

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ТЕСТУВАННЯ ДЛЯ
ПЕРЕВІРКИ ЗНАНЬ
SOFTWARE IMPLEMENTATION OF A KNOWLEDGE TESTING SYSTEM

Виконав: студент(ка) 4 курсу, групи КНТ-217
Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

Дерев'яно А.С.

(прізвище та ініціали)

Керівник Олійник А.О.

(прізвище та ініціали)

Рецензент Скрупський С.Ю.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2021 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА(КИ)

Дерев'янка Артема Сергійовича
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Програмна реалізація системи тестування для перевірки знань
Software Implementation of a Knowledge Testing System

керівник проекту (роботи) Олійник Андрій Олександрович, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року № 103

2. Строк подання студентом проекту (роботи) 13 травня 2021 року

3. Вихідні дані до проекту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Розробка архітектури програми. 3. Основні рішення щодо реалізації компонентів системи. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Олійник А.О., доцент		
Нормоконтролер	Андрєєв М.О., асистент		

7. Дата видачі завдання “10” березня 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз та дослідження предметної області	2-4 тижні	Розділи 1, 2
3	Розробка архітектури програми	5 тиждень	Розділ 2
4	Розробка програми	6-7 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми	8 тиждень	Розділ 4
8	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	9 тиждень	Додатки
9	Захист роботи	10 тиждень	

Студент(ка)

(підпис)

Дерев'янко А.С.

(прізвище та ініціали)

Керівник проекту (роботи)

(підпис)

Олійник А.О.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 91 с., 42 рис., 3 табл., 5 дод., 25 джерел.

ОНЛАЙН ТЕСТУВАННЯ, МОНІТОРИНГ, ЗДОБУВАЧІ ОСВІТИ, ВИКЛАДАЧ, РОЗРОБКА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес онлайн-тестування.

Предмет дослідження – методи організації онлайн-тестування.

Метою роботи є розроблення програмного забезпечення для організації та проведення онлайн-тестування здобувачів освіти.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова програмування PHP, система керування базою даних MySQL.

Результати. Розроблено програмне забезпечення у вигляді веб-застосунку для організації та проведення онлайн-тестування.

Наукова новизна роботи полягає в розробці кроссплатформного веб-застосунку для організації та проведення онлайн-тестування здобувачів освіти за різними дисциплінами з використанням різних форм тестових завдань.

Висновки. Розроблено вебзастосунок для організації та проведення онлайн-тестування.

Галузь використання – заклади середньої та вищої освіти.

ABSTRACT

Explanatory note to the final qualifying work of the bachelor: 91 pages, 42 figures, 3 tables, 5 appendixes, 25 sources.

ONLINE TESTING, MONITORING, EDUCATION APPLICANTS, TEACHER, DEVELOPMENT, SOFTWARE.

The object of research is the online testing process.

The subject of the research is methods of organizing online testing.

The aim of the work is to develop software for organizing and conducting online testing of educational applicants.

Materials, methods, and technical tools: Object-Oriented Programming, PHP programming language, MySQL database management system.

Results. Software has been developed in the form of a web application for organizing and conducting online testing.

The scientific novelty of the work consists in the development of a cross-platform web application for organizing and conducting online testing of educational applicants in various disciplines using various forms of test tasks.

Conclusions. A web application for organizing and conducting online testing has been developed.

Field of usage: institutions of secondary and higher education.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Основні принципи тестування. Використання тестування як форми контролю знань	11
1.1.1 Форми тестування.....	12
1.2 Огляд існуючих рішень	13
1.2.1 Освітній проект «На урок».....	14
1.2.2 Сайт Proprofs.com	15
1.2.3 Розширення Google Форми	16
1.2.4 Порівняння існуючих аналогів	17
1.3 Постановка завдання.....	18
1.4 Висновки за розділом 1	18
2 Розробка архітектури програми	20
2.1 Вибір середовища проектування.....	20
2.1.1 Мови програмування для розробки веб-застосунку	20
2.1.2 Системи керування базами даних.....	24
2.1.3 Веб-сервер.....	25
2.1.4 Середовище розробки.....	26
2.2 Загальна структура додатку.....	27
2.3 Висновки за розділом 2.....	30
3 Основні рішення щодо реалізації компонентів системи	32
3.1 Функціонування системи.....	32
3.2 Функціонування серверної частини.....	33
3.3 Проектування бази даних	35
3.3.1 Аналіз предметної області.....	35
3.3.2 Інфологічне проектування.....	35
3.3.3 Фізичне проектування	39

3.4 Функціонування клієнтської частини	40
3.5 Висновки за розділом 3.....	41
4 Експлуатація, тестування та експериментальне дослідження програми	42
4.1 Опис застосування	42
4.2 Умови виконання програми	42
4.3 Інструкція по експлуатації програми.....	43
4.3.1 Характеристики програми.....	43
4.3.2 Перелік файлів, у яких знаходиться програма	43
4.3.3 Звернення до програми.....	43
4.3.4 Вхідні й вихідні дані.....	44
4.3.5 Запуск програми	44
4.4 Методика та результати тестування	44
Висновки.....	49
Перелік джерел посилання	51
Додаток А Технічне завдання.....	54
Додаток Б Керівництво програміста	59
Додаток В Керівництво користувача	63
Додаток Д Текст програми	70
Додаток Е Слайди презентації.....	82

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CSS – Cascading Style Sheets;

GUI – Graphical user interface;

HTML – Hyper Text Markup Language;

PHP – Hypertext Preprocessor;

WWW – World Wide Web.

ЗВО – заклад вищої освіти;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

СКБД – система керування базами даних;

ВСТУП

Оснащеність сучасних закладів вищої освіти (ЗВО) комп'ютерною технікою, наявність власних локальних мереж, доступ до мережі Інтернет дозволяє перейти від традиційних методів оцінки отриманих знань до нових інформаційних технологій.

В даний час, в епоху інформаційного суспільства та вимушеного дистанціювання все більш актуальними стають різноманітні комп'ютерні системи тестування, здатні доповнити або замінити традиційні методи контролю та методики викладання. Завдяки комп'ютерним системам тестування стало набагато зручніше проводити тестування в установах, де застосовувалися і застосовуються звичайні тести. Сьогодні комп'ютерне тестування має низку переваг перед традиційним тестуванням. Воно відрізняється високою оперативністю, продуктивністю процесу тестування і об'єктивністю результатів контролю знань – вчитель може провести опитування набагато більшого числа учнів за менший час у порівнянні з очним опитуванням, і дозволяє проаналізувати якість підготовки по великому колу різних питань; також адаптація змісту і складності тестових питань рівню знань учнів, можливість самоконтролю.

Комп'ютерне тестування, що реалізоване в формі звичайного програмного забезпечення, що вимагає встановлення, налаштування, пристосування до інформаційної системи та інше, з часом будуть витіснені тому, що можливості Інтернету по взаємодії з користувачем доросли до певного рівня, на якому стало можливим створення повноцінних веб-застосунків.

Вебзастосунок – клієнт-серверний застосунок, в якому клієнтом виступає браузер, а сервером – веб-сервер. Логіка веб-застосунку розподілена між сервером і клієнтом, зберігання даних здійснюється, переважно, на сервері, обмін інформацією відбувається по мережі.

Однією з переваг використання веб-застосунків є той факт, що клієнти не залежать від конкретної операційної системи користувача, тому веб-застосунки є міжплатформними сервісами. Саме через це створення таких сервісів сьогодні вважається одним з перспективних напрямків діяльності для багатьох установ чи то комерційних, чи державних, в тому числі і для ЗВО.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основні принципи тестування. Використання тестування як форми контролю знань

Важливою ланкою процесу навчання є контроль знань і умінь учнів. Поступовий перехід від традиційних форм контролю і оцінювання знань до тестування відповідає тенденціям сучасності і загальній концепції модернізації системи освіти [1].

На сьогоднішній день в освітній практиці склалися дві основні форми контролю для перевірки знань учнів: усне опитування і письмова робота. Але усне опитування є вибірковою формою контролю знань окремих учнів, що віднімає значний обсяг дорогоцінного часу від заняття, а письмова робота - надзвичайно трудомістка і не оперативна. Найчастіше вчитель, не встигнувши впоратися з перевіркою робіт учнів, починає наступний урок без інформації про те, які розділи попереднього матеріалу не були засвоєні учнями в достатній мірі. Все це призводить до того, що учні, підготувавши заздалегідь відповіді на усні запитання вчителя, написавши контрольну роботу, відразу начисто забувають минулі теми. Процес підручник - дошка - вчитель поступово втрачає свою ефективність і об'єктивність при навчанні.

Вільної від цих недоліків є форма контролю у вигляді тестових завдань. Розглядаючи тестування, як спосіб розвитку навчально-пізнавальної діяльності, доцільно зупинитися на понятті тесту.

Тест - це система завдань специфічної форми зростаючої складності, що дозволяє оцінити структуру і виміряти рівень знань з певної навчальної дисципліни [2].

Розвиток нових інформаційних технологій дозволило організувати навчальний процес на більш високому рівні. Застосування комп'ютерних технологій для оперативного контролю знань учнів з використанням тестових завдань (онлайн-тестування) відкрило для них нові горизонти в освоєнні предметів.

У сфері освіти йде безперервний пошук нових форм організації процесу навчання, що забезпечують високу надійність навчання, ефективність засвоєння елементів навчально-пізнавальної діяльності, розвиток творчих здібностей учнів. Пошук надійних засобів вимірювання та контролю став причиною відродження інтересу до онлайн-тестування, зокрема.

Онлайн-тестування є значним кроком на шляху розвитку методики контролю за засвоєнням навчального матеріалу. Введення онлайн-тестування на заняттях дозволяє здійснити плавний перехід від суб'єктивних і багато в чому інтуїтивних оцінок до об'єктивних обґрунтованих методів оцінки результатів навчання [1].

Тест фіксує тільки результати роботи, але не хід їх виконання, можливо вгадування правильної відповіді, а також випадки, коли вибір неправильної відповіді пояснюється неухважністю студента, тому раціональніше поєднувати тестування з різними формами традиційного контролю. Таким чином, описані вище види контролю якості знань студентів, зручно використовувати і при організації самостійної роботи студентів в режимі самоконтролю, при повторенні навчального матеріалу[3].

1.1.1 Форми тестування

При створенні тестових завдань використовують їх різні форми. Рекомендованими міністерством освіти та науки України для застосування є тестові завдання закритої та відкритої форм.

Тестові завдання закритої форми пропонують вибрати одну правильну відповідь із декількох запропонованих.

У таких тестових завданнях відповіді необхідно розміщувати у певному порядку. При комп'ютерному тестуванні доцільно використовувати саме завдання закритої форми, оскільки їх легко опрацьовувати.

Окрім того, бажано щоб при створенні тестових завдань використовувалось якнайменше їх різновидів [4].

До тестів закритої форми можна віднести:

- вибір з множини. цей тип завдання потребує вибору однієї або декількох відповідей із запропонованого набору варіантів. Кожен варіант відповіді повинен мати свою вагу оцінювання. Загальна оцінка за завдання складається із оцінок обраних варіантів;

- вибір вірно – не вірно. Цей тип завдання потребує однозначну відповідь (вірно/не вірно) на поставлене питання;

- вибір відповідності. У завданні цього типу на задану тему створюється множина питань і множина правильних відповідей на ці питання. При тестуванні для кожного питання треба обрати із списку відповідну правильну відповідь;

- випадкові завдання короткої відповіді. Цей тип завдання по формі його представлення подібний завданню на вибір відповідності. Він базується на завданнях короткої відповіді, які створені у конкретному розділі.

Тестові завдання відкритої форми дозволяють вільно конструювати відповідь або доповнювати (завершувати) частковий варіант відповіді.

В комп'ютеризованому тестуванні відповіді на завдання відкритої форми повинні мати чітко визначені форму і зміст [4].

1.2 Огляд існуючих рішень

Комп'ютерне тестування як ефективний спосіб перевірки знань знаходить в освіті все більше застосування.

Застосування програм тестування та комп'ютерних тестів в освіті є економічно вигідним і забезпечує підвищення ефективності навчального процесу, об'єктивності оцінки рівня знань і є раціональним доповненням до інших методів перевірки знань. Важливу роль, при використанні комп'ютерного тестування, відіграє вибір програм для тестування знань, розробки і створення тестів. В даній роботі не розглядаються так би мовити

стаціонарне програмне забезпечення, що потребує інсталяції, постійних налаштувань та прив'язки до операційних систем.

Серед сучасних ресурсів Інтернет існує безліч сервісів для створення власних тестів. Кожному з цих сервісів притаманні свої переваги та недоліки для більш глибокого розуміння призначення веб-застосунків для тестування розглянемо декілька прикладів, що вже працюються.

1.2.1 Освітній проект «На урок»

«На урок» – це безкоштовний україномовний ресурс для проектування та реалізації власних електронних тестів [5], [6]. Ресурс надає можливість розробити різні типи тестів для проведення тестування знань, моніторингу успішності вивчення матеріалів та виконання домашнього завдання. До основних можливостей відноситься [5], [6]:

- авторизація із декількома рівнями доступу (учень або вчитель);
- створення та проходження тестів;
- виставлення оцінок;
- «ручна» перевірка результатів.

Головна сторінка проекту для авторизації вчителів наведена на рисунку

1.1.

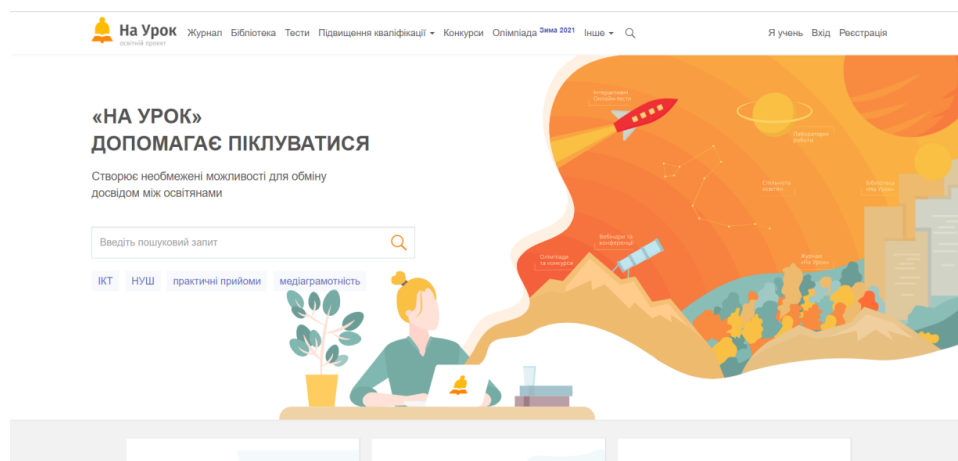


Рисунок 1.1 – Головна сторінка проекту «На урок» для авторизації вчителів

1.2.2 Сайт Proprofs.com

Мабуть, найвідоміший у всьому світі сайт для створення інтерактивних інструментів з перевірки знань або проведенню досліджень [7], [8]. Відмінно підійде викладачам, тренерам і маркетологам. З ним працюють такі корпорації, як Sony, Dell, Cisco, Adobe, Ford і т. д. Один недолік - все це платно. Проте завжди можна спочатку використовувати безкоштовну 7-денну пробну версію[9].

До основних можливостей можна віднести:

- необмежену кількість створюваних тестів;
- зручна статистика та аналітика;
- завантаження картинок, відео та інших файлів;
- створення окремих груп і онлайн –класів;
- брендінг свого облікового запису;
- крім тестів, тут можна створювати опитування, ігри, архіви, флеш - карти та ін.

Головна сторінка проекту наведена на рисунку 1.2.

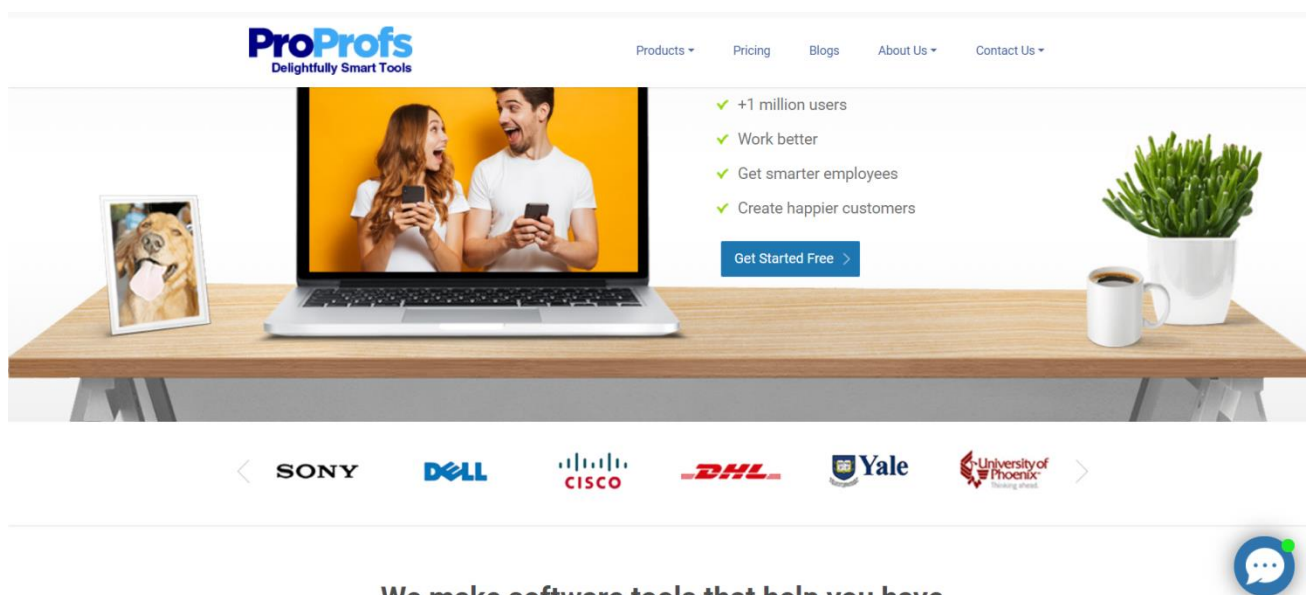


Рисунок 1.2 – Головна сторінка сайту Proprofs.com

1.2.3 Розширення Google Форми

Google Форми – програмне забезпечення для проведення опитувань. Входить до складу безкоштовного веб-пакету Google Docs Editors, пропонованого Google [10], [11]. Проте для навчальних закладів може використовуватися в рамках пакету Google Workspace (G Suite) разом із платформою Google Classroom, наприклад. Форми функціонують у форматі вебзастосунок. Застосунок надає змогу користувачам створювати та редагувати опитування в Інтернеті. Взаємодія із користувачами проходить в режимі реального часу. Зібрана інформація може бути автоматично згенерована в електронну таблицю. Функціонал «Тестування» робить можливим проведення тестів із оцінюванням результатів.

До основних можливостей можна віднести:

- велику кількість створюваних тестів;
- зручна статистика та аналітика;
- завантаження картинок, відео та інших файлів;
- брендінг форм;
- можливість «ручної» перевірки результатів.

Приклад Google Форми для тестування представлено на рисунку 1.3.

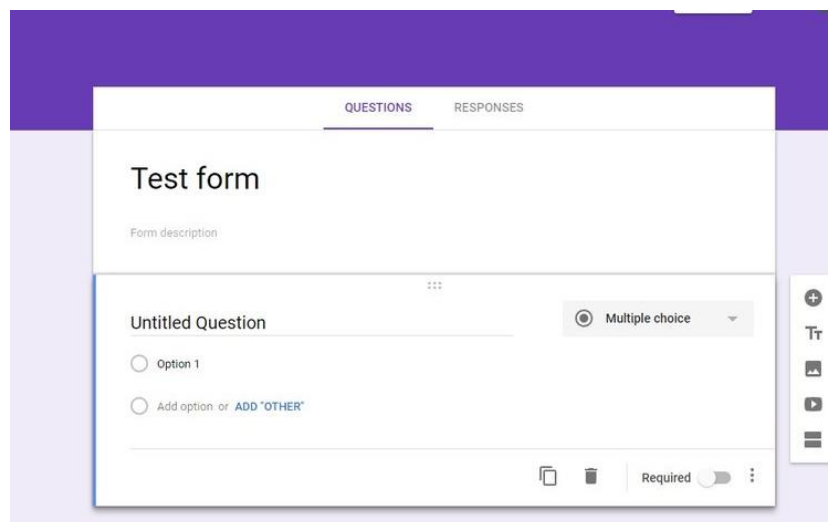


Рисунок 1.3 – Приклад Google Форми для тестування

1.2.4 Порівняння існуючих аналогів

Проведемо порівняння за певними критеріями, хід якого представимо у вигляді таблиці (табл. 1.1). Серед критеріїв виділимо наступні: вартість використання (платний ресурс чи безкоштовний), кроссплатформність, види тестувань (кількість типів тестових запитань та тестів), види перевірки результатів (чи повністю автоматичною є перевірка або можливо вносити певні корективи), підтримка мультимедійних даних для тестування (чи можуть для тестових питань використовуватися зображення, відео- та аудіо-матеріали).

Таблиця 1.1 – Порівняння існуючих аналогів

Критерій порівняння	Проект «На урок»	Сайт Proprofs.com	Google Форми	Онлайн тестування
Вартість	Безкоштовний	Платний (наявна пробна версія)	Безкоштовні з обмеженнями	Безкоштовний
Кроссплатформність	Так	Так	Так	Так
Види тестувань	Середнє	Багато	Середнє	Багато
Види перевірки результатів	Автоматична	Автоматична	Автоматична, ручна	Автоматична, ручна
Підтримка мультимедійних даних для тестування	Так	Так	Так	Так

З результатів порівняння можна зробити висновок, що в більшості випадків ресурси тестування не є повністю безкоштовними, адже, наприклад, Google Форми не дають повноцінних можливостей через обмеження в пам'яті на Google Диску. З іншого боку, ручна перевірка результатів, що може бути необхідна у деяких випадках (наприклад, при перевірці відкритих або теоретичних запитань) також доступне не усюди. Також слід відзначити, що певні ресурси не завжди надають повноцінні можливості для вибору видів

тестувань: іноді за замовченням (проект «На урок»), іноді через те, що з початку не розроблялися як тестові платформи (Google Форми).

Саме тому, можна зробити висновок, що не дивлячись на велику кількість засобів для онлайн-тестувань актуальною задачею залишається розробка такого застосунку, що виправляв би недоліки аналогів та надавав повноцінний функціонал для проведення тестувань.

1.3 Постановка завдання

За результатами аналізу проблемної області та дослідження існуючих аналогів для вирішення проблемних питань, сформулюємо постановку задачі.

Розробити вебзастосунок для контролю знань у формі тестування, застосунок повинен відповідати наступним вимогам:

- проведення тестування в закритій форм;
- тести можуть бути різної структури;
- тести можуть бути організовані на будь-якій мові;
- необмежена кількість тестів, що створюється;
- можливість реєстрації користувача;
- можливість налаштування системи оцінювання;
- зручна статистика.

1.4 Висновки за розділом 1

Виконано огляд проблеми пошуку проведення онлайн-тестування. Виділено основні шляхи для направи для розроблення нового програмного забезпечення (ПЗ) для проведення онлайн-тестувань.

Тест – це система завдань специфічної форми зростаючої складності, що дозволяє оцінити структуру і виміряти рівень знань з певної навчальної дисципліни [9].

У сфері освіти йде безперервний пошук нових форм організації процесу навчання, що забезпечують високу надійність навчання, ефективність засвоєння елементів навчально-пізнавальної діяльності, розвиток творчих здібностей учнів. Пошук надійних засобів вимірювання та контролю став причиною відродження інтересу до онлайн-тестування, зокрема.

При створенні тестових завдань використовують їх різні форми. Тестові завдання закритої форми пропонують вибрати одну правильну відповідь із декількох запропонованих. Тестові завдання відкритої форми дозволяють вільно конструювати відповідь або доповнювати (завершувати) частковий варіант відповіді.

Серед сучасних ресурсів Інтернет існує безліч сервісів для створення власних тестів. Кожному з цих сервісів притаманні свої переваги та недоліки. Проаналізувавши сучасні сервіси, що надають можливості Інтернету можна зробити висновок, що більшість таких сервісів призначенні не для освітніх установ, мають іноземну мову реалізації, потребують коштів для вільного користування всіма можливостями. Одним з найбільших недоліків використання вже готового сервісу тестування для освітніх установ є неможливість прив'язки веб-застосунку до власного сайту ЗВО.

2 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

В першому розділі був проведений аналіз систем онлайн-тестування, а також проаналізовані форми представлення тестів. На основі проведеного аналізу були сформовані основні можливості, які повинні бути притаманні системі, що розробляється, були виявленні основні технічні та програмні особливості.

В результаті аналізу можна зробити висновок, що для розробки необхідно обрати мову програмування для реалізації функціональної частини, засоби для організації візуального оформлення додатку, систему керування базами даних для розробки власної бази даних для централізованого збереження необхідної формації, веб-сервер для реалізації сумісної роботи раніше обраних технологій.

2.1 Вибір середовища проектування

2.1.1 Мови програмування для розробки веб-застосунку

Існує безліч мов, що використовуються при розробці веб-сайтів та веб-застосунків. Їх можна розділити на дві основні групи. До першої групи відносяться мови, код яких виконується на стороні клієнта, тобто на комп'ютері відвідувача сайту. У другу групу потрапляють ті мови, програми на яких виконуються на тому комп'ютері де знаходиться веб-сервер.

Застосунок складається з двох рівноцінних частин: серверної та клієнтської. Можливо використовувати одну і ту ж мову програмування у всьому додатку, однак це не є самим оптимальним варіантом.

Для реалізації клієнтської частини для організації графічного інтерфейсу користувача (GUI) використовують: HyperText Markup Language (HTML) та Cascading Style Sheets (CSS).

HTML – стандартна мова розмітки документів у Всесвітній павутині. Більшість веб-сторінок містять опис розмітки на мові HTML (або XHTML).

Мова HTML інтерпретується браузером і відображається у вигляді документа в зручній для людини формі [12].

CSS – формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки.

CSS використовується розробниками веб-сторінок для завдання кольорів, шрифтів, розташування окремих блоків та інших аспектів представлення зовнішнього вигляду цих веб-сторінок. Основною метою розробки CSS було розділення опису логічної структури веб-сторінки (яке проводиться за допомогою HTML або інших мов розмітки) від опису зовнішнього вигляду цієї веб-сторінки (яке тепер проводиться за допомогою формальної мови CSS). Такий поділ може збільшити доступність документа, надати велику гнучкість і можливість управління його поданням, а також зменшити складність і повторюваність в структурному вмісті [13].

Існує великий вибір мов, що підходять для реалізації серверної частини, серед яких існують визнані лідери в своїх областях. Розглянемо лише деякі мови для створення веб-додатків.

Perl – мова програмування загального призначення, яка була спочатку створена для маніпуляцій з текстом, але на даний момент використовується для виконання широкого спектру завдань, включаючи системне адміністрування, веб-розробку, ігри, біоінформатику, розробку графічних користувацьких інтерфейсів [14].

Мову можна охарактеризувати скоріше як практичну (легкість у використанні, ефективність, повнота), ніж гарну (елегантність, мінімалістичність). Головними перевагами мови є підтримка різних парадигм (процедурний, об'єктно-орієнтований та функціональний стилі програмування), контроль за пам'яттю (без збирача сміття, заснованого на циклах), вбудована підтримка обробки тексту, а також велика колекція модулів сторонніх розробників [14].

PHP – скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов,

що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP – проект відкритого програмного забезпечення [15].

PHP – мова, яка може бути вбудована безпосередньо в html-код сторінок, які, в свою чергу коректно будуть оброблені PHP-інтерпретатором. Механізм PHP просто починає виконувати код після першої екрануючої послідовності і продовжує виконання до того моменту, коли він зустрине парну екрануючу послідовність [15].

Велика різноманітність функцій PHP дають можливість уникнути написання багаторядкових призначень для користувача функцій на C або Pascal.

Наявність інтерфейсів до багатьох баз даних в PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase. через стандарт відкритого інтерфейсу зв'язку з базами даних можна підключатися до всіх баз даних, до яких існує драйвер.

Багато конструкцій мови запозичені з C, Perl. Код PHP дуже схожий на той, який зустрічається в типових програмах на C або Pascal. Це помітно знижує початкові зусилля при вивченні PHP. PHP – мова, що поєднує переваги Perl і C та спеціально спрямована на роботу в Інтернеті, мова з універсальним і зрозумілим синтаксисом. І хоча PHP є досить молодого мовою, вона здобула таку популярність серед web-програмістів, що в наш час є мало не найпопулярнішою мовою для створення веб-застосунків (скриптів).

Важливою перевагою PHP є те, що ця мова належить до інтерпретованих. Це дозволяє обробляти сценарії з достатньо високою швидкістю. За деякими оцінками, більшість PHP-сценаріїв (особливо не дуже великих розмірів) обробляються швидше за аналогічні їм програми, написані на Perl. Проте хоч би що робили розробники PHP, виконувані файли, отримані за допомогою компіляції, працюватимуть значно швидше — в десятки, а іноді і в сотні разів. Але продуктивність PHP достатня для створення цілком серйозних веб-застосунків [15].

Проведемо порівняння мов програмування для реалізації серверної частини веб-застосунку, результати порівняння представимо в табличній формі (табл. 2.1). У якості критеріїв будемо використовувати наступні: швидкість роботи (умовна швидкість роботи технології та мови програмування за оцінками експертів), бібліотеки (у загальному випадку програмні бібліотеки, модулі чи плагіни, що можуть спростити та скоротити час розробки), парадигми програмування (загальна можливість мови підтримувати парадигми програмування, що може скоротити час розробки та зробити код більш зрозумілим для подальшої підтримки), спільнота (критерій для визначення кількості активних розробників, що використовують мову програмування та кількості ресурсів присвячених мові програмування), фреймворки (критерій для оцінки кількості фреймворків, використання яких може скоротити час розробки).

Таблиця 2.1 – Порівняння мов програмування

Критерій	Perl	Java	Python	PHP
Швидкість роботи	–	+	±	+
Бібліотеки	±	±	++	+
Парадигми програмування	±	+	+	+
Спільнота	–	+	+	++
Фреймворки	–	±	+	++

За результатами порівняння можна зробити висновок, що найбільш оптимальним та простим буде використання PHP у якості мови програмування для серверної частини. Не дивлячись на стрімке зростання популярності більш нових мов програмування PHP залишається найбільш популярною мовою програмування при розробці веб-застосунків та підтримується більшою кількістю серверів.

2.1.2 Системи керування базами даних

Система керування базою даних (СКБД) являє собою програмне забезпечення, яке керує всім доступом до бази даних [16].

Для системою, що розробляється надійність і продуктивність СКБД є критичним моментом, оскільки додаток припускає наявність великої кількості записів в базі даних (БД), а також велику кількість користувачів, чії дії так чи інакше будуть пов'язані з маніпулюванням інформації з БД [17].

У світі існує безліч СКБД, огляд їх всіх не входить в рамки цієї випускної роботи. Розглянуто будуть найбільш популярні СКБД, які мають багаторічну історію розробки і підтримки.

Oracle Database або Oracle RDBMS - об'єктно-реляційна система управління базами даних компанії Oracle [18].

Сучасна СУБД Oracle це потужний програмний комплекс, що дозволяє створювати додатки будь-якого ступеня складності. Ядром цього комплексу є база даних, яка зберігає інформацію, кількість якої за рахунок коштів, що надаються масштабування практично безмежно. С високою ефективністю працювати з цією інформацією одночасно може практично будь-яка кількість користувачів (за наявності достатніх апаратних ресурсів), не проявляючи тенденції до зниження продуктивності системи при різкому збільшенні їх числа [19].

MySQL – вільна реляційна система управління базами даних. Продукт поширюється як під GNU General Public License, так і під власною комерційною ліцензією. Крім цього, розробники створюють функціональність за замовленням ліцензійних користувачів. Саме завдяки такому замовленню майже в найраніших версіях з'явився механізм реплікації [20].

MySQL є рішенням для малих і середніх додатків. Входить до складу серверів WAMP, AppServ, LAMP і в портативні збірки серверів Денвер, ХАМРР [21].

Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць : користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Більш того, СУБД MySQL поставляється із спеціальним типом таблиць EXAMPLE, що демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі і GPL – ліцензуванню, в СКБД MySQL постійно з'являються нові типи таблиць[22].

2.1.3 Веб-сервер

Веб-сервер – це сервер, що приймає HTTP–запити від клієнтів і викликає HTTP–відповіді, як правило, разом з HTML–сторінкою, зображенням або мультимедійним об'єктом. Говорячи про веб-сервері, мають на увазі як ПО, так і окремий комп'ютер, на якому це ПЗ встановлене.

Клієнт (веб -браузер) подає веб-серверу запити на отримання ресурсів, що позначаються URL–адресами [23].

Існує багато веб-серверів. Розглянемо лише деякі з них:

- Nginx – високопродуктивний HTTP-сервер, призначений в основному для роздачі клієнтам статичного контенту (зображень, javascript-файлів, css-стилів і т.д.). В основі лежить технологія неблокуючих з'єднань, що при великій кількості одночасних підключень істотно заощаджує ресурси сервера;

- Apache HTTP-сервер – вільний веб-сервер. За статистикою на травень 2018 року використовується в 46 % всіх веб-серверів, що робить його найпопулярнішим веб-сервером. Версія 2.2.x володіє перевіреною роками і мільйонами користувачів стабільністю і надійністю. Сервер має велику кількість модулів для роботи з багатьма серверними технологіями.

2.1.4 Середовище розробки

PhpStorm – це інтегроване середовище розробки при розробці на PHP з інтелектуальним редактором, яка глибоко аналізує код, підтримує основні версії PHP (7.0, 5.6 та більш старі) для сучасних і класичних проектів, забезпечує краще в індустрії автодоповнення коду, рефакторинги, запобігання помилок нальоту і підтримує змішування мов [24], [25].

Програмне забезпечення PhpStorm являє собою спеціалізований засіб веб-розробки, орієнтовані на веб-застосунки та інші види програм, які можна створювати за допомогою мови PHP і з використанням HTML, JavaScript і CSS. Рішення PhpStorm здійснює розгортання і синхронізацію проектів через протокол FTP. Середовище PhpStorm пропонує функції автоматичного завершення мовних конструкцій PHP в коді, інспектування коду, різні алгоритми рефакторингу і швидку навігацію по коду [24].

Реалізований в PhpStorm графічний PHP-відладчик підтримує умовні точки зупинки, відстеження значень і автоматизований вхід в налагодження окремих процедур. Для тестування додатків підтримується каркас тестових модулів PHPUnit і графічний інтерфейс для запуску тестів. При редагуванні коду виділяються конструкції синтаксису, здійснюється розширене форматування конфігурації, виявлення помилок в режимі реального часу і завершення коду. PhpStorm-редактор враховує коментарі до коду при його завершенні, автоматично вибираючи оптимальне рішення проблеми. PHP-рефакторинг і редагування шаблонів гарантує зміну проекту в найкоротші терміни. PhpStorm дозволяє візуалізувати код в ієрархічному вигляді і забезпечує швидку навігацію по всіх елементах [25].

Завдяки застосуванню PHPUnit-тестів можна швидко переглядати результати генерації коду окремих блоків або всього додатка. Якщо тест був проведений невдало, продукт дозволяє переглядати окремі кодові рядки, в яких виявлено помилку. PhpStorm забезпечує налагодження коду JavaScript і надає широкий діапазон можливостей: знаходження точки зупинки в HTML і

JavaScript, настройка параметрів точки зупинки, тестування синтаксису коду в режимі реального часу і т.д. [24], [25].

У таблиці 2.2 наведено порівняння найбільш популярних середовищ розробки для PHP.

Таблиця 2.2 – Порівняння середовищ розробки

Критерії	Anaconda	PyCharm	Atom	PhpStorm
Наявність безкоштовної версії (open-source)	++	++	++	±
Підтримка бібліотек та налаштувань	++	+	±	++
Кросс-платформність	++	++	++	—
Швидкість роботи	++	++	++	++
Підтримка та оновлення	++	±	±	++

Як видно з таблиці зв'язка PhpStorm найбільш оптимальний варіант середовища розробки для подальшої розробки. Додатково слід зазначити, що не так давно розробники представили версію із безкоштовну ліцензією для індивідуального використання: розробки та вивчення програмування для студентів та викладачів ЗВО.

2.2 Загальна структура додатку

В основі системи, що розробляється лежить архітектура «клієнт – сервер».

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних застосунків і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;

– мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів [23].

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загально прийнятим є положення, що клієнти та сервери — це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна виокремити три рівні операцій:

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку застосунку і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

- модель тонкого клієнта, в рамках якої вся логіка застосунку та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

- модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка застосунку. Програми проміжного рівня можуть функціювати під управлінням спеціальних серверів застосунків, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних [23].

Для роботи з системою користувач використовує стандартне програмне забезпечення – звичайний браузер. Це позбавляє його необхідності завантажувати та інсталювати спеціальні програми (хоча інколи така необхідність все-таки виникає). Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного

рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

2.3 Висновки за розділом 2

Для програмної реалізації веб-додатку для онлайн-тестування було проаналізовано сучасні засоби для веб-проектування та веб-програмування на основі виявлених особливостей та архітектури застосунку, що розробляється в рамках даної випускної роботи. Було виявлені необхідні засоби. Отже, для розробки практичної частини роботи обрано:

1) мова PHP для організації функціонування серверної частини, так як це найпоширеніша мова для розробки веб-додатків, що підтримується більшістю веб-серверів;

2) HTML та CSS для розробки графічного інтерфейсу користувача на стороні клієнта;

3) СКБД MySQL для проектування та розробки бази даних з метою централізованого збереження інформації;

4) веб-сервер Apache як найпопулярніший вільний веб-сервер для сумісної роботи попередньо обраних технологій проектування.

При написанні веб-застосунку необхідно враховувати особливості самого веб-застосунку і ті програмні пакети, якими воно буде доповнено. Будемо виходити з того, що пишеться веб-застосунку в основному на HTML, має PHP-сценарії, звертається до баз даних MySQL, посилається на CSS-файли, тому для розробки додатку була обрана XAMPP – кроссплатформенная збірка веб-сервера, що включає всі необхідні технології. Він не потребує створення віртуального диску, включає останні версії складових частин, швидко працює та має ряд інших переваг перед своїми аналогами.

Проаналізувавши обрану клієнт-серверну архітектуру можна зробити висновок: застосування, що розроблюється є надійним, зручним у

використанні, має відкриту кроссплатформну структуру, що відкриває широкі можливості по адаптації та масштабуванню.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Функціонування системи

Функціонування системи загалом залежить від обраної клієнт-серверної архітектури, в якій завдання або мережеве навантаження розподілені між постачальниками послуг (сервісів), що називають серверами, і замовниками послуг, званих клієнтами. Як середовище взаємодії клієнта з сервером використовується Інтернет (рис. 3.1).



Рисунок 3.1 – Клієнт-серверна архітектура

Основними перевагами архітектури «клієнт-сервер» є:

- можливість, в більшості випадків, розподілити функції обчислювальної системи між декількома незалежними комп'ютерами в мережі. Це дозволяє спростити обслуговування обчислювальної системи. Зокрема, заміна, ремонт, модернізація або переміщення сервера, не зачіпають клієнтів;

- всі дані зберігаються на сервері, який, як правило, захищений набагато краще більшості клієнтів. На сервері простіше забезпечити контроль

повноважень, щоб дозволяти доступ до даних тільки клієнтам з відповідними правами доступу;

- дозволяє об'єднати різні клієнти. Використовувати ресурси одного сервера часто можуть клієнти з різними апаратними платформами, операційними системами і таке інше.

Основні недоліки:

- у разі використання централізованої системи, непрацездатність основного сервера може зробити непрацездатним всі застосунки;
- адміністрування даної системи вимагає кваліфікованого професіонала;
- висока вартість обладнання.

3.2 Функціонування серверної частини

Почнемо проектування з деталізації серверної частини програми.

Оскільки основною метою додатку є надання послуг по засобам мережі Інтернет, необхідно включити в його серверну частину веб-сервер – апаратно-програмний комплекс, призначений для обслуговування HTTP -запитів. HTTP-запит – сформований згідно з протоколом HTTP/1.1 запит на сервер на заздалегідь певний порт (за замовчуванням, порти 80 і 8080) з метою виконання будь-якої віддаленої дії (маніпуляції з інформацією, виконання певних команд і т.д.). Як правило, такі запити посилає інтернет-браузер клієнта.

Оскільки додаток передбачає велику кількість операцій з читання значного обсягу даних, найбільш зручним варіантом буде включення в серверну частину технологій баз даних.

На рисунку 3.2 зображено процес деталізації первісної архітектури застосунку, а точніше його серверної частини. Серверна частина вміщує себе веб-сервер і сервер баз даних. У завдання веб-сервера входять:

- отримання і відповідь на HTTP - запити;

- перенаправлення запитів на необхідний додаток (сайт), як правило, приписаний до певного домену чи піддомену;
- надання додаткам доступу до необхідних модулів (наприклад, до модулю зв'язку з СКБД, модулю обробки PHP-програм та інші);
- авторизація і аутентифікація користувачів;
- реалізація функцій файл-серверу;
- інші функції.

У завдання сервера баз даних входить:

- обслуговування запитів на маніпуляції з даними на основі мови SQL;
- обслуговування бази даних;
- забезпечення цілісності даних;
- надання утиліт для адміністративного управління СКБД.

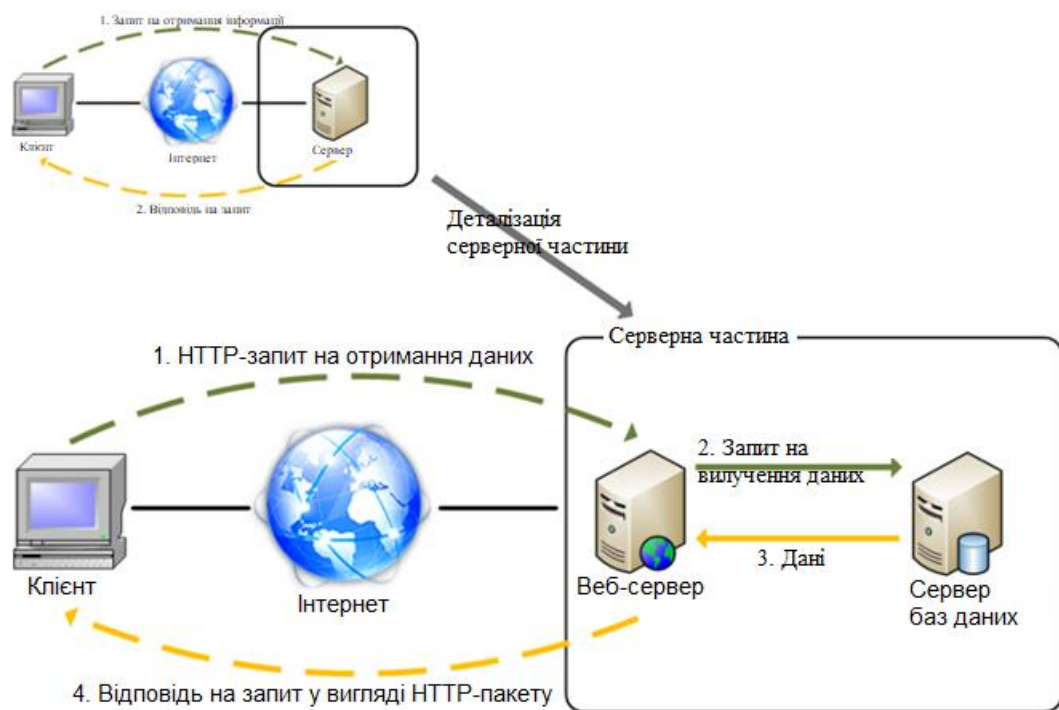


Рисунок 3.2 – Деталізація серверної частини

3.3 Проектування бази даних

3.3.1 Аналіз предметної області

Предметною областю називається частина реального світу, що представляє інтерес для даного дослідження. В основі системи що розроблюється лежить сховище тестових завдань, з даними про їх ціннісні шкали та зв'язки між сутностями бази даних.

Дані про тестові завдання, їх структура та взаємозв'язки між структурними елементами можна умовно поєднати у логічний блок тестових завдань. Кожне тестове завдання характеризується наступною інформацією:

- назва тесту;
- дані про користувача, який створив даний тест;
- кількість логічних блоків тесту;
- кількість питань та варіантів відповіді для кожного завдань;
- систему оцінювання кожного окремого завдання.

Крім того в системі присутній масив інформації що стосується процесу проходження тесту, який містить наступні дані:

- дані про користувача, що здійснює спробу проходження тесту;
- дані про тест, що проходиться в даний момент;
- дані про результати відповіді на кожне з поставлених питань.

3.3.2 Інфологічне проектування

Мета інфологічного етапу проектування полягає в отриманні семантичних (смыслових) моделей, що відображають інформаційний зміст проблеми. Однією з найбільш популярних семантичних моделей даних на етапі інфологічне проектування є неформальна модель «Сутність-Зв'язок» (Entity-Relationship - ER-модель). Моделювання предметної області базується на використанні графічних діаграм, що включають невелике число

різномірних компонентів. На рисунку 3.3 зображена інфологічна модель описаної предметної області.

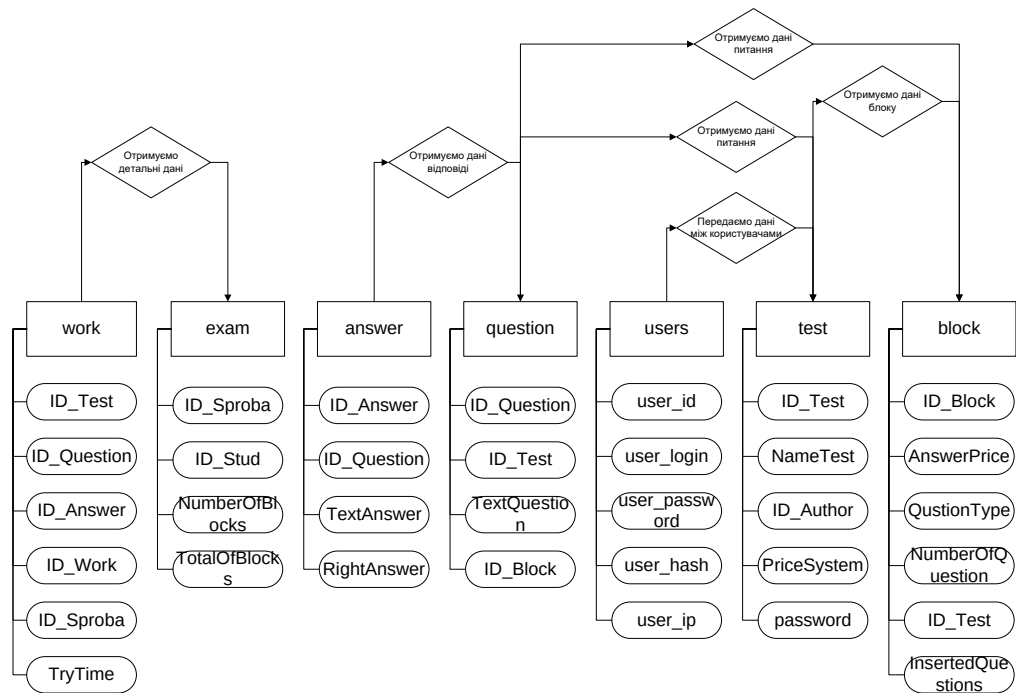


Рисунок 3.3 – Інфологічна модель

Відповідно до поставленого завдання була побудована модель прецедентів предметної області. Діаграма варіантів використання представлена на малюнку 3.4.

Для наочного уявлення поведінки прецедентів зручно застосовувати діаграму діяльності. Діаграма діяльності для прецеденту «Додати питання» (рис. 3.5), діаграма діяльності для прецеденту «Пройти тест» (рис. 3.6).

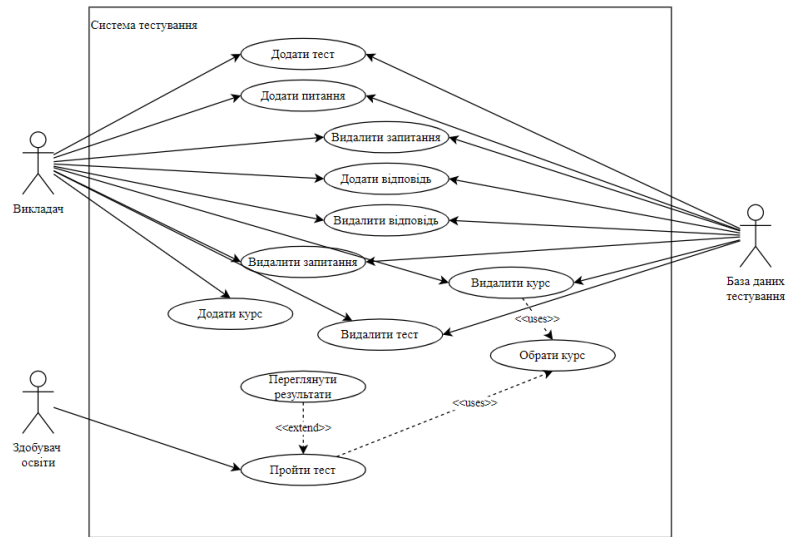


Рисунок 3.4 – Діаграма варіантів використання

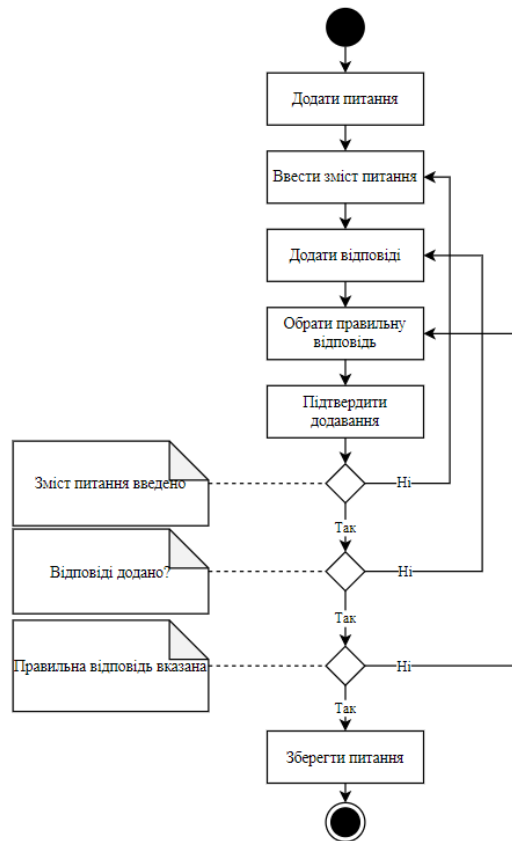


Рисунок 3.5 – Діаграма діяльності прецеденту «Додати питання»

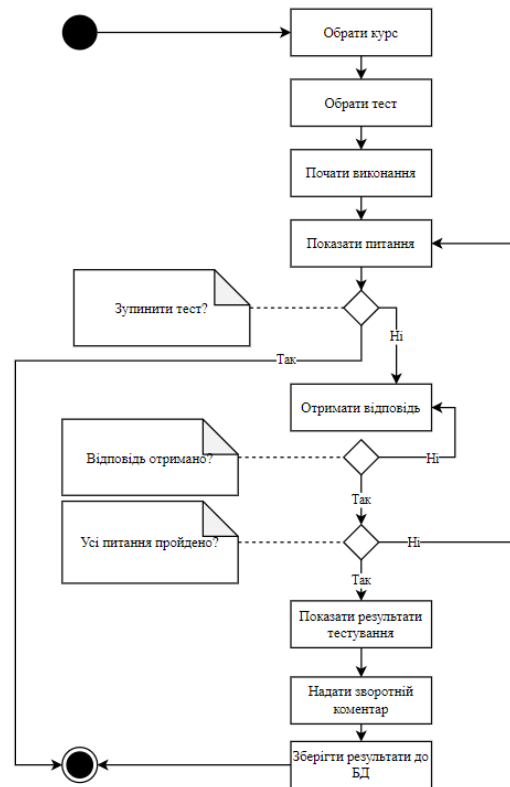


Рисунок 3.6 – Діаграма діяльності прецеденту «Пройти тест»

Для відображення взаємодії між виділеними класами на етапі побудови моделі аналізу розглянемо діаграму послідовностей для прецеденту «Додати питання» (рис. 3.7).

Діаграма послідовності допомагає встановити часовий взаємозв'язок між класами, іншими словами вона лише показує реалізацію поведінки прецеденту, але не є точним поданням кожного його кроку. На діаграмі бере участь інтерфейс користувача, яким не займаються до етапу проектування, тому на діаграмі послідовностей він міг бути опущений. В аналізі цікавить тільки основна поведінка класів аналізу. В даному випадку розглядається успішний варіант розвитку сценарію, при якому викладач вибрав тест, ввів зміст питання, можливі відповіді і вказав правильний з них. Система перевірила дані і за результатами перевірки додала нове питання в БД.

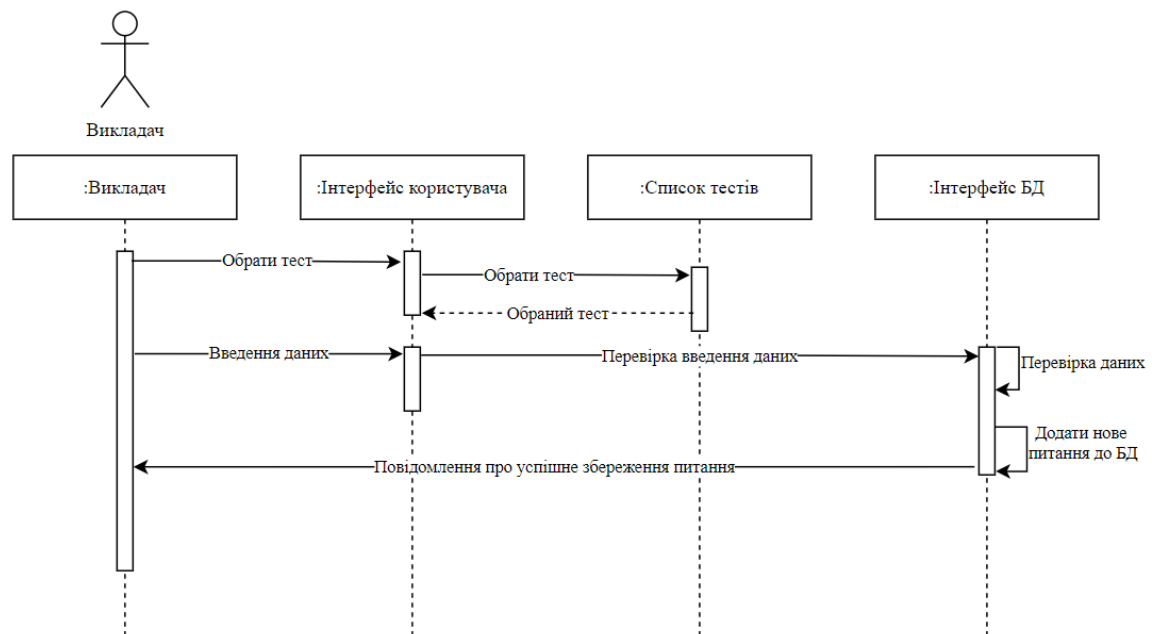


Рисунок 3.7 – Діаграма послідовності прецеденту «Додати питання»

3.3.3 Фізичне проектування

Етап фізичного проектування БД спрямований на розробку методів взаємодії з БД, які дозволяють забезпечити ефективність (високу швидкість) запитів до бази даних. Тобто на цьому етапі варто вирішити яким чином розташувати дані у зовнішній пам'яті, визначити потребу у створенні додаткових структур (наприклад, індексів), зважаючи на особливості конкретної СКБД.

Індекс – об'єкт бази даних, який створюється з метою підвищення продуктивності виконання запитів. Таблиці в базі даних можуть мати велику кількість рядків, які зберігаються в довільному порядку, і їх пошук за заданим значенням шляхом послідовного перегляду таблиці рядок за рядком може займати багато часу. Індекс формується із значень одного або декількох стовпців таблиці і покажчиків на відповідні рядки таблиці і, таким чином, дозволяє знаходити потрібний рядок за заданим значенням. Прискорення роботи з використанням індексів досягається в першу чергу за рахунок того, що індекс має структуру, оптимізовану під пошук – наприклад, збалансованого дерева.

На рисунку 3.8 зображено уточнення (деталізація) процесу проходження запиту до сервера баз даних. У цей процес також включена операційна система, яка працює як «міст» між СКБД та фізичним сховищем даних.

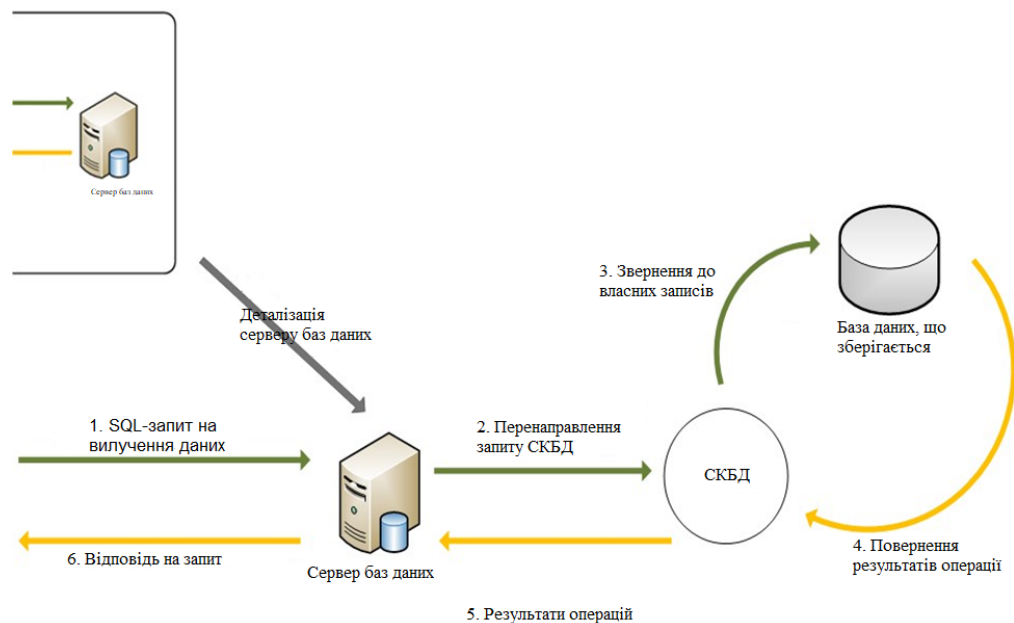


Рисунок 3.8 – Деталізація процесу проходження запиту до серверу БД

На рисунку 3.8 зображено схему даних розроблену для виконання завдань поставлених в ході розв’язання проекту. На схемі детально представлені зв’язки між сутностями БД.

3.4 Функціонування клієнтської частини

Клієнтська частина програми повинна підтримувати такі технології:

- доступ до мережі Інтернет;
- можливість роботи по протоколу HTTP/1.1;
- підтримка пристроїв взаємодії з людиною для введення даних.

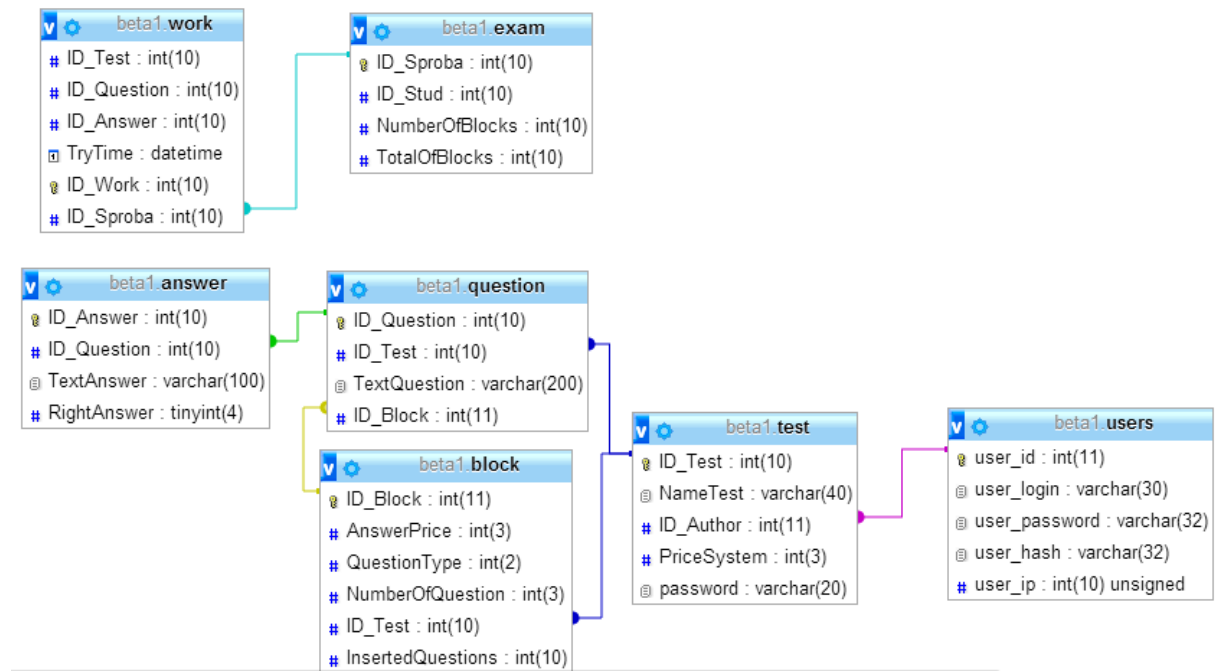


Рисунок 3.9 – Схема даних

3.5 Висновки за розділом 3

Запропоновано схему функціонування системи в цілому. Розроблена схема відповідає архітектурі клієнт-сервер і дозволяє виконувати всі поставлені перед системою функції, основними з яких є: онлайн тестування та додавання нових тестів.

Крім того, розглянуті схеми функціонування окремо серверної та клієнтської частин розроблюваної системи.

Запропоновані способи функціонування як системи в цілому, так і частин окремо, дозволяє забезпечити логічно вірну і програмно раціональну роботу програмних засобів веб-застосунку.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Опис застосування

ПЗ призначене для моніторингу рівня засвоєння знань здобувачами освіти шляхом проходження тестування. Кінцевими користувачами ПЗ можуть бути викладачі – для складання тестів та здобувачі освіти – для проходження онлайн-тестування.

4.2 Умови виконання програми

Для роботи з програмними засобами пропонується використовувати наступну рекомендовану мінімальну конфігурацію апаратного забезпечення сервера:

- процесор із тактовою частотою від 1,75 ГГц (для безперебійної роботи веб-браузеру);
- оперативний запам'ятовуючий пристрій від 3 Гб (для безперебійної роботи веб-браузеру);
- вільне місце на жорсткому диску від 1 Гб;
- мережева карта, яка забезпечує підтримку пропускної здатності не менше 100Mbps.

На сервері повинно бути встановлене наступне програмне забезпечення:

- СКБД MySQL версії 5.1.32 або вище;
- інтерфейс phpMyAdmin;
- PHP версії 5.6 або вище;
- веб-сервер Apache 2.0 і вище.

Єдиними вимогами до персональних комп'ютерів з клієнтської сторони є: наявність веб-браузера.

4.3 Інструкція по експлуатації програми

4.3.1 Характеристики програми

Розроблений вебзастосунок складається з серверної і клієнтської частини. Клієнтська частина за організацію користувацького інтерфейсу для проходження тестування, при використанні користувача з відповідними правами доста клієнтський інтерфейс дозволяє додавати нові тести та змінювати старі.

Серверна частина в свою чергу відповідає за перевірку користувача, реєстрацію користувача, відправлення запитань тесту, отримання і обробку відповідей, що підтверджує користувач, передачу миттєвих повідомлень при необхідності, організацію зручної статистики, зберігання статистики в базі даних, зберігання бази користувачів, які можуть проходити тести.

4.3.2 Перелік файлів, у яких знаходиться програма

Вебзастосунок складається в загальному обсязі з 26 файлів, які умовно можна розділити на категорії:

- файли з розширення `php`, що необхідні для організації функціональної частини;
- файли з розширенням `html` та `css` для організації GUI частини;
- файл бази даних;
- графічні файли для створення зручного і приємного інтерфейсу.

4.3.3 Звернення до програми

Для можливості роботи з програмним комплексом необхідно розмістити `php`-файли у відповідній серверній директорії. Також необхідно розгорнути базу даних користувачів у СКБД `MySQL`.

Після необхідних серверних налаштувань до програмної системи можна звертатися за допомогою веб-браузера, за відповідними адресами.

4.3.4 Вхідні й вихідні дані

Вхідними даними є логін та пароль користувача, тест, питання та варіанти відповіді для тесту.

Вихідними даними є статистика проходження тестування.

4.3.5 Запуск програми

Для можливості роботи з програмним комплексом необхідно розмістити php-файли у відповідній серверній директорії. Також необхідно розгорнути базу даних користувачів у СКБД MySQL.

Після необхідних серверних налаштувань до програмної системи можна звертатися за допомогою веб-браузера, за відповідними адресами.

Для запуску веб-застосунку необхідно зайти за відповідної адресою стартової сторінки.

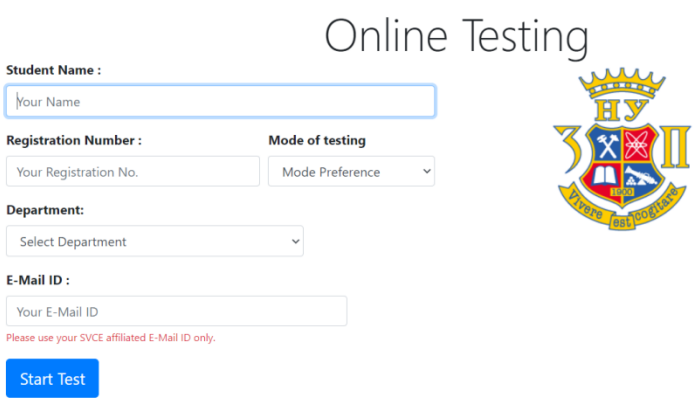
4.4 Методика та результати тестування

Для запуску додатку наберіть в адресному рядку браузеру відповідну адресу, після чого можна побачити початкову сторінку (рис. 4.1). Після чого користувач може вибрати одну з дій: зареєструватися або виконати вхід до системи. Відповідно від обраної дії залежить наступна сторінка. Якщо обрати процес входу до системи отримаю сторінку зображену на рисунку 4.2.

Для більшої зрозумілості будемо використовувати лише головний фрейм сторінки для відображення виконання програми.

Після входу до системи пропонується декілька варіантів дій, можна пройти тест, створити новий тест, якщо є відповідні права доступу, або повернутися на головну сторінку та зайти під іншим користувачем (рис. 4.3).

Якщо користувач обрав проходження тесту, то на рисунку 4.4 відображено процес вибору доступного тесту для проходження. Після вибору тесту починається процес тестування, тобто видаються питання з обраного тесту на які користувач має дати відповідь за допомогою маніпулювання перемикачами (рис. 4.5).



Online Testing

Student Name :

Registration Number : **Mode of testing**

Department:

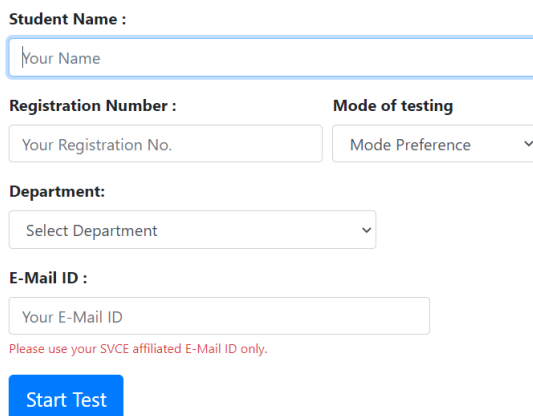
E-Mail ID :

Please use your SVCE affiliated E-Mail ID only.

Start Test

Copyright © 2021 NUZP
Department of Software Tools

Рисунок 4.1 – Початкова сторінка



Student Name :

Registration Number : **Mode of testing**

Department:

E-Mail ID :

Please use your SVCE affiliated E-Mail ID only.

Start Test

Рисунок 4.2 – Вхід до системи

Question Input

Enter Question...

This field is mandatory.

Choose File

No file chosen

Select Topic

This field is mandatory.

Enter Option A

This field is mandatory.

Enter Option B

This field is mandatory.

Enter Option C

This field is mandatory.

Enter Option D

This field is mandatory.

Select Correct Option

This field is mandatory.

Submit

Reset

Рисунок 4.3 – Вибір виду роботи з додатком

Search:

Register Num. ↑↓	Name ↑↓	Dept. ↑↓	Sec. ↑↓	E-Mail ID ↑↓	Preference ↑↓	Sec. 1 ↑↓	Sec. 2 ↑↓	Sec. 3 ↑↓	Sec. 4 ↑↓	Total ↑↓
180101001	AAKASH MOHAN	AUT		2018aut0351@svce.ac.in	Online	14	4	4	4	26
180101001	AAKASH MOHAN	AUT		2018aut0351@svce.ac.in	Online	14	4	4	4	26
180101003	ARIHANTH S S	AUT		2018aut0227@svce.ac.in	Offline	16	6	1	4	27
180101005	CHANDRU S	AUT		2018aut0783@svce.ac.in	Online	8	4	2	2	16
180101005	CHANDRU S	AUT		2018aut0783@svce.ac.in	Online	8	4	2	2	16
180101006	DARSHAN SOMANI	AUT		2018aut0718@svce.ac.in	Offline	7	1	4	1	13
180101006	DARSHAN SOMANI	AUT		2018aut0718@svce.ac.in	Offline	7	1	4	1	13
180101015	K.C.JAYAKAR	AUT		2018aut0232@svce.ac.in	Online	12	2	4	2	20
180101015	K.C.JAYAKAR	AUT		2018aut0232@svce.ac.in	Online	12	2	4	2	20
180101017	S.KAARTHIKEYAN	AUT		2018aut0297@svce.ac.in	Online	12	5	5	3	25
180101018	KAMALESH D	AUT		2018aut0368@svce.ac.in	Offline	5	4	2	2	13

Рисунок 4.4 – Вибір тесту

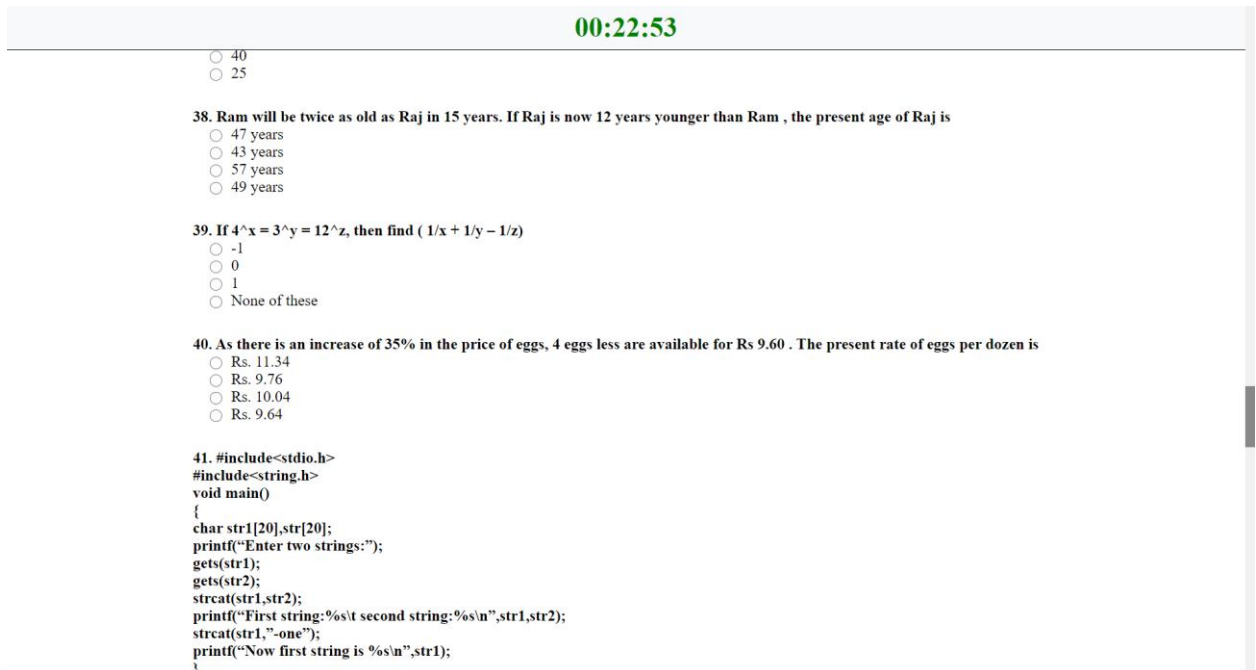


Рисунок 4.5 – Відповіді на питання

Після проходження тесту користувачу відображається статистика результатів його тестування (рис. 4.6).

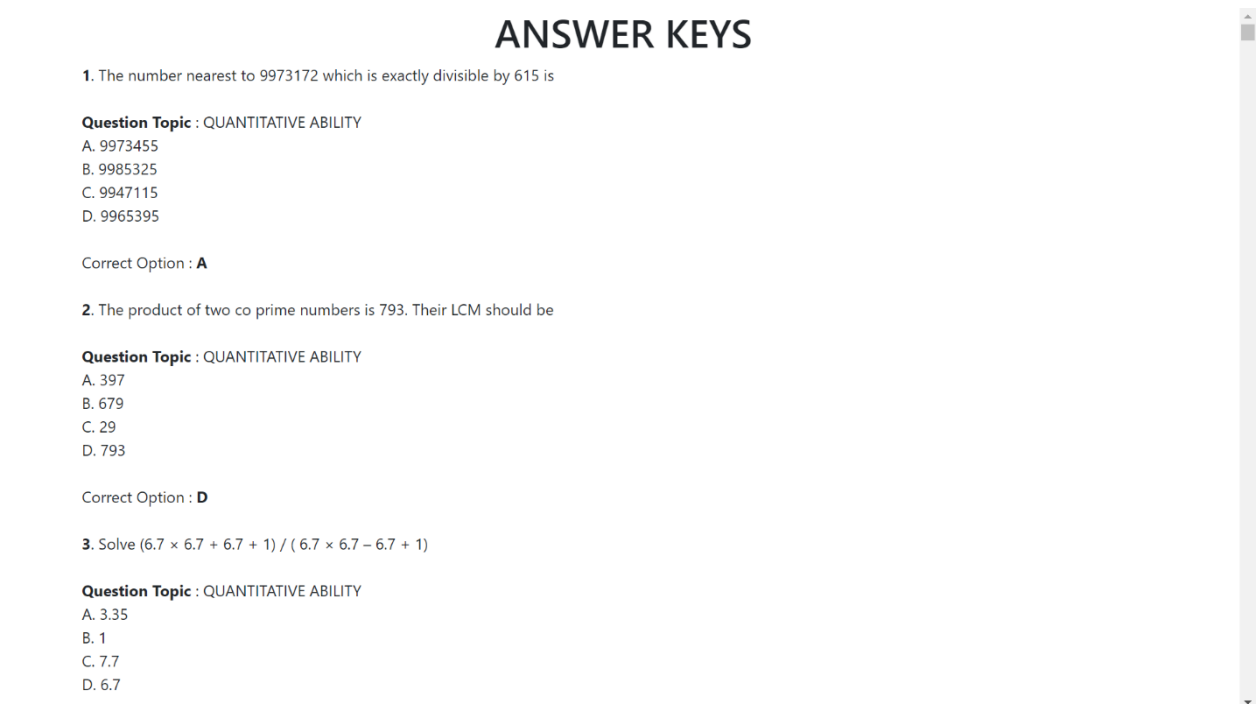


Рисунок 4.6 – Результати тестування

Якщо користувач має відповідні права доступу, то він може додати власний тест. Тому після вибору дій «Створити тест» користувач повинен вести основні характеристики тесту (рис. 4.7) та задати систему оцінювання.

Question Input

Enter Question...

This field is mandatory.

Choose File

No file chosen

Select Topic

This field is mandatory.

Enter Option A

This field is mandatory.

Enter Option B

This field is mandatory.

Enter Option C

This field is mandatory.

Enter Option D

This field is mandatory.

Select Correct Option

This field is mandatory.

Submit

Reset

Рисунок 4.7 – Створення тесту

Процес тесту інтуїтивно зрозумілий – користувачу необхідне ввести питання, варіанти відповідей та відмітити вірний варіант відповіді.

По закінченню тестування користувачі можуть надати свої коментарі із тесту, який вони проходили.

Feedback Page

1. Please rate the quality of the questions:-

☐ Very Bad
 ☐ Bad
 ☐ Average
 ☐ Good
 ☒ Very Good

2. Please rate your experience with test interface:-

☐ Very Bad
 ☐ Bad
 ☐ Average
 ☐ Good
 ☒ Very Good

3. Would you like to take up similar tests sometime in the near future?

☒ Yes
 ☐ No

Enter your Comments...

We really would appreciate some constructive feedback about the test.

Submit

Рисунок 4.8 – Форма зворотного зв'язку

ВИСНОВКИ

У випускній дипломній роботі бакалавра виконано аналіз проблеми проведення та організації онлайн-тестування. Для цього було виконано наступні завдання:

- розглянуто проблемну область;
- проаналізовано сучасні найбільш популярні аналоги для проведення тестування;
- виконано проектування програми;
- розроблено програму;
- розроблено сценарій та проведено тестування.

Для програмної реалізації веб-застосунку для онлайн-тестування було проаналізовано сучасні засоби для веб-проектування та веб-програмування на основі виявлених особливостей та архітектури застосунку, що розробляється в рамках даної випускної роботи. Було виявлені необхідні засоби. Отже, для розробки практичної частини була розроблена за допомогою: мова PHP для організації функціонування серверної частини; HTML та CSS для розробки графічного інтерфейсу користувача на стороні клієнта; СКБД MySQL для проектування та розробки бази даних з метою централізованого збереження інформації; веб-сервер Apache для сумісної роботи попередньо обраних технологій проектування.

При написанні веб-додатку були враховувати особливості самого веб-застосунку. Виходить з того, що вебзастосунок написано в основному на HTML, має PHP-сценарії, звертається до баз даних MySQL, посилається на CSS-файли, тому для розробки додатку була обрана XAMPP – кроссплатформна збірка веб-сервера, що включає всі необхідні технології.

При створенні тестових завдань використовують їх різні форми. Для програмної реалізації більш підходящими є тести закритої форми. Тому після аналізу об'єкта системи проектування були сформовані основні вимоги до

застосування. В результаті розробки випускної роботи, було розроблено вебзастосунок з наступними можливостями:

- проведення тестування в закритій формі;
- проведення тестів різної структури;
- організація тестів будь-якою мовою;
- створення необмеженої кількості тестів;
- реєстрація користувачів;
- налаштування системи оцінювання;
- організація зручної статистики.

Застосування, що розроблюється є надійним, зручним у використанні, має відкриту кроссплатформну структуру, що відкриває широкі можливості по адаптації та масштабуванню.

Усі завдання випускної дипломної роботи бакалавра повністю виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Маринова, Т. В. «Тестирование – форма контроля знаний учащихся» [Электрон. ресурс]. – Режим доступа : <http://festival.1september.ru/articles/598231/>.
2. «Разработка Web-сайта проверки уровня знаний математики учениками младших классов» [Электрон. ресурс]. – Режим доступа : <http://bibliofond.ru/view.aspx?id=656449#1/>.
3. Маматова, О. Г. «Формы контроля знаний студентов педагогических вузов» / О. Г. Маматова // Молодой ученый. — 2012. — №8. — С. 353-355.
4. Буйницька, О. П. «Форми тестових завдань. Форма подання тестового завдання» [Електрон. ресурс]. – Режим доступу : http://pidruchniki.ws/12590605/informatika/formi_testovih_zavdan_forma_podannya_testovogo_zavdannya.
5. Освітній проект «На урок»» [Електрон. ресурс]. – Режим доступу : <https://naurok.ua/student/tests>
6. УМІТИ На Урок: онлайн-тести [Електрон. ресурс]. – Режим доступу : <https://umity.in.ua/instrum65/>
7. ProProfs: SAAS Software for Training, Customer Support & More [Electronic resource]. – Access mode : <https://www.proprofs.com/>
8. ProProfs Quiz Maker Reviews [Electronic resource]. – Access mode : <https://www.g2.com/products/proprofs-quiz-maker/reviews>
9. Где создать и провести свои онлайн-тесты [Электрон. ресурс]. – Режим доступа : <http://uchinovoe.ru/articles/gde-sozdat-i-provesti-svoi-onlajn-testi>.
10. Google Формы [Электрон. ресурс]. – Режим доступа : https://www.google.com/intl/ru_ua/forms/about/
11. 5 Ways to Prevent Cheating on Your Google Form Quiz [Electronic resource]. – Access mode : <https://www.techlearning.com/news/5-ways-to-prevent->

cheating-on-your-google-form-quiz

12. HTML [Електрон. ресурс]. – Режим доступа :
<https://developer.mozilla.org/ru/docs/Web/HTML>

13. CSS [Електрон. ресурс]. – Режим доступа :
https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics

14. Perl [Електрон. ресурс]. – Режим доступа :
https://www.opennet.ru/docs/RUS/perl_help/

15. PHP [Електрон. ресурс]. – Режим доступа :
<https://www.php.net/manual/ru/index.php>

16. MySQL [Електрон. ресурс]. – Режим доступа :
<https://www.mysql.com/>

17. Пучкова, Д. М. «Обзор современных СУБД, веб-серверов, языков написания сценариев» [Електрон. ресурс]. – Режим доступа :
<http://www.webbcare.org/lectures/web-servers/Lecture11.pdf>.

18. Клієнт-серверна архітектура [Електрон. ресурс]. – Режим доступу :
<https://medium.com/@IvanZmerzlyi/%D0%BA%D0%BB%D1%96%D1%94%D0%BD%D1%82%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0-%D1%82%D0%B0-%D1%80%D0%BE%D0%BB%D1%96%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D1%96%D0%B2-9893d8048229>

19. СКБД Oracle [Електрон. ресурс]. – Режим доступу :
<https://www.oracle.com/ua/index.html>

20. Грабер, М. Справочное руководство по SQL. – М.: Лори, 1997. – 291 с.

21. Методичні вказівки до лабораторних робіт "Вивчення пакетів MS Excel і MS Access" (частина 1) з дисципліни "Організація баз даних і знань" для студентів спеціальності 8.080403 "Програмне забезпечення автоматизованих систем" (всіх форм навчання) /Укл.: С. К. Корнієнко. –

Запоріжжя: ЗНТУ, 2003 – 70 с.

22. Методичні вказівки до лабораторних робіт "Вивчення пакетів MS Excel і MS Access" (частина 2) з дисципліни "Організація баз даних і знань" для студентів спеціальності 8.080403 "Програмне забезпечення автоматизованих систем" (всіх форм навчання) /Укл.: С. К. Корнієнко. – Запоріжжя: ЗНТУ, 2003 – 62 с.

23. Вайк А. PHP. Справочник / А. Вайк, Д. Уокер, Р. Кокс. – К. : DiaSoft, 2001. – 447 с.

24. PhpStorm [Electronic resource]. – Access mode : <https://www.jetbrains.com/ru-ru/phpstorm/>

25. Top 6 PHPStorm Plugins Every PHP Developer Should Know [Electronic resource]. – Access mode : <https://dev.to/codebetteracademy/top-5-phpstorm-plugins-every-php-developer-should-know-16gp>

ДОДАТОК А
Технічне завдання

Вступ

Для організації та проведення онлайн-тестування з метою моніторингу успішності засвоєння навчального матеріалу здобувачами освіти.

A.1 Підстави для розробки

Розробка виконується на основі завдання на дипломну кваліфікаційну роботу бакалавра, яке було визначене за темою «Програмна реалізація системи тестування для перевірки знань» і затверджене наказом № 103 від 30 березня 2021 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Продукт «Вебзастосунок для онлайн-тестування» призначений для забезпечення онлайн-тестування за різноманітними навчальними дисциплінами, що можуть проводитися в сучасних навчальних закладах.

Вебзастосунок надає наступні можливості:

- проведення тестування в закритій формі;
- проведення тестів різної структури;
- організація тестів будь-якою мовою;
- створення необмеженої кількості тестів;
- реєстрація користувачів;
- авторизація користувачів;
- аутентифікація користувачів;
- налаштування системи оцінювання;
- організація зручної статистики.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Функціональні характеристики, виконання яких має забезпечуватися програмою:

- проведення тестування в закритій формі;
- проведення тестів різної структури;
- організація тестів будь-якою мовою;
- створення необмеженої кількості тестів;
- реєстрація користувачів;
- налаштування системи оцінювання;
- організація зручної статистики.

A.3.2 Вимоги до надійності

Основним чином надійність роботи програми має забезпечуватися шляхом надійності надання доступу до серверу.

Окрім того програма повинна контролювати дані, які потрапляють їй на вхід або отримуються в результаті її роботи.

Вхідні дані охоплюють зокрема дані, отримані з бази даних, від користувача, між підсистемами програмної системи.

A.3.3 Умови експлуатації

Для експлуатації програми має бути виділено окремий сервер, через який буде відбуватися доступ до веб-застосунку. Окрім того на сервері має розташовуватися база даних та система MySQL, але для збереження їх може виділятися також окремий сервер бази даних.

Відповідно необхідно перевіряти, слідкувати і керувати роботою сервера у відповідності з заданими вимогами до серверів.

A.3.4 Вимоги до складу та параметрів технічних засобів

Для роботи з програмними засобами пропонується використовувати наступну рекомендовану мінімальну конфігурацію апаратного забезпечення сервера:

- процесор із тактовою частотою від 1,75 ГГц (для безперебійної роботи веб-браузеру);
- оперативний запам'ятовуючий пристрій від 3 Гб (для безперебійної роботи веб-браузеру);
- вільне місце на жорсткому диску від 1 Гб;
- мережева карта, яка забезпечує підтримку пропускну здатності не менше 100Mbps.

На сервері повинно бути встановлене наступне програмне забезпечення:

- СКБД MySQL версії 5.1.32 або вище;
- інтерфейс phpMyAdmin;
- PHP версії 5.6 або вище;
- веб-сервер Apache 2.0 і вище.

Єдиними вимогами до персональних комп'ютерів з клієнтської сторони є: наявність веб-браузера.

A.3.5 Вимоги до транспортування та збереження

Передбачається, що програма буде зберігатися на сервері і таким чином буде виконуватися її розповсюдження.

A.3.6 Вимоги до програмної документації

Необхідно розробити програмні документи, що включають дане технічне завдання, текст програми, що оформлюються додатками

пояснювальної записки до випускної дипломної роботи бакалавра, та зокрема опис експлуатації та тестування програми.

А.3.7 Вимоги до маркування та упакування

Упаковка повинна бути цілісною у тому випадку, якщо програма зберігається у вигляді фізичної копії на носії інформації, включаючи зокрема флешки, компакт-диски, карти пам'яті.

А.4 Порядок контролю та приймання

Дипломна випускна робота бакалавра повинна виконуватися у повній відповідності з календарним планом, виконання етапів має контролюватися керівником роботи та консультантами відповідних розділів. Останім етапом є захист роботи перед екзаменаційною комісією у дистанційній формі.

ДОДАТОК Б
Кервіництво програміста

Б.1 Призначення програми і умови застосування програми

Продукт «Вебзастосунок для тестування» призначений для забезпечення онлайн-тестування за різноманітними навчальними дисциплінами, що можуть проводитися в сучасних навчальних закладах.

Веб- застосування надає наступні можливості:

- проведення тестування в закритій формі;
- проведення тестів різної структури;
- організація тестів будь-якою мовою;
- створення необмеженої кількості тестів;
- реєстрація користувачів;
- налаштування системи оцінювання;
- організація зручної статистики.

Для роботи з програмними засобами пропонується використовувати наступну рекомендовану мінімальну конфігурацію апаратного забезпечення сервера:

- процесор із тактовою частотою від 1,75 ГГц (для безперебійної роботи веб-браузеру);
- оперативний запам'ятовуючий пристрій від 3 Гб (для безперебійної роботи веб-браузеру);
- вільне місце на жорсткому диску від 1 Гб;
- мережева карта, яка забезпечує підтримку пропускної здатності не менше 100Mbps.

На сервері повинно бути встановлене наступне програмне забезпечення:

- СКБД MySQL версії 5.1.32 або вище;
- інтерфейс phpMyAdmin;
- PHP версії 5.6 або вище;
- веб-сервер Apache 2.0 і вище.

Єдиними вимогами до персональних комп'ютерів з клієнтської сторони є: наявність веб-браузера.

Тексти програми наведені в додатку Д.

Б.2 Характеристики програми

Розроблений веб-додаток складається з серверної і клієнтської частини. Клієнтська частина за організацію користувацького інтерфейсу для проходження тестування, при використанні користувача з відповідними правами доста клієнтський інтерфейс дозволяє додавати нові тести та змінювати старі.

Серверна частина в свою чергу відповідає за перевірку користувача, реєстрацію користувача, відправлення запитань тесту, отримання і обробку відповідей, що підтверджує користувач, передачу миттєвих повідомлень при необхідності, організацію зручної статистики, зберігання статистики в базі даних, зберігання бази користувачів, які можуть проходити тести.

Б.3 Перелік файлів, у яких знаходиться програма

Веб-застосування складається в загальному обсязі з 26 файлів, які умовно можна розділити на категорії:

- файли з розширення `php`, що необхідні для організації функціональної частини;
- файли з розширенням `html` та `css` для організації інтерфейсної частини;
- файл бази даних;
- графічні файли для створення зручного і приємного інтерфейсу.

Б.4 Звертання до програми

Для можливості роботи з програмним комплексом необхідно розмістити php-файли у відповідній серверній директорії. Також необхідно розгорнути базу даних користувачів у СКБД MySQL.

Після необхідних серверних налаштувань до програмної системи можна звертатися за допомогою веб-браузера, за відповідними адресами.

Б.5 Вхідні й вихідні дані

Вхідними даними є логін та пароль користувача, тест, питання та варіанти відповіді для тесту.

Вихідними даними є статистика проходження тестування.

ДОДАТОК В
Керівництво користувача

В.1 Призначення програми

Продукт «Вебзастосунок для тестування» призначений для забезпечення онлайн-тестування за різноманітними навчальними дисциплінами, що можуть проводитися в сучасних навчальних закладах.

Веб- застосування надає наступні можливості:

- проведення тестування в закритій формі;
- проведення тестів різної структури;
- організація тестів будь-якою мовою;
- створення необмеженої кількості тестів;
- реєстрація користувачів;
- налаштування системи оцінювання;
- організація зручної статистики.

В.2 Умови виконання програми

Для роботи з програмними засобами пропонується використовувати наступну рекомендовану мінімальну конфігурацію апаратного забезпечення сервера:

- процесор із тактовою частотою від 1,75 ГГц (для безперебійної роботи веб-браузеру);
- оперативний запам'ятовуючий пристрій від 3 Гб (для безперебійної роботи веб-браузеру);
- вільне місце на жорсткому диску від 1 Гб;
- мережева карта, яка забезпечує підтримку пропускної здатності не менше 100Mbps.

На сервері повинно бути встановлене наступне програмне забезпечення:

- СКБД MySQL версії 5.1.32 або вище;
- інтерфейс phpMyAdmin;

- PHP версії 5.6 або вище;
- веб-сервер Apache 2.0 і вище.

Єдиними вимогами до персональних комп'ютерів з клієнтської сторони є: наявність веб-браузера.

В.3 Як запустити програму

Для можливості роботи з програмним комплексом необхідно розмістити php-файли у відповідній серверній директорії. Також необхідно розгорнути базу даних користувачів у СКБД MySQL.

Після необхідних серверних налаштувань до програмної системи можна звертатися за допомогою веб-браузера, за відповідними адресами.

Для запуску веб-додатку необхідно зайти за відповідної адресою стартової сторінки.

В.4 Виконання програми

Для запуску додатку наберіть в адресному рядку браузеру відповідну адресу, після чого можна побачити початкову сторінку (рис. В.1). Після чого користувач може вибрати одну з дій: зареєструватися або виконати вхід до системи. Відповідно від обраної дії залежить наступна сторінка. Якщо обрати процес входу до системи отримаю сторінку зображену на рисунку В.2.

Для більшої зрозумілості будемо використовувати лише головний фрейм сторінки для відображення виконання програми.

Після входу до системи пропонується декілька варіантів дій, можна пройти тест, створити новий тест, якщо є відповідні права доступу, або повернутися на головну сторінку та зайти під іншим користувачем (рис. В.3).

Якщо користувач обрав проходження тесту, то на рисунку В.4 відображено процес вибору доступного тесту для проходження. Після вибору тесту починається процес тестування, тобто видаються питання з обраного

тесту на які користувач має дати відповідь за допомогою маніпулювання перемикачами (рис. В.5).

Online Testing



Student Name :

Registration Number :

Mode of testing

Department:

E-Mail ID :

Please use your SVCE affiliated E-Mail ID only.

Copyright © 2021 NUZP
Department of Software Tools

Рисунок В.1 – Початкова сторінка

Student Name :

Registration Number :

Mode of testing

Department:

E-Mail ID :

Please use your SVCE affiliated E-Mail ID only.

Рисунок В.2 – Вхід до системи

Question Input

Enter Question...

This field is mandatory.

Choose File

No file chosen

Select Topic

This field is mandatory.

Enter Option A

This field is mandatory.

Enter Option B

This field is mandatory.

Enter Option C

This field is mandatory.

Enter Option D

This field is mandatory.

Select Correct Option

This field is mandatory.

Submit

Reset

Рисунок В.3 – Вибір виду роботи з додатком

Search:

Register Num. ↑↓	Name ↑↓	Dept. ↑↓	Sec. ↑↓	E-Mail ID ↑↓	Preference ↑↓	Sec. 1 ↑↓	Sec. 2 ↑↓	Sec. 3 ↑↓	Sec. 4 ↑↓	Total ↑↓
180101001	AAKASH MOHAN	AUT		2018aut0351@svce.ac.in	Online	14	4	4	4	26
180101001	AAKASH MOHAN	AUT		2018aut0351@svce.ac.in	Online	14	4	4	4	26
180101003	ARIHANTH S S	AUT		2018aut0227@svce.ac.in	Offline	16	6	1	4	27
180101005	CHANDRU S	AUT		2018aut0783@svce.ac.in	Online	8	4	2	2	16
180101005	CHANDRU S	AUT		2018aut0783@svce.ac.in	Online	8	4	2	2	16
180101006	DARSHAN SOMANI	AUT		2018aut0718@svce.ac.in	Offline	7	1	4	1	13
180101006	DARSHAN SOMANI	AUT		2018aut0718@svce.ac.in	Offline	7	1	4	1	13
180101015	K.C.JAYAKAR	AUT		2018aut0232@svce.ac.in	Online	12	2	4	2	20
180101015	K.C.JAYAKAR	AUT		2018aut0232@svce.ac.in	Online	12	2	4	2	20
180101017	S.KAARTHIKEYAN	AUT		2018aut0297@svce.ac.in	Online	12	5	5	3	25
180101018	KAMALESH D	AUT		2018aut0368@svce.ac.in	Offline	5	4	2	2	13

Рисунок В.4 – Вибір тесту

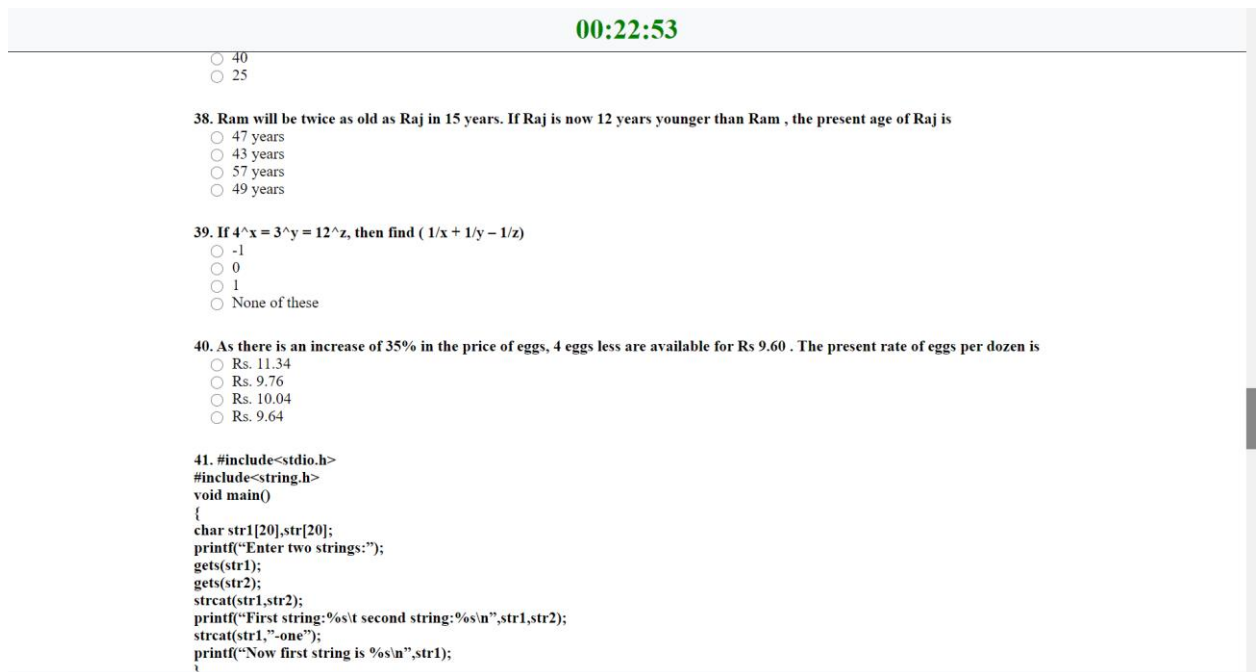


Рисунок В.5 – Відповіді на питання

Після проходження тесту користувачу відображається статистика результатів його тестування (рис. В.6).

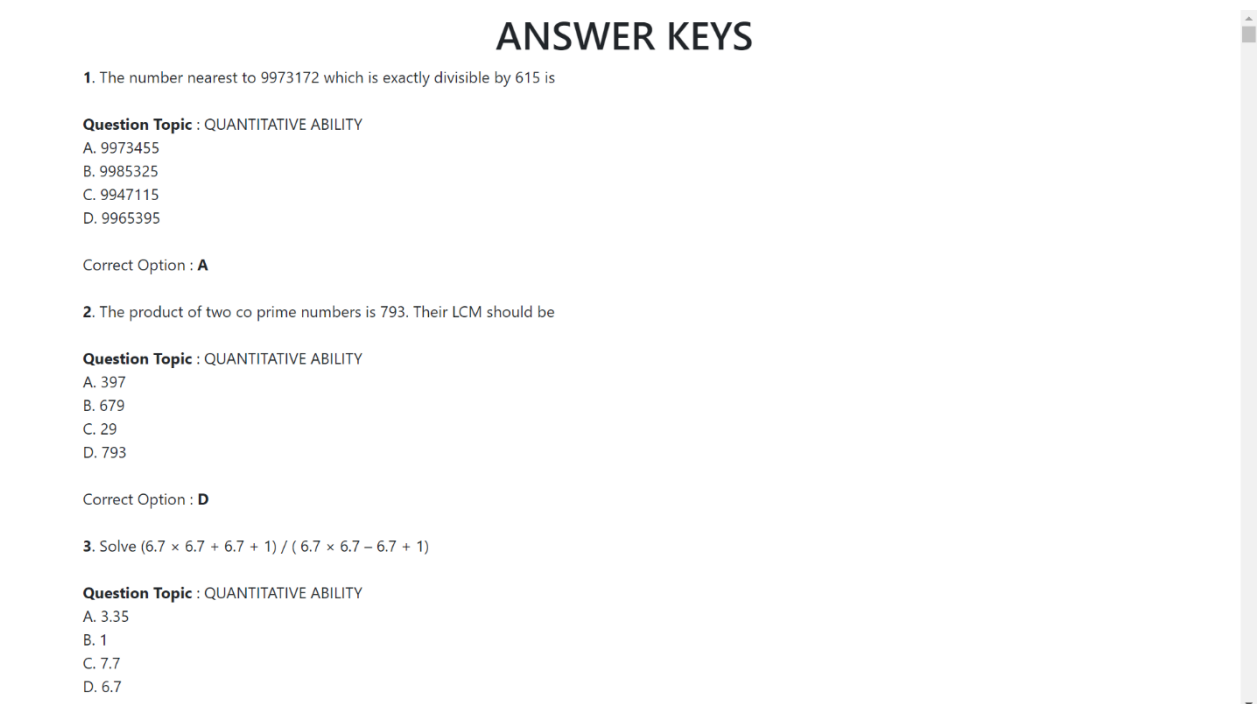


Рисунок В.6 – Результати тестування

Якщо користувач має відповідні права доступу, то він може додати власний тест. Тому після вибору дій «Створити тест» користувач повинен вести основні характеристики тесту (рис. В.7) та задати систему оцінювання.

Question Input

Enter Question...

This field is mandatory.

Choose File

 No file chosen

Select Topic

This field is mandatory.

Enter Option A

This field is mandatory.

Enter Option B

This field is mandatory.

Enter Option C

This field is mandatory.

Enter Option D

This field is mandatory.

Select Correct Option

This field is mandatory.

Submit

Reset

Рисунок В.7 – Створення тесту

В.5 Повідомлення користувачу

Для користувача передбачено повідомлення про помилки, які можуть виникнути при реєстрації на авторизації.

ДОДАТОК Д
Текст програми

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-
width, initial-scale=1.0" />

    <title>Online testing</title>
    <link                                rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.
1/css/bootstrap.min.css" />
    <link href="../../../static/css/login-index.css"
rel="stylesheet" />
  </head>

  <body>
    <div class="container">
      <div id="head">
        <header id="pageHead" class="display-4
text-center">Online Testing</header>
      </div>
      <div class="row">
        <div class="col-md-7">
          <form
            id="loginform"
            name="loginForm"
            method="POST"
            action="../../../test/index.php"
            class="needs-validation"
            novalidate
          >
            <div class="form-row">
              <div class="form-group col-md-10">
                <label><b>Student          Name          :
</b></label>
                <input type="text" name="nameIn"
placeholder="Your Name" class="form-control" autofocus
required />
              </div>
            </div>
            <div class="form-row">
              <div class="form-group col-md-6">
                <label><b>Registration    Number    :
</b></label>
                <input

```

```

        type="text"
        name="regIn"
        placeholder="Your      Registration
No."

        class="form-control"
        pattern="18\d{7}"
        required
    />
</div>
<div class="form-group col-md-4">
    <label      for=""><b>Mode      of
testing</b></label>
    <select name="modeIn" class="form-
control" required>
        <option      value=""      disabled
selected>Mode Preference</option>
        <option
value="Online">Online</option>
        <option
value="Offline">Offline</option>
    </select>
</div>
</div>
<div class="form-row">
    <div class="form-group col-md-7">
        <label><b>Department: </b></label>
        <select      name="deptIn"
id="deptSelect" class="form-control" required>
            <option      value=""      disabled
selected>Select Department</option>
            <option      value="ST">Software
Tools</option>
        </select>
    </div>
    <div class="form-group col-md-3 mt-
2">
        <select      name="sectionInput"
class="form-control d-none mt-4">
            <option      value=""      selected
disabled>Section</option>
            <option value="A">A</option>
            <option value="B">B</option>
            <option      value="C"      class="d-
none">C</option>
        </select>

```

```
</div>
</div>
<div class="form-row">
    <div class="form-group col-md-8">
        <label><b>E-Mail ID : </b></label>
        <input type="email" name="emailIn"
placeholder="Your E-Mail ID" class="form-control"
required />
        <small class="text-danger">Please
use your SVCE affiliated E-Mail ID only.</small>
    </div>
</div>
<button type="submit" class="formInput
btn btn-primary btn-lg" id="submitButton">Start
Test</button>
    </form>
</div>
<div class="col-md-5">
    <a href=""></a>
</div>
</div>
</div>
</div>
<footer>
    Copyright &copy; 2021 <b>NUZP</b>
    <p><b>Department of Software Tools<b>
</p>
</footer>
<script
src="../../../static/js/loginForm.js"></script>
<script
src="../../../static/js/validateForms.js"></script>
</body>
</html>

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css"
href="diplom.css">
<meta http-equiv="Content-Type" content="text/html;
charset="windows-1251">
<title>Добро пожаловать</title>
```

```

</head>
<body background="bottom.jpg">
<h1>Вітаємо</h1>
<p>Увійдіть або зареєструйтесь</p>
<br>
<input type="button" value="Увійти" onclick="
location.href='login.php' ">
<br>
<input type="button" value="Зареєструватися"
onclick=" location.href='register.php' ">
</body>
</html>

```

```

register.php
<html>
<head>
<link rel="stylesheet" type="text/css"
href="diplom.css">
<meta http-equiv="Content-Type" content="text/html;
charset="windows-1251">
<title> Вітаємо </title>
</head>
<body>
<div><h1> Вітаємо </h1></div>
<div>Раді новим користувачам</div>
<p></p>
<?php
// Сопінка реєстрації
# Підєднуємося с БД
mysql_connect("localhost", "root", "");
mysql_select_db("beta1");
if(isset($_POST['submit']))
{
    $err = array();
    # перевіряємо логін
    if(!preg_match("/^[a-zA-Z0-
9]+$/", $_POST['login']))
    {
        $err[] = "Логін може тільки з англійських
літер";
    }
    if(strlen($_POST['login']) < 3 or
strlen($_POST['login']) > 30)
    {

```

```

        $err[] = "Логін повинен бути не менше 3
символів ";
    }
    # перевірка унікальності
    записів
    $query = mysql_query("SELECT COUNT(user_id)
FROM users WHERE
user_login='".mysql_real_escape_string($_POST['login'])
.'");
    if(mysql_result($query, 0) > 0)
    {
        $err[] = "Пользователь с таким логином уже
существует в базе данных";
    }
    # Если нет ошибок, то добавляем в БД нового
пользователя
    if(count($err) == 0)
    {
        $login = $_POST['login'];
        # Убираем лишние пробелы и делаем двойное
шифрование
        $password =
md5(md5(trim($_POST['password'])));
        mysql_query("INSERT INTO users SET
user_login='".$login."',
user_password='".$password.'");
        header("Location: login.php"); exit();
    }
    else
    {
        print "<b>При регистрации произошли
следующие ошибки:</b><br>";
        foreach($err AS $error)
        {
            print $error."<br>";
        }
    }
}
?>
<table align="center">
<form method="POST">
<tr>
<td>Логин</td>
<td><input name="login" type="text"><br></td>
</tr>

```

```

        <tr>
        <td>Пароль</td>
        <td><input                                name="password"
type="password"><br></td>
        </tr>
        </table>
        <input                                name="submit"                type="submit"
value="Зарегистрироваться">
        </form>
        </body>
        </html>

```

```

login.php
<!DOCTYPE html>
<html>
<head>
<link            rel="stylesheet"                type="text/css"
href="diplom.css">
    <meta http-equiv="Content-Type" content="text/html;
charset="windows-1251">
    <title>Добро пожаловать</title>
</head>
<body>
<div><h2>Войдите </h1></div>
<?php
// Страница авторизации
# Функция для генерации случайной строки
function generateCode($length=6) {
    $chars  = "abcdefghijklmnopqrstuvwxyzABCDEFGHI
JKLMNOPQRSTUVWXYZ0123456789";
    $code = "";
    $clen = strlen($chars) - 1;
    while (strlen($code) < $length) {
        $code .= $chars[mt_rand(0,$clen)];
    }
    return $code;
}
# Соединяюсь с БД
mysql_connect("localhost", "root", "");
mysql_select_db("beta1");
if(isset($_POST['submit']))
{
    # Вытаскиваем из БД запись, у которой логин
равняется введенному

```

```

        $query      =      mysql_query("SELECT      user_id,
user_password      FROM      users      WHERE
user_login='".mysql_real_escape_string($_POST['login'])
.'" LIMIT 1");
        $data = mysql_fetch_assoc($query);
        # Сравниваем пароли
        if($data['user_password']      ===
md5(md5($_POST['password'])))
        {
            # Генерируем случайное число и шифруем его
            $hash = md5(generateCode(10));
            if(!@$_POST['not_attach_ip'])
            {
                # Если пользователя выбрал привязку к
IP
                # Переводим IP в строку
                $insip      =      ",
user_ip=INET_ATON('".$_SERVER['REMOTE_ADDR']."'");
            }
            # Записываем в БД новый хеш авторизации и
IP
            mysql_query("UPDATE      users      SET
user_hash='".$hash."'      ".$insip."      WHERE
user_id='".$data['user_id']."'");
            # Ставим куки
            setcookie("id",      $data['user_id'],
time()+60*60*24*30);
            setcookie("hash",      $hash,
time()+60*60*24*30);
            # Переадресовываем браузер на страницу
проверки нашего скрипта
            header("Location: check.php"); exit();
        }
        else
        {
            print "Вы ввели неправильный логин/пароль";
        }
    }
?>
<table align="center">
<form method="POST">
<tr>
<td>Логин</td>      <td><input      name="login"
type="text"><br></td>
</tr>

```

```

        <tr>
        <td>Пароль</td>
        <td><input                                name="password"
type="password"><br></td>
        </tr>
        <!--Не прикреплять к IP(не безопасно) <input
type="checkbox" name="not_attach_ip"><br>-->
        </table>
        <input name="submit" type="submit" value="Войти">
        </form>
        </body>
        </html>

```

```

stud.php
<html>
    <head>
        <link          rel="stylesheet"          type="text/css"
href="diplom.css">
        <meta          http-equiv="Content-Type"
content="text/html; charset="windows-1251">
        <title>Работа с тестами</title>
    </head>
    <body>
        <h3>Выберите тест</h3>
        <?php
        // Соединение с сервером БД
        mysql_connect("localhost", "root", "") or die
(mysql_error());
        // Выбор БД
        mysql_select_db("beta1") or die(mysql_error());
        $nema = "Нет больше тестов";
        $arrResult= array();
        $result = mysql_query("SELECT NameTest FROM test
Limit 0,10"); //потом сделать лимит 10
        //$userdata = mysql_fetch_assoc($query);
        /*while($data= mysql_fetch_array($result))
        {
            $arrResult[]=$data['NameTest'];
            $var=$arrResult;
        }
        print_r($var);
        echo $nema;
        */
        ?>
        <form action = "studOpen.php" method="post">

```

```

        <?php
        $i=0;
        while      ($row      =      mysql_fetch_array($result,
MYSQL_NUM) )
        {
            ?>
            <table align="center">
            <tr>
            <td>
            <?php
            printf($row[0]);
            ?>
            </td>
            <td>
            <?php
            echo  "<input  type=\"radio\"  value=\"\${i}\"
name=\"type\">";
            ?>
            </td>
            </tr>
            </table>
            <?php
            //<input          type="radio"          name="type"
value="<?php echo rad2"><br>
            $query = mysql_query("SELECT  ID_Test  FROM
test Limit ${i},1");
            $arResult = mysql_fetch_array($query);
            $QueryData= $arResult[0];
            $Arr[${i}] = $QueryData;
            //echo $Arr[${i}];
            echo "<br>";
            $i++;
        }
        ?>
        <input type = "submit" name ="submit" value=
"Пройти"/>
        <?php
        mysql_free_result($result);
        //print_r($arResult);
        //ORDER BY ID_Test DESC"
        /*$array_result = mysql_fetch_array($result);
        $foo = NULL;
        $nema = "Нет больше тестов";
        for ($i = 0; ; $i++)
        {

```

```

        $var = $array_result[$i];
        if (!isset($var))
        {
            echo $nema;
            break;
        }
        echo $var;
        echo '<br>';
    }
*/
mysql_close();
?>
</body>
</html>

1.php
<html>
<head>
<link          rel="stylesheet"          type="text/css"
href="diplom.css">
<meta http-equiv="Content-Type" content="text/html;
charset="windows-1251">
<title> Таблица </title>
</head>
<body>
<table align="center">
<form action = "test.php" method="post">
<!--АвторИД: <input type= "text" required name
= "ID_Author"><br>-->
<tr>
<td>Тест: </td>
<td><input type= "text" required name =
"Test"><br> </td>
</tr>
<tr>
<td>Пароль: </td>
<td><input type='password' name =
"Password"><br> </td>
</tr>
<tr>
<td>Система оценивания: </td>
<td><input type="number" required min="1"
max="100" name = "PriceSystem"></td>
</tr>
</table>

```

```
        <input type = "submit" name ="submit" value=
"Сохранить"/>
    </form>
</body>

</html>
```

ДОДАТОК Е
Слайди презентації



Рисунок Е.1 – Слайд 1

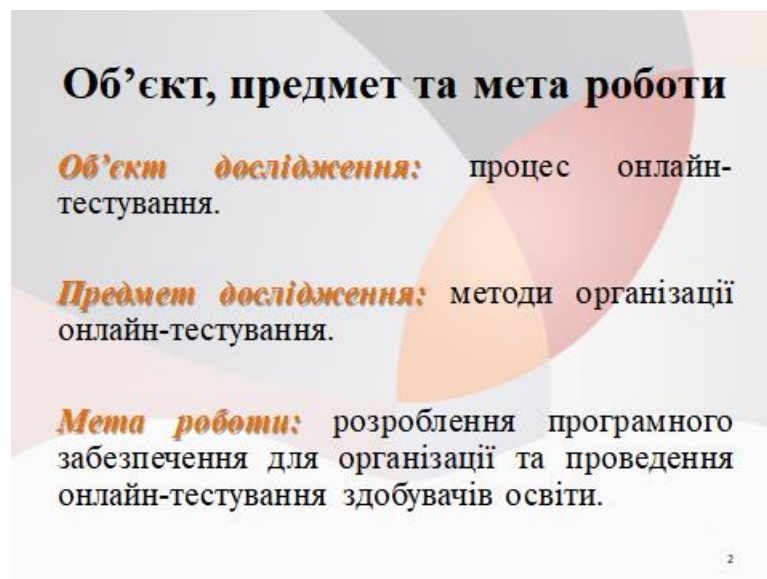


Рисунок Е.2 – Слайд 2

Постановка завдання

Розробити вебзастосунок для контролю знань у формі тестування.

Застосунок повинен відповідати наступним вимогам:

- проведення тестування в закритій формі.
- тести можуть бути різної структури.
- тести можуть бути організовані на будь-якій мові.
- необмежена кількість тестів, що створюється.
- можливість реєстрації користувача.
- можливість налаштування системи оцінювання.
- зручна статистика.

3

Рисунок Е.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Проект «На урок»	Сайт Proprofs.com	Google Форми	Онлайн тестування
Вартість	Безкоштовний	Платний (наявна пробна версія)	Безкоштовні з обмеженнями	Безкоштовний
Кроссплатформність	Так	Так	Так	Так
Види тестувань	Середнє	Багато	Середнє	Багато
Види перевірки результатів	Автоматична	Автоматична	Автоматична, ручна	Автоматична, ручна
Підтримка мультимедійних даних для тестування	Так	Так	Так	Так

4

Рисунок Е.4 – Слайд 4

Порівняння мов програмування

Критерій	Perl	Java	Python	PHP
Швидкість роботи	-	+	±	+
Бібліотеки	±	±	++	+
Парадигми програмування	±	+	+	+
Спільнота	-	+	+	++
Фреймворки	-	±	+	++

5

Рисунок Е.5 – Слайд 5

Порівняння середовищ розробки

Критерії	Anaconda	PyCharm	Atom	PhpStorm
Наявність безкоштовної версії (open-source)	++	++	++	±
Підтримка бібліотек та налаштувань	++	+	±	++
Кросс-платформність	++	++	++	-
Швидкість роботи	++	++	++	++
Підтримка та оновлення	++	±	±	++

6

Рисунок Е.6 – Слайд 6

Архітектура програми



- XAMPP, LAMP, WAMP (Apache, PHP, MySQL, Windows / Linux)
- HTML, CSS, JavaScript (клієнтський)
- Протокол HTTP

7

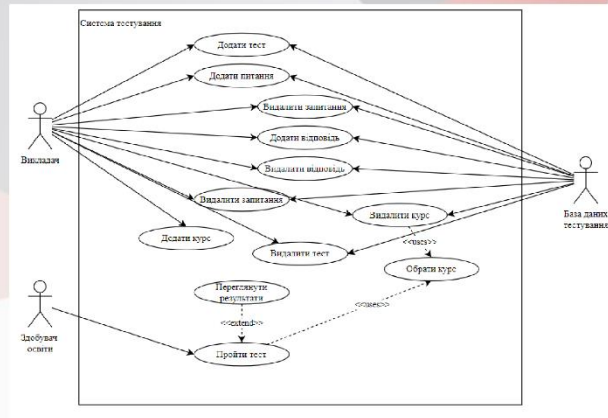
Рисунок Е.7 – Слайд 7

Інфологічна модель



Рисунок Е.8 – Слайд 8

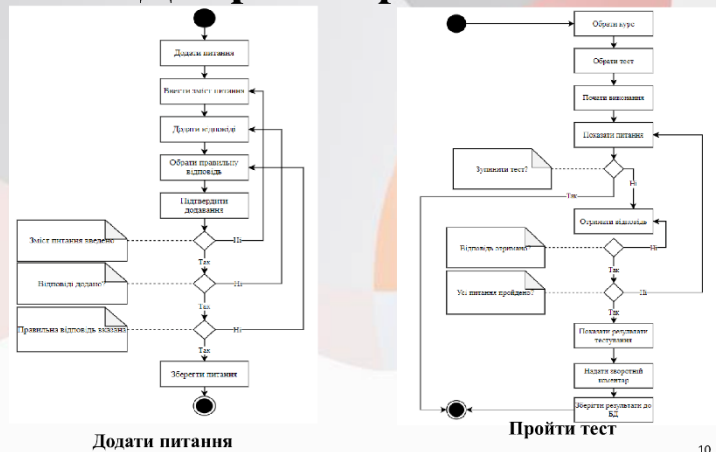
Діаграма варіантів використання



9

Рисунок Е.9 – Слайд 9

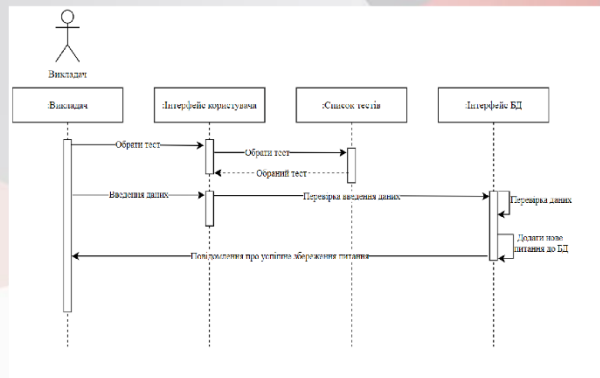
Діаграми прецедентів



10

Рисунок Е.10 – Слайд 10

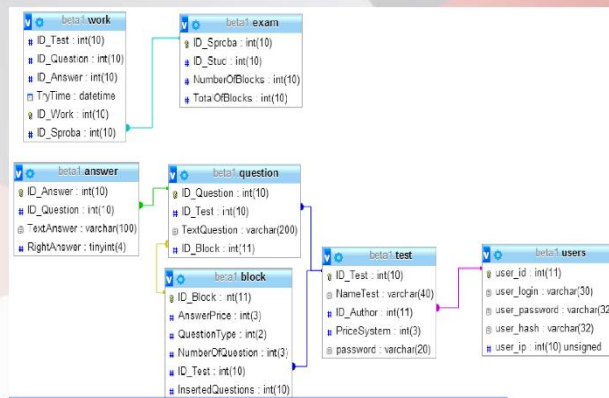
Діаграма послідовності прецеденту «Додати питання»



11

Рисунок Е.11 – Слайд 11

Схема бази даних



12

Рисунок Е.12 – Слайд 12

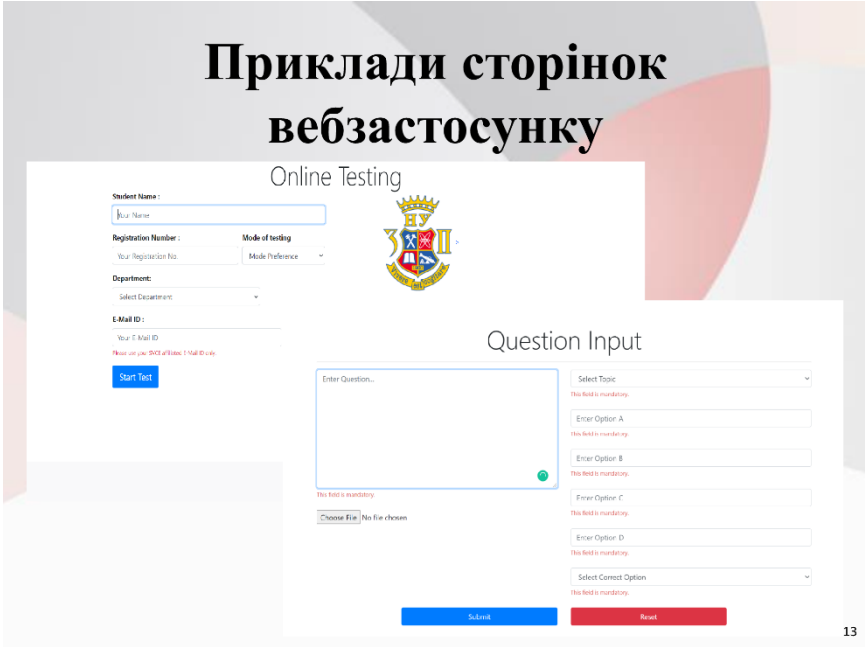


Рисунок Е.13 – Слайд 13

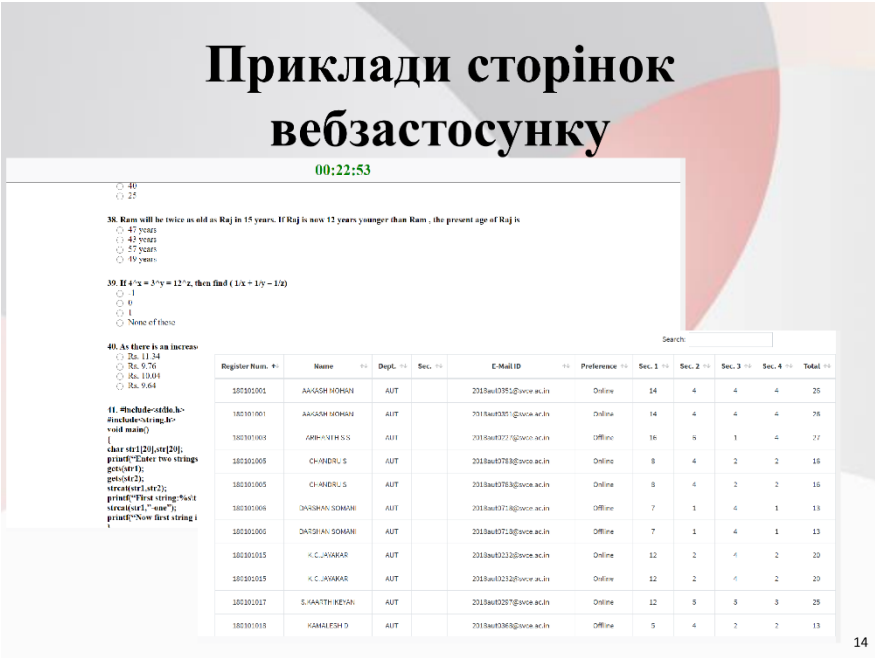


Рисунок Е.14 – Слайд 14

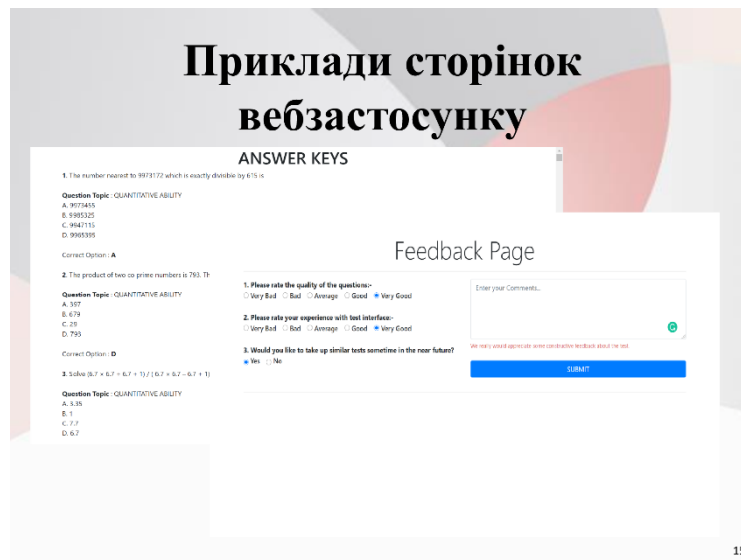


Рисунок Е.15 – Слайд 15

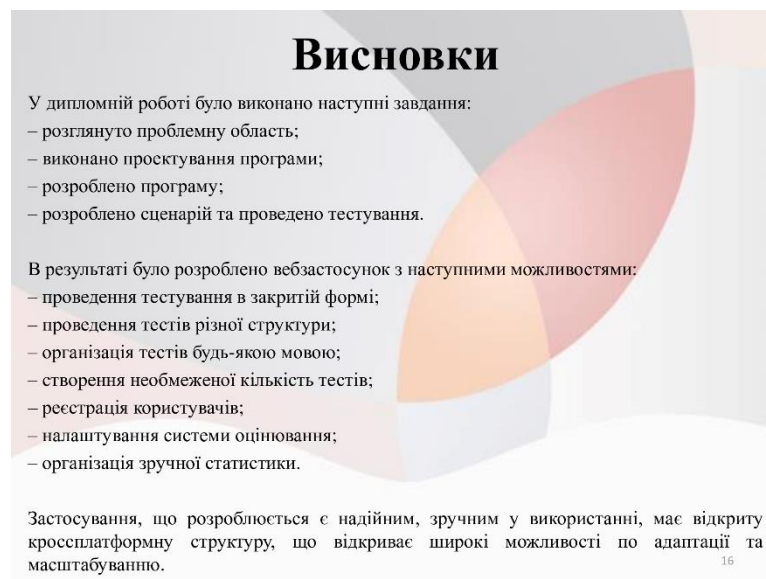


Рисунок Е.16 – Слайд 16

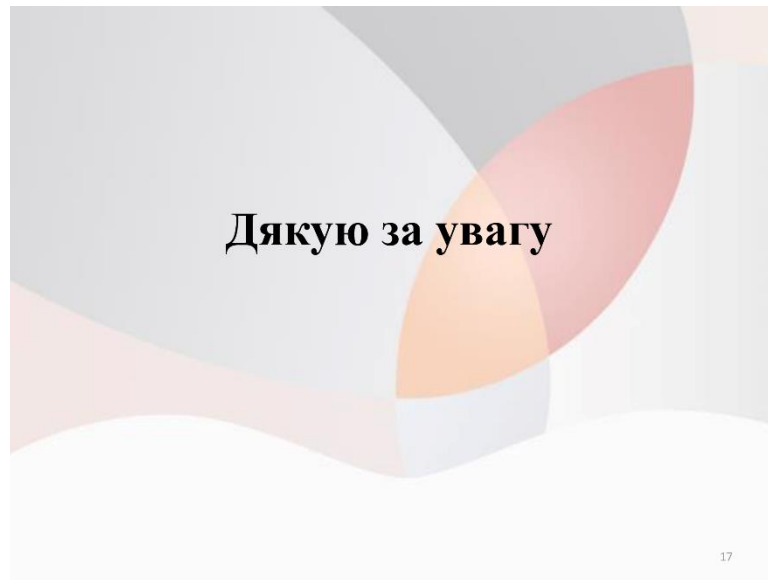


Рисунок Е.17 – Слайд 17