

Вінницький Національний Технічний Університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Бакалаврська кваліфікаційна робота

на тему: «Розробка інтерактивного вебзастосунку для підготовки до
національного мультипредметного тесту»

Виконала: студентка IV курсу, групи 2ПІ-216
спеціальності

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

Олійник Яна Сергіївна
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. ПЗ Рейда О.М.
(прізвище та ініціали)

Рецензент: к.т.н., доцент каф. ОТ Томчук М.А
(прізвище та ініціали)

Допущено до захисту

Зав. кафедри ПЗ О.Н. Романюк

«09» 06 2025 р.

ВНТУ – 2025

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

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
О.Н. Романюк
«20» березня 2025 р.

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

Олійник Яні Сергіївни

1. Тема роботи – «Розробка інтерактивного вебзастосунку для підготовки до національного мультипредметного тесту».

Керівник роботи: Рейда Олександр Миколайович, к.т.н., доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від «20» березня 2025 р. №97.

2. Строк подання студентом роботи 2 червня 2025.

3. Вихідні дані до роботи: Опис структури та формату НМТ; методики створення тестових систем; специфіка адаптивного навчання; основні типи тестових завдань; функціональні вимоги до навчальних вебзастосунків; сучасні вебтехнології (JavaScript, Vue.js, PHP, Laravel, PostgreSQL); приклади реалізації дашбордів, інтеграція з API мовних моделей для генерації пояснень; вимоги до швидкості користування, адаптивності та безпеки освітніх платформ.

4. Зміст текстової частини: вебзастосунок; вхідні дані: різної складності завдання для тестування, персональні дані користувача, вихідні дані: аналітика особистого прогресу користувача; мова програмування PHP.

5. Схема архітектури вебзастосунку; макети інтерфейсу користувача; графіки схеми алгоритмів тестування та повторення; структура бази даних; графіки аналітики навчального прогресу; приклади пояснень до помилок; діаграми маршруту користувача; демонстраційні скріншоти.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Рейда О.М., к.т.н., доцент каф. ІІЗ	24.03.25	01.06.25

7. Дата видачі завдання 24 березня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи	Приміт
1	Аналіз предметної області, дослідження технологій підготовки до НМТ	25.03.25 - 05.04.25	Вик.
2	Формування вимог до системи, обґрунтування вибору технологій	06.04.25 - 18.04.25	Вик.
3	Проектування архітектури вебзастосунку та структури бази даних	19.04.25 - 21.04.25	Вик.
4	Розробка програмного забезпечення	22.04.25 - 17.05.25	Вик.
5	Тестування програмного забезпечення	18.05.25 - 25.05.25	Вик.
6	Оформлення матеріалів до захисту БДР	26.05.25 - 30.05.25	Вик.

Студентка

Керівник бакалаврської кваліфікаційної роботи

(підпис)

(підпис)

Я. С. Олійник
(ініціали та прізвище)

О. М. Рейда
(ініціали та прізвище)

АНОТАЦІЯ

УДК 00492

Олійник Я. С. Розробка інтерактивного вебзастосунку для підготовки до національного мультипредметного тесту: бакалаврська кваліфікаційна робота зі спеціальності 121 – інженерія програмного забезпечення, освітня програма – інженерія програмного забезпечення. Вінниця: ВНТУ, 2025. 66 с.

На укр. мові. Бібліогр. : 25 назв ; рис.: 37; табл. : 5.

У бакалаврській кваліфікаційній роботі було методи та програмні засоби для інтерактивної підготовки до національного мультипредметного тесту (НМТ). Проведено аналіз сучасних онлайн-рішень для підготовки до ЗНО/НМТ, виявлено недоліки існуючих платформ та обґрунтовано необхідність створення більш адаптивного та зручного сервісу.

Розроблено архітектуру клієнт-серверного застосунку з інтерактивним інтерфейсом, реалізовано систему генерації тестів, автоматичної перевірки результатів, збереження статистики проходження тестів та візуалізації прогресу користувача. Інтерфейс створено з урахуванням принципів UX/UI для забезпечення зручності користування.

Проведено тестування застосунку, яке підтвердило коректність реалізованого функціоналу та його придатність до використання в освітньому процесі. Розроблене програмне забезпечення може бути використано в навчальних закладах, репетиторських платформах або для самостійної підготовки учнів.

Ключові слова: вебзастосунок, НМТ, тестування, підготовка до іспитів, інтерфейс користувача, клієнт-серверна архітектура.

ABSTRACT

UDC 00492

Oliinyk Ya. S. Development of an Interactive Web Application for Preparation for the National Multi-Subject Test : bachelor's thesis in specialty 121 – Software Engineering, educational program – Software Engineering. Vinnytsia: VNTU, 2025. 66 p.

In Ukrainian. Bibliography.: 25 titles; Figures.: 37; Tables: 5.

This bachelor's thesis presents the development of methods and software tools for interactive preparation for the National Multi-Subject Test (NMT). The work includes an analysis of existing online platforms for ZNO/NMT preparation, identifies their limitations, and substantiates the need for a more adaptive and user-friendly service.

A client-server application architecture was designed with an interactive interface. The system implements test generation, automatic result evaluation, tracking of test performance statistics, and visualization of user progress. The interface was developed according to UX/UI design principles to ensure ease of use.

Application testing confirmed the correctness of the implemented functionality and its suitability for use in educational contexts. The developed software can be used by schools, tutoring platforms, or students for independent test preparation.

Keywords: web application, NMT, testing, exam preparation, user interface, client-server architecture.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ РОЗВИТКУ ТЕХНОЛОГІЙ ПІДГОТОВКИ ДО ЕКЗАМЕНАЦІЙНИХ ТЕСТУВАНЬ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ.....	7
1.1 Аналіз стану технологій підготовки до НМТ з використанням інтерактивних вебзастосунків.....	7
1.2 Порівняльний аналіз аналогів.....	9
1.3 Аналіз методів розв’язання задачі.....	15
1.4 Постановка задачі дослідження.....	18
1.5 Висновки.....	19
2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ....	20
2.1 Формалізація вхідних даних та постановка функціональних вимог.....	20
2.2 Розробка інтерфейсу програмного додатку.....	22
2.3 Розробка структур програмного продукту.....	28
2.4 Розробка алгоритмів роботи системи.....	30
2.5 Висновки.....	34
3 РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ ПЛАТФОРМИ ПІДГОТОВКИ ДО НМТ.....	35
3.1 Варіантний аналіз і обґрунтування вибору засобів ля реалізації програмного забезпечення.....	35
3.2 Розробка модуля аналітики результатів тестування.....	38
3.3 Розробка модуля керування курсами й прогресом користувача.....	40
3.4 Розробка модуля інтегрування штучного інтелекту.....	43
3.5 Висновки.....	47
4 ТЕСТУВАННЯ ПРОГРАМИ.....	48
4.1 Методи тестування програми.....	48
4.2 Результати тестування програмного забезпечення.....	49
4.3 Розробка інструкції користувача.....	51
4.4 Висновки.....	58

	3
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61
ДОДАТКИ.....	64
Додаток А – ТЕХНІЧНЕ ЗАВДАННЯ.....	65
Додаток Б – ПРОТОКОЛ ПЕРЕВІРКИ НА ПЛАГІАТ.....	69
Додаток В – ЛІСТИНГ ПРОГРАМИ.....	70
Додаток Г – ІЛЮСТРАТИВНИЙ МАТЕРІАЛ.....	88

ВСТУП

Обґрунтування вибору теми дослідження. Розвиток цифрових технологій суттєво трансформує освітнє середовище. Сучасні реформи в системі української освіти, зокрема трансформація зовнішнього незалежного оцінювання у формат національного мультипредметного тесту (НМТ), обумовили необхідність створення інноваційних цифрових інструментів для якісної підготовки школярів. НМТ став основною формою вступного випробування для українських абітурієнтів, що охоплює кілька предметів водночас і проводиться у цифровому форматі.

Традиційні підходи до підготовки, зокрема друковані збірники тестів або неструктуровані онлайн-матеріали, не забезпечують достатнього рівня адаптації до індивідуальних потреб користувача. Крім того, відсутність інтерактивності, автоматичного зворотного зв'язку та системи оцінки прогресу знижує мотивацію до навчання. Тому зростає потреба у створенні таких інструментів, які б відповідали принципам сучасного навчання: адаптивність, зручність та легкість використання, візуальна привабливість, статистична обґрунтованість аналізу результатів для відслідковування прогресу.

НМТ, як комп'ютеризований формат оцінювання, суттєво відрізняється від традиційного ЗНО: він поєднує декілька предметів в одному тесті, має обмежений час виконання та проводиться онлайн в умовах спеціально підготовлених центрів тестування. Відповідно, підготовка до такого типу іспиту вимагає нових інструментів.

У такому контексті розробка інтерактивного вебзастосунку для підготовки до НМТ є актуальним і необхідним завданням. Такий вебзастосунок повинен не лише моделювати формат тестування, але й містити функціональність персонального кабінету, систему оцінювання результатів, рекомендації щодо тем, які потребують додаткового вивчення, а також надавати зручний і зрозумілий інтерфейс для користувачів різного рівня підготовки, що особливо важливо в умовах війни, пандемії та інших соціальних викликів.

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

Мета та завдання дослідження. Метою кваліфікаційної роботи є підвищення рівня знань та ефективності самостійної підготовки абітурієнтів шляхом створення інтерактивного навчального середовища з адаптивною логікою тестування, візуалізацією результатів та персоналізованими рекомендаціями.

Відповідно до поставленої мети необхідно вирішити такі завдання:

- провести аналіз існуючих програмних засобів, їхні функціональні особливості та обмеження;
- дослідити особливості функціонування та побудови інтерактивного вебзастосунку, орієнтованого на абітурієнтів;
- модифікувати метод аналізу помилок у відповідях користувача, що дозволяє автоматично формувати пояснення до неправильних відповідей із урахуванням запиту користувача та контексту завдання для покращення персоналізації та ефективності навчального процесу;
- розробити архітектуру системи з урахуванням вимог до масштабованості, доступності та безпеки;
- реалізувати адаптивний механізму формування тестів на основі результатів користувача, персоналізованих пояснень та модулі візуального аналізу навчального прогресу;
- провести тестування роботи вебзастосунку.

Об'єкт дослідження – процес розробки інтерактивного вебзастосунку для підготовки учнів до національного мультипредметного тесту.

Предмет дослідження – методи та засоби розробки інтерактивних вебсистем для навчальних цілей, зокрема для підготовки до стандартизованих іспитів.

Методи дослідження. Під час розробки інтерактивного вебзастосунку для підготовки до НМТ використовувались такі методи: методи об'єктно-орієнтованого моделювання, методи структурного та модульного програмування, методи

візуалізації даних, методи статистичного аналізу, а також методи штучного інтелекту, зокрема використання мовної моделі ChatGPT, функціональний аналіз, комп'ютерне моделювання для аналізу та перевірки отриманих теоретичних положень.

Наукова новизна отриманих результатів. У роботі модифіковано метод аналізу помилок у відповідях користувача, який на відміну від існуючих, базується на інтеграції NLP моделей за допомогою API ChatGPT, що дозволяє автоматично формувати пояснення до неправильних відповідей із урахуванням запиту користувача та контексту завдання для покращення персоналізації та ефективності навчального процесу.

Практичне значення отриманих результатів полягає в тому, що на основі виконаного дослідження та інтеграції штучного інтелекту було розроблено діючий прототип вебзастосунку для підготовки до НМТ. Запропоноване рішення може бути використано в освітніх закладах, на онлайн-платформах або самостійно – як ефективний інструмент для персоналізованої підготовки до національного мультипредметного тесту.

Апробація результатів роботи. Ключові ідеї бакалаврської кваліфікаційної роботи були представлені на LIV Всеукраїнській науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії у 2025 році [1].

Публікації. За темою бакалаврської роботи надруковано 1 працю у матеріалах конференції ВНТУ.

Структура та обсяг БКР. Бакалаврська кваліфікаційна робота включає вступ, чотири основні розділи, висновки, перелік використаних джерел, а також додатки. У додатках наведено технічне завдання, звіт про перевірку на плагіат, програмний код та візуальні матеріали для презентації результатів під час захисту.

1 АНАЛІЗ РОЗВИТКУ ТЕХНОЛОГІЙ ПІДГОТОВКИ ДО ЕКЗАМЕНАЦІЙНИХ ТЕСТУВАНЬ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз стану технологій підготовки до НМТ з використанням інтерактивних вебзастосунків

У зв'язку з воєнним станом в Україні та складнощами, пов'язаними з проведенням зовнішнього незалежного оцінювання, у 2022 році Міністерством освіти і науки було впроваджено новий формат вступного іспиту – національний мультипредметний тест. Його особливістю є те, що він охоплює одразу кілька предметів (українська мова, математика, історія України), проводиться в комп'ютеризованій формі й має обмеження в часі. Такий формат суттєво змінив підхід до підготовки абітурієнтів і актуалізував потребу в нових інструментах навчання, які були б наближеними до реальних умов складання НМТ [2].

На сьогодні переважна більшість учнів продовжують готуватись до вступного випробування за допомогою традиційних засобів: друкованих збірників тестів, конспектів, PDF-файлів або статичних сайтів, які надають можливість пройти обмежену кількість тестів без фіксації результатів. У таких підходах відсутня адаптація до рівня знань учня, не передбачено автоматичного аналізу помилок, а сам процес підготовки зводиться до механічного вибору варіантів відповідей без належного зворотного зв'язку. Крім того, учень не має змоги відстежувати свій навчальний прогрес, повторно проходити складні теми або працювати з індивідуально підібраними завданнями.

Інтерактивні вебзастосунки для підготовки до тестувань мають низку суттєвих переваг у порівнянні з традиційними методами. Головною перевагою є можливість миттєвої перевірки відповідей, автоматичного формування результатів, надання підказок та коментарів, а також ведення статистики за різними темами та тестами. Це дозволяє учневі не лише фіксувати помилки, а й аналізувати власний рівень знань та коригувати підготовку.

Також інтерактивні системи можуть містити різні типи тестових завдань: як класичні з одним правильним варіантом, так і завдання на встановлення відповідностей, вибір кількох відповідей чи введення власного варіанту. Це дає змогу моделювати реальний формат НМТ та створювати повноцінні тренувальні сесії. Крім того, такі вебзастосунки можуть забезпечувати доступ до навчальних матеріалів, рекомендацій та тематичних тестів з можливістю формувати індивідуальні добірки завдань.

У межах цифрової трансформації освіти поступово з'являються інтерактивні вебзастосунки, які намагаються вирішити частину цих проблем. Серед ключових характеристик таких застосунків можна відзначити:

- підтримку роботи з різними типами тестових завдань (одна правильна відповідь, кілька правильних варіантів, встановлення відповідностей, введення власного варіанту тощо);
- миттєву перевірку результатів з автоматичним підрахунком балів;
- збереження історії проходження тестів та можливість переглядати попередні відповіді;
- базову статистику результатів без глибокої аналітики.

Однак аналіз доступних на ринку інструментів демонструє, що більшість із них:

- мають обмежену функціональність і не забезпечують повноцінної адаптації під конкретного користувача;
- не дозволяють отримати пояснення до зроблених помилок;
- не дають можливості автоматично повторювати проблемні теми;

Також важливо зауважити, що майже всі існуючі вебресурси не враховують потребу в індивідуалізованій підтримці, яка особливо актуальна для учнів із різним рівнем підготовки. У більшості випадків відсутні механізми формування тематичних добірок, системи рекомендацій або динамічної побудови траєкторії навчання на основі попередніх результатів.

Аналіз сучасного стану технологій у сфері підготовки до НМТ підтверджує необхідність переходу до більш функціональних та гнучких систем навчання, які

забезпечують адаптивність, інтерактивність і аналітику результатів. Створення власного інтерактивного вебзастосунку для підготовки до НМТ є актуальним завданням, оскільки існуючі сервіси часто обмежені у функціоналі, мають платний доступ або не відповідають сучасним вимогам швидкодії, адаптивності та зручності користування [3].

Звернення до досвіду розробників сучасного програмного забезпечення, використання популярних фреймворків та технологій дозволяє створювати інноваційні рішення, які підвищують ефективність навчання, покращують користувацький досвід та забезпечують конкурентні переваги на ринку освітніх ІТ-продуктів. З огляду на постійний розвиток цифрових технологій, важливо стежити за актуальними тенденціями у цій галузі для успішної адаптації та модернізації освітніх сервісів.

1.2 Порівняльний аналіз аналогів

Національний мультипредметний тест, який було запроваджено як альтернативу ЗНО в умовах повномасштабної війни, кардинально змінив підходи до оцінювання знань випускників шкіл. На відміну від ЗНО, НМТ проводиться у форматі комп'ютерного тестування, має сувору структуру, обмежений час виконання і вимагає високої готовності до роботи з цифровими інтерфейсами. Ці особливості зумовили зростання попиту на інструменти для самостійної підготовки, які не тільки відображають формат реального тесту, а й дають змогу виявляти слабкі місця, отримувати підказки, відстежувати прогрес і повторювати теми.

У таких умовах на перший план вийшли інтерактивні вебзастосунки, які стали найбільш зручним та доступним способом організації підготовки для широкого кола абітурієнтів. З огляду на швидкий розвиток цифрових технологій, особливо в сфері веброзробки, з'явилися технічні можливості створювати адаптивні, функціонально гнучкі та інтуїтивно зрозумілі системи підготовки до тестувань. Такі застосунки працюють у браузері, не потребують встановлення,

підтримують роботу з мобільних пристроїв і можуть масштабуватись без втрати продуктивності.

Однак попри наявність різноманітних освітніх платформ, які частково реалізують подібний функціонал, глибший аналіз показує, що більшість з них або спеціалізуються винятково на теоретичному матеріалі, або не мають достатньо гнучких і масштабованих механізмів для роботи з тестами. Часто платформи обмежуються шаблонними варіантами завдань без можливості адаптації під рівень знань конкретного учня чи тематику курсу. Більше того, серйозним недоліком є відсутність повноцінного аналітичного інструменту, який би дозволяв учню не просто бачити суху оцінку, а й отримувати розгорнутий зворотний зв'язок: які саме теми опановані недостатньо глибоко, які типи завдань викликають найбільші труднощі, які помилки повторюються найчастіше та як динаміка змінюється з часом. Без такої діагностики процес навчання втрачає персоналізацію та стає менш ефективним для учня.

Для порівняння розглянемо функціонал низки популярних освітніх платформ:

- Zno.ua;
- На Урок;
- Prometheus;
- iLearn;
- EdEra;
- Всеосвіта.

Одним із прикладів є платформа «Zno.ua» (рис. 1.1) – онлайн-ресурс для підготовки до ЗНО та НМТ, що надає доступ до тестів за офіційною структурою. Користувач може проходити тести, переглядати правильні відповіді та отримувати загальну оцінку. Однак функціонал цієї платформи обмежений тестуванням без інтерактивних підказок, тематичних аналітик чи адаптивних тестів [4].

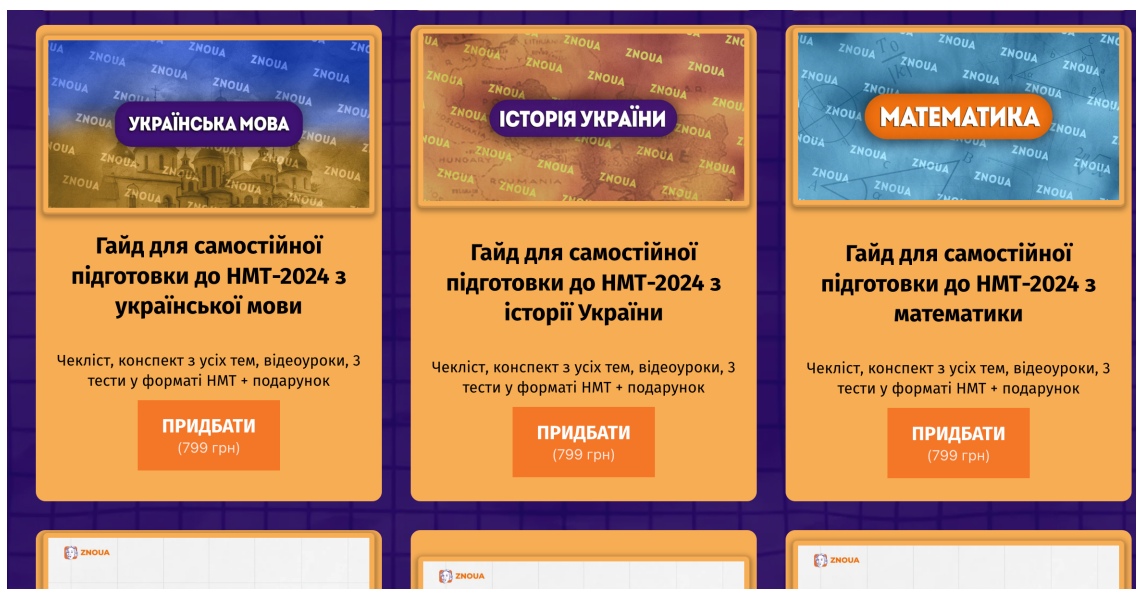


Рисунок 1.1 – Інтерфейс платформи «Zno.ua»

Ще одним популярним рішенням є «На Урок» (рис. 1.2) – освітня платформа, що містить велику кількість тестів та навчальних матеріалів. Вона дозволяє створювати власні тести та проходити тематичні добірки. Проте платформа не підтримує повноцінної аналітики прогресу, адаптивного підбору завдань або автоматичного формування тренувальних сесій відповідно до рівня знань користувача [5].

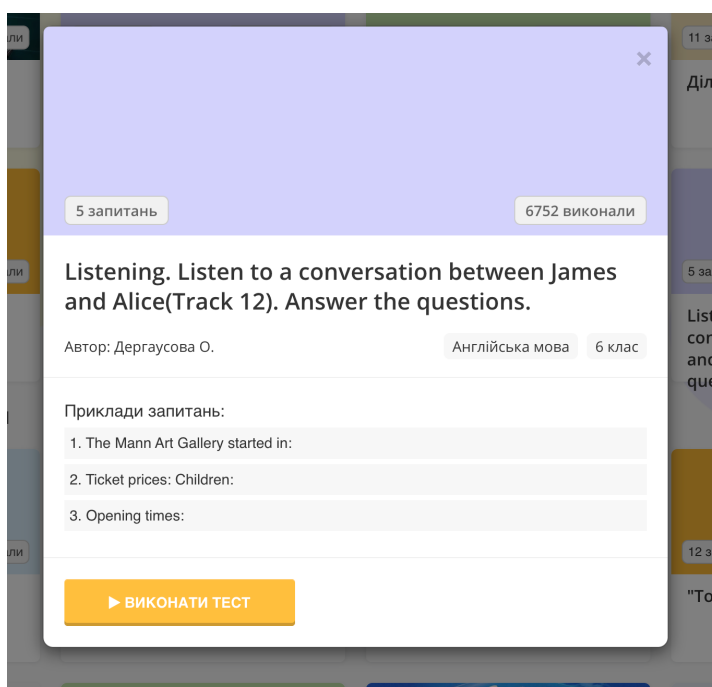


Рисунок 1.2 – Інтерфейс платформи «На Урок»

«Prometheus» (рис. 1.3) є онлайн-курсовою платформою з інтегрованими тестами у курсах. Основний акцент зроблено на теоретичні модулі, а тести слугують для перевірки засвоєння матеріалу. Платформа не надає можливості проходити повноцінні тренувальні НМТ-тести у форматі реального іспиту [6].

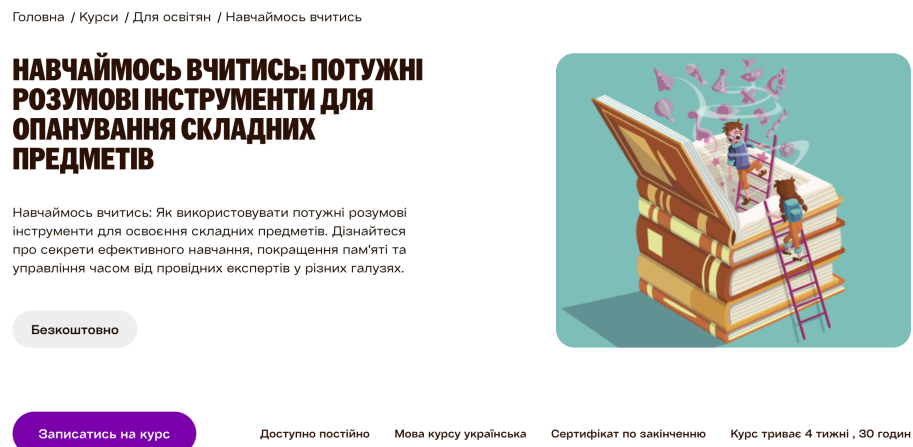


Рисунок 1.3 – Інтерфейс платформи «Prometheus»

Інтерактивна платформа «iLearn» (рис. 1.4) надає доступ до онлайн-курсів і тестів з можливістю перегляду пояснень до завдань. Особливістю є система заохочення та рейтингів, що підвищує мотивацію учнів. Проте функціонал побудови адаптивних тестів та розгорнутої статистики за розділами у платформі відсутній [7].

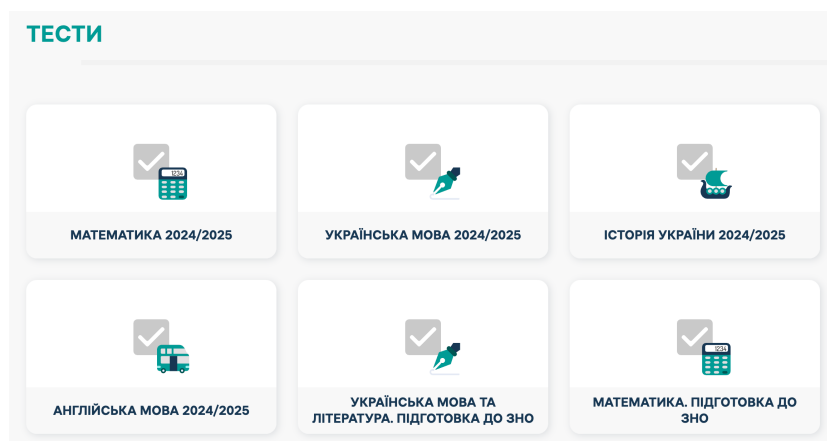


Рисунок 1.4 – Інтерфейс платформи «iLearn»

«EdEra» (рис. 1.5) – це навчальна платформа з інтерактивними курсами, тестами та лекціями. Вона пропонує тематичні модулі для самостійного навчання, однак не забезпечує швидкої перевірки тестів у режимі реального часу та комплексної аналітики результатів за конкретними темами чи блоками. Однак, попри широку тематику та якісне подання контенту, платформа має низку обмежень з точки зору практичної підготовки до стандартизованого тестування. Зокрема, вона не передбачає швидкої перевірки тестів у режимі реального часу, що ускладнює оперативний зворотний зв'язок для користувача.



Рисунок 1.5 – Інтерфейс платформи «EdEra»

Розробка власного інтерактивного вебзастосунку для підготовки до НМТ передбачає впровадження функціоналу, який не лише поєднує ключові переваги наявних освітніх сервісів, а й усуває низку їхніх поширених обмежень. Зокрема, планується створення адаптивної системи формування тестів відповідно до рівня знань користувача, інтеграція підказок і пояснень до відповідей, а також розробка блоку персональної аналітики з можливістю відстежувати прогрес за темами і видами завдань [8].

Результати порівняння функціональних можливостей існуючих рішень зведено в таблицю 1.1.

Таблиця 1.1 – Порівняльна характеристика аналогів

Критерії	Zno.ua	На урок	Prometheus	iLearn	EdEra	Власний застосунок
Повноцінний НМТ-формат тестів	1	1	0	1	0	1
Адаптивне формування тестів	0	0	0	0	0	1
Пояснення до відповідей	0	1	1	1	1	1
Персональна аналітика за темами	0	1	0	0	0	1
Зручність і швидкість перевірки результатів	1	1	1	1	0	1
Сумарний коефіцієнт	2	4	2	3	1	5

Відповідно до порівняльного аналізу, розробка власного вебзастосунку для підготовки до НМТ є доцільною, оскільки дозволить об'єднати ключові переваги наявних рішень та реалізувати функціонал, якого бракує конкурентам. Зокрема – адаптивне формування тестів, персоналізовану аналітику прогресу та інтерактивні пояснення.

Розробка власного вебзастосунку дозволяє поєднати найкращі практики реалізації тестування з урахуванням індивідуальних особливостей кожного користувача. Планується реалізувати:

- механізм підбору завдань на основі попередніх помилок;
- пояснення до відповідей, які надаються автоматично;
- дашборд аналітики з графіками прогресу;
- тематичні добірки для повторення.

Отже, розробка інтерактивного вебзастосунку для підготовки до НМТ відкриває широкі можливості для підвищення якості та ефективності підготовки випускників до тестування. Завдяки інтерактивним тестам, адаптивному формуванню завдань та персоналізованій аналітиці, користувачі зможуть отримувати актуальні знання, вчасно виявляти прогалини у підготовці та коригувати навчальний план відповідно до власного рівня. Це сприятиме кращим результатам на тестуванні та забезпечить більш індивідуалізований підхід до навчання.

1.3 Аналіз методів розв'язання задачі

Розробка інтерактивного вебзастосунку для підготовки до національного мультипредметного тесту вимагає комплексного підходу до вирішення завдань, пов'язаних із формуванням тестових завдань, адаптацією навчального контенту до рівня знань користувача та забезпеченням зручної інтерактивної взаємодії. У цьому розділі розглядаються основні методи, які застосовуються в сучасних цифрових освітніх платформах, їхні переваги, обмеження, а також доцільність використання у межах реалізації поставленої мети.

Перший із поширених методів – реалізація статичної системи тестування. У таких системах набір завдань попередньо визначений, тестові сесії формуються з фіксованого переліку питань, а результати зберігаються в базі даних без глибокого аналізу. Цей підхід є технічно простим, добре масштабованим і достатнім для базової перевірки знань. Однак він має суттєві обмеження: відсутність динамічного підбору завдань, нездатність враховувати рівень підготовки учня, відсутність зворотного зв'язку та персоналізованих підказок [9].

Другий метод – використання алгоритмів адаптивного тестування. Суть цього підходу полягає в тому, що складність та тематика завдань змінюються в залежності від попередніх відповідей користувача. Наприклад, якщо учень не справляється з темою, система може зменшити складність або запропонувати повторення схожих завдань. До реалізації таких механізмів залучаються прості евристики (підрахунок правильних відповідей за темами), а в складніших випадках

– моделі на основі машинного навчання. Перевага – індивідуалізація навчального процесу. Обмеження – складність налаштування логіки переходу між темами та потреба в накопиченні достатнього обсягу навчальних даних [10].

Окремо варто виділити модуль аналітики прогресу, як окремий напрям підвищення ефективності навчання. Такий модуль збирає статистику за тестами (наприклад, час проходження, правильність відповідей, повторюваність помилок) та виводить її у вигляді графіків, діаграм або таблиць. На основі цієї інформації учень може самостійно оцінити динаміку своїх результатів, а викладач або консультант – вчасно скоригувати стратегію підготовки. Дашборд з аналітикою, на відміну від звичайної таблиці результатів, дозволяє візуально оцінювати тенденції та виявляти теми, які потребують додаткового опрацювання.

Особливо перспективним на сьогодні є підхід, що поєднує в собі адаптивне тестування, аналітичний дашборд і використання LLM (великих мовних моделей) для формування пояснень і навчальної підтримки. В межах цієї роботи запропоновано інтеграцію ChatGPT API – сучасної мовної моделі, здатної формувати зрозумілі й точні пояснення до тестових завдань. Такий підхід дозволяє:

- надавати пояснення до відповідей у форматі, адаптованому до рівня учня;
- забезпечувати персоналізоване повторення на основі зроблених помилок;
- зменшити залежність від заздалегідь підготовлених текстів і матеріалів.

На відміну від шаблонних підказок, LLM може генерувати пояснення до будь-якого завдання на льоту, що підвищує масштабованість і актуальність матеріалу. Це не лише покращує розуміння складних тем, але й значно підвищує мотивацію до навчання, оскільки користувач отримує чітку відповідь на своє питання в зручному форматі.

У таблиці 1.2 наведено узагальнене порівняння методів, які можуть бути використані при розробці навчального вебзастосунку.

Таблиця 1.2 – Порівняльна характеристика методів реалізації систем підготовки до тестувань

Метод	Переваги	Обмеження
Статичне тестування	Простота реалізації, стабільність	Відсутність адаптивності, персоналізації, аналізу
Адаптивне тестування	Добір завдань за рівнем знань, покращення результатів	Складна логіка, залежність від даних
Аналітичний модуль (дашборд)	Візуалізація прогресу, виявлення слабких місць	Не дає пояснень до помилки, потребує коректної інтеграції
LLM (ChatGPT API) для пояснень	Персоналізовані пояснення, масштабованість, актуальність	Потреба в зовнішньому API, контроль якості відповідей

Результатом аналізу стало формування підходу, який поєднує три ключові компоненти:

1. Адаптивне формування тестів – динамічне підлаштування рівня складності.
2. Аналітичний дашборд – інструмент самоконтролю та прогресу.
3. Механізм пояснень на основі API LLM – підтримка учня після помилки.

Ця комбінація дозволяє створити не просто сервіс для перевірки знань, а повноцінну цифрову навчальну систему, здатну реагувати на індивідуальні труднощі користувача, формувати стійке розуміння тем і забезпечити ефективну підготовку до НМТ.

1.4 Постановка задачі дослідження

У межах бакалаврської кваліфікаційної роботи передбачається розробка системи, яка поєднує адаптивне тестування, інтерактивний інтерфейс, модулі аналітики та використання великої мовної моделі (LLM) для покращення навчального процесу. Особливістю системи є застосування API ChatGPT для генерації пояснень до помилок та підтримки персоналізованого повторення. Такий підхід дозволяє уникнути жорстко закладених шаблонів та забезпечує масштабованість без втрати якості зворотного зв'язку.

З метою досягнення поставленої в роботі цілі, доцільно реалізувати наступні задачі:

- розробити механізм формування тестових завдань, який дозволяє динамічно змінювати складність тестів залежно від попередніх результатів користувача, частоти помилок і рівня правильності у конкретних темах;
- створити модуль автоматизованого аналізу результатів тестування, що дає змогу виявляти слабкі місця у знаннях учня, зберігати статистику виконання завдань та групувати її за темами;
- інтегрувати API LLM для реалізації функціоналу пояснення помилок. Це дозволить автоматично генерувати текстові пояснення до запитів користувача у відповідній навчальній темі та формувати персоналізовані добірки для повторення;
- реалізувати візуальний дашборд аналітики прогресу, який у зручній графічній формі відображає динаміку знань за темами, частотою спроб, рівнем складності, що сприяє самоконтролю користувача та ефективнішому плануванню підготовки;
- розробити інтерфейс користувача, орієнтований на зручність, інтуїтивність і адаптивність до різних розмірів екранів.
- забезпечити повноцінне функціонування вебзастосунку на основі сучасного технологічного стеку: JavaScript та Vue.js для клієнтської частини, PHP (Laravel) для серверної логіки, PostgreSQL – для зберігання даних користувачів, результатів тестування та статистики;

- провести тестування вебзастосунку відповідно до визначених функціональних та експлуатаційних вимог, оцінити коректність роботи основних модулів, зручність інтерфейсу, відповідність поставленим завданням та очікуванням цільової аудиторії.

Зазначені задачі мають комплексний характер та передбачають не лише реалізацію програмного рішення, а й апробацію ефективності запропонованих методів взаємодії користувача із системою. Результатом виконання цих задач стане створення багатофункціонального вебзастосунку з підтримкою адаптивного навчання та інтерактивного супроводу, який може використовуватись як індивідуальними учнями, так і освітніми закладами для організації сучасної підготовки до НМТ.

1.5 Висновки

У першому розділі було проведено аналіз сучасного стану програмних засобів для підготовки до НМТ. Розглянуто найбільш популярні системи онлайн-тестування та адаптивного навчання, їхні можливості, переваги та обмеження.

На основі проведеного аналізу було визначено доцільність розробки власного інтерактивного вебзастосунку, який поєднуватиме адаптивне тестування, персональні рекомендації та інтерактивний інтерфейс. В результаті сформовано перелік основних технічних та функціональних задач, які необхідно вирішити в межах розробки: від алгоритмів формування тестів до модулів збереження та обробки навчальних даних. Запропоновано структуру та концепцію програмного забезпечення, яка дозволяє забезпечити адаптивне, гнучке та ефективне навчальне середовище для самопідготовки до НМТ.

Таким чином, на основі проведеного аналізу визначено концепцію та функціональну структуру майбутнього застосунку, що дозволяє перейти до етапу безпосередньої розробки.

2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ

2.1 Формалізація вхідних даних та постановка функціональних вимог

Ефективна розробка інтерактивного вебзастосунку для підготовки до національного мультипредметного тесту (НМТ) вимагає чіткого визначення типів вхідних даних та формулювання функціональних вимог до системи. Формалізація цих елементів дозволяє створити узгоджену архітектуру застосунку, забезпечити коректну обробку інформації, зручність взаємодії для користувача та можливість масштабування функціоналу.

Вебзастосунок повинен забезпечувати стабільну роботу з такими основними групами вхідних даних:

- Облікові дані користувачів – передбачають введення імені, електронної адреси, пароля, а також вибір ролі у системі (учень, викладач, адміністратор). Ці дані необхідні для авторизації, диференційованого доступу до функціоналу та збереження індивідуальної статистики.

- Навчальний контент – включає тестові завдання з предметів НМТ (українська мова, математика, історія України). Кожне завдання містить текст питання, список варіантів відповідей, правильну відповідь, тематичну категорію та пояснення. Контент може створюватися вручну через адміністративну панель або імпортуватися із зовнішніх джерел.

- Результати тестування – після проходження тесту користувачем система зберігає обрані варіанти, час виконання, кількість правильних відповідей, відсоткову оцінку, а також згенеровані на основі результатів рекомендації. Ці результати є основою для побудови індивідуальних навчальних маршрутів.

- Аналітичні та статистичні дані – охоплюють інформацію про теми, у яких користувач робить найбільше помилок, середні показники успішності, динаміку прогресу, частоту спроб та інші метрики. Дані використовуються для генерації звітів, побудови графіків у дашборді й формування персоналізованих порад.

- Запити до API великої мовної моделі (LLM) – нововведенням у системі є можливість надсилати запит на генерацію пояснення помилкової відповіді або пояснення до теми, яка викликає труднощі. На основі відповіді моделі ChatGPT користувач отримує персоналізоване пояснення у доступній формі. Для цього необхідно враховувати параметри запиту: текст помилки або тема, рівень складності, мова інтерфейсу.

Для реалізації застосунку обрано перевірений стек технологій, що дозволяє забезпечити стабільну роботу, гнучкість розробки та масштабованість:

PHP та Laravel – для реалізації авторизації, взаємодії з базою даних, обробки логіки формування тестів, збереження результатів та організації запитів до зовнішніх сервісів.

JavaScript та Vue.js – для створення адаптивного, швидкого інтерфейсу користувача, підтримки SPA-підходу (Single Page Application), інтерактивних елементів та безперервної взаємодії з бекендом.

PostgreSQL – як основна СУБД для надійного зберігання структурованих даних: облікових записів, тестових завдань, відповідей користувачів, статистики та логів взаємодії з мовною моделлю.

ChatGPT API – інтегрується для реалізації функціоналу генерації пояснень до помилок та тематичних запитів користувача. Вона працює як зовнішній інтелектуальний модуль, що формує природномовні відповіді в режимі реального часу, адаптовані до контексту та рівня знань користувача.

Додатково: бібліотеки візуалізації даних (Chart.js або ApexCharts) – для побудови графіків у дашборді користувача; Fetch API – для асинхронного обміну даними між фронтендом і сервером без повного оновлення сторінки.

Таким чином, правильно визначена структура вхідних даних і функціональні вимоги дозволяють сформулювати технічну основу для створення масштабованого та ефективного вебзастосунку. Особливу роль у цьому відіграє інтеграція з API LLM, що значно підвищує адаптивність і навчальну цінність розробленої системи, відрізняючи її від більшості аналогічних рішень на ринку.

2.2 Розробка інтерфейсу програмного додатку

Інтерфейс користувача (UI) є одним із найважливіших елементів інтерактивного вебзастосунку, оскільки саме він визначає зручність використання системи, швидкість навігації, рівень залученості та загальне враження користувача від взаємодії з продуктом. При розробці інтерфейсу було дотримано таких ключових принципів: адаптивність, мінімалізм, візуальна ієрархія та швидкий доступ до основних функцій.

Верстка реалізована на основі HTML5, CSS3 та Bootstrap 5, що дозволило скористатися готовою сіткою, типографікою, стилізованими елементами форм та модульною структурою для швидкої побудови інтерфейсних блоків. Завдяки цьому сторінки коректно відображаються як на великих екранах, так і на смартфонах, без втрати функціональності чи контексту. Інтерфейс містить кілька основних екранів, що будуть детально описані нижче в тексті.

На головному екрані (рис. 2.1) розміщено шапку з логотипом, посиланнями «Мої курси» та «Вийти» й hero-секцію з коротким слоганом і кнопкою швидкого переходу до особистого кабінету. Нижче виведено розділ «Наші курси», де картки формуються динамічно з масиву даних, кожна картка містить зображення, назву дисципліни й короткий опис, що полегшує вибір.

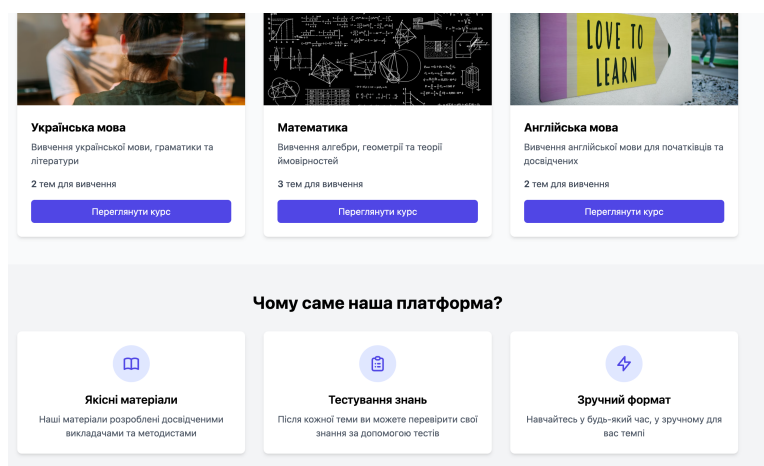


Рисунок 2.1 – Головний екран вебзастосунку

Після авторизації користувач потрапляє до особистого кабінету (рис. 2.2). Тут відображається персональне привітання й лічильник обраних курсів. Використано помірну кількість вільного простору, тому блок виглядає легким і не перевантаженим текстом.

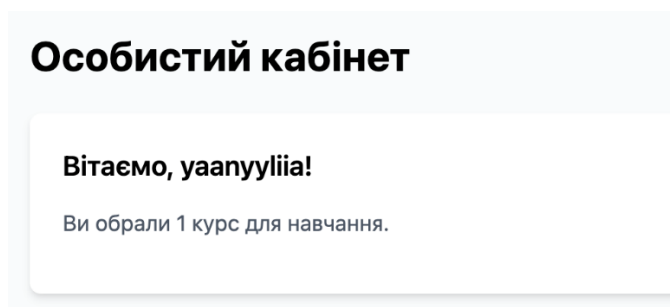


Рисунок 2.2 – Особистий кабінет абітурієнта

Секція «Мої курси» (рис. 2.3) подає кожен курс у вигляді картки з обкладинкою, детальним описом і прогрес-баром, що обчислюється на основі кількості пройдених уроків. Кнопка «Перейти до курсу» веде до матеріалів, а «Видалити» дає змогу миттєво відмовитися від дисципліни.

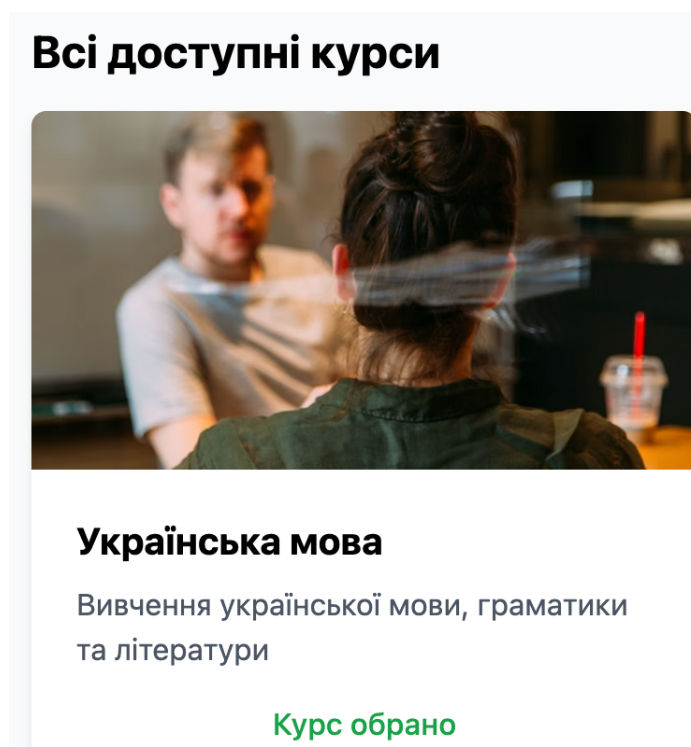


Рисунок 2.3 – Картка окремого обраного курсу

Усередині курсу (рис. 2.4) екран ділиться на дві колонки: ліворуч – список уроків із підсвічуванням активного, праворуч – навчальний контент. Кнопка «Позначити як пройдений» активує оновлення прогресу як на клієнті, так і на сервері. Така структура дозволяє легко повертатися до будь-якого уроку та фіксувати досягнення.

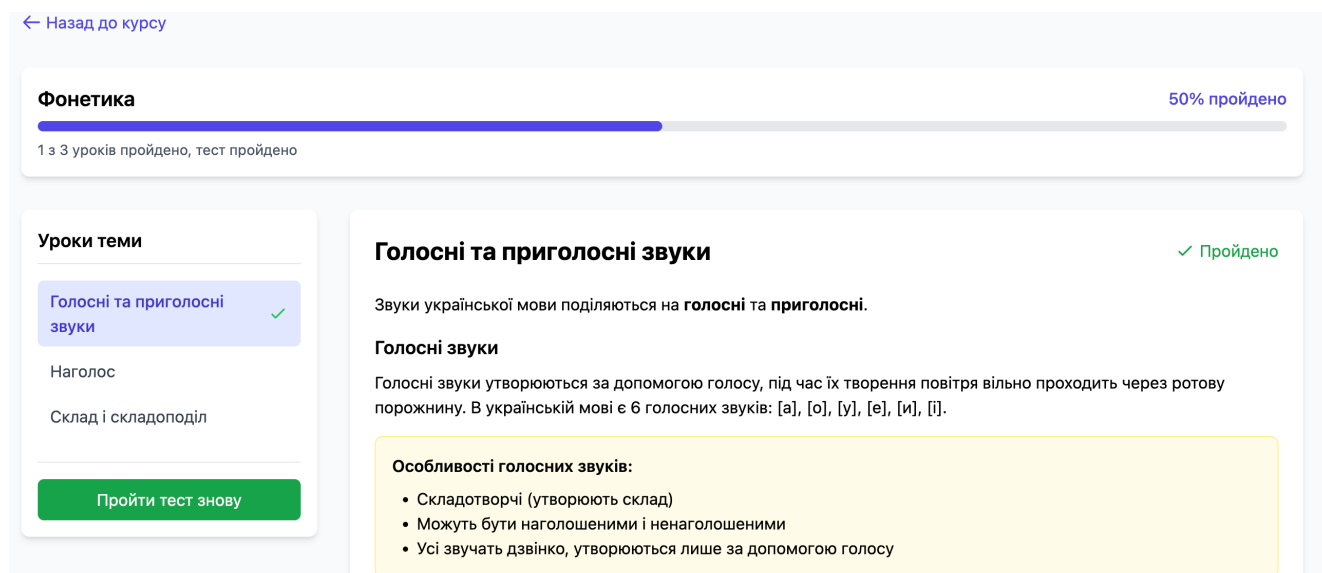


Рисунок 2.4 – Інтерфейс уроку із можливістю відслідковування прогресу

Тестовий модуль (рис. 2.5) відтворює структуру офіційного НМТ: запитання подано дрібним номерованим заголовком, під ним – варіанти відповіді, оформлені у вигляді радіокнопок із радіальним індикатором вибору. Після надсилання відповіді прогрес-бар у картці курсу оновлюється миттєво, а результати тесту передаються на сервер. Тестовий модуль (рис. 2.5) побудовано у стилістиці офіційного НМТ. Кожне запитання супроводжується варіантами відповідей, оформленими у вигляді радіокнопок. Після надсилання тесту, результати передаються на сервер, а прогрес автоматично оновлюється. Надалі передбачено відображення пояснення до помилок, які надаються через LLM на основі тексту запитання та помилкового вибору користувача.

Тест з фонетики

1. Скільки голосних звуків в українській мові?

☐ 5

☒ 6

☐ 10

☐ 12

Рисунок 2.5 – Графічний інтерфейс тестового модуля

Блок «Аналітика результатів тестів» (рис. 2.6) відображає зведену статистику навчання. Верхня панель містить три інформаційні картки: середній результат, найкращий результат і кількість завершених тестів. Ці візуальні компоненти сприяють самоконтролю та допомагають користувачу самостійно визначити, які теми потребують додаткової уваги.

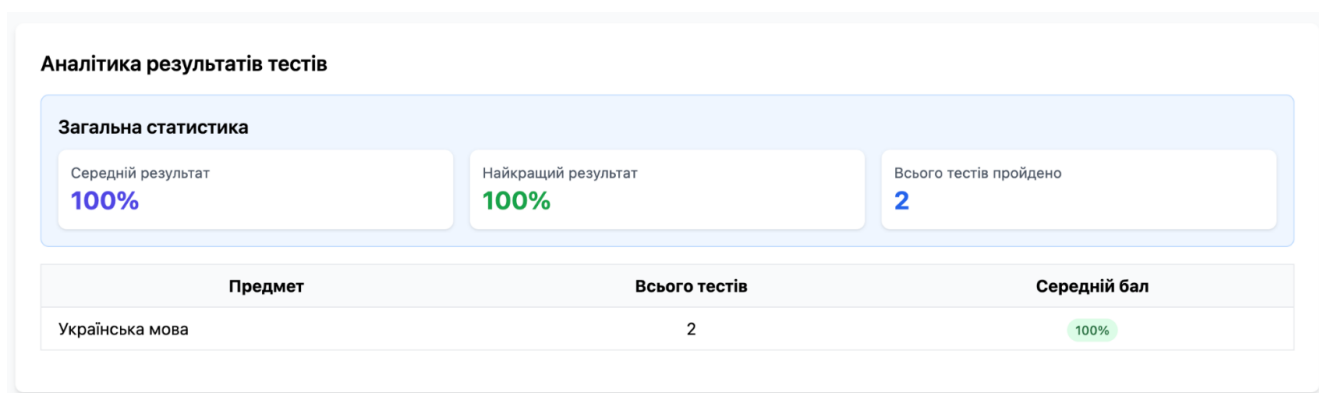


Рисунок 2.6 – Блок із аналітикою результатів тестування

На рисунку 2.7 зображено компонент інтерфейсу модуля перевірки тестових відповідей вебзастосунку. Користувач дав неправильну відповідь на питання про кількість голосних звуків в українській мові, система автоматично підсвітила правильний варіант та позначила помилковий. В нижній частині блоку наведено пояснення до відповіді, згенероване за допомогою інтеграції з мовною моделлю

GPT-4o Mini. Цей фрагмент інтерфейсу реалізує функціональність надання зворотного зв'язку після тестування.

Детальний розбір відповідей

Питання 1 X Неправильно ?

Які числа відносяться до натуральних?

☒ -1, 0, 1 Ваша відповідь

☒ 1, 2, 3 Правильна відповідь

☐ 0, 1, 2

☐ 0, -1, 2

? Пояснення від GPT-4o Mini

Привіт! Натуральні числа – це ті числа, які ми використовуємо для підрахунку і позначають кількість. Вони завжди натуральні, позитивні і починаються з 1: 1, 2, 3 і так далі. Тому правильна відповідь – це 2, адже 1, 2, 3 – це натуральні числа, а 0 і -1 – ні.

Рисунок 2.7 – Інтерфейс перевірки результатів з поясненням від LLM

На рисунку 2.8 представлено інтерфейс модуля генерації навчального контенту, що використовує штучний інтелект GPT-4o Mini. Користувач має змогу автоматично створювати нові уроки, натиснувши кнопку «Генерувати урок». Цей функціонал полегшує створення освітніх матеріалів і дозволяє формувати навчальні блоки на основі заданих тем або параметрів.

Генерація уроків з ШІ
Створюй нові уроки автоматично за допомогою GPT-4o Mini

+ СХОВАТИ

Оберіть тему для уроку

Алгебра

Ідеї для уроків

ОТРИМАТИ ІДЕЇ

Основи функцій: Відображення та графіки ✓

Математичні нерівності: Розв'язування та застосування

Системи рівнянь: Графічний та алгебраїчний методи

Моделювання задач: Використання алгебри у реальному житті

Алгебра в іграх: Розв'язування головоломок та задач з використанням рівнянь

Або введіть свою назву уроку

Основи функцій: Відображення та графіки

Генерація може зайняти 10-30 секунд

ГЕНЕРУЮ УРОК...

Рисунок 2.8 – Інтерфейс генерації уроків за допомогою LLM

На рисунку 2.9 показано повідомлення системи про завершення процесу генерації уроку.

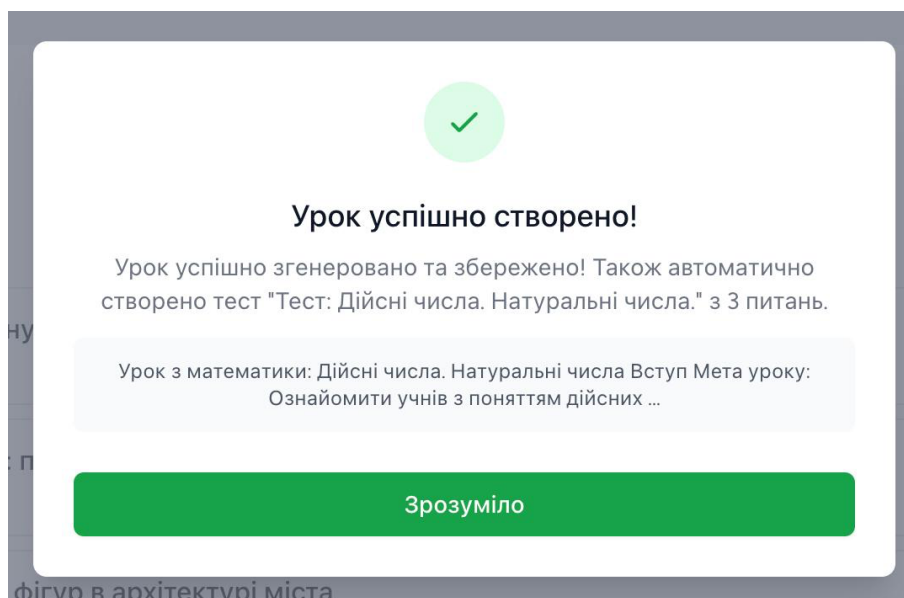


Рисунок 2.9 – Вікно із повідомленням про успішне створення уроку за допомогою

Інтерфейс розроблено за принципом Mobile First, тобто першочергово оптимізовано для смартфонів, що забезпечує зручність використання на найпоширеніших пристроях серед учнівської аудиторії.

Кольорова схема підібрана нейтральна, з акцентами на кнопках та елементах навігації, що полегшує орієнтацію та не перевантажує зір.

Використано адаптивну типографіку, піктограми та іконки для полегшення сприйняття дій навіть без читання інструкцій.

Отже, розроблений інтерфейс вийшов простим і зрозумілим: усі сторінки швидко відкриваються, адаптивно підлаштовуються під екран, а користувач без зусиль знаходить потрібні курси, стежить за прогресом і одразу бачить результати тестів.

2.3 Розробка структур програмного продукту

Застосунок для підготовки до національного мультипредметного тесту має чітко визначену структуру, що включає кілька основних компонентів, кожен з яких виконує свою функцію для забезпечення інтерактивного процесу навчання та підготовки. Структура застосунку побудована на основі модульної архітектури, що дозволяє користувачам проходити інтерактивні тести, виконувати вправи та аналізувати свої результати. Основною частиною програми є клас TestManager, який інкапсулює бізнес-логіку для відображення питань, обробки відповідей і збереження результатів тестувань. Цей клас взаємодіє з базою даних, що містить різноманітні тестові завдання для кожної дисципліни національного мультипредметного тесту.

Застосунок складається з кількох основних компонентів, кожен з яких виконує свою функцію для ефективної підготовки користувача до тесту (рис 2.10).

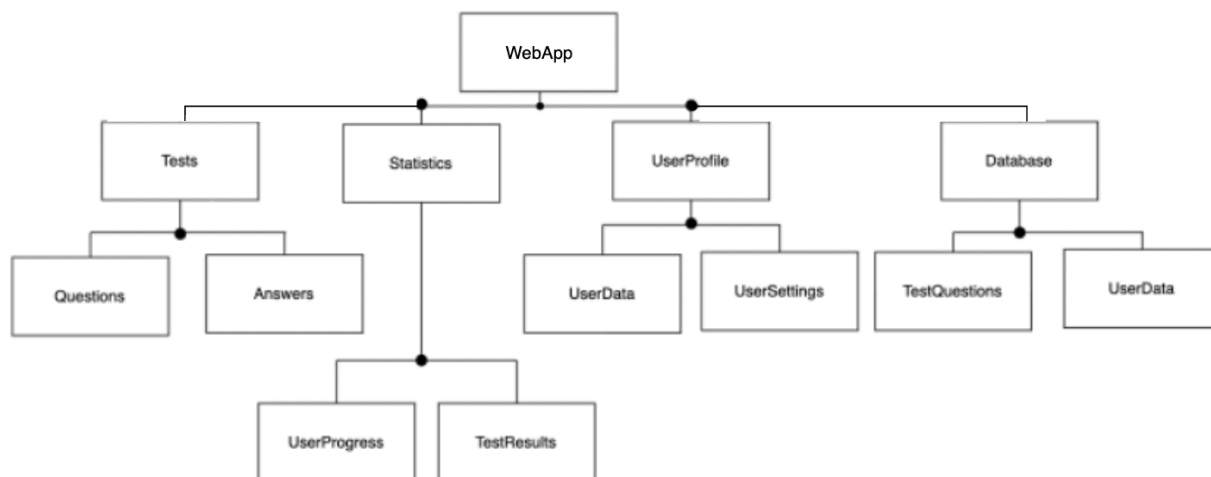


Рисунок 2.10 – Структурна схема застосунку

Головна сторінка застосунку включає інтерфейс для взаємодії з користувачем та відображення тестових завдань і результатів. Структурна схема головної сторінки вебзастосунку зображена на рисунку 2.12.

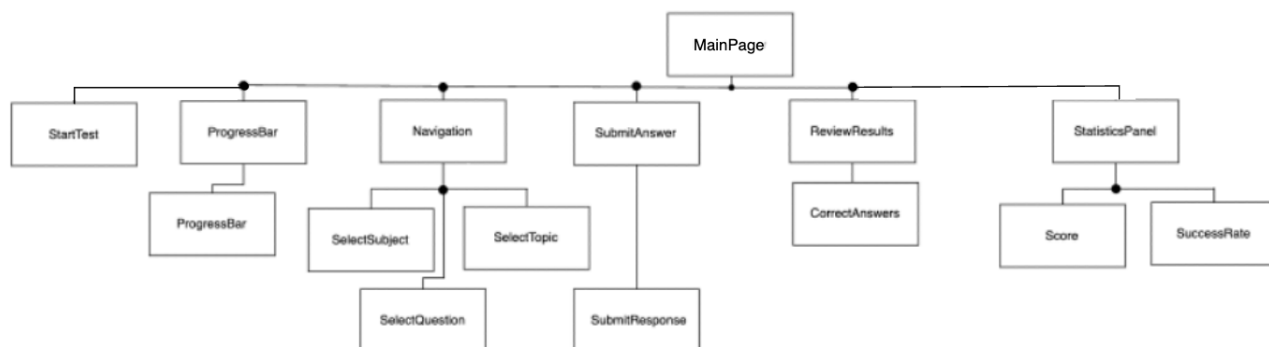


Рисунок 2.11 – Структурна схема головної сторінки вебзастосунку

Вся інформація про питання та відповіді зберігається в базі даних, що дозволяє швидко оновлювати та змінювати дані, а також забезпечує інтерактивність тестування. База даних взаємодіє з компонентом TestManager, який отримує запити для відображення нових питань та зберігання результатів користувачів. Застосунок здійснює оновлення тестових завдань у реальному часі, а також автоматично генерує статистику по кожному тестуванню. Це дозволяє користувачу стежити за своїм прогресом, а також отримувати рекомендації щодо покращення результатів. Усі результати зберігаються у UserProfile та

відображаються на панелі статистики, що дає змогу здійснювати детальний аналіз успішності та виділяти слабкі місця для подальшого навчання.

Розробка інтерактивного вебзастосунку для підготовки до національного мультипредметного тесту передбачає чітке структуроване середовище, яке дозволяє користувачам зручно і ефективно проходити навчання. Забезпечення інтерактивності, відображення результатів та адаптація тестів до рівня знань користувача є важливими аспектами, які допоможуть створити зручний інструмент для підготовки до тестування. Модульна структура застосунку дозволяє легко впроваджувати нові функціональні можливості та тестові завдання, адаптуючи програму під вимоги користувачів.

2.4 Розробка алгоритмів роботи системи

У процесі створення вебзастосунку для підготовки до національного мультипредметного тесту, було розроблено низку алгоритмів, які забезпечують коректну та стабільну роботу користувацького інтерфейсу, обробку введених даних і взаємодію з навчальними модулями. Ключовими функціональними елементами стали оновлення навчального прогресу, пошук і фільтрація тестових завдань, обробка результатів проходження тестів, запуск нової сесії тестування, завершення поточної або всієї серії запитань. Такий комплекс алгоритмів створює стабільну основу, на якій будується персоналізований досвід підготовки й дозволяє масштабувати систему без втрати продуктивності.

Алгоритм оновлення списку доступних або активних тестів реалізується при завантаженні сторінки або при натисканні кнопки «Оновити». Система запитує актуальні дані про сесію користувача, прогрес проходження та статус тестів, після чого відображає їх у таблиці. На рисунку 2.12 представлено блок-схему цього процесу.

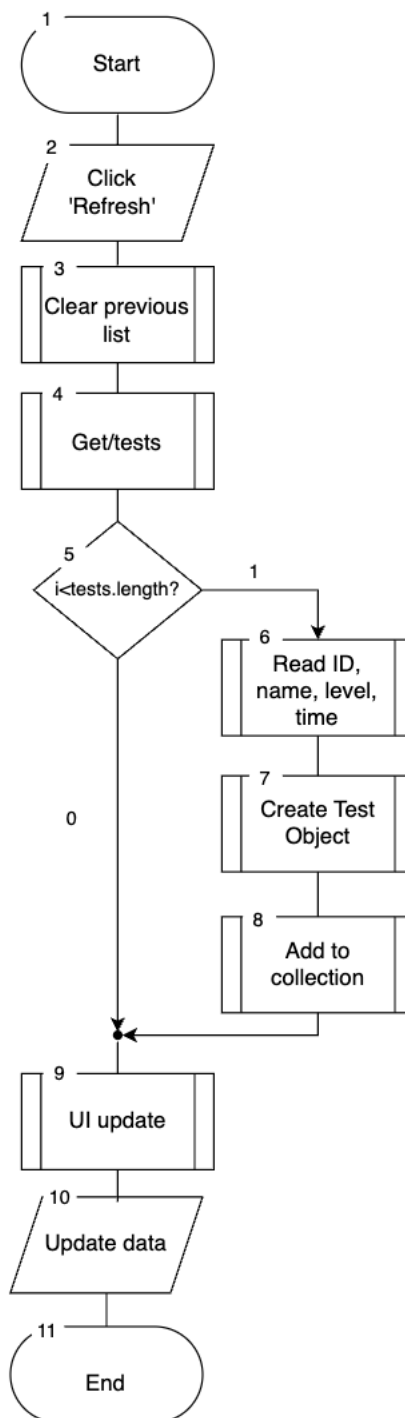


Рисунок 2.12 – Блок-схема оновлення списку тестів

Фільтрація тестів реалізована через подію `TextChanged`, яка виникає під час введення тексту у відповідне поле. Залежно від введеного фрагмента (назва теми, тип завдання, рівень складності), система динамічно оновлює вміст таблиці, виводячи лише релевантні варіанти. Алгоритм подано на рисунку 2.13.

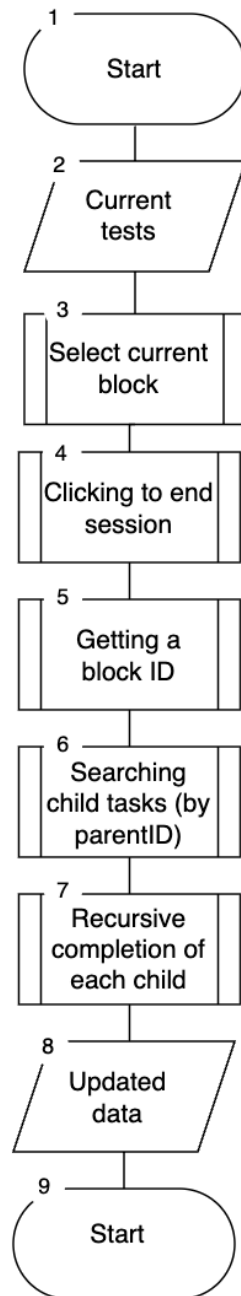


Рисунок 2.13 – Блок-схема алгоритму пошуку/фільтрації

У разі потреби користувач може завершити окрему сесію до її логічного завершення (наприклад, у разі помилки або за бажанням почати з іншого модуля). Перед виконанням дії система перевіряє наявність вибраного завдання, після чого закриває відповідний запис, обробляє виключення і оновлює таблицю. Алгоритм представлено на рисунку 2.14.

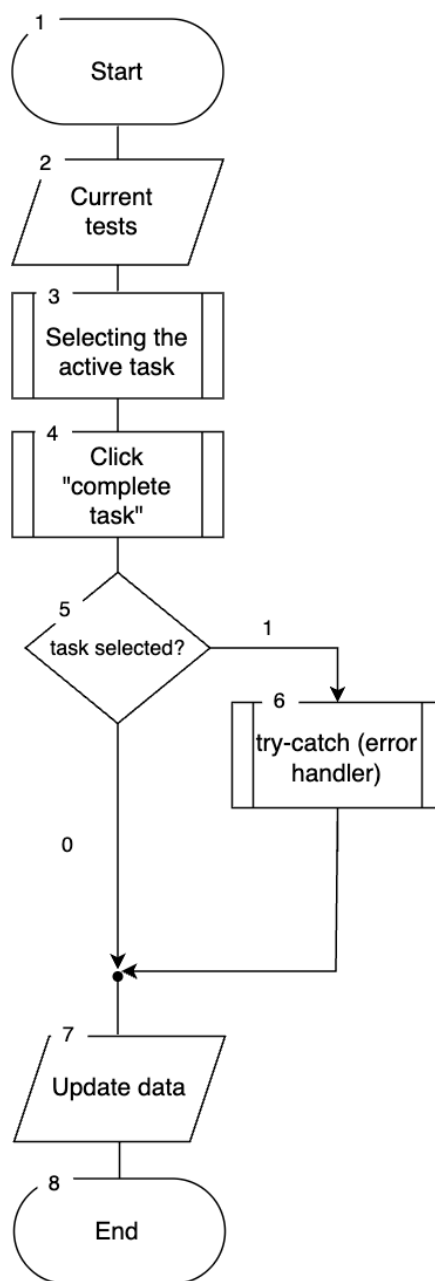


Рисунок 2.14 – Блок схема алгоритму завершення вибраного завдання

Після завершення попередньої сесії або вручну користувач може ініціювати запуск нового тесту. Для цього вводиться шлях до відповідного ресурсу або обирається тест із каталогу. Система перевіряє коректність введення, формує об'єкт запуску, обробляє можливі помилки й оновлює інтерфейс. Сценарій зображено на рисунку 2.15.

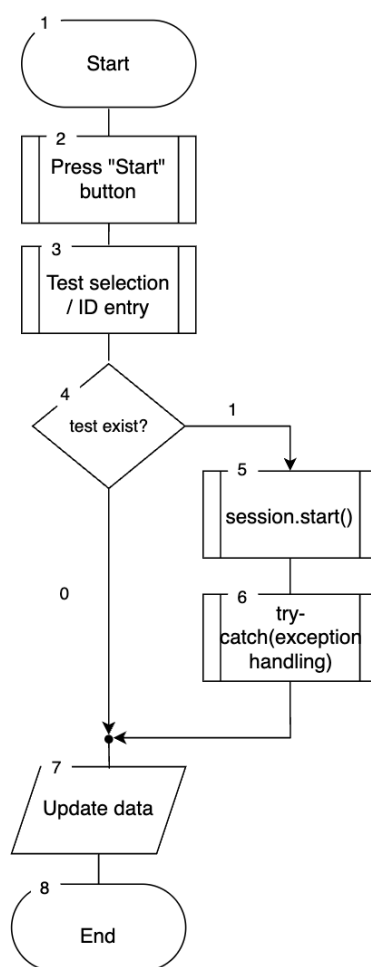


Рисунок 2.15 – Блок-схема алгоритму запуску нової сесії

Таким чином, алгоритмічна логіка вебзастосунку охоплює повний цикл взаємодії з тестовими модулями. Кожен блок реалізовано з урахуванням можливих виключень і забезпечує безпечну та передбачувану поведінку системи.

2.5 Висновки

У другому розділі здійснено ґрунтовний аналіз інформаційних потоків інтерактивного вебзастосунку для підготовки до національного мультипредметного тесту: розібрано формат вхідних даних та вихідних результатів. Окрему увагу приділено проєктуванню адаптивного, mobile-first інтерфейсу, що динамічно підлаштовується під сценарії роботи майбутніх абітурієнтів.

3. РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ ПЛАТФОРМИ ПІДГОТОВКИ ДО НМТ

3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення

Щоб обрати оптимальний серверний стек для платформи онлайн-підготовки до НМТ, було проаналізовано п'ять найуживаніших технологій: PHP (Laravel), Node.js (Express), Python (Django), Java (Spring Boot) та Ruby (Ruby on Rails). Для кожної з них оцінили шість параметрів, критичних саме для освітнього веб-проєкту.

Таблиця 3.1 – Функціональні характеристики мов програмування

Критерії	Мова				
	PHP	Node.js	Python	Java	Ruby
Продуктивність БД	1	1	0	1	0
Простота розгортання	1	0	0	0	1
Наявність SCROM/LTI пакетів	1	0	0	0	0
Зрілість екосистеми	1	1	1	1	0
Вартість хостингу	1	1	1	0	0
Швидкість створення MVP	1	1	1	0	0
Сумарний коефіцієнт	6	4	3	2	1

Сумарні результати показали, що зв'язка PHP 8 + Laravel 10 набрала найбільше балів, суттєво випередивши найближчого конкурента Node.js. Основними чинниками, що забезпечили таку перевагу, стали: простота розгортання на більшості популярних хостингів (у тому числі бюджетних), низька вартість обслуговування, висока зрілість екосистеми, а також наявність готових пакетів і бібліотек, що суттєво спрощують створення освітніх вебзастосунків. Крім того, платформа відзначилась швидким створенням MVP (мінімально життєздатного продукту), що стало можливим завдяки вбудованим засобам Laravel — Artisan CLI, шаблонам Blade, механізмам міграцій і роутінгу. Розвинена підтримка SCORM і LTI-модулів дозволяє без зайвого кодування підключати стандартизовані тести та курси [11].

Беручи до уваги результати порівняння й специфіку курсового проєкту, у ролі серверної платформи остаточно обрано PHP з фреймворком Laravel. MVC-архітектура Laravel чітко відокремлює бізнес-логіку від представлення, забезпечує зручне тестування та масштабування, тоді як фронтенд на HTML 5, CSS 3 і JavaScript (Bootstrap 5) у поєднанні з Fetch API гарантує миттєві оновлення даних без перезавантаження сторінки [12].

Така комбінація утворює економічну, швидку та технологічно зрілу основу для подальшого розвитку інтерактивної платформи підготовки до НМТ. Для реалізації клієнтської частини інтерфейсу було використано компонентний підхід, рекомендований у сучасних гайдлайнах для Vue 3 [13].

Основні методи роботи з DOM та асинхронною обробкою подій реалізовано згідно з рекомендаціями провідного довідника з JavaScript [14].

Окрім вибору серверного стеку, проєкт передбачав інтеграцію штучного інтелекту для пояснення помилок у тестах. Тому проведено окремий аналіз популярних API LLM, які можуть бути використані в освітньому контексті.

Таблиця 3.2 – Порівняння API LLM

Критерії	LLM				
	GPT-4o	Claude 3	Gemini	Mistral	Cohere
Якість відповідей	1	1	1	0	0
Адаптація до рівня користувача	1	1	0	0	0
Простота інтеграції через API	1	1	0	0	0
Документація та підтримка	1	0	1	0	0
Вартість при середньому навантаженні	0	0	1	1	1
Стабільність і масштабованість	1	1	1	0	0
Сумарний коефіцієнт	5	4	4	1	1

Найвищі бали отримав API від OpenAI, зокрема модель GPT-4o. Вона забезпечує високу якість генерації відповідей, вміє адаптувати стиль пояснення до рівня користувача, легко інтегрується в існуючу архітектуру через REST API, має чудову документацію і підтримку, а також стабільно працює при високому навантаженні. Попри вартість, яка дещо перевищує середню, баланс між функціональністю, гнучкістю й точністю пояснень робить її найкращим вибором для реалізації освітнього функціоналу.

Таким чином, вибір API від OpenAI є стратегічно обґрунтованим рішенням, що дозволяє впровадити адаптивні пояснення на основі відповідей користувача й забезпечити персоналізоване навчання без складного програмування або глибоких NLP-компетенцій.

3.2 Розробка модуля аналітики результатів тестування

У межах платформи підготовки до НМТ одним із ключових компонентів став модуль аналітики, що обчислює статистику проходження тестів і передає її у форматі, зручному для графічного відображення (див. рис. 3.1). Модуль реалізовано як Laravel-сервіс: він приймає сиру сесійну інформацію, агрегує її за предметами й повертає зведений JSON для візуалізації через Chart.js. основний стек базується на Laravel, методи організації MVC-архітектури запозичено з класичного підходу, описаного у навчальному курсі з Ruby on Rails [15].

Перший крок – визначення структури даних. У базі зберігаються сутності users, tests, attempts. Таблиця attempts містить user_id, test_id, кількість правильних відповідей та тривалість проходження. На основі цих полів формується запит до сервісу аналітики.

```
namespace App\Services;

use Illuminate\Support\Facades\DB;

class StatsService
{
    public function getUserStats(int $userId): array
    {
        $aggregate = DB::table('attempts')
            ->where('user_id', $userId)
            ->selectRaw('AVG(score)      as avg_score,
                    MAX(score)      as best_score,
                    COUNT(id)       as tests_done')
            ->first();

        $bySubject = DB::table('attempts as a')
            ->join('tests as t', 't.id', '=', 'a.test_id')
            ->where('a.user_id', $userId)
            ->groupBy('t.subject')
            ->selectRaw('t.subject,
                    COUNT(a.id)          as total,
                    ROUND(AVG(a.score)) as avg_score')
            ->get();

        return [
            'summary' => [
                'avg'   => (int) $aggregate->avg_score,
                'best'  => (int) $aggregate->best_score,
                'count' => (int) $aggregate->tests_done,
            ],
            'subjects' => $bySubject
        ];
    }
}
```

Рисунок 3.1 – Фрагмент коду сервісу StatsService, що агрегує результати тестувань

Контролер викликає сервіс і повертає відповідь у форматі JSON. На клієнті JavaScript-функція `fetch('/api/stats')` підтягує ці дані та передає їх до `Chart.js`, який будує стовпчикову та кругову діаграми, показані на скріншоті аналітики (рис. 3.2).

```
namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use App\Services\StatsService;
use Illuminate\Http\JsonResponse;

class StatsController extends Controller
{
    public function __construct(private
        StatsService $stats) {}

    public function show(): JsonResponse
    {
        $data = $this->stats->getUserStats(auth
            ()->id());
        return response()->json($data);
    }
}
```

Рисунок 3.2 – Фрагмент коду контролера StatsController

У процесі розробки особливу увагу приділено оптимізації запитів: замість декількох окремих звернень використано агрегатні функції SQL, що зменшує час відповіді й навантаження на базу. Кешування результатів у Redis гарантує миттєве відображення статистики при повторному відкритті секції «Аналітика результатів тестів». Фронтенд отримує об'єкт (рис. 3.3).

```
{
  "summary": { "avg": 78, "best": 92, "count": 27
  },
  "subjects": [
    { "subject": "Математика", "total": 10,
      "avg_score": 80 },
    { "subject": "Українська мова", "total": 9,
      "avg_score": 75 },
    { "subject": "Історія України", "total": 8,
      "avg_score": 79 }
  ]
}
```

Рисунок 3.3 – Приклад JSON-відповіді API

Chart.js формує лінійну діаграму динаміки й кругову діаграму розподілу балів. Комбінація PHP-бекенду та асинхронного Fetch-запиту робить блок аналітики повністю інтерактивним: користувач одразу бачить актуальні показники.

Таким чином, модуль аналітики не лише збирає й обробляє значний обсяг сесійних даних, а й забезпечує їх наочне представлення, підвищуючи інформативність інтерфейсу та мотивацію абітурієнтів до подальшого навчання.

3.3 Розробка модуля керування курсами й прогресом користувача

HTML-розмітка створювалась відповідно до сучасних стандартів, з використанням практик, наведених на W3Schools [18]. CSS-стилізація виконувалась згідно з рекомендаціями MDN Web Docs [19], що дозволило реалізувати адаптивний дизайн та зручну типографіку. Щоб прискорити розробку UI-компонентів і зменшити обсяг повторного коду, було інтегровано CSS-фреймворк Bootstrap 5 [20], що дозволив реалізувати адаптивні інтерфейси з єдиним набором стилів для мобільних і десктопних пристроїв.

Основний функціональний модуль, який відповідає за взаємодію користувача з переліком навчальних курсів, реалізує наступні визначені у попередніх розділах задачі: формування каталогу дисциплін, реєстрація абітурієнтів на обрані курси, збереження та обробка результатів проходження навчальних блоків, що включає в себе візуалізацію прогресу окремо обраного тесту, предметного курсу і, звісно, загалом прогресу користувача (таблиці, графіки, текстовий матеріал, тощо).

Модуль відповідає за формування каталогу дисциплін, запис абітурієнта на курс та фіксацію результатів проходження уроків. Його логіка зосереджена у сервісі CourseService (рис. 3.4), а власне HTTP-запити обробляє REST-контролер CourseController (рис. 3.5). Така двошарова побудова полегшує тестування: бізнес-методи перевіряються окремо від транспортного рівня, а контролер залишається тонким і відповідає лише за валідацію та серіалізацію даних.

```

class CourseService
{
    public function listWithProgress(int $userId)
    {
        return Course::withCount([
            'lessons as completed' => function ($q) use ($userId) {
                $q->whereExists(function ($sub) use ($userId) {
                    $sub->select(DB::raw(1))
                        ->from('lesson_user')
                        ->whereColumn('lesson_user.lesson_id', 'lessons.id')
                        ->where('lesson_user.user_id', $userId);
                });
            }
        ])->get();
    }

    public function toggleEnrollment(int $userId, int $courseId)
    {
        Course::findOrFail($courseId)
            ->students()
            ->toggle($userId);

        return $this->listWithProgress($userId);
    }

    public function markLessonComplete(int $userId, int $lessonId)
    {
        Lesson::findOrFail($lessonId)
            ->students()
            ->syncWithoutDetaching($userId);

        return $this->listWithProgress($userId);
    }
}

```

Рисунок 3.4 – Фрагмент коду CourseService

Метод `listWithProgress` запитує курси разом з обчисленою кількістю пройдених уроків. Для цього використано підзапит `whereExists`, який дає змогу отримати дані одним SQL-зверненням без післяобробки у PHP. Дві інші функції – `toggleEnrollment` та `markLessonComplete` – змінюють зв'язок `many-to-many` між користувачем і курсами або уроками, а потім одразу повертають оновлений каталог. Завдяки такій схемі фронтенд одержує повну картину прогресу одразу після будь-якої дії.

```

class CourseController extends Controller
{
    public function __construct(private CourseService $courses) {}

    public function index(): JsonResponse
    {
        return response()->json(
            $this->courses->listWithProgress(auth()->id())
        );
    }

    public function enroll(Request $r): JsonResponse
    {
        return response()->json(
            $this->courses->toggleEnrollment(
                auth()->id(),
                (int) $r->course_id
            )
        );
    }

    public function complete(Request $r): JsonResponse
    {
        return response()->json(
            $this->courses->markLessonComplete(
                auth()->id(),
                (int) $r->lesson_id
            )
        );
    }
}

```

Рисунок 3.5 – Контролер CourseController, який приймає запити каталогу, запису та позначення уроків

Контролер експонує три ендпоїнти: `/api/courses` повертає список дисциплін з прогресом, `/api/enroll` додає або знімає курс, а `/api/complete` фіксує проходження конкретного уроку. Усі відповіді мають однакову структуру, тому на клієнті достатньо одного обробника, що вставляє отримані дані у DOM і перераховує ширину індикатора прогресу на картці.

```
[
  {
    "id": 7,
    "title": "Математика",
    "lessons_count": 12,
    "completed": 8
  },
  {
    "id": 9,
    "title": "Українська мова",
    "lessons_count": 10,
    "completed": 10
  }
]
```

Рисунок 3.6 – JSON-відповідь, яку фронтенд перетворює на картки курсів зі смугою прогресу

Кожен об'єкт містить загальну кількість уроків і значення `completed`. На основі цих двох чисел JavaScript вираховує відсоток, який відтворюється у вигляді прогрес-бару. Оскільки сервер щоразу повертає повну колекцію курсів, інтерфейс оновлюється миттєво й залишається консистентним без додаткових локальних підрахунків.

3. 4 Розробка модуля інтегрування штучного інтелекту

У межах створення інтерактивної вебплатформи для підготовки до національного мультипредметного тесту (НМТ) було реалізовано спеціальний модуль, що інтегрує можливості штучного інтелекту (ШІ) на основі OpenAI ChatGPT-4o API. Основне завдання цього модуля — автоматизація пояснення помилок учнів після проходження тестів та генерація персоналізованих завдань для повторного опрацювання слабких тем.

Після завершення кожного тесту результати проходження автоматично фіксуються системою. Далі відбувається аналіз відповідей, у ході якого визначаються неправильні варіанти та встановлюються теми, які потребують додаткового вивчення. Для кожної помилки серверна частина формує запит до моделі ChatGPT-4o. Цей запит містить інструкцію пояснити, чому правильна відповідь є саме такою, а також включає додаткові параметри: рівень знань користувача, мову інтерфейсу та контекст самого завдання. У відповідь від API надходить коротке навчальне пояснення, яке зберігається в базі даних і

відображається учню у спеціальному інформаційному блоці одразу після завершення тесту.

Окрім пояснень, модуль також використовується для генерації нових завдань. На основі тем, у межах яких було допущено помилки, система формує додатковий запит до ChatGPT-4o. У ньому надається інструкція створити нове тестове запитання, що максимально відповідає формату національного мультипредметного тесту. Згенероване запитання автоматично додається до індивідуального списку учня для повторного проходження і пропонується під час наступної навчальної сесії.

Для реалізації взаємодії з OpenAI API використовувалась офіційна документація [21], що дозволило реалізувати стабільне з'єднання через HTTP-запити з подальшою обробкою відповідей у форматі JSON. Інтеграцію було реалізовано на серверному рівні за допомогою HTTP-клієнта фреймворку Laravel. Структура кожного запиту включає системне повідомлення з інструкцією, поточне тестове питання, відповідь користувача та відповідний контекст теми. Відповіді моделі зберігаються у базі даних із кешуванням, що дає змогу уникнути дублювання запитів при повторному зверненні до вже опрацьованих випадків.

Взаємодія з OpenAI API реалізована на серверному рівні за допомогою Laravel HTTP-клієнта. Структура запиту містить системне повідомлення з інструкцією, поточне питання, відповідь користувача та контекст теми. Отримані результати кешуються у базі даних, щоб уникнути дублювання запитів при повторному використанні. На рисунку 3.7 зображено діаграму послідовностей, яка описує взаємодію між компонентами системи під час завершення тесту користувачем. Діаграма демонструє, як дані проходять через фронтенд, бекенд, сервіс, базу даних і зовнішній API (OpenAI). Діаграму послідовностей було створено в PlantUML, оскільки даний інструмент дозволяє швидко будувати послідовності у текстовому форматі, що зручно для постійних оновлень продукту.

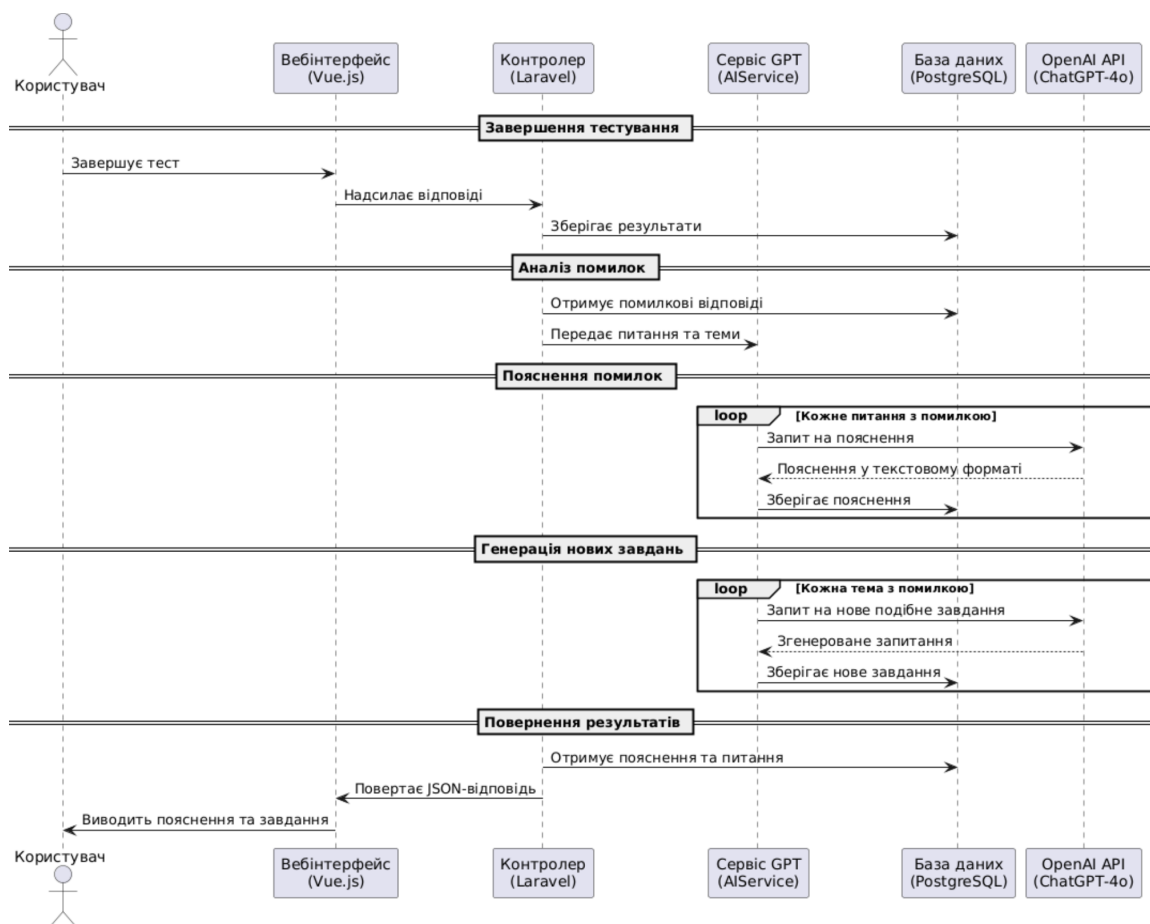


Рисунок 3.7– UML-діаграма послідовності інтеграції модуля III в застосунок підготовки до НМТ

Завдяки використанню моделі ChatGPT-4o стало можливим інтегрувати високоякісну навчальну підтримку без потреби в написанні великої кількості вручну підготовлених пояснень або статичних сценаріїв. Користувач отримує персоналізовані пояснення та тренувальні питання, що підвищує ефективність самостійної підготовки.

Рисунок 3.8 відображає логічну структуру бази даних у вигляді діаграми сутностей і зв'язків (ERD), що була створена для моделювання основних об'єктів системи та взаємодій між ними. У схемі визначено ключові сутності: користувачі, предмети, теми, уроки, тести та результати тестування, кожна з яких представлена окремою таблицею з відповідними атрибутами. Встановлено реляційні зв'язки між таблицями, зокрема відношення багато-до-багатьох між користувачами та предметами, а також ієрархічні залежності між предметами, темами, уроками і

тестами. Окрема таблиця для збереження результатів тестування забезпечує збирання інформації про спроби проходження тестів, що включає дату, час, кількість балів і статус. Така структура дає змогу ефективно організувати дані, підтримувати узгодженість і масштабованість системи при розширенні функціональності.

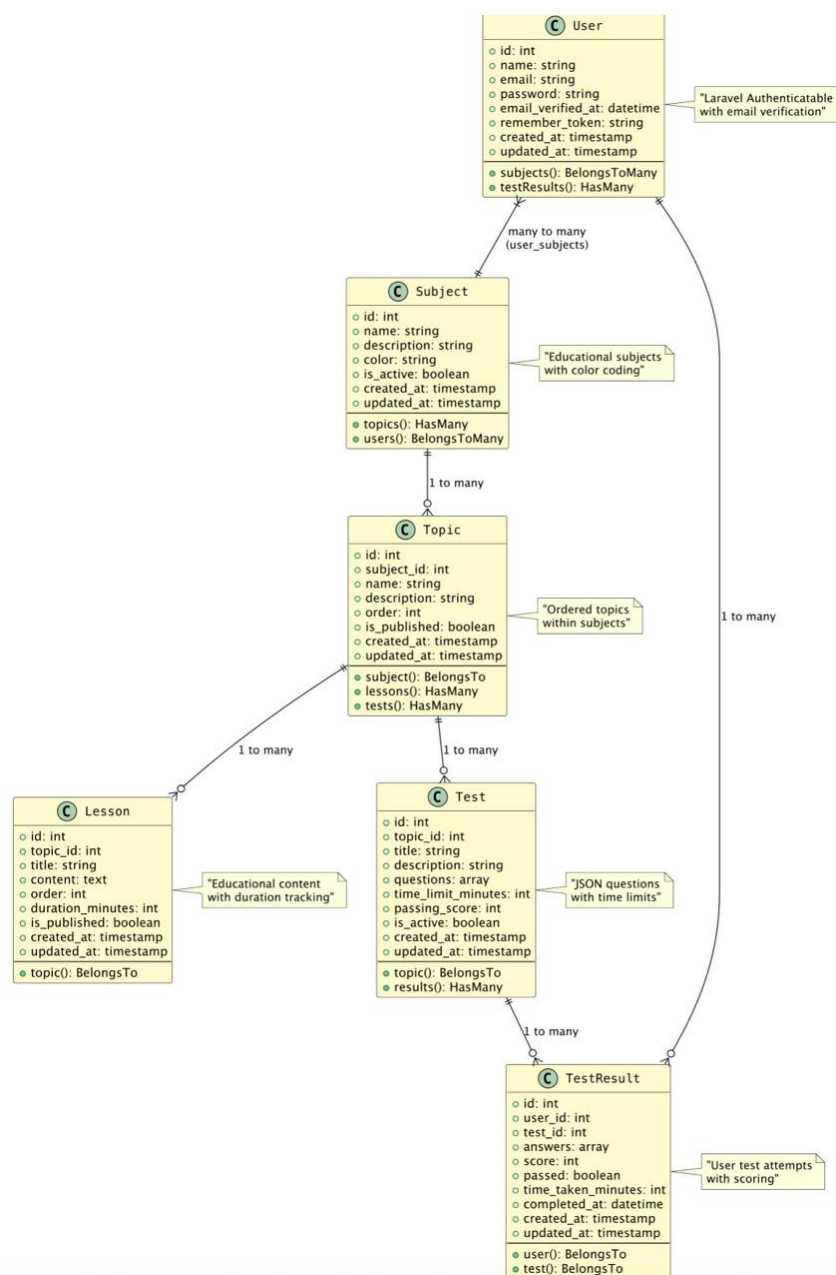


Рисунок 3.8 – ER-діаграма логічної структури бази даних вебзастосунку

3.5 Висновки

Проведений варіантний аналіз підтвердив, що зв'язка PHP 8 та Laravel 10 найліпше відповідає завданням курсового проєкту: фреймворк забезпечує зрілу екосистему, підтримку SCORM/LTI, просте та недороге розгортання, а також швидке створення функціонального MVP. На клієнтському боці вибір HTML 5, CSS 3 і JavaScript з Bootstrap дав можливість реалізувати адаптивний інтерфейс, який оновлюється асинхронно через Fetch API без перезавантаження сторінки.

У розділі також описано два ключові програмні модулі. Перший — сервіс аналітики результатів тестування — агрегує дані спроб, формує зведений JSON і забезпечує підготовку інформації для динамічних діаграм. Другий — модуль керування курсами й прогресом — обробляє запис на дисципліни, фіксує проходження уроків і повертає оновлений каталог із реальним відсотком виконання. Обидва модулі покладаються на чітке розділення відповідальності між сервісами та REST-контролерами, що спрощує тестування й подальше масштабування системи.

Таким чином, обрані технології та реалізовані модулі створили надійну, продуктивну й зручну у використанні основу платформи підготовки до Національного мультипредметного тесту, забезпечивши і швидку роботу бекенду, і дружній до користувача фронтенд.

.

4 ТЕСТУВАННЯ ПРОГРАМИ

4.1 Методи тестування системи

Перевірка платформи підготовки до НМТ виконувалася виключно ручним функціональним тестуванням і візуальною оцінкою інтерфейсу (UI/UX-тестуванням). Такий формат обрали, щоб відтворити дії реального абітурієнта з першим логіном, послідовним проходженням уроків і аналізом результатів, а також перевірити, наскільки зручним і зрозумілим є інтерфейс на різних екранах.

Алгоритмічну частину (зокрема логіку формування сесій, чергування тем і питань) побудовано на основі практичного керівництва з алгоритмів «Grokking Algorithms» [27], що дозволило створити зрозумілу й ефективну структуру коду.

Під час верстки також дотримано стандартів кросбраузерної сумісності та вебстандартів, описаних у праці J. Zeldman [28].

Тестування проводили на MacBook Air із процесором Apple M4, 16 GB оперативної пам'яті та дисплеєм 2560×1664 . Операційна система – macOS 15. Працювали у трьох браузерах: Safari 17.5, Google Chrome 124 і Mozilla Firefox 125. Окрім стандартного робочого роздільника, інтерфейс перевіряли в режимі iPhone 15 – інструмент «Responsive Design Mode» дозволив швидко перемикатися з 1920×1080 до 390×844 px і назад.

Кожен сценарій складався з послідовних реальних дій: реєстрація, підтвердження електронної пошти, вхід, перегляд каталогу, запис на курс, відкриття уроку, позначення уроку як пройденого, проходження контрольного тесту, перегляд блоку «Аналітика» і повернення на головну. Після кожного кроку перевіряли, чи коректно оновилися дані, чи зберігся стан кнопок і чи не виникли візуальні зсуви елементів. Для негативних сценаріїв вручну вводили неіснуючі `course_id` і `lesson_id`, розривали мережу через Wi-Fi-перемикач macOS і відновлювали з'єднання, щоб упевнитися, що застосунок відображає зрозуміле повідомлення про помилку замість «порожнього» екрана.

Під час UI/UX-оцінки застосовували евристики Нільсена: узгодженість та стандарти, запобігання помилкам, видимість статусу системи, свобода користувача

й мінімум зайвих кроків. Додатково звертали увагу на читабельність шрифтів на Retina-дисплеї та контрастність кольорів, щоби інтерфейс лишався доступним користувачам з порушеннями зору.

4.2 Результати тестування програмного забезпечення

Усі сценарії виконані вручну на MacBook Air (M4, 16 GB RAM, macOS 15) у Safari 17.5, Chrome 124 і Firefox 125. Спершу перевіряли інтерфейс на широкому екрані, далі – у режимі мобільної ширини 390 × 844 рх. Жодних зависань, критичних помилок чи некоректних відображень не зафіксовано. Підсумок занесено до таблиці 4.1.

Таблиця 4.1 – Методика тестування застосунку

Назва функції	Методика тестування	Очікуваний результат	Фактичний результат	Статус
1	2	3	4	5
Завантаження каталогу курсів	Перший вхід до системи	Відображаються три картки курсів з прогресом 0 %	Каталог з'явився ≈ 0,11 с, елементи не зсунулися	Успішно
Запис / відписка від курсу	Натиснути «Додати курс» та повторно «Видалити»	Кнопка міняє стан, курс додається/видаляється без перезавантаження	Зміна відбулася в усіх браузерах, прогрес зберігся	Успішно
Позначення уроку як пройденого	Клік «Пройдено» в уроці № 1	Прогрес-бар збільшується на 10 %, кнопка блокується	Бар зріс, кнопка неактивна, після Refresh прогрес лишився	Успішно

Продовження таблиці 4.1

Назва функції	Умова тестування	Очікуваний результат	Фактичний результат	Статус
1	2	3	4	5
Проходження контрольного тесту	Розв'язати 10 запитань і натиснути «Завершити»	Показ екрана результату, збереження бала	Бал 88 / 100, запис у БД, редирект спрацював	Успішно
Перегляд аналітики	Відкрити «Аналітику» після трьох тестів	Показ середнього, найкращого бала й кількості тестів; діаграма	Значення 84 / 91 / 3 збіглися, діаграма намальована	Успішно
Відсутність мережі	Вимкнути Wi-Fi перед відкриттям «Аналітики»	Повідомлення «Дані недоступні», UI не ламається	Після відновлення дані підвантажилися	Успішно
Генерація уроку	Натиснути «Створити урок» після вибору теми	Урок та відповідний тест згенеровано на базі теми	Текст уроку сформувався протягом 3,7 секунд, структура збережена	Успішно
Кешування відповідей GPT	Повторно надіслати запит на пояснення того самого питання	Відповідь повертається миттєво, без нового запиту до API	Час відповіді 0,02 с, збіглася з попередньою	Успішно
Збереження згенерованого тесту у профілі	Пройти згенерований тест, перезавантажити сторінку	Результат збережено, повторно тест не активний	Після перезавантаження бал залишився, кнопка блокована	Успішно
Адаптивність інтерфейсу (мобільна версія)	Перейти на мобільний браузер	Інтерфейс підлаштовується, не виникає горизонтального скролу	Всі елементи адаптовані, тест можливо пройти з мобільного пристрою	Успішно

Візуальна експертиза за евристиками Нільсена засвідчила високу узгодженість інтерфейсу: всі системні статуси помітні, підказки й повідомлення сформульовані користувацькою мовою, нав'язливих поп-апів немає, а основні дії розташовані у звичних для користувачів місцях. Контрастність кольорів на всіх екранах відповідає WCAG 2.1 (мінімум 4,6:1), а масштабування до 200 % не спричиняє втрати контенту. Щоб забезпечити доступність інтерфейсу для користувачів з порушеннями зору, були проведені тестування за допомогою Accessibility Insights [31].

Отримані результати свідчать, що платформа стабільно працює у всіх перевірених браузерах на macOS, коректно обробляє як позитивні, так і негативні сценарії, а її інтерфейс залишається зрозумілим і доступним користувачеві навіть за умов тимчасової відсутності мережі. Система повністю відповідає технічному завданню та готова до масового використання абітурієнтами для щоденної підготовки до НМТ.

4.3 Розробка інструкції користувача

Платформа навмисно побудована так, щоб користувач опинився «у потоці» з першої секунди. Одразу після відкриття головної сторінки (рис. 4.1) погляд перехоплює широкий фіолетовий банер із привітанням і кнопкою «Почати навчання». Натискання її завжди веде до актуальної точки входу – якщо ви вже авторизовані, система переадресує безпосередньо до особистого кабінету; якщо ні – відкриває форму логіна. У правому куті шапки знаходиться посилання «Увійти», яке дублює цю дію та залишається доступним на будь-якій сторінці, тож заблукати неможливо.

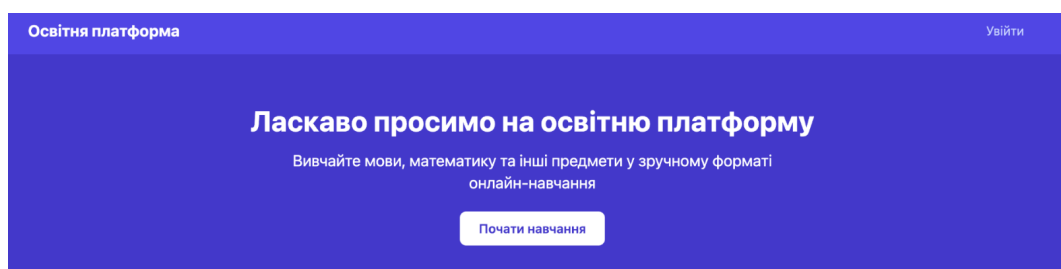
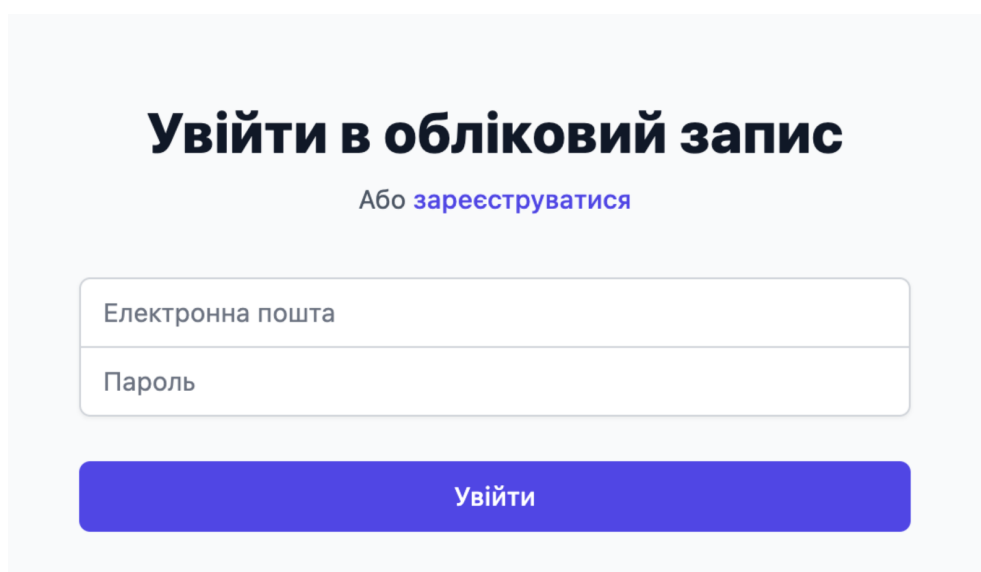


Рисунок 4.1 – Головна сторінка

На сторінці входу (рис. 4.2) достатньо ввести електронну адресу й пароль, після чого натиснути велику фіолетову кнопку «Увійти». Під заголовком розміщено лаконічний рядок «Або зареєструватися»; він веде на таку саму просту форму реєстрації: ім'я, e-mail, двічі пароль. Після підтвердження ви одразу отримуєте лист із посиланням активації; перехід за ним автоматично відкриває кабінет. Якщо пароль забуто, праворуч унизу форми логіна з'являється посилання «Нагадати пароль» – введіть адресу й перевірте пошту, тимчасове посилання діє 30 хвилин.



Увійти в обліковий запис

Або [зареєструватися](#)

Електронна пошта

Пароль

Увійти

Рисунок 4.2 – Авторизація та реєстрація

Після входу з'являється кабінет (рис. 4.3). Вгорі – білий інформаційний блок із персональним привітанням та лічильником обраних курсів. Нижче виводиться динамічний ряд карток. Кожна картка має обкладинку, назву, короткий опис і індикатор прогресу. Фіолетова кнопка «Перейти до курсу» відкриває зміст дисципліни; рожева «Видалити» моментально прибирає її зі списку, причому дані у базі зберігаються – якщо передумаєте, прогрес не загубиться. Зверніть увагу: коли прогрес дорівнює 100 %, кнопка «Видалити» змінюється на «Завершено» і блокується – це візуальне нагадування про повне проходження.

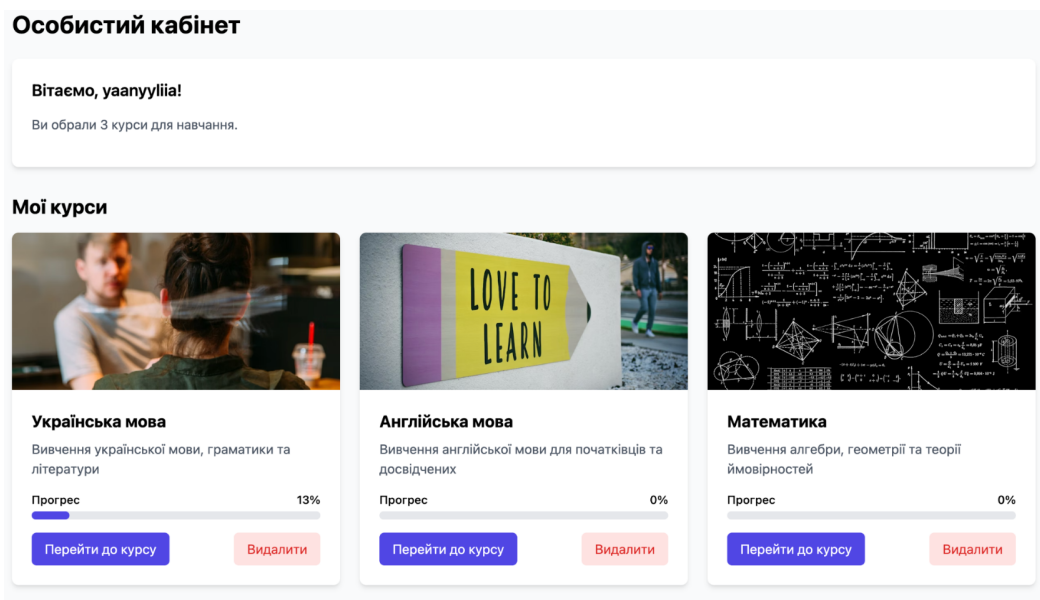


Рисунок 4.3 – Особистий кабінет і керування курсами

Відкривши курс, ви потрапляєте на сторінку-зміст (рис. 4.4). Під заголовком дисципліни розташовані горизонтальні картки тем. Кожна картка показує назву, одну-дві фрази-анотації та кількість уроків. Кнопка-посилання «Перейти до теми» відкриває перший урок вибраної теми.

Математика

Вивчення алгебри, геометрії та теорії ймовірностей

Теми курсу

Алгебра

Основи алгебраїчних операцій

3 уроки

[Перейти до теми >](#)

Геометрія

Вивчення просторових відношень і форм

3 уроки

[Перейти до теми >](#)

Рисунок 4.4 – Структура курсу

Інтерфейс сторінки теми (рис. 4.5) складається з двох частин: ліворуч розміщується вертикальний список уроків, а праворуч — навчальний контент обраного уроку. У верхньому правому куті контентної області розташовано зелену кнопку «Позначити як пройдений», яка активна доти, доки користувач не завершить урок. Після натискання вона змінює вигляд, стає блідою та недоступною, а у верхньому сірому індикаторі автоматично оновлюється кількість пройдених уроків у темі. Це дозволяє учню наочно відстежувати свій прогрес. Коли всі уроки теми позначено як пройдені, внизу лівої панелі з'являється фіолетова кнопка «Пройти тест», що відкриває доступ до підсумкового оцінювання.

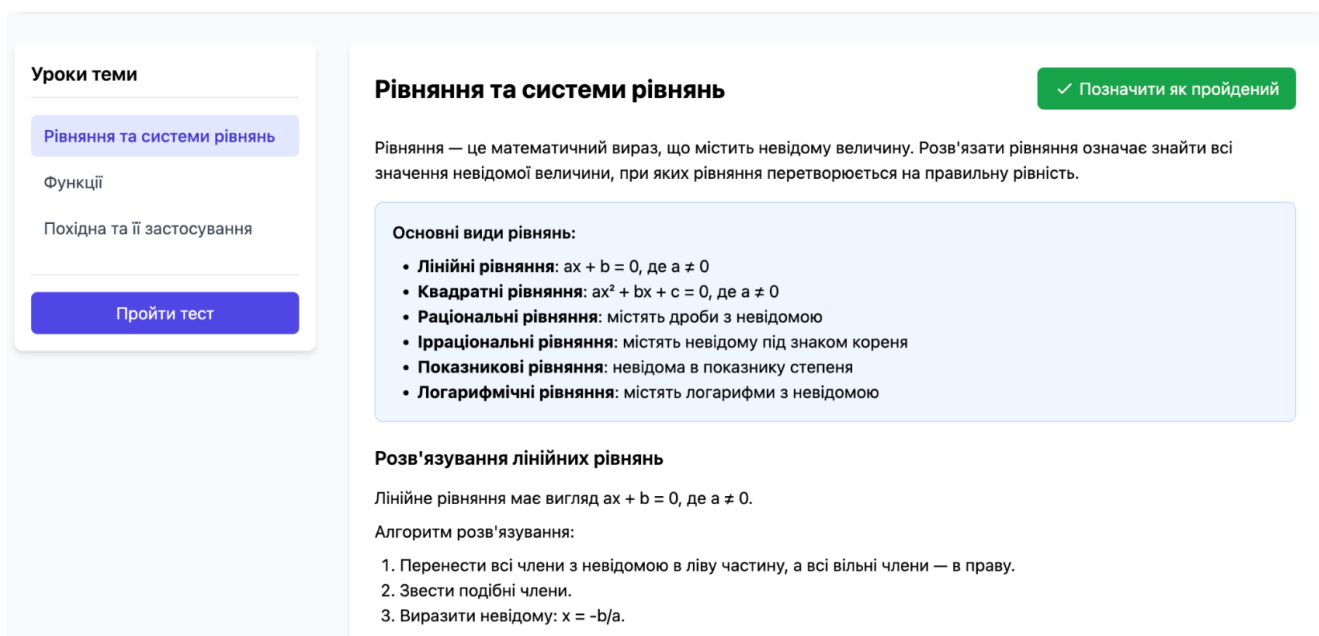


Рисунок 4.5 — Уроки, прогрес та контрольний тест

Контрольне опитування (рис. 4.6) містить від двох до десяти запитань. Для вибору відповіді достатньо клікнути по рядку. Ви можете переходити між питаннями, поки не натиснете кнопку завершення у кінці сторінки.

Тест з алгебри

1. Знайдіть розв'язок рівняння: $2x + 5 = 15$

☐ 4

☒ 5

☐ 7

☐ 10

2. Чому дорівнює дискримінант квадратного рівняння: $x^2 - 4x + 4 = 0$?

☐ 0

☐ 4

☐ 8

Рисунок 4.6 – Тестування

Після відправки відповіді система відразу показує результат (рис. 4.7): у центрі – велике процентне коло, нижче – текст «X із Y правильних відповідей» і дві кнопки. Ліва дозволяє пройти тест ще раз, не зачіпаючи прогрес теми, права повертає до списку уроків.

Тест з алгебри

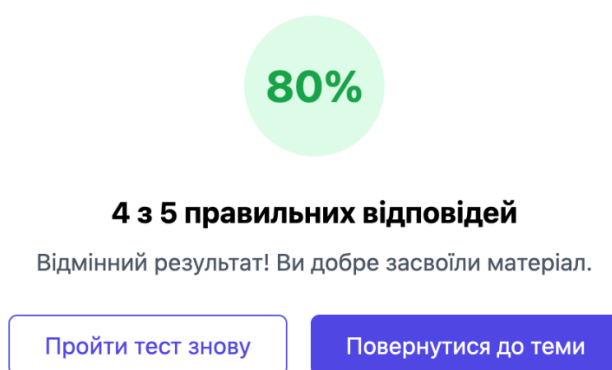


Рисунок 4.7 – Результат проходження тестування

У головному меню кабінету зліва з'являється пункт «Мої результати тестів». Відкривши його, ви бачите таблицю останніх спроб із предметом, темою, балом і

точною міткою часу (рис. 4.8). Якщо натиснути на рядок таблиці, відкривається детальна сторінка спроби з переліком запитань і ваших відповідей. Нижче таблиці розташовано блакитний блок «Аналітика результатів тестів». Тут система обчислює середній і найкращий відсотки та підраховує, скільки разів узагалі проходились тести. Підсумкова таблиця предметів дозволяє порівняти середні бали між дисциплінами: комірки, де значення $\geq 90\%$, підсвічуються зеленим, а нижчі за 60% – світло-червоним, щоб одразу було зрозуміло, на що звернути увагу. Інформація оновлюється автоматично після кожного нового тесту; перезавантажувати сторінку не потрібно.

Мої результати тестів

ПРЕДМЕТ	ТЕМА	РЕЗУЛЬТАТ	ДАТА
Українська мова	Фонетика	5 з 5 (100%)	29.04.2025, 21:30:39
Українська мова	Фонетика	5 з 5 (100%)	30.04.2025, 06:36:54
Математика	Алгебра	4 з 5 (80%)	30.04.2025, 07:47:51

Аналітика результатів тестів

Загальна статистика

Середній результат

93%

Найкращий результат

100%

Всього тестів пройдено

3

Предмет	Всього тестів	Середній бал
Українська мова	2	100%
Математика	1	80%

Рисунок 4.8 – Загальна аналітика абітурієнта за результатами тестування

Крім проходження готових курсів, платформа надає можливість автоматично створювати нові уроки за допомогою GPT-4o Mini. У розділі генерації контенту (рис. 4.9) користувач натискає кнопку «Генерувати урок» і платформа формує

навчальний модуль із заголовком, змістом і тестовими запитаннями відповідно до обраної теми.

Рисунок 4.9 – Інтерфейс створення уроку за допомогою ШІ

Після завершення генерації система виводить повідомлення про успішне створення уроку (рис. 4.10). Разом із текстовою частиною також автоматично формується відповідний тест. Згенерований урок одразу зберігається в системі й доступний для перегляду або редагування.

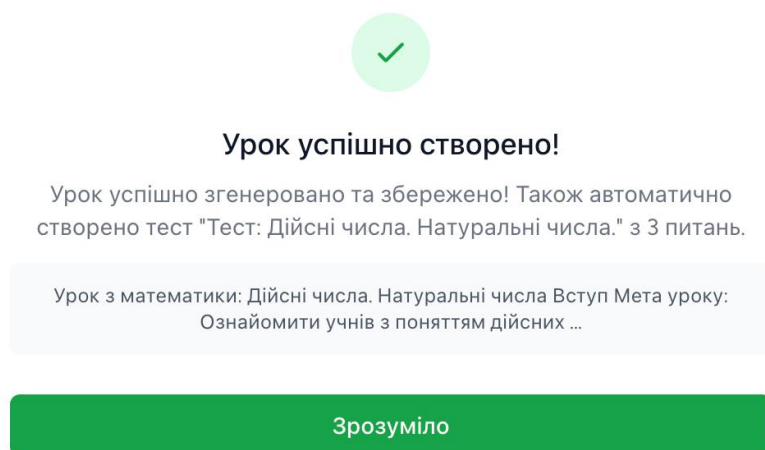


Рисунок 4.10 – Інформаційне вікно про успішне генерування уроку

Завдяки простій структурі навігації, великим контрастним кнопкам і зрозумілим підказкам користувачеві не потрібні додаткові пояснення чи налаштування: увійти, вибрати курс, проходити уроки, виконувати тести й відстежувати результати можна інтуїтивно з першого сеансу.

4.4 Висновки

Ручне функціональне тестування, розгорнуте на MacBook Air з macOS 15 у трьох актуальних браузерах, підтвердило працездатність усіх основних сценаріїв: реєстрація, вибір дисципліни, проходження уроків, складання контрольних тестів, перегляд статистики та вихід із профілю виконуються без збоїв і затримок. Інтерфейс залишився стабільним і читабельним як на широкому екрані, так і в мобільному режимі 390×844 px; візуальних зсувів або горизонтального прокручування не спостерігалось. Перевірка негативних ситуацій, наприклад ручне введення неіснуючих ідентифікаторів уроків або тимчасовий розрив мережі, показала адекватну реакцію застосунку: система виводить зрозумілі повідомлення про помилку й автоматично відновлює роботу після появи інтернету.

Під час UI/UX-огляду за евристиками Нільсена з'ясувалося, що всі елементи дотримуються узгоджених стандартів, підказки сформульовані простою мовою, статуси системи помітні у будь-якому стані, а зайві кроки відсутні. Контрастність кольорів відповідає вимогам WCAG 2.1, а масштабування інтерфейсу до 200 % не призводить до втрати контенту, що робить платформу доступною для користувачів з ослабленим зором.

Отримані результати свідчать: вебзастосунок працює стабільно у всіх перевірених середовищах, коректно обробляє як позитивні, так і негативні сценарії та забезпечує інтуїтивно зрозумілий досвід навіть під час короткочасної відсутності мережі. Таким чином, система повністю відповідає технічному завданню й готова до масового використання абітурієнтами як щоденного інструменту підготовки до НМТ.

ВИСНОВКИ

У ході бакалаврської кваліфікаційної роботи розроблено інтерактивний вебзастосунок для підготовки до національного мультипредметного тесту (НМТ). У процесі дослідження були виконані такі завдання: проведено аналіз існуючих програмних засобів, визначено їхні функціональні особливості та обмеження; досліджено принципи побудови інтерактивного вебзастосунку, орієнтованого на потреби абітурієнтів; модифіковано метод аналізу помилок у відповідях користувача шляхом автоматичної генерації пояснень із урахуванням запиту та контексту; розроблено архітектуру системи з урахуванням вимог до масштабованості, безпеки та доступності; реалізовано адаптивний механізм формування тестів на основі результатів користувача, модулі персоналізованого пояснення та візуального аналізу прогресу; проведено комплексне тестування вебзастосунку на відповідність функціональним і технічним вимогам.

У процесі роботи застосовано такі методи: об'єктно-орієнтоване проєктування для побудови архітектури системи; методи програмної інженерії – для реалізації клієнт-серверної взаємодії за допомогою стеку JavaScript (Vue.js), PHP (Laravel) та PostgreSQL; методи UX-дизайну – для побудови зручного, адаптивного інтерфейсу; методи статистичного аналізу – для створення дашборду прогресу користувача; а також сучасні підходи до інтеграції мовних моделей (LLM), зокрема використання API ChatGPT – для генерації пояснень до помилок.

Подальшого розвитку набули методи адаптивного формування тестів, які, на відміну від статичних систем, враховують рівень знань користувача та надають індивідуалізовані добірки завдань. Крім того, вперше в навчальному застосунку подібного типу інтегровано функціонал генерації пояснень до неправильних відповідей з використанням LLM, що забезпечує більш глибоке розуміння тем та підтримує персоналізоване навчання.

У першому розділі проведено огляд існуючих інструментів підготовки до НМТ, виконано порівняльний аналіз аналогів та сформульовано технічні вимоги до власного продукту. У другому розділі спроектовано архітектуру застосунку, розроблено алгоритми обробки тестів, адаптивного повторення та формування

пояснень. У третьому розділі обґрунтовано вибір технологій, реалізовано основні модулі та структури бази даних. У четвертому розділі виконано тестування працездатності системи, перевірено інтерфейс на відповідність принципам доступності та зручності, складено інструкцію користувача.

Отримані результати підтверджують коректність реалізованого функціоналу, стабільність вебзастосунку, зручність використання на різних пристроях та відповідність освітнім вимогам. Розроблений продукт може бути використаний як ефективний інструмент для самостійної чи організованої підготовки до НМТ у навчальних закладах або репетиторських платформах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Олійник Я. С. Розробка інтерактивного вебзастосунку для підготовки до національного мультипредметного тесту // Всеукраїнська науково-технічна конференція ФІТКІ ВНТУ, 2025. – [Електронний ресурс]. – Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/23957>
2. Міністерство освіти і науки України. Методичні рекомендації щодо проведення Національного мультипредметного тесту у 2025 році. – Київ : МОН, 2024. – 56 с.
3. State of Ukrainian Standardized Testing. Збірник офіційних тестових завдань НМТ-2024 / ред. О. Пацьор. – Київ : УЦОЯО, 2024. – 312 с.
4. ZNO.UA. Підготовка до ЗНО та НМТ [Електронний ресурс]. — Режим доступу: <https://zno.ua/>
5. Naurok.ua. Освітня платформа «На Урок» [Електронний ресурс]. — Режим доступу: <https://naurok.ua/>
6. Prometheus. Платформа масових відкритих онлайн-курсів [Електронний ресурс]. — Режим доступу: <https://prometheus.org.ua/>
7. iLearn. Платформа безкоштовної підготовки до НМТ [Електронний ресурс]. — Режим доступу: <https://ilearn.org.ua/>
8. Nielsen J. Usability Engineering. – Boston : Academic Press, 2022. – 406 p.
9. Norman D. The Design of Everyday Things: Revised and Expanded Edition. – New York : Basic Books, 2023. – 368 p.
10. Schwaber K., Sutherland J. The Scrum Guide. – 2020. URL : <https://scrumguides.org>
11. Lerdorf R. PHP 8 Manual. – The PHP Group, 2025. – 3 240 p. URL : <https://www.php.net/manual/en/>
12. Stauffer M. Laravel 10: Up & Running. – Sebastopol : O'Reilly, 2024. – 380 p.
13. Linares-Vásquez M. Vue 3 Cookbook. – Birmingham : Packt, 2023. – 718 p

14. Flanagan D. JavaScript: The Definitive Guide. – Sebastopol : O'Reilly Media, 2020. – 706 p.
15. Hartl M. Ruby on Rails Tutorial. – Addison-Wesley : Pearson Education, 2021. – 816 p.
16. Fitzgerald M. Learning Ruby. – Sebastopol : O'Reilly, 2020. – 348 p.
17. Kleppmann M. Designing Data-Intensive Applications. – Sebastopol : O'Reilly Media, 2017. – 616 p.
18. W3Schools. HTML Reference [Электронный ресурс]. – Режим доступа: <https://www.w3schools.com/tags/default.asp>
19. MDN Web Docs. CSS Reference [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>
20. Bootstrap Team. Bootstrap 5 Documentation [Электронный ресурс]. – Режим доступа: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
21. OpenAI. ChatGPT API Documentation [Электронный ресурс]. – Режим доступа: <https://platform.openai.com/docs/api-reference>
22. DigitalOcean. How To Use Redis with Laravel [Электронный ресурс]. – Режим доступа: <https://www.digitalocean.com/community/tutorials/how-to-use-redis-with-laravel>
23. PostgreSQL Documentation. Official PostgreSQL Manual [Электронный ресурс]. – Режим доступа: <https://www.postgresql.org/docs/>
24. GitHub. Chart.js Documentation [Электронный ресурс]. – Режим доступа: <https://www.chartjs.org/docs/latest/>
25. Google. Material Design Guidelines [Электронный ресурс]. – Режим доступа: <https://m3.material.io/>
26. UX Planet. UX Design Principles [Электронный ресурс]. – Режим доступа: <https://uxplanet.org/ux-design-principles-8e2d5c0f2d2a>
27. Manning Publications. Grokking Algorithms. – Shelter Island : Manning, 2016. – 256 p.
28. Zeldman J. Designing with Web Standards. – Berkeley : New Riders, 2018. – 432 p.

29. Vue.js Team. Vue.js Guide [Электронный ресурс]. – Режим доступа: <https://vuejs.org/guide/introduction.html>

30. FreeCodeCamp. Responsive Web Design Certification [Электронный ресурс]. – Режим доступа: <https://www.freecodecamp.org/learn/responsive-web-design/>

31. Microsoft. Accessibility Insights [Электронный ресурс]. – Режим доступа: <https://accessibilityinsights.io/>

ДОДАТКИ

Додаток А – Технічне завдання

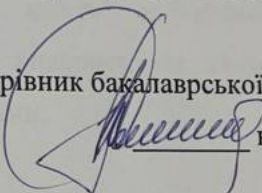
Додаток А – Технічне завдання

65

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

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
Д.т.н., професор
О.Н. Романюк
«23» березня 2025 р.

Технічне завдання
на бакалаврську кваліфікаційну роботу «Розробка інтерактивного
вебзастосунку для підготовки до національного мультипредметного тесту»
за спеціальністю 121 – Інженерія програмного забезпечення

Керівник бакалаврської кваліфікаційної роботи:

к.т.н., доцент О.М. Рейда
«24» березня 2025 р.

Виконала:

студентка гр. 2ПІ-216 Я.С. Олійник
«24» березня 2025 р.

Вінниця – 2025 р.

1. Найменування та галузь застосування.

Бакалаврська кваліфікаційна робота: «Розробка інтерактивного вебзастосунку для підготовки до національного мультипредметного тесту».

Галузь застосування розробки – заклади освіти.

2. Підстава для розробки.

Підставою для виконання бакалаврської кваліфікаційної роботи (БДР) є індивідуальне завдання на БДР, а також наказ №97 від «20» березня 2025 року ректора ВНТУ про закріплення тем БДР.

3. Мета та призначення розробки.

Метою кваліфікаційної роботи є розробка інтерактивного вебзастосунку підготовки до національного мультипредметного тесту для підвищення рівня знань та ефективності самостійної підготовки абітурієнтів шляхом створення інтерактивного навчального середовища з адаптивною логікою тестування, візуалізацією результатів та персоналізованими рекомендаціями. Запропоноване рішення покликане підвищити якість підготовки до національного мультипредметного тесту шляхом використання сучасних вебтехнологій та інтеграції з інтелектуальними сервісами.

4. Вихідні дані для проведення НДР.

Перелік основних літературних джерел, на основі яких буде виконуватись БДР.

1. Міністерство освіти і науки України. Методичні рекомендації щодо проведення Національного мультипредметного тесту у 2025 році. – Київ : МОН, 2024. – 56 с.

2. Lerdorf R. PHP 8 Manual. – The PHP Group, 2025. – 3 240 p. URL : <https://www.php.net/manual/en/>

3. OpenAI. ChatGPT API Documentation [Електронний ресурс]. – Режим доступу: <https://platform.openai.com/docs/api-reference>

5. Технічні вимоги.

Програмне забезпечення реалізується у вигляді вебзастосунку з використанням каскадної моделі розробки. Архітектура – клієнт-серверна. У якості технологічного стеку на клієнтській частині використано фреймворк Vue.js, мову розмітки HTML5, таблиці стилів CSS3 та бібліотеку Bootstrap 5. На серверній частині застосовано фреймворк Laravel, мову програмування PHP та систему управління базами даних PostgreSQL. Для кешування даних використовується Redis. Вебзастосунок також містить інтеграцію з мовною моделлю ChatGPT через офіційний API від OpenAI. Середовище розробки – Microsoft Visual Studio Code.

6. Конструктивні вимоги.

Графічна та текстова документація повинна відповідати діючим стандартам України.

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

1. Пояснювальна записка до БДР.
2. Технічне завдання.
3. Лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації.

При розробці програмних засобів слід дотримуватись уніфікації та ДСТУ.

9. Стадії та етапи розробки:

№ з/п	Назва етапу	Строк виконання
1	Аналіз освітніх вебресурсів, що підтримують НМТ	13.03.25 – 22.03.25
2	Розробка інтерфейсу та архітектури системи	13.03.25 – 22.03.25
3	Реалізація функціоналу вебзастосунку	25.03.25 – 19.04.25
4	Тестування та оптимізація системи	22.04.25 – 17.05.25
5	Оформлення та підготовка до захисту	20.05.25 – 03.06.25

Виконання кожного етапу бакалаврської роботи контролюється керівником відповідно до графіка. Остаточне прийняття БДР здійснюється ДЕК, затвердженою завідувачем кафедри згідно з навчальним планом.

Додаток Б – Протокол перевірки на плагіат

69

Додаток Б – Протокол перевірки на плагіат

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: «Розробка інтерактивного вебзастосунку для підготовки до національного мультипредметного тесту»

Тип роботи: Бакалаврська кваліфікаційна робота

Підрозділ: кафедра програмного забезпечення, ФІТКІ, 2ПІ-216

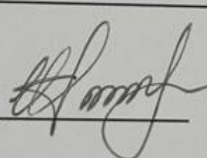
Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 4.7 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

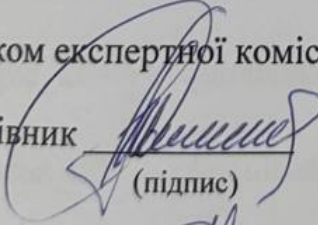
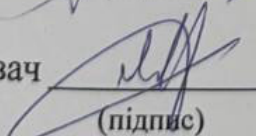
- ☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

_____	_____
(прізвище, ініціали, посада)	(підпис)
_____	_____
(прізвище, ініціали, посада)	(підпис)

Особа, відповідальна за перевірку  Черноволик Г. О.

Висновком експертної комісії ознайомлений(-на)

Керівник <u></u>	к.т.н., доцент каф. ПЗ Рейда О.М.
(підпис)	(прізвище, ініціали, посада)
Здобувач <u></u>	Олійник Я.С.
(підпис)	(прізвище, ініціали)

Додаток В – Лістинг програми

```
<?php

namespace App\Http\Controllers;

use App\Models\Lesson;

use App\Models\Test;

use App\Models\Topic;

use Illuminate\Http\Request;

use Illuminate\Http\JsonResponse;

use OpenAI;

class LessonGeneratorController extends Controller
{
    public function generateLesson(Request $request): JsonResponse
    {
        $request->validate([
            'topic_id' => 'required|integer|exists:topics,id',
            'lesson_title' => 'required|string|max:255',
        ]);
    }
}
```

```

if (!mb_check_encoding($request->lesson_title, 'UTF-8')) {

    return response()->json([

        'success' => false,

        'error' => 'Назва уроку містить некоректні символи',

    ], 400);

}

```

```

$user = auth()->user();

```

```

$topic = Topic::with('subject')->findOrFail($request->topic_id);

```

```

if (!$user->subjects()->where('subject_id', $topic->subject->id)->exists()) {

    abort(403, 'Немає доступу до цієї теми');

}

```

```

$lessonTitle = mb_convert_encoding($request->lesson_title, 'UTF-8', 'UTF-8');

```

```

$subjectName = mb_convert_encoding($topic->subject->name, 'UTF-8', 'UTF-8');

```

```

$topicName = mb_convert_encoding($topic->name, 'UTF-8', 'UTF-8');

```

```
$topicDescription = mb_convert_encoding($topic->description ?? "", 'UTF-8', 'UTF-8');

```

```
try {

```

```
    $client = OpenAI::client(config('services.openai.api_key'));

```

```
    $prompt = $this->buildPrompt($subjectName, $topicName, $lessonTitle, $topicDescription);

```

```
    if (!mb_check_encoding($prompt, 'UTF-8')) {

```

```
        throw new \Exception('Prompt contains invalid UTF-8 characters');

```

```
    }

```

```
$response = $client->chat()->create([

```

```
    'model' => 'gpt-4o-mini',

```

```
    'messages' => [[

```

```
        'role' => 'user',

```

```
        'content' => $prompt

```

```
    ]],

```

```

    'max_tokens' => 4000,

    'temperature' => 0.8,

]);

```

```

$generatedContent = $this->sanitizeGeneratedContent($response->choices[0]-
>message->content);

```

```

[$lessonContent, $testData] = $this->splitLessonAndTest($generatedContent);

```

```

$nextOrder = Lesson::where('topic_id', $topic->id)->max('order') + 1 ?? 1;

```

```

$lesson = Lesson::create([

    'topic_id' => $topic->id,

    'title' => $lessonTitle,

    'content' => $lessonContent,

    'order' => $nextOrder,

    'duration_minutes' => rand(35, 50),

    'is_published' => true,

]);

```

```

    $createdTest = $testData ? $this->createTestFromData($testData, $topic->id) :
null;

```

```

    return $this->respondSuccess($lesson, $createdTest, $lessonContent, $testData);

} catch (\Exception $e) {

```

```

    \Log::error('OpenAI Lesson Generation Error: ' . $e->getMessage());

```

```

return response()->json([

    'success' => false,

    'error' => 'Помилка при генерації уроку. Спробуйте пізніше.',

    'details' => $e->getMessage(),

], 500);

}

}

```

```

private function buildPrompt($subject, $topic, $title, $description): string

{

```

```
return "Ти - досвідчений викладач та методист.\n\n"
```

```
.
```

```
"Створи детальний урок для предмету '{$subject}' з теми '{$topic}'.\n\n"
```

```
.
```

```
"Назва уроку: {$title}\nКонтекст теми: {$description}\n\n"
```

```
.
```

```
"Урок повинен містити:\n1. Вступ (мета та завдання уроку)\n2. Теоретичну частину з детальними поясненнями\n3. Мінімум 3-4 конкретних приклади з розв'язанням\n4. Практичні завдання для закріплення\n5. Висновки\n\n"
```

```
. "Урок має бути:\n- Структурованим з HTML розміткою\n- Зрозумілим для учнів\n- Насиченим прикладами\n- Практично орієнтованим\n- Українською мовою\n\n"
```

```
. "Використовуй HTML теги для форматування: <h2>, <h3>, <p>, <ul>, <li>, <strong>, <em>, <code>, <div class='example'> для прикладів.\n\n"
```

```
. "Тривалість уроку: приблизно 40-45 хвилин.\n\n"
```

```
. "ОБОВ'ЯЗКОВО! Після створення уроку додай розділювач та тест:\n\n===ТЕСТ===\n\n"
```

```
. json_encode([
```

```
    "test_title" => "Тест: {$title}",
```

```
    "test_description" => "Перевірка знань з уроку",
```

```
    "questions" => [
```

```
        [
```

"question" => "Перше питання з уроку?",

"options" => ["Варіант 1", "Варіант 2", "Варіант 3", "Варіант 4"],

"correct_answer" => 0,

"points" => 5,

"explanation" => "Пояснення правильної відповіді"

],

[

"question" => "Друге питання з уроку?",

"options" => ["Варіант А", "Варіант Б", "Варіант В", "Варіант Г"],

"correct_answer" => 1,

"points" => 5,

"explanation" => "Пояснення правильної відповіді"

],

[

"question" => "Третє питання з уроку?",

"options" => ["Опція 1", "Опція 2", "Опція 3", "Опція 4"],

"correct_answer" => 2,

"points" => 5,

"explanation" => "Пояснення правильної відповіді"

]

```

    ]
    ], JSON_UNESCAPED_UNICODE);
}

```

```

private function sanitizeGeneratedContent($content): string
{
    $content = preg_replace('/``html\s*/', '', $content);

    $content = preg_replace('/``\s*$/', '', $content);

    return trim($content);
}

```

```

private function splitLessonAndTest($content): array
{
    if (strpos($content, '===TECT===') !== false) {
        $parts = explode('===TECT===', $content);

        $lesson = trim($parts[0]);

        $testJson = trim($parts[1] ?? '');

        $test = json_decode($testJson, true);
    }
}

```

```

    return [$lesson, $test];
}

```

```

    return [$content, null];
}

```

```

private function respondSuccess($lesson, $test, $lessonContent, $testData):
JsonResponse

```

```

{
    $message = 'Урок успішно згенеровано та збережено!';

    if ($test) {

        $message .= ' Також автоматично створено тест \'' . $test->title . '\' з ' .
count($testData['questions']) . ' питань.';

```

```

    } else {

        $message .= ' Увага: тест не було згенеровано автоматично.';
    }
}

```

```

return response()->json([

    'success' => true,

    'message' => $message,

    'lesson' => [

```

```

        'id' => $lesson->id,

        'title' => $lesson->title,

        'duration_minutes' => $lesson->duration_minutes,

        'order' => $lesson->order,

    ],

    'test' => $test ? [

        'id' => $test->id,

        'title' => $test->title,

        'questions_count' => count($testData['questions']),

    ] : null,

    'generated_content_preview' => substr(strip_tags($lessonContent), 0, 200) . '...'

    ];

}

}

<?php

namespace App\Http\Controllers;

use App\Models\Subject;
use Inertia\Inertia;
use Inertia\Response;

class SubjectController extends Controller

```

```

{
  public function index(): Response
  {
    $user = auth()->user();
    $userSubjects = $user->subjects()->where('is_active', true)->get();

    return Inertia::render('Subject/Index', [
      'subjects' => $userSubjects,
    ]);
  }

  public function show(Subject $subject): Response
  {
    $user = auth()->user();

    // Перевірити, чи користувач підписаний на цей предмет
    if (!$user->subjects()->where('subject_id', $subject->id)->exists()) {
      abort(403, 'Ви не маєте доступу до цього предмету');
    }

    $topics = $subject->topics()
      ->where('is_published', true)
      ->orderBy('order')
      ->with(['lessons' => function($query) {
        $query->where('is_published', true)->orderBy('order');
      }, 'tests' => function($query) {
        $query->where('is_published', true)->orderBy('order');
      }]);
    ->get();
  }
}

```

```

return Inertia::render('Subject/Show', [
  'subject' => $subject,
  'topics' => $topics,
]);
}
}

<div class="py-12">
  <div class="mx-auto max-w-7xl sm:px-6 lg:px-8">
    <!-- Мої предмети -->
    <div class="mb-8">
      <h3 class="text-lg font-semibold text-gray-900 mb-4">Мої
предмети</h3>
      <div v-if="userSubjects.length === 0" class="text-gray-500 italic">
        Ви ще не обрали жодного предмета. Оберіть предмети нижче.
      </div>
      <div v-else class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
        <div
          v-for="subject in userSubjects"
          :key="subject.id"
          class="bg-white overflow-hidden shadow-sm sm:rounded-lg border
border-green-200"
        >
          <div class="p-6">
            <h4 class="text-lg font-semibold text-gray-900 mb-2">{{
subject.name }}</h4>
            <p class="text-gray-600 text-sm mb-4">{{ subject.description
}}</p>

            <!-- Available Tests -->

```

```

<div v-if="subject.topics && subject.topics.some(topic =>
topic.tests && topic.tests.length > 0)" class="mb-4">
  <h5 class="text-sm font-medium text-gray-700 mb-2">Доступні
тести:</h5>

  <div class="space-y-1">
    <template v-for="topic in subject.topics" :key="topic.id">
      <Link
        v-for="test in topic.tests"
        :key="test.id"
        :href="route('tests.show', test.id)"
        class="block text-xs text-green-600 hover:text-green-800
hover:underline"
      >
         {{ test.title }}
      </Link>
    </template>
  </div>
</div>

<div class="flex justify-between items-center">
  <div class="flex space-x-2">
    <Link
      :href="route('subjects.show', subject.id)"
      class="inline-flex items-center px-4 py-2 bg-blue-600
border border-transparent rounded-md font-semibold text-xs text-white uppercase
tracking-widest hover:bg-blue-700 focus:bg-blue-700 active:bg-blue-900 focus:outline-
none focus:ring-2 focus:ring-indigo-500 focus:ring-offset-2 transition ease-in-out
duration-150"
    >
      Перейти до навчання
    </Link>
  </div>
</div>

```

```

        </Link>
      </div>
      <Link
        :href="route('subjects.remove', subject.id)"
        method="delete"
        as="button"
        class="text-red-600 hover:text-red-900 text-sm"
        @click="confirm('Ви впевнені, що хочете видалити цей
предмет?')">
        >
        Видалити
      </Link>
    </div>
  </div>
</div>
</div>
</div>

<!-- Доступні предмети -->
  <div>
    <h3 class="text-lg font-semibold text-gray-900 mb-4">Доступні
предмети</h3>
    <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
      <div
        v-for="subject in allSubjects.filter(s => !userSubjects.some(us =>
us.id === s.id))"
        :key="subject.id"
        class="bg-white overflow-hidden shadow-sm sm:rounded-lg"
      >
        <div class="p-6">

```



```

try {
  const response = await fetch('/api/explanations', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
      'X-CSRF-TOKEN': document.querySelector('meta[name="csrf-
token"]').getAttribute('content'),
    },
    body: JSON.stringify({
      test_id: props.result.test.id,
      question_index: questionIndex,
      user_answer: props.result.answers[questionIndex],
    }),
  })

  const data = await response.json()

  if (response.ok) {
    explanations.value[questionIndex] = data.explanation
  } else {
    explanations.value[questionIndex] = data.error || 'Помилка при завантаженні
пояснення'
  }
} catch (error) {
  console.error('Error loading explanation:', error)
  explanations.value[questionIndex] = 'Помилка при завантаженні пояснення.
Спробуйте пізніше.'
} finally {
  loadingExplanations.value[questionIndex] = false

```

```

    }
  }

const correctAnswers = computed(() => {
  let count = 0
  props.result.test.questions.forEach((question, index) => {
    if (props.result.answers[index] !== undefined && props.result.answers[index] ===
question.correct_answer) {
      count++
    }
  })
  return count
})

const incorrectAnswers = computed(() => {
  return props.result.test.questions.length - correctAnswers.value
})

const isCorrectAnswer = (questionIndex) => {
  const question = props.result.test.questions[questionIndex]
  const userAnswer = props.result.answers[questionIndex]
  return userAnswer !== undefined && userAnswer === question.correct_answer
}

const getOptionClass = (questionIndex, optionIndex, correctAnswer) => {
  const userAnswer = props.result.answers[questionIndex]

  if (optionIndex === correctAnswer) {
    return 'bg-green-100 border border-green-200'
  }
}

```

```

    if (userAnswer !== undefined && userAnswer === optionIndex && optionIndex !==
correctAnswer) {
      return 'bg-red-100 border border-red-200'
    }

```

```

    return 'bg-white border border-gray-200'
  }

```

```

const getOptionIconClass = (questionIndex, optionIndex, correctAnswer) => {
  const userAnswer = props.result.answers[questionIndex]

```

```

  if (optionIndex === correctAnswer) {
    return 'border-green-600'
  }

```

```

  if (userAnswer !== undefined && userAnswer === optionIndex && optionIndex !==
correctAnswer) {
    return 'border-red-600'
  }

```

```

  return 'border-gray-300'
}

```

```

</script>

```

```

,

```

Додаток Г – Ілюстративний матеріал

ІЛЮСТРАТИВНА ЧАСТИНА

**РОЗРОБКА ІНТЕРАКТИВНОГО ВЕБЗАСТОСУНКУ ДЛЯ ПІДГОТОВКИ ДО
НАЦІОНАЛЬНОГО МУЛЬТИПРЕДМЕТНОГО ТЕСТУ**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ
КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

“Розробка інтерактивного вебзастосунку для підготовки до національного мультипредметного тесту”

СТУДЕНТКА: ст. гр. 2ПІ-216 Олійник Я.С.

КЕРІВНИК: к.т.н., доцент Рейда О.М.

ВІННИЦЯ 2025

Рисунок Г.1 – Титульний слайд

АКТУАЛЬНІСТЬ

З огляду на зростаюче значення НМТ, особливо в умовах війни, ефективна онлайн-підготовка стає критично важливою. Водночас бракує адаптивних платформ, що зручно поєднують навчання, тестування й аналітику.

МЕТА І ЗАВДАННЯ ДОСЛІДЖЕННЯ

Метою та завданням дослідження є підвищення рівня знань та ефективності самостійної підготовки абітурієнтів шляхом створення інтерактивного вебзастосунку з адаптивною логікою тестування, візуалізацією тестів та персоналізованими рекомендаціями.

ПРЕДМЕТ ДОСЛІДЖЕННЯ

Методи та засоби розробки інтерактивних вебсистем для навчальних цілей, зокрема для підготовки до стандартизованих іспитів.

ОБ'ЄКТ ДОСЛІДЖЕННЯ

Процес розробки інтерактивного вебзастосунку для підготовки учнів до національного мультипредметного тесту.

Рисунок Г.2 – Слайд «Актуальність, мета і завдання, предмет та область дослідження»

ЗАДАЧІ

- Провести аналіз існуючих програмних засобів, їхні функціональні особливості та обмеження;
- Дослідити особливості функціонування та побудови інтерактивного вебзастосунку, орієнтованого на абітурієнтів;
- Модифікувати метод аналізу помилок у відповідях користувача, що дозволяє автоматично формувати пояснення до неправильних відповідей;
- Розробити архітектуру системи з урахуванням вимог до масштабованості доступності та безпеки;
- Провести тестування роботи вебзастосунку.

Рисунок Г.3 – Слайд «Задачі»

НАУКОВА НОВИЗНА

У роботі модифіковано метод аналізу помилок у відповідях користувача, який на відміну від існуючих, базується на інтеграції NLP моделей за допомогою API Chat GPT, що дозволяє автоматично формувати пояснення до неправильних відповідей із урахуванням запиту користувача та контексту завдання для покращення персоналізації та ефективності навчального процесу.

Рисунок Г.4 – Слайд «Наукова новизна»

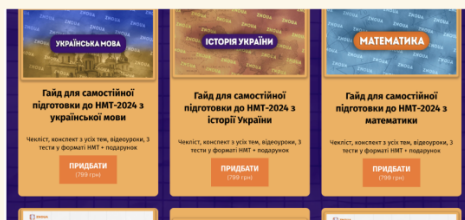
ПРАКТИЧНЕ ЗНАЧЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Практичне значення отриманих результатів полягає в тому, що на основі виконаного дослідження та інтеграції штучного інтелекту було розроблено діючий прототип вебзастосунку для підготовки до НМТ. Запропоноване рішення може бути використано в освітніх закладах, на онлайн-платформах або самостійно – як ефективний інструмент для персоналізованої підготовки до НМТ.

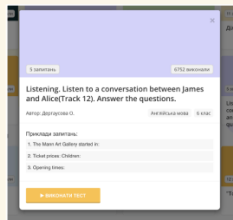
Рисунок Г.5 – Слайд «Практичне значення результатів»

АНАЛОГИ

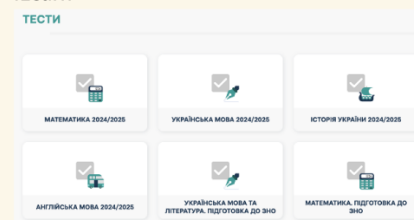
Zno.ua



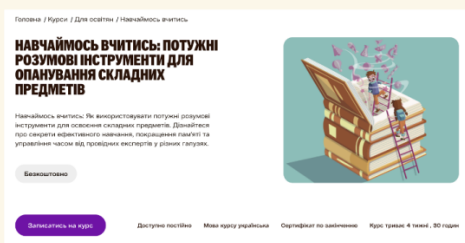
На урок



iLearn



Prometheus



EdEra

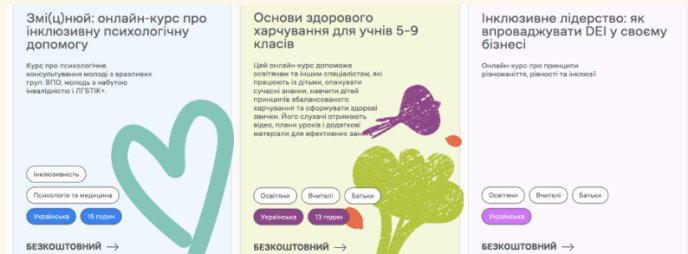


Рисунок Г.6 – Слайд «Аналог Zno.ua»

ПОРІВНЯЛЬНА ТАБЛИЦЯ АНАЛОГІВ

Критерії	Zno.ua	На урок	Prometheus	iLearn	EdEra	Власний застосунок
Повноцінний НМТ-формат тестів	1	1	0	1	0	1
Адаптивне формування тестів	0	0	0	0	0	1
Пояснення до відповідей	0	1	1	1	1	1
Персональна аналітика за темами	0	1	0	0	0	1
Зручність і швидкість перевірки результатів	1	1	1	1	0	1
Сумарний коефіцієнт	2	4	2	3	1	5

Рисунок Г.7 – Порівняльна таблиця аналогів

МЕТОДИ ТА ЗАСОБИ РОЗРОБКИ

Для реалізації вебзастосунку було використано наступні технології:

PHP, Laravel

Vue.js, Bootstrap

ChatGPT API (GPT-4o)

PostgreSQL, Redis

Рисунок Г.8 – Слайд «Методи та засоби розробки»

LARAVEL

Laravel — це сучасний PHP-фреймворк, який забезпечує зручну структуру коду, високу безпеку та швидку розробку вебзастосунків. У цій роботі Laravel було обрано завдяки його вбудованим можливостям маршрутизації, авторизації, роботи з базами даних та інтеграції сторонніх API, що відповідає поставленим вимогам вебзастосунку.



Рисунок Г.9 – Слайд «Laravel»

OPEN AI

API ChatGPT 4o — це інтерфейс, який дозволяє підключити можливості штучного інтелекту безпосередньо до застосунку. У своїй роботі його було використано для генерації навчальних уроків і пояснень до тестів. Цей API особливий тим, що здатен швидко формувати змістовні й адаптивні відповіді, а модель GPT-4o працює швидше та дешевше за попередні версії.



Рисунок Г.10 – Слайд «API ChatGPT 4o»

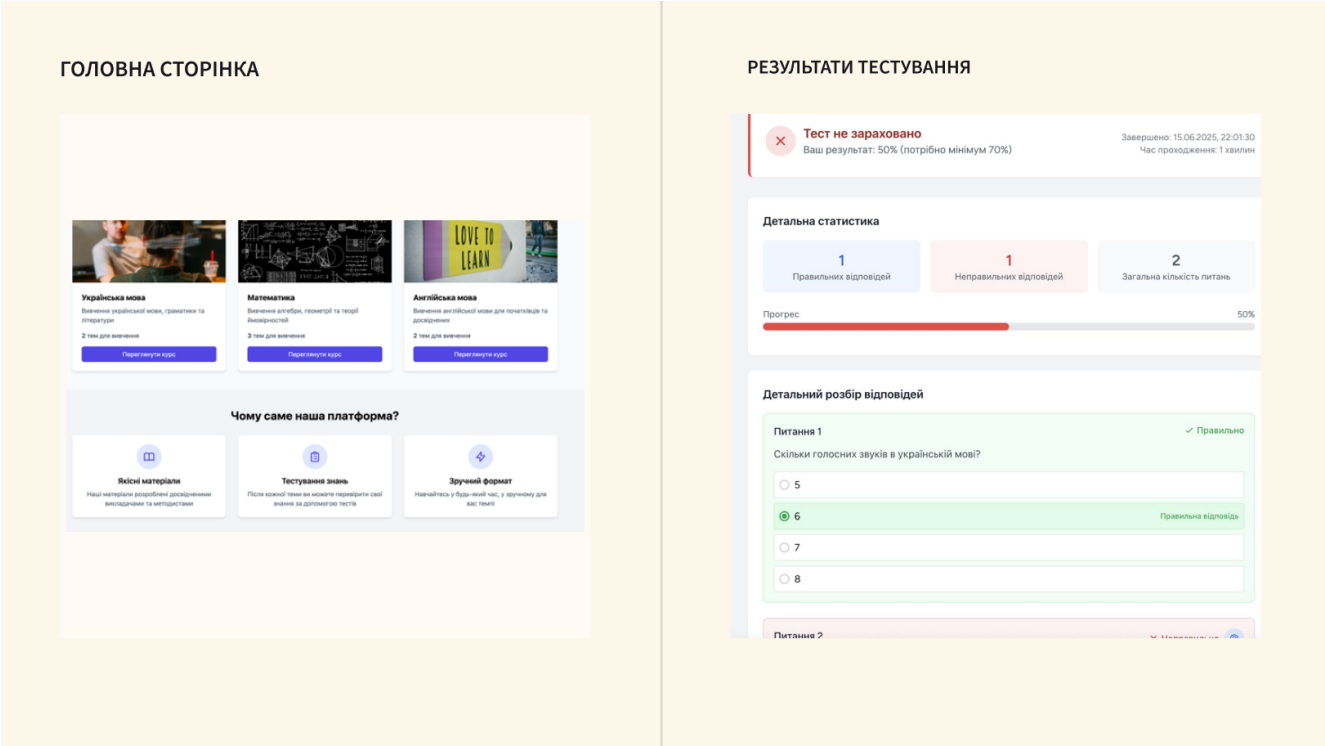


Рисунок Г.11 – Слайд «Інтерфейс вебзастосунку»

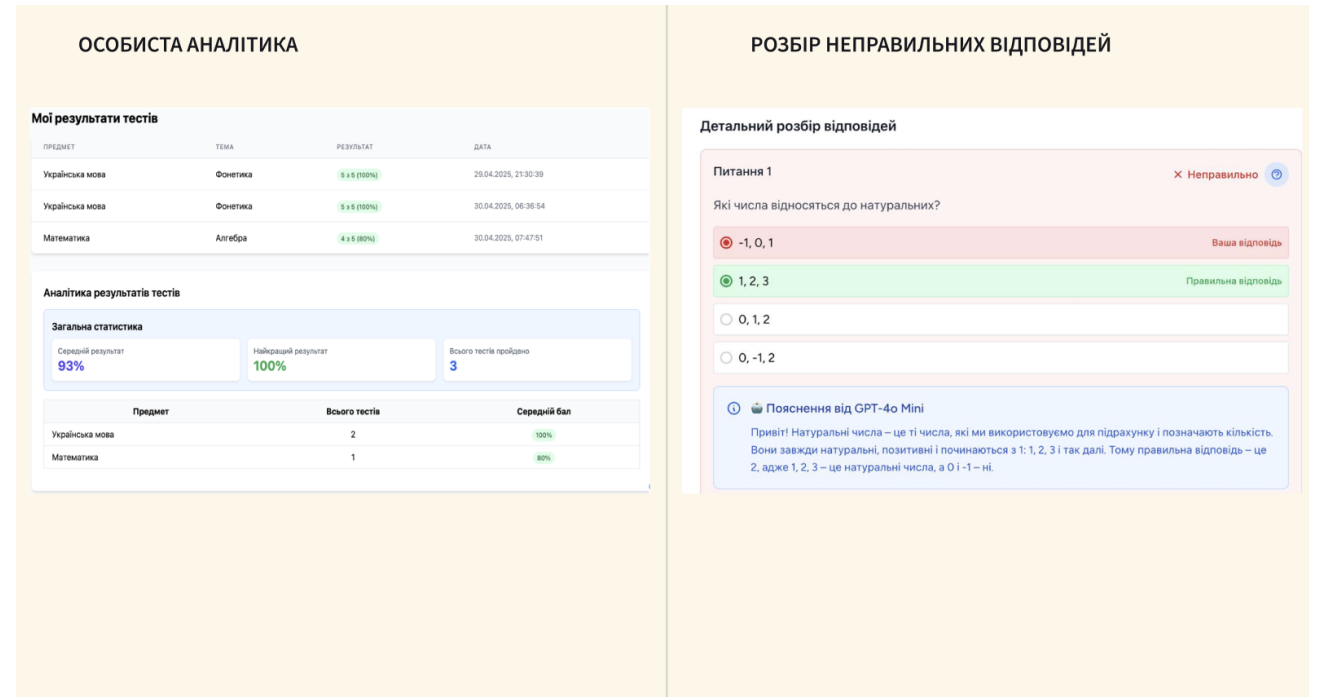


Рисунок Г.12 – Слайд «Інтерфейс вебзастосунку»

ГЕНЕРАЦІЯ УРОКІВ ЗА ДОПОМОГОЮ ШІ

Генерація уроків з ШІ
Створюй нові уроки автоматично за допомогою GPT-4o Mini

Оберіть тему для уроку

Алгебра

Ідеї для уроків

ОТРИМАТИ ІДЕЇ

Основи функцій: Відображення та графіки

Математичні нерівності: Розв'язування та застосування

Системи рівнянь: Графічний та алгебраїчний методи

Моделювання задач: Використання алгебри у реальному житті

Алгебра в іграх: Розв'язування головоломок та задач з використанням рівнянь

Або введіть свою назву уроку

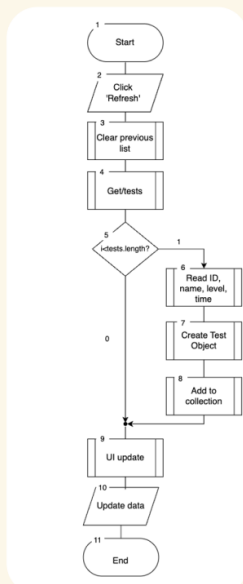
Основи функцій: Відображення та графіки

Генерація може зайняти 10-30 секунд

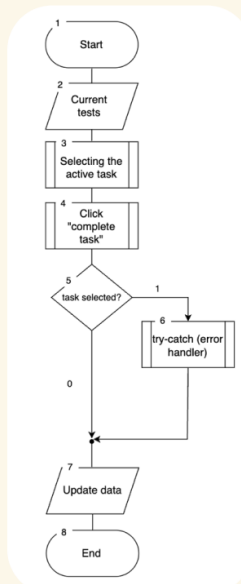
ГЕНЕРУЮ УРОК...

Рисунок Г.13 – Слайд «Інтерфейс вебзастосунку»

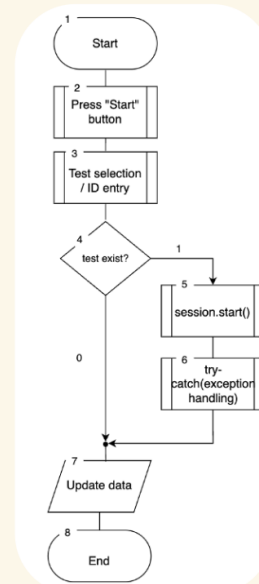
АЛГОРИТМИ



Алгоритм оновлення списку тестів



Алгоритм завершення вибраного процесу



Алгоритм запуску нової сесії

Рисунок Г.14 – Слайд «Алгоритми роботи вебзастосунку»

ВИСНОВКИ	○ Проведено аналіз існуючих рішень для підготовки до екзаменаційного тестування; ○ Розроблено інтерфейс вебзастосунку; ○ Проведено варіантний аналіз і обґрунтування вибору методів та засобів для реалізації вебзастосунку; ○ Розроблено повноцінний вебзастосунок із підтримкою генерації навчальних матеріалів; ○ Проведено тестування та оцінку ефективності розробленої системи в реальних умовах.
----------	---

Рисунок Г.15 – Слайд «Висновки»

ПУБЛІКАЦІЇ ТА АПРОБАЦІЇ	УДК 004.056 Я. С. Олійник О. М. Рейда АНАЛІЗ ВПЛИВУ ШТУЧНОГО ІНТЕЛЕКТУ НА ПРОЦЕС ПІДГОТОВКИ ДО НМТ Вінницький національний технічний університет Анотація У роботі розглянуто сучасні підходи до підготовки до Національного мультипредметного тестування (НМТ) та інтеграцію штучного інтелекту (ШІ) у цей процес. Проаналізовано ефективність традиційних методів навчання, а також інноваційні рішення на базі ШІ: адаптивні системи, предиктивну аналітику та автоматизовану генерацію контенту. Визначено ключові переваги, зокрема персоналізацію навчання та оптимізацію часу, та обмеження, пов'язані з етикою та технічною складністю. Зроблено висновки щодо перспектив застосування ШІ у масштабних освітніх системах. Ключові слова: НМТ, штучний інтелект, адаптивне навчання, персоналізація, освітні технології. Abstract The paper examines modern approaches to preparation for the National Multisubject Test (NMT) and the integration of artificial intelligence (AI) into this process. The efficiency of traditional learning methods, as well as innovative AI-based solutions—adaptive systems, predictive analytics, and automated content generation—is analyzed. Key advantages, including learning personalization and time optimization, and limitations related to ethics and technical complexity are identified. Conclusions are drawn regarding the prospects of AI implementation in large-scale educational systems. Keywords: NMT, artificial intelligence, adaptive learning, personalization, educational technologies. Вступ Національне мультипредметне тестування (НМТ) є вирішальним етапом для вступників до вищих навчальних закладів України. За даними УЦОЯО, у 2024 році 14,37 % від загальної кількості абітурієнтів не набрали необхідного проходного бала хоча б з одного предмета, даний факт змушує
-------------------------	---

Рисунок Г.16 – Слайд «Публікації й апробації»



ДЯКУЮ ЗА УВАГУ!

Рисунок Г.17 – Фінальний слайд