

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та кібербезпеки

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**WEB-РЕСУРС ДЛЯ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ ТЕСТІВ І
ОПИТУВАНЬ У НАВЧАЛЬНОМУ ПРОЦЕСІ**

**WEB-RESOURCES FOR ORGANIZING AND CONDUCTING
TESTS AND ASSESSMENTS IN THE EDUCATIONAL PROCESS**

спеціальність 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія
(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІ-41
Манькут Максим Михайлович

(підпис)

Керівник:
к.т.н., старший викладач
Гринюк Сергій Васильович

(підпис)

Кваліфікаційну роботу
допущено до захисту
«____» червня _____ 2023 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2023 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

проф. Н. Черняшук

« _____ » _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Манькут Максим Михайлови

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи WEB-Ресурс для організації та проведення тестів і опитувань у навчальному процесі

Керівник роботи Гринюк Сергій Васильович, к.т.н., асистент

затверджені наказом закладу вищої освіти від «28» грудня 2023 р. №982/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 01.06.2023 р.

3. Вихідні дані до роботи Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Вступ

Загальні відомості про веб-ресурси

Детальний аналіз існуючих засобів

Розробка ресурсу для тестів та опитувань

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Існуючі рішення

Структурні та архітектурні схеми

Моделі бази даних

Інтерфейс системи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Загальні відомості про веб-ресурси</i>	<i>Гринюк С.В.</i>		
<i>Детальний аналіз існуючих засобів</i>	<i>Гринюк С.В.</i>		
<i>Розробка ресурсу для тестів та опитувань</i>	<i>Гринюк С.В.</i>		
<i>Висновки</i>			

7. Дата видачі завдання 01.11.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Обґрунтування теми</i>	10.11.2022	Виконано
2.	<i>Огляд літератури на тему розробки веб-ресурсу</i>	22.11.2022	Виконано
3.	<i>Дослідження предметної області</i>	12.01.2023	Виконано
4.	<i>Вибір засобів розробки</i>	14.02.2023	Виконано
5.	<i>Розробка та тестування веб-ресурсу для проходження тестів та опитувань</i>	26.03.2023	Виконано
6.	<i>Оформлення матеріалів роботи</i>	03.04.2023	Виконано
7.	<i>Нормоконтроль</i>	17.05.2023	Виконано
8.	<i>Інструментальна перевірка на академічний плагіат</i>	24.05.2023	Виконано
9.	<i>Представлення кваліфікаційної роботи бакалавра до захисту</i>	13.06.2023	Виконано

Здобувач вищої освіти

(підпис)

Манькут М.М.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Гринюк С. В.

(прізвище, ініціали)

АНОТАЦІЯ

Манькут. М. М. Ресурс для організації та проведення тестів і опитувань у навчальному процесі.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2023.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел, трьох додатків.

Перший розділ присвячено огляду предметної області, тут розглядаються основні значення ресурсів для навчання і функцій які в них мають знаходитись.

В другому розділі проведений огляд існуючих рішень для проведення дистанційних опитувань.

Третій розділ присвячено опису інструментів та розробки застосунку.

Об'єкт – технології створення сайтів для покращення процесу дистанційного навчання

Метою роботи є розробка Web-ресурсу для організації та проведення тестів і опитувань у навчальному процесі. Для створення ресурсу використано мови програмування JavaScript, PHP, веб-сервер Apache, систему керування базами даних MySQL, технологію AJAX.

Ключові слова: навчання, веб-ресурс, тести, опитування.

ANNOTATION

Mankut. M. M. Resource for organizing and conducting tests and surveys in the educational process.

Qualification work of the bachelor's degree program "Computer Engineering", specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2023.

The qualification work consists of an introduction, three chapters, conclusions, a list of references, and three appendices.

The first chapter is devoted to an overview of the subject area, it discusses the main meanings of resources for learning and the functions they should contain.

The second section provides an overview of existing solutions for conducting distance learning surveys.

The third section is devoted to the description of tools and application development.

Object – technologies for creating websites to facilitate the process of distance learning

The purpose of the study is to develop a Web resource for organizing and conducting tests and surveys in the educational process. To create the resource, the programming languages JavaScript, PHP, Apache web server, MySQL database management system, AJAX technology were used.

Keywords: training, web resource, tests, surveys.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 ВЕБ РЕСУРС ДЛЯ ПРОВЕДЕННЯ ТЕСТІВ ТА ОПИТУВАНЬ	8
1.1 Переваги використання web-технологій.....	10
РОЗДІЛ 2 ІСНУЮЧІ ПРОГРАМНІ РІШЕННЯ В ОРГАНІЗАЦІЇ ПРОВЕДЕННЯ ТЕСТІВ ТА ОПИТУВАНЬ.....	13
2.1 Існуючі засоби для проведення дистанційного навчання.....	13
2.2 Поняття веб ресурсів для проведення тестів.....	14
РОЗДІЛ 3 МЕТОДИ ТА ЗАСОБИ ОРГАНІЗАЦІЇ ВЕБ РЕСУРСІВ	20
3.1 Вибір архітектури програмного комплексу	20
3.2 Опис архітектури серверу	21
3.3 Опис інструментів розробки	23
3.3.1 Мова програмування JavaScript, Vue.js та Vuetify.....	23
3.3.2 Мова програмування PHP	25
3.3.3 Серверна платформа Open Server та веб-сервер Apache.....	28
3.4 Обґрунтування програмної реалізації.....	32
3.5 Архітектурна та функціональна схема програмної системи	35
3.6 Методика роботи користувача.....	39
ВИСНОВКИ.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47
ДОДАТКИ.....	48

ВСТУП

Актуальність теми полягаєв тому що в останні роки широка доступність і використання інтернет-комунікацій разом з розвитком сучасних технологій відкрили унікальні можливості в освіті. Інтернет є джерелом активної інтелектуальної діяльності для сучасних студентів, з безмежними можливостями для отримання знань та вдосконалення навичок. Викладачі, зі свого боку, отримують можливість оптимізувати свої системи контролю, зокрема, шляхом переведення тестування в онлайн-режим. Великий інтерес до створення та ефективного використання освітніх веб-ресурсів зумовлений значною мірою виправданими сподіваннями на те, що вони можуть підвищити ефективність навчального процесу та якість засвоєння знань.

Метою роботи створення програмного проекту, який допоможе упростити процес навчання в дистанційному режимі.

Об'єкт дослідження – технології створення сайтів для організації тестів та опитувань.

Предмет дослідження веб-розробка.

Завдання, які необхідно виконати:

- Проаналізувати готові програмні рішення.
- Розробити інтерфейс проекту.
- Реалізувати веб-ресурс завдяки якому можна проводити тести та опитування у навчальному процесі.
- Зробити висновки за результатами роботи.

РОЗДІЛ 1

ВЕБ РЕСУРС ДЛЯ ПРОВЕДЕННЯ ТЕСТІВ ТА ОПИТУВАНЬ

У сучасних умовах розвиток сучасних технологій, поширення і доступність інтернет-комунікацій відкрили унікальні можливості для освіти.

Інтернет – це велика інформаційна база, яка надає безліч можливостей для покращення навчального процесу. Завдяки Інтернету студенти можуть швидко та ефективно знаходити необхідну навчальну інформацію, а також спілкуватися зі своїми викладачами та одногрупниками. Викладачі ж, з свого боку, можуть використовувати різні електронні ресурси для підвищення якості навчального процесу, зокрема, проводити онлайн-тестування та оцінювання знань студентів. Використання освітніх веб-ресурсів може значно полегшити та покращити процес навчання, дозволяючи зосередитися на засвоєнні матеріалу та підвищенні рівня знань. Тому створення та використання таких ресурсів є дуже важливим і закономірним кроком для оптимізації навчального процесу та підвищення якості освіти (рис. 1.1).

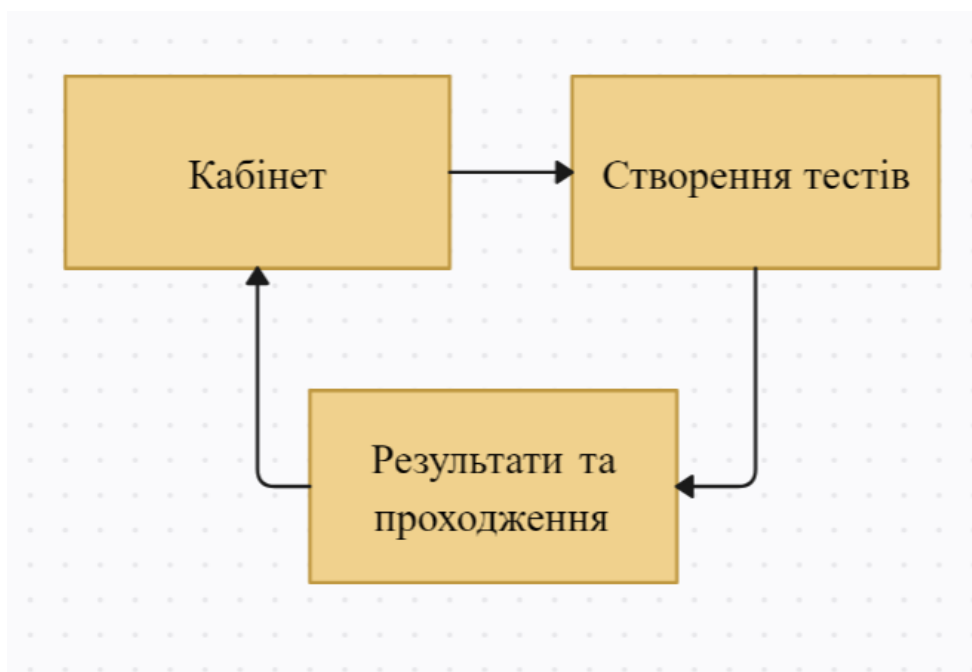


Рисунок 1.1 – Схема

Задачі, які повинні вирішуватись:

- Забезпечення користувача персональним кабінетом, де він зможе створювати тести і опитування;
- Швидке впровадження управління знаннями та проведення опитувань;
- Надання можливості максимально автоматизувати процеси тестування та опитування, використовувати їх у дистанційному навчанні;
- Наявність простого у використанні графічного інтерфейсу, який можна використовувати як на ПК, так і на мобільних телефонах.

Вхідні дані повинні мати наступну інформацію Інформація про випробування та опитування, що зберігається в нашій базі даних.

Вихідні дані будуть виглядати таким чином HTML-сторінка, яка показує інформацію про тест або опитування.

Результати роботи повинні включати в себе результати тестів та опитувань. Для легкості використання створена пошукова система, зручна відображення інформації.

Програмне забезпечення повинно мати декілька програмних засобів реалізації, а архітектура програмної системи повинна складатися з трьох компонентів: клієнт, сервер і база даних.

Користувачами системи мають бути (рис. 1.2):

- адміністратор системи;
- викладач;
- студент.

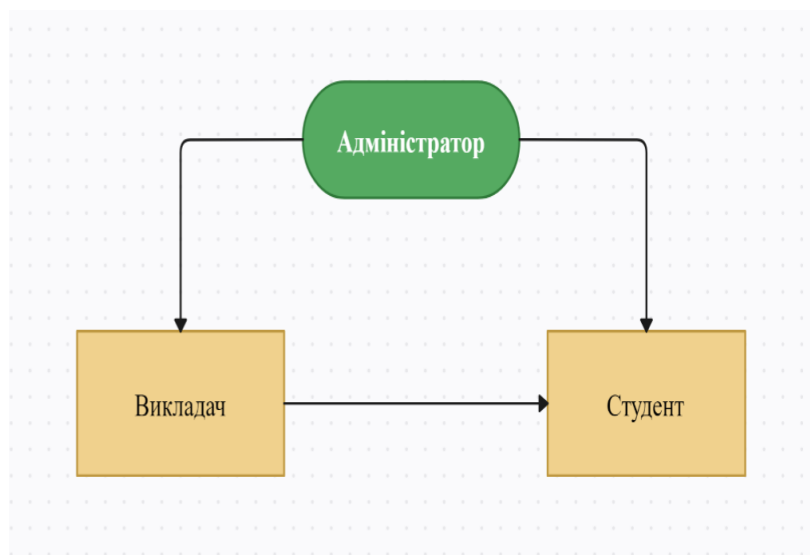


Рисунок 1.2 – Користувачі системи

Метою розробки web-ресурсу для організації та проведення тестів і опитувань у навчальному процесі є надання зручного та ефективного інструменту для вимірювання та оцінювання знань студентів. Цей ресурс дозволить автоматизувати процес проведення тестувань та опитувань, спрощуючи завдання викладачів та забезпечуючи зручну платформу для студентів.

1.1 Переваги використання web-технологій

Основні переваги використання web-технологій полягають у доступності та мобільності. Відкритий доступ до ресурсу через веб-браузер дозволить студентам проходити тести та опитування з будь-якого пристрою з Інтернет-підключенням. Крім того, цей підхід спрощує адміністрування та збереження результатів тестувань та опитувань.

Процес розробки такого ресурсу включатиме вибір та використання сучасних інструментів розробки, таких як фреймворки та мови програмування, для створення надійного та функціонального програмного забезпечення. Крім того, дизайн інтерфейсу ресурсу буде приділено особливу увагу, забезпечуючи зручну навігацію та привабливий зовнішній вигляд.

Цей web-ресурс буде спрямований на використання в навчальних закладах різного рівня. Викладачі матимуть можливість створювати тести та опитування, встановлювати правила та параметри, а також аналізувати результати. Студенти зможуть зручно проходити тести та опитування, отримувати результати та відстежувати свій прогрес у навчанні.

Розробка такого web-ресурсу є важливим кроком у покращенні навчального процесу, забезпечуючи швидкий та зручний спосіб проведення тестів та опитувань. Він допоможе збільшити ефективність оцінювання знань студентів та полегшить завдання викладачів у підготовці та аналізі результатів.

Використання веб-ресурсу для навчання має багато плюсів, які сприяють покращенню якості освіти та навчального процесу. Ось деякі з переваг, які надають веб-ресурси:

Доступність і нучність: Веб-ресурси доступні з будь-якого місця, де є Інтернет-підключення. Студенти можуть навчатися в зручний для них час і місце, не залежно від географічного розташування чи розкладу. Це дає їм більшу гнучкість та можливість самостійно контролювати своє навчання.

Розширені можливості навчання: Веб-ресурси не можуть використовувати різноманітні медіа-ресурси, такі як відео, аудіо, інтерактивні презентації, вебінари тощо. Це студентам отримують інформацію більш науково, цікаво і ефективно.

Інтерактивність та залученість: Веб-ресурси надають можливість взаємодії та залучення студентів до навчального процесу через використання форумів, чатів, обговорень, групових проектів тощо. Це сприяє активному навчанню, співпраці та обміну думками між студентами та викладачами.

Автоматизація та ефективність: Веб-ресурси дозволяють автоматизувати багато процесів, таких як проведення тестів, збір результатів, оцінювання та надання зворотного зв'язку. Це робить процес навчання більш ефективним, економить час викладачів та студентів, а також дозволяє швидко виявляти прогрес та робити корекції в навчальному процесі.

Індивідуалізація навчання: Веб-ресурси не дозволяють створювати індивідуальні навчальні шляхи для студентів. Вони можуть вибирати рівень працездатності завдань, виконувати самостійні завдання, отримувати персоналізовані рекомендації та підтримку в навчанні.

Моніторинг та аналітика: Веб-ресурси надають можливість викладачам та адміністраторам відстежувати прогрес студентів, аналізувати результати, виявляти труднощі та виявляти якісь проблеми в навчанні. Це дозволяє вчасно втручатися та забезпечити якісну підтримку студентам.

В цілому, використання веб-ресурсів для навчання має багато переваг, які покращують доступність, якість та ефективність навчання. Вони створюють нові можливості для студентів та викладачів, розширюють інтерактивність та сприяють активному навчанню.

РОЗДІЛ 2

ІСНУЮЧІ ПРОГРАМНІ РІШЕННЯ В ОРГАНІЗАЦІЇ ПРОВЕДЕННЯ ТЕСТІВ ТА ОПИТУВАНЬ

2.1 Існуючі засоби для проведення дистанційного навчання

Різноманітні веб-ресурси з кожним роком стають все більш популярними. На їх базі вже створено різне програмне забезпечення, що вирішує дуже широкий спектр задач, серед яких можна виділити наступні: організація та виконання тестів та опитувань у навчальному процесі.

Під час пошуку інформації та аналізу існуючих рішень ми виявили, що ці рішення наразі дуже складні у використанні або не повністю вирішують поставлені перед ними завдання:

– Освітній веб-ресурс "Perfect English Grammar" є одночасно і довідником, і збірником різноманітних вправ та онлайн-тестів з англійської граматики. Мінусом цієї системи для поставленого завдання є неможливість входу в систему, оскільки вона дозволяє зберігати отримані результати через авторизацію;

– Сайт Quizlet - це ресурс для занять з іноземної мови, який дозволяє користувачам практикуватись, виконуючи різноманітні завдання та тести. Програма не повністю реалізує це своє завдання, оскільки ресурс використовується лише для підготування до тесту, а не для проходження тесту;

Ще одним з недоліків цих системи є їхня складність. Для початківців ці програм можуть бути перевантаженими функціями і дуже вимогливими в плані конфігурації.

Оскільки тестувальна автоматизація в галузі освіти набуває все більшого поширення, існує нагальна необхідність у розробці зручних, інтуїтивно зрозумілих, адаптивних і не вибагливих до ресурсів платформ для проведення опитувань та тестувань.

Одним з варіантів, який може функціонувати на будь-яких сучасних пристроях, є розробка програмного системи, що базується на веб-технологіях.

2.2 Поняття веб ресурсів для проведення тестів

Ключові слова: веб-ресурс, навчальний процес, електронні освітні ресурси, освітні веб-ресурси, тестування, опитування.

Веб-ресурс - це складова частина інформаційного простору Інтернету, яка включає в себе набір інтегрованих засобів технічного і програмно-апаратного характеру, а також інформацію, що призначена для публікації та доступності для користувачів у глобальній мережі. Веб-ресурси можуть включати в себе різноманітні елементи, такі як текст, зображення, відео, аудіо, графіку та інші матеріали, які створюються для різних цілей, таких як реклама, навчання, розваги та інше. Веб-ресурси є важливим інструментом взаємодії між користувачами та різними організаціями, підприємствами та установами.

Навчальний процес можна описати як сукупність діяльностей, що мають організаційний та дидактичний характер і спрямовані на досягнення певного рівня освіти відповідно до державних стандартів. Цей процес включає у себе різноманітні заходи, такі як лекції, семінари, практичні заняття, самостійну роботу, контрольні роботи та інші, які сприяють засвоєнню та усвідомленню знань, вмінь та навичок з певної галузі знань. Метою навчального процесу є формування особистості з відповідним рівнем знань та навичок, готової до життя в суспільстві та здатної до самореалізації..

Навчальний процес включає в себе використання сучасних інформаційних технологій навчання, що дозволяє організувати інтерактивний та цікавий процес навчання, а також підвищує ефективність засвоєння знань. Основна мета навчального процесу - формування освіченої, гармонійно розвиненої особистості, здатної до саморозвитку та постійного оновлення наукових знань. Така особистість може швидко адаптуватися до змін у соціокультурній сфері та в галузях техніки, технологій, систем управління та організації праці в умовах ринкової економіки. Використання інформаційних технологій в навчальному процесі є ключовим, оскільки вони дозволяють забезпечити якісну освіту, незалежно від часу та місцезнаходження студента.

Електронні освітні ресурси є невід'ємною складовою сучасної освіти та мають важливе навчально-методичне призначення, що дозволяє забезпечувати ефективну навчальну діяльність студентів та забезпечувати доступ до якісної інформації. Одним з найбільш поширених різновидів електронних освітніх ресурсів є освітні веб-ресурси, які забезпечують навчальну підтримку студентів та є одним з ключових інструментів інформаційно-освітнього простору.

Використання сучасних веб-технологій дає змогу суттєво вдосконалити систему освіти, а отже, її подальша інформатизація - процес незворотний та обов'язковий. Освітні веб-ресурси у даній ситуації є оптимальним інструментом удосконалення професійної підготовки сучасних педагогів.

Розвиток дистанційної освіти вимагає правильного вибору програмного та апаратного забезпечення, а також знань платформ дистанційного навчання для реалізації дистанційних курсів університетських спеціальностей. Необхідно також забезпечити належний рівень матеріалів для дистанційного навчання. Сучасні веб-ресурси, включаючи веб-сайти та блоги, дедалі більше використовуються як ефективні інструменти електронного навчання, зокрема для дистанційного навчання. Тому розробка та вдосконалення цих ресурсів є важливим кроком у підвищенні якості дистанційного навчання і забезпеченні доступу до нього.

Освітні веб-ресурси - це електронні ресурси, що містять освітні матеріали та розміщені в мережі Інтернет у різних форматах.

За функціональним призначенням вони поділяються на:

- навчальні;
- навчально-методичні;
- наукові;
- педагогічні;
- програмні засоби.

Доступно та ефективно забезпечувати різноманітні аспекти освітнього процесу через електронні ресурси, що знаходяться в Інтернеті, включаючи

інформаційну, консультативну, методичну та навчальну підтримку, а також сприяють взаємодії з громадськістю:

- Освітні веб-ресурси мають перевагу перед традиційними методами навчання, оскільки вони дозволяють інтерактивно донести інформацію до аудиторії незалежно від її територіального місцезнаходження;

- оперативно висвітлювати діяльність веб-ресурсу на основі публікації новин, оглядів, каталогів видань, а також наукових, методичних і практичних матеріалів;

- використовувати сучасні засоби спілкування, такі як: електронна пошта інтерактивні конференції, форум - та ефективно організувати службу підтримки порталу;

- активізувати участь педагогічних працівників та учнів в Інтернет-олімпіадах, конкурсах, конференціях.

Використання можливостей інформаційного простору в галузі освіти та в діяльності педагогів є необхідність у сучасному світі. Освітні веб ресурси можуть виступити не лише пріоритетним засобом самоосвіти для освітянської спільноти, але й ефективним засобом організації місцевої системи освіти. Нові інформаційні ресурси впливають на всі складові освіти, даючи змогу педагогам вирішувати складні та актуальні завдання для розвитку інтелектуальних і творчих можливостей учнів.

Одним із шляхів до підвищення якості освіти є проведення ефективного контролю знань студентів. Мета контролю - визначити якість засвоєння знань, виміряти їх цінність та виставити певну оцінку. Процес перевірки та оцінювання знань є активним, і викладачі не лише фіксують фактичні знання учнів, а й впливають на результати навчання. Тому важливо знайти найефективніші способи перевірки знань для визначення досягнень студентів та їх заохочення до здобуття більшого багажу знань.

Одним із засобів досягнення цієї мети є контроль знань. Тестування допомагає визначити рівень знань студента за допомогою контекстних тестів. Такий підхід до управління знаннями дозволяє забезпечити об'єктивність і

незалежність від суб'єктивних факторів, що дозволяє отримати найбільш точні результати.

Переваги тестового контролю знань:

- простота процедури запису (введення) відповіді, незалежність оцінки від техніки письма;

- Кількісні критерії оцінки, які забезпечують повноту та глибину вивчення матеріалу, є важливими для оцінки знань студентів.;

- Простота та формалізованість процедури визначення оцінки є необхідними для забезпечення об'єктивності оцінювання та зручності для викладачів та студентів;

- Чітке та зрозуміле формулювання завдань тесту є ключовим для забезпечення правильної інтерпретації завдань студентами та їх правильної відповіді;

- Забезпечення еквівалентних критеріїв оцінювання знань та вмінь студентів дозволяє забезпечити справедливе порівняння між різними студентами та групами;

- Тестовий контроль знань забезпечує можливість перевірки значної кількості студентів одночасно та багаторазової перевірки для з'ясування змін в рівні підготовки;

- Використання комп'ютерних систем навчання та контролю дозволяє орієнтуватися на найновіші освітні технології та забезпечити ефективну та точну оцінку знань студентів;

- Тестовий контроль знань дозволяє забезпечити перевірку знань, умінь та навичок студентів на необхідному, заздалегідь запланованому рівні та забезпечує можливість самоконтролю;

- Використання тестового контролю знань як засобу підвищення якості освіти веде до різних висновків. Розширення методів і засобів оцінки якості знань майбутніх фахівців є важливим завданням, яке впливає з потреб педагогічної практики та майбутніх інтеграційних процесів в системі освіти України з європейським простором;

– Зокрема, комп'ютерні навчальні дисципліни та використання тестових завдань у комп'ютерному форматі широко використовуються в багатьох зарубіжних вишах і знаходять відгук в українській вищій освіті. Це свідчить про те, що використання комп'ютерного тестування є актуальним і ефективним підходом до контролю та оцінки знань студентів;

– Впровадження українських педагогічних та методичних розробок у навчальний процес здійснюється поступово у співпраці з викладачами вищих технічних та вищих навчальних закладів.;

– Оскільки наукова Інформація швидко застаріває, викладачам необхідно оновлювати не лише лекційні матеріали, а й відповідні контрольні тести та супровідні навчальні посібники.;

– Студенти вищих освітніх закладів зобов'язані систематично займатися підготовкою до складання іспитів протягом всіх курсів навчання та проходження практики.

Традиційні методи контролю та оцінювання знань у поєднанні з новими технологіями відкривають перед викладачами широкий спектр можливостей. Незаперечним залишається той факт, що підвищення якості навчання студентів безпосередньо пов'язане зі створенням і послідовним використанням систем тестового контролю засвоєння знань. Особливо актуальним це питання є сьогодні, коли робляться важливі кроки щодо інтеграції української системи вищої освіти в європейську університетську спільноту, а популярності і підтримки зазнають ідеї так названого "Болонського процесу".

Якість навчання значною мірою залежить від якості роботи викладачів, тому проблема контролю та оцінювання якості роботи викладачів є одним із найскладніших і найважливіших завдань в процесі управління якістю освіти..

Одним із способів оцінки роботи педагога може бути опитування учнів, наприклад, через визначені проміжки часу або під час засвоєння певної дисципліни.

Опитування – це метод збору соціологічної інформації про досліджуваний об'єкт під час безпосереднього або опосередкованого спілкування того хто опитує з особою, яка відповідає на питання.

Використання опитувань учнів для визначення якості викладання надає можливість більш ефективно здійснювати моніторинг та оцінювати якість праці викладачів і є необхідною складовою для вдосконалення професійних знань та навичок викладання вчителів. Для студентів опитування надають можливості не лише поставити оцінку своїм викладачам, але й оцінити себе як учасників навчальної діяльності. Також опитування можуть допомогти у вирішенні організаційних питань для учнів та вчителів.

Використання постійного контролю знань у навчальному процесі значно підвищує рівень мотивації до навчання, а опитування покращує якість освітнього процесу, але для проведення тесту чи опитування викладачеві доводиться докладати чимало зусиль для постановки завдання та організації роботи. Тому важливим напрямком вирішення цієї проблеми є дуже оперативний контроль, максимальна деталізація та автоматизація процесу тестування та опитування, а також використання веб-ресурсів для організації тестів та анкетування в дистанційному навчанні.

РОЗДІЛ 3

МЕТОДИ ТА ЗАСОБИ ОРГАНІЗАЦІЇ ВЕБ-РЕСУРСІВ

Розглядаючи задачу та можливі методи її вирішення, було вирішено створити програмний комплекс, який буде базуватися на веб-технологіях. Головна перевага веб-застосунку перед альтернативними варіантами полягає в тому, що він може бути використаний на будь-яких пристроях без необхідності портування на конкретну операційну систему. Браузер та його віртуальна машина виступають у ролі універсальної операційної системи та комп'ютера, що значно спрощує процес використання програмного комплексу. Використання веб-технологій дозволяє досягти найкращих результатів за мінімальний час та витрати, а також забезпечити максимальну доступність користувачів до програмного продукту.

3.1 Вибір архітектури програмного комплексу

Для реалізації задачі використовується триланкова архітектура з клієнтом, сервером та базою даних (див. рис. 3.1).

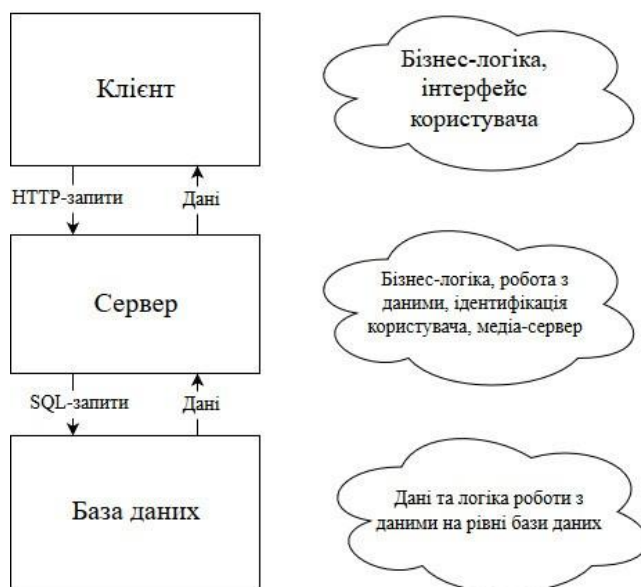


Рисунок 3.1 – Триланкова архітектура програмного комплексу.

Це правильно, сервер дійсно є центром програмного комплексу, який забезпечує збереження та обробку даних. Важливою функцією сервера є забезпечення безпеки даних шляхом ідентифікації користувачів та надання їм індивідуальних прав доступу до програмного застосунку та бази даних. Якщо авторизація та контроль доступу до даних здійснюються на рівні користувача, то це може призвести до неконтрольованого доступу до даних та їхньої порушення цілісності. Тому логіку авторизації та контролю доступу до даних краще реалізовувати на рівні сервера.

Це програмний комплекс з сервером, який забезпечує основну бізнес-логіку та доступ до бази даних. Користувачі взаємодіють з веб-сайтом, що дозволяє налаштувати програму та переглядати результати роботи. Логіка аутентифікації також реалізується на рівні користувача, щоб уникнути неконтрольованого доступу до даних.

Рівень бази даних забезпечує збереження та цілісність даних за допомогою зв'язків та ключів. Тут також можна реалізувати бізнес-логіку, яка не вимагає зовнішніх джерел даних.

3.2 Опис архітектури серверу

Для виконання серверу було використано API, який є інтерфейсом прикладного програмування. API (Application Programming Interface) – це набір чітко визначених методів і протоколів, які дозволяють різним компонентам програмного забезпечення взаємодіяти один з одним. Використовуючи API, розробники можуть швидко створювати нові програми або розширювати функціональність існуючих. API може бути застосований в різних областях, таких як веб-системи, операційні системи, бази даних, апаратне забезпечення та програмні бібліотеки. Він виступає як мости між компонентами, надаючи зручний спосіб взаємодії та обміну даними між ними.

API є незамінним інструментом у розробці програмного забезпечення, особливо для веб-додатків. Вони надають розробникам доступ до готового

набору функціональності, що дозволяє їм ефективно вирішувати конкретні завдання, не займаючись повторною реалізацією вже наявних функцій. Завдяки API розробники можуть швидко і просто інтегрувати зовнішні сервіси, бібліотеки та інструменти у свої додатки, забезпечуючи їм більшу функціональність та гнучкість. API спрощує розробку, робить її більш продуктивною та допомагає розробникам швидше досягати бажаних результатів.

API може бути частиною комплексного набору розробки програмного забезпечення або існувати самостійно як окрема реалізація. У контексті веб-розробки використання API зазвичай передбачає визначення повідомлень запиту HTTP та структури повідомлень-відповідей, які можуть мати формат XML або JSON. Це дозволяє розробникам здійснювати взаємодію з веб-серверами, обмінюватись даними та виконувати різноманітні операції, такі як отримання, оновлення чи видалення ресурсів. Використання стандартних протоколів та форматів спрощує інтеграцію між різними системами та дозволяє розробникам ефективно використовувати функціональність, яку надає API.

Сучасні тенденції веб-розробки переходять від SOAP до REST та ROA. Це пов'язано з розвитком Семантичного вебу та гібридних API, які комбінують кілька інтерфейсів для створення нових додатків.

Для клієнтського застосунку використовувався фреймворк, заснований на шаблоні проектування MVVM (Model-View-ViewModel). Цей шаблон дозволяє відокремити розробку графічного інтерфейсу від бізнес-логіки, забезпечуючи легке управління даними та відображенням. MVVM особливо ефективний у платформах з підтримкою "зв'язування даних".

MVVM (Model-View-ViewModel) сприяє чистоті коду, полегшує тестування та розширення програмного забезпечення. Він поліпшує підтримку, зрозуміння та обслуговування програми, розділяючи роль кожного компонента. Розробники можуть одночасно працювати над графічним інтерфейсом та бізнес-логікою, що допомагає розподіляти завдання та прискорює розробку програмного забезпечення.

Шаблон MVVM (Model-View-ViewModel) складається з трьох основних частин:

- Модель (Model): Вона представляє фундаментальні дані, необхідні для роботи застосунку.
- Представлення (View): Воно відповідає за графічний інтерфейс, такий як вікна, кнопки і т.д.
- Модель вигляду (ViewModel): Вона служить абстракцією для представлення і забезпечує зв'язування з даними з Моделі.

3.3 Опис інструментів розробки

Цей програмний комплекс побудований за принципами триланкової архітектури з використанням різних технологій на кожному рівні. Головною метою було створення мультиплатформного рішення, яке базується на відкритих технологіях. Для клієнтського рівня використовувалися JavaScript як мова програмування, фреймворк Vue.js для побудови веб-застосунків і фреймворк Vuetify для створення графічних інтерфейсів. Для серверного рівня була використана мова програмування PHP, серверна платформа Open Server та веб-сервер Apache. В якості реляційної бази даних використовувався MySQL. Технологія AJAX була використана під час розробки веб-системи.

3.3.1 Мова програмування JavaScript, Vue.js та Vuetify

JavaScript є прототипною мовою програмування, що означає, що об'єкти можуть успадковувати властивості та методи інших об'єктів. Це дає програмістам гнучкість і можливість швидко розширювати функціональність програм. Вона також має скриптовий характер та використовує динамічну типізацію, що дозволяє змінювати типи даних залежно від потреб програми. JavaScript підтримує імперативну парадигму програмування, що дає змогу створювати послідовність команд для виконання завдань. Крім того, вона частково підтримує функціональну парадигму, що дозволяє використовувати функції як об'єкти першого класу.

су. Завдяки своїм особливостям, JavaScript став незамінним інструментом для розробки різноманітних застосунків та скриптів.

JavaScript є прототипною мовою програмування, що означає, що об'єкти можуть успадковувати властивості та методи інших об'єктів. Це дає програмістам гнучкість і можливість швидко розширювати функціональність програм. Вона також має скриптовий характер та використовує динамічну типізацію, що дозволяє змінювати типи даних залежно від потреб програми. JavaScript підтримує імперативну парадигму програмування, що дає змогу створювати послідовність команд для виконання завдань. Крім того, вона частково підтримує функціональну парадигму, що дозволяє використовувати функції як об'єкти першого класу. Завдяки своїм особливостям, JavaScript став незамінним інструментом для розробки різноманітних застосунків та скриптів.

Мова JavaScript має широке застосування і використовується для різних цілей, зокрема:

- Написання сценаріїв веб-сторінок, що надає їм інтерактивність та динаміку;
- Розробки односторінкових веб-застосунків (React, AngularJS, Vue.js), які працюють без перезавантаження сторінки;
- Серверного програмування за допомогою платформи Node.js, що дозволяє виконувати JavaScript на стороні сервера;
- Розробки стаціонарних застосунків з використанням Electron або NW.js, які базуються на веб-технологіях;
- Розробки мобільних застосунків з використанням фреймворків React Native або Cordova, які дозволяють перенести JavaScript-код на мобільні платформи;
- Написання сценаріїв для різного прикладного програмного забезпечення, такого як Adobe Creative Suite чи Apache JMeter;

Так, ви правильно зауважуєте. Мови Java і JavaScript є двома різними мовами програмування з відмінною семантикою. Хоча вони мають деякі схожі риси у стандартних бібліотеках та правилах іменування, їх синтаксис був

вплинутий мовою С. Однак, семантика та дизайн JavaScript були вплинуті мовами Self та Scheme, що надає їй унікальні особливості та можливості.

Ви правильно описали процес включення JavaScript-коду в HTML-сторінку в рамках технології DHTML. JavaScript-код додається в HTML-файл за допомогою тегів `<script></script>`. Зазвичай, браузері вбудовують інтерпретатор JavaScript за умовчанням, тому мова сценаріїв, яку вони розуміють, є JavaScript. Однак, браузері також можуть підтримувати інші мови сценаріїв, які можна вказати за допомогою атрибуту `type` в тезі `<script>`.

При розробці великих і нетривіальних веб-застосунків з використанням JavaScript, критично важливим є доступ до інструментів відлагодження. Браузери можуть мати відмінності в інтерпретації JavaScript і реалізації Об'єктної моделі документа, тому для кожного браузера рекомендується мати власний набір інструментів для відлагодження. Сучасні розробницькі інструменти, такі як Google Chrome DevTools, Mozilla Firefox Developer Tools, Microsoft Edge DevTools, надають потужні засоби для відлагодження JavaScript-коду в різних браузерах. Вони дозволяють аналізувати, відстежувати та налагоджувати код, перевіряти значення змінних, виконувати кроки відлагодження, а також аналізувати мережевий трафік і багато іншого. Таким чином, розробники повинні мати набір відповідних інструментів для відлагодження веб-застосунків в різних браузерах, щоб забезпечити ефективну розробку та налагодження.

3.3.2 Мова програмування PHP

Мова програмування PHP є скриптовою мовою, спеціально призначеною для створення динамічних веб-сторінок на веб-сервері. Вона надає зручний і потужний інструментарій для розробки веб-додатків, який включає в себе вбудовані бібліотеки для роботи з базами даних, обробки форм, роботи з файлами, управління сеансами користувачів та багато іншого.

Однією з головних переваг PHP є його широка підтримка і популярність серед хостинг-провайдерів. Багато веб-серверів підтримують PHP і мають встановлені необхідні компоненти для його виконання, що робить його

доступним для багатьох веб-розробників. Крім того, PHP є проектом відкритого програмного забезпечення, що сприяє його постійному розвитку та вдосконаленню.

PHP використовується для роботи з різними системами управління базами даних, такими як MySQL, PostgreSQL, Oracle та іншими. Це дозволяє розробникам з легкістю підключатися до різних баз даних і працювати з ними, виконуючи операції зчитування, запису та зміни даних.

PHP також має простий синтаксис, що робить його легким для вивчення і використання. Він пропонує розробникам широкі можливості для створення динамічного контенту, взаємодії з користувачем та розширення функціональності веб-сайтів і додатків.

Загалом, PHP є потужним і популярним інструментом для веб-розробки, який надає розробникам можливість створювати динамічні веб-сторінки та додатки з легкістю та ефективністю.

Так, PHP дійсно надає широкий набір вбудованих засобів для розробки веб-додатків. Декілька основних засобів включають:

- Автоматичне збирання даних з форм та інших джерел: PHP має зручний механізм для автоматичного збирання POST і GET-параметрів, а також змінних оточення веб-сервера. Ці дані автоматично доступні у вигляді асоціативних масивів, що спрощує отримання та обробку надісланих даних;

- Підтримка різних систем управління базами даних (СУБД): PHP має вбудовану підтримку для багатьох популярних СУБД, таких як MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape і Apache Derby, Informix, Ovrimos SQL, Lotus Notes, DB++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird/InterBase, Paradox File Access, MaxDB і інтерфейс PDO. Це дозволяє зручно взаємодіяти з різними базами даних, виконувати запити, отримувати результати і здійснювати інші операції з базами даних безпосередньо з PHP-скриптів;

- Автоматизована відправка HTTP-заголовків, що дозволяє налаштовувати різні аспекти комунікації з клієнтом, такі як кешування, кодування, типи відповіді і багато іншого;

- Робота з HTTP-авторизацією, що дозволяє здійснювати автентифікацію користувачів на основі HTTP-протоколу;

- Робота з cookies і сесіями, що дозволяє зберігати стан інформації між різними запитами користувача і забезпечувати ідентифікацію сеансів;

- Робота з локальними і віддаленими файлами та сокетами, що відкриває можливості для роботи з файловою системою сервера, а також взаємодії з іншими системами через сокети;

- Обробка файлів, що завантажуються на сервер, дозволяє отримувати і обробляти файли, які користувачі відправляють на сервер через форми або інші механізми завантаження файлів. Це дозволяє зберігати, обробляти або використовувати ці файли у веб-додатках за потребою;

- Робота з XForms надає засоби для створення складних форм, які дозволяють здійснювати більш складну валідацію даних, залежності між полями, обробку подій та інші функції. XForms є розширенням мови XML і дозволяє створювати багатофункціональні форми для взаємодії з користувачами.

Мова програмування PHP використовує специфічний механізм для вбудовування PHP-коду безпосередньо в HTML-сторінки. Виконання PHP-коду починається після першої екрануючої послідовності (<?) і триває до зустрічі парної екрануючої послідовності (?>).

Синтаксис мови PHP взятий з мов C і Perl, що робить його знайомим для багатьох програмістів і спрощує процес вивчення. PHP є мовою програмування, спеціально спрямованою на роботу в Інтернеті, і поєднує переваги цих двох мов. Тому вона стала надзвичайно популярною серед веб-програмістів і є найпоширенішою мовою для створення веб-застосунків.

Правильно, ви правильно відзначили переваги PHP. Його вбудовані функції роблять розробку веб-додатків швидшою і простішою, дозволяючи

уникнути написання складних функцій з нуля. Синтаксис PHP легкий для вивчення і розуміння, що полегшує розробку веб-додатків.

Інтерпретованість PHP дозволяє швидко тестувати та випробовувати код без необхідності компіляції, що сприяє швидкому розвитку. Відкритий характер PHP і його безкоштовна доступність дозволяють багатьом розробникам використовувати його для створення веб-додатків без фінансових обмежень. Наявність великої спільноти розробників PHP також забезпечує підтримку та доступ до різноманітних ресурсів, документації та сторонніх бібліотек, що сприяє продуктивності та розширюваності веб-додатків на PHP.

Одним з застосувань мови PHP є розробка парсерів, що виконують синтаксичний аналіз HTML-сторінок та опрацьовують дані згідно з поставленою задачею. Це може бути корисно, наприклад, для збору даних зі сторінок веб-сайтів. PHP надає зручні засоби для роботи з HTML, такі як DOM-об'єкти та функції для розбору HTML-коду.

3.3.3 Серверна платформа Open Server та веб-сервер Apache

Платформа Open Server є зручним інструментом для веб-розробників, оскільки вона надає комплексне середовище для розробки, налагодження та тестування веб-проектів. Вона включає в себе веб-сервери Apache і nginx, які є популярними серверами у веб-розробці і підтримують різноманітні можливості.

Apache є одним з найпоширеніших веб-серверів у світі і підтримує різні операційні системи. Він може бути розширений за допомогою модулів, що дозволяє додавати додаткові функції і підтримку різних мов програмування, таких як Perl, Python, Tcl і PHP. Apache також підтримує функцію віртуального хостингу, що дозволяє обслуговувати кілька веб-сайтів на одному сервері.

Open Server також включає PHP, яка є однією з найпоширеніших мов програмування для веб-розробки. PHP підтримує роботу з базами даних, включаючи MySQL, і забезпечує можливість створення динамічних веб-сторінок та взаємодії з користувачем.

Для зручного управління базами даних MySQL у складі Open Server є phpMyAdmin, який надає графічний інтерфейс для виконання операцій з базами даних.

Додатково, Open Server включає FTP-сервер FileZilla, який дозволяє передавати файли між комп'ютером і сервером, спрощуючи процес розробки та розгортання веб-додатків.

Загалом, Open Server забезпечує повноцінне середовище для розробки веб-додатків, яке об'єднує різні компоненти і надає зручний інтерфейс для адміністрування та налаштування.

Веб-сервер Apache дійсно використовується для передачі статичних та динамічних веб-сторінок через протокол HTTP в Інтернеті. Однак, крім цього, Apache може виконувати й інші функції, у тому числі працювати як кешувальний проксі-сервер.

Так, Apache використовує модульну архітектуру, що дозволяє розширювати його функціональність за допомогою модулів. Ядро Apache надає основну функціональність веб-сервера, включаючи обробку конфігураційних файлів, виконання протоколу HTTP та систему завантаження модулів. Але багато з функціональних можливостей Apache реалізуються саме через модулі.

Модуль `mod_php` робить сервер Apache ідеальним вибором для розробки веб-додатків і динамічних веб-сторінок з використанням мови програмування PHP. Він інтегрується з Apache і дозволяє виконувати PHP-скрипти, які генерують вихідний HTML-код. Цей HTML-код потім надсилається клієнту через протокол HTTP. Модуль `mod_php` забезпечує швидку та ефективну обробку PHP-коду, що дозволяє розробникам створювати потужні та функціональні веб-додатки.

Модулі можуть бути включені або виключені в залежності від потреб конкретного веб-проекту або конфігурації сервера. Додавання або видалення модулів зазвичай вимагає редагування конфігураційних файлів Apache та перезавантаження сервера для застосування змін.

Таким чином, модулі Apache розширюють базові можливості сервера, дозволяючи використовувати різні функції і розширення для задоволення конкретних потреб веб-додатків та сервісів. Ядро Apache реалізоване мовою програмування C, що забезпечує високу ефективність та швидкодію. Використання мови C дозволяє оптимально використовувати системні ресурси і зменшити накладні витрати. Система конфігурації Apache базується на текстових конфігураційних файлах, що спрощує налаштування сервера. Існують три рівні конфігурації, кожен з яких дозволяє налаштовувати параметри сервера, віртуальних хостів або окремих директорій. Мова конфігураційних файлів Apache базується на блоках директив, що дозволяє зручно організувати та структурувати налаштування сервера. Деякі модулі використовують конфігураційні файли операційної системи, а параметри також можуть бути задані через ключі командного рядка.

Для веб-сервера Apache доступні різні моделі симетричного мультипроцесування, які можуть використовуватись в залежності від потреб і особливостей серверного середовища. Ось кілька з них:

- Модель *worker* у веб-сервері Apache поєднує переваги мультипроцесорних та багатонитевих рішень, дозволяючи ефективно обслуговувати багато клієнтів з мінімальним використанням ресурсів.

- Модель *pre-fork*: Ця модель ґрунтується на попередньому створенні окремих процесів для обробки запитів клієнтів. Кожен процес обслуговує окремого клієнта.

- Модель *perchild*: Це гібридна модель з фіксованою кількістю процесів. Кожен процес виконується під відповідним користувачем і групою, що дозволяє запускати віртуальні хости з різними правами доступу.

- Модель *netware*: Ця модель оптимізована для роботи в середовищі NetWare, і вона використовує багатопотоковий підхід.

- Модель *winnt*: Ця модель створена для операційної системи Microsoft Windows і також використовує багатопотоковий підхід.

– Модель Apache-ІТК: Ця модель ґрунтується на моделі pre-fork, але дозволяє запускати кожен віртуальний хост під окремими ідентифікаторами користувача (UID) та групи (GID).

– Модель peruser: Ця модель також базується на моделі perchild і дозволяє запускати кожен віртуальний хост під окремими UID та GID, але не використовує ниті.

– Кожна з цих моделей має свої особливості і може бути вибрана залежно від вимог до продуктивності, безпеки та ресурсів сервера.

MySQL – це безкоштовна вільна система керування реляційними базами даних та невеликий однопотоковий сервер баз даних, який відрізняється швидкістю, надійністю та простотою використання. Вона зберігає дані в окремих базах даних, що підвищує швидкість та гнучкість, а таблиці пов'язані зв'язками, що дозволяє виконувати запити, об'єднуючи дані з різних таблиць за допомогою мови структурованих запитів SQL. MySQL є популярним рішенням для малих і середніх проектів, а його сервер може бути скомпільований на різних платформах, зокрема на UNIX-системах, що підтримують багатопотоковість і сприяють загальній продуктивності системи.

Система vMySQL є відкритим програмним забезпеченням, доступним для використання та модифікації будь-яким користувачем. Це означає, що кожен має право користуватися системою MySQL і змінювати її відповідно до своїх потреб. Будь-хто може вивчати вихідний код системи та вносити зміни для поліпшення або налаштування системи під свої вимоги. Це робить систему MySQL гнучкою та налаштовуваною для різних потреб користувачів.

MySQL сервер постійно працює на комп'ютері, а клієнтські програми, наприклад, PHP-скрипти, взаємодіють з сервером, відправляючи йому SQL-запити через мережеві інструменти або сокети. Сервер MySQL обробляє ці запити, зберігає дані та надсилає результати клієнту. Запити вказують серверу, яку інформацію необхідно отримати з бази даних. Сервер БД повертає відповідь, яка містить результати запиту, клієнту (скрипту).

MySQL базується на трирівневій організації даних. Найвищий рівень - база даних, яка складається з пов'язаних таблиць. Кожна таблиця містить рядки, які представляють окремі записи, і стовпці, що визначають типи даних. Фізичне представлення баз даних і таблиць MySQL здійснюється за допомогою файлів з розширеннями, такими як `frm`, `MYD` і `MYI`. Ця структура дозволяє ефективно організувати дані і забезпечує швидкий доступ до них. Більше того, один сервер MySQL може підтримувати кілька баз даних одночасно, а кожна база даних має свій унікальний логін і пароль для забезпечення доступу до неї.

MySQL є потужною, швидкою та надійною системою управління реляційними базами даних. Вона пропонує широкі можливості і використовується для задоволення потреб різноманітних проєктів. MySQL відзначається простотою в установленні та використанні, а також високою надійністю. Ця система надає зручний інструментарій для роботи з базами даних, що включає можливість підключення до сервера MySQL за допомогою функції `mysql_connect()`. Одна з функцій MySQL, `mysql_fetch_array()`, повертає асоціативний масив з даними, і параметр `$result_type` дозволяє вказати тип повернутої структури даних. Значення `MYSQL_BOTH` використовується за замовчуванням і повертає масив з подвійними індексами. MySQL є популярним вибором серед розробників і користувачів завдяки своїм перевагам у широких можливостях, ефективності та активній спільноті підтримки.

3.4 Обґрунтування програмної реалізації

Під час розробки системи було проведено дослідження та аналіз цільових областей і вимог замовника. В результаті ретельного аналізу було вирішено розробити програмний продукт на основі веб-технологій для використання у веб-браузерах.

Для серверної частини системи була обрана технологія, яка відповідає принципам простоти використання, актуальності та можливості запуску на будь-якій операційній системі. Платформа `Open Server` є потужним і корисним

сервером, який дозволяє швидко налаштувати доступ до веб-сервера Apache і обслуговувати різноманітні сайти з використанням веб-хостингу. Для роботи з базою даних використовувалася мова програмування PHP.

На клієнтському рівні було обрано технології, які відповідають таким же принципам, але з урахуванням виконання в браузері. Фреймворк Vue.js був обраний через його здатність створювати складні графічні інтерфейси, які легко модифікувати, тестувати та розширювати в подальших етапах розробки. Мова програмування JavaScript була обрана через її прототипну природу та динамічну типізацію. Для стилізації клієнтського додатку був використаний фреймворк Vuetify, який надає широку бібліотеку готових інтерфейсів та підтримку масштабування до різних розмірів пристроїв, включаючи мобільні телефони.

Вищезазначені технології та інструменти були обрані з метою забезпечення якості, простоти розробки та зручності використання програмного продукту.

Базою даних було обрано MySQL через те, що це проект з відкритим вихідним кодом і надзвичайно великою підтримкою з боку розробників. Ще однією важливою рисою є висока швидкість, стабільність і багато вбудованих можливостей.

Також для створення системи була обрана технологія AJAX через її можливість використовувати API-запит веб-клієнта (браузера) до веб-сервера за протоколом HTTP у фоновому режимі.

Ці технології захищені ліцензіями, які виключають можливість патентних прав розробників і надають доступ до початкового коду проекту, що дозволяє створювати високоякісні та надійні продукти.

Система для організації та проведення тестів та опитувань у навчальному процесі складатиметься з декількох модулів, кожен з яких буде розділений на різну кількість функціональних підблоків. Основним модулем є кабінет користувача, який також буде елементом зовнішнього оформлення системи. Така структура зробить систему легкою та інтуїтивно зрозумілою у

використанні, оскільки модулі розташовані в порядку, який зазвичай використовується для вирішення подібних завдань.

На рисунку 3.1 наведена схема структури системи, на якій розташовані програмні модулі.



Рисунок 3.1 – Схема структури системи

Після введення та обробки відповідей на запитання, які є вхідними даними, система виконує розрахунки та виводить на екран отримані результати тестові бали та результати опитування в цілому.

На рисунку 3.2 представлена діаграма прецедентів, яка описує функції та дії гравців у системі.



Рисунок 3.2 – Діаграма прецедентів системи

Діаграма прецедентів передбачає існування трьох типів користувачів: студентів, викладачів та адміністраторів. Викладачі створюють тести та опитування, студенти проходять їх, а адміністратори відповідають за реєстрацію викладачів та студентів у системі, а також можуть видаляти застарілу інформацію. Усі три типи користувачів мають доступ до перегляду результатів тестування та опитування.

3.5 Архітектурна та функціональна схема програмної системи

Весь контент системи, а також її основні дані, це дані про тести і опитування зберігаються в базі даних. Концептуальна модель бази даних має такий вигляд (рис. 3.3).

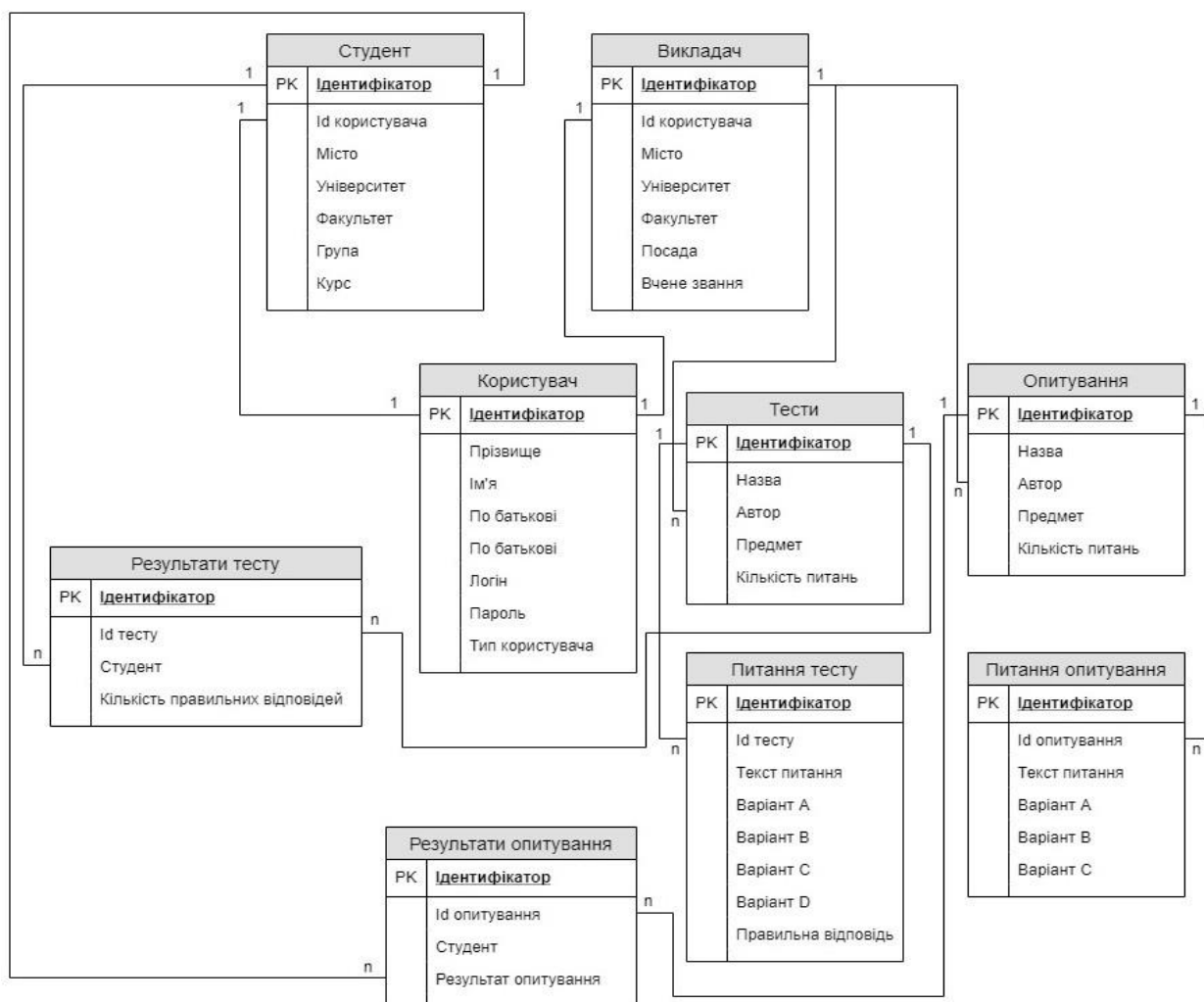


Рисунок 3.3 – Концептуальна модель бази даних

Таблиці бази даних складаються з: “Користувач”, “Учень”, “Викладач”, “Тест”, “Опитування”, “Запитування тесту”, “Запитування опитування”, “Відповіді та результати”, “Результати опитування”. Структура таблиці “Користувач» показано так (таблиця 3.1):

Таблиця 3.1 – Структура таблиці “Користувач”

Ім'я поля	Тип і розмір поля	Опис поля
Id_	Int_	Первинний ключ
Surname	varchar	Прізвище
Name	varchar	Ім'я
Patronymic	varchar	По батькові
UserName	varchar	Логін користувача
Password	varchar	Пароль
Status	int	Тип користувача

Таблиця складається з таких полів як Ідентифікатор(id), Ім'я, Прізвище, По батькові, Лог., Пар. і Варіація користувача.

Таблиця “Учень” містить зв'язок з таблицею “Користувач” по ідентифікатору користувача, так як виражає точні дані (таб. 3.2).

Таблиця 3.2 – Структура таблиці “Студент”

Ім'я поля	Тип і розмір поля	Опис поля
Id_	Int_	Первинний ключ
IdUser	int	Id користувача
IdTown	int	Місто
IdUniversity	int	Університет
IdFaculty	int	Факультет
IdGroup	int	Група
Course	int	Курс

В таблиці “Викладач” знаходиться зв’язок з таблицею “Користувач(Студент)” по ідентифікатору(id) користувача, так як виражає точні дані, що стосуються будь-якого викладача (таблиця 3.3).

Таблиця 3.3 – Структура таблиці “Викладач”

Ім'я поля	Тип і розмір поля	Опис поля
Id_	Int_	Первинний ключ
IdUser	int	Id користувача
IdTown	int	Місто
IdUniversity	int	Університет
IdFaculty	int	Факультет
Position	int	Посада
AcademicTitle	int	Вчене звання

В таблиці “Тести” знаходяться поля Ідентифікатор(id), Name, Автор, Предмет і Кількість питань. Має зв’язок з таблицею “Викладачі” по ідентифікатору(id) автора (таблиця 3.4).

Таблиця 3.4 – Структура таблиці “Тести”

Ім'я поля	Тип і розмір поля	Опис поля
Id_	Int_	Первинний ключ
Name	varchar	Назва
IdAuthor	int	Автор
IdSubject	int	Предмет
AmountQuestions	int	Кількість питань

Таблиця "Опитування" містить такі поля, як ідентифікатор, назва, автор, тематика та кількість запитань. Ця таблиця пов'язана з таблицею "Викладач" за допомогою ідентифікатора автора для відображення детальних даних, пов'язаних з опитуваннями, створеними викладачем (Таблиця 3.5).

Таблиця 3.5 – Структура таблиці “Опитування”

Ім'я поля	Тип і розмір поля	Опис поля
Id_	Int_	Первинний ключ
Name	varchar	Назва
IdAuthor	int	Автор
IdSubject	int	Предмет
AmountQuestions	int	Кількість питань

В таблиці “Запитання тесту” знаходяться Ідентифікатор(id) поля, Ідентифікатор(id) тесту, Запитання, Правильні відповіді. Ця таблиця зв'язана з таблицею “Тест” по ідентифікатору(id) тесту: (таблиця 3.6).

Таблиця 3.6 – Структура таблиці “Питання тесту”

Ім'я поля	Тип і розмір поля	Опис поля
Id_	Int_	Первинний ключ
IdTest	int	Ідентифікатор тесту
TextQuestion	varchar	Текст питання
VariantA	varchar	Варіант А
VariantB	varchar	Варіант В
VariantC	varchar	Варіант С
VariantD	varchar	Варіант D
RightAnswer	int	Правильна відповідь

В таблиці “Питання опитування” знаходяться поля Ідентифікатор(id) поля, Ідентифікатор(id) опитування, Питання, Правильні відповіді. Таблиця зв'язана з таблицею “Опитування” по ідентифікатору(id) опитування: (таблиця 3.7).

Таблиця 3.7 – Структура таблиці “Питання опитування”

Ім'я поля	Тип і розмір поля	Опис поля
Id_	Int_	Первинний ключ
IdPoll	int	Ідентифікатор тесту
TextQuestion	varchar	Текст питання
VariantA	varchar	Варіант А
VariantB	varchar	Варіант В
VariantC	varchar	Варіант С

Таблиця “Результати тесту” знаходяться Ідентифікатор(id) поля, Ідентифікатор(id) тесту, Студент та Кількість правильних питань. (таблиця 3.8).

Таблиця 3.8 – Структура таблиці “Результати тесту”

Ім'я поля	Тип і розмір поля	Опис поля
Id_	Int_	Первинний ключ
IdTest	int	Ідентифікатор тесту
IdStudent	int	Студент
AmountRightAnswers	int	Кількість правильних відповідей

3.6 Методика роботи користувача

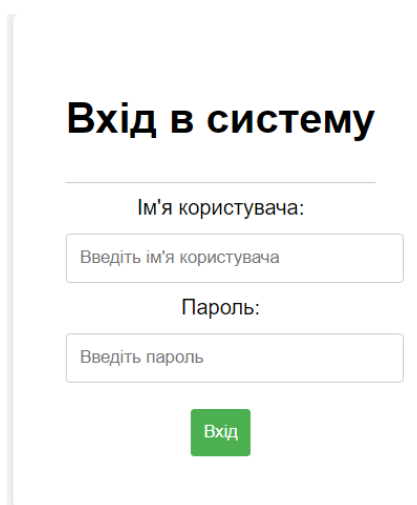
Програмний комплекс побудований з використанням сучасних веб-технологій для використання у сумісних веббраузерах, що підтримують актуальні веб-стандарти.

Оскільки цей комплекс працює на основі вебтехнологій, його не потрібно встановлювати на користувацький пристрій. Проте для використання комплексу необхідний веббраузер, який підтримує сучасні вебстандарти. Також необхідно мати стабільне інтернетпідключення з мінімальною швидкістю 50 мегабіт на секунду. Щоб користуватися функціоналом програми для проведення тестів і

опитувань, необхідно адати відповідний дозвіл клієнтському додатку.

Для початку роботи в системі, користувач повинен авторизуватися на клієнтському додатку. На рисунку 3.4 показана форма авторизації, де користувач вводить свої облікові дані для входу до системи.

Будь ласка, зверніть увагу, що цей текст був згенерований штучним інтелектом і може містити загальну інформацію, а не конкретні деталі вашого програмного комплексу.



Вхід в систему

Ім'я користувача:

Введіть ім'я користувача

Пароль:

Введіть пароль

Вхід

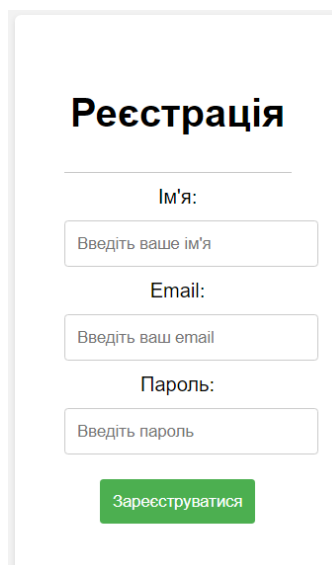
Рисунок 3.4 – Форма авторизації

Якщо користувач раніше не користувався системою або бажає створити новий профіль, йому потрібно пройти процедуру реєстрації. На рисунку 3.5 зображена форма реєстрації, де користувач повинен ввести необхідні дані для створення облікового запису.

Форма реєстрації може містити такі поля, як ім'я, прізвище, електронна пошта, пароль та інші важливі дані, які вимагаються для створення профілю в системі. Користувач повинен заповнити всі необхідні поля згідно вказаних вимог.

Після успішної реєстрації користувач отримує доступ до системи та може використовувати її функціонал для проведення тестів і опитувань.

Зверніть увагу, що це загальна інформація, і форма реєстрації в реальному програмному комплексі може мати інші специфічні поля та вимоги.



Реєстрація

Ім'я:
Введіть ваше ім'я

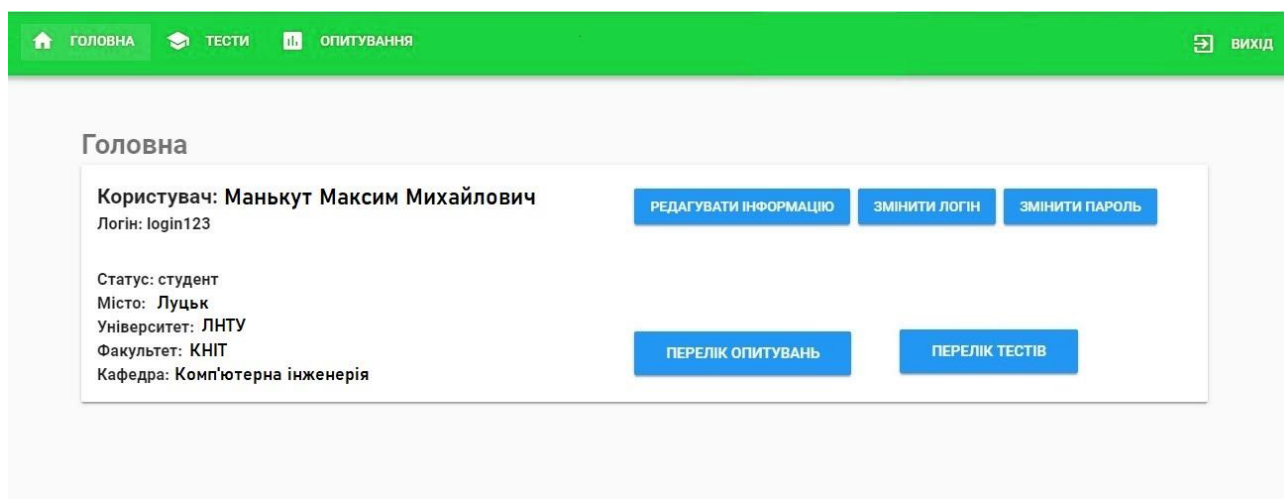
Email:
Введіть ваш email

Пароль:
Введіть пароль

[Зареєструватися](#)

Рисунок 3.5 – Форма реєстрації

Після авторизації користувач отримує доступ до головного меню системи, з якого він може відкривати різні сторінки, перейти до свого кабінету користувача або вийти з профілю. Рисунок відображає головне меню системи (рис. 3.6).



ГОЛОВНА ТЕСТИ ОПИТУВАННЯ ВИХІД

Головна

Користувач: Манькут Максим Михайлович
Логін: login123

РЕДАГУВАТИ ІНФОРМАЦІЮ ЗМІНИТИ ЛОГІН ЗМІНИТИ ПАРОЛЬ

Статус: студент
Місто: Луцьк
Університет: ЛНТУ
Факультет: КНІТ
Кафедра: Комп'ютерна інженерія

ПЕРЕЛІК ОПИТУВАНЬ ПЕРЕЛІК ТЕСТІВ

Рисунок 3.6 – Головне меню системи

У системі доступні різні функції, включаючи:

– Тести: Перегляд результатів пройдених тестів, створення нових тестів та їх проходження (залежно від типу користувача).

– Опитування: Перегляд результатів пройдених опитувань, створення нових опитувань та їх проходження (залежно від типу користувача).

– Користувачі: Адміністратор може реєструвати нових викладачів і студентів, переглядати список зареєстрованих користувачів, змінювати їх дані або видаляти облікові записи.

Ці пункти меню надають користувачам можливість виконувати різноманітні завдання і взаємодіяти з системою залежно від їх ролі і дозволів. записи.

Для того щоб переглянути список пройдених тестів потрібно натиснути на кнопку “Тести” (рис. 3.7).

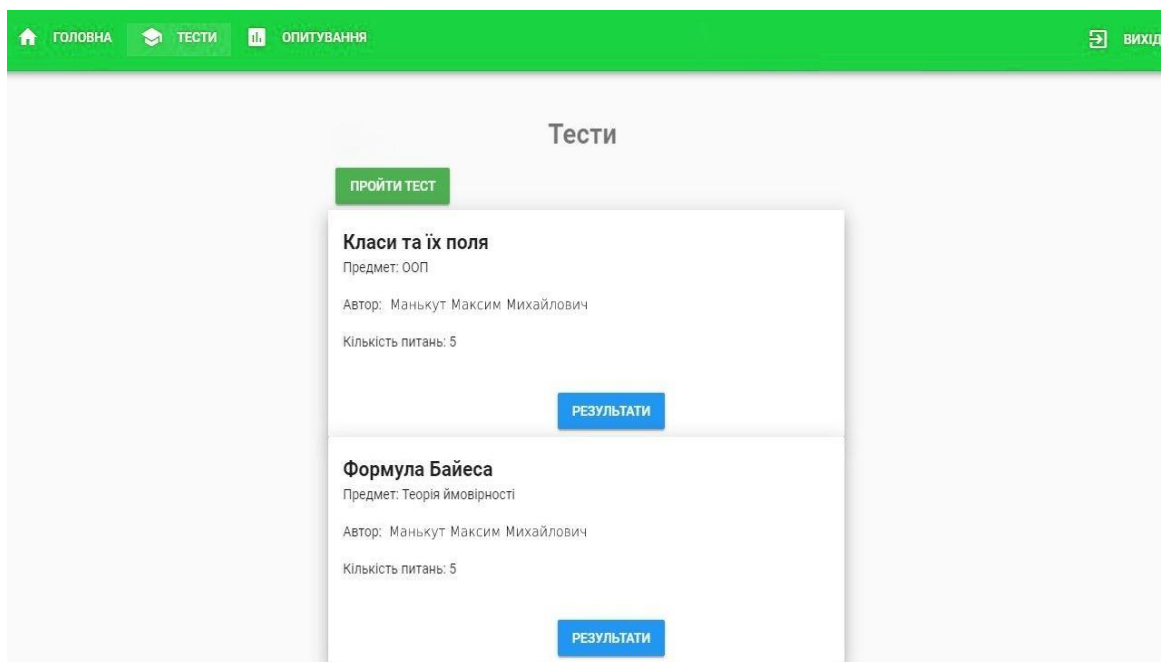


Рисунок 3.7 - Пункт меню “Тести”

Для того, щоб переглянути результати вже пройденого тесту, користувач може скористатися кнопкою "Результати", розташованою на елементі списку.

Для того, щоб створити новий тест, викладачам доступна кнопка «Створити тест», яка дозволяє їм швидко створити новий тест і налаштувати його параметри. Ця кнопка доступна тільки системам викладачів. Студентам, які бажають пройти новий тест, доступна кнопка «Пройти тест». Вони можуть натиснути на цю кнопку, щоб розпочати проходження тесту. Ця кнопка доступна лише студентам. Для того, щоб переглянути список пройдених опитувань, можна натиснути кнопку «Опитування», яка відображається на екрані (рис. 3.8).

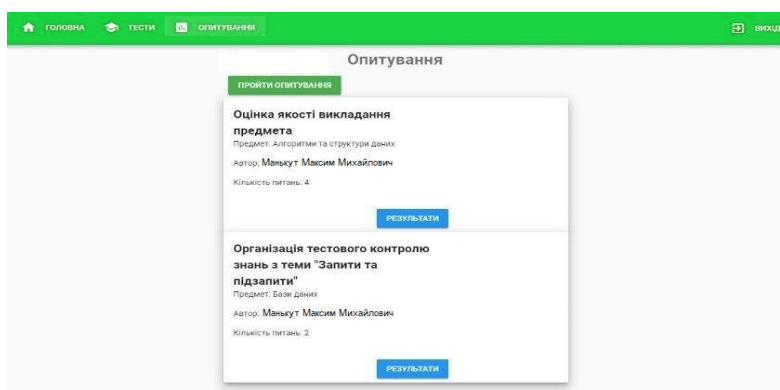


Рисунок 3.8 - Пункт меню “Опитування”

Для перегляду результатів вже пройденого опитування, користувачу потрібно натиснути кнопку "Результати" на елементі списку, пов'язаному з цим опитуванням.

Для створення нового опитування викладачеві доступна кнопка "Створити опитування", яка йому доступна ексклюзивно. Ця кнопка дозволяє викладачеві швидко створити нове опитування та налаштувати його параметри.

Студентам, які бажають пройти нове опитування, доступна кнопка "Пройти опитування". Вони можуть натиснути цю кнопку, щоб розпочати проходження опитування. Ця кнопка доступна тільки студентам.

Для перегляду списку зареєстрованих викладачів та студентів існує кнопка "Опитування" (рис. 3.9). Цю кнопку може натиснути тільки адміністратор системи, і вона дає доступ до перегляду списку зареєстрованих користувачів.

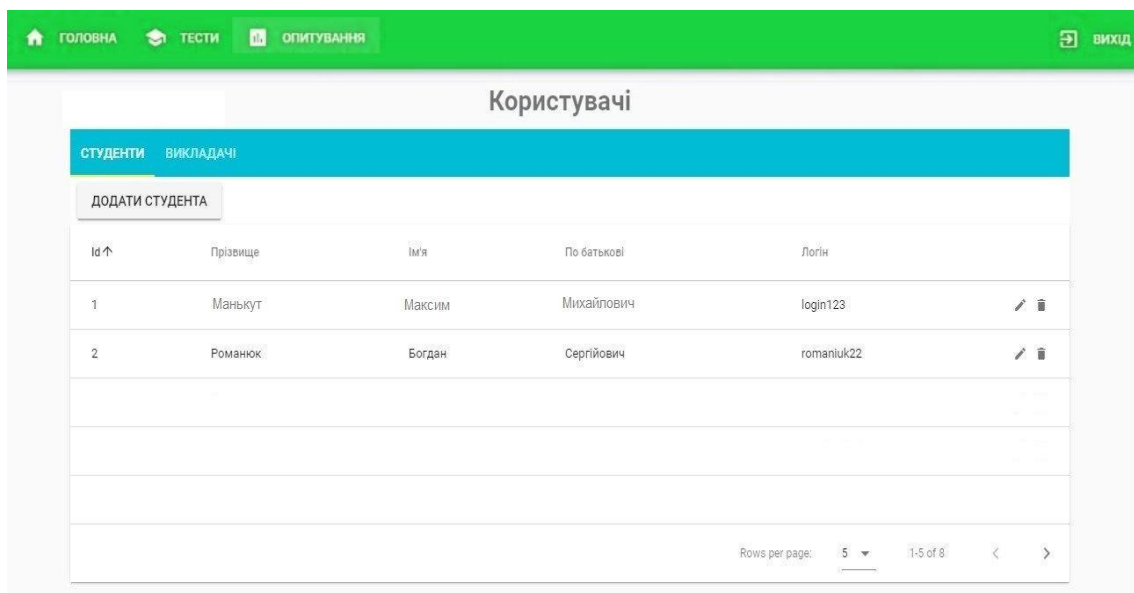


Рисунок 3.9 - Пункт меню “Користувачі”

Адміністратор має можливість редагувати інформацію про користувача, натиснувши на кнопку редагування, яка розташована поряд з відповідним рядком користувача. Для видалення облікового запису користувача адміністратор повинен натиснути на кнопку видалення, яка також знаходиться поряд з рядком користувача. На рисунку 3.10 представлена форма створення опитування.

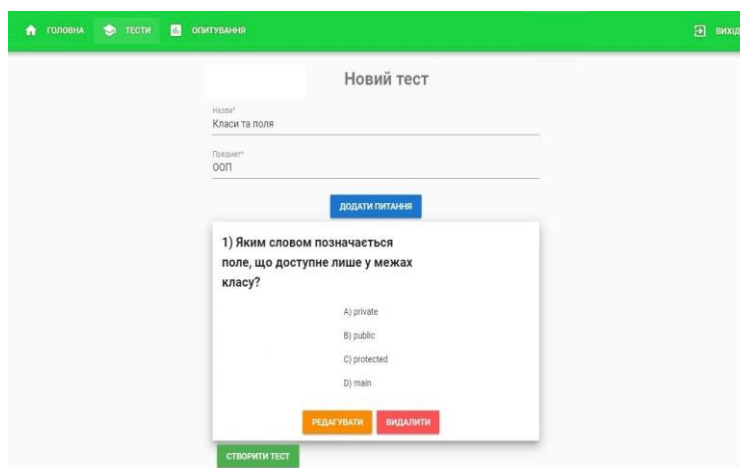


Рисунок 3.10 – Створення тесту (опитування)

Щоб додати нове питання до тесту чи опитування, натисніть кнопку "Додати питання". Для редагування питання натисніть кнопку "Редагувати" поряд з відповідним питанням у списку, а для видалення питання - кнопку "Видалити".

Якщо всі питання до тесту готові, натисніть кнопку "Створити тест" ("Створити опитування"). Щоб знайти необхідний тест або опитування для проходження, введіть всі необхідні дані до форми та оберіть знайдений тест чи опитування. На рисунку 3.11 зображено вікно проходження тесту (опитування).

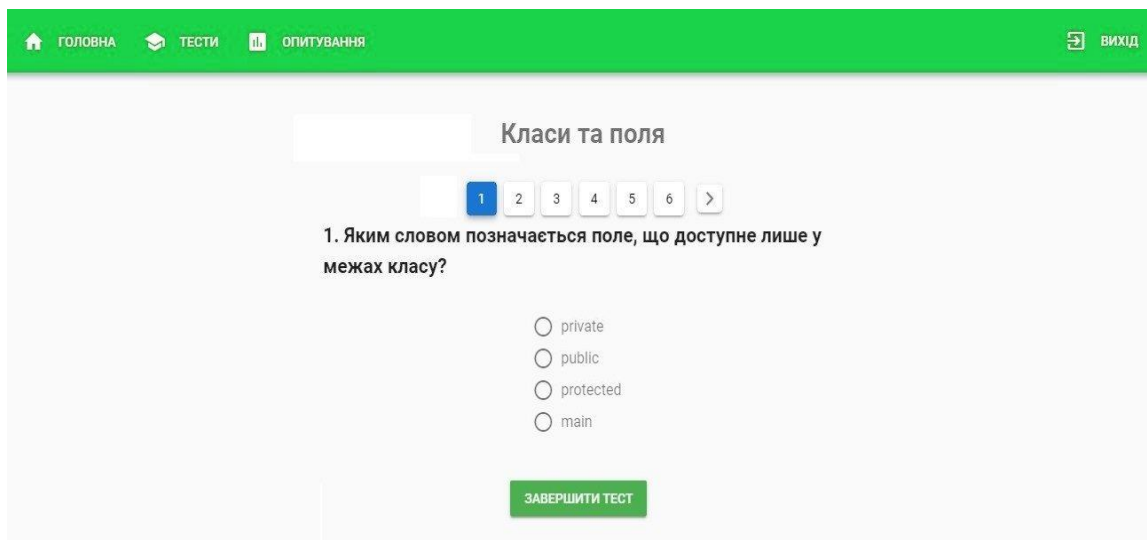


Рисунок 3.11 - Проходження тесту (опитування)

Для відповіді на питання достатньо обрати один з варіантів відповіді, натиснувши на радіокнопку, що знаходиться поряд з ним. Для переходу до іншого питання потрібно натиснути на номер цього питання. Щоб завершити тест, необхідно натиснути кнопку "Завершити тест".

ВИСНОВКИ

Під час дослідження існуючого програмного забезпечення для організації тестування та опитування у навчальному процесі були детально розглянуті системи, які вирішують схожі завдання. Наш розроблений програмний продукт забезпечує автоматизацію процесів тестування та опитування, дозволяючи здійснювати їх швидко та ефективно.

Детально проаналізовано різні методи та засоби розробки програмних систем і прийшли до висновку, що оптимальним рішенням є створення веб-орієнтованої системи з трьома рівнями архітектури. Це дозволяє нам забезпечити гнучкість та зручність як під час розробки, так і у подальшому використанні системи.

Результати тестових завдань підтвердили правильність отриманих результатів, що свідчить про повну відповідність системи всім поставленим вимогам.

В системі можуть працювати різні категорії користувачів, включаючи викладачів, які задають питання для тестів і опитувань, студентів які їх проходять, а також адміністраторів веб-ресурсу, які мають можливість змінювати інформацію про викладачів та студентів. Наше програмне забезпечення може працювати на будь-якій операційній системі, яка підтримує браузер, що відповідає сучасним веб-стандартам та має стабільний доступ до Інтернету.

Цей проект дозволив значно поглибити наші знання про різноманітні технології, які використовуються в процесі розробки програмного забезпечення. Крім того, ми створили декілька прототипів, які допомогли нам розв'язати різні аспекти задачі і послужили основою для розробки кінцевого програмного продукту. Завдяки цьому ми маємо потужний та надійний інструмент для покращення навчального процесу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методологічні та методичні основи створення і використання електронних засобів навчального призначення. URL: https://lib.iitta.gov.ua/864/1/New_PDF_Document.pdf (дата звернення: 02.01.2023).
2. Тестові завдання для контролю знань в курсі "Обчислювальна техніка і технічні засоби навчання". URL: <http://kukh.ho.ua/kurs/zao/lit/Kukh.pdf> (дата звернення: 25.12.2022).
3. Про технологію створення системи тестування. URL: <https://naurok.com.ua/stattya-vikoristannya-suchasnih-testovih-tehnologiy-dlya-perevirki-znan-uchniv-na-urokah-273884.html> (дата звернення: 08.03.2023).
4. What is REST? URL: <http://www.restapitutorial.com/lessons/whatisrest.html> (дата звернення: 07.03.2022).
5. DOM Scripting: Web Design with JavaScript and the Document Object Model. URL: <http://surl.li/hthqb> (дата звернення: 05.01.2023).
6. Створення і використання веб-ресурсів у навчальному процесі. URL: <https://cutt.ly/PweiKbdI> (дата звернення: 11.02.2023).
7. Основи створення веб-сайтів. URL: <https://naurok.com.ua/stvorennya-vebsaytiv-posibnik-dlya-vchitalya-332550.html> (дата звернення: 01.03.2023).
8. Веб-дизайн і програмування веб-сторінок. URL: http://school1k24.at.ua/10CLASS_WEB/OsnovyWebDis.pdf (дата звернення: 08.02.2023).
9. Розробка веб-ресурсів на HTML та CSS. URL: <http://surl.li/hczq> (дата звернення: 07.01.2023).
10. Веб-дизайн і програмування. URL: <https://cutt.ly/wweiLIZi> (дата звернення: 05.02.2023).
11. Веб-дизайн: підручник. URL: <http://dspace.onua.edu.ua/handle/11300/11777> (дата звернення: 01.02.2023).

ДОДАТКИ

Додаток А

Програмний модуль перегляду тестів

```
<template>
  <v-container>
    <v-layout row>
      <v-flex xs12 sm6 offset-sm3>
        <h1 class="text--secondary mb3">Тести</h1>
        <v-btn
          :to="'/tests/new'"
        >Створити тест</v-btn>
        <v-btn class="success" :to="'/tests/search'">Пройти тест</v-btn>
      <v-card
        class="elevation-10 mb2"
        v-for="(test, i) in uniqueTests"
        :key="i"
      >
        <v-layout>
          <v-flex xs8>
            <v-card-text>
              <h2 class="text--primary">{{ test.name }}</h2>
              <p>Предмет: {{ test.subject }}</p>
              <p>Автор: {{ test.author }}</p>
              <p>Кількість питань: {{ test.amountQuestions }}</p>
            </v-card-text>
            <v-card-actions>
              <v-spacer></v-spacer>
              <v-btn
                class="info"
                :to="'/test/' + test.id + '/result'"
              >Результати</v-btn>
            </v-card-actions>
          </v-flex>
        </v-layout>
      </v-card>
    </v-flex>
  </v-layout>
</v-container>
</template>
<script>
export default {
  data() {
    return {}
  },
  computed: {
    uniqueTests() {
      // Вибір унікальних тестів, щоб уникнути повторень
      const tests = this.$store.getters.tests;
      const uniqueTests = [];
      const testIds = [];
      for (const test of tests) {
        if (!testIds.includes(test.id)) {
          uniqueTests.push(test);
          testIds.push(test.id);
        }
      }
      return uniqueTests;
    }
  }
}
</script>
```

Програмний модуль перегляду опитувань

```
<template>
  <v-container>
    <v-layout row>
      <v-flex xs12 sm6 offset-sm3>
        <h1 class="text--secondary mb3">Опитування</h1>
        <v-btn
          :to="'/polls/new'"
          >Створити опитування</v-btn>
        <v-btn class="success">Пройти опитування</v-btn>
        <v-card
          class="elevation-10 mb2"
          v-for="(poll, i) in uniquePolls"
          :key="i"
          >
          <v-layout>
            <v-flex xs8>
              <v-card-text>
                <h2 class="text--primary">{{ poll.title }}</h2>
                <p>Предмет: {{ poll.subject }}</p>
                <p>Автор: {{ poll.author }}</p>
                <p>Кількість питань: {{ poll.amountQuestions }}</p>
              </v-card-text>
              <v-card-actions>
                <v-spacer></v-spacer>
                <v-btn
                  class="info"
                  :to="'/poll/' + poll.id"
                  >Результати</v-btn>
              </v-card-actions>
            </v-flex>
          </v-layout>
        </v-card>
      </v-flex>
    </v-layout>
  </v-container>
</template>
<script>
  export default {
    data() {
      return {}
    },
    computed: {
      uniquePolls() {
        // Вибір унікальних опитувань, щоб уникнути повторень
        const polls = this.$store.getters.polls;
        const uniquePolls = [];
        const pollIds = [];
        for (const poll of polls) {
          if (!pollIds.includes(poll.id)) {
            uniquePolls.push(poll);
            pollIds.push(poll.id);
          }
        }
        return uniquePolls;
      }
    }
  }
</script>
```


Програмний модуль створення тесту

```
<template>
  <v-container>
    <v-layout row v-if="!loading">
      <v-flex xs12 sm6 offset-sm3>
        <h1 class="text--secondary mb-3">Новий тест</h1>
        <v-form
          ref="form"
          v-model="valid">
          <v-text-field
            name="title"
            label="Назва*"
            type="text"
            required
            :rules="[v => !!v || 'Це поле обов\`язкове для введення']"
            v-model="title"></v-text-field>
          <v-text-field
            name="amountQuestions"
            label="Кількість питань*"
            type="number"
            min="1"
            step="1"
            required
            :rules="[v => !!v || 'Це поле обов\`язкове для введення']"
            v-model="amountQuestions"></v-text-field>
          <EditTestQuestion :question="question"/>
        </v-form>
      <v-layout>
        <v-flex xs12>
          <v-card
            class="elevation-10 mb2"
            v-for="(question, i) in questions"
            :key="i"
            >
            <v-layout>
              <v-flex xs8>
                <v-card-text>
                  <h2>{{i + 1}}) {{question.name}}</h2>
                  <p></p>
                  <p>A) {{question.variantA}}</p>
                  <p>B) {{question.variantB}}</p>
                  <p>C) {{question.variantC}}</p>
                  <p>D) {{question.variantD}}</p>
                </v-card-text>
                <v-card-actions>
                  <v-btn class="warning">Редагувати</v-btn>
                  <v-btn class="error">Видалити</v-btn>
                </v-card-actions>
              </v-flex>
            </v-layout>
          </v-card>
          <v-spacer></v-spacer>
          <v-btn
            :disabled="!valid"
            class="success"
            @click="createTest">Створити тест</v-btn>
        </v-flex>
      </v-layout>
    </v-flex>
  </v-layout>
</v-progress-circular :value="100"></v-progress-circular>
```

```

</v-layout>
</v-container>
</template>
<script>
import EditTestQuestion from "../TestQuestions/EditTestQuestion"
import axios from "axios"

export default {
  data() {
    return {
      title: '',
      amountQuestions: '1',
      select: null,
      subjects: [
        'Item1',
        'Item2'
      ],
      valid: false,
      questions: []
    }
  },
  components: {
    EditTestQuestion
  },
  computed: {
    loading() {
      return this.$store.getters.loading
    }
  },
  mounted() {
    axios("http://testandpoll.ua/api.php?action=read&obj=questions")
      .then(response => {
        this.questions = response.data;
      })
      .catch(error => {
      })
  },
  methods: {
    createTest() {
      if(this.$refs.form.validate()) {
        const test = {
          title: this.title,
          select: this.select,
          amountQuestions: this.amountQuestions
        }
      }
    }
  }
}
</script>

```

Програмний модуль керування запитом до бази даних

```

<?php
$db = mysql_connect('localhost', 'root', '') or die('');
mysql_select_db('test_and_poll') or die('');
$res = array('error', false);

$action = 'read';
$obj = 'tests';
if(isset($_GET['action'])) {
  $action = $_GET['action'];
}
if(isset($_GET['obj'])) {
  $obj = $_GET['obj'];
}

```

```

if ($action == 'read' && $obj == 'tests') {
61
$result = mysql_query("SELECT * FROM `tests`");
$tests = array();

while ($row = mysql_fetch_assoc($result)){
array_push($users, $row);
}
$res['tests'] = $tests;
echo json_encode($res['tests']);
}
if ($action == 'read' && $obj == 'polls') {
$result = mysql_query("SELECT * FROM `polls`");
$polls = array();

while ($row = mysql_fetch_assoc($result)){
array_push($users, $row);
}
$res['polls'] = $polls;
echo json_encode($res['polls']);
}
if ($action == 'create' && $obj == 'tests') {
$name = $_POST['name'];
$subject = $_POST['subject'];
var_dump($_POST);

$result = mysql_query("INSERT INTO `tests`(`name`, `subject`) VALUES ('$name',
'$subject') ");
if ($result) {
$res['message'] = "Test Added successfully";
$res['name'] = $name;
$res['subject'] = $subject;
} else{
$res['error'] = true;
$res['message'] = "Insert Test fail";
}
echo json_encode($res);
}
if ($action == 'create' && $obj == 'polls') {
$name = $_POST['name'];
$subject = $_POST['subject'];
var_dump($_POST);

$result = mysql_query("INSERT INTO `polls`(`name`, `subject`) VALUES ('$name',
'$subject') ");
if ($result) {
$res['message'] = "Poll Added successfully";
$res['name'] = $name;
$res['subject'] = $subject;
} else{
$res['error'] = true;
$res['message'] = "Insert Poll fail";
}
echo json_encode($res);
}
if ($action == 'search' && $obj == 'tests') {
$name = $_POST['name'];
$subject = $_POST['password'];

$result = mysql_query("SELECT * FROM `tests` WHERE `name` = '$name' AND `subject`
= '$subject'");
if ($result) {
$result = [
'name' => $result['name'],
'subject' => $result['subject']

```

```
];
$res['message'] = "Test Search success";
$res_int = mysql_numrows($result);
} else{
$res['error'] = true;
$res['message'] = "Test Search failed";
}
echo json_encode($result);
}
62
if ($action == 'search' && $obj == 'polls') {
$name = $_POST['name'];
$subject = $_POST['password'];

$result = mysql_query("SELECT * FROM `polls` WHERE `name` = '$name' AND `subject`
= '$subject'");
if ($result) {
$result = [
'name' => $result['name'],
'subject' => $result['subject']
];
$res['message'] = "Poll Search success";
$res_int = mysql_numrows($result);
} else{
$res['error'] = true;
$res['message'] = "Poll Search failed";
}
echo json_encode($result);
}
?>
```