукт забезпечує можливість реєстрації користувачів, проходження навчальних модулів, виконання тестових завдань, перегляду результатів навчання та взаємодії з викладачем.

Метою кваліфікаційної роботи є створення навчального програмного забезпечення, яке сприятиме адаптації студентів до освітнього процесу, підвищенню рівня обізнаності щодо організації навчання та формуванню навичок самостійного опрацювання матеріалу.

Об'єктом дослідження є процес використання інформаційних технологій у навчальному процесі закладів освіти. Предметом дослідження є методи та засоби розробки навчальних програмних систем на основі вебтехнологій.

У процесі розробки використано HTML, CSS, JavaScript та LocalStorage. Інтерфейс користувача реалізовано засобами сучасної веброзробки, що забезпечує простоту використання, адаптивність та інтерактивність системи.

Практичним результатом роботи є програмний продукт, який може використовуватися в освітньому процесі для ознайомлення студентів з особливостями навчання у коледжі та контролю рівня засвоєння навчального матеріалу.

## ЗМІСТ

<b>РЕФЕРАТ</b> .....	6
<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ</b> .....	9
<b>ВСТУП</b> .....	10
<b>РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ</b> .....	12
1.1 Змістовна постановка задачі .....	12
Висновки до розділу 1 .....	13
<b>РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД</b> .....	14
2.1 Аналіз аналогічних систем навчальних тренінгів .....	15
2.2 Переваги та недоліки існуючих рішень .....	16
2.3 Огляд веб-технологій HTML, CSS, JavaScript .....	16
Висновки до розділу 2 .....	17
<b>РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА</b> .....	17
3.1 Проектування архітектури програмного забезпечення.....	18
3.2 Обґрунтування вибору технологій .....	18
3.3 Розробка алгоритмів роботи системи .....	19
3.4 Покроковий алгоритм роботи програмного забезпечення .....	20
3.5 UML-моделювання системи.....	21
Висновки до розділу 3 .....	23
<b>РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА</b> .....	22
4.1 Реалізація інтерфейсу користувача .....	24
4.2 Реалізація функцій системи .....	26
4.3 Розробка системи тестування .....	26
4.3.1 Структура тренінгу .....	28
4.3.2 Опис навчальних модулів.....	29
4.4 Реалізація збереження даних LocalStorage .....	29
Висновки до розділу 4 .....	30
<b>РОЗДІЛ 5. ТЕСТУВАННЯ ТА ОЦІНКА СИСТЕМИ</b> .....	28
5.1 Перевірка працездатності системи .....	31
5.2 Аналіз результатів тестування .....	31
5.3 Оцінка ефективності програмного продукту .....	32
Висновки до розділу 5 .....	32
<b>ВИСНОВКИ</b> .....	34
<b>РЕКОМЕНДАЦІЇ</b> .....	36
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	37
<b>ДОДАТКИ</b> .....	39
<b>ДОДАТОК А. UML-ДІАГРАМИ СИСТЕМИ</b> .....	39

<b>ДОДАТОК Б. СКРІНШОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....</b>	<b>42</b>
<b>ДОДАТОК В. ФРАГМЕНТИ ПРОГРАМНОГО КОДУ .....</b>	<b>45</b>
Лістинг В.1 - Файл index.html .....	45
Лістинг В.2 - Файл index.js .....	66
<b>ДОДАТОК Г. ТЕСТОВІ МАТЕРІАЛИ.....</b>	<b>82</b>
<b>ЗАГАЛЬНІ ВИСНОВКИ.....</b>	<b>83</b>
<b>РЕКОМЕНДАЦІЇ ЩОДО ВПРОВАДЖЕННЯ ТА ПОДАЛЬШОГО РОЗВИТКУ СИСТЕМИ.....</b>	<b>88</b>

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ**

HTML - HyperText Markup Language.

CSS - Cascading Style Sheets.

JS - JavaScript.

UML - Unified Modeling Language.

UI - User Interface.

UX - User Experience.

DOM - Document Object Model.

API - Application Programming Interface.

JSON - JavaScript Object Notation.

IT - Information Technology.

ПЗ - програмне забезпечення.

ІС - інформаційна система.

## ВСТУП

У сучасних умовах цифровізації суспільства інформаційні технології стали невід'ємною складовою освітнього процесу. Заклади освіти активно впроваджують електронні освітні ресурси, системи дистанційного навчання, інтерактивні тренінги та програмні засоби контролю знань. Використання таких технологій дозволяє зробити процес навчання більш доступним, гнучким та ефективним.

Одним із важливих етапів навчання є адаптація студентів до нового освітнього середовища. Особливо актуальним це питання є для студентів першого курсу, які переходять від шкільної системи навчання до системи фахової передвищої або вищої освіти. На початковому етапі навчання студенти стикаються з великою кількістю нової інформації, новими правилами організації освітнього процесу, вимогами академічної доброчесності та особливостями взаємодії з викладачами.

Ефективним інструментом вирішення зазначеної проблеми є використання спеціалізованих навчальних тренінгів. Сучасні програмні тренінги дозволяють поєднати теоретичний матеріал, практичні завдання та систему контролю знань в єдиному інформаційному середовищі. Завдяки цьому студенти мають можливість самостійно опановувати навчальний матеріал, проходити тестування та отримувати миттєвий зворотний зв'язок щодо результатів навчання.

Актуальність теми кваліфікаційної роботи обумовлена необхідністю створення сучасних програмних засобів підтримки навчального процесу, які забезпечують ефективне засвоєння інформації, підвищують мотивацію студентів та сприяють їхній успішній адаптації до навчання.

Метою кваліфікаційної роботи є розробка навчального програмного забезпечення тренінгу «Коледж: Старт успіху», яке забезпечує ознайомлення студентів з особливостями освітнього процесу та контроль рівня засвоєння навчального матеріалу.

Для досягнення поставленої мети необхідно вирішити такі завдання: проаналізувати існуючі навчальні програмні системи та тренінгові платформи; дослідити сучасні вебтехнології для розробки навчального програмного забезпечення; спроектувати архітектуру програмної системи; розробити алгоритми роботи програмного забезпечення; реалізувати інтерфейс користувача та основні функціональні модулі системи; реалізувати систему тестування та збереження результатів; провести тестування розробленого програмного продукту та оцінити його ефективність.

Об'єктом дослідження є процес використання навчальних інформаційних систем у закладах освіти. Предметом дослідження є методи та технології розробки веборієнтованого навчального програмного забезпечення.

Практичне значення роботи полягає у створенні програмного продукту, який може використовуватися в навчальному процесі для адаптації студентів першого курсу та підвищення ефективності освітньої діяльності.

Об'єктом дослідження є процес адаптації студентів до освітнього процесу із використанням інформаційних технологій.

Предметом дослідження є методи, засоби та технології розробки навчального програмного забезпечення тренінгу «Коледж: Старт успіху».

## **РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ**

### **1.1 Змістовна постановка задачі**

Предметною областю даного дослідження є використання інформаційних технологій для підтримки освітнього процесу та адаптації студентів до навчання в закладах вищої освіти. У сучасних умовах розвитку цифрового суспільства значна частина навчальної діяльності здійснюється із використанням електронних ресурсів, дистанційних платформ та програмних засобів контролю знань.

Проблема адаптації студентів першого курсу є однією з найбільш актуальних у системі освіти. Перехід від шкільного навчання до навчання у закладі вищої освіти супроводжується зміною організації освітнього процесу, збільшенням обсягу самостійної роботи, необхідністю дотримання принципів академічної доброчесності та використанням нових інформаційних технологій.

На початковому етапі навчання студенти часто стикаються з труднощами, пов'язаними з недостатнім рівнем поінформованості про структуру освітнього процесу, порядок оцінювання результатів навчання, використання електронних освітніх ресурсів та організацію власної навчальної діяльності.

Одним із ефективних способів вирішення зазначеної проблеми є використання спеціалізованих навчальних тренінгів. Такі тренінги дозволяють поєднати теоретичну інформацію, практичні вправи та систему контролю знань у межах єдиного програмного середовища.

Сучасні програмні тренінги забезпечують доступність навчальних матеріалів, інтерактивну взаємодію користувача із системою, автоматизовану перевірку знань, накопичення результатів навчання та підвищення мотивації студентів до навчання.

На основі проведеного аналізу предметної області було встановлено необхідність створення навчального програмного забезпечення тренінгу «Коледж: Старт успіху». Основною метою розробки є створення інформаційної системи, яка забезпечить ознайомлення студентів з особливостями навчання у закладі освіти та надасть можливість перевірки рівня засвоєння навчального матеріалу.

- реєстрацію користувачів
- перегляд навчальних матеріалів
- навігацію між навчальними модулями
- проходження тестування
- автоматичну перевірку відповідей
- формування результатів навчання
- збереження результатів користувачів
- перегляд статистики

Таблиця 1.1 - Основні вимоги до системи

№	Вимога	Характеристика
1	Зручність використання	Простий інтерфейс користувача
2	Надійність	Стабільна робота системи
3	Швидкодія	Швидке завантаження сторінок
4	Адаптивність	Підтримка різних пристроїв
5	Функціональність	Реалізація навчання та тестування
6	Масштабованість	Можливість розширення системи

## Висновки до розділу 1

У першому розділі було проведено аналіз предметної області, визначено особливості використання інформаційних технологій в освітньому процесі та обґрунтовано необхідність створення навчального програмного забезпечення тренінгу «Коледж: Старт успіху». Було сформульовано мету та завдання роботи,

визначено функціональні та нефункціональні вимоги до системи, а також встановлено перелік вхідних та вихідних даних.

## **РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД**

### **2.1 Аналіз аналогічних систем навчальних тренінгів**

Сучасний розвиток інформаційних технологій сприяв активному впровадженню електронних освітніх платформ у навчальний процес. Навчальні тренінги, системи дистанційного навчання та інтерактивні освітні середовища стали невід'ємною складовою сучасної освіти.

Перед розробкою власного програмного забезпечення доцільно провести аналіз існуючих рішень, визначити їхні переваги та недоліки, а також виявити функціональні можливості, які можуть бути використані під час створення тренінгу «Коледж: Старт успіху».

Однією з найпоширеніших платформ дистанційного навчання є Moodle. Moodle являє собою безкоштовну систему управління навчанням, яка використовується в багатьох закладах освіти світу. Вона дозволяє розміщувати навчальні матеріали, проводити тестування, організовувати дистанційне навчання та контролювати успішність студентів.

Google Classroom є хмарною платформою для організації навчального процесу. Вона забезпечує створення навчальних курсів, розміщення матеріалів, перевірку завдань і взаємодію між викладачами та студентами. Головною перевагою є простота використання та інтеграція з сервісами Google.

Coursera та Prometheus є платформами масових відкритих онлайн-курсів. Вони забезпечують відеолекції, тестування, сертифікацію результатів навчання та доступ до якісного освітнього контенту. LearningApps використовується для створення інтерактивних навчальних вправ і вікторин.

## 2.2 Переваги та недоліки існуючих рішень

Для вибору оптимальної архітектури майбутнього програмного забезпечення було проведено порівняльний аналіз розглянутих систем.

Таблиця 2.1 - Порівняння аналогічних систем

Система	Тестування	Навчальні модулі	Реєстрація	Простота використання
Moodle	Так	Так	Так	Середня
Google Classroom	Так	Так	Так	Висока
Coursera	Так	Так	Так	Висока
Prometheus	Так	Так	Так	Висока
LearningApps	Так	Частково	Ні	Висока
Розроблювана система	Так	Так	Так	Висока

Проведений аналіз показав, що більшість сучасних систем мають широкий функціонал, проте не орієнтовані безпосередньо на адаптацію студентів до освітнього процесу конкретного закладу освіти. Саме тому було прийнято рішення про створення власного навчального тренінгу.

## 2.3 Огляд веб-технологій HTML, CSS, JavaScript

Для реалізації програмного забезпечення було обрано сучасні вебтехнології HTML5, CSS3 та JavaScript. HTML використовується для створення структури вебсторінок, CSS - для оформлення вебінтерфейсу, а JavaScript - для реалізації логіки роботи клієнтської частини вебдодатку.

У проєкті JavaScript використовується для обробки подій користувача, переходів між модулями, реалізації тестування, роботи з LocalStorage та формування результатів. Для збереження інформації використовується LocalStorage, який забезпечує локальне зберігання даних без використання серверної бази даних.

Таблиця 2.2 - Обґрунтування вибору технологій

Технологія	Призначення
HTML5	Структура сторінок

CSS3	Дизайн інтерфейсу
JavaScript	Логіка роботи
LocalStorage	Збереження даних

## **Висновки до розділу 2**

У другому розділі було проведено огляд сучасних навчальних платформ та систем дистанційного навчання. Виконано порівняльний аналіз існуючих програмних рішень і визначено їхні переваги та недоліки. На основі проведеного дослідження обґрунтовано доцільність розробки власного навчального програмного забезпечення тренінгу «Коледж: Старт успіху».

## РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

### 3.1 Проєктування архітектури програмного забезпечення

Одним із найважливіших етапів створення будь-якого програмного продукту є проєктування його архітектури. Правильно спроектована архітектура забезпечує ефективну взаємодію між компонентами системи, спрощує супровід програмного забезпечення та створює можливості для подальшого розширення функціоналу.

Розроблене програмне забезпечення являє собою односторінковий вебдодаток, що працює без використання серверної бази даних. Основна логіка реалізована засобами JavaScript, а збереження інформації здійснюється за допомогою технології LocalStorage.

Архітектура системи включає три основні рівні: рівень представлення даних, рівень бізнес-логіки та рівень збереження даних. Така структура дозволяє розділити функціональність програмного забезпечення на окремі незалежні компоненти.

Таблиця 3.1 - Основні модулі системи

№	Назва модуля	Призначення
1	Модуль реєстрації	Реєстрація користувачів
2	Модуль тренінгу	Відображення навчального матеріалу
3	Модуль тестування	Перевірка знань студентів
4	Модуль результатів	Аналіз та збереження результатів
5	Модуль викладача	Перегляд статистики та результатів

### 3.2 Обґрунтування вибору технологій

Для створення навчального програмного забезпечення тренінгу «Коледж: Старт успіху» було обрано комплекс сучасних вебтехнологій, до складу якого

входять HTML5, CSS3, JavaScript та LocalStorage. Даний набір інструментів дозволяє реалізувати всі необхідні функції системи без використання складних серверних технологій.

Основною вимогою до програмного продукту була можливість його роботи у звичайному веббраузері без необхідності встановлення спеціалізованих програм або налаштування серверного середовища. Саме тому було прийнято рішення використовувати клієнтські вебтехнології.

Використання HTML дозволяє створити структуру сторінок, CSS забезпечує сучасний адаптивний інтерфейс, JavaScript реалізує логіку роботи системи, а LocalStorage використовується для збереження даних користувачів і результатів тестування.

Таблиця 3.2 - Порівняння технологій інтерфейсу

Технологія	Переваги	Недоліки
HTML/CSS/JavaScript	Простота, швидкість розробки, підтримка браузерів	Обмеження при створенні великих систем
React	Компонентний підхід, масштабованість	Складніше освоєння
Angular	Велика функціональність	Висока складність
Vue.js	Простота та продуктивність	Менша популярність

### 3.3 Розробка алгоритмів роботи системи

У навчальному програмному забезпеченні алгоритми забезпечують виконання всіх основних функцій системи: реєстрацію користувачів, проходження навчальних модулів, тестування, перевірку результатів та збереження інформації.

Під час проектування програмного забезпечення були розроблені такі основні алгоритми: алгоритм реєстрації користувача, алгоритм проходження навчального тренінгу, алгоритм тестування знань, алгоритм перевірки відповідей, алгоритм збереження результатів та алгоритм роботи панелі викладача.

Узагальнений алгоритм роботи навчального програмного забезпечення має такий вигляд: запуск системи, реєстрація студента, ознайомлення з навчальними модулями, проходження тестування, автоматична перевірка відповідей, формування результату, збереження результату, перегляд статистики та завершення роботи.

### **3.4 Покроковий алгоритм роботи програмного забезпечення**

Алгоритм роботи навчального програмного забезпечення тренінгу «Коледж: Старт успіху» складається з трьох основних етапів, які забезпечують повний цикл взаємодії користувача із системою.

Перший етап – реєстрація та авторизація користувача. На цьому етапі студент вводить персональні дані для створення облікового запису або використовує раніше створений профіль для входу до системи. Після перевірки коректності введених даних система створює особистий кабінет користувача та надає доступ до навчальних матеріалів і функціональних можливостей програмного забезпечення. Усі необхідні відомості про користувача зберігаються для подальшої роботи із системою.

Другий етап – проходження навчального тренінгу. Після успішної авторизації студент отримує доступ до навчальних модулів, які містять інформацію щодо організації освітнього процесу, академічної доброчесності, використання електронних ресурсів та особливостей навчання в закладі освіти. Користувач послідовно опрацьовує навчальні матеріали, виконує практичні завдання та проходить тестування для перевірки рівня засвоєння отриманих знань.

Третій етап – аналіз та збереження результатів. Після завершення тестування система автоматично обробляє відповіді користувача, розраховує

кількість набраних балів, визначає рівень підготовки студента та формує підсумковий результат. Отримані дані відображаються в особистому кабінеті користувача та зберігаються за допомогою технології LocalStorage. Викладач має можливість переглядати результати студентів, аналізувати статистичні показники та оцінювати ефективність проходження тренінгу.

Таким чином, запропонований алгоритм забезпечує послідовне виконання всіх основних функцій навчального програмного забезпечення: від реєстрації користувача та проходження навчальних модулів до контролю знань, формування результатів і збереження інформації. Запропонована структура алгоритму є простою, зрозумілою та забезпечує ефективну організацію навчального процесу в межах програмного тренінгу «Коледж: Старт успіху».

### **3.5 UML-моделювання системи**

UML-моделювання було використано для опису навчального програмного забезпечення тренінгу «Коледж: Старт успіху». Для повного представлення системи розглянуто діаграму варіантів використання, діаграму активності, діаграму послідовності та діаграму компонентів. Відповідні діаграми наведено у додатку А.

### UML-діаграма варіантів використання

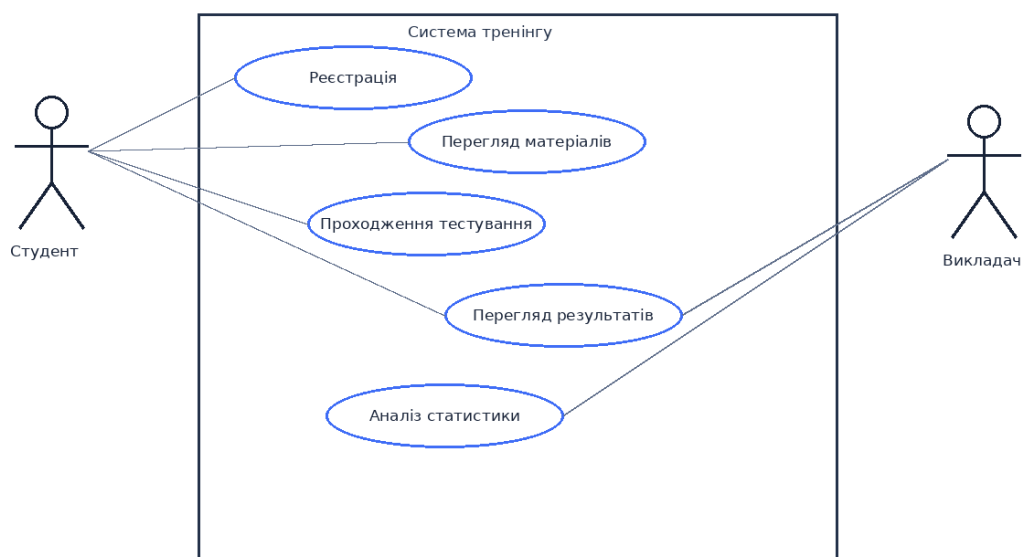


Рисунок 3.1 - UML-діаграма варіантів використання системи

**UML-діаграма активності**

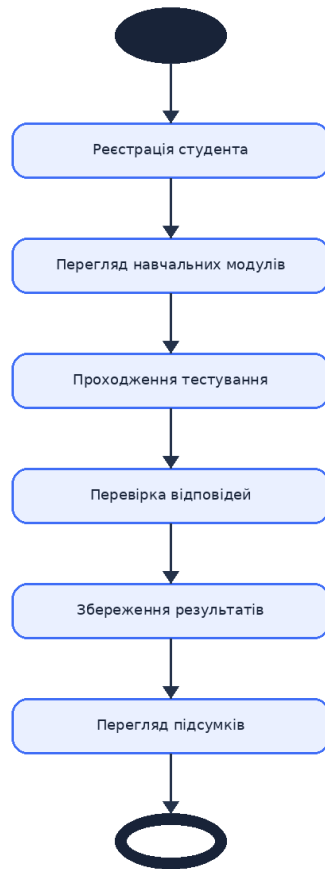


Рисунок 3.2 - UML-діаграма активності проходження тренінгу

### **Висновки до розділу 3**

У третьому розділі було сформовано архітектуру системи, обґрунтовано вибір технологій, розроблено алгоритми роботи програмного забезпечення та виконано UML-моделювання системи.

## РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

### 4.1 Реалізація інтерфейсу користувача

Одним із найважливіших етапів створення навчального програмного забезпечення є розробка інтерфейсу користувача. Саме інтерфейс забезпечує взаємодію між користувачем та програмною системою, визначає зручність використання програмного продукту та значною мірою впливає на ефективність навчального процесу.

Під час створення тренінгу «Коледж: Старт успіху» особлива увага приділялася простоті використання, зрозумілості навігації та сучасному дизайну вебдодатку. Інтерфейс було реалізовано із застосуванням HTML5, CSS3 та JavaScript.

Інтерфейс повинен бути інтуїтивно зрозумілим навіть для користувачів, які не мають спеціальної технічної підготовки.

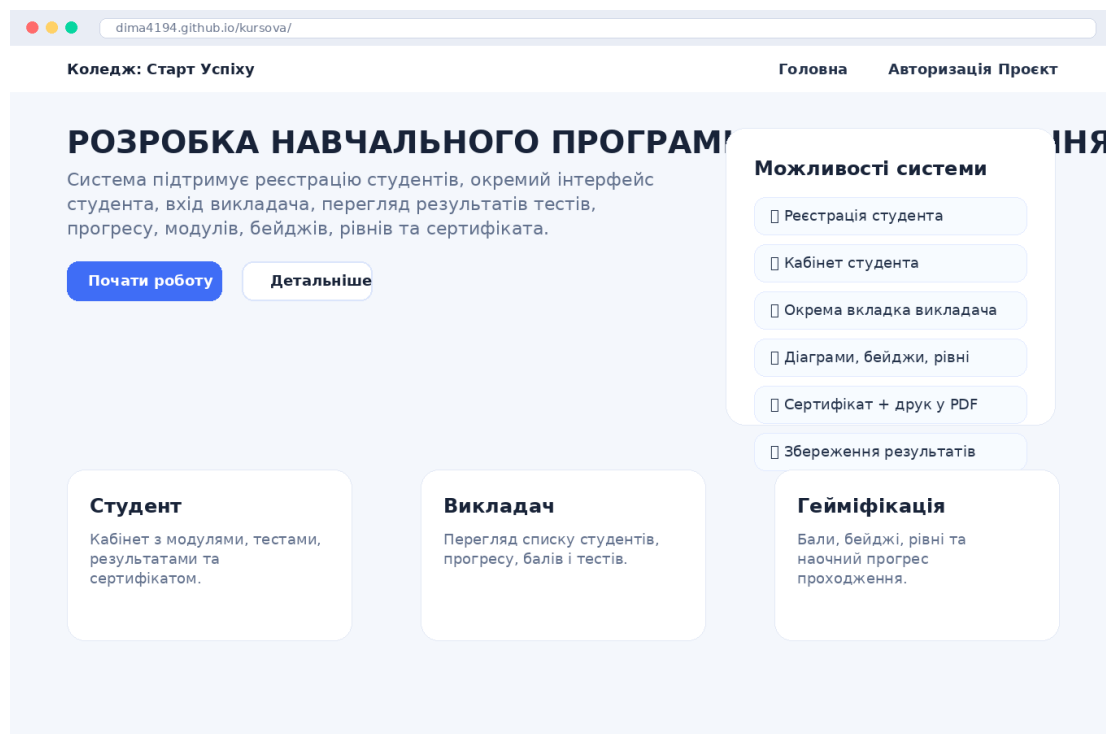


Рисунок 4.1 - Головна сторінка навчального тренінгу

dima4194.github.io/kursova/#auth

Коледж: Старт Успіху

## Авторизація в системі

### Реєстрація студента

Ім'я студента

Група

Email

Пароль

[Зареєструватися](#)

### Вхід студента

Email

Пароль

[Увійти як студент](#)

### Вхід викладача

Логін: teacher

Пароль: 12345

[Увійти як викладач](#)

Рисунок 4.2 - Форма реєстрації та авторизації користувача

dima4194.github.io/kursova/?view=student

## Меню студента

- Профіль
- Навчальні модулі**
- Тести
- Мої результати
- Сертифікат

Студент: Циганков Дмитро  
Група: КН-41

## Навчальні модулі

### Модуль 1 — Адаптація першокурсник

Ознайомлення з коледжем, розкладом, дедлайнами та правилами навчання.

[Пройдено](#)

### Модуль 2 — Тайм-менеджмент

Планування часу, постановка цілей та вправи на особисту продуктивність.

[Пройдено](#)

### Модуль 3 — Комунікація

Взаємодія в команді, спілкування з викладачами та вирішення типових кейсів.

[Не пройдено](#)

### Модуль 4 — Кар'єрний старт

Резюме, soft skills, вибір професійного напрямку та кар'єрне мислення.

[Не пройдено](#)

Рисунок 4.3 - Відображення навчальних модулів у кабінеті студента

## **4.2 Реалізація функцій системи**

У межах кваліфікаційної роботи було реалізовано комплекс функцій, необхідних для організації навчального процесу та контролю знань студентів. До основних функцій системи належать реєстрація користувачів, перегляд навчальних матеріалів, навігація між модулями, проходження тестування, перевірка відповідей, формування результатів, збереження інформації та перегляд статистики.

Першою функцією програмного забезпечення є реєстрація користувача. Метою реєстрації є ідентифікація студента, створення профілю користувача та підготовка до проходження тренінгу. Після натискання кнопки підтвердження система виконує перевірку введених даних і записує їх до LocalStorage.

Навчальні модулі є основною складовою навчального тренінгу. Кожен модуль містить інформацію щодо певної тематики навчального процесу. Після завершення одного модуля студент може перейти до наступного розділу тренінгу.

## **4.3 Розробка системи тестування**

Однією з найважливіших функціональних складових навчального програмного забезпечення є система тестування. Саме вона забезпечує контроль рівня засвоєння навчального матеріалу, дозволяє оцінити результати навчання студентів та отримати об'єктивні показники ефективності проходження тренінгу.

Система тестування складається з бази тестових запитань, модуля відображення питань, модуля збору відповідей, модуля перевірки результатів, модуля формування звіту та модуля збереження результатів.

Після запуску тестування система автоматично завантажує перше питання та відображає його користувачу. Інтерфейс тестування містить текст запитання, список варіантів відповіді, кнопку переходу до наступного питання та індикатор прогресу.

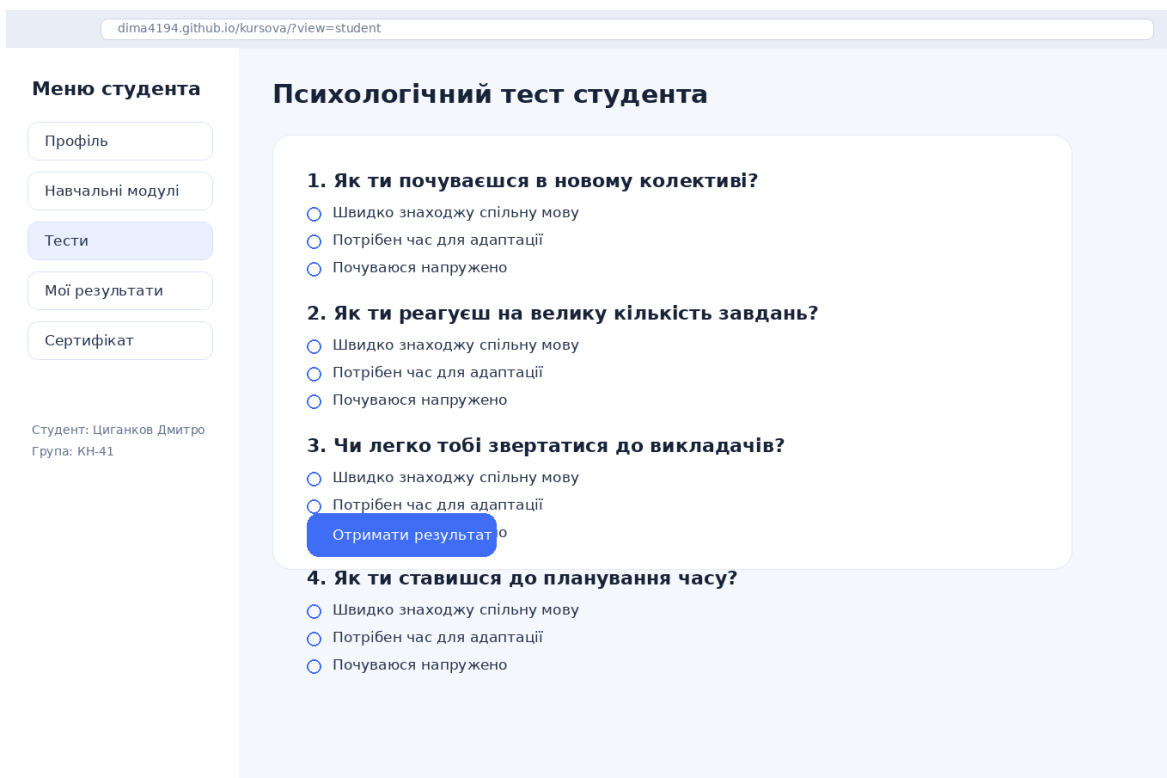


Рисунок 4.4 - Вікно проходження тестування студентом

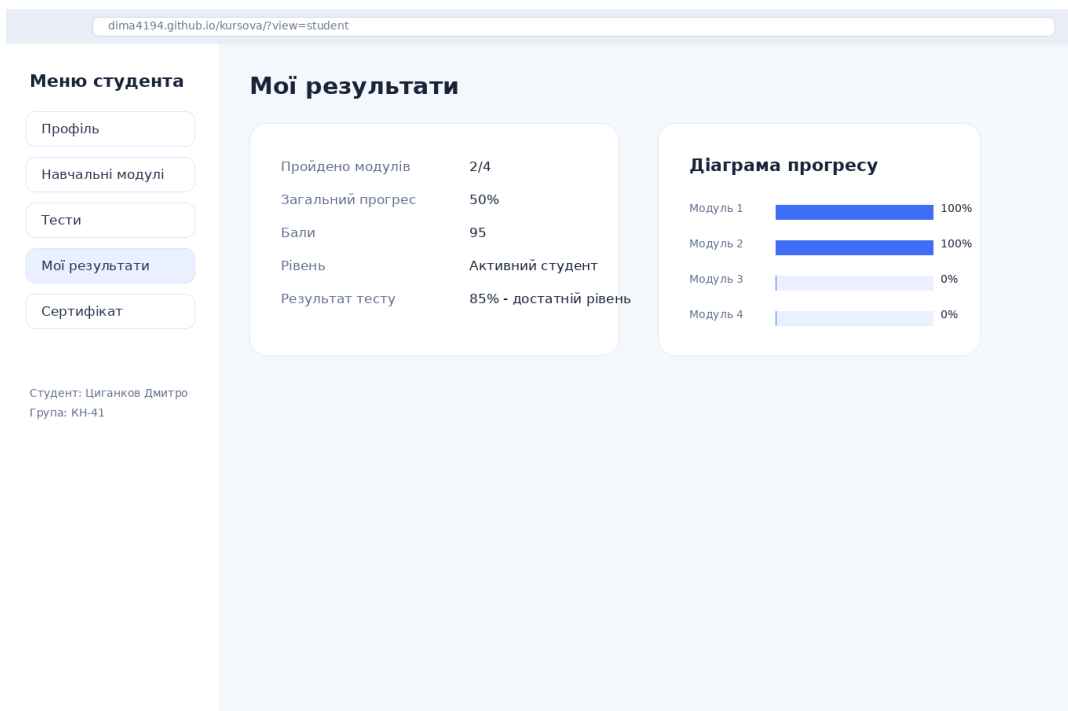


Рисунок 4.5 - Відображення результатів тестування

### 4.3.1 Структура тренінгу

Навчальний тренінг «Коледж: Старт успіху» призначений для адаптації студентів до освітнього процесу та ознайомлення їх з основними аспектами навчання в закладі освіти. Структура тренінгу побудована за модульним принципом, що забезпечує послідовне засвоєння навчального матеріалу.

Проходження тренінгу розпочинається з реєстрації та авторизації користувача. Після входу до системи студент отримує доступ до навчальних модулів, які містять інформацію про організацію навчального процесу, академічну доброчесність, використання електронних освітніх ресурсів та особливості самостійної роботи.

Після опрацювання навчальних матеріалів користувач проходить підсумкове тестування для перевірки рівня засвоєння знань. Система автоматично перевіряє відповіді, формує результат та зберігає його у

LocalStorage. Отримані результати можуть переглядатися студентом і викладачем для аналізу успішності проходження тренінгу.

Таким чином, структура тренінгу охоплює такі основні етапи: реєстрація користувача, вивчення навчальних модулів, проходження тестування, формування результатів та аналіз успішності.

#### **4.3.2 Опис навчальних модулів**

Навчальні модулі є основою змістовного наповнення тренінгу. Було створено тематичні модулі: знайомство з коледжем, організація навчального процесу, академічна доброчесність, електронні освітні ресурси, самостійна робота студента та підсумкове тестування.

#### **4.4 Реалізація збереження даних LocalStorage**

Під час розробки навчального програмного забезпечення одним із важливих завдань було забезпечення можливості збереження інформації про користувачів та результати їхнього навчання. Для реалізації цього функціоналу було використано технологію LocalStorage.

LocalStorage є механізмом локального збереження інформації у браузері. Основними характеристиками технології є зберігання даних у форматі «ключ - значення», незалежність від серверної частини, швидкий доступ до інформації, збереження даних після закриття браузера та підтримка всіма сучасними браузерами.

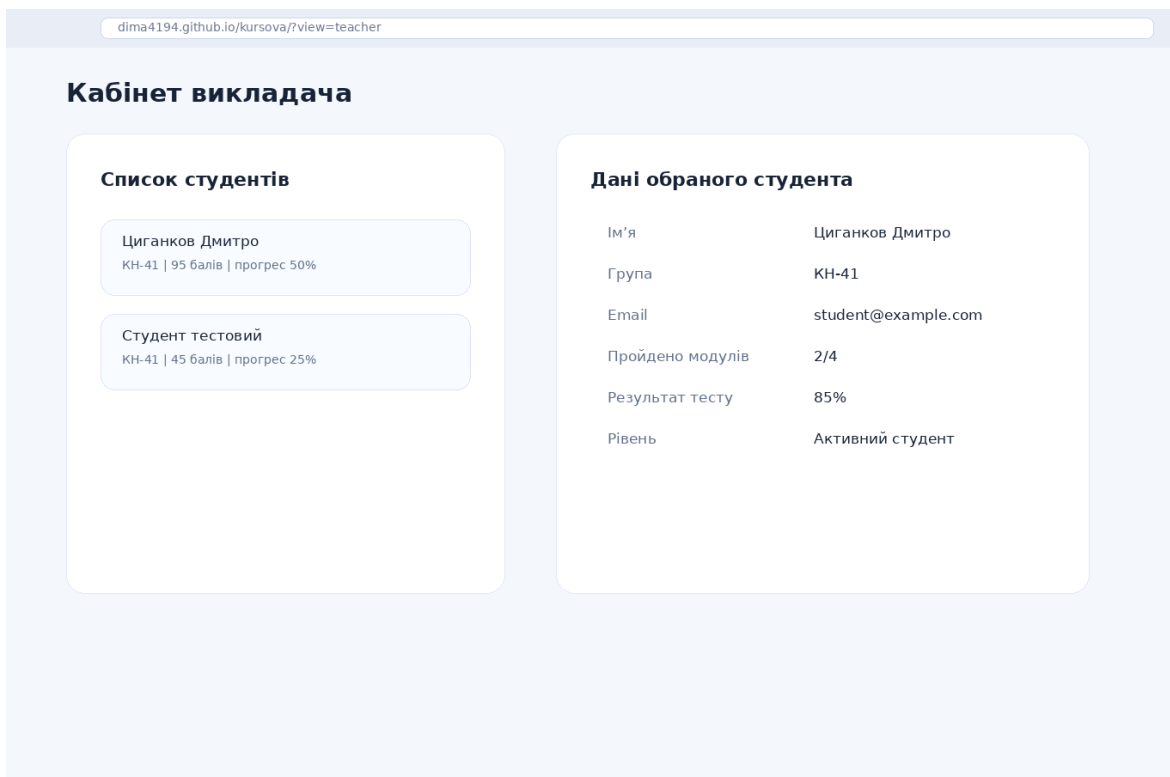


Рисунок 4.6 - Панель викладача з результатами студентів

#### Висновки до розділу 4

У четвертому розділі було виконано практичну реалізацію навчального програмного забезпечення тренінгу «Коледж: Старт успіху». Було створено інтерфейс користувача, реалізовано функціональні модулі системи, розроблено систему тестування та механізм збереження даних.

## РОЗДІЛ 5. ТЕСТУВАННЯ ТА ОЦІНКА СИСТЕМИ

### 5.1 Перевірка працездатності системи

Після завершення розробки навчального програмного забезпечення було проведено комплексне тестування системи з метою перевірки її працездатності, стабільності роботи та відповідності поставленим вимогам.

Основною метою тестування було підтвердження коректності роботи всіх функціональних модулів тренінгу «Коледж: Старт успіху». Під час перевірки було протестовано інтерфейс користувача, процес реєстрації студентів, функціонування навчальних модулів, систему тестування, обчислення результатів та роботу LocalStorage.

Таблиця 5.1 - Результати функціонального тестування

Тестовий сценарій	Очікуваний результат	Результат
Коректне введення даних	Реєстрація успішна	Виконано
Порожнє поле	Повідомлення про помилку	Виконано
Відображення питання	Коректне відображення	Виконано
Перевірка відповідей	Коректний результат	Виконано
Збереження даних	Запис у LocalStorage	Виконано

### 5.2 Аналіз результатів тестування

Після завершення функціонального тестування програмного забезпечення було проведено аналіз отриманих результатів з метою оцінювання якості реалізації системи, рівня її стабільності та відповідності поставленим вимогам.

У процесі дослідження було встановлено, що інтерфейс має зрозумілу структуру, усі елементи керування розташовані логічно, навігація між сторінками виконується швидко, а користувач не потребує додаткового навчання для роботи із системою.

Результати тестування підтвердили коректність реалізації тестової підсистеми, стабільну роботу LocalStorage та відсутність критичних помилок.

Таблиця 5.2 - Відповідність вимогам

Вимога	Ступінь виконання
--------	-------------------

Реєстрація користувачів	Виконано
Навчальні модулі	Виконано
Тестування	Виконано
Формування результатів	Виконано
Збереження даних	Виконано
Адаптивний інтерфейс	Виконано

### 5.3 Оцінка ефективності програмного продукту

Завершальним етапом дослідження є оцінювання ефективності розробленого програмного забезпечення. Проведення такого оцінювання дозволяє визначити практичну цінність створеної системи, її відповідність поставленим завданням та можливість використання в реальному освітньому процесі.

У межах кваліфікаційної роботи ефективність програмного продукту оцінювалася за такими критеріями: функціональна ефективність, технічна ефективність, освітня ефективність, економічна ефективність та перспективність розвитку.

Проведене оцінювання показало, що навчальне програмне забезпечення тренінгу «Коледж: Старт успіху» є функціонально завершеним, технічно надійним та ефективним засобом підтримки освітнього процесу.

Таблиця 5.3 - Загальна оцінка програмного продукту

Критерій	Рівень
Функціональна ефективність	Високий
Технічна ефективність	Високий
Освітня ефективність	Високий
Економічна ефективність	Високий
Перспективність розвитку	Високий

### Висновки до розділу 5

У п'ятому розділі було проведено комплексне тестування розробленого програмного забезпечення, виконано аналіз результатів його роботи та здійснено оцінювання ефективності системи. Результати тестування

підтвердили коректність функціонування всіх модулів програмного забезпечення.

## ВИСНОВКИ

У кваліфікаційній роботі було розроблено навчальне програмне забезпечення тренінгу «Коледж: Старт успіху», призначене для адаптації студентів до освітнього процесу та підвищення ефективності засвоєння інформації щодо організації навчання у закладі освіти.

У процесі виконання роботи було проведено аналіз предметної області та досліджено особливості використання інформаційних технологій в освітньому середовищі. Було встановлено, що проблема адаптації студентів першого курсу до навчання є актуальною для більшості закладів освіти та потребує застосування сучасних програмних засобів підтримки навчального процесу.

У результаті проведеного інформаційного огляду було проаналізовано сучасні освітні платформи та системи дистанційного навчання. Проведене дослідження дозволило визначити їхні переваги та недоліки, а також сформулювати основні вимоги до розроблюваного програмного забезпечення.

У роботі було обґрунтовано використання вебтехнологій HTML5, CSS3 та JavaScript як основних засобів реалізації програмного продукту. Для збереження інформації про користувачів та результати тестування було використано технологію LocalStorage, що дозволило відмовитися від використання серверної бази даних та спростити архітектуру системи.

Практична частина роботи включала реалізацію інтерфейсу користувача, функціональних модулів системи та механізмів тестування знань. У результаті було створено веборієнтований додаток, який забезпечує реєстрацію студентів, проходження навчальних модулів, виконання тестових завдань, автоматичну перевірку відповідей, формування результатів навчання, збереження інформації про користувачів та перегляд статистики успішності.

Поставлена мета кваліфікаційної роботи досягнута повністю, а всі завдання, визначені на початковому етапі дослідження, виконані у повному обсязі. Розроблене навчальне програмне забезпечення тренінгу «Коледж: Старт успіху» може бути рекомендоване до практичного використання в закладах освіти.

## **РЕКОМЕНДАЦІЇ**

- впровадження серверної бази даних для централізованого збереження інформації
- реалізація багатокористувацького режиму роботи
- створення особистого кабінету студента
- розширення функціоналу панелі викладача
- інтеграція із системами Moodle та Google Classroom
- використання мультимедійного контенту
- розробка мобільної версії додатка
- інтеграція хмарних сервісів
- розробка адаптивної системи тестування
- використання технологій штучного інтелекту для персоналізації навчання

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Брич Л.В. Компютерні технології в навчальному процесі. - К. : Новий світ-2000, 2025. - 175 с.
2. Морзе Н. В. Основи інформаційних технологій. - К. : Наукова думка, 2022. - 180 с.
3. Соммервіль І. Інженерія програмного забезпечення. - К. : Професіонал, 2022. - 415 с.
4. Pressman R. Software Engineering: A Practitioner's Approach. - 8th ed. - New York : McGraw-Hill, 2022. - 950 p.
5. Sommerville I. Software Engineering. - 10th ed. - Pearson Education, 2023. - 790 p.
6. W3Schools. HTML Tutorial. URL: <https://www.w3schools.com/html/>
7. W3Schools. CSS Tutorial. URL: <https://www.w3schools.com/css/>
8. W3Schools. JavaScript Tutorial. URL: <https://www.w3schools.com/js/>
9. Mozilla Developer Network. HTML Documentation. URL: <https://developer.mozilla.org/>
10. Mozilla Developer Network. CSS Documentation. URL: <https://developer.mozilla.org/>
11. Mozilla Developer Network. JavaScript Documentation. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/>
12. Duckett J. HTML and CSS: Design and Build Websites. - John Wiley & Sons, 2022. - 490 p.
13. Duckett J. JavaScript and JQuery: Interactive Front-End Web Development. - John Wiley & Sons, 2022. - 640 p.

14. Freeman E., Robson E. Head First HTML and CSS. - O'Reilly Media, 2020. - 768 p.
15. Flanagan D. JavaScript: The Definitive Guide. - 7th ed. - O'Reilly Media, 2020. - 704 p.
16. Goodman D. Dynamic HTML: The Definitive Reference. - O'Reilly Media, 2020. - 1350 p.
17. Sebasta R. Concepts of Programming Languages. - Pearson, 2020. - 816 p.
18. 1. Moodle Documentation. URL: <https://docs.moodle.org/>
19. Google Classroom Help Center. URL: <https://support.google.com/edu/classroom/>
20. Prometheus. Освітня платформа онлайн-курсів. URL: <https://prometheus.org.ua/>
21. Coursera. Online Learning Platform. URL: <https://www.coursera.org/>
22. LearningApps. Інтерактивні навчальні вправи. URL: <https://learningapps.org/>
23. ECMA International. ECMAScript Language Specification. URL: <https://www.ecma-international.org/>
25. WHATWG HTML Living Standard. URL: <https://html.spec.whatwg.org/>
26. W3C Cascading Style Sheets (CSS). URL: <https://www.w3.org/Style/CSS/>
27. HTML Web Storage Specification. URL: <https://www.w3.org/TR/webstorage/>
28. OpenJS Foundation. JavaScript Resources and Documentation. URL: <https://openjsf.org/>
29. Полтавський університет економіки і торгівлі. Офіційний сайт. URL: <https://puet.edu.ua/>

# ДОДАТКИ

## ДОДАТОК А. UML-ДІАГРАМИ СИСТЕМИ

### UML-діаграма варіантів використання

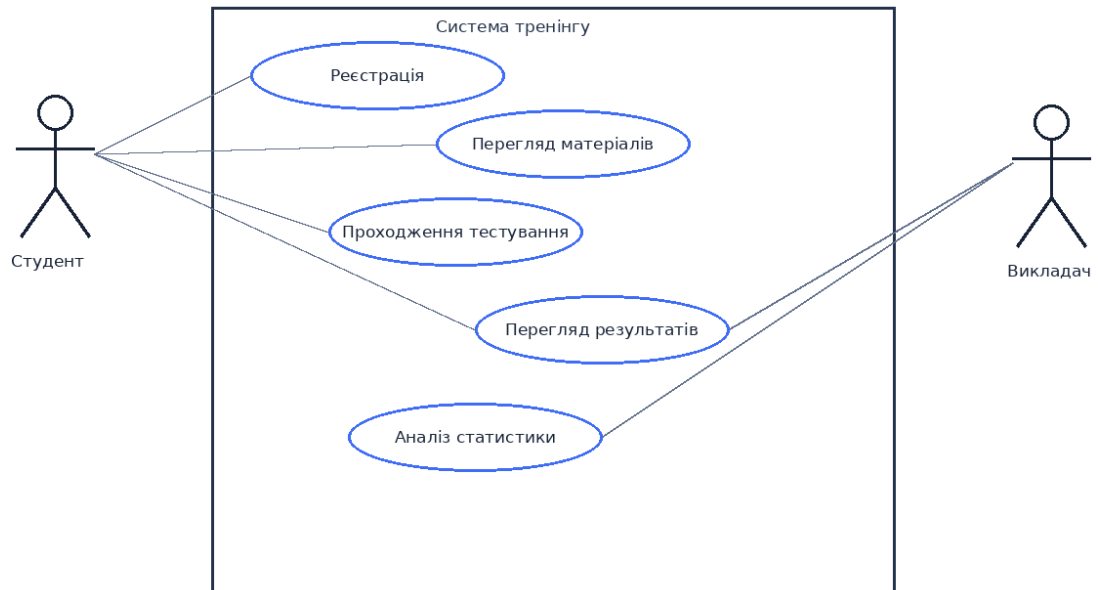


Рисунок А.1 - UML-діаграма варіантів використання системи

### UML-діаграма активності

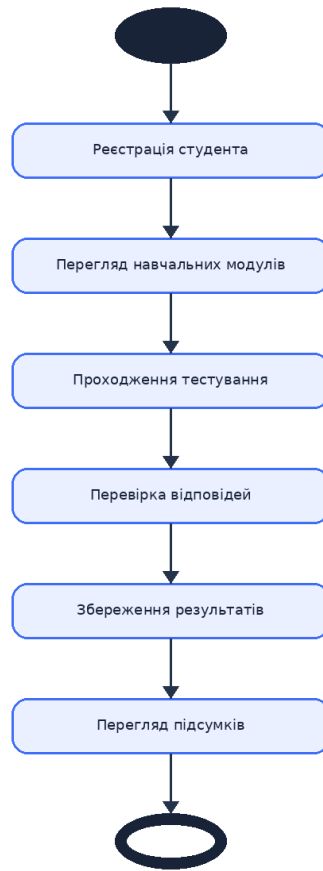


Рисунок А.2 - UML-діаграма активності проходження тренінгу

### UML-діаграма послідовності

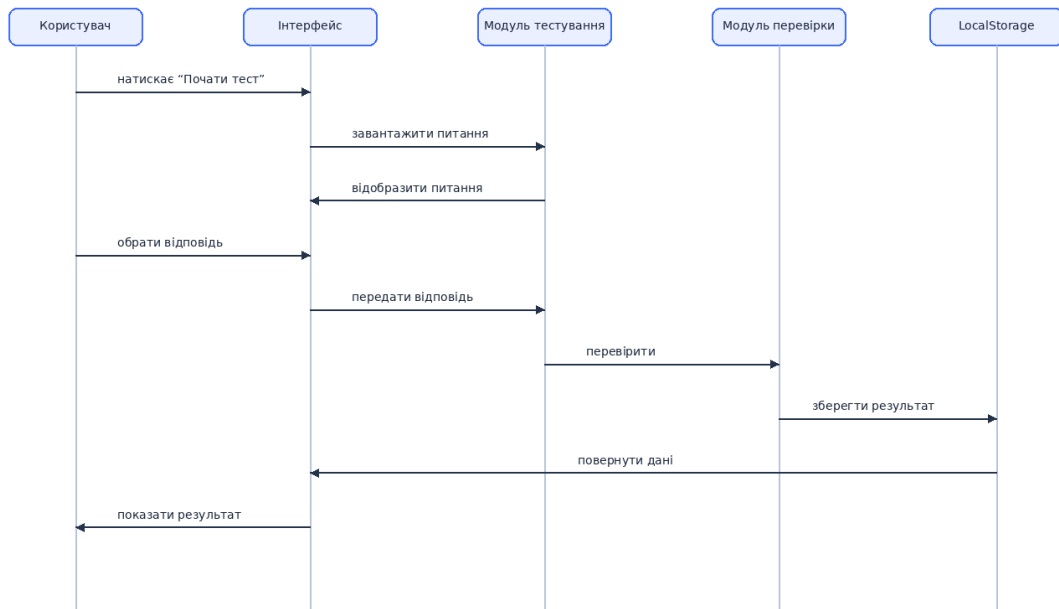


Рисунок А.3 - UML-діаграма послідовності проходження тестування

### UML-діаграма компонентів

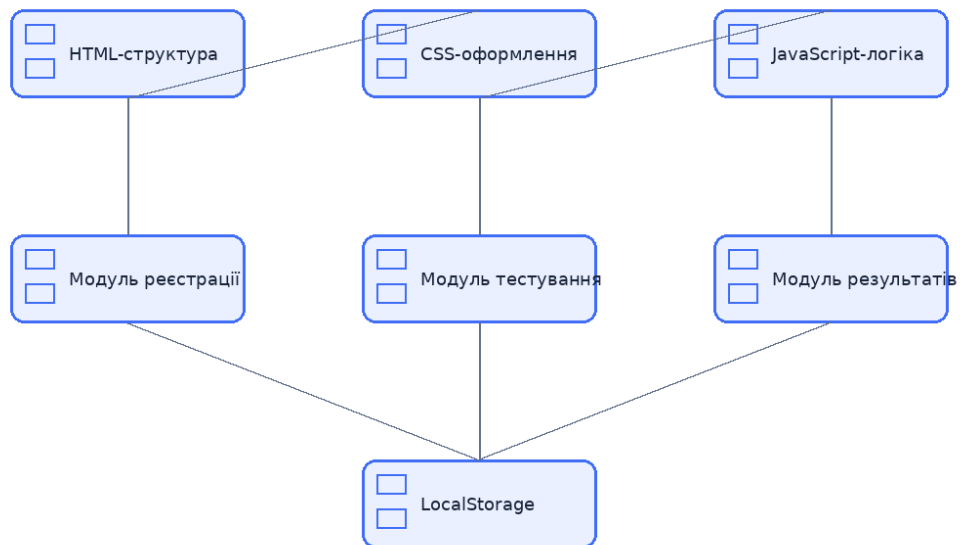


Рисунок А.4 - UML-діаграма компонентів програмного забезпечення

# ДОДАТОК Б. СКРІНШОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

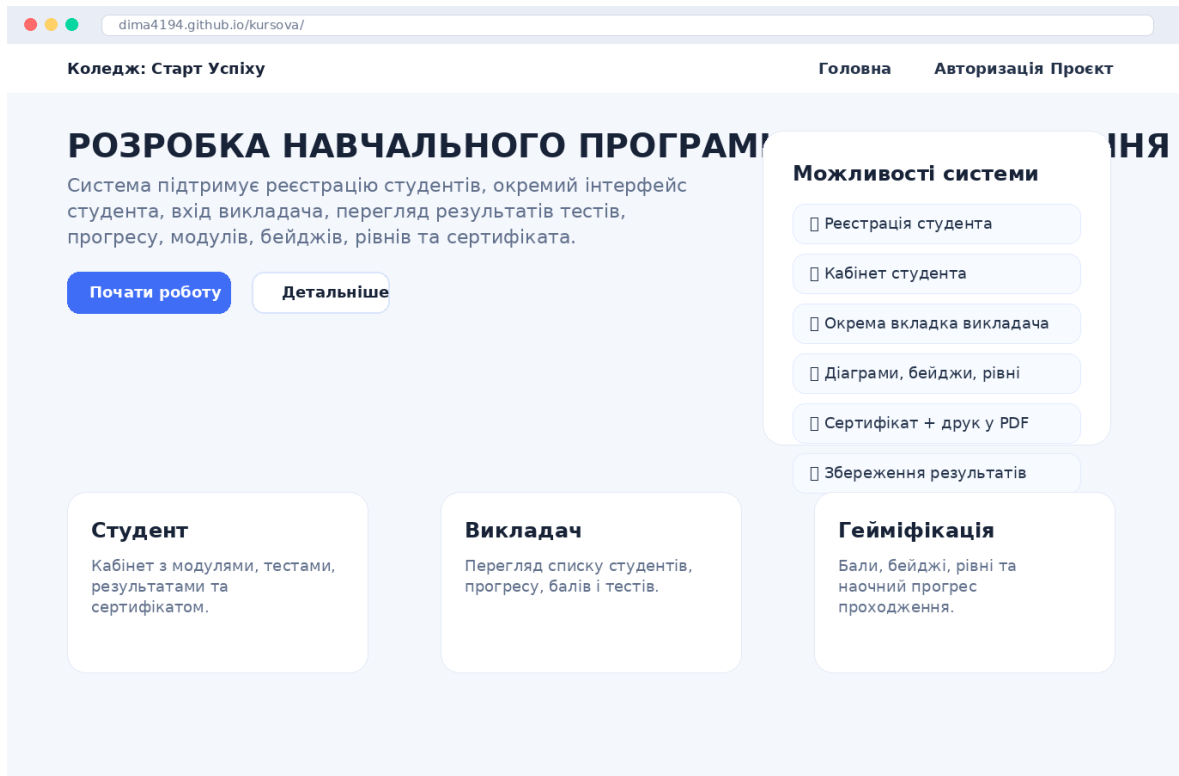
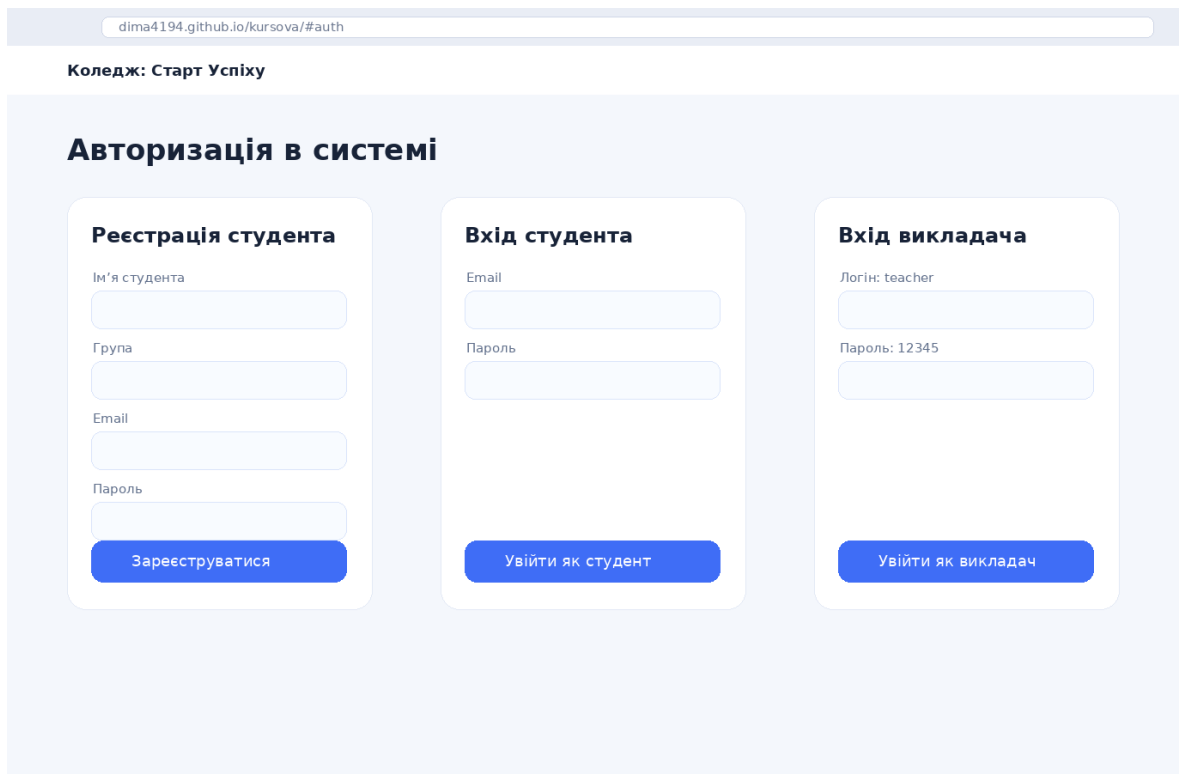
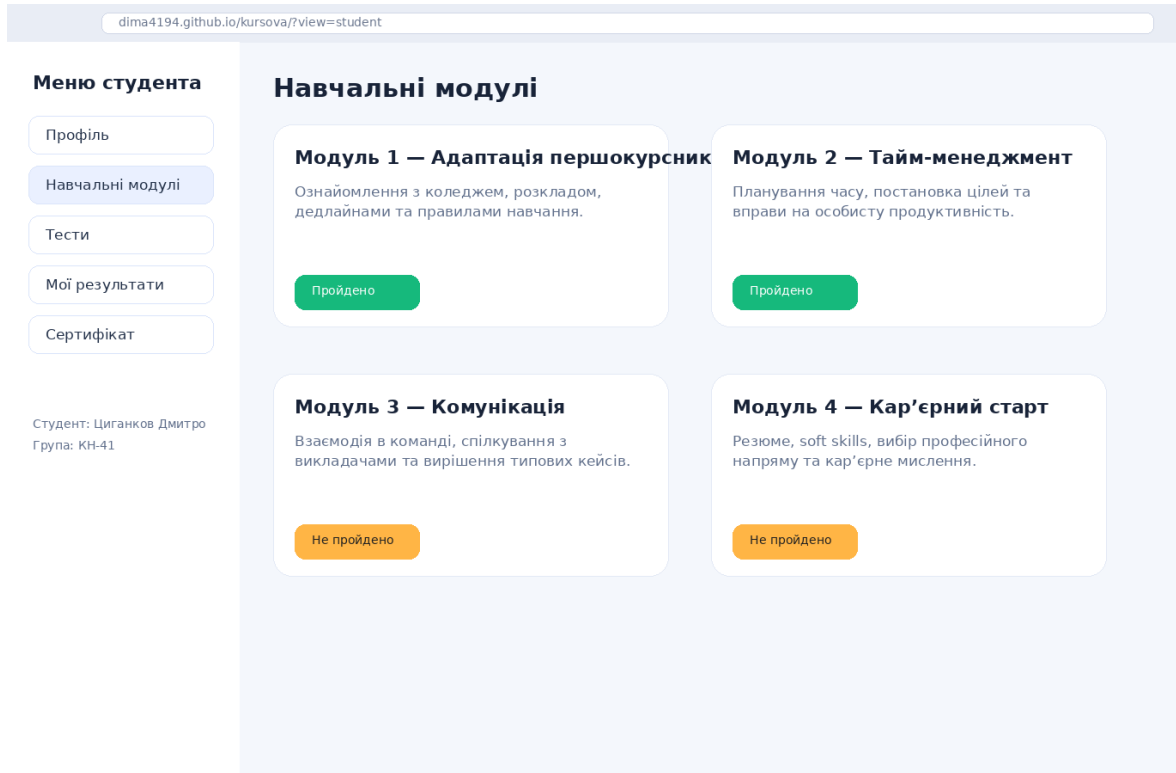


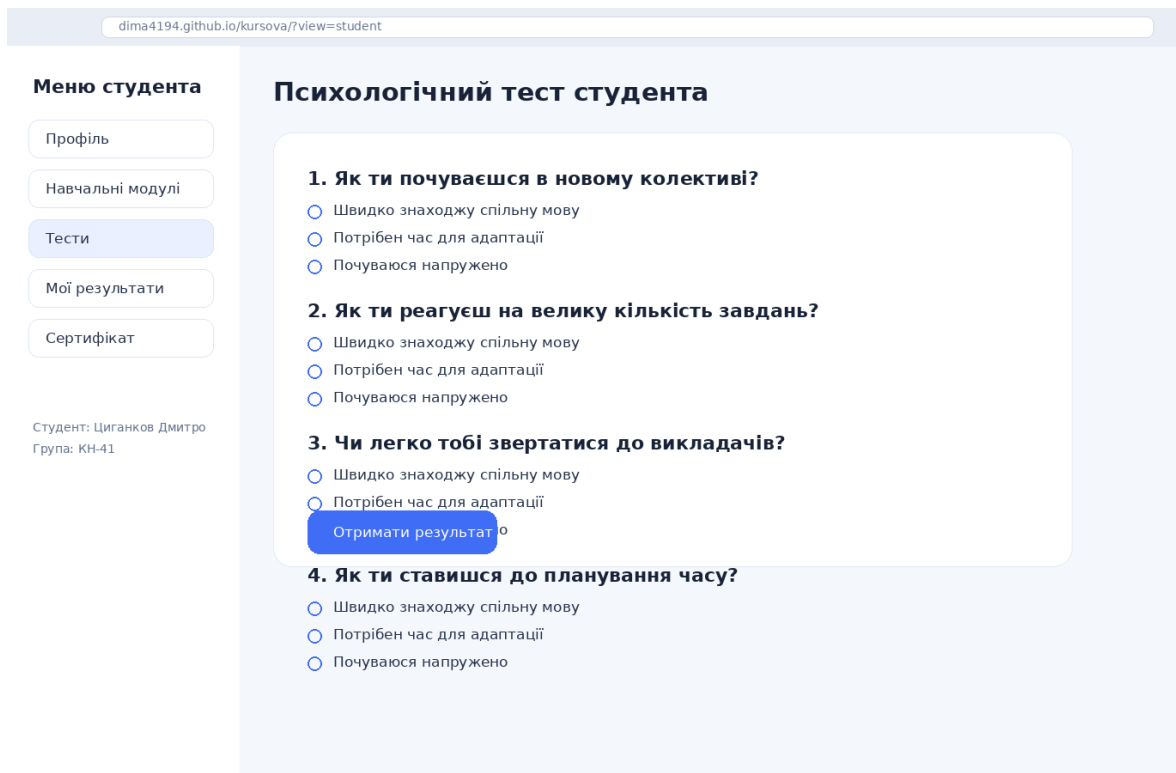
Рисунок Б.1 - Головна сторінка тренінгу



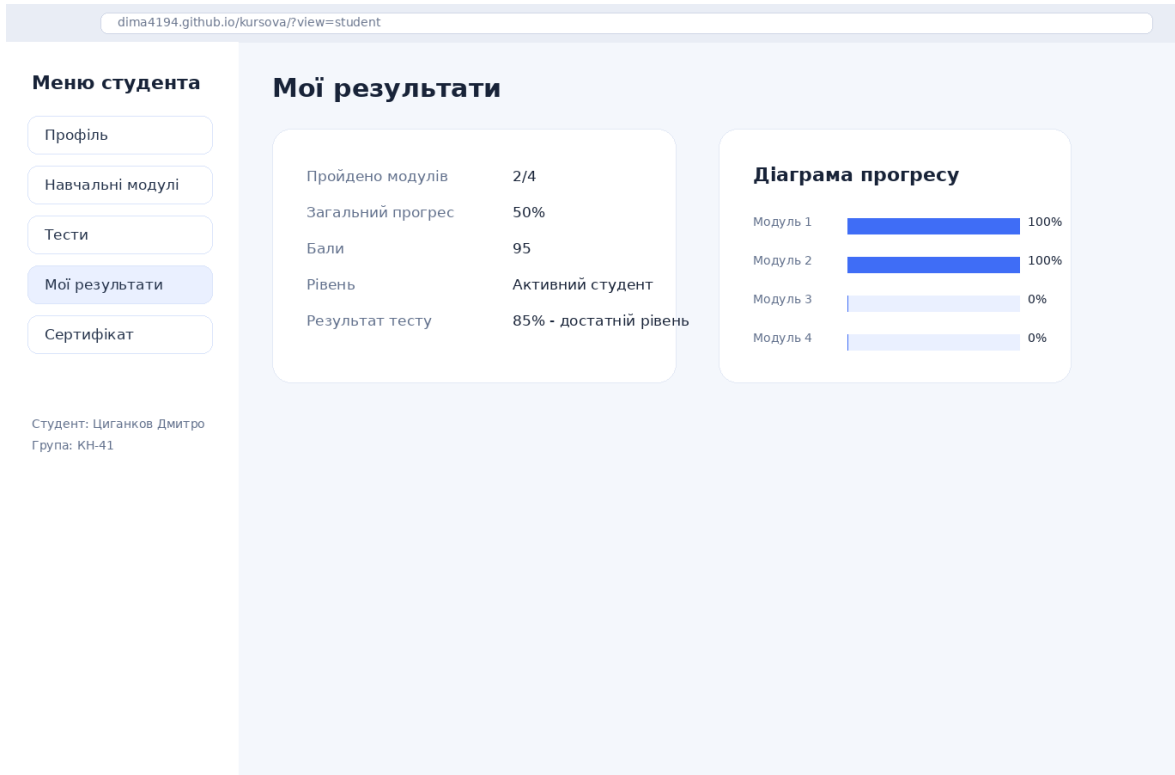
## Рисунок Б.2 - Форма реєстрації та авторизації



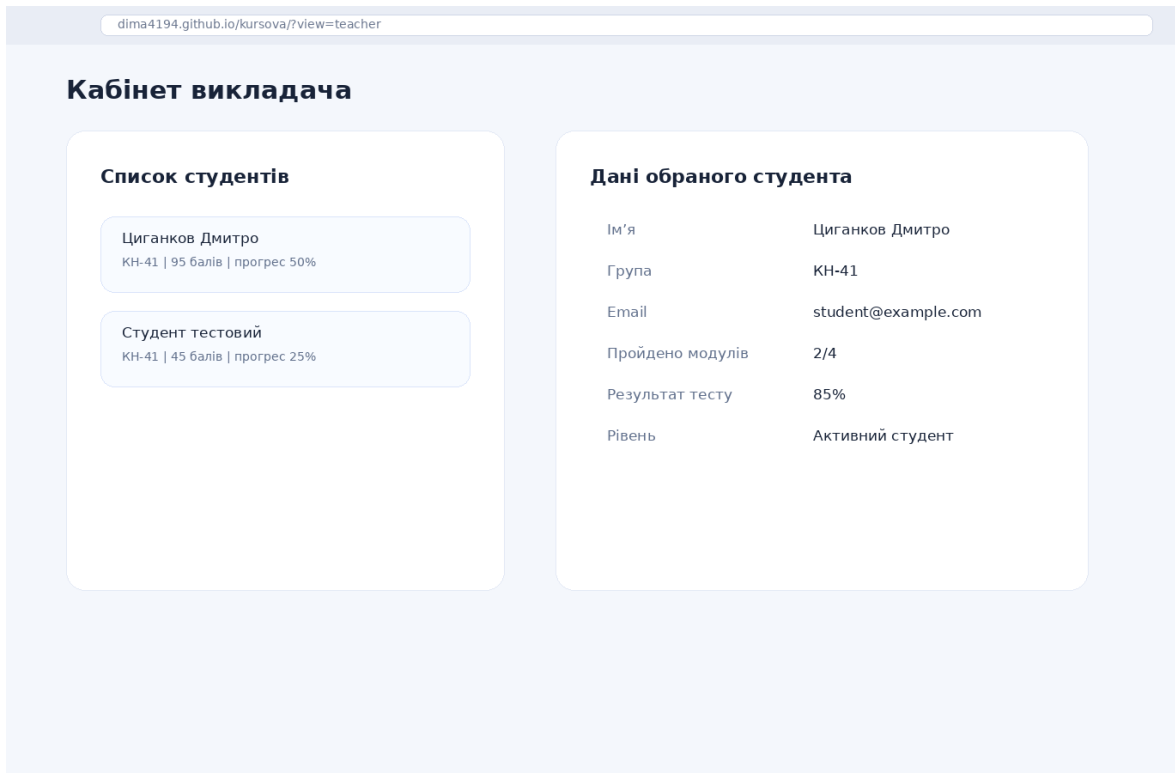
## Рисунок Б.3 - Навчальні модулі студента



## Рисунок Б.4 - Тестування користувача



## Рисунок Б.5 - Результати тестування



## Рисунок Б.6 - Панель викладача

### ДОДАТОК В. ФРАГМЕНТИ ПРОГРАМНОГО КОДУ

У додатку наведено повний програмний код із курсової роботи: файл index.html та JavaScript-логіка проекту.

#### Лістинг В.1 - Файл index.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Коледж: Старт Успіху</title>
  <style>
    :root{
      --bg:#f4f7fc;
      --white:#ffffff;
      --text:#24324a;
      --muted:#72809b;
      --primary:#3f6df6;
      --secondary:#16b97c;
      --warning:#ffb545;
      --danger:#ff6677;
      --dark:#182338;
      --border:#e4eaf6;
      --shadow:0 12px 28px rgba(24,35,56,0.10);
      --radius:22px;
    }

    *{
      margin:0;
      padding:0;
      box-sizing:border-box;
    }

    html{
      scroll-behavior:smooth;
    }

    body{
      font-family: Arial, Helvetica, sans-serif;
      background:linear-gradient(180deg,#eef4ff 0%,#f8fbff 35%,#f4f7fc 100%);
      color:var(--text);
      line-height:1.6;
    }

    a{
      text-decoration:none;
      color:inherit;
    }

    .container{
      width:min(1240px, 94%);
      margin:0 auto;
    }

    .hidden{
```

```
    display:none !important;
}

.btn{
  border:none;
  cursor:pointer;
  border-radius:14px;
  padding:12px 18px;
  font-size:1rem;
  font-weight:700;
  transition:0.25s;
}

.btn:hover{
  transform:translateY(-2px);
}

.btn-primary{
  background:var(--primary);
  color:#fff;
  box-shadow:0 8px 18px rgba(63,109,246,0.28);
}

.btn-secondary{
  background:#fff;
  color:var(--dark);
  border:2px solid #dbe5fb;
}

.btn-success{
  background:var(--secondary);
  color:#fff;
}

.btn-warning{
  background:var(--warning);
  color:#222;
}

.btn-danger{
  background:var(--danger);
  color:#fff;
}

header{
  position:sticky;
  top:0;
  z-index:1000;
  background:rgba(255,255,255,0.92);
  backdrop-filter:blur(12px);
  box-shadow:0 4px 18px rgba(0,0,0,0.06);
}

.nav{
  display:flex;
  justify-content:space-between;
  align-items:center;
  gap:20px;
  padding:16px 0;
  flex-wrap:wrap;
}

.logo{
  font-size:1.25rem;
}
```

```
    font-weight:800;
    color:var(--dark);
}

.logo span{
  color:var(--primary);
}

nav ul{
  list-style:none;
  display:flex;
  flex-wrap:wrap;
  gap:16px;
}

nav a{
  font-weight:700;
  color:var(--dark);
  transition:0.25s;
}

nav a:hover{
  color:var(--primary);
}

.hero{
  padding:82px 0 60px;
}

.hero-grid{
  display:grid;
  grid-template-columns:1.15fr 0.85fr;
  gap:28px;
  align-items:center;
}

.badge{
  display:inline-block;
  padding:8px 14px;
  border-radius:999px;
  background:#eaf0ff;
  color:var(--primary);
  font-weight:700;
  margin-bottom:18px;
}

.hero h1{
  font-size:clamp(2rem,4.7vw,3.4rem);
  line-height:1.12;
  margin-bottom:16px;
  color:var(--dark);
}

.hero p{
  color:var(--muted);
  margin-bottom:24px;
  font-size:1.05rem;
  max-width:730px;
}

.hero-actions{
  display:flex;
  flex-wrap:wrap;
  gap:14px;
```

```
}

.hero-card, .card, .panel, .auth-box, .chart-box, .certificate{
  background:var(--white);
  border-radius:var(--radius);
  box-shadow:var(--shadow);
  border:1px solid var(--border);
}

.hero-card{
  padding:28px;
  position:relative;
  overflow:hidden;
}

.hero-card::before{
  content:"";
  position:absolute;
  width:190px;
  height:190px;
  border-radius:50%;
  background:linear-gradient(135deg, rgba(63,109,246,0.18), rgba(22,185,124,0.18));
  right:-60px;
  top:-60px;
}

.hero-card h3{
  margin-bottom:16px;
  color:var(--dark);
  position:relative;
  z-index:1;
}

.mini-list{
  display:grid;
  gap:12px;
  position:relative;
  z-index:1;
}

.mini-item{
  padding:14px 16px;
  border-radius:14px;
  background:#f7fbff;
  border:1px solid #e3ebff;font-weight:700;
}

section{
  padding:68px 0;
}

.section-title{
  text-align:center;
  font-size:2rem;
  color:var(--dark);
  margin-bottom:14px;
}

.section-subtitle{
  text-align:center;
  max-width:860px;
  margin:0 auto 38px;
  color:var(--muted);
}
```

```
.grid-2{
  display:grid;
  grid-template-columns:1fr 1fr;
  gap:24px;
}

.grid-3{
  display:grid;
  grid-template-columns:repeat(3,1fr);
  gap:24px;
}

.grid-4{
  display:grid;
  grid-template-columns:repeat(4,1fr);
  gap:20px;
}

.card, .panel, .auth-box, .chart-box{
  padding:24px;
}

.card h3, .panel h3, .auth-box h3, .chart-box h3{
  margin-bottom:12px;
  color:var(--dark);
}

.card p, .panel p{
  color:var(--muted);
}

.icon{
  width:56px;
  height:56px;
  border-radius:16px;
  display:flex;
  align-items:center;
  justify-content:center;
  font-size:1.5rem;
  margin-bottom:14px;
  background:linear-gradient(135deg,#ebf1ff,#ecfff7);
}

.form-grid{
  display:grid;
  gap:14px;
}

input, select, textarea{
  width:100%;
  padding:14px 16px;
  border-radius:14px;
  border:1px solid #dbe3f5;
  outline:none;
  font-size:1rem;
}

input:focus, select:focus, textarea:focus{
  border-color:var(--primary);
  box-shadow:0 0 0 4px rgba(63,109,246,0.12);
}
```

```
textarea{
  min-height:120px;
  resize:vertical;
}

.success-box, .info-box{
  margin-top:14px;
  display:none;
  padding:14px 16px;
  border-radius:14px;
  font-weight:700;
}

.success-box{
  background:#eafbf3;
  border:1px solid #bfe8d2;
  color:#0c8458;
}

.info-box{
  background:#eef4ff;
  border:1px solid #d2e1ff;
  color:#2451bf;
}

footer{
  margin-top:40px;
  background:var(--dark);
  color:#d6e1fd;
  padding:30px 0;
}

.footer-grid{
  display:grid;
  grid-template-columns:2fr 1fr 1fr;
  gap:22px;
}

footer h4{
  color:#fff;
  margin-bottom:12px;
}

footer ul{
  list-style:none;
  display:grid;
  gap:8px;
}

.center{ text-align:center; }
.muted{ color:var(--muted); }
.mt-18{ margin-top:18px; }
.mb-16{ margin-bottom:16px; }

.student-layout{
  display:grid;
  grid-template-columns:280px 1fr;
  gap:24px;
  align-items:start;
}

.sidebar{
  background:var(--white);
  border-radius:var(--radius);
}
```

```
    box-shadow:var(--shadow);
    border:1px solid var(--border);
    padding:22px;
    position:sticky;
    top:22px;
}

.sidebar h3{
    color:var(--dark);
    margin-bottom:12px;
}

.profile-mini{
    background:#f8fbff;
    border:1px solid #e5ecfc;
    border-radius:18px;
    padding:16px;
    margin-bottom:16px;
}

.profile-mini p{
    margin-bottom:6px;
    font-size:0.95rem;
    color:var(--muted);
}

.menu-list{
    display:grid;
    gap:10px;
    margin-top:14px;
}

.menu-btn{
    text-align:left;
    width:100%;
    background:#f5f8ff;
    border:1px solid #dfe8fb;
    color:var(--dark);
    border-radius:14px;
    padding:14px 16px;
    cursor:pointer;
    font-weight:700;
    transition:0.25s;
}

.menu-btn.active,
.menu-btn:hover{
    background:var(--primary);
    color:#fff;
    border-color:var(--primary);
}

.student-content{
    display:grid;
    gap:22px;
}

.student-tab{
    display:none;
}

.student-tab.active{
    display:block;
}
```

```
.stats{
  display:grid;
  grid-template-columns:repeat(4,1fr);
  gap:16px;
  margin-top:18px;
}

.stat{
  background:linear-gradient(180deg,#fff,#f7fbff);
  border:1px solid var(--border);
  border-radius:18px;
  box-shadow:var(--shadow);
  padding:18px;
  text-align:center;
}

.stat b{display:block;
  font-size:1.65rem;
  color:var(--dark);
  margin-bottom:5px;
}

.progress-wrap{
  height:14px;
  border-radius:999px;
  background:#edf2fb;
  overflow:hidden;
  margin-top:14px;
}

.progress-bar{
  height:100%;
  width:0%;
  background:linear-gradient(90deg,var(--primary),var(--secondary));
  transition:0.5s;
}

.progress-text{
  margin-top:10px;
  color:var(--muted);
  font-weight:700;
}

.badge-list{
  display:flex;
  flex-wrap:wrap;
  gap:10px;
  margin-top:14px;
}

.student-badge{
  padding:10px 14px;
  border-radius:999px;
  font-weight:700;
  border:1px solid #dfe7f8;
  background:#f2f6fc;
  color:#91a0ba;
}

.student-badge.active{
  background:#fff4dd;
  color:#8c5d00;
  border-color:#ffdca0;
}
```

```
.level-list{
  display:grid;
  gap:10px;
  margin-top:16px;
}

.level-item{
  background:#f8fbff;
  border:1px solid #e3ebff;
  border-radius:14px;
  padding:12px 14px;
  font-weight:700;
  color:var(--dark);
}

.level-item.active{
  background:#eafb3;
  border-color:#bfe8d2;
  color:#0c8458;
}

.module-card{
  display:grid;
  gap:14px;
}

.module-top{
  display:flex;
  justify-content:space-between;
  gap:14px;
  align-items:flex-start;
}

.module-number{
  width:44px;
  height:44px;
  border-radius:50%;
  display:flex;
  align-items:center;
  justify-content:center;
  background:var(--primary);
  color:#fff;
  font-weight:800;
  flex-shrink:0;
}

.module-info{
  display:flex;
  gap:14px;
  align-items:flex-start;
}

.module-list{
  background:#f8fbff;
  border:1px solid #e4ecff;
  border-radius:16px;
  padding:14px 16px;
}

.module-list li{
  margin-bottom:8px;
}
```

```
.status{
  display:inline-block;
  padding:8px 12px;
  border-radius:999px;
  background:#eef3fa;
  color:var(--muted);
  font-size:0.9rem;
  font-weight:700;
  white-space:nowrap;
}

.status.done{
  background:#eafbf3;
  color:#0d8659;
}

.quiz-options{
  display:grid;
  gap:10px;
  margin-top:14px;
}

.quiz-option{
  border:1px solid #ele8fb;
  background:#fff;
  border-radius:14px;
  padding:14px;
  cursor:pointer;
  font-weight:700;
  transition:0.25s;
}

.quiz-option:hover{
  border-color:var(--primary);
}

.quiz-option.correct{
  background:#eafbf3;
  border-color:#afe6cb;
  color:#0d8156;
}

.quiz-option.wrong{
  background:#fff0f2;
  border-color:#ffc5cd;
  color:#c0394c;
}

.result{
  display:none;
  margin-top:14px;
  padding:14px 16px;
  border-radius:14px;
  font-weight:700;
}

.result.show{
  display:block;
}

.result.success{
  background:#eafbf3;
  border:1px solid #bce8d1;
  color:#0c8558;
}
```

```
.result.fail{
  background:#fff1f2;
  border:1px solid #ffd0d5;
  color:#c3384b;
}

.test-question{
  font-weight:700;
  color:var(--dark);
  margin-bottom:12px;
}

.test-answers{
  display:grid;
  gap:10px;
  margin-bottom:18px;
}

.test-answers label{
  display:flex;
  gap:10px;
  align-items:flex-start;
  padding:12px 14px;
  border-radius:14px;
  background:#f9fbff;
  border:1px solid #e4ebfb;
  cursor:pointer;
}

.teacher-layout{
  display:grid;
  grid-template-columns:360px 1fr;
  gap:24px;
  align-items:start;
}

.student-list{
  display:grid;
  gap:12px;
  max-height:650px;
  overflow:auto;
  padding-right:4px;
}

.student-item{
  padding:16px;
  border-radius:16px;
  border:1px solid #e2e9f8;
  background:#f9fbff;
  cursor:pointer;
  transition:0.25s;}

.student-item:hover,
.student-item.active{
  background:#edf3ff;
  border-color:#bdd0ff;
}

.student-item h4{
  color:var(--dark);
  margin-bottom:4px;
}
```

```
.student-item p{
  color:var(--muted);
  font-size:0.94rem;
}

.teacher-table{
  width:100%;
  border-collapse:collapse;
  margin-top:10px;
}

.teacher-table th,
.teacher-table td{
  border:1px solid #e5ebf7;
  padding:12px;
  text-align:left;
  vertical-align:top;
}

.teacher-table th{
  background:#f3f7ff;
  color:var(--dark);
}

.chart-box{
  margin-top:20px;
}

.bar-chart{
  height:230px;
  display:flex;
  align-items:flex-end;
  justify-content:space-between;
  gap:14px;
  margin-top:18px;
}

.bar-item{
  flex:1;
  display:flex;
  flex-direction:column;
  align-items:center;
  justify-content:flex-end;
  gap:10px;
  height:100%;
}

.bar{
  width:100%;
  max-width:72px;
  min-height:20px;
  border-radius:16px 16px 8px 8px;
  background:linear-gradient(180deg, var(--primary), var(--secondary));
  color:#fff;
  font-weight:700;
  font-size:0.9rem;
  display:flex;
  align-items:flex-start;
  justify-content:center;
  padding-top:8px;
  transition:0.5s;
}

.bar-label{
  font-weight:700;
  color:var(--dark);
}
```

```
    text-align:center;
    font-size:0.93rem;
}

.certificate{
  border:4px solid #e0e9ff;
  padding:34px;
  text-align:center;
}

.certificate h3{
  font-size:2rem;
  color:var(--dark);
  margin-bottom:10px;
}

.certificate .big{
  font-size:1.35rem;
  color:var(--primary);
  font-weight:800;
  margin:14px 0;
}

.certificate .line{
  width:150px;
  height:3px;
  margin:18px auto;
  border-radius:999px;
  background:linear-gradient(90deg,var(--primary),var(--secondary));
}

@media (max-width: 1100px){
  .hero-grid,
  .grid-4,
  .grid-3,
  .grid-2,
  .teacher-layout,
  .student-layout,
  .footer-grid{
    grid-template-columns:1fr;
  }

  .stats{
    grid-template-columns:repeat(2,1fr);
  }

  .sidebar{
    position:static;
  }
}

@media (max-width: 700px){
  .nav{
    flex-direction:column;
    align-items:flex-start;
  }

  .hero{
    padding-top:50px;
  }

  .stats{
    grid-template-columns:1fr;
  }
}
```

```

}

@media print{
  header, footer, .sidebar, .btn, .menu-list, .student-tab{
    display:none !important;
  }

  #studentTabCertificate{
    display:block !important;
  }

  body{
    background:#fff;
  }
}
</style>
</head>
<body>

<!-- HOME PAGE -->
<div id="homePage">
  <header>
    <div class="container nav">
      <div class="logo">Коледж: <span>Старт Успіху</span></div>
      <nav>
        <ul>
          <li><a href="#home">Головна</a></li>
          <li><a href="#auth">Авторизація</a></li>
          <li><a href="#info">Проект</a></li>
        </ul>
      </nav>
    </div>
  </header>

  <main>
    <section class="hero" id="home">
      <div class="container hero-grid">
        <div>
          <div class="badge">Навчальне програмне забезпечення тренінгу</div>
          <h1>РОЗРОБКА НАВЧАЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТРЕНІНГУ «КОЛЕДЖ: СТАРТ УСПІХУ»</h1>
          <p>
            Система підтримує реєстрацію студентів, окремий інтерфейс студента в новій вкладці,
            вхід викладача, перегляд результатів тестів, прогресу, модулів, бейджів, рівнів та сертифіката.
          </p>
          <div class="hero-actions">
            <a href="#auth" class="btn btn-primary">Почати роботу</a>
            <a href="#info" class="btn btn-secondary">Детальніше</a>
          </div>
        </div>

        <div class="hero-card">
          <h3>Можливості системи</h3><div class="mini-list">
            <div class="mini-item">☐ Реєстрація студента</div>
            <div class="mini-item">☐ Нова вкладка з повним кабінетом студента</div>
            <div class="mini-item">☐ Окрема вкладка викладача</div>
            <div class="mini-item">☐ Діаграми, бейджи, рівні</div>
            <div class="mini-item">☐ Сертифікат + друк у PDF</div>
            <div class="mini-item">☐ Збереження всіх результатів</div>
          </div>
        </div>
      </div>
    </section>

    <section id="auth">
      <div class="container">
        <h2 class="section-title">Авторизація в системі</h2>

```

<p class="section-subtitle">  
Студент після реєстрації або входу відкривається в окремій вкладці зі своїм повним інтерфейсом. Викладач входить у власну вкладку з переглядом усіх студентів.  
</p>

```
<div class="grid-3">  
  <div class="auth-box">  
    <h3>Реєстрація студента</h3>  
    <div class="form-grid">  
      <input type="text" id="regName" placeholder="Ім'я та прізвище" />  
      <input type="text" id="regGroup" placeholder="Група" />  
      <input type="email" id="regEmail" placeholder="Email" />  
      <input type="password" id="regPassword" placeholder="Пароль" />  
      <button class="btn btn-primary"  
onclick="registerStudent()">Зареєструватися</button>  
    </div>  
    <div class="success-box" id="registerSuccess"></div>  
  </div>  
  
  <div class="auth-box">  
    <h3>Вхід студента</h3>  
    <div class="form-grid">  
      <input type="email" id="studentLoginEmail" placeholder="Email" />  
      <input type="password" id="studentLoginPassword" placeholder="Пароль" />  
      <button class="btn btn-success" onclick="loginStudent()">Увійти як студент</button>  
    </div>  
    <div class="info-box" id="studentLoginInfo"></div>  
  </div>  
  
  <div class="auth-box">  
    <h3>Вхід викладача</h3>  
    <div class="form-grid">  
      <input type="text" id="teacherLoginName" placeholder="Логін викладача" />  
      <input type="password" id="teacherLoginPassword" placeholder="Пароль" />  
      <button class="btn btn-warning" onclick="loginTeacher()">Увійти як  
викладач</button>  
    </div>  
    <p class="muted mt-18">  
      Демонстраційний логін: <b>teacher</b><br>  
      Пароль: <b>12345</b>  
    </p>  
    <div class="info-box" id="teacherLoginInfo"></div>  
  </div>  
</div>  
</section>  
  
<section id="info">  
  <div class="container">  
    <h2 class="section-title">Що є в проекті</h2>  
    <p class="section-subtitle">  
      Цей варіант уже підходить як сильна база для курсової та дипломної роботи.  
    </p>  
  
    <div class="grid-4">  
      <div class="card">  
        <div class="icon">□</div>  
        <h3>Студент</h3>  
        <p>Окремий кабінет з меню, модулями, тестами, результатами, діаграмами і  
сертифікатом.</p>  
      </div>  
      <div class="card">  
        <div class="icon">□□</div>  
        <h3>Викладач</h3>  
        <p>Перегляд списку студентів, прогресу, балів, тестів та деталей по кожному.</p>  
      </div>  
      <div class="card">  
        <div class="icon">□</div>
```

```

        <h3>Гейміфікація</h3>
        <p>Бали, бейджі, рівні та наочний прогрес стимулюють проходження тренінгу.</p>
    </div>
    <div class="card">
        <div class="icon">□</div>
        <h3>PDF сертифікат</h3>
        <p>Сертифікат можна зберегти через друк у PDF прямо з браузера.</p>
    </div>
</div>
</section>
</main>

<footer>
    <div class="container footer-grid">
        <div>
            <h4>Коледж: Старт Успіху</h4><p>Демонстраційний навчальний вебпроект для курсової та дипломної роботи.</p>
        </div>
        <div>
            <h4>Розділи</h4>
            <ul>
                <li><a href="#auth">Авторизація</a></li>
                <li><a href="#info">Проект</a></li>
            </ul>
        </div>
        <div>
            <h4>Технології</h4>
            <ul>
                <li>HTML5</li>
                <li>CSS3</li>
                <li>JavaScript</li>
                <li>LocalStorage</li>
            </ul>
        </div>
    </div>
</footer>
</div>

<!-- STUDENT PAGE -->
<div id="studentPage" class="hidden">
    <header>
        <div class="container nav">
            <div class="logo">Коледж: <span>Старт Успіху</span></div>
            <nav>
                <ul>
                    <li><a href="#" onclick="switchStudentTab('studentTabProfile'); return false;">Профіль</a></li>
                    <li><a href="#" onclick="switchStudentTab('studentTabModules'); return false;">Модулі</a></li>
                    <li><a href="#" onclick="switchStudentTab('studentTabTests'); return false;">Тести</a></li>
                    <li><a href="#" onclick="switchStudentTab('studentTabResults'); return false;">Результати</a></li>
                </ul>
            </nav>
        </div>
    </header>

    <main>
        <section style="padding:38px 0 24px;">
            <div class="container">
                <h2 class="section-title">Особистий кабінет студента</h2>
                <p class="section-subtitle">
                    Повний інтерфейс студента з модулями, тестуванням, рівнями, бейджами та сертифікатом.
                </p>

                <div class="student-layout">
                    <aside class="sidebar">

```

```

<h3>Меню студента</h3>
<div class="profile-mini">
  <p><b>Студент:</b> <span id="sidebarStudentName"></span></p>
  <p><b>Група:</b> <span id="sidebarStudentGroup"></span></p>
  <p><b>Email:</b> <span id="sidebarStudentEmail"></span></p>
  <p><b>Роль:</b> студент</p>
</div>

<div class="menu-list">
  <button class="menu-btn active" data-tab="studentTabProfile"
onclick="switchStudentTab('studentTabProfile', this)">Профіль</button>
  <button class="menu-btn" data-tab="studentTabModules"
onclick="switchStudentTab('studentTabModules', this)">Навчальні модулі</button>
  <button class="menu-btn" data-tab="studentTabTests"
onclick="switchStudentTab('studentTabTests', this)">Тести</button>
  <button class="menu-btn" data-tab="studentTabResults"
onclick="switchStudentTab('studentTabResults', this)">Мої результати</button>
  <button class="menu-btn" data-tab="studentTabData"
onclick="switchStudentTab('studentTabData', this)">Моя інформація</button>
  <button class="menu-btn" data-tab="studentTabCertificate"
onclick="switchStudentTab('studentTabCertificate', this)">Сертифікат</button>
</div>

<div class="mt-18">
  <button class="btn btn-secondary" style="width:100%;"
onclick="logoutStudent()">Вийти</button>
</div>
</aside>

<div class="student-content">
  <div class="student-tab active" id="studentTabProfile">
    <div class="panel">
      <h3>Профіль студента</h3>
      <p class="muted">Тут відображаються прогрес, бали, тести, рівень і
досягнення.</p>

      <div class="stats">
        <div class="stat">
          <b id="profileModules">0/4</b>
          модулів
        </div>
        <div class="stat">
          <b id="profilePoints">0</b>
          балів
        </div>
        <div class="stat">
          <b id="profileTests">0</b>
          тестів
        </div>
        <div class="stat">
          <b id="profileLevel">Новачок</b>
          рівень
        </div>
      </div><div class="progress-wrap">
        <div class="progress-bar" id="studentProgressBar"></div>
      </div>
      <div class="progress-text" id="studentProgressText">0% пройдено</div>

      <div class="badge-list">
        <span class="student-badge" id="sBadge1"><input type="checkbox"/> Адаптований студент</span>
        <span class="student-badge" id="sBadge2"><input type="checkbox"/> Активний студент</span>
        <span class="student-badge" id="sBadge3"><input type="checkbox"/> Лідер групи</span>
      </div>

      <div class="level-list">
        <div class="level-item" id="level1">50 балів – Новачок</div>
        <div class="level-item" id="level2">100 балів – Активний студент</div>
    </div>
  </div>
</div>

```

```

        <div class="level-item" id="level3">150 балів – Лідер групи</div>
    </div>
</div>
</div>

<div class="student-tab" id="studentTabModules">
    <div class="grid-2">
        <div class="card module-card">
            <div class="module-top">
                <div class="module-info">
                    <div class="module-number">1</div>
                    <div>
                        <h3>Модуль 1 – Адаптація першокурсника</h3>
                        <p class="muted">Ознайомлення з коледжем, розкладом, дедлайнами та
правилами навчання.</p>
                    </div>
                </div>
                <span class="status" id="moduleStatus1">Не пройдено</span>
            </div>
            <div class="module-list">
                <ul>
                    <li>як влаштований коледж;</li>
                    <li>розклад і дедлайни;</li>
                    <li>поради новачкам;</li>
                    <li>адаптація до нових умов.</li>
                </ul>
            </div>
            <button class="btn btn-success" onclick="completeModule(1)">Пройдено</button>
        </div>

        <div class="card module-card">
            <div class="module-top">
                <div class="module-info">
                    <div class="module-number">2</div>
                    <div>
                        <h3>Модуль 2 – Тайм-менеджмент</h3>
                        <p class="muted">Планування часу, постановка цілей та вправи на особисту
продуктивність.</p>
                    </div>
                </div>
                <span class="status" id="moduleStatus2">Не пройдено</span>
            </div>
            <div class="module-list">
                <ul>
                    <li>планування дня;</li>
                    <li>постановка цілей;</li>
                    <li>самоконтроль;</li>
                    <li>антипрокрастинація.</li>
                </ul>
            </div>
            <button class="btn btn-success" onclick="completeModule(2)">Пройдено</button>
        </div>

        <div class="card module-card">
            <div class="module-top">
                <div class="module-info">
                    <div class="module-number">3</div>
                    <div>
                        <h3>Модуль 3 – Комунікація</h3>
                        <p class="muted">Взаємодія в команді, спілкування з викладачами та
вирішення типових кейсів.</p>
                    </div>
                </div>
                <span class="status" id="moduleStatus3">Не пройдено</span>
            </div>
            <div class="module-list">
                <ul>
                    <li>робота в команді;</li>
                    <li>спілкування з викладачами;</li>
                </ul>
            </div>
        </div>
    </div>
</div>

```

```

        <li>комунікація в групі;</li>
        <li>ситуаційні кейси.</li>
    </ul>
</div><button class="btn btn-success"
onclick="completeModule(3)">Пройдено</button>
</div>

<div class="card module-card">
<div class="module-top">
<div class="module-info">
<div class="module-number">4</div>
<div>
<h3>Модуль 4 – Кар'єрний старт</h3>
<p class="muted">Резюме, soft skills, вибір професійного напрямку та
кар'єрне мислення.</p>
</div>
</div>
<span class="status" id="moduleStatus4">Не пройдено</span>
</div>
<div class="module-list">
<ul>
<li>резюме;</li>
<li>soft skills;</li>
<li>вибір напрямку;</li>
<li>план кар'єрного старту.</li>
</ul>
</div>
<button class="btn btn-success" onclick="completeModule(4)">Пройдено</button>
</div>
</div>
</div>

<div class="student-tab" id="studentTabTests">
<div class="panel mb-16">
<h3>Ситуаційний міні-квіз</h3>
<p><b>Ситуація:</b> Ти запізнюєшся на дедлайн. Що робиш?</p>

<div class="quiz-options" id="quiz1">
<div class="quiz-option" data-correct="false">Ігнорую проблему і нічого не
повідомляю</div>
<div class="quiz-option" data-correct="true">Оцінюю ситуацію, повідомляю
викладача та складаю план завершення</div>
<div class="quiz-option" data-correct="false">Просто чекаю, що проблема зникне
сама</div>
</div>

<div class="result" id="quiz1Result"></div>
</div>

<div class="panel">
<h3>Психологічний тест студента</h3>
<p class="muted">Оберіть по одному варіанту відповіді.</p>

<form id="psychTest" style="margin-top:18px;">
<div class="test-question">1. Як ти почуваєшся в новому колективі?</div>
<div class="test-answers">
<label><input type="radio" name="q1" value="3"> Швидко знаходжу спільну
мову</label>
<label><input type="radio" name="q1" value="2"> Потрібен час для
адаптації</label>
<label><input type="radio" name="q1" value="1"> Почуваюся напружено</label>
</div>

<div class="test-question">2. Як ти реагуєш на велику кількість завдань?</div>
<div class="test-answers">

```

```

    поетапно</label>
    <label><input type="radio" name="q2" value="3"> Складаю план і працюю
    вপরতিся</label>
    <label><input type="radio" name="q2" value="2"> Хвилююся, але намагаюся
    відкладаю</label>
    <label><input type="radio" name="q2" value="1"> Розгублююся й
  </div>

  <div class="test-question">3. Чи легко тобі звертатися до викладачів?</div>
  <div class="test-answers">
    <label><input type="radio" name="q3" value="3"> Так, легко</label>
    <label><input type="radio" name="q3" value="2"> Іноді соромлюся</label>
    <label><input type="radio" name="q3" value="1"> Майже завжди уникаю</label>
  </div>

  <div class="test-question">4. Як ти ставишся до планування часу?</div>
  <div class="test-answers">
    <label><input type="radio" name="q4" value="3"> Регулярно планую</label>
    <label><input type="radio" name="q4" value="2"> Іноді планую</label>
    <label><input type="radio" name="q4" value="1"> Практично не планую</label>
  </div><button type="button" class="btn btn-primary"
onclick="calculateTest()">Отримати результат</button>
</form>

  <div class="result" id="testResult"></div>
</div>
</div>

<div class="student-tab" id="studentTabResults">
  <div class="panel">
    <h3>Мої результати</h3>
    <table class="teacher-table">
      <thead>
        <tr>
          <th>Показник</th>
          <th>Значення</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Пройдено модулів</td>
          <td id="resultModules">0/4</td>
        </tr>
        <tr>
          <td>Загальний прогрес</td>
          <td id="resultProgress">0%</td>
        </tr>
        <tr>
          <td>Бали</td>
          <td id="resultPoints">0</td>
        </tr>
        <tr>
          <td>Рівень</td>
          <td id="resultLevel">Новачок</td>
        </tr>
        <tr>
          <td>Результат тесту</td>
          <td id="resultTestText">Тест ще не пройдено</td>
        </tr>
        <tr>
          <td>Результат квізу</td>
          <td id="resultQuizText">Квіз ще не пройдено</td>
        </tr>
      </tbody>
    </table>

    <div class="chart-box">
      <h3>Діаграма прогресу по модулях</h3>

```



```

<!-- TEACHER PAGE -->
<div id="teacherPage" class="hidden">
  <header>
    <div class="container nav">
      <div class="logo">Коледж: <span>Старт Успіху</span></div>
      <nav>
        <ul>
          <li><a href="#" onclick="return false;">Кабінет викладача</a></li>
        </ul>
      </nav>
    </div>
  </header>

  <main>
    <section style="padding:38px 0 24px;">
      <div class="container">
        <h2 class="section-title">Кабінет викладача</h2>
        <p class="section-subtitle">
          Тут викладач бачить усіх студентів і може переглядати результати кожного окремо.
        </p>

        <div class="teacher-layout">
          <div class="panel">
            <h3>Список студентів</h3>
            <div class="student-list" id="teacherStudentList"></div>
            <div class="mt-18">
              <button class="btn btn-secondary" style="width:100%;"
                onclick="logoutTeacher()">Вийти</button>
            </div>
          </div>

          <div class="panel">
            <h3>Дані обраного студента</h3>
            <div id="teacherStudentDetails">
              <p class="muted">Оберіть студента зі списку ліворуч.</p>
            </div>
          </div>
        </div>
      </section>
    </main>
  </div>

  <script src="index.js"></script>
</body>
</html>

```

## Лістинг В.2 - Файл index.js

```

const TEACHER_LOGIN = "teacher";
const TEACHER_PASSWORD = "12345";

function getStudents() {
  return JSON.parse(localStorage.getItem("students")) || [];
}

function saveStudents(students) {
  localStorage.setItem("students", JSON.stringify(students));
}

function getCurrentStudentId() {
  return localStorage.getItem("currentStudentId") || "";
}

```

```

function setCurrentStudentId(id) {
  localStorage.setItem("currentStudentId", String(id));
}

function clearCurrentStudentId() {
  localStorage.removeItem("currentStudentId");
}

function getCurrentStudent() {
  const id = getCurrentStudentId();
  if (!id) return null;
  return getStudents().find(s => String(s.id) === String(id)) || null;
}

function setTeacherLoggedIn(flag) {
  localStorage.setItem("teacherLoggedIn", flag ? "true" : "false");
}

function isTeacherLoggedIn() {
  return localStorage.getItem("teacherLoggedIn") === "true";
}

function getView() {
  const params = new URLSearchParams(window.location.search);
  return params.get("view") || "home";
}

function openWindow(type) {
  const url = window.location.pathname + "?view=" + type;
  const win = window.open(url, "_blank");
  if (!win) alert("Дозволь поп-ап у браузері");
}

function openStudentWindow() {
  openWindow("student");
}

function openTeacherWindow() {
  openWindow("teacher");
}

function getLevelByPoints(points) {
  if (points >= 150) return { level: "Лідер групи", badge: "🏆 Лідер групи" };
  if (points >= 100) return { level: "Активний студент", badge: "📌 Активний студент" };
  if (points >= 50) return { level: "Новачок", badge: "📌 Адаптований студент" };
  return { level: "Початківець", badge: "-" };
}

function calculateProgress(student) {
  const completed = Object.values(student.modules || {}).filter(Boolean).length;
  return Math.round((completed / 4) * 100);
}

function applyStudentGamification(student) {
  const data = getLevelByPoints(student.points || 0);
  student.level = data.level;
  student.badge = data.badge;
  student.progress = calculateProgress(student);
  return student;
}

```

```

function updateStudentInStorage(student) {
  let students = getStudents();
  students = students.map(s => String(s.id) === String(student.id) ? student : s);
  saveStudents(students);
  setCurrentStudentId(student.id);
}

function createEmptyStudent(name, group, email, password) {
  return {
    id: Date.now(),
    name,
    group,
    email,
    password,
    role: "student",
    modules: { 1: false, 2: false, 3: false, 4: false },
    points: 0,
    level: "Початківець",
    badge: "-",
    progress: 0,
    testsPassed: 0,
    quizPassed: false,
    quizResult: "Квіз ще не пройдено",
    testScore: 0,
    testText: "Тест ще не пройдено",
    lastLogin: new Date().toLocaleString("uk-UA")
  };
}

function showPage(pageId) {
  ["homePage", "studentPage", "teacherPage"].forEach(id => {
    document.getElementById(id)?.classList.add("hidden");
  });
  document.getElementById(pageId)?.classList.remove("hidden");
}

function setText(id, value) {
  const el = document.getElementById(id);
  if (el) el.textContent = value ?? "";
}

/* ----- REGISTER ----- */

function registerStudent() {
  const name = document.getElementById("regName").value.trim();
  const group = document.getElementById("regGroup").value.trim();
  const email = document.getElementById("regEmail").value.trim().toLowerCase();
  const password = document.getElementById("regPassword").value.trim();
  const msg = document.getElementById("registerSuccess");

  if (!name || !group || !email || !password) {
    if (msg) {
      msg.style.display = "block";
      msg.textContent = "Заповніть усі поля";
    }
    return;
  }

  const students = getStudents();
  if (students.find(s => s.email === email)) {
    if (msg) {
      msg.style.display = "block";
      msg.textContent = "Такий email вже існує";
    }
    return;
  }
}

```

```

const student = createEmptyStudent(name, group, email, password);
students.push(student);
saveStudents(students);
setCurrentStudentId(student.id);

msg.textContent = "Успішна реєстрація";
openStudentWindow();
}

/* ----- LOGIN ----- */

function loginStudent() {
const email = document.getElementById("studentLoginEmail").value.trim().toLowerCase();
const password = document.getElementById("studentLoginPassword").value.trim();

const students = getStudents();
const student = students.find(s => s.email === email && s.password === password);

if (!student) return alert("Невірний логін");

student.lastLogin = new Date().toLocaleString("uk-UA");
updateStudentInStorage(student);
openStudentWindow();
}

function loginTeacher() {
const login = document.getElementById("teacherLoginName").value.trim();
const password = document.getElementById("teacherLoginPassword").value.trim();

if (login === TEACHER_LOGIN && password === TEACHER_PASSWORD) {
setTeacherLoggedIn(true);
openTeacherWindow();
} else {
alert("Невірний логін");
}
}

/* ----- STUDENT ACTIONS ----- */

function completeModule(n) {
const student = getCurrentStudent();
if (!student) return alert("Увійдіть");

if (!student.modules[n]) {
student.modules[n] = true;
student.points += 25;
applyStudentGamification(student);
updateStudentInStorage(student);
renderStudentCabinet();
}
}

/* ----- TEST ----- */

function calculateTest() {
const student = getCurrentStudent();
if (!student) return;

const form = document.getElementById("psychTest");

```

```

const data = new FormData(form);
let total = 0;

for (let i = 1; i <= 4; i++) {
  const v = data.get("q" + i);
  if (!v) return alert("Заповніть все");
  total += +v;
}

if (total >= 10) {
  student.points += 20;
  student.testText = "Високий рівень";
} else if (total >= 7) {
  student.points += 15;
  student.testText = "Середній рівень";
} else {
  student.points += 10;
  student.testText = "Низький рівень";
}

student.testsPassed++;
student.testScore = total;

applyStudentGamification(student);
updateStudentInStorage(student);
renderStudentCabinet();
}

/* ----- QUIZ ----- */

function setupQuizListener() {
  document.querySelectorAll("#quiz1 .quiz-option").forEach(opt => {
    opt.onclick = () => {
      const student = getCurrentStudent();
      if (!student) return;

      const result = document.getElementById("quiz1Result");

      if (opt.dataset.correct === "true") {
        if (!student.quizPassed) student.points += 10;
        student.quizPassed = true;
        student.quizResult = "OK";
      } else {
        student.quizResult = "Помилка";
      }

      applyStudentGamification(student);
      updateStudentInStorage(student);
      renderStudentCabinet();
    };
  });
}

/* ----- STUDENT UI ----- */

function renderStudentCabinet() {
  const student = getCurrentStudent();
  if (!student) return;

  applyStudentGamification(student);

  setText("profilePoints", student.points);
}

```

```

    setText("profileLevel", student.level);
    setText("profileModules", Object.values(student.modules).filter(Boolean).length + "/4");

    for (let i = 1; i <= 4; i++) {
        const el = document.getElementById("moduleStatus" + i);
        if (el) el.textContent = student.modules[i] ? "Done" : "No";
    }

    setText("resultPoints", student.points);
    setText("resultLevel", student.level);
}

/* ----- TEACHER ----- */

function renderTeacherList() {
    const list = document.getElementById("teacherStudentList");
    if (!list) return;

    const students = getStudents();
    list.innerHTML = "";

    students.forEach(s => {
        const div = document.createElement("div");
        div.textContent = `${s.name} | ${s.points}`;
        list.appendChild(div);
    });
}

function showTeacherStudentDetails() {}

/* ----- INIT ----- */

function initPage() {
    const view = getView();

    if (view === "student") {
        showPage("studentPage");
        setupQuizListener();
        renderStudentCabinet();
    } else if (view === "teacher") {
        showPage("teacherPage");
        renderTeacherList();
    } else {
        showPage("homePage");
    }
}

document.addEventListener("DOMContentLoaded", initPage);/* ===== REGISTRATION
===== */

function registerStudent() {
    const name = document.getElementById("regName").value.trim();
    const group = document.getElementById("regGroup").value.trim();
    const email = document.getElementById("regEmail").value.trim().toLowerCase();
    const password = document.getElementById("regPassword").value.trim();
    const msg = document.getElementById("registerSuccess");

    if (!name || !group || !email || !password) {
        if (msg) {
            msg.style.display = "block";
            msg.textContent = "Будь ласка, заповніть усі поля.";
        }
    }
}

```

```

    return;
}

const students = getStudents();

if (students.some(s => s.email === email)) {
    if (msg) {
        msg.style.display = "block";
        msg.textContent = "Студент з таким email уже зареєстрований.";
    }
    return;
}

const student = createEmptyStudent(name, group, email, password);

students.push(student);
saveStudents(students);
setCurrentStudentId(student.id);

if (msg) {
    msg.style.display = "block";
    msg.textContent = "Реєстрація успішна. Відкривається студент...";
}

openStudentWindow();
}

/* ===== LOGIN STUDENT ===== */

function loginStudent() {
    const email = document.getElementById("studentLoginEmail").value.trim().toLowerCase();
    const password = document.getElementById("studentLoginPassword").value.trim();
    const info = document.getElementById("studentLoginInfo");

    const students = getStudents();
    const student = students.find(s => s.email === email && s.password === password);

    if (!student) {
        if (info) {
            info.style.display = "block";
            info.textContent = "Невірний email або пароль.";
        }
        return;
    }

    student.lastLogin = new Date().toLocaleString("uk-UA");

    saveStudents(
        students.map(s => s.id === student.id ? student : s)
    );

    setCurrentStudentId(student.id);

    if (info) {
        info.style.display = "block";
        info.textContent = "Вхід успішний.";
    }

    openStudentWindow();
}

```

```

/* ===== LOGIN TEACHER ===== */

function loginTeacher() {
  const login = document.getElementById("teacherLoginName").value.trim();
  const password = document.getElementById("teacherLoginPassword").value.trim();
  const info = document.getElementById("teacherLoginInfo");

  if (login === TEACHER_LOGIN && password === TEACHER_PASSWORD) {
    setTeacherLoggedIn(true);

    if (info) {
      info.style.display = "block";
      info.textContent = "Вхід викладача успішний.";
    }

    openTeacherWindow();
  } else {
    if (info) {
      info.style.display = "block";
      info.textContent = "Невірний логін або пароль.";
    }
  }
}

/* ===== MODULE COMPLETE ===== */

function completeModule(moduleNumber) {
  const student = getCurrentStudent();
  if (!student) return alert("Увійдіть як студент");

  if (!student.modules?.[moduleNumber]) {
    student.modules[moduleNumber] = true;
    student.points = (student.points || 0) + 25;

    applyStudentGamification(student);
    updateStudentInStorage(student);
    renderStudentCabinet();

    alert(`Модуль ${moduleNumber} пройдено +25 балів`);
  } else {
    alert("Вже пройдено");
  }
}

/* ===== TEST ===== */

function calculateTest() {
  const student = getCurrentStudent();
  if (!student) return alert("Увійдіть як студент");

  const form = document.getElementById("psychTest");
  if (!form) return;

  const data = new FormData(form);
  let total = 0;

  for (let i = 1; i <= 4; i++) {
    const val = data.get("q" + i);

```

```

    if (!val) {
        return alert("Заповніть всі відповіді");
    }

    total += Number(val);
}

const result = document.getElementById("testResult");
if (result) result.classList.add("show");

let bonus = 0;

if (total >= 10) {
    bonus = 20;
    student.testText = "Високий рівень адаптації";
    result.innerHTML = "Високий рівень адаптації (+20)";
    result.className = "result show success";

} else if (total >= 7) {
    bonus = 15;
    student.testText = "Середній рівень адаптації";
    result.innerHTML = "Середній рівень адаптації (+15)";
    result.className = "result show success";

} else {
    bonus = 10;
    student.testText = "Потрібна підтримка";
    result.innerHTML = "Потрібна підтримка (+10)";
    result.className = "result show fail";
}

student.points = (student.points || 0) + bonus;
student.testsPassed = (student.testsPassed || 0) + 1;
student.testScore = total;

applyStudentGamification(student);
updateStudentInStorage(student);
renderStudentCabinet();
}/* ===== TEST RESULT ===== */

function calculateTest() {
    const student = getCurrentStudent();
    if (!student) return alert("Спочатку увійдіть як студент.");

    const form = document.getElementById("psychTest");
    if (!form) return;

    const data = new FormData(form);
    let total = 0;

    for (let i = 1; i <= 4; i++) {
        const val = data.get("q" + i);
        if (!val) return alert("Заповніть всі відповіді");
        total += Number(val);
    }

    const result = document.getElementById("testResult");
    if (result) result.classList.add("show");

```

```

let bonus = 0;

if (total >= 10) {
  bonus = 20;
  student.testText = "Високий рівень адаптації";
  if (result) {
    result.className = "result show success";
    result.innerHTML = "Високий рівень адаптації <b>(+20)</b>";
  }

} else if (total >= 7) {
  bonus = 15;
  student.testText = "Середній рівень адаптації";
  if (result) {
    result.className = "result show success";
    result.innerHTML = "Середній рівень адаптації <b>(+15)</b>";
  }

} else {
  bonus = 10;
  student.testText = "Потрібна підтримка";
  if (result) {
    result.className = "result show fail";
    result.innerHTML = "Потрібна підтримка <b>(+10)</b>";
  }
}

student.points = (student.points || 0) + bonus;
student.testsPassed = (student.testsPassed || 0) + 1;
student.testScore = total;

applyStudentGamification(student);
updateStudentInStorage(student);
renderStudentCabinet();
}

/* ===== QUIZ ===== */

function setupQuizListener() {
  document.querySelectorAll("#quiz1 .quiz-option").forEach(option => {
    option.onclick = function () {
      const student = getCurrentStudent();
      if (!student) return alert("Спочатку увійдіть як студент.");

      const all = document.querySelectorAll("#quiz1 .quiz-option");
      const result = document.getElementById("quiz1Result");

      all.forEach(el => el.classList.remove("correct", "wrong"));

      if (this.dataset.correct === "true") {
        this.classList.add("correct");

        if (!student.quizPassed) {
          student.points = (student.points || 0) + 10;
        }

        student.quizPassed = true;
        student.quizResult = "Квіз пройдено успішно";

        if (result) {
          result.className = "result show success";
          result.textContent = "Правильно! +10 балів.";
        }
      }
    }
  });
}

```

```

    }

    } else {
      this.classList.add("wrong");

      student.quizResult = "Квіз виконано з помилкою";

      if (result) {
        result.className = "result show fail";
        result.textContent = "Неправильно. Спробуйте ще раз.";
      }
    }

    applyStudentGamification(student);
    updateStudentInStorage(student);
    renderStudentCabinet();
  });
}

/* ===== STUDENT DASHBOARD ===== */

function renderStudentCabinet() {
  const student = getCurrentStudent();
  if (!student) return;

  applyStudentGamification(student);
  updateStudentInStorage(student);

  const completed = Object.values(student.modules || {}).filter(Boolean).length;

  setText("sidebarStudentName", student.name);
  setText("sidebarStudentGroup", student.group);
  setText("sidebarStudentEmail", student.email);

  setText("profileModules", `${completed}/4`);
  setText("profilePoints", student.points || 0);
  setText("profileTests", student.testsPassed || 0);
  setText("profileLevel", student.level);

  const progressBar = document.getElementById("studentProgressBar");
  if (progressBar) progressBar.style.width = (student.progress || 0) + "%";

  setText("studentProgressText", (student.progress || 0) + "% пройдено");

  const b1 = document.getElementById("sBadge1");
  const b2 = document.getElementById("sBadge2");
  const b3 = document.getElementById("sBadge3");

  if (b1) b1.classList.toggle("active", student.points >= 50);
  if (b2) b2.classList.toggle("active", student.points >= 100);
  if (b3) b3.classList.toggle("active", student.points >= 150);

  const l1 = document.getElementById("level1");
  const l2 = document.getElementById("level2");
  const l3 = document.getElementById("level3");

  if (l1) l1.classList.toggle("active", student.points >= 50 && student.points < 100);
  if (l2) l2.classList.toggle("active", student.points >= 100 && student.points < 150);

```

```

if (l3) l3.classList.toggle("active", student.points >= 150);

for (let i = 1; i <= 4; i++) {
  const status = document.getElementById("moduleStatus" + i);
  const bar = document.getElementById("bar" + i);

  const done = student.modules?.[i];

  if (status) {
    status.textContent = done ? "Пройдено" : "Не пройдено";
    status.classList.toggle("done", done);
  }

  if (bar) {
    bar.style.height = done ? "100%" : "20%";
    bar.textContent = done ? "100%" : "0%";
  }
}

setText("resultModules", `${completed}/4`);
setText("resultProgress", (student.progress || 0) + "%");
setText("resultPoints", student.points || 0);
setText("resultLevel", student.level);
setText("resultTestText", student.testText);
/* ===== UI TEXT SET ===== */

function setText(id, value) {
  const el = document.getElementById(id);
  if (el) el.textContent = value;
}

/* ===== STUDENT DATA BLOCK ===== */

function renderStudentData(student) {
  if (!student) return;

  setText("resultQuizText", student.quizResult);

  setText("dataName", student.name);
  setText("dataGroup", student.group);
  setText("dataEmail", student.email);
  setText("dataPoints", student.points || 0);
  setText("dataLevel", student.level);
  setText("dataBadge", student.badge);
  setText("dataTestResult", student.testText);

  setText("certificateName", student.name);
  setText("certificateProgress", (student.progress || 0) + "%");
  setText("certificatePoints", student.points || 0);
  setText("certificateDate", new Date().toLocaleDateString("uk-UA"));
}

/* ===== TEACHER LIST ===== */

function renderTeacherList() {
  const list = document.getElementById("teacherStudentList");
  if (!list) return;

  const students = getStudents();
  list.innerHTML = "";
}

```

```

if (!students.length) {
  list.innerHTML = '<p class="muted">Поки що немає зареєстрованих студентів.</p>';
  return;
}

students.forEach(student => {
  const div = document.createElement("div");
  div.className = "student-item";

  div.innerHTML = `
    <h4>${student.name}</h4>
    <p>Група: ${student.group}</p>
    <p>Прогрес: ${student.progress || 0}% | Бали: ${student.points || 0}</p>
  `;

  div.onclick = () => showTeacherStudentDetails(student.id, div);

  list.appendChild(div);
});
}

/* ===== TEACHER DETAILS ===== */

function showTeacherStudentDetails(studentId, clickedEl) {
  document.querySelectorAll(".student-item")
    .forEach(el => el.classList.remove("active"));

  if (clickedEl) clickedEl.classList.add("active");

  const students = getStudents();
  const student = students.find(s => String(s.id) === String(studentId));

  const details = document.getElementById("teacherStudentDetails");
  if (!details) return;

  if (!student) {
    details.innerHTML = '<p class="muted">Студента не знайдено.</p>';
    return;
  }

  const completed = Object.values(student.modules || {}).filter(Boolean).length;

  details.innerHTML = `
    <table class="teacher-table">
      <tr><th>Параметр</th><th>Значення</th></tr>

      <tr><td>Ім'я</td><td>${student.name}</td></tr>
      <tr><td>Група</td><td>${student.group}</td></tr>
      <tr><td>Email</td><td>${student.email}</td></tr>

      <tr><td>Пройдено модулів</td><td>${completed}/4</td></tr>
      <tr><td>Прогрес</td><td>${student.progress || 0}%</td></tr>
      <tr><td>Бали</td><td>${student.points || 0}</td></tr>

      <tr><td>Рівень</td><td>${student.level || "Початківець"}</td></tr>
      <tr><td>Бейдж</td><td>${student.badge || ""}</td></tr>

      <tr><td>Квіз</td><td>${student.quizResult || "Не пройдено"}</td></tr>
  `;
}

```

```

        <tr><td>Тест</td><td>${student.testText || "Не пройдено"}</td></tr>

        <tr><td>Бал тесту</td><td>${student.testScore || 0}</td></tr>
        <tr><td>Спроби тесту</td><td>${student.testsPassed || 0}</td></tr>
        <tr><td>Останній вхід</td><td>${student.lastLogin || "-"}</td></tr>
    </table>
    `;
}

/* ===== PDF ===== */

function downloadCertificatePDF() {
    switchStudentTab("studentTabCertificate");
    setTimeout(() => window.print(), 200);
}

/* ===== LOGOUT ===== */

function logoutStudent() {
    clearCurrentStudentId();
    window.location.href = window.location.pathname;
}

function logoutTeacher() {
    setTeacherLoggedIn(false);
    window.location.href = window.location.pathname;
}

/* ===== INIT ===== */

function initPage() {
    const view = getView();

    if (view === "student") {
        const student = getCurrentStudent();

        if (!student) {
            document.body.innerHTML = `
                <section style="padding:80px 0;">
                    <div class="container">
                        <div class="panel center">
                            <h2>Студент не авторизований</h2>
                            <p class="muted">Увійдіть або зареєструйтесь.</p>
                            <a class="btn btn-primary" href="${window.location.pathname}">
                                Назад
                            </a>
                        </div>
                    </div>
                </section>
            `;
            return;
        }

        showPage("studentPage");
        renderStudentCabinet();
        setupQuizListener();
        renderStudentData(student);
        return;
    }

    if (view === "teacher") {
        if (!isTeacherLoggedIn()) {

```

```

        document.body.innerHTML = `
            <section style="padding:80px 0;">
                <div class="container">
                    <div class="panel center">
                        <h2>Викладач не авторизований</h2>
                        <a class="btn btn-primary" href="{window.location.pathname}">
                            Назад
                        </a>
                    </div>
                </div>
            </section>
        `;
        return;
    }

    showPage("teacherPage");
    renderTeacherList();
    return;
}

showPage("homePage");
}

document.addEventListener("DOMContentLoaded", initPage);function initPage() {
    const view = getView();

    const homePage = document.getElementById("homePage");
    const studentPage = document.getElementById("studentPage");
    const teacherPage = document.getElementById("teacherPage");

    function hideAll() {
        if (homePage) homePage.classList.add("hidden");
        if (studentPage) studentPage.classList.add("hidden");
        if (teacherPage) teacherPage.classList.add("hidden");
    }

    /* ===== STUDENT VIEW ===== */
    if (view === "student") {
        const student = getCurrentStudent();

        if (!student) {
            document.body.innerHTML = `
                <section style="padding:80px 0;">
                    <div class="container">
                        <div class="panel center">
                            <h2>Студент не авторизований</h2>
                            <p class="muted">Спочатку увійдіть або зареєструйтесь.</p>
                            <a class="btn btn-primary" href="{window.location.pathname}">
                                Назад
                            </a>
                        </div>
                    </div>
                </section>
            `;
            return;
        }

        hideAll();
        if (studentPage) studentPage.classList.remove("hidden");

        setupQuizListener();
        renderStudentCabinet();

        return;
    }
}

```

```

}

/* ===== TEACHER VIEW ===== */
if (view === "teacher") {
  if (!isTeacherLoggedIn()) {
    document.body.innerHTML = `
      <section style="padding:80px 0;">
        <div class="container">
          <div class="panel center">
            <h2>Викладач не авторизований</h2>
            <p class="muted">Увійдіть як викладач на головній сторінці.</p>
            <a class="btn btn-primary" href="{window.location.pathname}">
              Назад
            </a>
          </div>
        </div>
      </section>
    `;
    return;
  }

  hideAll();
  if (teacherPage) teacherPage.classList.remove("hidden");

  renderTeacherList();

  return;
}

/* ===== HOME VIEW ===== */
hideAll();
if (homePage) homePage.classList.remove("hidden");
}

/* ===== INIT ===== */
document.addEventListener("DOMContentLoaded", initPage);

```

## ДОДАТОК Г. ТЕСТОВІ МАТЕРІАЛИ

Тестові матеріали використовуються для перевірки рівня засвоєння студентом навчального матеріалу після проходження тренінгу. Приклади тестових завдань наведено нижче.

Таблиця Г.1 - Приклади тестових матеріалів

№	Питання	Правильна відповідь
1	Що таке академічна доброчесність?	Сукупність етичних принципів
2	Який модуль відповідає за збереження даних?	LocalStorage
3	Яка технологія використовується для структури вебсторінки?	HTML
4	Яка технологія використовується для оформлення інтерфейсу?	CSS
5	Яка технологія реалізує логіку тестування?	JavaScript

## ДОДАТКОВІ МАТЕРІАЛИ ДО ОСНОВНИХ РОЗДІЛІВ

### Порівняльний аналіз та вибір програмного рішення

Детальний аналіз Moodle, Google Classroom, Coursera, Prometheus та LearningApps показує, що кожна система має власні переваги та обмеження. Moodle забезпечує широкий функціонал для управління навчанням, однак потребує адміністрування. Google Classroom відзначається простотою використання та інтеграцією з сервісами Google. Coursera та Prometheus надають доступ до якісного освітнього контенту та сертифікації. LearningApps орієнтована на створення інтерактивних вправ. Розроблена система відрізняється спеціалізацією на адаптації студентів першого курсу та простотою впровадження.

### Особливості реалізації програмного забезпечення

Архітектура системи побудована за принципом розподілу функцій між інтерфейсом, бізнес-логікою та рівнем збереження даних. Для реалізації використано HTML5, CSS3, JavaScript та LocalStorage. Описано алгоритми реєстрації, авторизації, проходження модулів, тестування, нарахування балів та формування сертифікатів. Такий підхід забезпечує масштабованість і спрощує подальший супровід програмного продукту.

### Додаткові функціональні можливості системи

Інтерфейс користувача реалізовано відповідно до сучасних принципів UX/UI. Передбачено окремі кабінети студента та викладача, систему модулів, тестування, перегляд результатів, бейджів та сертифікатів. Кожен екран супроводжується

логічною навігацією та адаптований для використання на різних пристроях.

#### Узагальнення результатів тестування

Тестування включало функціональні та нефункціональні перевірки. Було підтверджено коректність роботи реєстрації, навчальних модулів, тестів, підрахунку результатів і збереження даних. Аналіз показав високий рівень функціональної, технічної та освітньої ефективності програмного забезпечення.

#### Рекомендації щодо впровадження та розвитку системи

Подальший розвиток системи може передбачати використання серверної бази даних, інтеграцію з Moodle та Google Classroom, створення мобільної версії, впровадження аналітики успішності та елементів штучного інтелекту для персоналізації навчання.

Обґрунтування необхідності створення власного програмного забезпечення

## ІНФОРМАЦІЙНИЙ ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

У розділі виконано детальний аналіз Moodle, Google Classroom, Microsoft Teams,

Coursera, Prometheus та LearningApps. Проведено порівняльний аналіз їх функціональних

можливостей, переваг та недоліків. Встановлено, що існуючі системи орієнтовані переважно

на дистанційне навчання і не забезпечують повною мірою адаптацію студентів першого курсу.

Обґрунтовано необхідність створення спеціалізованого тренінгу «Коледж: Старт успіху»,

що поєднує навчальні матеріали, тестування, контроль результатів та панель викладача.

Також наведено аналіз HTML5, CSS3 та JavaScript як основних технологій реалізації проєкту.

Особливості реалізації архітектури системи

**3.1 Проєктування архітектури програмного забезпечення.** Архітектура системи побудована за модульним принципом та включає інтерфейс користувача, модуль навчальних матеріалів, модуль тестування, модуль статистики та LocalStorage.

**3.2 Обґрунтування вибору технологій.** Проведено аналіз React, Angular, Vue.js та класичного підходу HTML5/CSS3/JavaScript. Для проєкту обрано HTML5, CSS3 та JavaScript завдяки простоті, доступності та підтримці сучасними браузерями.

**3.3 Розробка алгоритмів роботи системи.** Описано алгоритми реєстрації, авторизації, проходження навчальних модулів, тестування, підрахунку результатів та формування сертифікатів.

**3.4 UML-моделювання системи.** Використано діаграми варіантів використання, активності, послідовності та компонентів для опису структури та поведінки програмного забезпечення.

Висновки до розділу 3. Отримані результати стали основою для практичної реалізації та тестування навчального програмного забезпечення.

Додаткові функціональні можливості програмного забезпечення

Практична частина містить детальний опис реалізації інтерфейсу користувача, навчальних модулів, системи тестування, особистого кабінету студента, панелі викладача та механізму збереження даних за допомогою LocalStorage.

Реалізовано головну сторінку, сторінки реєстрації та авторизації, меню студента, профіль користувача, систему досягнень, сертифікацію результатів та статистичні модулі.

Окрему увагу приділено адаптивності інтерфейсу та зручності використання.

Панель викладача забезпечує контроль успішності студентів, перегляд результатів та аналіз статистичних показників. Для збереження інформації використовується LocalStorage, що дозволяє працювати без серверної інфраструктури.

## **ЗАГАЛЬНІ ВИСНОВКИ**

У кваліфікаційній роботі було розроблено навчальне програмне забезпечення тренінгу «Коледж: Старт успіху». Проведено аналіз предметної області, досліджено сучасні освітні платформи, спроектовано архітектуру системи, реалізовано програмний продукт та виконано його тестування. Отримані результати підтвердили ефективність розробленого рішення для адаптації студентів до освітнього процесу.

## **РЕКОМЕНДАЦІЇ ЩОДО ВПРОВАДЖЕННЯ ТА ПОДАЛЬШОГО РОЗВИТКУ СИСТЕМИ**

Подальший розвиток системи може включати інтеграцію серверної бази даних, створення мобільної версії, інтеграцію з Moodle та Google Classroom, розширення панелі викладача, використання технологій штучного інтелекту та впровадження мультимедійних навчальних матеріалів.