

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕКОНОМІКИ І МЕНЕДЖМЕНТУ**

Кафедра кібернетики та інформатики

КВАЛІФІКАЦІЙНА РОБОТА

освітній ступінь - «Бакалавр»

**на тему: Розробка інформаційної системи перевірки практичних
робіт із вибірки даних**

Виконав:

студент спеціальності

**126 «Інформаційні системи та технології»
ОП «Інформаційні системи та технології»
*Єременко Олександр Володимирович***

Керівник:

Агаджанова Світлана Володимирівна
доцент кафедри кібернетики та інформатики

Рецензент:

Яценко Валерій Валерійович
доцент кафедри економічної кібернетики
ННІ бізнесу, економіки та менеджменту
Сумського державного університету

СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

Факультет	Економіки і менеджменту
Кафедра	Кібернетики та інформатики
Освітній ступінь	«Бакалавр»
Спеціальність	126 «Інформаційні системи та технології» ОП «Інформаційні системи та технології»

Затверджую:
Завідувач кафедри Світлана АГАДЖАНОВА

« _____ » « _____ » 2023 ____ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ *Єременку Олександрову Володимировичу*

1. Тема роботи: Розробка інформаційної системи перевірки практичних робіт із вибірки даних

2. База практичного дослідження: Сумський НАУ.

керівник роботи Агаджанова Світлана Володимирівна к.т.н., доцент
затверджена наказом по університету від « _____ » _____ 2023 р. № _____ -н

3. Строк подання студентом закінченого проекту (роботи) «23» червня 2023 року

4. Вихідні дані до проекту (роботи):

Результати аналітичних досліджень зарубіжних та вітчизняних вчених, інтернет-джерела, нормативні документи за чинним законодавством.

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

- Виконати постановку задачі;
- Вибрати середовище для реалізації;
- розробити технічне завдання;
- розробити програмний продукт відповідно до поставлених задач;
- розробити інструкцію для користувача

6. Дата видачі завдання:

« »

2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/П	Назва етапів дипломного проекту(роботи)	Строк виконання етапів проекту	Примітка
1	Визначення об'єкту, предмету дослідження, формулювання мети та задач кваліфікаційної роботи, складання плану	вересень-жовтень 2022 р.	виконано
2	Підбір та вивчення літературних джерел, технічної документації	листопад- грудень 2022 р.	виконано
3	Розробка технічного завдання	січень 2023 р..	виконано
4	Розробка веб-додатку	лютий-березень 2023 р.	виконано
5	Створення інструкції користування веб-додатком для користувачів	до 10.05.2023 р.	виконано
6	Оформлення пояснювальної записки	до 30.05.2023 р..	виконано
7	Оформлення презентації	до 1.06.2023 р.	виконано
8	Опрацювання зауважень керівника, перевірка на автентичність	02.06.2023 р.	виконано
9	Підготовка доповіді та її попередній захист	06.06.2023 р.	виконано
10	Перевірка з академічної доброчесності кваліфікаційної роботи	09.06.2023 р.	виконано
11	Представлення кваліфікаційної роботи на кафедрі, деканат	23.06.2023 р.	виконано
12	Захист кваліфікаційної роботи	28.06.2023р.	виконано

Студент

(підпис)

Олександр ЄРЕМЕНКО

Керівник роботи

(підпис)

Світлана АГАДЖАНОВА

Перевірку на автентичність проведено

(підпис)

Надія БАРАНИК

Дипломна робота допущена до захисту

(підпис)

Маргарита ЛИШЕНКО

АНОТАЦІЯ

Єременко О.В. Розробка інформаційної системи перевірки практичних робіт із вибірки даних.

Кваліфікаційна робота зі спеціальності 126 «Інформаційні системи та технології» ОП Інформаційні системи та технології, СНАУ, Суми-2023р. – Рукопис.

Ця робота присвячена розробці інформаційної системи для перевірки практичних робіт з вибірки даних. Інформаційна система має на меті автоматизувати процес перевірки практичних робіт студентів з вибірки даних, забезпечуючи швидкий та ефективний збір, обробку та аналіз результатів.

Система включає в себе різноманітні функції, що допомагають викладачам та асистентам у проведенні перевірки практичних завдань. Вона надає можливість завантажувати практичні роботи студентів у визначений формат, що сприяє їхній однорідності та порівнянню. Далі система автоматично виконує перевірку робіт за заданими критеріями та алгоритмами, аналізує отримані результати і генерує звіт про оцінки та коментарі до кожної роботи.

Крім того, система має інтерфейс для студентів, де вони можуть завантажити свої практичні роботи та отримати результати перевірки. Це сприяє прозорості та взаємодії між викладачами та студентами, а також допомагає студентам зрозуміти свої помилки та вдосконалювати навички.

Розроблена інформаційна система забезпечує значну економію часу та зусиль для викладачів, що пов'язані з перевіркою практичних робіт студентів. Вона також сприяє об'єктивності та послідовності в оцінюванні робіт, забезпечуючи більш справедливий процес для всіх сторін. Загалом, розроблена інформаційна система є корисним інструментом для університетів та навчальних закладів, які займаються навчанням та оцінюванням практичних навичок з вибірки даних.

Загалом, розробка інформаційної системи перевірки практичних робіт з вибірки даних має великий потенціал для покращення якості навчання та ефективності оцінювання студентів. Вона спрощує та автоматизує процес перевірки робіт, забезпечуючи більш об'єктивні та консистентні результати.

Ключові слова: PHP, база даних, SQL, веб-додаток, лабораторні роботи.

SUMMARY

Yeremenko O.V. Development of an information system for checking practical works on data sampling.

Qualification work in the specialty 126 "Information Systems and Technologies" OP Information Systems and Technologies, SNAU, Sumy-2023 - Manuscript.

This work is devoted to the development of an information system for checking practical work on data sampling. The information system aims to automate the process of checking students' practical work on data sampling, providing fast and efficient collection, processing and analysis of results.

The system includes a variety of functions that help teachers and assistants in checking practical assignments. It provides the ability to upload students' practical works in a specific format, which facilitates their homogeneity and comparison. Then the system automatically checks the papers according to the specified criteria and algorithms, analyzes the results and generates a report on the grades and comments for each paper.

In addition, the system has an interface for students, where they can upload their practical work and get the results of the check. This promotes transparency and interaction between teachers and students, and helps students understand their mistakes and improve their skills.

The developed information system provides significant savings in time and effort for teachers related to the examination of students' practical work. It also promotes objectivity and consistency in the evaluation of work, ensuring a fairer process for all parties. All in all, the developed information system is a useful tool for universities and training institutions involved in teaching and assessing practical skills in data sampling.

In general, the development of an information system for checking practical work in data sampling has great potential to improve the quality of teaching and the efficiency of student assessment. It simplifies and automates the process of checking papers, providing more objective and consistent results.

Key words: PHP, database, SQL, web application, laboratory work.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	
1.1 Огляд останніх досліджень і публікацій.....	8
1.2 Аналіз продуктів-аналогів.....	9
1.3 Мета та задачі дослідження	14
1.4 Засоби реалізації.....	14
РОЗДІЛ 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗОВАНОЇ ПЕРЕВІРКИ ЛАБОРАТОРНИХ РОБІТ З ВИБІРКИ ДАНИХ	
2.1 Структурно функціональне моделювання процесу «Організація виконання та оцінювання завдань з лабораторних робіт»	17
2.2 Проектування моделі бази даних	20
2.3 Моделювання варіантів використання.....	22
РОЗДІЛ 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗОВАНОЇ ПЕРЕВІРКИ ЛАБОРАТОРНИХ РОБІТ З ВИБІРКИ ДАНИХ	
3.1 Архітектура програмного забезпечення	25
3.2 Програмна реалізація	28
3.3 Використання інформаційної системи.....	34
ВИСНОВКИ.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	47
ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ	50
ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ	58
ДОДАТОК В. КОПІЇ ПУБЛІКАЦІЙ	66
ДОДАТОК Г. ЛІСТИНГ ФАЙЛІВ ПРОГРАМНОГО КОДУ	74

ВСТУП

Розробка інформаційної системи перевірки практичних робіт із вибірки даних є актуальним завданням у сучасному світі, де зростає значення технологій та автоматизації процесів. Практичні роботи є важливою складовою навчального процесу в багатьох освітніх закладах та курсах, але їх перевірка та оцінювання може бути трудомісткою та часоємною задачею для викладачів.

Інформаційна система перевірки практичних робіт із вибірки даних пропонує ефективний та автоматизований підхід до оцінювання студентів, що дозволяє викладачам значно збільшити продуктивність та зосередитися на якісному навчанні. Ця система базується на аналізі вибірки даних, яку студенти виконують у вигляді практичних завдань.

Основна мета розробки цієї інформаційної системи полягає в тому, щоб забезпечити викладачам можливість швидкого та об'єктивного оцінювання практичних робіт студентів, а також створити зручне середовище для навчання та самостійного розвитку студентів.

Система перевірки практичних робіт із вибірки даних пропонує такі можливості:

- Автоматизована перевірка практичних завдань з використанням алгоритмів та методів аналізу даних.
- Об'єктивна оцінка результатів студентів на основі критеріїв, заданих викладачем.
- Статистичний аналіз та візуалізація результатів для забезпечення звітності та аналізу прогресу студентів.
- Зручне середовище для студентів, де вони можуть виконувати практичні завдання, завантажувати свої роботи та отримувати детальний звіт про свої успіхи та помилки.
- Можливість надання зворотного зв'язку та рекомендацій студентам для поліпшення їх робіт та розвитку навичок.
- Керування групами студентів, розподіл завдань та призначення викладачів для їх перевірки.

- Забезпечення конфіденційності та безпеки даних студентів та викладачів.
- Інтеграція з існуючими освітніми платформами та системами, що дозволяє спрощувати робочий процес викладачів та студентів.

Розробка інформаційної системи перевірки практичних робіт із вибірки даних має значний потенціал у покращенні якості освіти, забезпеченні об'єктивності та ефективності процесу оцінювання. Вона дозволяє зосередитися на розвитку студентів та забезпеченні їх успіху шляхом надання конструктивного зворотного зв'язку та індивідуального підходу до кожного студента.

Розробка такої системи вимагає використання сучасних технологій, аналітичних методів та експертного досвіду. Правильна імплементація цієї системи може стати важливим кроком у поліпшенні якості навчання та підготовки майбутніх фахівців.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Огляд останніх досліджень і публікацій пов'язаний з розробкою інформаційних систем для перевірки практичних робіт із вибірки даних відображає актуальні тенденції та досягнення у цій галузі. Наведу деякі приклади останніх досліджень та публікацій, що стосуються даної тематики:

"Automated Assessment Systems for Practical Assignments in Computer Science Education"[2] - дослідження, що розглядає використання автоматизованих систем оцінювання для практичних завдань у навчанні комп'ютерних наук. Автори дослідження висвітлюють переваги та виклики, пов'язані зі впровадженням таких систем, а також пропонують методи та рекомендації для їх розробки та використання.

"Intelligent Tutoring Systems for Practical Exercises in Data Analysis"[6] - публікація, що досліджує використання інтелектуальних систем підтримки навчання для практичних вправ з аналізу даних. Автори пропонують моделі та алгоритми для автоматичної перевірки та оцінювання робіт студентів з фокусом на індивідуальному навчанні та адаптивному підході.

"Assessment of Practical Skills in Online Learning Environments"[1] - дослідження, що розглядає проблематику оцінювання практичних навичок у онлайн-середовищах навчання. Автори дослідження аналізують різні методи та підходи до перевірки та оцінювання практичних робіт та роблять акцент на важливості забезпечення об'єктивності та надійності оцінювання.

Ці дослідження та публікації свідчать про постійний інтерес до розробки інформаційних систем для перевірки практичних робіт із вибірки даних і їх впровадження у навчальні процеси. Вони підкреслюють важливість автоматизації та об'єктивності оцінювання, а також підтримки індивідуального навчання та адаптивного підходу до студентів.

Деякі з цих досліджень пропонують використання інтелектуальних систем, таких як системи підтримки навчання та аналітичних алгоритмів, що дозволяють

автоматично аналізувати та оцінювати роботи студентів на основі заданих критеріїв. Інші дослідження розглядають використання онлайн-середовищ та віддалених засобів для зручності та доступності перевірки практичних завдань.

Однак, незважаючи на ці досягнення, є деякі виклики, що потребують уваги у розробці інформаційних систем перевірки практичних робіт із вибірки даних. Серед них - складність автоматичного розпізнавання креативності та індивідуального підходу студентів, а також забезпечення конфіденційності та безпеки даних.

1.2 Аналіз продуктів-аналогів

Серед програмних продуктів, які вирішують схожі завдання, як поставлені у даному проекті, можна відмітити деякі веб-орієнтовані рішення для вивчення мови SQL. Усі ці рішення надають можливість виконувати практичні завдання шляхом написання запитів до тренувальної бази даних.

Для тренування навичок створення вибірки даних існує інтерактивний онлайн тренажер мови SQL під назвою SQLCourse [13]. На рисунку 1.1 зображено зовнішній вигляд сторінки тренування на цьому онлайн тренажері.

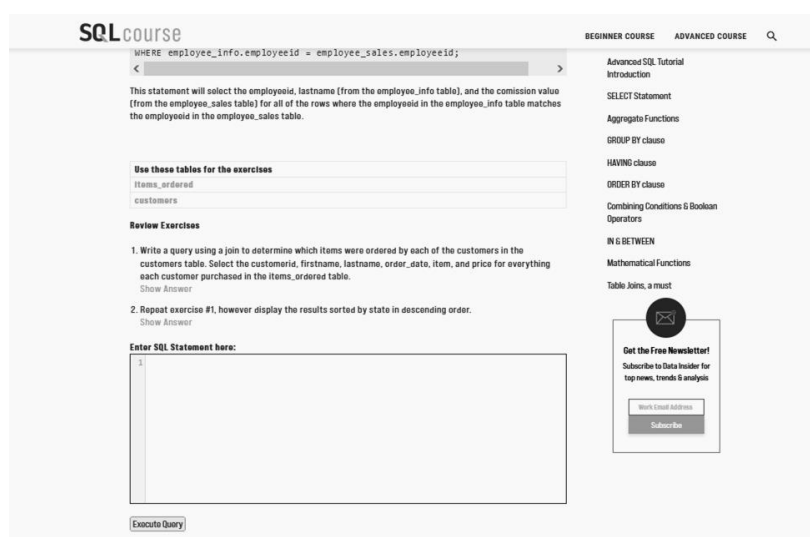


Рисунок 1.1 – Сторінка для тренування навичок створення вибірки даних SQLCourse

Джерело: [13]

SQLCourse[13] має структуру, що нагадує підручник для вивчення мови SQL та систем управління базами даних. При користуванні цим онлайн тренажером, користувач може обрати певну тему, ознайомитися з теоретичним матеріалом та прикладами побудови запитів, пов'язаних з цією темою. В кінці тренування, користувач може створити запит до тестової бази даних та отримати практичні навички.

Інтерфейс SQLCourse може виглядати застарілим, але він простий у використанні, тому знаходження необхідних опцій не викликає проблем. Однак, функціональний модуль, який використовується для перевірки введених користувачем запитів та виведення результатів на екран, має деякі проблеми і не завжди відображає отриманий результат у відповідному форматі.

Інтерактивний онлайн тренажер мов програмування – Codecademy [4]. Вигляд сторінки тренування навичок мов програмування для цього онлайн тренажера наведено на рисунку 1.2.

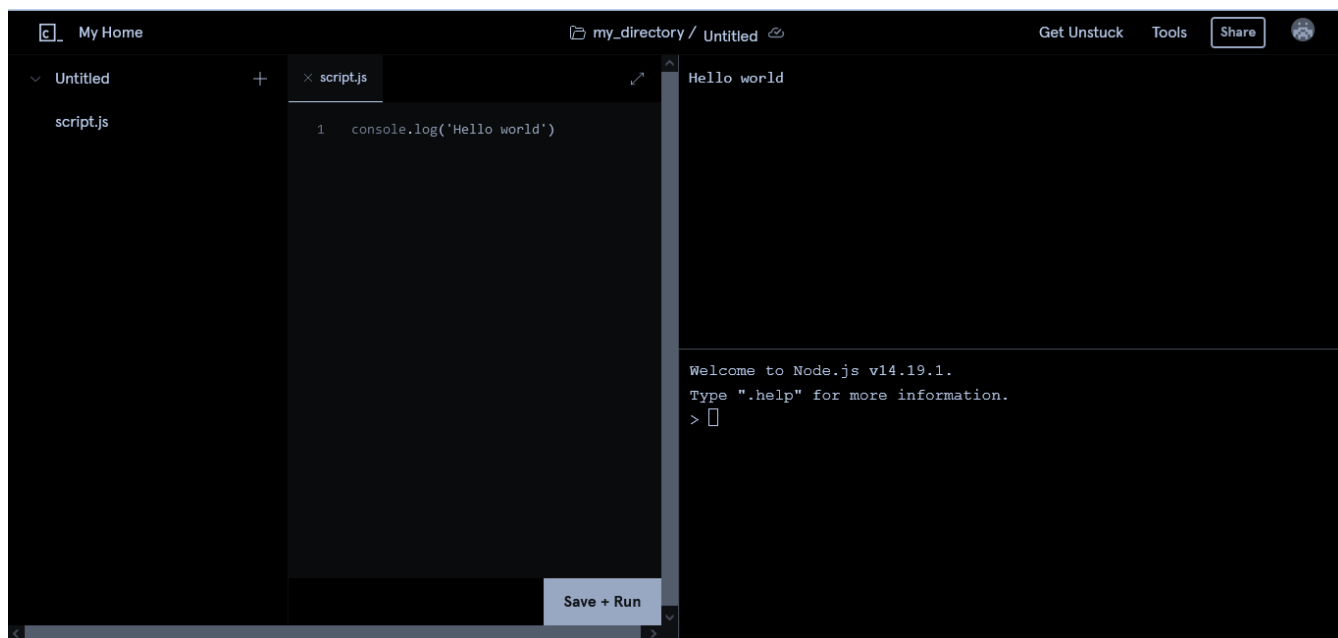


Рисунок 1.2 – Сторінка для тренування навичок мов програмування Codecademy

Джерело: [4]

Codecademy[4] є платформою для навчання програмуванню, включаючи також курси з мови SQL. Цей ресурс пропонує широкий спектр онлайн-курсів, що дозволяє навчитися програмуванню з нуля або покращити вже наявні навички.

Курси з мови SQL на Codecademy охоплюють різні рівні від початкового до більш продвинутого. Ви можете вчитися основам SQL, роботі з базами даних, складним запитам, операціям з даними та іншим аспектам мови SQL. Кожен курс має структурований навчальний план, доповнений завданнями, практичними вправами та підтримкою спільноти для обміну досвідом та отримання допомоги.

Загальний досвід навчання на Codecademy є інтерактивним, практичним та орієнтованим на результат. Цей ресурс може бути відмінним вибором для тих, хто бажає вивчити мову SQL або покращити свої навички у цій області.

Платформа для оцінювання навичок спеціалістів CodeSignal [5]. Вигляд сторінки тренування навичок програмування наведено на рисунку 1.3.

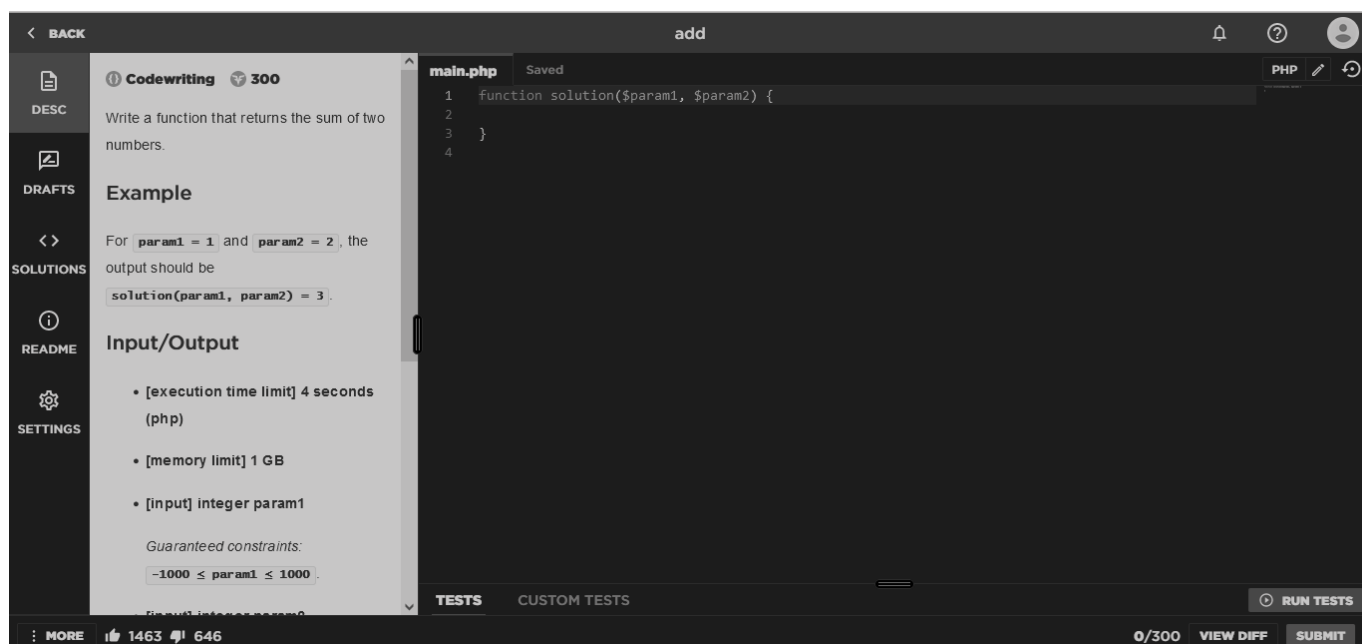


Рисунок 1.3 – Сторінка для тренування навичок мов програмування CodeSignal

Джерело: [5]

CodeSignal[5] - це онлайн-платформа для тестування та оцінювання навичок програмування. Вона пропонує широкий вибір завдань, які охоплюють різні мови програмування та рівні складності.

Основні особливості CodeSignal включають:

1. Завдання: CodeSignal надає багато різноманітних програмувальних завдань, які допомагають вам вдосконалити навички програмування. Ці завдання можуть бути

орієнтовані на алгоритмічне програмування, структури даних, бази даних, веб-розробку та багато іншого.

2. Аналіз коду: CodeSignal надає можливість аналізувати ваш код та отримувати рекомендації щодо його вдосконалення. Це допомагає покращити стиль коду, ефективність та читабельність вашого програмування.

3. Поділ на рівні: CodeSignal пропонує поділ завдань на рівні складності, що дозволяє вам вибрати завдання, які відповідають вашому рівню навичок.

4. Конкурси та рейтинг: CodeSignal проводить регулярні конкурси, де ви можете порівняти свої навички з іншими програмістами та отримати рейтинг за ваші досягнення.

5. Підготовка до співбесіди: CodeSignal також пропонує підготовку до технічних співбесід, де ви можете практикуватися вирішенням типових завдань, які часто ставляться на співбесідах.

CodeSignal є популярною платформою для програмістів, яка допомагає розвивати навички програмування та підготуватися до технічних співбесід.

Інтерактивний онлайн тренажер мови SQL – SQLZoo [14]. Вигляд сторінки тренування навичок створення вибірки даних для цього онлайн тренажера наведено на рисунку 1.4.

SELECT basics

Language: English • Deutsch • español • 日本語 • 中文

name	continent	area	population	gdp
Afghanistan	Asia	652230	25500100	20343000000
Albania	Europe	28748	2831741	12960000000
Algeria	Africa	2381741	37100000	168681000000
Andorra	Europe	468	78115	3712000000
Angola	Africa	1246700	20609294	100990000000
...				

Introducing the world table of countries

1. The example uses a WHERE clause to show the population of France. Note that strings (pieces of text that are data) should be in 'single quotes'.

Modify it to show the population of Germany

SELECT population FROM world
WHERE name = "France"

Submit SQL Restore default

result

Рисунок 1.4 – Сторінка для тренування навичок створення вибірки даних SQLZoo

Джерело: [14]

SQLZoo[14] - це інтерактивний онлайн-ресурс, який надає навчальні матеріали та завдання для вивчення мови SQL. Він створений для того, щоб допомогти користувачам освоїти основи SQL та практично застосувати свої знання.

Основні особливості SQLZoo включають:

1. Інтерактивні уроки: SQLZoo пропонує інтерактивні уроки, які дозволяють вам навчатися та виконувати SQL-запити прямо на веб-сторінці. Ви можете виконувати запити, перевіряти їх результати та отримувати зворотній зв'язок щодо виконання завдань.

2. Багато різноманітних завдань: SQLZoo містить велику кількість завдань, які охоплюють різні аспекти мови SQL. Ви можете вивчати основи запитів SELECT, використовувати JOIN-операції, виконувати агрегатні функції, працювати з підзапитами та багато іншого.

3. Різні рівні складності: SQLZoo пропонує завдання різного рівня складності, починаючи з основних запитів і до більш складних випадків. Це дозволяє вам прогресувати в навичках SQL поступово, розвиваючи свої знання.

4. Реальні дані: SQLZoo надає можливість працювати з реальними наборами даних, що допомагає вам навчитися виконувати запити на реальних прикладах.

5. Додаткові матеріали: Крім інтерактивних завдань, SQLZoo надає додаткові матеріали, які допомагають розібратися в конкретних аспектах SQL та надають додаткову підтримку в процесі вивчення.

SQLZoo є популярним ресурсом для вивчення мови SQL.

Таблиця 1.1 – Порівняльна характеристика веб-додатків-аналогів

Назва сайту	SQLCourse	Codecademy	CodeSignal	sqlzoo	db-sql-practics
Перегляд правильної відповіді	-	+	+	+	+
Обмеження на повторні спроби	-	-	-	-	+
Вказує на помилки	+	+	-	-	-
Публічні результати	+	-	-	-	-
Підказки	+	+	-	-	-

Джерело: [4, 5, 14, 13]

Після ретельного аналізу переваг та недоліків зазначених аналогів можна визначитися з актуальністю веб-додатків і зрозуміти необхідність сучасного та інтуїтивно зрозумілого інтерфейсу, а також повного функціоналу для керування користувачами, редагування тестових завдань та автоматичної перевірки відповідей у розроблюваному програмному продукті.

1.3 Мета та задачі дослідження

Основною метою дипломного проекту є створення web-додатку, який забезпечує автоматизовану перевірку лабораторних робіт з написання SQL-запитів до бази даних.

Проект має бути реалізований у формі веб-сайту.

Для досягнення встановленої мети необхідно вирішити наступні завдання:

- Сформулювати та узгодити технічне завдання;
- Провести аналіз предметної області;
- Створити базу даних;
- Розробити шаблон для веб-сайту;
- Реалізувати систему перевірки SQL-запитів;

Аналіз предметної області дозволив визначити наступні функціональні вимоги до розроблюваної системи:

1. Зберігання результатів робіт студентів;
2. Підтримка збереження/виконання результатів завдань;
3. Управління лабораторними завданнями;
4. Функція авто перевірки лабораторних робіт ;
5. Імпорт виставлених оцінок до Excel;
6. Можливість перевірки викладачем;
7. Кабінети для студентів та викладача.

1.4 Засоби реалізації

Для зручної розробки програмного забезпечення ІС застосовується редактор коду Visual Studio Code(рис.1.5), який є потужним інструментом для створення та

налагодження веб-додатків. Цей редактор дозволяє використовувати різноманітні розширення, що полегшують рефакторинг та аналіз програмного коду.

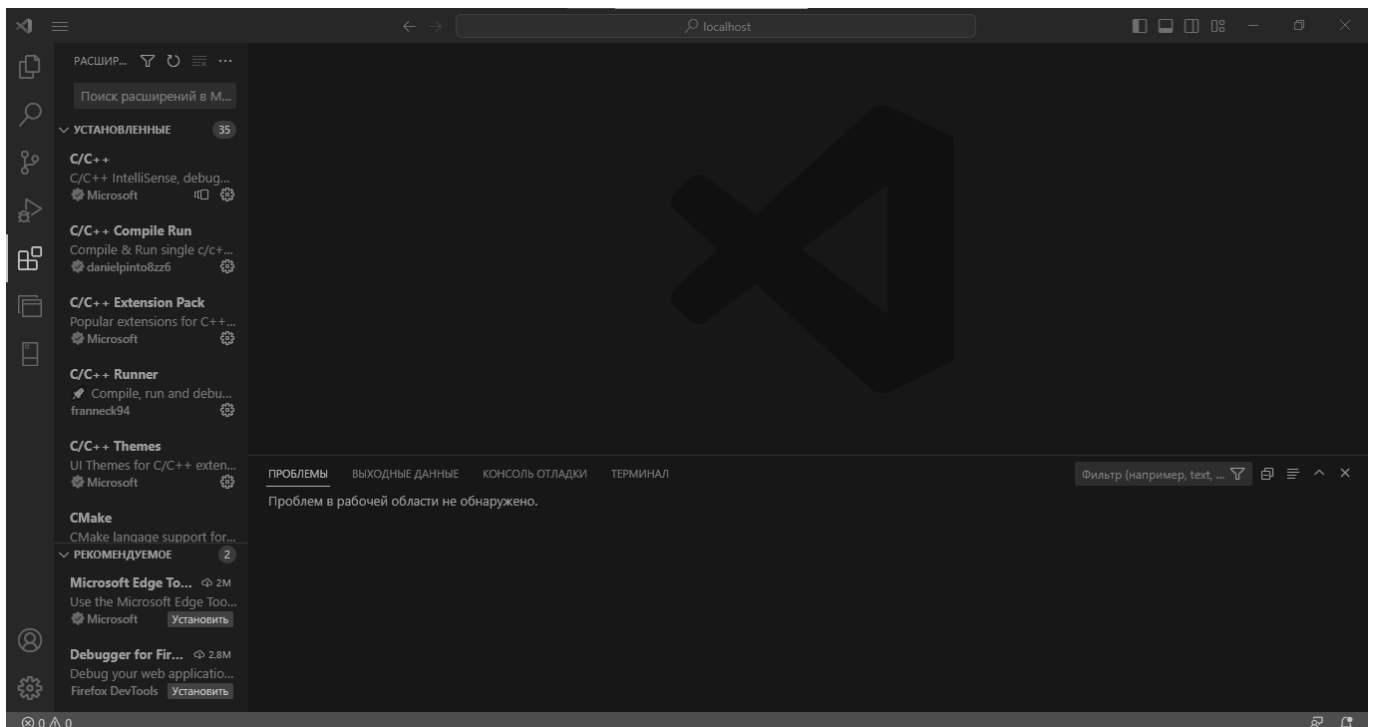


Рисунок 1.5 – Вигляд Visual Studio Code

Джерело: [11]

Для розробки застосунку використовується локальне середовище - Open Server Panel (OSPanel) (рис. 1.6). OSPanel є переносним програмним забезпеченням, яке надає локальний веб-сервер та набір інструментів для розробки серверного програмного забезпечення.

Одним з важливих компонентів OSPanel є phpMyAdmin, який використовується для адміністрування MySQL через Інтернет. PhpMyAdmin надає можливість виконувати операції з базою даних через інтерфейс або за допомогою текстового редактора для SQL-запитів.

Згідно з функціональними вимогами до серверної частини, було вирішено використовувати систему управління базами даних MySQL версії 8.0, а також мову програмування PHP версії 7.2.

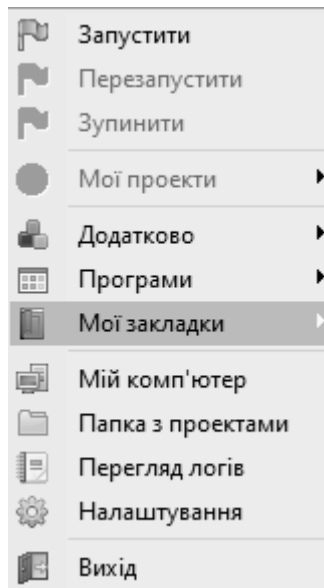


Рисунок 1.6 – Інтерфейс open server panel

Джерело: [7]

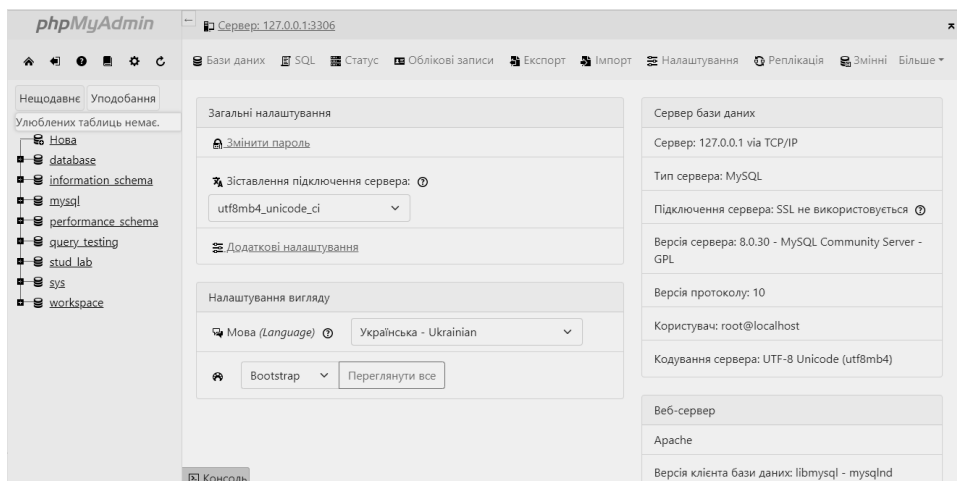


Рисунок 1.7 – Інтерфейс phpMyAdmin

Джерело: [7]

Клієнтська частина передбачає розробку користувацького інтерфейсу, який відображатиметься в обраному користувачем браузері. Інтерфейс має бути універсальним та адаптивним, тому для створення структури web-сторінок використовується мова розмітки HTML5, для стилізації цих сторінок – таблиці стилів CSS3

РОЗДІЛ 2

ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗОВАНОЇ ПЕРЕВІРКИ ЛАБОРАТОРНИХ РОБІТ З ВИБІРКИ ДАНИХ

2.1 Структурно функціональне моделювання процесу «Організація виконання та оцінювання завдань з лабораторних робіт»

Моделювання - це метод дослідження, який використовує створення та аналіз спрощених замінників реальних об'єктів для отримання пізнавальних результатів. Вихідний об'єкт, який використовується для створення моделі, називається прототипом або оригіналом. Модель є замінником цього об'єкта.

Один з основних мотивів для створення моделей полягає в зменшенні ризиків і затрат, збереженні часу розробника та пошуку оптимальних рішень.

IDFE[8] (Integrated Definition for Function Modeling) - це набір взаємопов'язаних блоків, де стрілки вказують вхідні та вихідні дані, а також управління (керуючі об'єкти) та механізми. Основною метою створення IDEF-діаграм є побудова ієрархічної системи.

Вхідні дані для перевірки та оцінки студентів включають:

- Номер лабораторної роботи;

Вихідними даними є:

- Таблиця успішності;
- Оцінки.

Керуючі об'єкти включають:

- Еталонні запити;
- Рівні доступу;
- Завдання;
- Вхідну базу даних.

Механізми включають:

- Адміністратора;
- Студента;
- Систему.



Рисунок 2.1 – Контекстна діаграма

Джерело: запропоноване автором

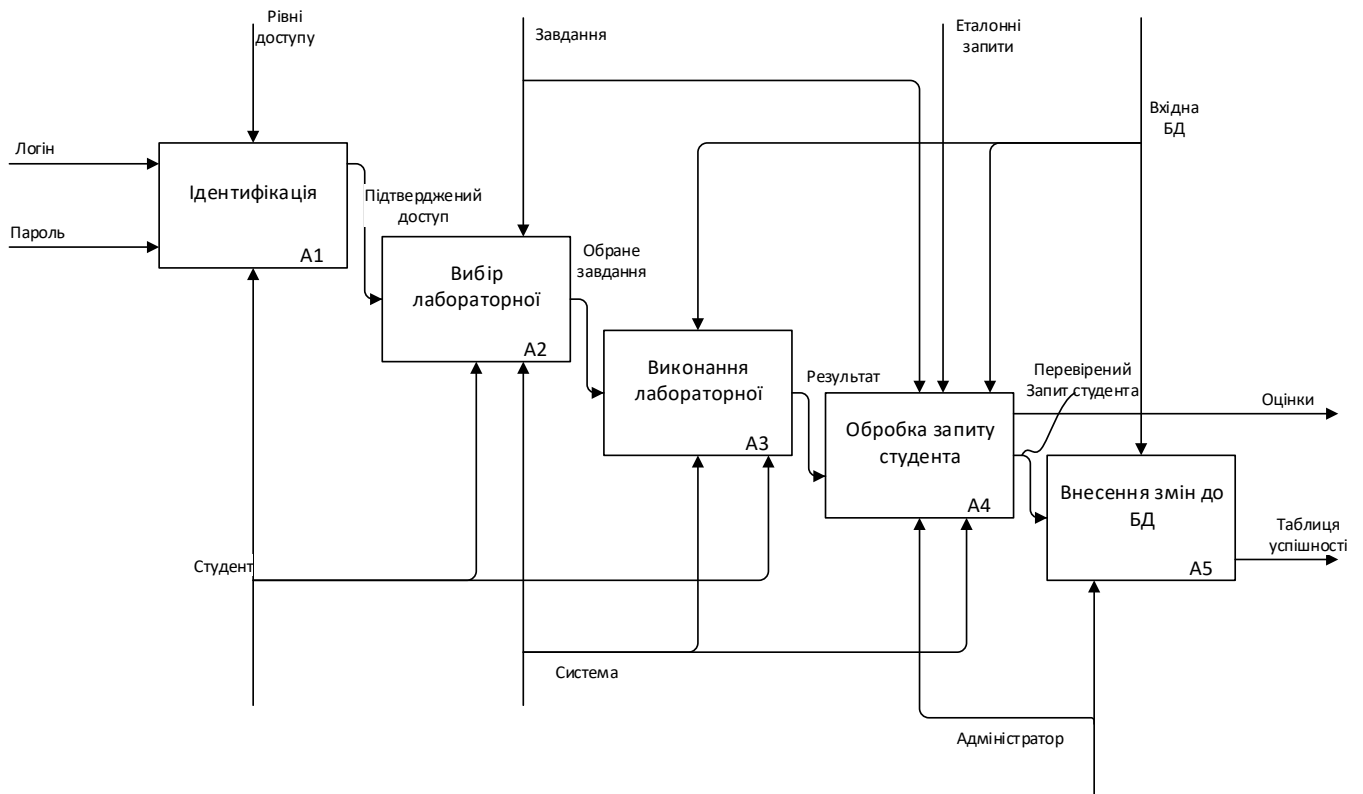


Рисунок 2.2 – Діаграма декомпозиції 1-го рівня

Джерело: запропоноване автором

При побудові діаграми декомпозиції були використані такі блоки:

- Ідентифікація;

- Вибір лабораторної роботи;
- Виконання роботи;
- Обробка запиту студента;
- Внесення змін до бази даних;

Головне завдання даного проекту полягало у впровадженні обробки запитів студентів. Для детального вивчення цього процесу була розроблена декомпозиція обробки запиту студента (див. рис. 2.3).

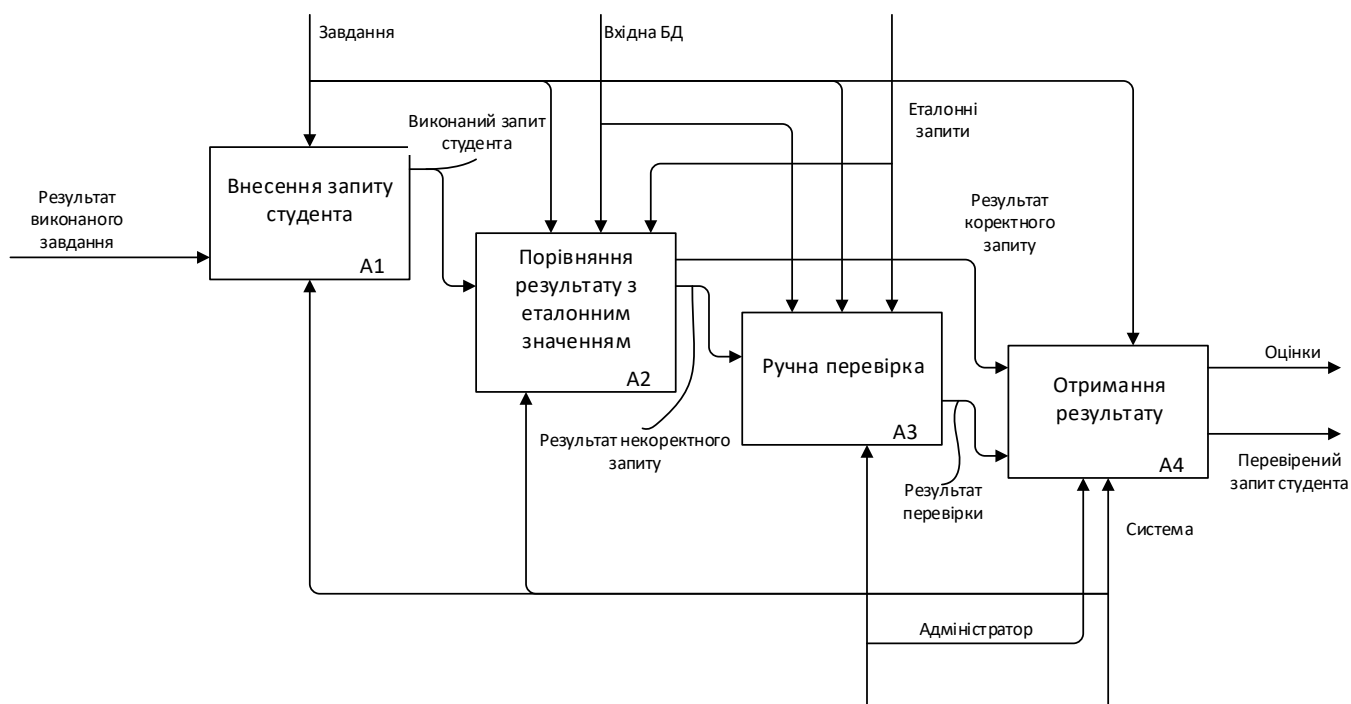


Рисунок 2.3 – Діаграма декомпозиції 2-го рівня «обробка запиту студента»

Джерело: запропоноване автором

В даний момент розробляється веб-додаток, призначений для оцінювання практичних навичок студентів. Основними користувачами цієї системи є студенти, які можуть отримувати завдання з лабораторних робіт, переглядати свої оцінки та проглядати інформацію про свої попередні практичні роботи. Крім того, адміністратор також може використовувати веб-додаток і мати повноваження вносити зміни в дані.

2.2 Проектування моделі бази даних

Під час проектування моделі бази даних визначаються структура та взаємозв'язки між таблицями, що використовуються для збереження даних. Основні кроки проектування моделі бази даних включають:

1. Визначення сутностей: Встановлення основних сутностей (сутності представляють об'єкти, про які зберігається інформація) і їхніх взаємозв'язків в системі. Наприклад, у веб-додатку для управління студентськими записами можуть бути сутності "Студент" і "Група" з взаємозв'язком "належить до".

2. Визначення атрибутів: Для кожної сутності визначаються її атрибути (атрибути представляють характеристики сутностей). Наприклад, атрибути для сутності "Студент" можуть включати "Ім'я", "Прізвище", "Номер студента" тощо.

3. Визначення первинного ключа: Визначення унікального ідентифікатора для кожної сутності, який використовується для унікальної ідентифікації записів в таблицях бази даних. Наприклад, у сутності "Студент" первинним ключем може бути "Номер студента".

4. Визначення зв'язків: Встановлення взаємозв'язків між сутностями, що визначають, як вони пов'язані між собою. Це можуть бути зв'язки один до одного, один до багатьох або багато до багатьох. Наприклад, зв'язок між сутностями "Студент" і "Група" може бути один до багатьох (одна група має багатьох студентів).

5. Нормалізація: Використовується для оптимізації структури бази даних та уникнення аномалій при збереженні даних. Даний процес розбиває таблиці

База даних можна описати як місце зберігання інформації, де таблиці зв'язуються між собою за допомогою зв'язків. Вміст цих таблиць є постійно змінюваним та постійно поповнюється новою інформацією. В рамках проекту було проведено проектування бази даних згідно з поставленим завданням, а потім ця база даних була реалізована[8].

Після аналізування основних аспектів та використання web-додатку моделі, переходимо до створення структури бази даних. Для цього нам потрібно визначити імена таблиць, їх атрибути, типи даних, призначення та обмеження. Цю інформацію представлено в таблиці 2.1.

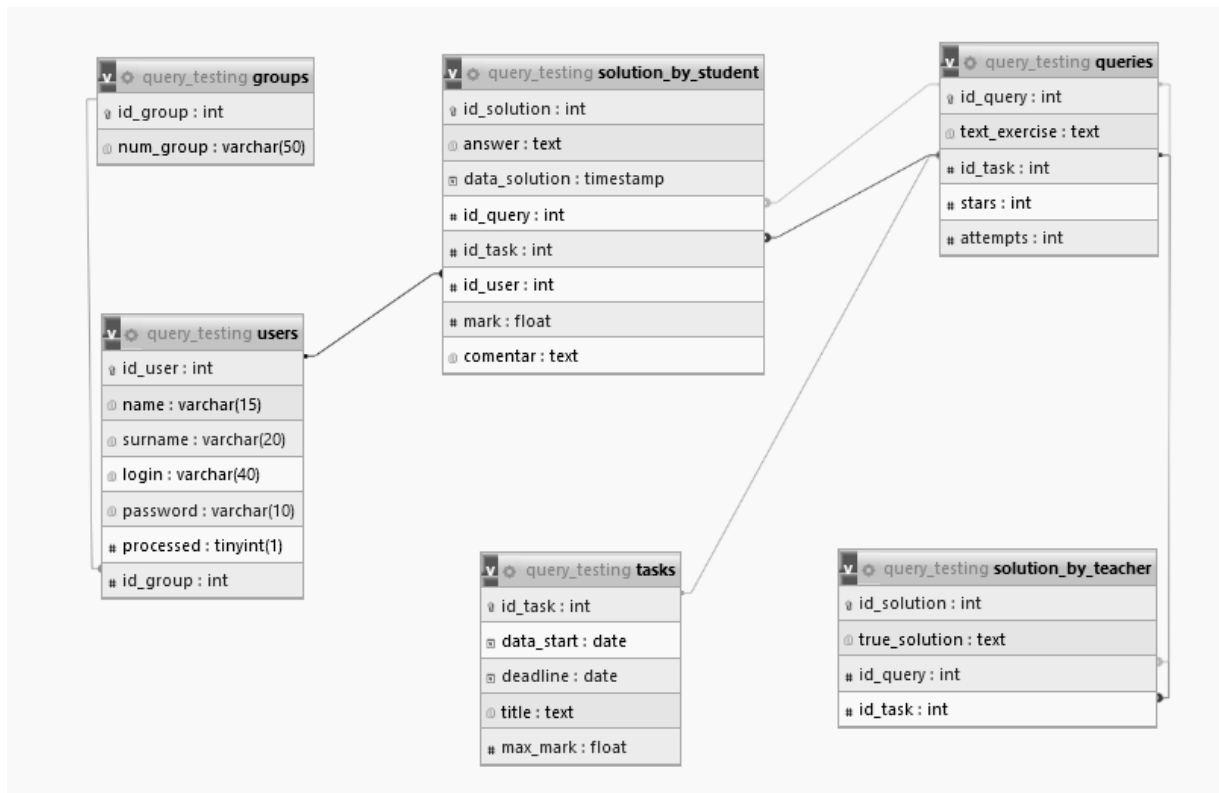


Рисунок 2.4 – Структура бази даних

Джерело: запропоноване автором

Таблиця 2.1 – Структура бази даних

Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
Tasks	Idt-asks	Номер	Int	PK	Автоінкремент
	Date-start	Початок роботи	Date	-	-
	Dead-line	Локаут на роботу	Date	-	-
	Title	Заголовок завдання	Text	-	-
	Max mark	Найвища оцінка	Float	-	-
User	Id-user	Номер студента	Int	PK	Автоінкремент
	Name	Ім'я користувача	Varchar	-	-
	Surname	Прізвище користувача	Varchar	-	-
	Login	Логін користувача	Varchar	-	-
	Pasword	Пароль користувача	Varchar	-	-
	Id-group	Група студента	Int	PFK	-
	Processed	Обробка на реєстрацію	Tinyint	-	-
Queries	Id query	№ Запиту	Int	PK	Автоінкремент
	Text exercise	Завдання	Text	-	-
	Id task	№ Задачі	Int	PFK	-
	Stars	Складність	Int	-	-
	Attempts	К-сть спроб	Int	-	-

Продовження таблиці 2.1

Solution-by-teacher	Id solution	№ Рішення	Int	PK	Автоінкремент
	True-solution	Еталонний запит	Text	-	-
	Id-query	№ Запиту	Int	FK	-
	I-task	№ Задачі	Int	FK	-
Solution-by-student	Id-solution	№ Рішення	Int	PK	Автоінкремент
	Answer	Відповідь	Text	-	-
	Date-solution	Дата рішення	Time-stamp	-	-
	Id-query	№ Запиту	Int	FK	-
	Id-task	№ Задачі	Int	FK	-
	Id-user	№ користувача	Int	PFK	-
	Mark	Оцінка	Float	-	-
	Comentar	Коментар	Text	-	-
Group	Id-group	№ Групи	Int	PK	Автоінкремент
	Num-group	Група	Varchar	-	

Джерело: запропоноване автором

2.3 Моделювання варіантів використання

Моделювання варіантів використання є процесом, який дозволяє аналізувати та визначати різні можливі сценарії використання системи або програмного продукту. Цей процес включає в себе ідентифікацію акторів (користувачів, системи, зовнішніх компонентів), опис послідовності дій, які вони можуть виконувати, та визначення залежностей між ними.

Моделювання варіантів використання дозволяє зрозуміти, як система буде взаємодіяти з користувачами та іншими компонентами, які функції будуть ними використовуватися і які результати будуть отримані. Це допомагає визначити вимоги до системи, спроектувати її архітектуру та забезпечити належну функціональність.

Моделювання варіантів використання використовує різні методи та інструменти, такі як діаграми варіантів використання (Use Case diagrams), сценарії, таблиці або специфікації. Цей процес допомагає забезпечити повноту та зрозумілість вимог до системи, а також виявити потенційні проблеми та ризики, що можуть виникнути в процесі використання системи



Рисунок 2.5 – Діаграма варіантів використання

Джерело: запропоноване автором

Діаграма варіантів використання представляє собою графічне зображення всіх можливих сценаріїв використання, акторів і зв'язків між ними [12]. Ці варіанти використання описують послідовність дій, які система виконує у відповідь на діяльність користувачів або інших програмних систем, а також вони відображають

функціональність самої системи. У цьому контексті актори можуть бути розглянуті таким чином:

- Користувач - людина, яка використовує програму.
- Адміністратор - людина, яка здійснює адміністрування системи.

У кожного актора є різні можливості використання. Студент має наступні права:

- Переглядати результати своїх виконаних робіт.
- Переглядати роботи, які він надсилав, включаючи невдалі спроби.
- Переглядати завдання для лабораторних робіт.

Адміністратор має наступні права:

- Переглядати результати виконаних студентами робіт.
- Переглядати роботи, надіслані студентами, включаючи їх невдалі спроби.
- Переглядати завдання для лабораторних робіт студентів.
- Редагувати, видаляти та додавати матеріали контенту.
- Перевіряти роботи вручну за необхідністю.
- Додавати запити, за якими проводиться перевірка.
- Експортувати оцінки у форматі Excel.
- Встановлювати стандарти для виконання робіт, такі як час та максимальний

бал.

РОЗДІЛ 3

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗОВАНОЇ ПЕРЕВІРКИ ЛАБОРАТОРНИХ РОБІТ З ВИБІРКИ ДАНИХ

3.1 Архітектура програмного забезпечення

Діаграма компонентів є візуальним зображенням компонентів системи та залежностей між ними. Вона демонструє структуру системи на високому рівні, показуючи компоненти, їхні взаємозв'язки та інтерфейси.

Діаграма компонентів використовується для моделювання архітектури системи, визначення ролей компонентів та їхніх взаємозв'язків, а також для виявлення залежностей між компонентами. Вона дозволяє розглянути систему як набір окремих компонентів, які співпрацюють між собою для досягнення загальних цілей.

У діаграмі компонентів компоненти представляються у вигляді прямокутників, а залежності між компонентами відображаються стрілками. Кожен компонент може мати інтерфейси, які вказують, як інші компоненти можуть взаємодіяти з ним. Інтерфейси зазвичай позначаються на діаграмі за допомогою портів або інших позначень[12].

Діаграма компонентів допомагає розібратися в структурі системи, виділити основні компоненти та їхні взаємозв'язки, а також сприяє розумінню архітектурних рішень та організації роботи системи. Вона є потужним інструментом для аналізу та проектування системи перед реалізацією.

Для забезпечення правильної функціональності сайту була розроблена діаграма компонентів, яка включає чіткі вказівки щодо внутрішніх методів, які є ключовими для реалізації даного завдання. Діаграма також відображає зв'язки між цими методами та зовнішніми користувачами, які використовують ці методи[12].

Для ілюстрації розподілу програмної системи на структурні компоненти та взаємозв'язки між ними використовується діаграма компонентів

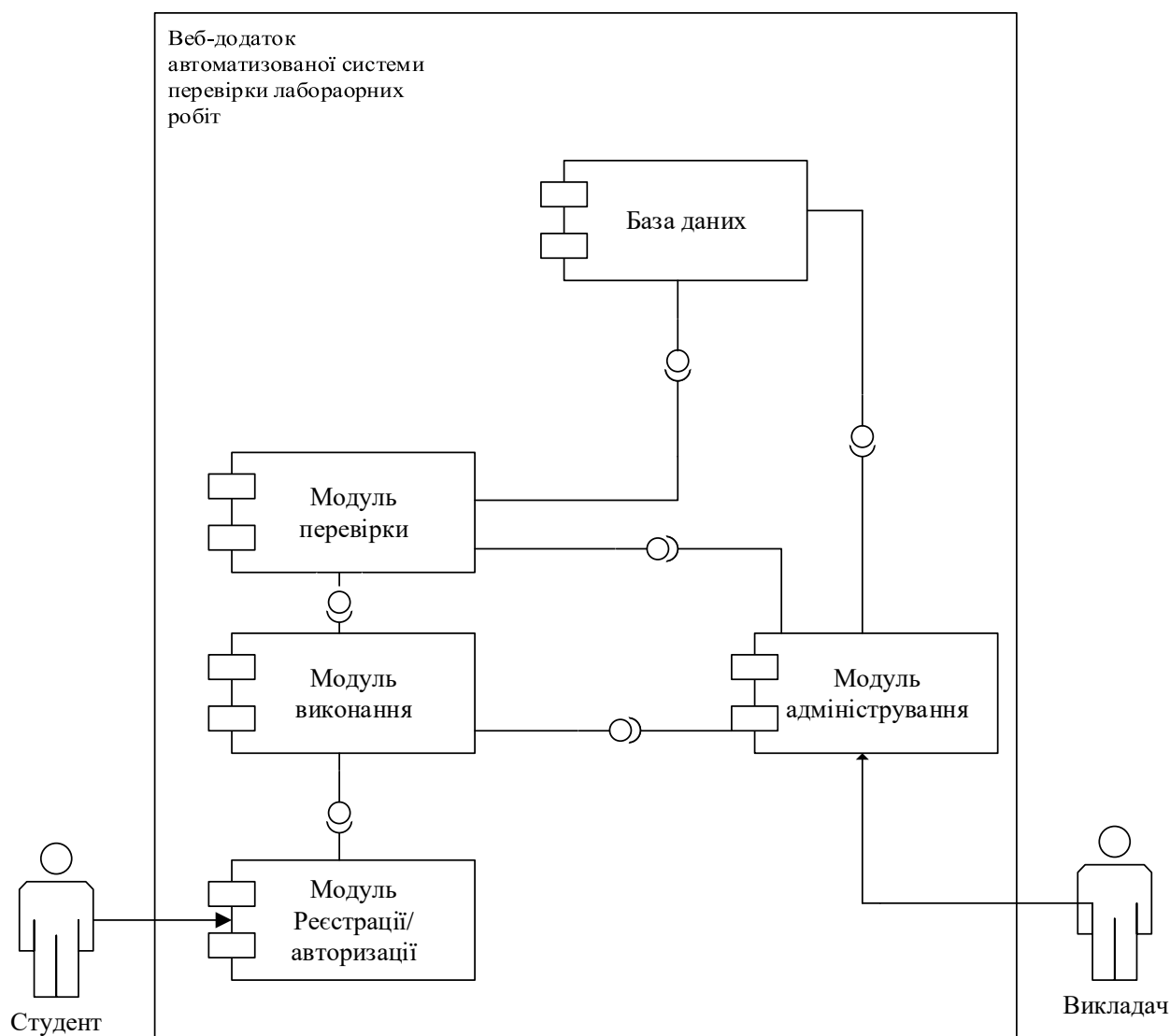


Рисунок 3.1 – Діаграма компонентів

Джерело: запропоноване автором

У цьому дипломному проєкті діаграма компонентів складається з таких елементів:

1. Модуль Реєстрації/Авторизації: Цей модуль відповідає за функції реєстрації та авторизації користувачів на сайті. Він забезпечує правильне управління обліковими записами та захист доступу до системи.

2. Модуль адміністрування: Цей модуль надає можливості адміністраторам системи керувати користувачами, правами доступу, змінювати налаштування та здійснювати інші дії, пов'язані з управлінням системою.

3. Модуль виконання: Цей модуль відповідає за виконання основної функціональності системи. Він містить логіку, необхідну для виконання завдань,

представлених на сайті, таких як обробка даних, відображення результатів тощо.

4. Модуль перевірки: Цей модуль відповідає за перевірку правильності та цілісності даних, які надходять до системи. Він забезпечує валідацію та перевірку даних на різних етапах обробки.

5. Бази даних: Це компонент, який містить фізичні або логічні бази даних, в яких зберігаються дані, необхідні для роботи системи. Він забезпечує збереження, доступ і оновлення даних, використовуючи відповідні запити та операції бази даних.

Ці компоненти взаємодіють між собою для забезпечення правильної роботи системи та надання необхідної функціональності користувачам.

З web-додатком взаємодіюють два типи користувачів: студенти та викладачі.

Викладач, виступаючи як адміністратор, взаємодіє з усіма компонентами системи за допомогою модуля адміністрування.

Студенти спочатку взаємодіють з модулем реєстрації/авторизації, і тільки після успішної авторизації вони мають доступ до модуля виконання.

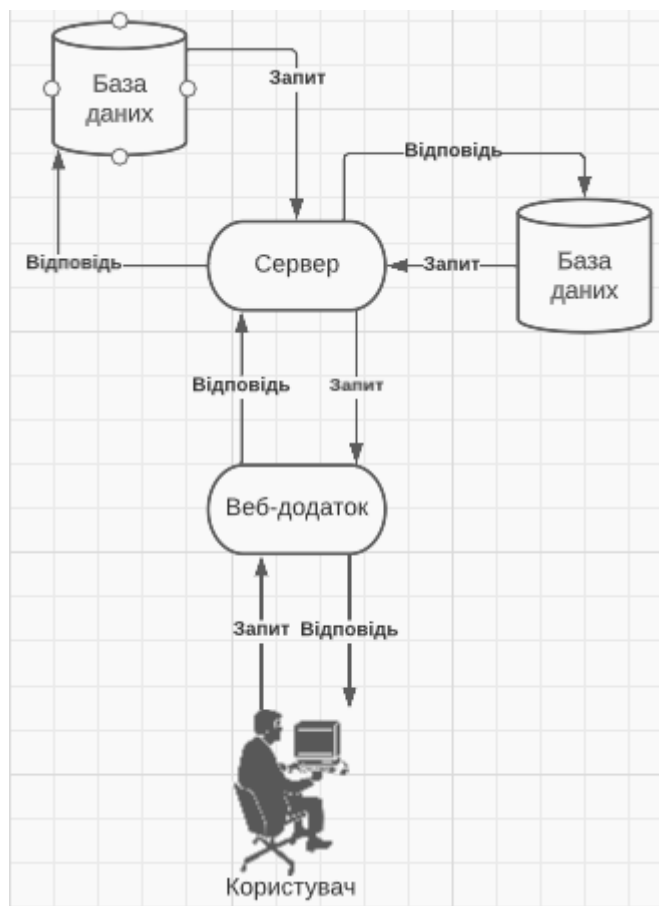


Рисунок 3.2 – Архітектура сайту

Джерело: запропоноване автором

Архітектура сайту включає наступні компоненти: користувач, веб-додаток, сервер та базу даних.

1. Користувач: Це особа або система, яка взаємодіє з веб-сайтом. Користувачі можуть виконувати різні дії, такі як перегляд сторінок, введення даних або здійснення операцій.

2. Веб-додаток: Це програмний компонент, який виконується на сервері і надає функціональність сайту. Веб-додаток обробляє запити користувачів, взаємодіє з базою даних та генерує відповіді, які відображаються на веб-сторінках.

3. Сервер: Це обчислювальний ресурс, який хостить веб-додаток і обробляє запити користувачів. Сервер забезпечує комунікацію між веб-додатком та користувачами, відправляє запити до бази даних та отримує результати.

4. База даних: Це зберігальний простір для збереження даних, які використовуються веб-додатком. База даних зберігає інформацію, таку як користувачі, деталі замовлень, контент сторінок тощо. Веб-додаток взаємодіє з базою даних для отримання, оновлення та збереження даних.

Користувачі взаємодіють з веб-додатком через браузер або інший клієнтський інтерфейс. Веб-додаток отримує запити від користувачів, обробляє їх, звертається до бази даних для отримання або збереження інформації і повертає відповідь.

3.2 Програмна реалізація

Основним завданням цього дипломного проекту було створення інструментарію, який забезпечує високу точність перевірки запитів студентів та оцінює їх роботу. Для досягнення цієї мети було прийнято рішення зберігати відповіді, отримані від студентів та викладачів, в базі даних та порівнювати їх. Це створило стандарт, за яким проводяться подальші перевірки.

Для кожного з основних архітектурних компонентів додатку було створено відповідні директорії, в яких розташовуються більш специфічні фрагменти моделі, контролера та вигляду. Файлова структура проекту наведена на рисунку 3.3.

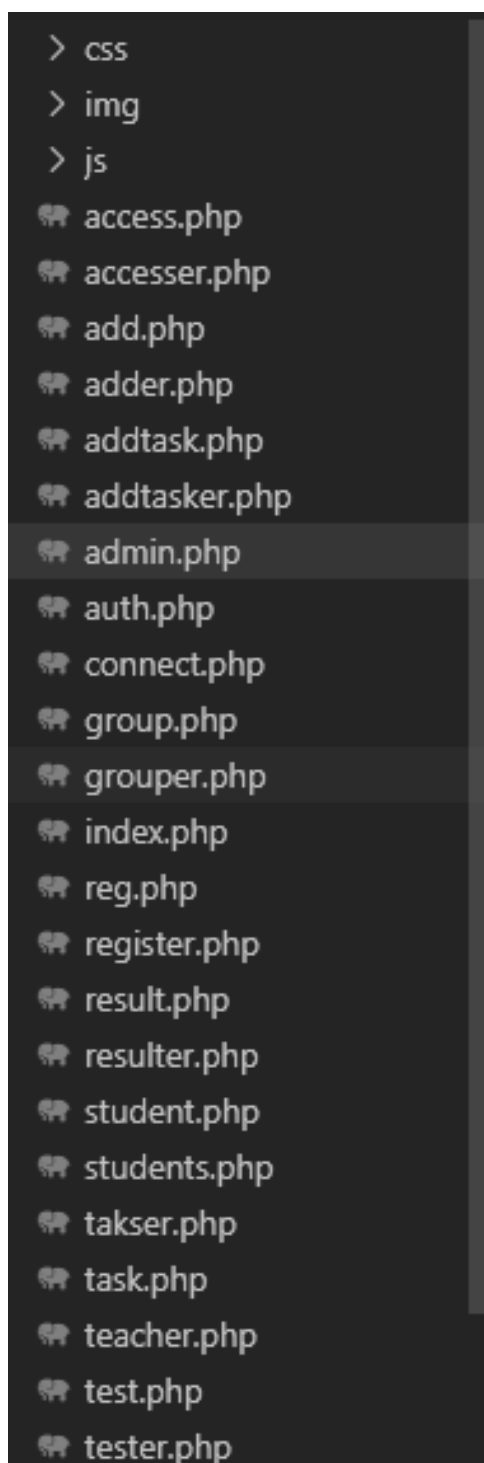


Рисунок 3.3 – Файлова структура проекту

Джерело: запропоноване автором

Для встановлення параметрів з'єднання з основною базою даних, а також для налаштування ряду параметрів навігації, був створений окремий файл - connect.php. У цьому файлі ці параметри задаються як константи (рис. 3.4).

```

connect.php > ...
1  <?php
2  $mysql = mysqli_connect('localhost', 'root', '', 'stud_lab');
3  if(!$mysql) {
4  die ('Error connect to database!');
5  }
6  ?>

```

Рисунок 3.4 Лістинг файлу connect.php

Джерело: запропоноване автором

Для реєстрації нових користувачів і збереження інформації про них у базі даних було створено файл reg.php. У цьому файлі реалізовані перевірки на валідність та унікальність доданих даних, а при успішній реєстрації відправляється запит на збереження інформації в базу даних (рис. 3.5).

```

reg.php > ...
6  $pass_conf = $_POST['pass_conf'];
7  $name = $_POST['name'];
8  $surname = $_POST['surname'];
9  $group = $_POST['group'];
10 $q = mysqli_fetch_assoc(mysqli_query($mysql, "SELECT COUNT(*) FROM `users` WHERE
11 `login`='$_login'"));
12 $check = join("", $q);
13 if (preg_match('/^([0-9a-zA-Z-_.]{5,16})$/', $login) || $login == "") {
14 if ($check > 0) {
15 $SESSION['alert'] = 'Такий логін вже існує!';
16 header('Location: register.php');
17 } elseif ($pass <> $pass_conf) {
18 $SESSION['alert'] = 'Паролі не співпадають!';
19 header('Location: register.php');
20 } elseif ($login == "" || $pass == "" || $name == "" || $surname == "") {
21 $SESSION['alert'] = 'Заповніть усі поля!';
22 header('Location: register.php');
23 } else {
24 $pass = md5($pass);
25 mysqli_query($mysql, "INSERT INTO `users` (`id_user`, `login`, `pass`, `name`,
26 `surname`, `id_group`) VALUES (NULL, '$login', '$pass', '$name', '$surname', '$group')");
27 $SESSION['msg'] = 'Ви зареєстровані!';
28 header('Location: index.php');
29 }
30 } else {
31 $SESSION['alert'] = 'Логін тільки латиницею!';
32 header('Location: register.php');
33 }

```

Рисунок 3.5 – Лістинг файлу reg.php

Джерело: запропоноване автором

Після додавання нового користувача та збереження його інформації, необхідно пройти процес авторизації, що означає перевірку його даних з даними, що зберігаються в базі даних (рис. 3.7). Якщо веб-додаток знайде відповідність,

відвідувач буде авторизований в своєму профілі та матиме можливість користуватися всіма функціями. Цей процес відповідає файлу авторизації - auth.php (рис. 3.6).

```

auth.php > ...
1  <?php
2  session_start();
3  require_once 'connect.php';
4  $login = $_POST['login'];
5  $pass = md5($_POST['pass']);
6  $check_user = mysqli_query($mysql, "SELECT * FROM `users` WHERE `login` = '$login' AND
7  `pass` = '$pass'");
8  if (mysqli_num_rows($check_user) > 0) {
9  if ($login === "admin") {
10 header('Location: admin.php');
11 } elseif ($login === "yeremenko") {
12 header('Location: teacher.php');
13 } else {
14 $user = mysqli_fetch_assoc($check_user);
15 $_SESSION['user'] = [
16 "id" => $user['id_user'],
17 "name" => $user['name'],
18 "surname" => $user['surname'],
19 "id_group" => $user['id_group']
20 ];
21 header('Location: student.php');
22 }
23 } else {
24 $_SESSION['alert'] = 'Не вірний логін або пароль!';
25 header('Location: index.php');
26 }
27 ?>

```

Рисунок 3.6 – Лістинг коду файлу auth.php

Джерело: запропоноване автором

	<u>id_user</u>	<u>name</u>	<u>surname</u>	<u>login</u>	<u>password</u>	<u>id_group</u>
☐   	1	Ivan	Ivanov	IST4	123	1
☐   	2	Oleksndr	Yeremenko	IST3	123	1
☐   	3	123	123	132	123	2

Рисунок 3.7 – Структура бази даних «users»

Джерело: запропоноване автором

Для адміністрування сайту була створена адміністраторська сторінка admin.php(рис. 3.8), яка надає доступ та управління адміністративним функціям та можливостям системи. Основна мета адміністраторської сторінки - надати

адміністратору зручний інтерфейс для керування різними аспектами веб-додатку.

```

echo"<img src = './images/star.png' width=15 height=15>";
}
echo"</td>";
}
else{
echo"<td style='max-width:330px'>".$buf."</td>";
}
}
echo"</tr>";
$p++;
}
echo "</table></div>";
//ТАБЛИЦА С завданнями -
echo"<datalist id='list_tasks'>";
for($i=0;$i<count($for_add_admin);$i++){
echo"<option>".$for_add_admin[$i]."</option>";
}echo"</datalist>";
echo"<datalist id='list_queries'>";
for($i=0;$i<count($for_add_admin1);$i++){
echo"<option>".$for_add_admin1[$i]."</option>";
}
echo"</datalist>";
echo"</div>";
echo"</div>";
}
?>
</body></html>

```

Рисунок 3.8 – Лістинг коду файла admin.php

Джерело: запропоноване автором

Для автоматизованої перевірки відповіді студента на завдання використовується контролера `resulter.php`. Цей метод отримує ідентифікатор конкретної лабораторної роботи та ідентифікатор завдання з цієї роботи (рис. 3.9).

На початку методу відбувається запит до моделі для отримання даних про завдання, таких як ідентифікатори наступного та попереднього завдань, правильна відповідь на поточне завдання та кількість спроб, що залишилися у студента. Якщо студент вперше відкриває цю сторінку, відбувається завантаження представлення з вищезазначеними даними.

Після введення студентом відповіді на завдання у відповідне поле та натискання кнопки для підтвердження, відбувається двоетапна перевірка цієї відповіді та встановлення потенційної оцінки за виконання завдання.

```

</div>
</div>
</div>
</div>
</header>
<form action="result.php" method="post">
<?
require_once 'connect.php';
$result1 = mysqli_query($mysql, "SELECT labs.lab, results.all_answer,
results.right_answer FROM `results` INNER JOIN `labs` ON results.id_lab=labs.id_lab INNER JOIN
`users` ON results.id_user=users.id_user WHERE results.id_user = '$id_user'");
?>
<div class="box">
<table class="table">
<thead>
<tr>
<th>Назва роботи</th>
<th>Кількість завдань</th>
<th>Кількість правильних відповідей</th>
</tr>
</thead>
<?
while($select_while1 = mysqli_fetch_assoc($result1)) {
?>
<tbody>
<tr>
<td><? echo $select_while1['lab']; ?></td>
<td><? echo $select_while1['all_answer']; ?></td>

```

Рисунок 3.9 - Лістинг коду файла result.php

Джерело: запропоноване автором

Було розроблено функціонал, який дозволяє експортувати дані про успішність студентів до файлу MS Excel. У цьому файлі дані про студентів зберігаються відповідно до їх групи. Повний код, який реалізує цей функціонал, наведено в Додатку Б.

Рік

Експорт оцінок в Excel

Рисунок 3.10 – Створення ехел файлу успішності студентів

Джерело: запропоноване автором

A	B	C	D
№	Initials and Surname of a Student	Labs for Unit 15 - Retrieving Data Using Set Operators	Total Score
1	Креслов Дмитро	1,533333333	1,533333
2	Петренко Катерина	1	1
3	Михайліченко Антон	1,066666667	1,066667

Рисунок 3.11 –Звіт успішності

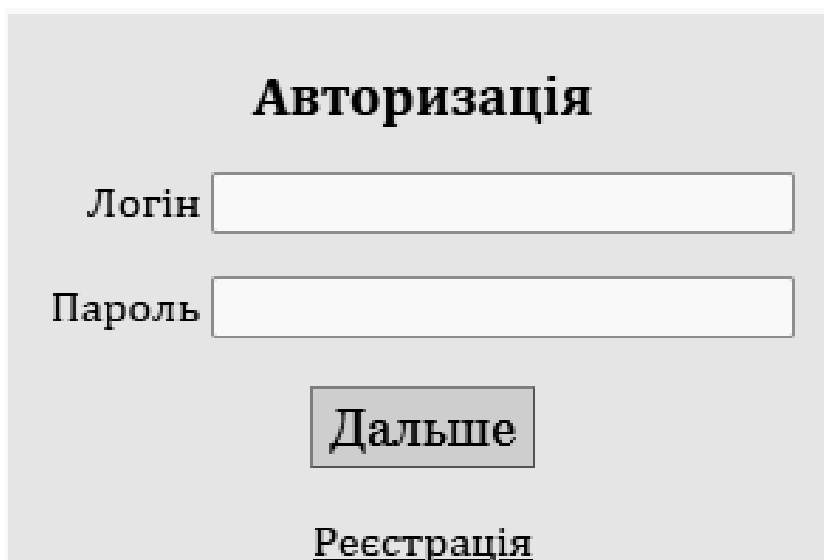
Джерело: запропоноване автором

Адміністратор має можливість управляти визначеними процесами за допомогою адміністративної панелі. Ці процеси включають змінення, видалення, редагування, додавання та встановлення терміну виконання робіт і багато іншого.

Викладач встановлює максимальну кількість балів за всю лабораторну роботу та кількість зірок за кожне завдання. Щоб визначити відсоток балів за одне завдання від загальної кількості балів, сумуються всі отримані зірки. Потім ця сума ділиться на максимальну кількість зірок для даного завдання, результат помножується на 100. Таким чином отримується відсоток балів за це завдання. Повний програмний код можна знайти в Додатку Г.

3.3 Використання інформаційної системи

Першою сторінкою, яку користувач бачить при запуску розробленого веб-застосунку, є сторінка авторизації. На цій сторінці розташована форма (див. рис. 3.12), де користувач повинен ввести свій логін і пароль, які визначалися під час реєстрації, щоб отримати доступ до системи. При авторизації також визначається роль користувача в системі: викладач або студент.



Авторизація

Логін

Пароль

Дальше

[Реєстрація](#)

Рисунок 3.12 – Форма авторизації

Джерело: запропоноване автором

Якщо користувач вводить неправильну інформацію, наприклад, логін, якого

немає в системі, або невірний пароль для введеного логіну, то після натискання кнопки "Далі" він отримує відповідне повідомлення про помилку при авторизації (див. рис. 3.13).

Користувача з такими логіном і паролем не знайдено.

Авторизація

Логін

Пароль

[Реєстрація](#)

Рисунок 3.13 - Перевірка даних під час авторизації

Джерело: запропоноване автором

Якщо користувач-студент вперше відвідує систему, він може зареєструватися, натиснувши на посилання "Реєстрація!", розташоване внизу форми авторизації. Після цього, його перенаправляють на сторінку реєстрації, де він повинен заповнити форму (див. рис. 3.14). В цій формі він повинен ввести своє ім'я, прізвище, обрати групу, визначити логін та пароль, а також підтвердити пароль, введений повторно. Якщо користувач уже зареєстрований, він може повернутися на сторінку авторизації, натиснувши на посилання "Авторизація!".

Для форми реєстрації також встановлені правила перевірки введених даних. Наприклад, ім'я та прізвище можуть складатися тільки з літер латинського алфавіту та кирилиці, а логін та пароль можуть містити лише літери латинського алфавіту. Крім того, їх довжина повинна відповідати визначеному значенню в системі. Якщо користувач натискає кнопку "Далі" і не відповідає встановленим обмеженням, відображається відповідне повідомлення про помилку (див. рис. 3.15).

Регістрація

Ім'я

Прізвище

Група

Логін

Пароль

Повторіть пароль

[Авторизація](#)

Рисунок 3.14 – Форма реєстрації
Джерело: запропоноване автором

Виникла помилка

Регістрація

Ім'я

Прізвище

Група

Логін

Пароль

Повторіть пароль

[Авторизація](#)

Рисунок 3.15 - Перевірка даних під час реєстрації
Джерело: запропоноване автором

Після успішної реєстрації користувач-студент не одразу отримує доступ до системи. Йому потрібно зачекати на підтвердження від викладача або адміністратора. Якщо він спробує авторизуватися без наданого доступу, йому буде показано відповідне повідомлення(рис.4.16).

Запит прийнято, очікуйте схвалення реєстрації від адміністратора

Реєстрація

Ім'я

Прізвище

Група

Логін

Пароль

Повторіть пароль

[Авторизація](#)

Рисунок 3.16 - Перевірка наявності прав на авторизацію

Джерело: запропоноване автором

Після успішної авторизації, студент перенаправляється на головну сторінку сайту, де він зустрічає список доступних лабораторних робіт і може обрати ту, яку він бажає виконати (рис. 3.17).

Головна Мої спроби Мої результати Структура бази Вийти

Оберіть роботу зі списку:

- Лабораторна робота №16 - Вибірка даних з використанням агрегатних функцій
- Labs for Unit 15 - Retrieving Data Using Set Operators
- Лабораторна робота до теми 14 - Вибірка даних з використанням підзапитів
- Лабораторна робота до теми 12 - Вибірка даних з більш ніж однієї таблиці
- Лабораторна робота до теми 11 - Вибірка даних з однієї таблиці
- Labs for Unit 17 - Hierarchical queries
- Labs for Unit 18 - Retrieving Data Using Cartesian Product

Рисунок 3.17 – Демонстрація головної сторінки студента

Джерело: запропоноване автором

Після вибору відповідної лабораторної роботи, студент бачить перелік завдань,

які необхідно виконати, разом з кількістю спроб для кожного завдання. Для вибору певного завдання, студент повинен клікнути на нього і натиснути кнопку "Ок" (рис. 3.18).

Головна Мої спроби Мої результати Структура бази Вийти

Лабораторна робота до теми 14 - Вибірка даних з використанням підзапитів ok

Оберіть завдання зі списку:

- ☐ 14.1* Display the names of all subordinates King. Відобразіть імена всіх підлеглих King. / Залишилось (2) спроби
- ☐ 14.2* Display names of employees in the same department as Smith. Відобразити імена працівників відділу, в якому працює Smith. / Залишилось (2) спроби
- ☐ 14.3** Display names of employees hired after Smith in his department Відобразіть імена працівників, найнятих після Smith у його відділі / Залишилось (3) спроби
- ☐ 14.4*** Display the names of managers whom all employees receive more than \$ 2500 (use IN, ALL, ANY) Відобразіть прізвища начальників, всі працівники яких отримують понад 2500 доларів (використовуй IN, ALL, ANY) / Залишилось (2) спроби
- ☐ 14.5** Display the names of employees hired after all employees of department No. 30 (use IN, ALL, ANY) Виведіть імена працівників, найнятих після всіх працівників відділу N30 (використовуй IN, ALL, ANY) / Залишилось (2) спроби
- ☐ 14.6* Display the name, department number and salaries of employees who work in departments where some employees are paid the comm (use IN, ALL, ANY) Відобразіть прізвище, номер підрозділу та заробітну плату працівників, які працюють у підрозділах, де деяким працівникам виплачується премія (використовуй IN, ALL, ANY) / Залишилось (2) спроби
- ☐ 14.7** Display name, post, and salary of employees whose salary is greater than the salary of any analysts. (use IN, ALL, ANY) Відобразіть ім'я, посаду та заробітну плату співробітників, які отримують більше будь-якого аналітика (використовуй IN, ALL, ANY) / Залишилось (2) спроби
- ☐ 14.8* Display the name and salary for all employees from Dallas (use IN, ALL, ANY) Відобразіть імена та зарплатню всіх співробітників Даласу (використовуй IN, ALL, ANY) / Залишилось (2) спроби
- ☐ 14.9* Display the number and name of the employee, number and name his manager using subquery in the SELECT clause Відобразіть номер та прізвище працівника, номер та прізвище його керівника, використовуючи підзапит в розділі SELECT / Залишилось (2) спроби

Рисунок 3.18 - Демонстрація головної сторінки студента

Джерело: запропоноване автором

Коли студент обрав потрібне завдання, йому потрібно скласти запит відповідно до вимог завдання і натиснути кнопку "Відправити відповідь". У випадку, якщо випадково студент обрав не те завдання, він може обрати інше завдання, клацнувши на кнопку "Обрати інше завдання" (рис. 3.19).

Головна Мої спроби Мої результати Структура бази Вийти

Лабораторна робота до теми 14 - Вибірка даних з використанням підзапитів ok

14.12.** Display the names, department number and the salary of employees from the department №10 whom paid above \$1500 using subquery in the FROM clause - Виведіть імена, номер відділу та заробітну плату працівників відділу №10, які отримують понад \$1500, використовуючи підзапит в розділі FROM

Обрати інше завдання

Відправити відповідь

Не забудь про ";" в кінці запиту!

Рисунок 3.19 - Демонстрація форми виконання завдання

Джерело: запропоноване автором

Якщо студент надішле невірний запит, він отримає повідомлення про помилку. В такому випадку, якщо студент має ще спроби на виконання завдання, він може спробувати ще раз, надіславши запит повторно/

Студент може переглянути свої спроби виконаних завдань на сторінці «Мої спроби»(рис.3.20)

Головна Мої спроби Мої результати Структура бази Вийти					
Дата	Робота	Текст завдання	Ваша відповідь	Бал	Коментар
2023-06-05 22:04:57	Лабораторна робота до теми 14 - Вибірка даних з використанням підзапитів	14.12.** Display the names, department number and the salary of employees from the department №10 whom paid above \$1500 using subquery in the FROM clause - Виведіть імена, номер відділу та заробітну плату працівників відділу №10, які отримують понад \$1500, використовуючи підзапит в розділі FROM	123		ИТОГИ АВТОМАТИЧЕСКОЙ ПРОВЕРКИ: Возникла ошибка при проверке. Запрос будет проверен в ручном режиме.

.Рисунок 3.20 – Сторінка «Мої спроби»

Джерело: запропоноване автором

Відкриваючи сторінку «Мої результати» студент бачить свої досягнення щодо певної роботи і відстежувати свій прогрес.(рис. 3.21)

Головна Мої спроби Мої результати Структура бази Вийти			
Робота		Максимальний бал	Бал
Лабораторна робота до теми 14 - Вибірка даних з використанням підзапитів		4	0.41379310344827586
Labs for Unit 15 - Retrieving Data Using Set Operators		2	0.26666666666666666

Рисунок 3.21 – Сторінка «Мої результати»

Джерело: запропоноване автором

Після успішної авторизації викладач перенаправляється на головну сторінку, де розміщена адміністративна панель та таблиці з інформацією лабораторні роботи. Для створення нової роботи, викладач заповнює форму інформацією про лабораторну роботу, таку як дата початку та дата закінчення, і натискає кнопку «Додати».(рис. 3.22)

Після того як викладач створив нову лабораторну роботу, її вміст додається до таблиці «Роботи».(рис.3.23)

Для створення нового завдання для обраної роботи необхідно виконати наступні кроки: ввести номер роботи, вказати назву завдання в полі «Текст завдання» і ввести еталонний запит без символу «;» в полі «Правильна відповідь». Крім того, потрібно вказати кількість зірок, яку можна отримати за виконання завдання, а також

максимальну кількість спроб на його виконання (рис. 3.24).

+ Роботу

Дата початку

Дата кінця

Заголовок (Коротко)

Максимальний бал

Рисунок 3.22 – Додавання роботи

Джерело: запропоноване автором

Роботи:				
Номер роботи	Дата початку	Дата кінця	Заголовок	MAX бал
16	2023-04-28	2021-05-04	Лабораторна робота №16 - Вибірка даних з використанням агрегатних функцій	4
15	2023-01-21	2023-01-25	Labs for Unit 15 - Retrieving Data Using Set Operators	2
14	2022-12-13	2022-12-21	Лабораторна робота до теми 14 - Вибірка даних з використанням підзапитів	4
12	2022-11-21	2022-12-07	Лабораторна робота до теми 12 - Вибірка даних з більш ніж однієї таблиці	4
11	2021-11-16	2021-11-30	Лабораторна робота до теми 11 - Вибірка даних з однієї таблиці	4
23	2022-02-11	2022-02-24	Labs for Unit 17 - Hierarchical queries	4
24	2022-02-25	2022-03-10	Labs for Unit 18 - Retrieving Data Using Cartesian Product	4

Рисунок 3.23 – Таблиця робіт

Джерело: запропоноване автором

The image shows a web form for adding tasks. It has a title '+ Завдання' at the top. Below the title are two large text input areas. The first is labeled 'Текст завдання' and the second is labeled 'Правильна відповідь'. Below these two areas are three input fields for 'Робота', 'Зірок', and 'Спроб'. At the bottom of the form is a button labeled 'Додати'.

Рисунок 3.24 –Процес додавання завданнь

Джерело: запропоноване автором

Після того як викладач записав всю необхідну інформацію про завдання, вона переноситься в таблицю «Завдання» (рис.3.25).

Для зміни зміста лабораторної роботи, необхідно спочатку вибрати номер даної роботи та записати її у відповідне поле після цього натиснути кнопку «Змінити», після змін потрібно натиснути на кнопку «Зберегти зміни».(рис.3.26)

Якщо потрібно змінити зміст завдання, потрібно ввести номер цього завдання у відповідне поле і натиснути на "Змінити". Тоді відкриється режим редагування, де можна внести необхідні зміни до завдання. Після завершення редагування треба натиснути кнопку "Зберегти зміни" для збереження внесених змін. (рис. 3.27)

Завдання:				
Номер запиту	Текст завдання	Номер роботи	Складність	МАХ спроб
29	12.1* Відобразити всю інформацію про працівників: імена працівників та місця розташування їх підрозділів	12	(1) ★	2
15	11.14 Виведіть ім'я працівників, у яких премія (comm) перевищує 20% від зарплати	11	(2) ★★	2
14	11.13 Виведіть ім'я та зарплату тих працівників, зарплата яких перевищує 1200 доларів.	11	(2) ★★	2
12	11.11 Змініть запит 11.10, щоб додати четвертий стовпець для відображення розміру збільшення зарплати. Надайте стовпцю псевдонім "Increased to"	11	(1) ★	2
11	11.10 Виведіть номер працівника, його поточну зарплату та підвищену зарплату на 20%. Назвіть останню колонку New Salary	11	(1) ★	2
10	11.9 Виведіть ім'я всіх працівників та розмір їх внеску до пенсійного фонду. Розмір внесків становить 6% від зарплати. Надайте стовпцю псевдонім "contribution"	11	(1) ★	2

Рисунок 3.25 – Вміст таблиці завдання

Джерело: запропоноване автором

Змінити роботу

Номер

Дата початку

Дата кінця

Заголовок (Коротко)

Лабораторна робота №16 - Вибірка даних з використанням агрегатних функцій

Максимальний бал

Рисунок 3.26 -Редагування роботи

Джерело: запропоноване автором

Змінити завдання

Номер

Текст завдання

11.13 Виведіть ім'я та зарплату тих працівників, зарплата яких перевищує 1200 доларів.

Правильна відповідь

```
SELECT ename,
sal FROM emp WHERE sal > 1200
```

Робота

Зірок

Спроб

Рисунок 3.27 – Редагування завдання

Джерело: запропоноване автором

Для того, щоб видалити роботу, необхідно вказати її номер у відповідному полі і натиснути кнопку "Видалити" (рис. 3.28). Після цього робота буде вилучена з системи і не буде доступна для подальшого виконання або редагування.

Видалити роботу

Номер

Рисунок 3.28 – Видалення роботи

Джерело: запропоноване автором

Для того, щоб видалити завдання, потрібно вказати його номер у відповідному полі і натиснути кнопку "Видалити" (рис. 3.29). Після цього завдання буде повністю вилучено з системи і не буде доступно для подальшого виконання чи редагування.

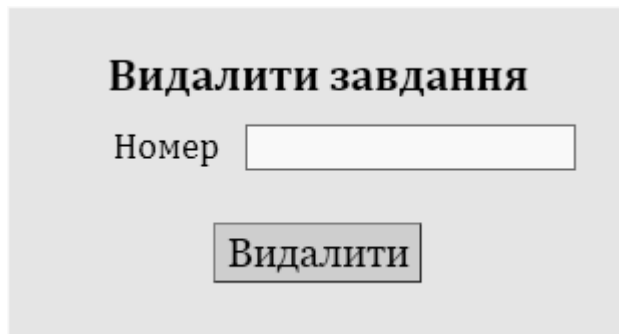


Рисунок 3.29 – Видалення завдання

Джерело: запропоноване автором

Відкриваючи сторінку "Нові користувачі" викладач може побачити запит на підтвердження реєстрації студента, (рис. 3.30). Натиснувши на «ОК» він підтверджує правильність введених даних студента та надає йому повний доступ до системи. Після цього, студент зможе авторизуватися та виконувати доступні завдання.

Головна	Нові користувачі		Нові відповіді		Перевірені відповіді		Правильні відповіді		Структура бази	Вийти
	id_user	name	surname	login	password	processed	id_group	Група	OK	
	126	12333	123123	123123	1	0	7	ІТ-92	ok	
	127	Петро	Петрович	Petro	1	0	8	ІТ-92-0	ok	

Рисунок 3.30 – Сторінка «Нові користувачі»

Джерело: запропоноване автором

На сторінці "Нові відповіді" викладач має доступ до переліку відповідей студентів, які можуть бути оцінені вручну. Викладач може ввести оцінку відповіді у відповідну комірку. Якщо викладач захоче то він зможе залишити коментар до роботи студента. (рис. 3.31) Це дозволяє викладачу оцінити роботу студента та надати йому зворотний зв'язок щодо його відповіді.

Головна													Нові користувачі	Нові відповіді	Перевірені відповіді	Правильні відповіді	Структура бази	Вийти		
	ID	Студент	Група	Робота	Текст завдання	MAX оцінка	Рішення	Відповідь	Оцінка	Коментар	OK									
15391	114	Rodion Chabanenko	IT-92	Labs for Unit 18 - Retrieving Data Using Cartesian Product	18.12** Display the department's list in which at least one manager works - Відобразити список назв підрозділів, де працює хоч один менеджер. Використовувати тільки Cross Join	2	Select DISTINCT dept.deptno, dept.dname, deptloc from emp, dept Where emp.deptno=deptdeptno && emp.job = 'Manager'	SELECT DISTINCT X.DEPTNO FROM EMP X, EMP Y WHERE X.EMPNO = Y.MGR AND X.DEPTNO = Y.DEPTNO; SELECT DISTINCT X.DEPTNO, X.DNAME, X.LOC FROM DEPT X, EMP Y, EMP Z WHERE X.DEPTNO = Y.DEPTNO AND Y.EMPNO = Z.MGR;	От 0 до 2	ИТОГИ АВТОМАТИЧЕСКОЙ ПРОВЕРКИ: Возникла ошибка при проверке. Запрос будет проверен в ручном режиме.	ok									

Рисунок 3.31 – Сторінка «Нові відповіді»

Джерело: запропоноване автором

Всі роботи які були перевірені викладачем зберігаються на сторінці «Перевірені відповіді».(рис. 3.32)

Головна		Нові користувачі		Нові відповіді		Перевірені відповіді		Правильні відповіді		Структура бази		Вийти	
Рік 2023				Експорт оцінок в Excel									
Робота		Текст завдання		MAX оцінка		Рішення		Відповідь		Оцінка			
Labs for Unit 18 - Retrieving Data Using Cartesian Product		18.15*** Modify request 18.14 so that pairs are combined employees working in the same city, and not in the same department. - Змініть запит 18.14, щоб пари були об'єднані працівниками, які працюють в одному місті, а не в одному відділі. Використовувати		3		SELECT e1.ename, e2.ename FROM emp e1, emp e2, dept d1, dept d2 WHERE e1.deptno != e2.deptno AND e1.ename != e2.ename AND d1.deptno = e1.deptno AND d2.deptno = e2.deptno GROUP BY e1.ename;		SELECT e1.ename, e2.ename, d1.loc FROM 'emp' e1 CROSS JOIN 'dept' d1 CROSS JOIN 'emp' e2 CROSS JOIN 'dept' d2 WHERE e1.deptno = d1.deptno AND e2.deptno = d2.deptno AND d1.loc = d2.loc AND e1.empno <> e2.empno AND e1.empno > e2.empno ORDER BY d1.loc ASC;		От 0 до 3 3			

Рисунок 3.32 – Сторінка «Правильні відповіді»

Джерело: запропоноване автором

Крім того, викладач має можливість експортувати оцінки до файлу Excel шляхом натискання на "Експорт оцінок в Excel". Це дозволяє викладачу отримати дані відповідно до обраного року в форматі .xls. Файл Excel міститиме всю необхідну інформацію про оцінки студентів

ВИСНОВКИ

У процесі розробки дипломного проекту було проведено аналітичний огляд публікацій щодо тематики предметної області. Розроблено концепцію програмного додатку. Розроблено базу даних програмного додатку. Розроблено веб-додаток для супроводу і перевірки практичних робіт із вибірки даних:

Для досягнення поставленої мети було проведено планування робіт проекту, що включало ідентифікацію ідеї проекту, деталізацію мети за методологією SMART, планування структури робіт у форматі WBS, побудову діаграми Ганта та визначення ризиків проекту (див. Додаток Б).

Створено окремі моделі для кожної сутності бази даних системи, контролери та вигляди для кожної функціональної сторінки системи. Реалізовано модуль автоматизованої перевірки відповідей надісланих студентами, який передбачає перевірку правильності синтаксису та семантики відповіді (SQL запиту), та відображення підказок за результатами перевірки, які допомагають визначити причину помилки.

Впроваджено функціонал для роботи з інформацією про користувачів. Покращено ефективність роботи з користувацьким інтерфейсом: спрощено навігацію між завданнями однієї роботи, до ряду таблиць додано фільтри за встановленими атрибутами та можливість сортування записів, виправлено проблему з перевантаженням елементами інтерфейсу окремих сторінок системи.

Розроблена панель адміністратора, яка дозволяє викладачу просто та зручно завантажувати нові завдання. Створено діаграми бізнес-процесів в нотації Integration Definition for Function Modeling, (UML)діаграми варіантів використання, моделі даних, архітектура веб-додатку.

Відтестовано веб-додаток для супроводу і перевірки практичних робіт із вибірки даних. Використання даного веб-додатку дозволяє зменшити часові витрати на коректну перевірку запитів, створених студентом, до бази даних викладачем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Assessment of Practical Skills in Online Learning Environments. URL: <https://journals.sagepub.com/doi/pdf/10.2190/gymq-78fa-wmtx-j06c/> (дата звернення: 02.03.2023)
2. Automated Assessment Systems for Practical Assignments in Computer Science Education. URL: <https://ieeexplore.ieee.org/abstract/document/8819909> // (дата звернення: 02.03.2023)
3. Bringing MySQL to the web. URL: <https://www.php.net/> (дата звернення: 04.05.2023)
4. Codecademy. URL: <https://www.codecademy.com/learn/> (дата звернення: 02.03.2023)
5. CodeSignal. URL: <https://codesignal.com/> (дата звернення: 02.03.2023)
6. Intelligent Tutoring Systems for Practical Exercises in Data Analysis. URL: <https://link.springer.com/article/10.1007/s40593-014-0014-z/> (дата звернення: 02.03.2023)
7. Локальне середовище - Open Server Panel. URL: <https://ospanel.io/> (дата звернення: 23.03.2023)
8. Методологія IDEF0. URL: https://stud.com.ua/87184/ekonomika/metodologiya_idef0/ (Дата звернення: 06.04.2023)
9. PHP . URL: <https://www.php.net/> (дата звернення: 04.05.2023)
10. PHP MySQL Database. URL: <https://bitly.su/jiKXII/> (дата звернення: 30.05.2023)
11. Редактор коду Visual Studio Code. URL: <https://code.visualstudio.com/docs> (дата звернення: 23.03.2023)
12. Розробка UML-діаграми варіантів використання. URL: <https://studfile.net/preview/5200239/page:6/> (Дата звернення: 30.05.2023)
13. Інтерактивний онлайн тренажер мови SQL SQLCourse. URL: <https://www.sqlcourse.com/> (дата звернення: 02.03.2023)
14. Інтерактивний онлайн тренажер мови SQL SQLZoo. URL: https://sqlzoo.net/wiki/SQL_Tutorial/ (дата звернення: 02.03.2023)

15. Татро, К., МакІнтайр, П. Programming PHP: Creating Dynamic Web Pages. 4-те видання. O'Reilly Media, 2020. С. 544.
16. ЗАКОН УКРАЇНИ про захист інформації в інформаційно-комунікаційних системах. URL: <https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80#Text/> (дата звернення: 30.05.2023)
17. Що таке CSS?. URL: https://css.in.ua/article/shcho-take-html_10/ (дата звернення: 04.05.2023)
18. Єременко О. В. Використання інформаційних Веб-тренажерів в навчальному процесі / Технології XXI сторіччя: Збірник тез за матеріалами 28-ої міжнародної науково-практичної конференції (23-25 листопада 2022 р.). Ч.2. – Суми: СНАУ, 2022 - с.176-177.
19. Єременко О. В. Розробка інформаційної системи перевірки практичних робіт із вибірки даних: Шляхи і технології. / Матеріали науково-практичної конференції викладачів, аспірантів та студентів Сумського НАУ (25-28 квітня 2023 р.). – Суми, 2023. – с.351.

ДОДАТКИ

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

**На розробку інформаційної системи перевірки практичних робіт із вибірки
даних**

Погоджено:

Кандидат технічних наук, доцент кафедри
кібернетики та інформатики

Студент групи ІСТ 2101 с.т.

Єременко Олександр Володимирович

1. Призначення й мета створення web-додатку

1.1 Призначення web-додатку

Web-додаток призначений для підвищення ефективності роботи викладачів університету, для скорочення часу перевірки лабораторних робіт студентів за рахунок автоматизації деяких етапів перевірки.

1.2 Мета створення web-додатку

Мета полягає у розробленні web-додатку автоматизованої перевірки лабораторних робіт з написання запитів до БД мовою SQL.

1.3 Цільова аудиторія

У цільовій аудиторії web-додатку можна виділити такі групи:

1. Викладач
2. Студенти

2 Вимоги до web-додатку

2.1 Вимоги до web-додатку в цілому

2.1.1 Вимоги до структури й функціонування web-додатку

Web-додаток повинен бути реалізована у вигляді сайту, доступного в мережі Інтернет. Сайт повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

2.1.2 Вимоги до персоналу

Від персоналу не має вимагатися особливих технічних навичок для підтримки й експлуатації web-додатку, окрім загальних навичок роботи з персональним

комп'ютером і стандартним веб-браузером (наприклад, MS Internet Explorer, Google Chrome).

2.1.3 Вимоги до збереженні web-додатку

Уся інформація надана у web-додатку буде зберігатися у базі даних реалізованій засобами системи управління базами даних MySQL.

2.1.4 Вимоги до розмежування доступу

Розроблюваний web-додаток має бути загальнодоступним.

Відповідно до прав доступу до інформації у web-додатку, усіх користувачів можна поділити на відвідувачів та адміністратора.

Відвідувачі можуть переглядати усі сторінки web-додатку, ознайомитись з видами задач.

Адміністратор може редагувати зовнішній вигляд та наповнення web-додатку.

2.2 Структура web-додатку

2.2.1 Загальна інформація про структуру web-додатку

Сайт повинен складатися з наступних розділів:

Для викладача:

- Головна – розміщена адміністративна панель.
- Нові відповіді – містить інформацію про роботи, які надсилають студенти.
- Перевірені відповіді – містить інформацію про роботи студентів і бали за них.
- Правильні відповіді – містить інформацію з еталонними відповідями.

Для студентів:

- Головна – містить загальну інформацію з матеріалами лабораторних робіт.
- Мої спроби – містить повну інформацію про всі спроби виконання запиту до бази даних.
- Мої результати – містить інформацію з оцінками за конкретну роботу

2.2.2 Навігація

Відповідно до бажаного замовником дизайну web-додатку, для навігації, у шапці буде створена система контент меню. Меню необхідне для швидкого переміщення користувача по усім доступним сторінкам. Меню буде відображатися на всіх сторінках, щоб відвідувач міг в будь-який момент часу перейти на будь-яку сторінку web-додатку.

2.2.3 Наповнення web-додатку (контент)

Сторінки всіх розділів сайту мають бути автоматично створені на основі даних з бази даних на сервері. Редагування вмісту розділів повинно бути можливим через адміністраторський web-інтерфейс, який дозволяє редагувати інформаційний вміст сторінок сайту без необхідності програмування. Наповнення сторінок інформацією повинне бути здійснюватися за допомогою шаблонів сайту.

2.2.4 Дизайн та структура додатку

Web-додаток має сучасний та професійний стиль, з лаконічним і зручним дизайном. Оформлення не має негативного впливу на інформативність: хоча графіка на сайті є обмеженою, він пропонує зручну навігацію для користувачів та зацікавлює їх для повторних відвідувань. Розташування елементів на головній сторінці сайту наглядно показано на рисунку А.1-А.2.

Головна		Нові відповіді	Перевірені відповіді	Правильні відповіді	Структура бази даних	Вийти
Роботи						
Форма додавання роботи	Форма додавання завдання	Номер роботи	Дата початку	Дата кінця	Заголовок	MAX бал
Форма редагування роботи	Форма редагування завдання	Завдання				
Форма видалення роботи	Форма видалення завдання	Номер запиту	Текст завдання	Номер роботи	Складність	MAX спроб

Рисунок А.1 – Головна сторінка адміністратора

Головна	Мії спроби	Мії результати	Вихід
▽ Лабораторна робота №1 ОК Текст завдання Оверати інше завдання Відправити відповідь			

Рисунок А.2 – Головна сторінка студента

2.2.5 Система навігації (карта web-додатку)

Взаємозв'язок між розділами й підрозділами сайту (карта сайту) представлено на рис. А.3-А.4.



Рисунок А.3– Карта сайту для адміністратора

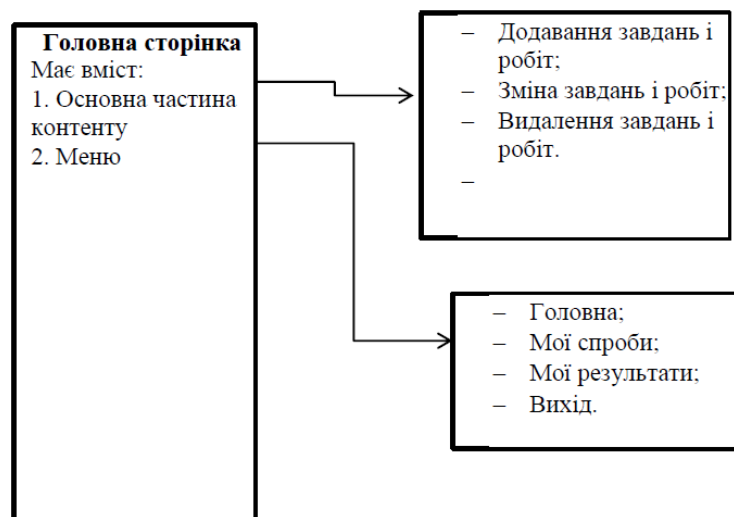


Рисунок А.2– Карта сайту для студента

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Система керування контентом (адміністративна частина) повинна надавати можливість додавання, редагування й видалення вмісту сторінок. Також повинна бути передбачена можливість додавання інформації без відображення на сайті для студентів.

2.3.2 Функціональні вимоги

На головній сторінці будуть представлені такі елементи:

- Назва та номер лабораторної роботи;
- Завдання;
- Час на виконання;
- Кількість спроб на виконання.

2.3.3 Системні вимоги

Системні вимоги відповідно до потреб користувачів:

- наявність реєстрації та авторизації відвідувачів.
- створення, редагування, видалення груп студентів;
- створення, редагування, видалення практичних робіт;
- перегляд результатів;
- перегляд, редагування, видалення користувачів;
- автоматична перевірка відповідей студента та формування списку з результатами.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація web-додатку відбувається з використанням:

- PHP 7.2
- MySQL 8.0
- CSS3
- HTML5

2.4.2 Вимоги до лінгвістичного забезпечення

Web-додаток має бути виконаний українською мовою.

2.4.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

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

3 Склад і зміст робіт зі створення web-додатку

Докладний опис етапів роботи зі створення web-додатку наведено в таблиці А.1.

Таблиця А.1 – Етапи створення web-додатку

№	Склад і зміст робіт	Строк розробки
1	Створення шаблону web-додатку	2 дні
2	Створення верстки сторінок web-додатку	14 днів
3	Розробка сторінок	10 днів
5	Розробка модулю з статистикою	7 днів
6	Розробка бази даних	7 днів
7	Додавання записів до бази даних	3 днів
8	Beta-тестування	4 днів
10	Розміщення на хостингу	1 день
11	Перевірка працездатності	2 дні
13	Завершення роботи	1 день
	Загальна тривалість робіт	51 днів

ДОДАТОК Б

ПЛАНУВАННЯ РОБІТ

Внаслідок пандемії та політичної ситуації в країні за останні роки, всі аспекти людської діяльності змушені були швидко адаптуватися до нових умов та розвиватися у незвичному форматі. Це також стосується освітнього процесу, який повністю перейшов до онлайн-формату.

Мета проекту полягає у створенні програмного продукту, який сприятиме автоматизації роботи викладача та заощадженню часу й зусиль. Досягнення цієї мети є реальним завдяки отриманим знанням на попередніх курсах з WEB, які допоможуть виконати поставлені завдання. Проект буде завершено вчасно, що підтверджується календарним планом проекту.

Деталізація мети методом SMART. Для того, щоб проект був успішним і конкурентоспроможним, необхідно на концептуальному етапі правильно визначити його цілі за допомогою підходу SMART (табл. Б.1). Його суть уточнюється в аббревіатурі назви: Specific, Measurable, Achievable, Relevant, Time-framed.

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Розробити модель веб додатку автоматичної перевірки лабораторних робіт
Measurable (вимірювана)	Створити якісний веб-додаток.
Achievable (досяжна)	Створити веб-додаток, використовуючи ресурси з застосуванням комп'ютерних технологій.
Relevant (реалістична)	У наявності є всі необхідні технічні та програмні засоби. Розробники достатньо кваліфіковані для виконання поставлених задач.
Time-framed (обмежена у часі)	Робота створюється з обмеженням у часі на основі сформованого календарного плану та матриці відповідальності.

Планування змісту робіт. Розподіл роботи на менші завдання – це зручний спосіб підвищити продуктивність розробки, зробивши роботу більш доступною та керованою. Для проектів робоча ієрархія (WBS) є інструментом, який використовує

цей підхід і є одним із найважливіших документів для управління проектом, командної роботи, що забезпечує комплексну оцінку термінів і контроль плану роботи. Елементами декомпозиції можуть бути дані, продукти, послуги.

Проектна продукція знаходиться на першому поверсі. Другий рівень декомпозиції – це основні дії та види діяльності, що забезпечують досягнення цілей проекту. Розбиття шматків виконується до тих пір, поки вони не стануть основними. Основна робота – це дія, яка завдяки своєму виконанню має чітко визначений і однозначний результат, для якого призначається особа, може бути розраховано робочий час та витрати праці. На малюнку Б.1 показано WBS, використаний для розробки web-додатку.

Планування структури виконавців. Після того, як процес розбитий, необхідно розробити організаційну структуру підрядника або OBS, яка визначається як графічна структура відповідальної особи або учасників виконання робіт, залучених до реалізації проекту. Як відповідальна особа, працівник несе відповідальність за основну роботу та організацію роботи, передбачену WBS. Кожну найважливішу роботу слід розглядати як окремий проект.

Планування структури організації. Наступним кроком після розбиття процесу є розробка організаційної структури виконавця або OBS, яка визначається як графічна структура відповідального або тих, хто бере участь у розробці проекту. Відповідальна особа – це працівник, який відповідає за організацію та виконання основної роботи, зазначеної в WBS. На рисунку Б.2 показана організаційна структура плану проекту. Перелік підрядників, залучених до проекту, наведено в таблиці Б.2.

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Єременко О.В.	Виконує front-end та back-end розробку.
Проектувальник	Єременко О.В.	Розроблює структуру web-додатку та виконує проектувальні роботи.

Продовження таблиці Б.2

Тестувальники	Єременко О.В.	Відповідальні за тестування функціоналу та дизайну web-додатку.
Керівник проекту	Агаджанова С.В.	Делегує розробкою та складає завдання на розробку проекту.

Планування структури організації, для впровадження готового проекту (OBS). Після побудови WBS розробимо організаційну структуру виконавців OBS. Організаційна структура проекту стосується тільки внутрішньої організаційної структури проекту і не стосується відносин проектних груп чи учасників з батьківськими організаціями. Діаграма OBS зображена на рис. Б.2. Список виконавців, що функціонують в проекті знаходиться в табл. Б.2.

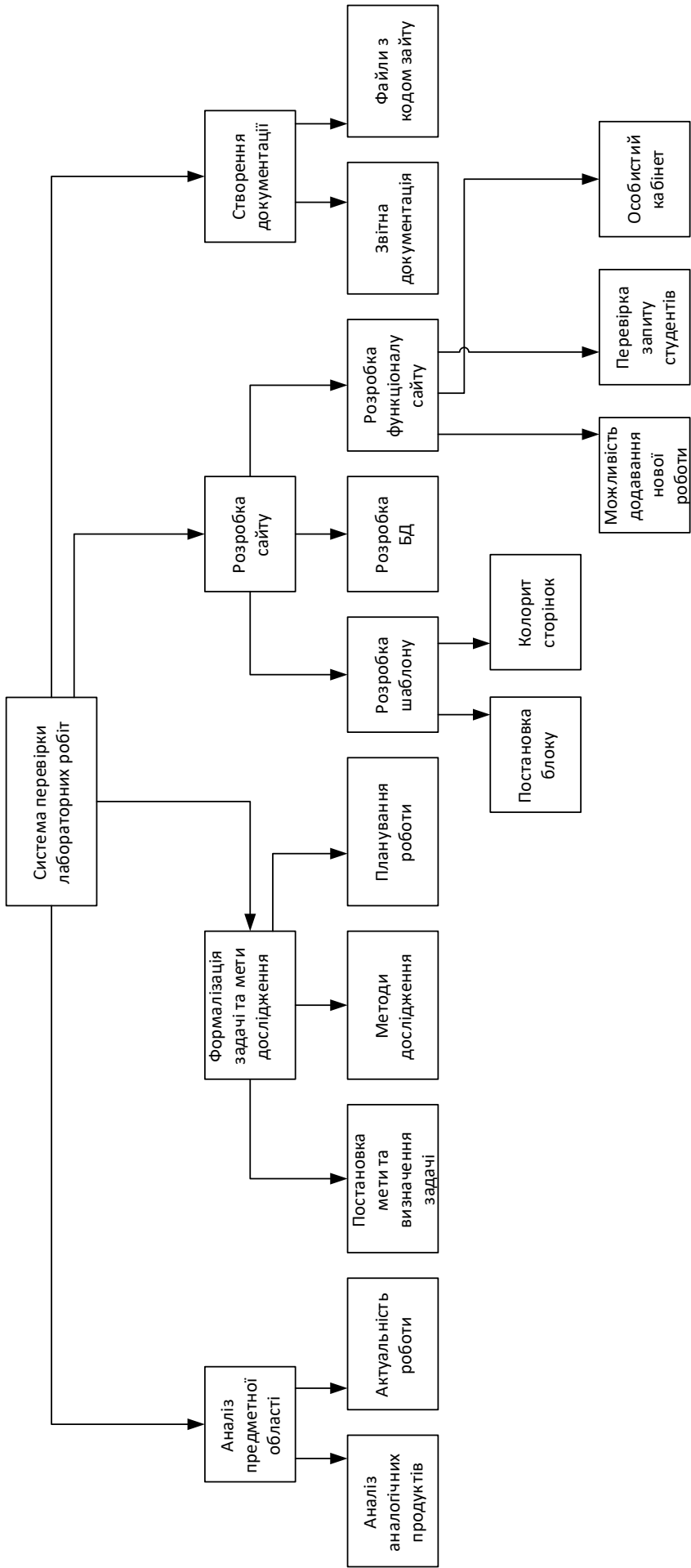


Рисунок Б.1 – Таблиця WBS
Джерело: власна розробка

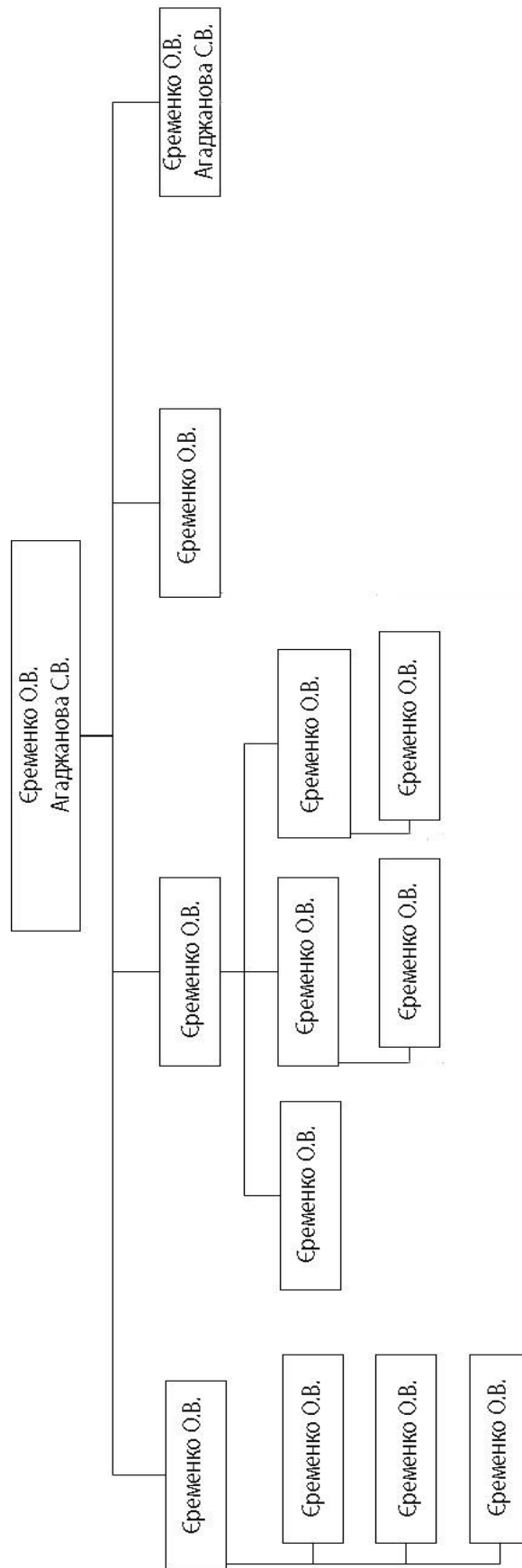


Рисунок Б.2 – Таблиця OBS
Джерело: власна розробка

Діаграма Ганта. Далі побудуємо календарний план виконання дипломного проекту. Найпоширеніший формат графіка в будь-якій галузі — діаграма Ганта. Цей графік дозволяє менеджерам проекту і всій команді розробників візуалізувати графіки часу і взаємозв'язок між окремими завданнями та етапами роботи над проектом. Тривалість виконання робіт зазначена в днях, але фактична тривалість виконання робіт приблизно дорівнює 2-3 години на день. Для того щоб мати реальне уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, з урахуванням вихідних та святкових днів, побудовано календарний графік. Діаграма Ганта та список робіт діаграми Ганта зображені на рис. Б.3.

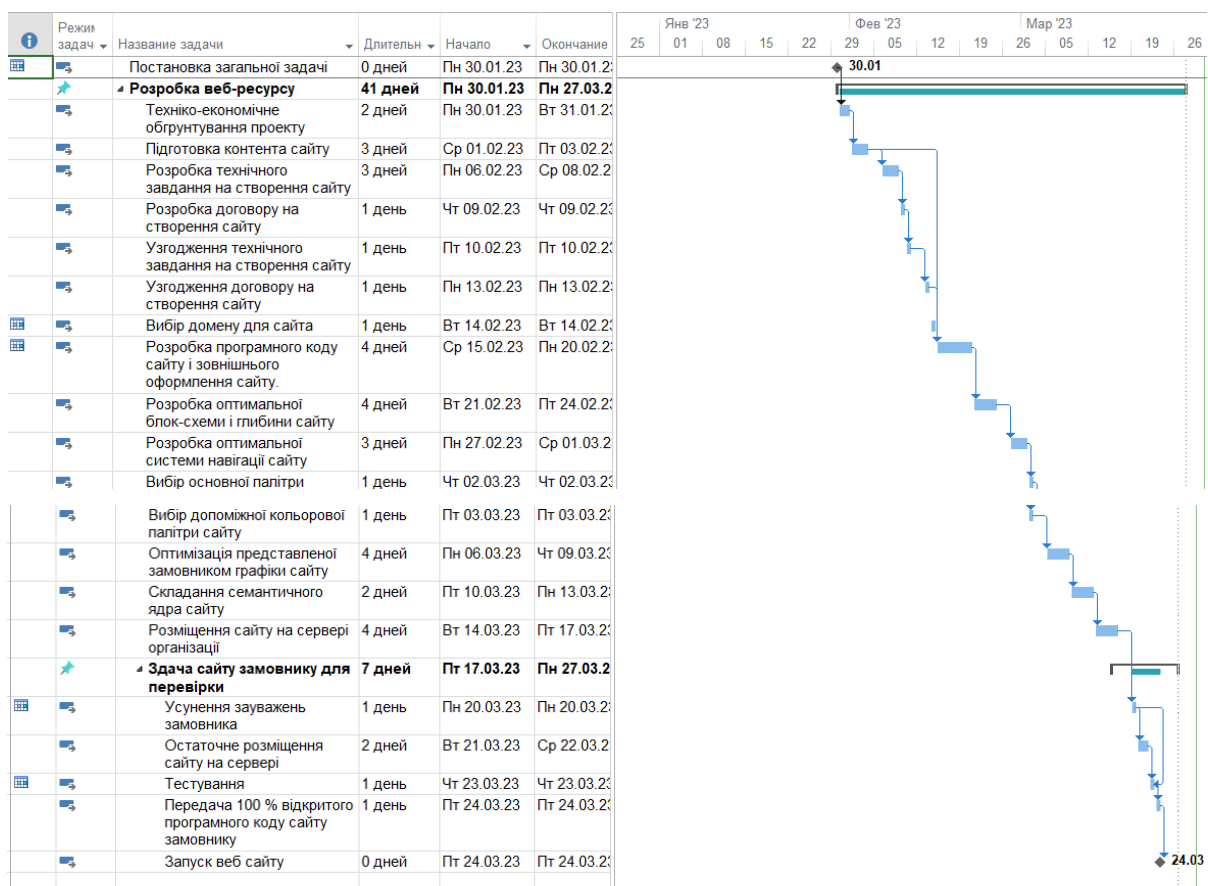


Рисунок Б.3 – Діаграма ганта

Планування ризиків проекту

Управління ризиками проекту – це процес виявлення, аналізу та реагування на будь-який ризик, що виникає протягом життєвого циклу проекту. Крім того, управління ризиками є частиною процесу планування, що необхідно для своєчасного з’ясування можливих ризиків, і контролю цих ризиків, якщо вони дійсно виникають. Ризик – це все, що потенційно може вплинути на терміни, продуктивність або бюджет проекту. Процес ідентифікації ризиків передбачає встановлення та документування характеристик ризиків, які можуть вплинути на проект. Цей процес є ітеративним, адже з розвитком проекту можуть з’являтися нові ризики, що потребуватимуть ідентифікації та реагування.

Перелічимо можливі ризики проекту, визначимо відносну критичність, а також можливі способи їх уникнення. Введемо умовну шкалу рівнів критичності ризиків (табл. Б.3):

Таблиця Б.3 – Рівні критичності ризиків

1	Незначний
2	Низький
3	Середній
4	Високий
5	Критичний

Ризики проекту, визначені в результаті ідентифікації, а також їх потенційний вплив та ймовірність виникнення представлені у таблиці Б.4.

Таблиця Б.4 – Ризики проекту

№	Опис	Ймовірність	Вплив
1	Втрата актуальності проекту	1	4
2	Зміна вимог на пізніх етапах розробки	2	4
3	Незаплановані роботи, що потребують обов’язкового виконання	2	3

Продовження таблиці Б.4

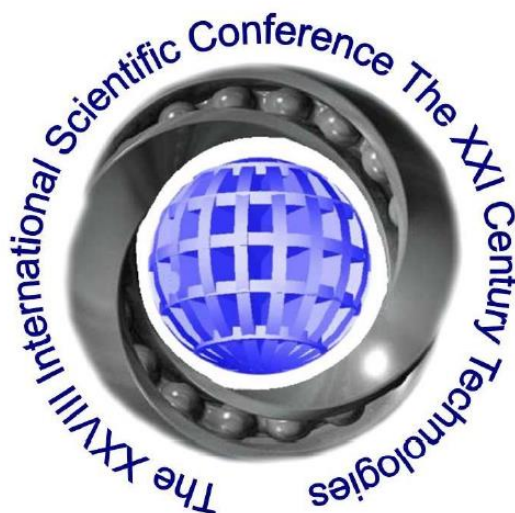
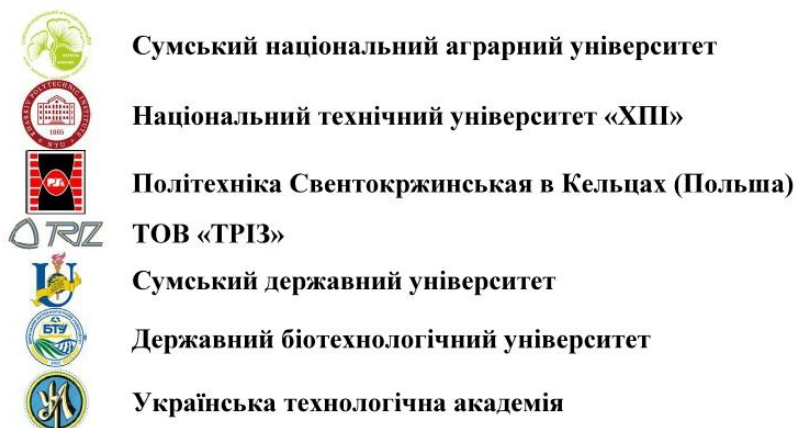
4	Вимушене зміщення термінів виконання завдань або паралельне виконання цих завдань	3	3
5	Нездійснені або важкі задачі	3	5
6	Пошкодження або відсутність обладнання	1	5
7	Непрацездатність розробника	1	5
8	Низьке залучення замовника	2	4

Тепер, складемо матрицю залежності втрат від ймовірності виникнення кожного із зазначених ризиків (табл. Б.5).

Таблиця Б.5 – Матриця «Ймовірність – Втрати»

Ймовірність \ Втрати	1	2	3	4	5
1				1	6,7
2			3	2,8	
3			4		5
4					
5					

ДОДАТОК В
КОПІЇ ПУБЛІКАЦІЙ



ТЕХНОЛОГІЇ ХХІ СТОРІЧЧЯ

Збірник тез за матеріалами 28^{ої} міжнародної науково-практичної конференції
(23-25 листопада 2022 р.)

Частина 2

Секції: «Прогресивні технології в економіці»,
«Прогресивні технології в екології та енергоефективність»,
«Прогресивні технології у педагогіці»

Суми – 2022

<i>Кушнірова Л.В., Сумський державний педагогічний університет ім. А.С.Макаренка, Суми, Україна</i>	
ПИТАННЯ ЛІНГВОДИДАКТИКИ В ДОРОБКУ В. МАСАЛЬСЬКОГО	164
<i>Мартинченко С.В., студентка, Агаджанов-Гонсалес К.Х., ст.викладач, СНАУ, Суми, Україна</i>	
СУЧАСНІ НАПРЯМКИ ЕКСТРАКЦІЇ МЕТАДАНИХ ДЛЯ СТРУКТУРУВАННЯ ТА ПОШУКУ ПРЕЗЕНТАЦІЙ	166
<i>Ребрій А.М., Сумський національний аграрний університет, Україна</i>	
ОСОБЛИВОСТІ ДИСТАНЦІЙНОГО НАВЧАННЯ ПІДГОТОВКИ СТУДЕНТІВ В АГРАРНІЙ ОСВІТІ.....	167
<i>Ребрій А.М., Сумський національний аграрний університет, Україна</i>	
ТЕХНОЛОГІЇ МОДЕЛЮВАННЯ ТА ВІЗУАЛІЗАЦІЯ ГЕОМЕТРИЧНИХ ОБ'ЄМНИХ ТІЛ.....	168
<i>Рибенко І.О., Сумський національний аграрний університет, Україна</i>	
СУЧАСНІ ТЕНДЕНЦІЇ ВИКЛАДАННЯ ГРАФІЧНИХ ДИСЦИПЛІН	169
<i>Рибенко І.О., Сумський національний аграрний університет, Україна</i>	
КОМП'ЮТЕРНА ГРАФІКА - ІНСТРУМЕНТ ДЛЯ ПІДВИЩЕННЯ ЯКОСТІ ЗНАНЬ	170
<i>Захарова І.О., к.п.н., доцент, СДПУ імені А.С.Макаренка, м.Суми, Україна</i>	
ПЕДАГОГІЧНІ ОСОБЛИВОСТІ ДИСТАНЦІЙНОГО НАВЧАННЯ СТУДЕНТІВ У ЗАКЛАДІ ВИЩОЇ ОСВІТИ.....	171
<i>Семерня О.В., Сумський національний аграрний університет</i>	
РОЛЬ ВИКЛАДАЧА В ОРГАНІЗАЦІЇ САМОСТІЙНОЇ ПІДГОТОВКИ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ З ПИТАНЬ ОХОРОНИ ПРАЦІ	173
<i>Сременко О. В., студент, Агаджанова С.В., к.т.н., доцент, СНАУ, Суми, Україна</i>	
ВИКОРИСТАННЯ ІНТЕРАКТИВНИХ ВЕБ-ТРЕНАЖЕРІВ В НАВЧАЛЬНОМУ ПРОЦЕСІ	176
<i>Гончар Д., Лапченко І., студенти, СНАУ, Суми, Україна</i>	
ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ У ОСВІТІ	177
<i>Статівка В.І., д.п.н., професор, Захарова І.О., к.п.н., доцент, СумДПУ ім. А.С. Макаренка, м. Суми, Україна</i>	
УПРАВЛІННЯ НАУКОВИМ ДОСЛІДЖЕННЯМ СТУДЕНТА: СПОСОБИ УПЕРЕДЖЕННЯ ПОМИЛОК У СТВОРЕННІ ТЕКСТОВОГО ВАРІАНТУ	178
<i>Баталова А.Б., старший викладач, СНАУ, Суми, Україна</i>	
МУЛЬТИМЕДІА, ЯК ЗАСІБ НАВЧАННЯ СТУДЕНТІВ В УМОВАХ ВІЙСЬКОВОГО СТАНУ	180
<i>Хурсенко С.М., к.ф.-м.н., доцент, СНАУ</i>	
САМОРОЗВИТОК ОСОБИСТОСТІ В УМОВАХ КОМПЕТЕНТНІСНО-ОРІЄНТОВАНОГО ПІДХОДУ	181
<i>Хурсенко С.М., к.ф.-м.н., доцент, СНАУ</i>	
МУЛЬТИМЕДІЙНІ ТЕХНОЛОГІЇ – ЗАСІБ НАВЧАННЯ НОВОГО ПОКОЛІННЯ.....	183
<i>Хіманова А.О., СНАУ, Україна</i>	
ВИКОРИСТАННЯ СИМУЛЯТОРІВ У ОСВІТНІЙ ДІЯЛЬНОСТІ УНІВЕРСИТЕТУ	184

- самостійна робота є організованою викладачем активною діяльністю здобувача освіти, направленою на виконання поставленої дидактичної мети, але здійснюється без посередньої участі викладача;
- самостійна робота студентів вимагає від викладача ретельного її планування;
- створення умов ефективної організації навчальної роботи здобувачів освіти припускає перш за все ґрунтовне науково-методичне їх забезпечення;
- від організації самостійної роботи студентів залежить результат засвоєння знань, умінь та навичок здобувача освіти, рівень самостійності.

Дії викладача спрямовують здобувачів освіти до організації і раціонального, ефективного здійснення активної, самостійної, свідомої і результативної пізнавальної діяльності для формування ризик-орієнтовного мислення у працівників і майбутніх керівників у галузі; зміна ролі здобувача освіти в навчальному процесі, перетворення його на активний, діяльний суб'єкт, а також перехід до демократичного стилю управління навчальним процесом, де сама особистість викладача виступає стимулом щодо формування інтересу до знань у здобувача освіти.

ЛІТЕРАТУРА

1. Гришук М.В. Основи охорони праці: Підручник. – К.: Кондор, 2005. – 240 с.
2. Дичківська І. М. Інноваційні педагогічні технології / І. М. Дичківська. – К. : Академвидав, 2004. – 351 с
3. Козаков В.А. Самостійна робота студентів як дидактична проблема / В.А. Козаков. - К. : НМК ВО, 1990. - 62 с.
4. Методика навчання і наукових досліджень у вищій школі: навч. посіб. / [ред. С.У. Гончаренко, П.М. Олійник, В.К. Федорченко]. - К.: Вища школа, 2003. - 323 с.
5. Платонов К.К., Голубев Г.Г. Психология. – М.: Просвещение, 1977. – С. 53-68.

УДК:378.14

Єременко О. В., студент, Агаджанова С.В., к.т.н., доцент, СНАУ, Суми, Україна

ВИКОРИСТАННЯ ІНТЕРАКТИВНИХ ВЕБ-ТРЕНАЖЕРІВ В НАВЧАЛЬНОМУ ПРОЦЕСІ

Вступ. З кожним роком система освіти стає все більш залежною від інформаційних технологій, і світ останнім часом переконався у цьому. За допомогою систем інтернет-комунікації студенти мають можливість дистанційно спілкуватися з викладачами, за допомогою хмарного сховища – обмінюватися даними, а за допомогою веб-тренажерів викладачі можуть навчати студентів та оцінювати їхні знання.

Основна частина. Інтерактивний веб-тренажер - це програмний комплекс, метою якого є імітація взаємодії людини з реальною системою. При використанні інтерактивних веб-тренажерів контроль якості знань відходить від узагальненості та стає індивідуальним для кожного студента з урахуванням темпу його навчання. Також автоматизований контроль підвищує об'єктивність самого контролю та дозволяє оцінити якість знань студентів не тільки поверхнево, а й дає кількісну оцінку якості засвоєння студентами того чи іншого навчального розділу [2].

Слід виділити наступні позитивні сторони веб-тренажерів [1]:

- враховується темп роботи кожного студента, який самостійно керує навчальним процесом;
- скорочується час, необхідний для розвитку певних навичок;
- збільшується кількість навчальних завдань;
- диференціація рівнів досягається порівняно просто;
- підвищується мотивація до навчальної діяльності.

Всі інтерактивні завдання в такому тренажері повинні припускати наявність зворотного зв'язку, можливість корекції дій і можливість виконання практичних дій. Інтерактивні зав-

дання в тренажері включають послідовність запитань та інтерактивних підказок і вправ. Навчальні комп'ютерні інтерактивні веб-тренажери можна використовувати на різних етапах підготовки здобувачів вищої освіти інженерних спеціальностей: актуалізація знань, вивчення та закріплення нового матеріалу, домашні завдання, самостійна робота, перевірка знань, і тд. На основі проходження навчальних інтерактивних комп'ютерних веб-тренажерів відбувається моніторинг відповідей студентів. У результаті з'являється можливість обробляти результати проходження навчальних інтерактивних комп'ютерних веб-тренажерів, аналізувати та оцінювати якість кожного завдання чи питання з точки зору його складності. Ефективне використання навчальних комп'ютерних веб-тренажерів у навчальному процесі дозволяє значно зменшити кількість помилок, підвищити швидкість маніпулювання та прийняття рішень, скоротити час навчання, більш адекватно оцінити рівень знань і набутих умінь, індивідуалізувати навчання, формувати висновки про дії майбутніх спеціалістів.

Можна виділити кілька класів тренажерів, які використовуються в навчальному процесі [3, ст.61]:

- тренажери, що навчають поведінці в незвичайних ситуаціях (наприклад, при моделюванні складних ситуацій при керуванні транспортними засобами);
- тренажери, що навчають працювати за алгоритмом (використовуються при навчанні роботі зі складним обладнанням);
- демонстраційний (ілюстративний) тренажер (показує деталі, пристрої, процеси);
- електронний екзаменатор програмного забезпечення (даний вид тренажера містить різні види тестів);
- тренажери, які навчають розпізнаванню образів (використовуються в навчанні з медичної діагностики);

Висновки. Отже, виходячи з матеріалу можна сказати, що навчальні веб-тренажери являються за своєю суттю електронними освітніми ресурсами, які передбачають багаторазове виконання здобувачами вищої освіти однотипних завдань з метою закріплення вивченого матеріалу та формування навчальних навичок, за допомогою веб-тренажерів студент здобуває необхідні йому знання, він може самостійно керувати навчальним процесом. Веб-тренажер у вузькому розумінні — це комп'ютерна навчальна програма для розвитку в студентів вищих навчальних закладів умінь і навичок певної діяльності, а також розвитку пов'язаних з нею здібностей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

6. Bielov, M. A., Antipov, O. E. (2010). Principles of designing a virtual computer lab based on cloud computing technology. Collection of proceedings of the international conference "Modern problems and solutions in science, transport, production and education", 1, 92
7. Особенности использования автоматизированного контроля знаний студентов: С.И. Дорошенко, В. С. Никольская, 2008. — с. 79
8. Палюх Б. В. Электронное обучение в инженерном образовании / Б. В. Палюх, А. В. Твардовский, В. К. Иванов // Качество образования. — 2012. — №10. — С.34 – 37

УДК 621.9.048

Гончар Д., Лапченко І., студенти, СНАУ, Суми, Україна

ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ У ОСВІТІ

Сучасний стан справ на ринку ІТ в Україні можна порівняти з золотою лихоманкою. Є певний вичерпний ресурс, котрим в нашому випадку є робота у сфері інформаційних технологій, популярність якого протягом кількох років зростає в геометричній прогресії. І є ті, хто зацікавлений у тому щоб цей вичерпний ресурс здавався нескінченним. Відповідно, щороку все більше і більше людей хочуть змінити свою основну роботу на роботу у сфері інформаційних технологій. Цим і обумовлюється актуальність даного дослідження.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

МАТЕРІАЛИ

науково-практичної конференції
викладачів, аспірантів та студентів
Сумського НАУ

(25-28 квітня 2023 р.)

Рудакова Д.О. РОЛЬ ЕКОЛОГІЧНОГО МАРКЕТИНГУ В ФОРМУВАННІ КОНКУРЕНТНИХ ПЕРЕВАГ ДЕРЖАВНОГО ПІДПРИЄМСТВА	339
Базарний С.О. УПРАВЛІННЯ ТОВАРНОЮ ПОЛІТИКОЮ НА ПІДПРИЄМСТВАХ МАЛОГО БІЗНЕСУ	340
Сергієнко В.Г. КАДРОВА БЕЗПЕКА ЯК ОСНОВНИЙ ЕЛЕМЕНТ ЕКОНОМІЧНОЇ БЕЗПЕКИ ПІДПРИЄМСТВА	341
Бельська А.О. РЕКЛАМНА ДІЯЛЬНІСТЬ НА ПІДПРИЄМСТВАХ РОЗДРІБНОЇ ТОРГІВЛІ ТА СПОСОБИ ЇЇ ВДОСКОНАЛЕННЯ	342
Дундук Д.С. РОЛЬ І ЗНАЧЕННЯ КРЕДИТУ В РОЗВИТКУ ЕКОНОМІКИ УКРАЇНИ	343
В'юненко О.Б. СУЧАСНІ ПРОБЛЕМИ ІНФОРМАЦІЙНОЇ КІБЕРГІГІЄНИ	344
Баталова А.Б. ГЕОІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ЯК ІНСТРУМЕНТ ЕКОЛОГІЧНИХ ДОСЛІДЖЕНЬ	345
Маяк А.О. ЦИФРОВІ ІНСТРУМЕНТИ ДЛЯ ПУБЛІКАЦІЇ ТА ПРОДАЖУ МУЗИКИ	346
Бусенко І.В. РОЗРОБКА ВІДЕОІГОР НА БАЗІ ТЕХНОЛОГІЇ P2E	347
Крамський А.С. ChatGPT – ТВІЙ МЕНТОР У НАВЧАННІ	348
Гервольський М.Д. АНАЛІТИЧНИЙ АНАЛІЗ РІЗНИХ ВИДИ ХОСТИНГІВ	349
Потапенко М.Т. АВТОМАТИЗОВАНА ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ СПОРТИВНОГО КОМПЛЕКСУ	350
Єременко О.В. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПЕРЕВІРКИ ПРАКТИЧНИХ РОБІТ ІЗ ВИБІРКИ ДАНИХ: ШЛЯХИ І ТЕХНОЛОГІЇ	351
Мотлюк М.В. А/В ТЕСТУВАННЯ ЯК МЕТОД МАРКЕТИНГОВИХ ДОСЛІДЖЕНЬ В ІТ	352
Олефіренко Д.В. ПРОГНОЗУВАННЯ КІЛЬКОСТІ ВЕЛИКОЇ РОГАТОЇ ХУДОБИ В УКРАЇНІ З ВИКОРИСТАННЯМ ЕКОНОМЕТРИЧНИХ МЕТОДІВ	353
Бабуров Д.В. ПРОГНОЗУВАННЯ КІЛЬКОСТІ СІЛЬСЬКОГОСПОДАРСЬКОЇ ПТИЦІ В УКРАЇНІ З ВИКОРИСТАННЯМ ЕКОНОМЕТРИЧНИХ МЕТОДІВ	354
Шакотько А.М. ІНФОРМАЦІЙНА ПІДСИСТЕМА МАТЕРІАЛЬНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВСП «ГАФК СНАУ»	355
Басанець Д.В. ІНФОРМАЦІЙНА ПІДСИСТЕМА ОБЛІКУ КАДРОВОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ КОЛЕДЖУ ВСП «ГАФК СНАУ»	356
Хіманова А.О. ПРОБЛЕМИ УПРАВЛІННЯ ІТ ПРОЄКТАМИ СУЧАСНИХ НАВЧАЛЬНИХ СИСТЕМ	357
Сіденко А.С. ПРОБЛЕМИ ПОБУДОВИ САЙТІВ ДЛЯ СИСТЕМ ДИСТАНЦІЙНОГО НАВЧАННЯ ...	358
Шакотько Є.С. СУЧАСНІ АЛГОРИТМИ АНАЛІЗУ ПРОДУКТИВНОСТІ ТА МОНІТОРИНГУ КОМП'ЮТЕРНИХ МЕРЕЖ	359
Грошовик Р.Ю. ОБГРУНТУВАННЯ НЕОБХІДНОСТІ АВТОМАТИЗАЦІЇ ВЕДЕННЯ ПОГОСПОДАРСЬКОГО ОБЛІКУ ТЕРИТОРІАЛЬНИХ ГРОМАД	360
Наконечний О.Б. АНАЛІЗ ПРОБЛЕМИ АВТОМАТИЗАЦІЇ РОЗРАХУНКУ БАЛАНСУ ГУМУСУ І ПОЖИВНИХ РЕЧОВИН В ҐРУНТІ ТА ПРОПОЗИЦІЇ ЩОДО ЇЇ ВИРІШЕННЯ	361
Мартинченко С.В. ОБРОБКА ВЕЛИКИХ ДАНИХ: ПРОБЛЕМИ І ШЛЯХИ ЇХ ВИРІШЕННЯ	362
Zhang Yuanyi THE KEY FACTORS IN PERSONAL DEVELOPMENT FROM THE PERSPECTIVE OF ENTERPRISES	363
Qin Fan IMPROVEMENT OF THE RISK MANAGEMENT SYSTEM AT THE ENTERPRISE	364
Jiang Maoli ORGANIZE THE CONCEPT AND METHOD OF BUSINESS PROCESS IMPROVEMENT	365
Zhang Buyao DISCUSSION ON THE MANAGEMENT OF CAREER PROMOTION OF ENTERPRISE EMPLOYEES	366
Cui Xiaoping THE IMPACT OF INNOVATIVE TRANSFORMATIONS IN THE AGRO-INDUSTRIAL SPHERE ON THE ACHIEVEMENT OF THE SUSTAINABLE DEVELOPMENT GOALS	367
Zhao Jing CONCEPTUAL APPROACHES TO MANAGING THE BEHAVIOR OF THE COMPANY'S PERSONNEL ON THE BASIS OF CORPORATE CULTURE	368
Гулько Б. В. МЕХАНІЗМ ДЕРЖАВНОГО РЕГУЛЮВАННЯ В СФЕРІ СТАЛОГО РОЗВИТКУ	369
Маслак О.М., Санін Р.С. ІНСТРУМЕНТИ ДЕРЖАВНОГО РЕГУЛЮВАННЯ ФЕРМЕРСЬКИХ ГОСПОДАРСТВ УКРАЇНИ В УМОВАХ ВОЄННОГО СТАНУ	370
Ніколаєнко Т.С. НАПРЯМИ РОЗВИТКУ ПУБЛІЧНОГО АДМІНІСТРУВАННЯ ОСВІТИ НА РІВНІ ТЕРИТОРІАЛЬНОЇ ГРОМАДИ	371
Гребченко Т.А., Стеценко Б.М. КОМУНІКАЦІЙНИЙ ЗВ'ЯЗОК ОРГАНІВ ДЕРЖАВНОГО УПРАВЛІННЯ ІЗ СУСПІЛЬСТВОМ	372
Майданевич С.В. АДМІНІСТРАТИВНО-ПРОЦЕДУРНІ ЗАСАДИ РЕГУЛЮВАННЯ В БАНКІВСЬКІЙ СФЕРІ	373
Chonglong Bi MANAGEMENT OF ORGANIZATIONAL CULTURE AS A FACTOR OF INCREASING COMPETITIVENESS	374
Тимошенко М.І. ПРИНЦИПИ СТВОРЕННЯ ТА УПРАВЛІННЯ КЛАСТЕРНИМИ ТА НАДКЛАСТЕРНИМИ ЛІКАРНЯМИ	375

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПЕРЕВІРКИ ПРАКТИЧНИХ РОБІТ ІЗ ВИБІРКИ ДАНИХ: ШЛЯХИ І ТЕХНОЛОГІЇ

Єременко О. В., студ. 2 с.т. курсу ФЕІМ, спец. «Інформаційні системи та технології»
Науковий керівник: к.т.н., доцент Агаджанова С.В.
Сумський НАУ

У сучасному світі навчання інформаційним технологіям набуває все більшої популярності. Зараз неможливо уявити освіту без використання комп'ютера та Інтернету. Однак, навчання в комп'ютерних класах, лабораторіях та інших приміщеннях, де проводяться практичні заняття, потребує контролю якості виконання студентами практичних робіт. Інформаційні системи перевірки практичних робіт є важливим елементом в цьому процесі.

Метою представленої роботи є розробка інформаційної системи для перевірки практичних робіт із вибірки даних. Дана система дозволить автоматизувати процес перевірки практичних робіт та спростити роботу викладачів та студентів. Одним із шляхів розробки такої системи є збір та обробка даних. Для цього можуть бути використані різні методи збору даних, такі як опитування та анкетування студентів. Після збору даних можна провести їх аналіз та обробку, що дає можливість створити вибірку даних для подальшої перевірки робіт.

Далі, для розробки інформаційної системи перевірки практичних робіт, необхідно створити алгоритми перевірки. Для цього можна використовувати різні підходи, включаючи стандартні методи перевірки робіт та методи машинного навчання. Використання машинного навчання та штучного інтелекту дозволяє підвищити точність та ефективність перевірки робіт.

Розроблена інформаційна система може також забезпечити можливість відстеження результатів навчання студентів, що дає можливість вчасно виявляти та вирішувати проблеми та недоліки у навчальному процесі.

Загалом, розробка інформаційної системи перевірки практичних робіт із вибірки даних є актуальною та важливою задачею для сучасної системи вищої освіти. Використання відповідних технологій та методів дозволяє зменшити навантаження на викладачів, підвищити якість та об'єктивність перевірки робіт студентів та покращити навчальний процес в цілому.

Отже, розробка інформаційної системи перевірки практичних робіт із вибірки даних може бути реалізована за допомогою шляхів та технологій, які забезпечують швидку та ефективну перевірку робіт студентів. Результатом розробки такої системи є значне зменшення часу та зусиль, які викладачі витрачають на перевірку лабораторних робіт.

За планом виконання дослідження в рамках підготовки кваліфікаційної роботи було визначено актуальність проблеми, головну мету та задачі проекту. Оглянуто актуальні літературні джерела за означеними темами: реінжиніринг ПЗ, синтаксичний аналіз SQL запитів та проектування бази знань для підсистеми перевірки завдань. Проведено детальний аналіз схожих за тематикою та функціоналом програмних продуктів, і сформовано порівняльну таблицю цих рішень за визначеними характеристиками.

За для досягнення поставленої мети було проведено планування робіт проекту. Ідентифіковано ідею проекту, проведено деталізацію мети за методологією SMART, планування змісту структури робіт у вигляді WBS та OBS, побудовано діаграму Ганта і визначено ризики проекту. Обрано наступні засоби реалізації: для серверної частини web-застосунку було обрано мову програмування PHP 7.2 та систему керування базами даних MySQL 8.0, а для реалізації клієнтської частини використовується мова розмітки HTML5, таблиці стилів CSS3, CSS-фреймворк Bootstrap 5.1 та мова програмування JavaScript за стандартом ES8.

Створено модель функціонування головного бізнес-процесу (Організація виконання та оцінювання завдань з лабораторних робіт) у нотатції IDEF0, а саме: її контекстну діаграму, та діаграми декомпозиції першого і другого рівня. Побудовано модель варіантів використання системи, спроектовано моделі бази даних системи: концептуальну і даталогічну, модель бази даних для бази знань з перевірки та оцінювання наданих відповідей, а також модель навчальної бази даних.

Визначено та впроваджено найбільш вдалий архітектурний шаблон програмного забезпечення – MVC. Згідно цього шаблону було модернізовано програмний код web-застосунку: створено окремі моделі для кожної сутності бази даних системи, контролери та вигляди для кожної функціональної сторінки системи.

Реалізовано модуль автоматизованої перевірки відповідей надісланих студентами, який передбачає перевірку правильності синтаксису та семантики відповіді (SQL запиту), та відображення підказок за результатами перевірки, які допомагають визначити причину помилки. Розширено функціонал для роботи з інформацією про користувачів. Значно покращено ефективність роботи з користувацьким інтерфейсом: спрощено навігацію між завданнями однієї роботи, до ряду таблиць додано фільтри за встановленими атрибутами та можливість сортування записів, виправлено проблему з перевантаженням елементами інтерфейсу окремих сторінок системи.

ДОДАТОК Г

ЛІСТИНГ ФАЙЛІВ ПРОГРАМНОГО КОДУ

Лістинг основних файлів керування роботою додатку:

```
index.php
<html>
<meta charset="utf-8">
<head>
<title>Авторизація</title>
<link type="text/css" rel="stylesheet" href="css/style-login.css">
</head>
<body>
<div class="cssload-container">
<div class="cssload-loading"><i></i><i></i></div>
</div>
<form class="box" action="auth.php" method="post">
<div class="header-logo">
<a href="index.php" class="logo">

</a>
</div>
<p>Перевірка лабораторних робіт!</p>
<div class="inputs">
<input type="text" name="login" placeholder="Логін">
<input type="password" name="pass" placeholder="Пароль">
</div>
<button type="submit">Увійти</button>
<a href="./register.php">Реєстрація</a>
<?php
if ($_SESSION['alert']) {
echo "<p class='alert'> <span class='closebtn'
onclick='this.parentElement.style.display='none';>x</span> " . $_SESSION['alert'] . '
</p>';
} elseif ($_SESSION['msg']) {
echo "<p class='success'> <span class='closebtn'
onclick='this.parentElement.style.display='none';>x</span> " . $_SESSION['msg'] . '
</p>';
}
unset($_SESSION['alert']);
unset($_SESSION['msg']);
?>
</div>
</form>
```

```

</body>
</html>
<script>
var close = document.getElementsByClassName("closebtn");
var i;
for (i = 0; i < close.length; i++) {
close[i].onclick = function() {
var div = this.parentElement;
div.style.opacity = "0";
setTimeout(function() {
div.style.display = "none";
}, 600);
}
}
let mask = document.querySelector('.cssload-container');
window.addEventListener('load', () => {
mask.classList.add('hide');
setTimeout(() => {
mask.remove();
}, 600);
});
</script>

```

```

admin.php
<html>
<head>
<meta charset="utf-8">
<title>Адміністратор </title>
<link type="text/css" rel="stylesheet" href="css/bootstrap.min.css" />
<link type="text/css" rel="stylesheet" href="css/style-main.css" />
</head>
<body>
<div class="exit">
<a href="index.php">Вийти</a>
</div>
<div class="cssload-container">
<div class="cssload-loading"><i></i><i></i></div>
</div>
<header>
<div id="header">
<div class="container">
<div class="row">
<div class="col-md-3">
<div class="header-logo">
<a href="admin.php" class="logo">

```



```

<div class="cat-body">
<h3>Додавання<br>груп</h3>
<a href="grouper.php" class="cta-btn">Відкрити<i class="fa
fa-arrow-circle-right"></i></a>
</div>
</div>
</div>
<div class="col-md-4 col-xs-6">
<div class="cat">
<div class="cat-img">

</div>
<div class="cat-body">
<h3>Переглянути<br>користувачів</h3>
<a href="users.php" class="cta-btn">Відкрити<i class="fa fa-
arrow-circle-right"></i></a>
</footer>
<script>
let mask = document.querySelector('.cssload-container');
window.addEventListener('load', () => {
mask.classList.add('hide');
setTimeout(() => {
mask.remove();
}, 600);
});
</script>
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>
<?php
session_start();
require_once 'connect.php';
$login = $_POST['login'];
$pass = md5($_POST['pass']);
$check_user = mysqli_query($mysql, "SELECT * FROM `users` WHERE `login` =
'$login' AND
`pass` = '$pass'");
if (mysqli_num_rows($check_user) > 0) {
if ($login === "admin") {
header('Location: admin.php');
} elseif ($login === "yerenenko") {
header('Location: teacher.php');
} else {
$user = mysqli_fetch_assoc($check_user);

```

```

$_SESSION['user'] = [
    "id" => $user['id_user'],
    "name" => $user['name'],
    "surname" => $user['surname'],
    "id_group" => $user['id_group']
];
header('Location: student.php');
}
} else {
$_SESSION['alert'] = 'Не вірний логін або пароль!';
header('Location: index.php');
}
?>

```

Файл коду auth.php

```

<?php
session_start();
require_once 'connect.php';
$login = $_POST['login'];
$pass = md5($_POST['pass']);
$check_user = mysqli_query($mysql, "SELECT * FROM `users` WHERE `login` =
'$login' AND
`pass` = '$pass'");
if (mysqli_num_rows($check_user) > 0) {
if ($login === "admin") {
header('Location: admin.php');
} elseif ($login === "yerenenko") {
header('Location: teacher.php');
} else {
$user = mysqli_fetch_assoc($check_user);
$_SESSION['user'] = [
    "id" => $user['id_user'],
    "name" => $user['name'],
    "surname" => $user['surname'],
    "id_group" => $user['id_group']
];
header('Location: student.php');
}
} else {
$_SESSION['alert'] = 'Не вірний логін або пароль!';
header('Location: index.php');
}
?>

```

Файл коду register.php

```

<?php
session_start();
?>
<html>
<meta charset="utf-8">
<head>
<title>Рєєстрація</title>
<!-- Custom stylesheet -->
<link type="text/css" rel="stylesheet" href="css/style-login.css">
</head>
<body>
<div class="cssload-container">
<div class="cssload-loading"><i></i><i></i></div>
</div>
<form class="box" action="reg.php" method="post">
<div class="header-logo">
<a href="register.php" class="logo">

</a>
</div>
<p>Перевірка твоєї якості!</p>
<input class="input" type="text" placeholder="Введіть ім'я" name="name">
<input class="input" type="text" placeholder="Введіть прізвище" name="surname">
<input class="input" type="text" placeholder="Введіть логін" name="login">
<input class="input" type="password" placeholder="Введіть пароль" name="pass">
<input class="input" type="password" placeholder="Повторіть пароль"
name="pass_conf">
<select name="group" class="sel" required>
<option selected value="">Оберіть свою групу</option>
<?php
require_once 'connect.php';
$result = mysqli_query($mysql, "SELECT * FROM `groups` WHERE `g_name` NOT IN
('Адміністратор','Викладач')");
while ($row = mysqli_fetch_assoc($result)) {
?>
<option class="sel_groups" value="<?php echo "$row[id_group]" ?>"><?php echo
"$row[g_name]" ?></option>
<?php } ?>
</select>
<button type="submit">Зареєструватися</button>
<a href="index.php">Увійти</a>
<?php
if ($_SESSION['alert']) {
echo "<p class='alert'> <span class='closebtn'

```

```

onclick='this.parentElement.style.display='none';>x</span> " . $_SESSION['alert'] . '
</p>';
}
unset($_SESSION['alert']);
?>
</form>
</body>
</html>
<script>
var close = document.getElementsByClassName("closebtn");
var i;
for (i = 0; i < close.length; i++) {
close[i].onclick = function() {
var div = this.parentElement;
div.style.opacity = "0";
setTimeout(function() {
div.style.display = "none";
}, 600);
}
}
let mask = document.querySelector('.cssload-container');
window.addEventListener('load', () => {
mask.classList.add('hide');
setTimeout(() => {
mask.remove();
}, 600);
});
</script>

```

tester.php

```

<?php
session_start();
?>
<html>
<head>
<meta charset="utf-8">
<title>Студент</title>
<link type="text/css" rel="stylesheet" href="css/bootstrap.min.css" />
<link type="text/css" rel="stylesheet" href="css/style-main.css" />
</head>
<body>
<div class="exit">
<a href="index.php">Вийти</a>
</div>

```



```

<form action="test.php" method="post">
<div class="box">
<h2>
<?
require_once 'connect.php';
$id = $_GET['id'];
$result1 = mysqli_query($mysql, "SELECT * FROM `labs` WHERE id_lab =
'$id'");
$lab = mysqli_fetch_assoc($result1);
$result2 = mysqli_query($mysql, "SELECT * FROM `tasks` WHERE id_lab =
'$id'");
$count_task = 0;
$i = 0;
echo $lab['lab'];
?>
<input type="text" value="<? echo $id; ?>" name="id" style="display:
none;">
<input type="text" value="<? echo $id_user; ?>" name="id_user"
style="display: none;">
</h2>
<?
while ($select_while1 = mysqli_fetch_assoc($result2)) {
$i++;
$count = 0;
$count_task++;
$result3 = mysqli_query($mysql, "SELECT * FROM `tasks` INNER JOIN
`answers` ON tasks.id_task=answers.id_task WHERE tasks.id_lab = '$id' AND
answers.id_task =
'$select_while1[id_task]'");
echo '<div class="task">' . $count_task, ' . ' . $select_while1['task'] .
'</div>';
if ($select_while1['id_type'] == '1') {
while ($select_while2 = mysqli_fetch_assoc($result3)) {
$count_radio++;
$count++;
echo '<div class="type1" id="type1"><div class="radio-
check"><label for="rr' . $count_radio . '">' . $select_while2['answer'] . '</label><input
type="radio" id="rr' . $count_radio . '" name="" . $i . '" value="" . $count . '"></div></div>
';
}
} else {
echo '<div class="type2" id="type2"><input class="input" type="text"
name="" . $i . '"></div>';
}
}
}

```

```

?>
<button type="submit">Відповісти</button>
<?
if ($_SESSION['alert']) {
echo "<p class='alert'> <span class='closebtn'
onclick='this.parentElement.style.display='none';>x</span> " . $_SESSION['alert'] . '
</p>';
} elseif ($_SESSION['msg']) {
echo "<p class='success'> <span class='closebtn'
onclick='this.parentElement.style.display='none';>x</span> " . $_SESSION['msg'] . '
</p>';
}
unset($_SESSION['alert']);
unset($_SESSION['msg']);
?>

```

Файл коду task.php

```

<?php
session_start();
require_once 'connect.php';
$lab_name = $_POST['lab_name'];
$q = mysqli_fetch_assoc(mysqli_query($mysql, "SELECT COUNT(*) FROM `labs`
WHERE
`lab`='$lab_name'"));
$check = join("", $q);
if ($lab_name == "") {
$_SESSION['alert'] = 'Заповніть усі поля!';
header('Location: tasker.php');
} elseif ($check > 0) {
$_SESSION['alert'] = 'Робота з такою назвою вже існує!';
header('Location: tasker.php');
} else {
mysqli_query($mysql, "INSERT INTO `labs` (`id_lab`, `lab`) VALUES (NULL,
'$lab_name')");
$_SESSION['msg'] = $lab_name. ' створена!';
header('Location: tasker.php');
}

```