МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: Web-додаток оцінювання лабораторних робіт з дисципліни «Організація баз даних та знань»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ-61-8 Васильченко Роман Сергійович

Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою _____ «___»
2020 р.

Науковий керівник _____ к. т. н., доц. Марченко А.В.
(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії _____ _____ Шифрін Д.
М.
(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі
немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2020

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ
Зав. секцією ІТП

_____
В. В. Шендрик
«__» _____ 2020 р.

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

Васильченко Роман Сергійович

1 Тема роботи *Web-додаток оцінювання лабораторних робіт з дисципліни «Організація баз даних та знань»*

керівник роботи *Марченко Анна Вікторівна, к.т.н., доцент*,

затверджені наказом по університету від « 14 » травня 2020 р. № 0576-III

2 Строк подання студентом роботи «1» червня 2020 р.

3 Вхідні дані до роботи технічне завдання на розробку web-додатку, методичні вказівки і вимоги до оформлення дипломної роботи

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, постановка задачі та методи дослідження, проектування програмного додатку, розробка web-додатку для перевірки лабораторних робіт студентів

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень постановка задачі, аналіз аналогів, моделювання програмного додатку, аналіз технологій, етапи розробки web-додатку, висновки

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7.Дата видачі завдання01,10,2019_____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	02.03.20 – 05.03.20	
2	Пошук аналогів та обґрунтування потреби	06.03.20 – 10.03.20	
3	Визначення вимог	11.03.20 – 11.03.20	
4	Визначення інструментарію	12.03.20 – 12.03.20	
5	Планування wbs та obs	13.03.20 – 13.03.20	
6	Складання календарного плану	16.03.20 – 17.03.20	
7	Визначення ризиків	18.03.20 – 19.03.20	
8	Розробка інтерфейсу користувача	20.03.20 – 26.03.20	
9	Проектування бази даних	27.03.20 – 03.04.20	
10	Розробка програмних модулів	06.04.20 – 30.04.20	
11	Компілювання програмного коду	01.05.20 – 11.05.20	
12	Тестування розробником	12.05.20 – 14.05.20	
13	Тестування незалежною особою	15.05.20 – 19.05.20	
14	Оформлення документації	11.03.20 – 21.05.20	
15	Архівація проекту	22.05.20 – 25.05.20	
16	Введення в експлуатацію	26.05.20 – 31.05.20	

Студент _____ Васильченко Р.С.
(підпис)

Керівник роботи _____ к.т.н., доц. Марченко А.В.
(підпис)

РЕФЕРАТ

Тема дипломної роботи: Web-додаток оцінювання лабораторних робіт з дисципліни «Організація баз даних та знань»

Дипломна робота складається зі вступу, трьох розділів, висновка, списку використаної літератури та додатків.

Пояснювальна записка містить 78 с., 59 рис., 8 табл., 3 додатки, 16 джерел.

У першому розділі досліджується актуальність проблеми, виконується пошук та аналіз існуючих аналогів, формується мета дипломної роботи та задачі проекту, виконується аналіз технологій для реалізації проекту.

Другий розділ присвячений проектуванню програмного додатку, де наведені діаграми у нотації IDEF0, Use Case, ER-діаграму та базу даних.

В останньому розділі виконується детальний опис практичної реалізації програмного додатку: створення прототипу діалогового вікна, розробка бази даних, розробка головного функціоналу та допоміжних модулів.

Результатом виконаної роботи є розроблений програмний додаток, який виконує автоматичнуу перевірку лабораторних робіт студентів, зберігає отримані дані у базі даних та у вигляді звіту за бажанням користувача.

Ключові слова: PHP, WEB-ДОДАТОК, БАЗИ ДАНИХ, ЛАБОРАТОРНІ РОБОТИ.

ЗМІСТ

ВСТУП 6
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 8
1.2 Аналіз програмних продуктів – аналогів 9
1.3 Постановка задачі 15
2 МОДЕЛЮВАННЯ WEB-ДОДАТКУ 16
2.1 Моделювання бізнес-процесів web-додатку 16
2.2 Моделювання варіантів використання web-додатку 19
2.3 Проектування моделі бази даних 21
3. РОЗРОБКА WEB-ДОДАТКУ 23
3.1 Архітектура web-додатку 23
3.3 Програмна реалізація 28
3.4 Використання web-додатку 34
ВИСНОВКИ 47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 49
Додаток А 50
Додаток Б 62
Додаток В 67

ВСТУП

Існує глобальна проблема сьогодення, що пов'язана з контактним зв'язком, а також потреба в реалізації методів її подолання. Кожен заклад виходить з ситуації, система навчання повністю автоматизувалась, але не завжди це вдало. Якщо це однотипні завдання, відповідь на які у всіх студентів однакова, автоматизована система навчання значно полегшить роботу викладача, інша річ коли предмет потребує творчого підходу, де ця система навпаки погіршить ситуацію.

Метою дипломного проекту є розроблення web-додатку автоматизованої перевірки лабораторних робіт з написання запитів до бази даних мовою SQL.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- Провести аналіз предметної області;
- Сформувати та затвердити технічне завдання;
- Розробити шаблон сайту;
- Створити базу даних;
- Реалізувати систему перевірки SQL-запитів;
- Провести тестування.

Студентів велика кількість і у кожного є своя робота котру потрібно перевірити викладачеві. Це все займає багато часу, тому було вирішено розробити сайт, на якому студенти матимуть змогу вирішувати завдання онлайн, та отримувати результат одразу.

Основні переваги:

- Економія часу при підготовці до лабораторних завдань.
- Простота у використанні. Мінімум найнеобхідніших функцій для оволодіння якими вам досить буде півгодини.
- Накопичення і систематизація лабораторних матеріалів, які надійно збережуться в електронному вигляді.

– Розвиток навичок в роботі з новими технологіями викладачів і студентів.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

У процесі аналізу задачі перевірки правильності побудови запитів до бази даних було виявлено декілька можливих варіантів її вирішення.

Перший метод базується на порівнянні тексту еталонного запиту, визначеного викладачем, та розв'язку студента. Але, нажаль, у цей спосіб не завжди можна буде з високою точністю оцінити роботу студента, оскільки часто два однакові за функціональністю запити мають синтаксичні відмінності.

Другий метод полягає у порівнянні результатів запитів студента та викладача. Результатом може бути сповіщення про успішне виконання операції, помилка або таблиця з вибіркою. Недоліком цього способу є відсутність можливості перевірки застосування у запиті потрібних операторів, згідно завданню та меті практичної роботи.

Враховуючи переваги і недоліки методів можна спробувати скомбінувати їх таким чином, щоб скористатись перевагами обох.

Визначений такий комбінований підхід :

Приведення рядків до верхнього регістру, розкладення їх на окремі слова, вважаючи символ пробілу роздільником, та перевірка наявність потрібних операторів у запити студента. Після цього застосувати другий описаний вище метод. [2]

1.2 Аналіз програмних продуктів – аналогів

Цей проект можна вважати унікальним, у вільному доступі мережі Інтернет було знайдено лише подібні сайти, які частково відповідають меті проекту. В кожному із них є слабкі та сильні сторони, але всі вони є гарним тренажером для вдосконалення навичок в написанні SQL-запитів [1].

– Sql-ex і sql-tutorial – два сайти які відносяться до спільного ресурсу. Так званий онлайн підручник, в якому користувач може змінити мову, зареєструватись, переглянути контент. [2]

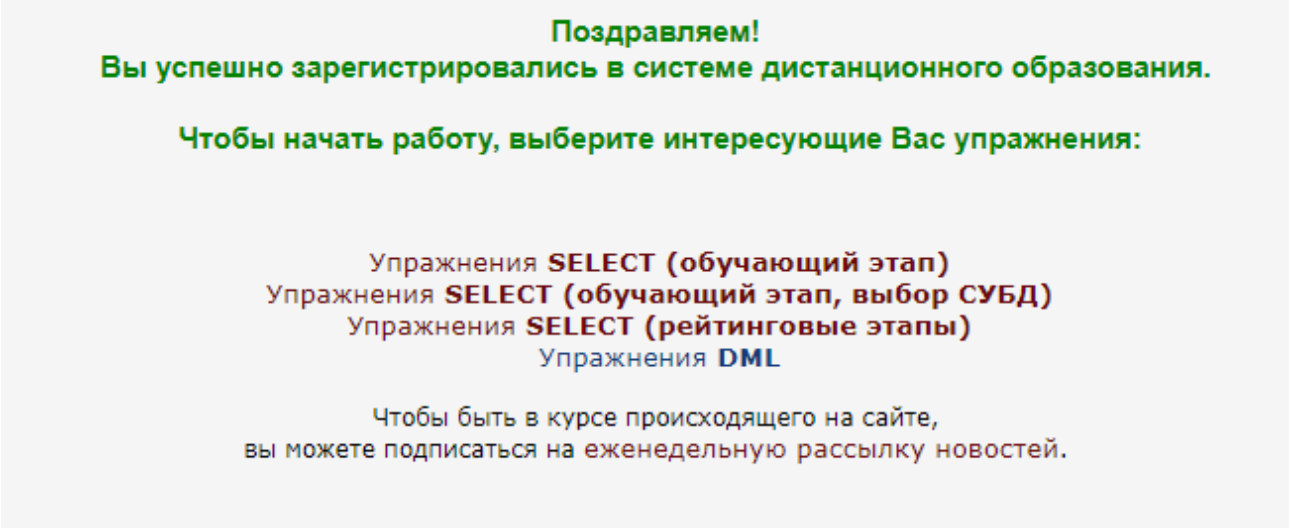


Рисунок 1.1 – Вікно вибору етапів вправ

Для початку роботи потрібно зареєструватися. Після чого слід обрати етап проходження завдань

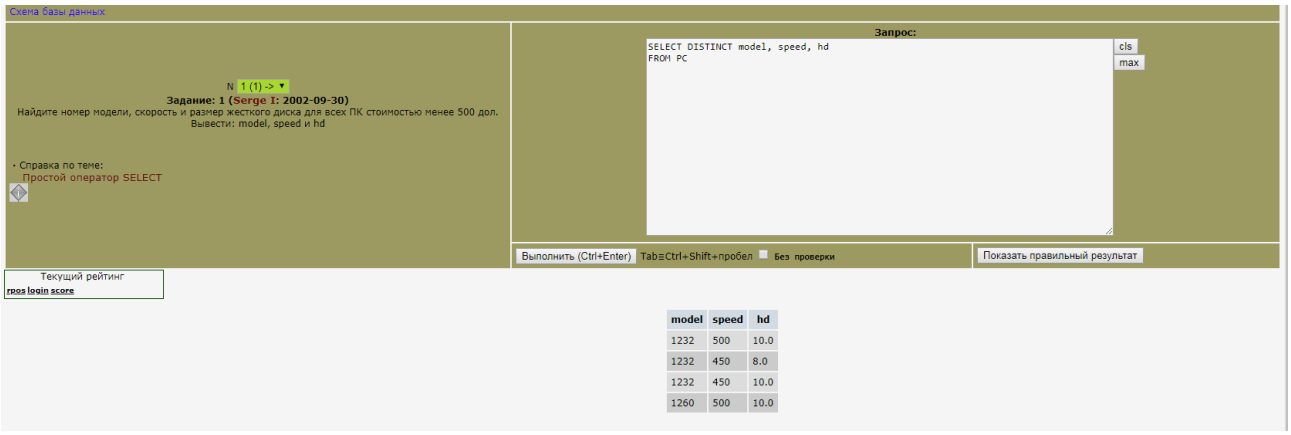


Рисунок 1. 2 – Вікно написання запиту

Також є сторінка з методичними матеріалами на якій користувач має змогу знайти потрібну йому інформацію.

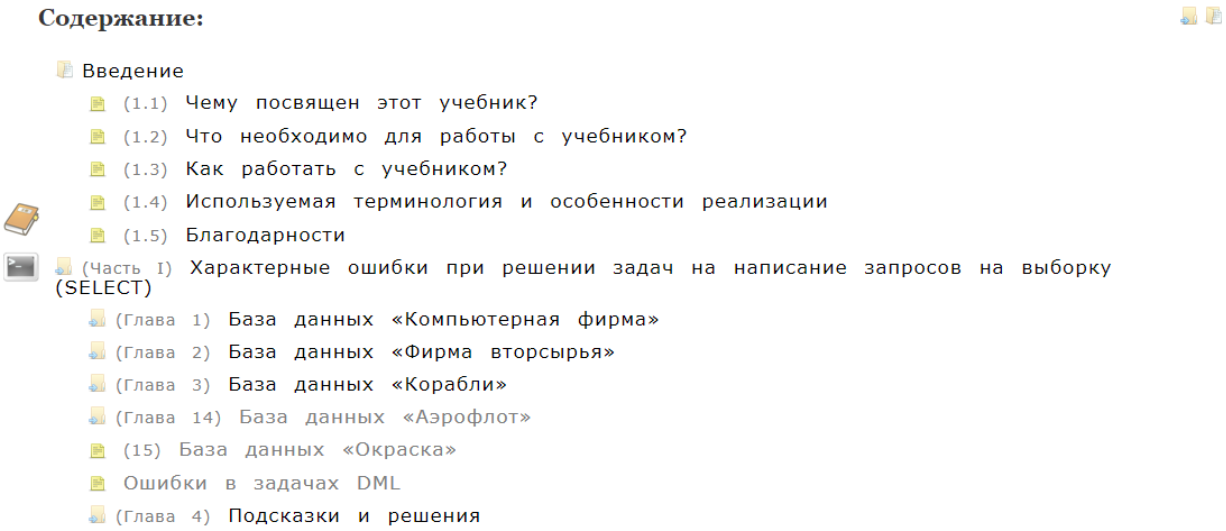


Рисунок 1. 3 – Вікно вибору етапів вправ

– Pgexercises – сайт виконаний в спокійних тонах, мінімалістично. Немає непотрібної реклами, зручно виконане меню з декількома рівнями.

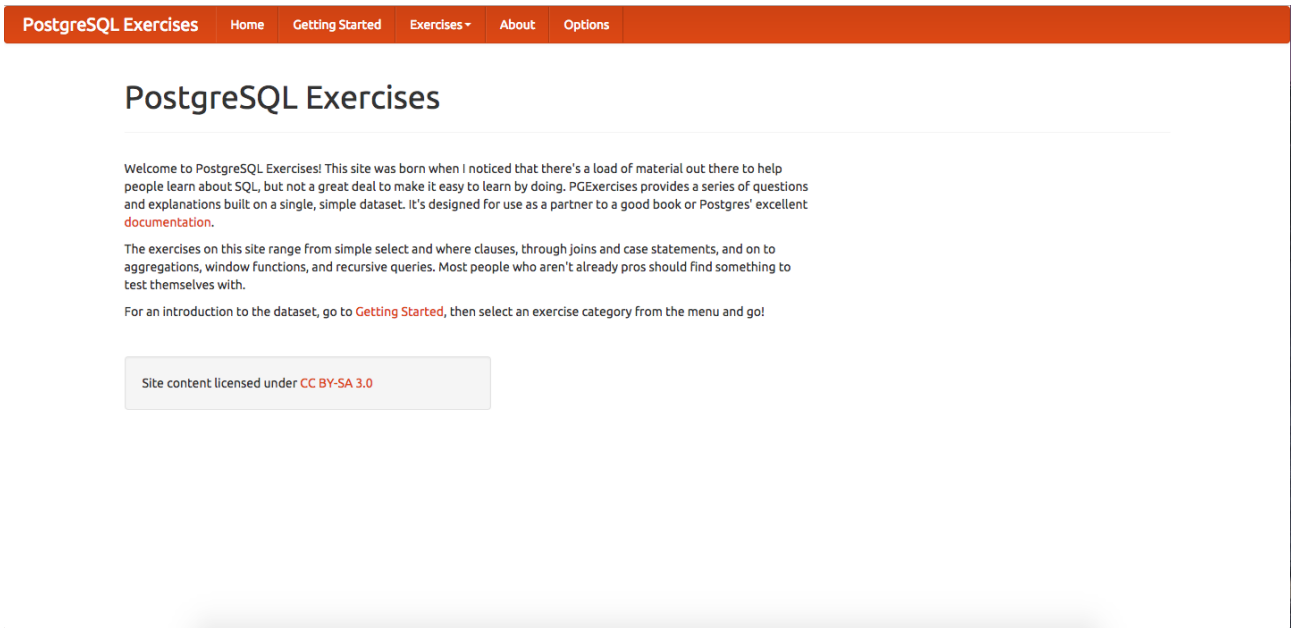


Рисунок 1.4 – Головне вікно сайту pgexercises

Цей сайт дає змогу вдосконалювати свої знання, має гарний функціонал. На вкладці «упражнения» розміщені завдання які необхідно виконати.

Рисунок 1.5 – Вибір завдання

Можна відмітити, якщо складно виповнити запит, на сторінці розміщено «намек», який полегшить роботу користувачу. [3]

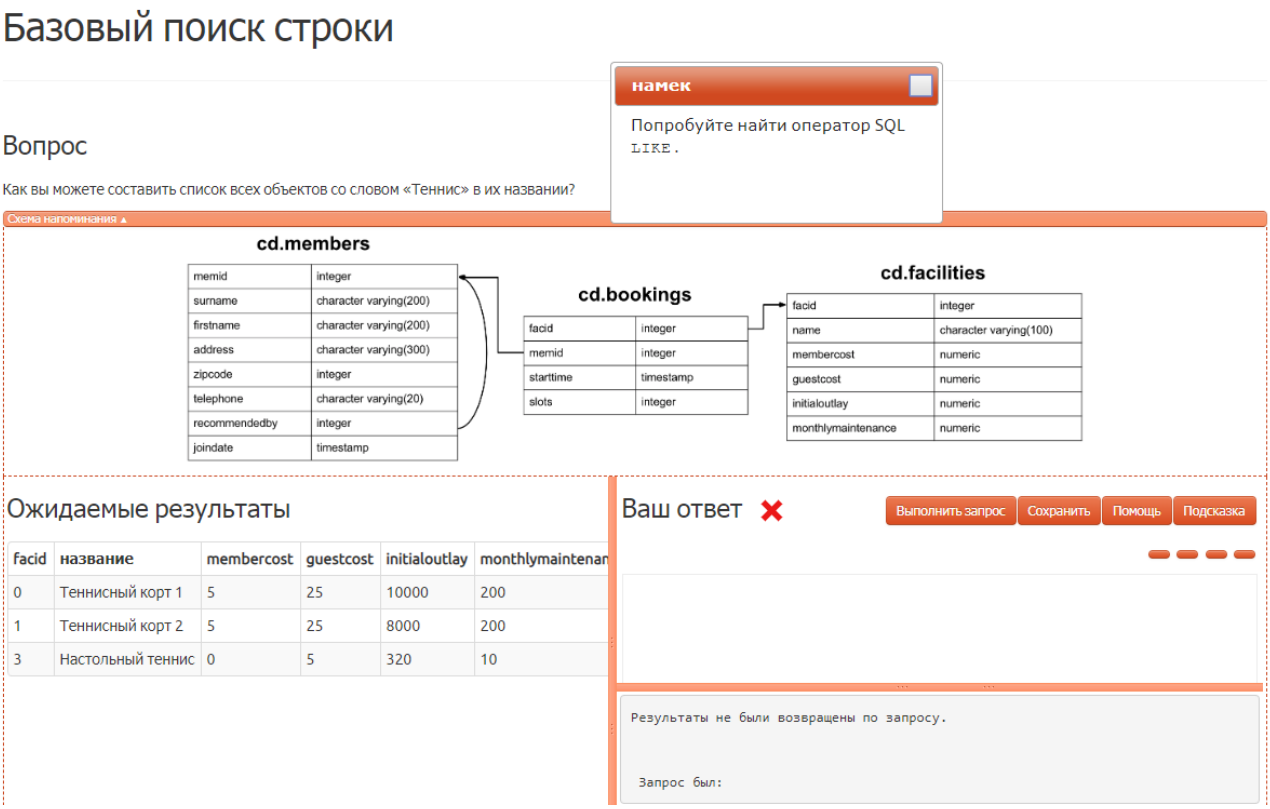


Рисунок 1.6 – Результат написання запиту

– SQLBolt – на сайті розміщений базовий курс роботи з sql-запитами, варто відмітити, що крім фіксованого курсу є можливість виконання додаткових завдань, посилання на які розміщені в футері.

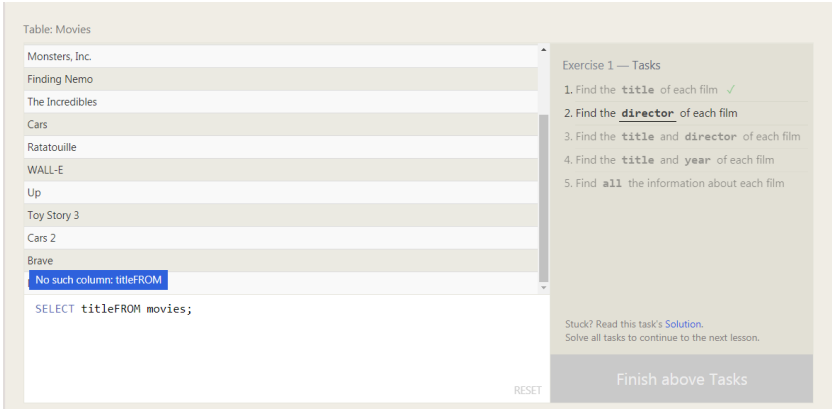


Рисунок 1.7 – Результат написання запиту

– Sqlzoo – дизайн виконано в спокійних тонах, головним плюсом слід відзначити пошук по заголовку з обраної категорії.

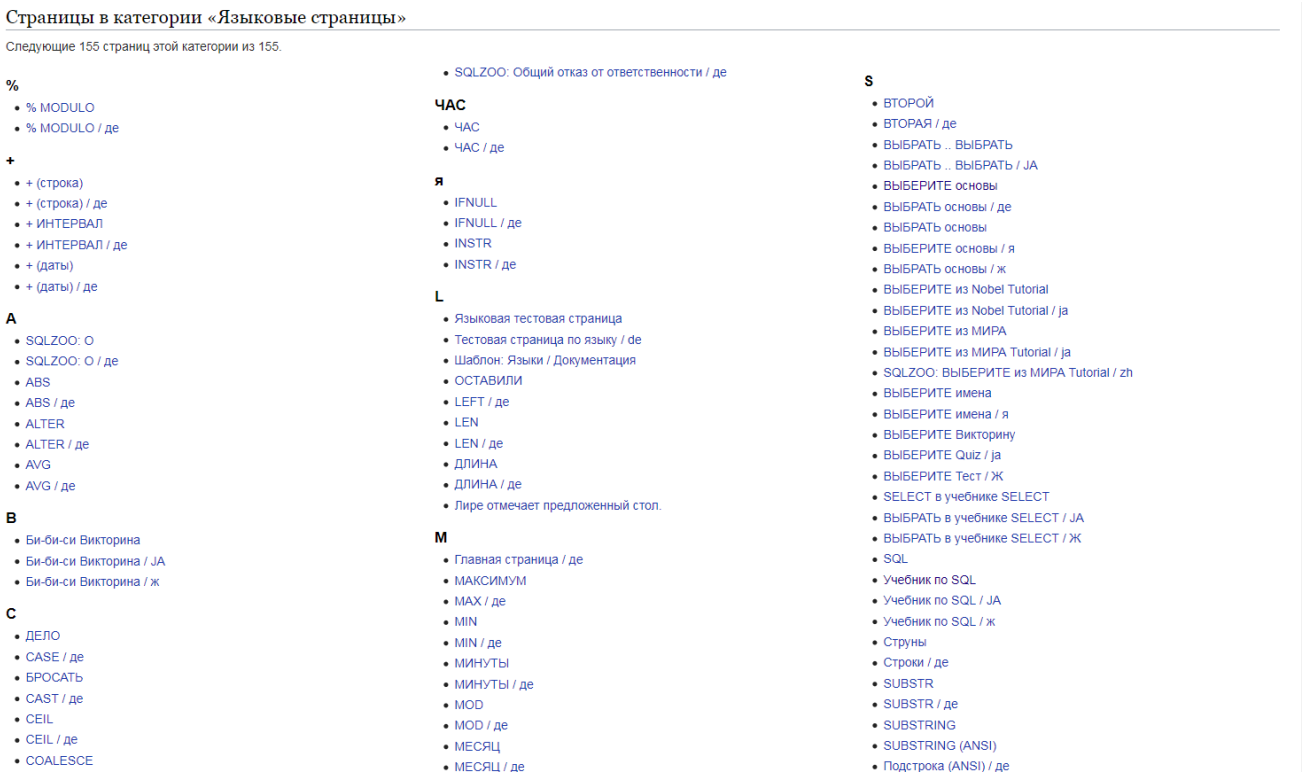


Рисунок 1.8 – Пошук з обраної категорії по заголовку

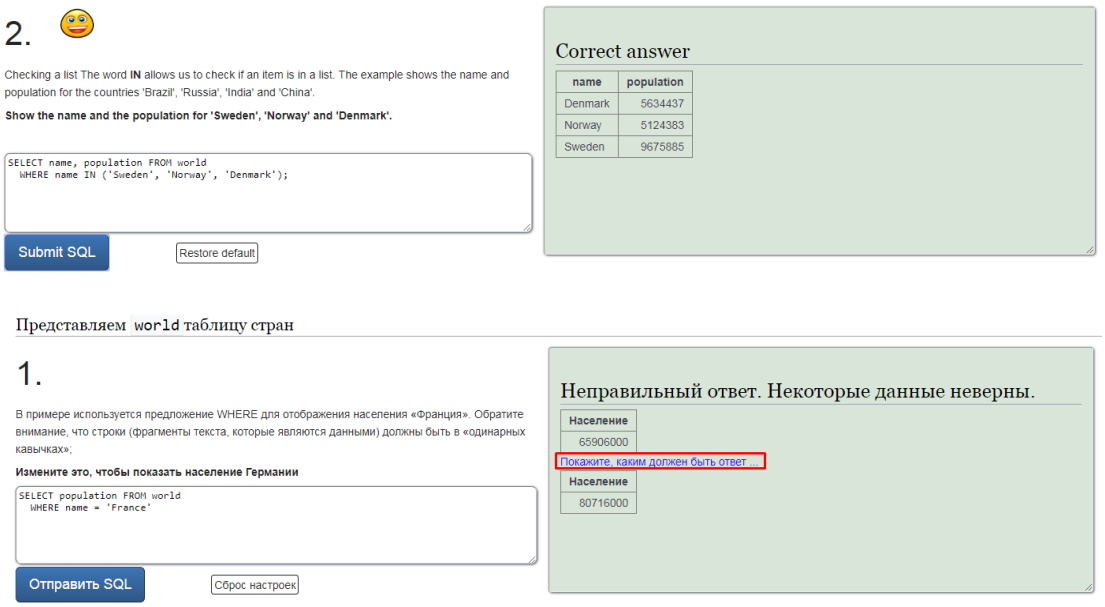


Рисунок 1.9 – Результат написання запиту

При написанні правильного запиту з’являється смайл. Є можливість переглянути правильну відповідь. [4]

Таблиця1.1 – Порівняльна таблиця сайтів-аналогів

Назва сайту	sql-tutorial	pgexercises	SQLBolt	sqlzoo	db-sql-practics
Можливість переглянути правильну відповідь	-	+	+	+	+
Обмежена кількість спроб проходження завдання	-	-	-	-	+
Підкреслення помилки в написанні запиту	+	-	+	-	-
Публічний результат роботи	+	+	-	-	-
Можливість вибору підказки	+	+	-	-	-

1.3 Постановка задачі

Головна мета дипломного проекту полягає у розробленні web-додатку автоматизованої перевірки лабораторних робіт з написання запитів до бази даних мовою SQL.

Проект повинен бути виконаний у вигляді сайту.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- Сформувати та затвердити технічне завдання;
- Провести аналіз предметної області;
- Розробити шаблон сайту;
- Створити базу даних;
- Реалізувати систему перевірки SQL-запитів;
- Провести тестування.

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

1. Зберігання результату робіт студентів;
2. Підтримка виконання завдань та збереження результатів;
3. Ведення лабораторних завдань;
4. Підтримка функцій автоматичної перевірки та виставлення оцінок;
5. Імпорт балів в EXEL;
6. Можливість ручної перевірки;
7. Особистий кабінет для кожного з користувачів.

2 МОДЕЛЮВАННЯ WEB-ДОДАТКУ

2.1 Моделювання бізнес-процесів web-додатку

Моделювання – це метод пізнання, сутью якого полягає в створенні та дослідженні спрощених замінників реальних об'єктів.[6]

Початковий обєкт прийнято називати прототипом чи оригіналом. Об’єкт-замінник – модель.

До створення моделей звертаються щоб зменшити ризики на витрати, заощадити час розробника, та знайти оптимальне рішення.

IDFE – набір взаємопов’язаних блоків, де стрілками зліва вказуються вхідні дані, справа – вихідні, зверху – управління (керуючі об’єкти), вниху - механізми. Головною метою створення IDEF-діаграм побудова ієрархічної системи.[7]

Вхідні дані перевірки і оцінки студентів:

– Логін;

– Пароль;

Вихідними даними:

– Таблиця успішності;

– Оцінки.

Керуючі об’єкти:

– Еталонні запити;

– Рівні доступу;

– Завдання;

– Вхідна БД.

Механізми:

– Адміністратор;

– Студент;

– Система.

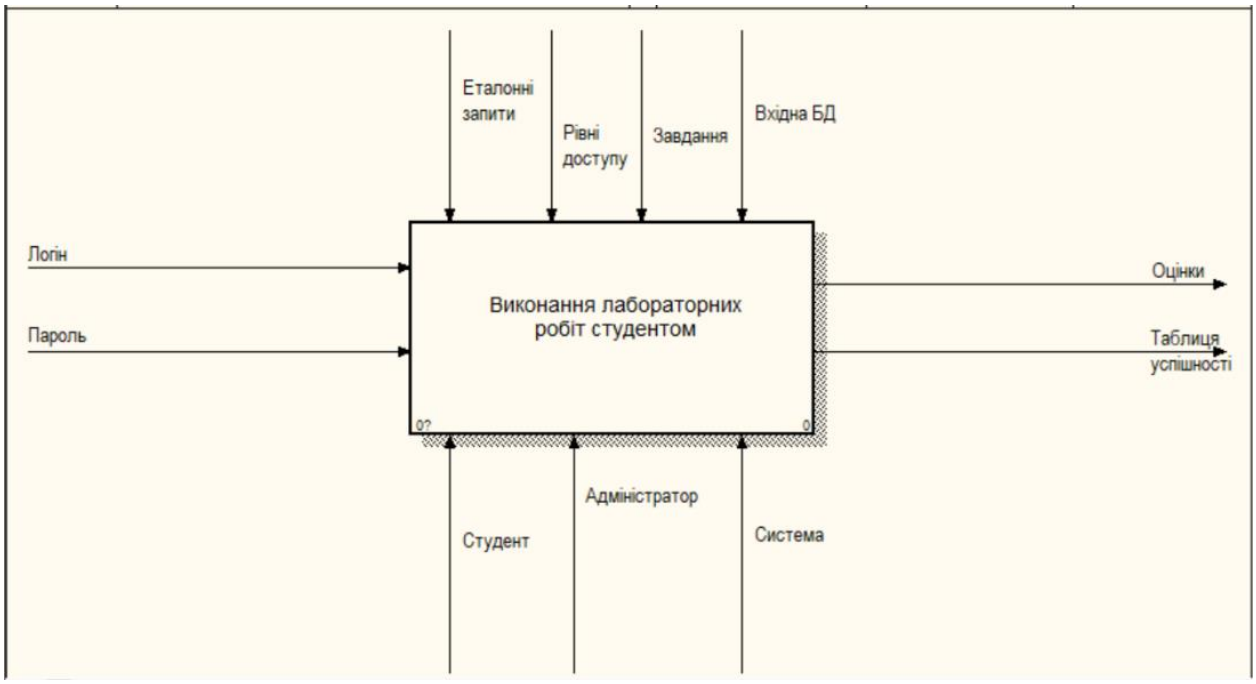


Рисунок 2.2 – Структурно-функціональне моделювання реалізації інтерактивного додатку

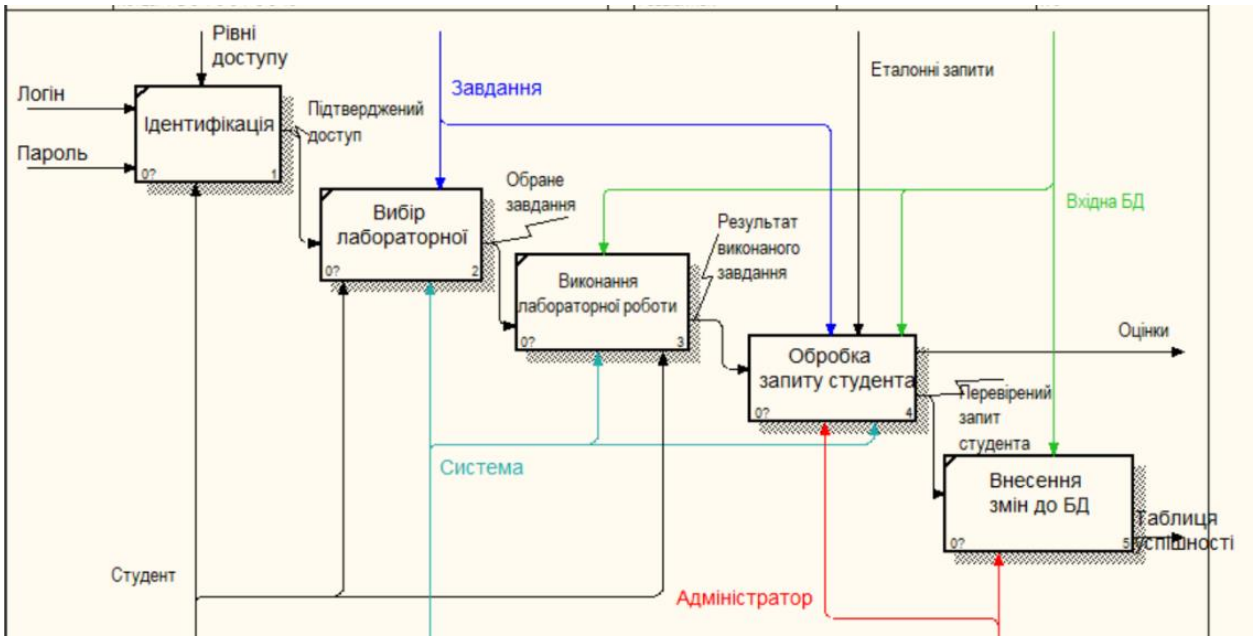


Рисунок 2.3 – Структурно-функціональне моделювання реалізації інтерактивного додатку (в IDEF0)

Будуючи діаграму декомпозиції відмічені блоки:

– Ідентифікація;

– Вибір лабораторної;
– Виконання лабораторної роботи;
– Обробка запиту студента;
– Внесення змін до БД.

Головним завданням даного проекту була реалізація обробки запитів студентів, щоб детальніше ознайомитися з даним процесом було розроблено декомпозицію обробки запиту студента (рис 2.4).

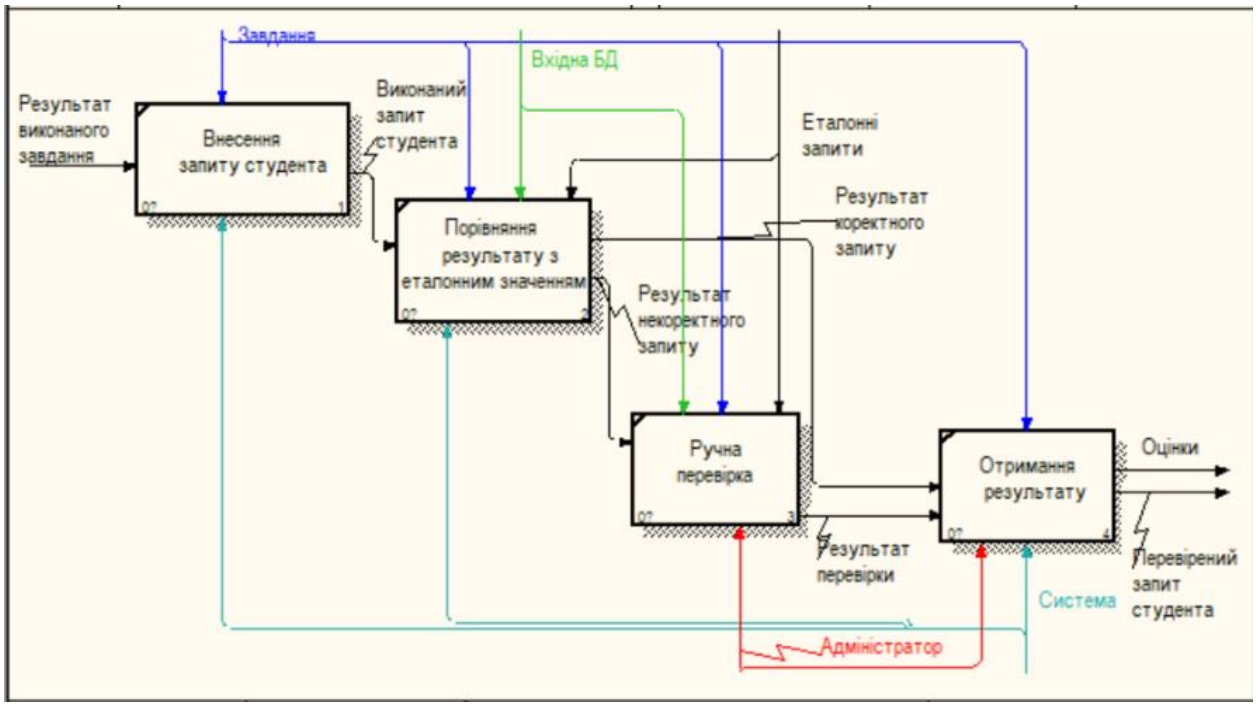


Рисунок 2.4 – Декомпозиція етапу «обробка запиту студента»

Розробляється web-додаток призначена для оцінювання практичних навичок студентів. Основні користувачі системи - студенти. За допомогою цієї системи вони можуть отримувати завдання з лабораторної роботи, переглядати оцінки, та минулі практичні. Також користувачем web-додатоку може бути адміністратор, який уповноважений вносити зміни в дані.

2.2 Моделювання варіантів використання web-додатку

Діаграма варіантів використання – це графічне зображення всіх варіантів використання, акторів, та відносини між ним [8]

Варіанти використання – це описання послідовності дій які робить система у відповідь на діяльність користувачів, або інших програмних систем. Вони відображають функціональність системи.

Актори:

Користувач – людина, яка використовую програмний продукт;

Адміністратор – людина, яка має більш розширений функціонал продукту.[11]

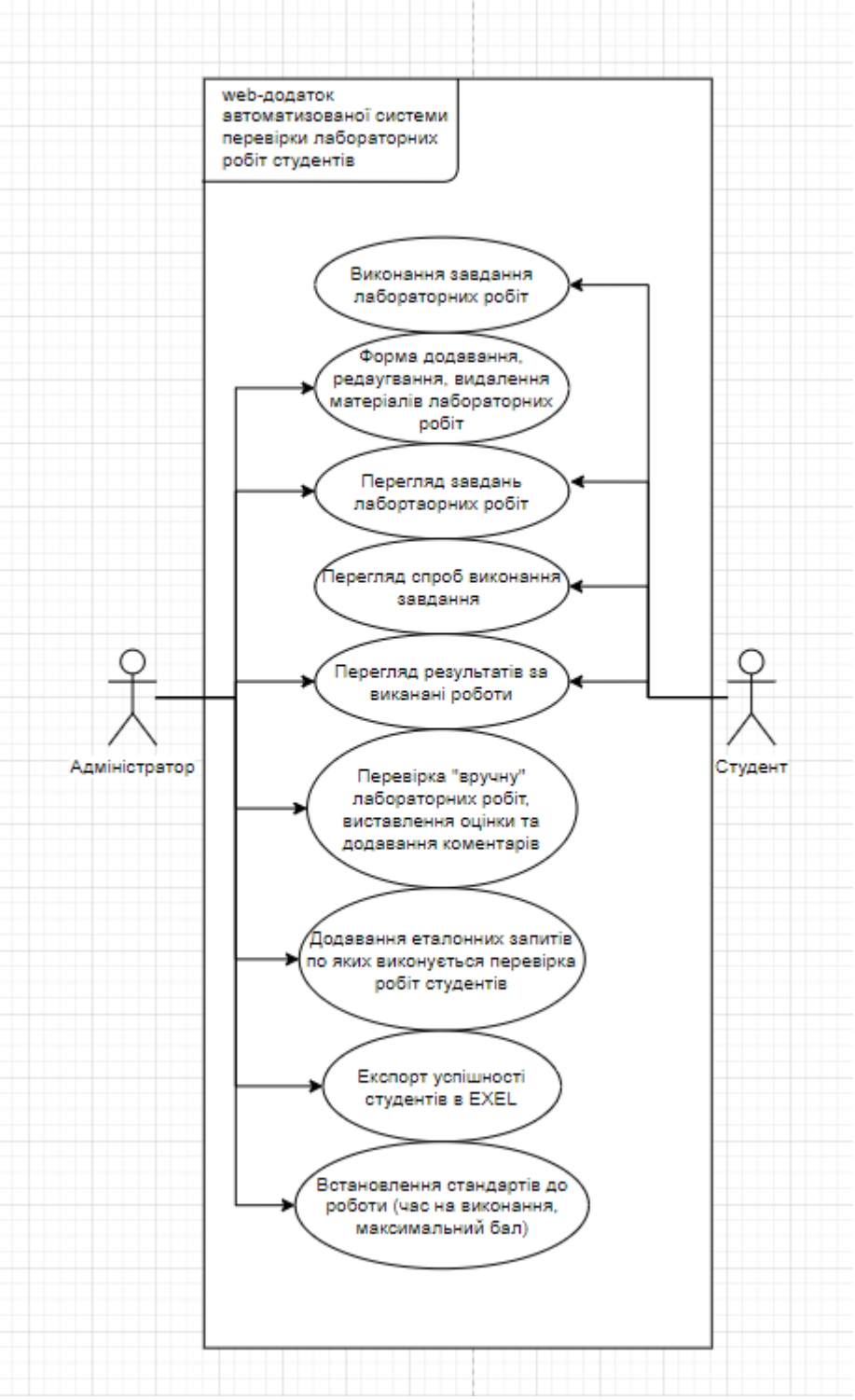


Рисунок 2.5 – Діаграма варіантів використання

У кожного актора різні варіанти використання

Студент має права:

– Переглядати результати за виконані ним роботи;
– Переглядати роботи які надсилав, та невдалі спроби;
– Переглядати завдання до лабораторних робіт.

Адміністратор має права:

– Переглядати результати за виконані студента роботи;

– Переглядати роботи які надсилав студент, та його невдалі спроби;

– Переглядати завдання до лабораторних робіт студентів;

– Редагувати, видаляти та додавати матеріали контенту;

– Перевіряти роботи вручну, виставляти оцінки самостійно та додавати коментарі до звітів;

– Додавати еталонні запити, за якими в подальшому відбувається перевірка;

– Експортувати оцінки в EXEL;

– Встановлювати стандарти на виконання роботи (час, максимальний бал).

2.3 Проектування моделі бази даних

База даних – це сховище для інформації, в якій за допомогою зв’язків поєднані таблиці, вміст таблиць, постійно змінюється, та наповнюється. Було проведено проектування бази даних згідно завдання, та її реалізація.[9]

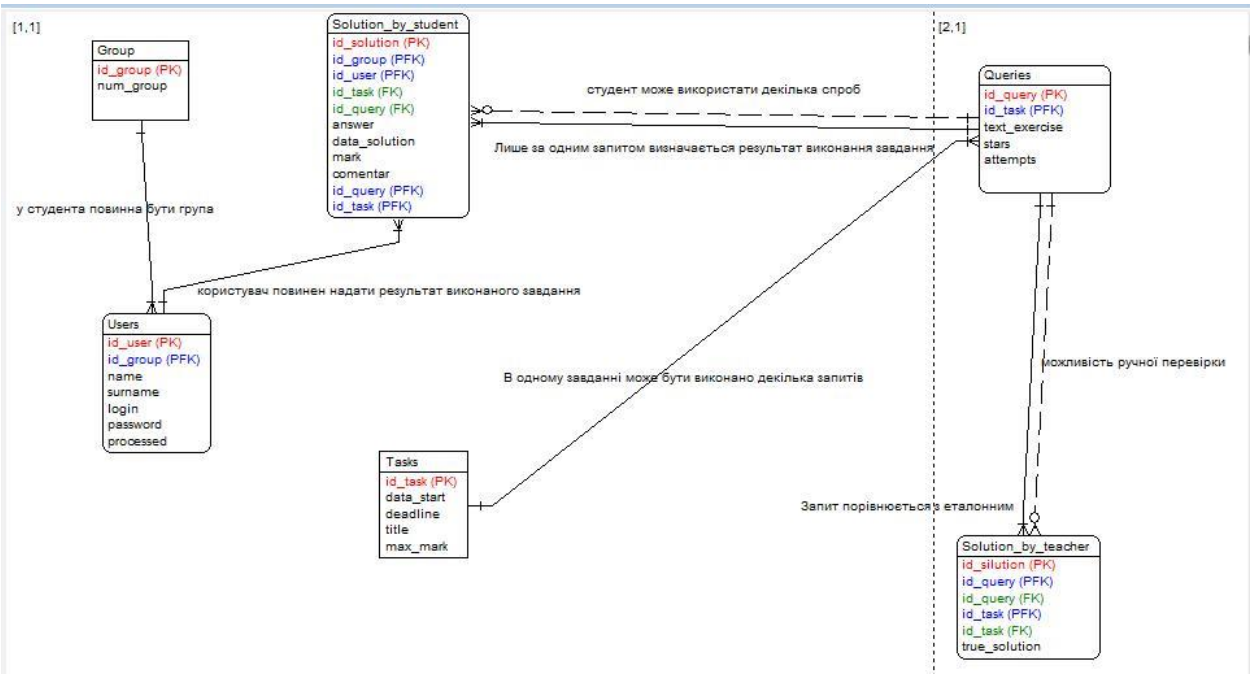


Рисунок 2.6 – Модель бази даних

Модель бази даних була розроблена за допомогою програмного забезпечення case studio. при розробці діаграми було визначено недоліки програмного забезпечення: при встановленні зв’язків між таблицями, а саме встановлення зв’язків не від сутності до сутності, а між таблицями, що провокує дублювання зв’язків. Надалі для виконання схожих робіт не рекомендовано використання даного пз.

3. РОЗРОБКА WEB-ДОДАТКУ

3.1 Архітектура web-додатку

Для коректної роботи сайту, було розроблено діаграму компонентів. В основі чітко вказані внутрішні методи по відношенню з додатком, які є ключовими в реалізації даного завдання, зв’язки між ними, та зовнішні користувачі, що використовують дані методи.

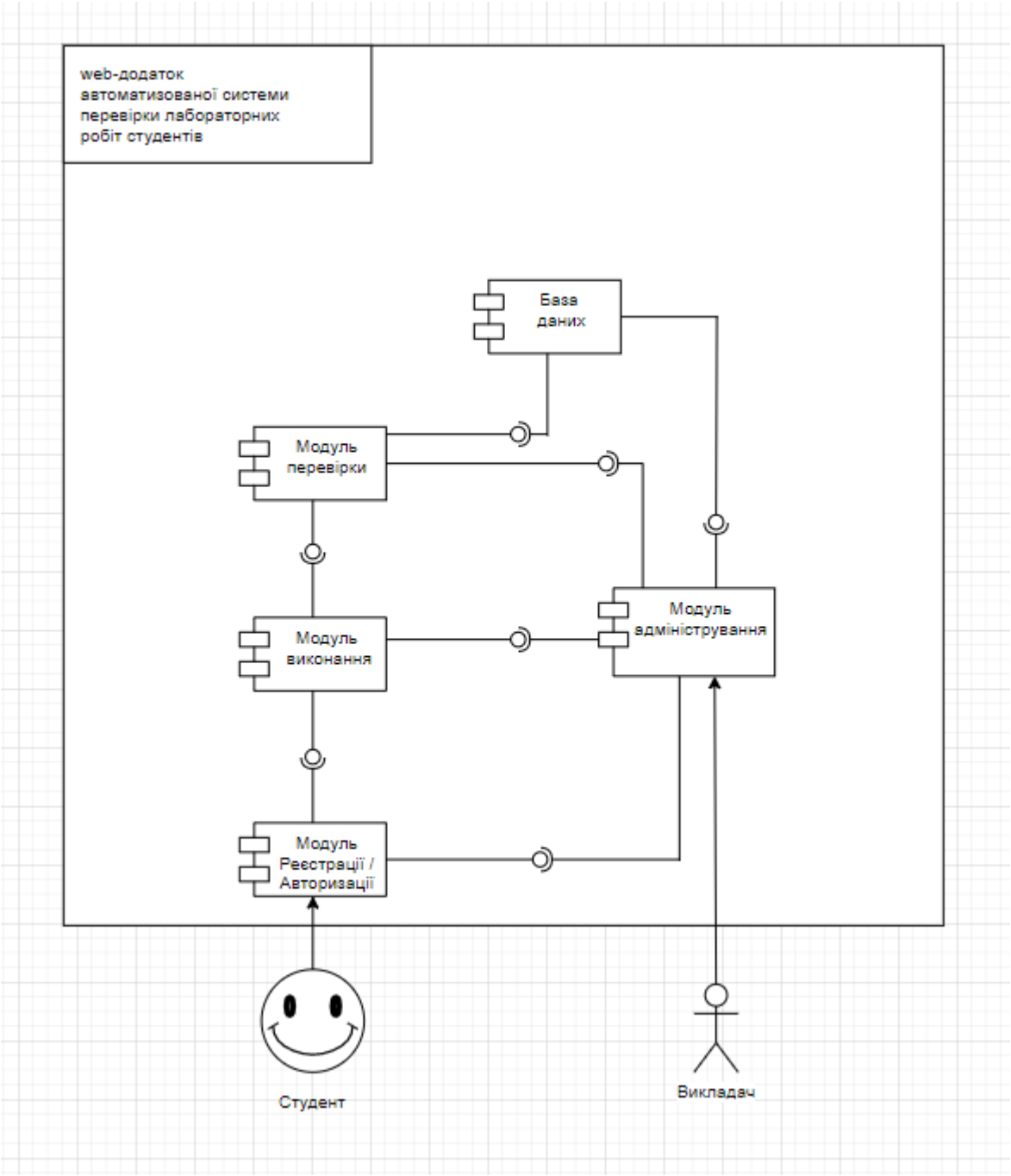


Рисунок 3.1 – Діаграма компонентів

Для демонстрації розбиття програмної системи на структурні компоненти та їх зв'язки використовують діаграму компонентів. Діаграма компонентів даного дипломного проекту складається з таких компонентів, як:

– Модуль Реєстрації/Авторизації;
– Модуль адміністрування;
– Модуль виконання;

– Модуль перевірки;

– Бази даних.

З web-додатком взаємодіють два типи користувачів: студент та викладач.

Викладач, як адміністратор взаємодіє з усіма компонентами через модуль адміністрування.

Студент одразу потрапляє на модуль реєстрації/авторизації, лише після нього має доступ до модуля виконання.

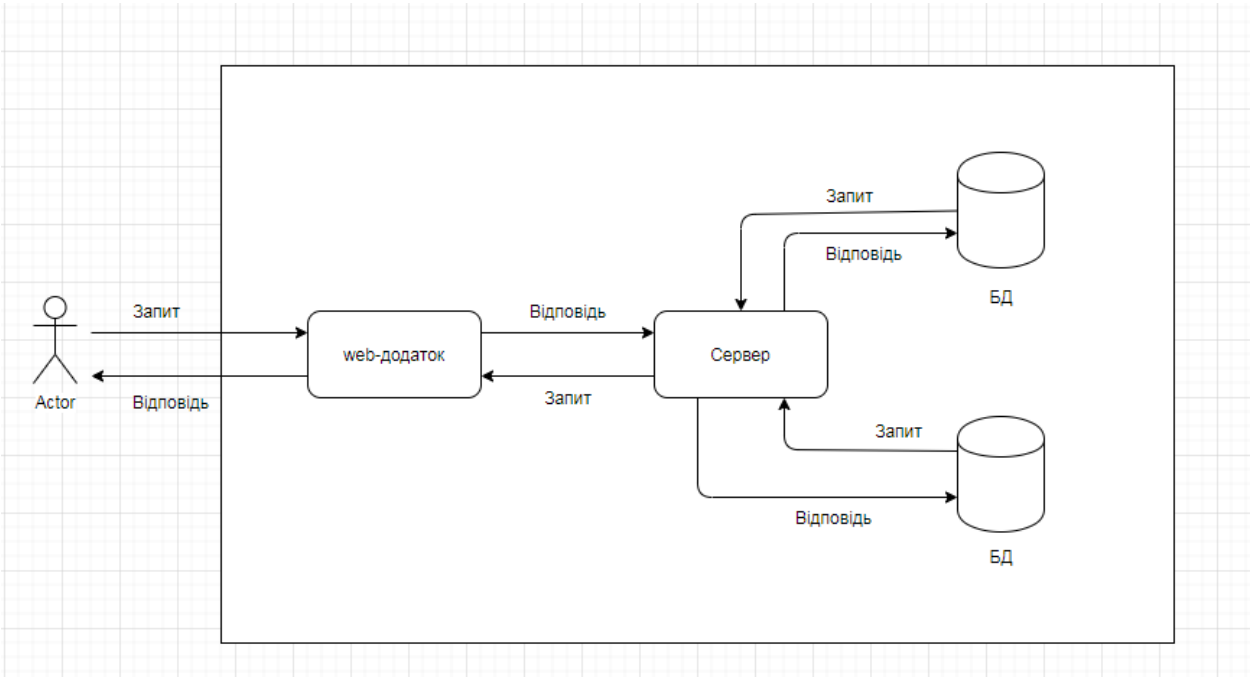


Рисунок 3.2 – Архітектура сайту

3.2 Фізична реалізація бази даних

Проаналізувавши сутність, використання в моделі web-додатку, перейдемо до реалізації структури бази даних. Для цього представимо імена необхідних таблиць, атрибутів, типів, їх призначення та обмеження (табл. 3.1).

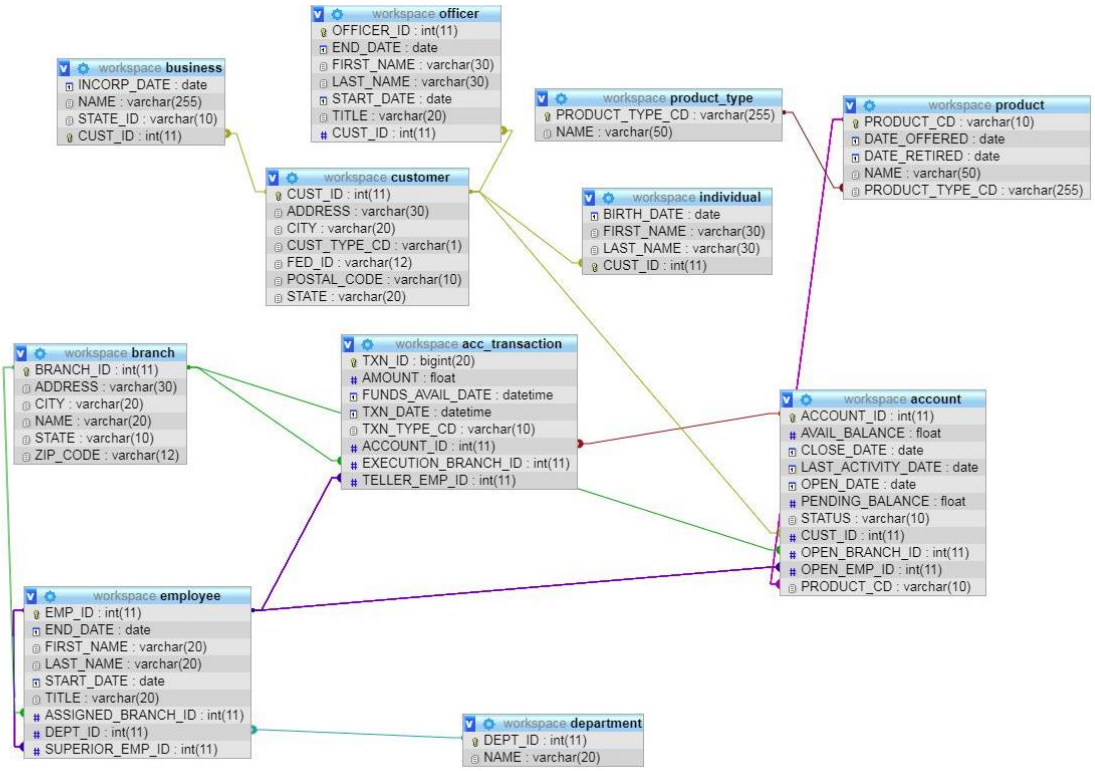


Рисунок 3.3 – Схема реалізованої бази даних для підтримки виконання запитів студентами

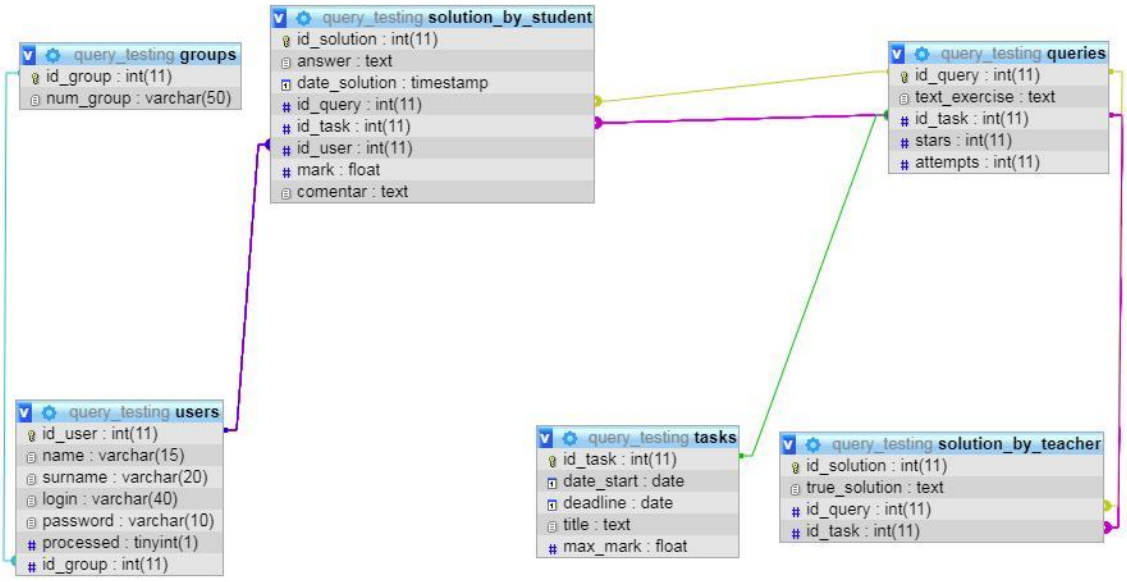


Рисунок 3.4 – Схема реалізованої бази даних для підтримки роботи веб-додатку

Таблица 3.1 - Структура бази даних

Таблиця	**Поле**	**Зміст**	**Тип**	**Ключі**	**Обмеження**
Tasks	Id_tasks	Номер задачі	Int	PK	Не пустий, Автоінкремент
	Date_start	Дата початку роботи	Date		Не пустий
	Deadline	Дата останнього дня здачі роботи	Date		Не пустий
	Title	Заголовок	Text		Не пустий
	Max_mark	Максимальна оцінка	Float		Не пустий
Users	Id_user	Номер користувача студента	Int	PK	Не пустий, Автоінкремент
	Name	Ім’я студента	Varchar2 (15)		Не пустий
	Surname	Прізвище студента	Varchar2 (20)		Не пустий
	Login	Логін студента	Varchar2 (40)		Не пустий
	Password	Пароль студента	Varchar2 (10)		Не пустий
	Id_group	Номер групи	Int	PFK	Не пустий
	Processed	Обробка заявки на реєстрацію	Tinyint (1)		Не пустий
Queries	Id_query	Номер запиту	Int	PK	Не пустий, Автоінкремент
	Text_exercise	Завдання	Text		Не пустий
	Id_task	Номер задачі	Int	PFK	Не пустий
	Stars	Кількість зірок	Int		Не пустий
	Attempts	Кількість спроб	Int		Не пустий

Продовження таблиці 3.1 Структура бази даних

Solution_by_teacher	Id_solution	Номер рішення	Int	PK	Не пустий, Автоінкремент
	True_solution	Еталонний запит	Text		Не пустий
	Id_query	Номер запиту	Int	FK	Не пустий
	Id_task	Номер задачі	Int	FK	Не пустий
Solution_by_student	Id_solution	Номер рішення	Int	PK	Не пустий, Автоінкремент
	Answer	Відповідь	Text		Не пустий
	Date_solution	Дата рішення	Time_ stamp		Не пустий
	Id_query	Номер запиту	Int	FK	Не пустий
	Id_task	Номер задачі	Int	FK	Не пустий
	Id_user	Номер користувача студента	Int	PFK	Не пустий
	Mark	Оцінка	Float		Не пустий
	Comentar	Коментар	Text		Не пустий
Group	Id_group	Номер групи	Int	PK	Не пустий, Автоінкремент
	Num_group	Назва групи	Varchar2 (50)		Не пустий

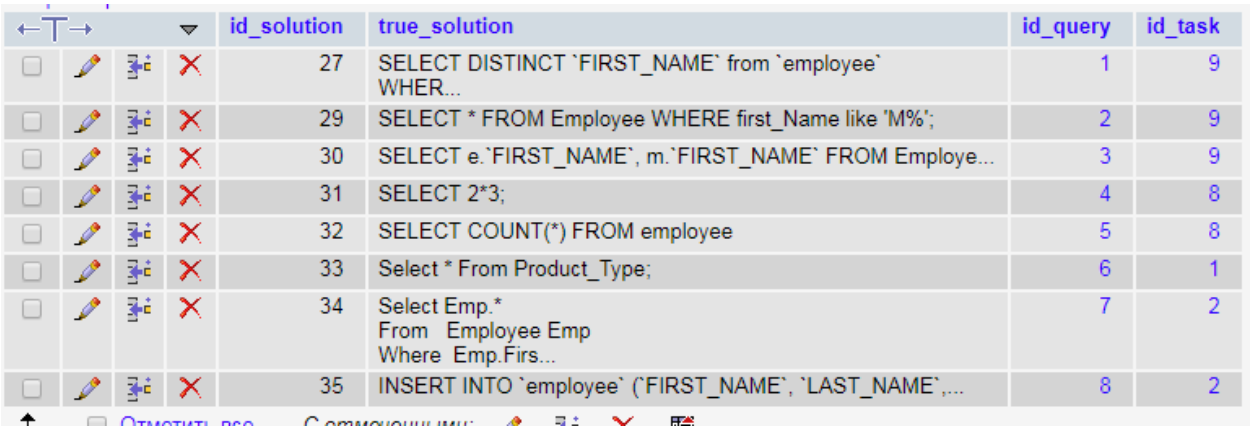
3.3 Програмна реалізація

Головною задачею даного дипломного проекту було створення інструментарію, що реалізує механізм перевірки запитів студентів з максимальною точністю, а також оцінки їх роботи. Було вирішено заносити до бази даних відповіді, отримані від запитів студента та викладача, порівнювати їх, що стало б еталоном для подальших перевірок.

```
}
if(isset($_POST['add_sub'])){
    $db=mysqli_connect("localhost","root","","query_testing");
    if(!$db){ exit();}

    $user_query=$_POST['text_query'];
    $id_task=$_SESSION['id_task'];
    $id_query=$_SESSION['id_query'];
    $mark = 0;
    $str_queries = "";
    $db1=mysqli_connect("localhost","student","mHE_UfK6$4j*GVJ","workspace");
    if(!$db1) exit();
    $str = "select st.*,q.stars from solution_by_teacher st join queries q on(q.id_query=st.id_query)
    where q.id_query=".$id_query." order by q.id_query desc limit 1";

    $query_true = mysqli_query($db,$str);
    if($row_query_true=mysqli_fetch_array($query_true, MYSQLI_ASSOC)){
        $str_queries = str_replace(";","",$row_query_true['true_solution']);
        $mark = $row_query_true['stars'];
    }
    $str_queries = $str_queries." except ".$user_query;
    //echo $str_queries;

    $coment = "";
    $name_savepoint     = "sp".date ('YmdHis', time());
    $query_autocommit   = mysqli_query($db1,"Set autocommit=0");
    $query_transaction  = mysqli_query($db1,"Start transaction");
    $query_savepoint    = mysqli_query($db1,"Savepoint sp".$name_savepoint);
    try{
        if(! ($query_autocommit & $query_transaction & $query_savepoint)){
            throw new Exception;
        }
        $query123 = mysqli_query($db1,$str_queries);
        if($query123 == false){
            throw new Exception;
        }
        else {
            if($row_query123=mysqli_fetch_array($query123, MYSQLI_ASSOC)){
                $coment = "ИТОГИ АВТОМАТИЧЕСКОЙ ПРОВЕРКИ: Есть отличия в результатах эталонного запроса и решения, присланного студентом.";
                $mark=null;
            }
```

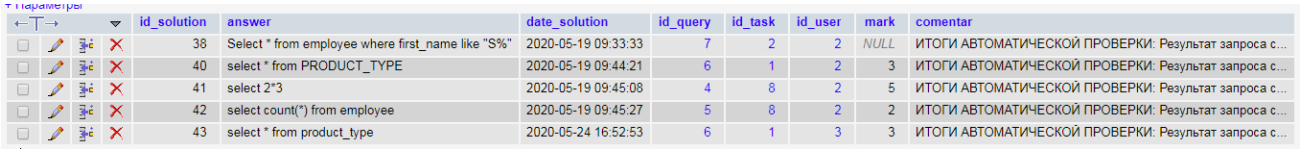
Рисунок 3.5 – Функція обробки запита студента

Після надсилання студентом завдання, відбувається його автоматична перевірка. Якщо завдання виконано правильно, студент отримує відповідне повідомлення та може переглянути отриману оцінку в розділі "Мої результати". У випадку надсилання невірно виконаного завдання, студент має можливість переробити завдання, якщо викладач заздалегідь передбачив таку можливість та визначив кількість спроб на виконання поставленого завдання. Також якщо студент надав неправильну відповідь, викладач має можливість перевірити якість виконання в ручному режимі, поставити оцінку та залишити особистий коментар. Всі ці дані заносяться до бази даних.



id_solution	true_solution	id_query	id_task
27	SELECT DISTINCT `FIRST_NAME` from `employee` WHER...	1	9
29	SELECT * FROM Employee WHERE first_Name like 'M%';	2	9
30	SELECT e.`FIRST_NAME`, m.`FIRST_NAME` FROM Employe...	3	9
31	SELECT 2*3;	4	8
32	SELECT COUNT(*) FROM employee	5	8
33	Select * From Product_Type;	6	1
34	Select Emp.* From Employee Emp Where Emp.Firs...	7	2
35	INSERT INTO `employee` (`FIRST_NAME`, `LAST_NAME`,...	8	2

Рисунок 3.6 – Структура бази даних «solution_by_titcher»



id_solution	answer	date_solution	id_query	id_task	id_user	mark	comentar
38	Select * from employee where first_name like "S%"	2020-05-19 09:33:33	7	2	2	NULL	ИТОГИ АВТОМАТИЧЕСКОЙ ПРОВЕРКИ: Результат запроса с...
40	select * from PRODUCT_TYPE	2020-05-19 09:44:21	6	1	2	3	ИТОГИ АВТОМАТИЧЕСКОЙ ПРОВЕРКИ: Результат запроса с...
41	select 2*3	2020-05-19 09:45:08	4	8	2	5	ИТОГИ АВТОМАТИЧЕСКОЙ ПРОВЕРКИ: Результат запроса с...
42	select count(*) from employee	2020-05-19 09:45:27	5	8	2	2	ИТОГИ АВТОМАТИЧЕСКОЙ ПРОВЕРКИ: Результат запроса с...
43	select * from product_type	2020-05-24 16:52:53	6	1	3	3	ИТОГИ АВТОМАТИЧЕСКОЙ ПРОВЕРКИ: Результат запроса с...

Рисунок 3.7 – Структура бази даних «solution_by_student»

Адміністратор має право вирішувати чи надавати доступ до сайту користувачу, чи ні. Такі маніпуляції слугують своєрідним захистом та зберігає конфіденційність.

```php
$query=mysqli_query($db,$str);
while ($myrow=mysqli_fetch_array($query, MYSQLI_ASSOC)){
        if ($p==1) {
            echo"<tr>";
            foreach($myrow as $in => $buf){
                if($in=='text_exercise'){ $col="Текст завдання";}
                if($in=='title'){ $col="Робота";}
                if($in=='answer'){ $col="Відповідь";}
                if($in=='id_user'){ $col="ID";}
                if($in=='name'){ $col="Студент";}
                if($in=='num_group'){
                    $col="Група";
                    echo"<th style = 'min-width: 60px'>".$col."</th>";
                    continue;
                }
                if($in=='mark'){ $col="Оцінка";}
                if($in=='stars'){ $col="MAX оцінка";}
                if($in=='comentar'){ $col="Коментар";}
                if($in=='true_solution'){ $col="Рішення";}
                echo"<th>".$in."</th>";
            }
            echo"<th>OK</th></tr>";
        }
        echo"<tr>";
        $i = $myrow['id_user'];

        foreach($myrow as $in=>$buf) {
            echo"<td>".$buf."</td>";
        }
        echo"<td><input type = 'submit' name = 'ok".$i."' value='ok' style='max-width:30px'></td></tr>";
        $p++;
        if(isset($_POST["ok"."".$i])){
            $str5="update `users` set processed = true where id_user = $i";
            //echo $str5;
            if($str5!=$_SESSION['last_add']){
                $query5=mysqli_query($db,$str5);
                $_SESSION['last_add']=$str5;
                echo"<script>location.href=location.href</script>";
                break;
            }
        }
}
```

Рисунок 3.8 – Функція підтвердження реєстрації студента

	id_user	name	surname	login	password	processed	id_group
	1	Tester	Tester	Tester	1	1	NULL
	2	Роман	Васильченко	ramzes	123	1	1
	3	Иван	Иванов	ivan	1	1	2
	4	Катя	Иванова	katya	12345678	0	1
	5	Иииигорь	Иванова	123	1	1	2
	6	Катя	ggggggg	789	1	0	1

Рисунок 3.9 – Структура бази даних «users»

Для формування звіту було розроблено можливість експорту даних успішності студентів до файлу MS EXCEL, де зберігаються дані студентів відповідно до групи. Повний програмний код представлено в Додатку В

Рисунок 3.10 – Демонстрація створення звіту успішності студента

A	B	C	D	E	F	G
№	Initials and Surname of a Student	Лаб/р №13	Лаб/р №14	Лаб.р №15	Л/р 10	Total Score
1	Васильченко Роман	0	8	0	6	14
2	Иванова Катя	0	0	0	6	0
3	gggggggg Катя	0	8	0	6	0

Рисунок 3.11 – Демонстрація звіту успішності студента

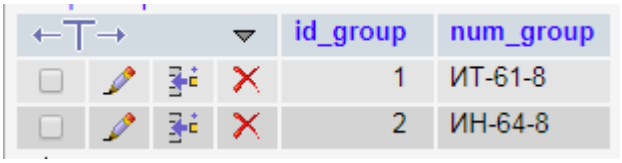


Рисунок 3.12 – Структура бази даних «group»

Можливість управління визначеними процесами адміністратору надається за допомогою адміністративної панелі. До таких процесів належать змінення, видалення, редагування, додавання, встановлення терміну виконання робіт та інше. Викладач задає максимум балів за всю лабораторну і кількість зірок за завдання. Щоб визначити частку балів за один запит від всього завдання, підсумовуються всі отримані зірки, дана сума ділиться на максимальну кількість зірок одного завдання, помножується на 100, та отримуємо відсоток. Повний програмний код представлено в Додатку В.

Роботи:

Номер роботи	Дата початку	Дата кінця	Заголовок	MAX бал
1	2020-05-01	2020-05-31	Л/р 10	6
2	2019-01-03	2019-01-14	Л/р 9	7
3	2019-01-15	2019-01-30	Л/р 11	5
4	2019-02-01	2019-02-10	Л/р 12	5.4
6	2019-02-15	2019-02-23	ОДЗ	25
7	2020-01-01	2020-01-07	Лаб/р №13	10
8	2020-01-08	2020-01-19	Лаб/р №14	8
9	2020-01-09	2020-02-01	Лаб.р №15	10

Завдання:

Номер запиту	Текст завдання	Номер роботи	Складність	MAX спроб
1	Напишите SQL запрос, чтобы напечатать имя отдельного работника у которого поле START_DATE между первым числом марта 2004 года и последним числом августа 2004 года.	9	(3) ★★★	2
2	Напишите SQL запрос, чтобы найти сотрудника, чья зарплата равна или превышает 20000	9	(1) ★	3
3	Найти имя руководителя и подчиненного.	9	(2) ★★	3
4	Получить результат умножения 2 на 3.	8	(5) ...	5
5	Получить количество всех сотрудников.	8	(2) ★★	2
6	Выбрать все типы продуктов	1	(3) ★★★	3
7	Выбрать всё о работниках, чье имя начинается на S	2	(5) ...	3
8	Добавить нового сотрудника по имени Роман Васильченко, дата приема на работу = начало 2020-го года. Остальное не заполнять.	2	(3) ★★★	2

Рисунок 3.13 – Демонстрація створення завдань та лабораторних робіт

id_query	text_exercise	id_task	stars	attempts
1	Напишите SQL запрос, чтобы напечатать имя отдельно...	9	3	2
2	Напишите SQL запрос, чтобы найти сотрудника, чья з...	9	1	3
3	Найти имя руководителя и подчиненного.	9	2	3
4	Получить результат умножения 2 на 3.	8	5	5
5	Получить количество всех сотрудников.	8	2	2
6	Выбрать все типы продуктов	1	3	3
7	Выбрать всё о работниках, чье имя начинается на S	2	5	3
8	Добавить нового сотрудника по имени Роман Васильче...	2	3	2

Рисунок 3.14 – Структура бази даних «queries»

id_task	date_start	deadline	title	max_mark
1	2020-05-01	2020-05-31	Л/р 10	6
2	2019-01-03	2019-01-14	Л/р 9	7
3	2019-01-15	2019-01-30	Л/р 11	5
4	2019-02-01	2019-02-10	Л/р 12	5.4
6	2019-02-15	2019-02-23	ОДЗ	25
7	2020-01-01	2020-01-07	Лаб/р №13	10
8	2020-01-08	2020-01-19	Лаб/р №14	8
9	2020-01-09	2020-02-01	Лаб.р №15	10

Рисунок 3.15 – Структура бази даних «tasks»

3.4 Використання web-додатку

Для звичайного користувача взаємодія з таким складним програмним об'єктом викликає певні проблеми. Тому для кращого розуміння та коректного використання сайту було вирішено створити інструкції користувача як для студента, так і для викладача.

3.4.1 Інструкція користування додатком для студента

У першу чергу студент потрапляє на сторінку авторизації. Якщо студент ще не має особистого кабінету, йому потрібно зареєструватися на сайті, для чого необхідно натиснути кнопку «Реєстрація». Далі користувачу необхідно правильно заповнити відповідні поля введення та натиснути кнопку «Далі» та очікувати, поки викладач не схвалить його заявку на реєстрацію. Якщо користувач при авторизації записав до поля введення некоректні дані або взагалі не заповнив поле, то отримує відповідне повідомлення про виникнення помилки (рис.3.16 – 3.19).

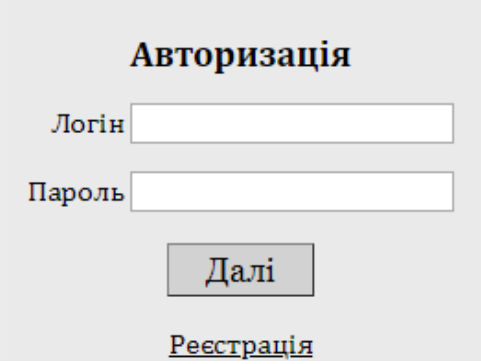


Рисунок 3.16 – Форма авторизації

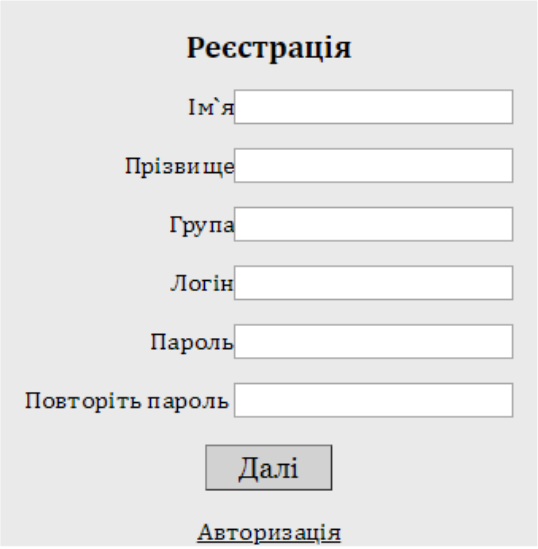


Рисунок 3.17 – Форма реєстрації

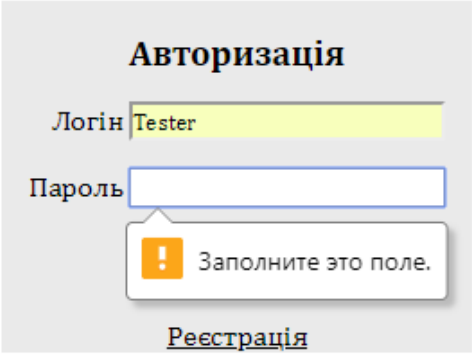


Рисунок 3.18 – Форми авторизації з повідомленням про незаповнене поле

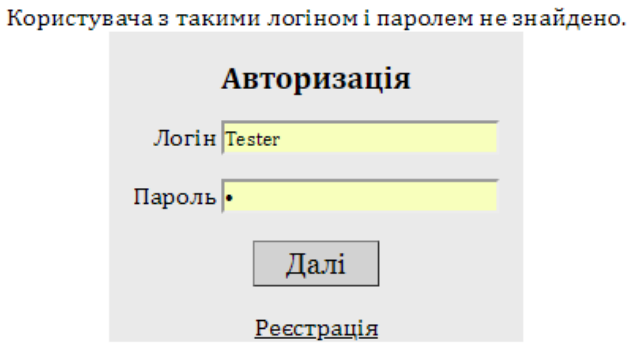


Рисунок 3.19 – Форми авторизації з повідомленням про помилку

Після успішної авторизації студент переходить до головної сторінки сайту, де він має можливість обрати з наявного списку роботу, яку він хоче виконати (рис.3.20).

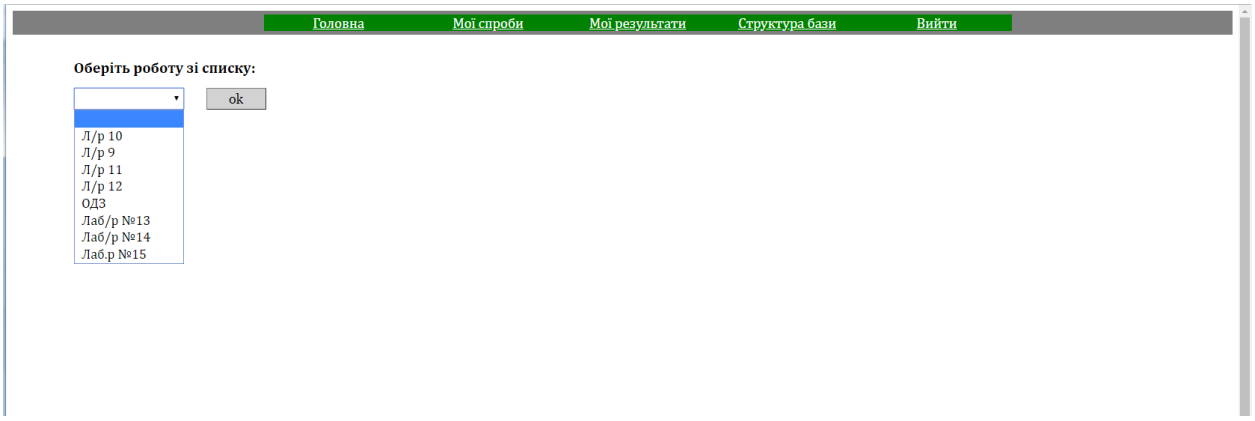


Рисунок 3.20 – Демонстрація головної сторінки студента

Обравши відповідну роботу, студент отримує певний перелік завдань для виконання та кількість спроб для кожного завдання, далі необхідно натиснути на потрібне завдання та натиснути кнопку «Ок» (рис.3.21).

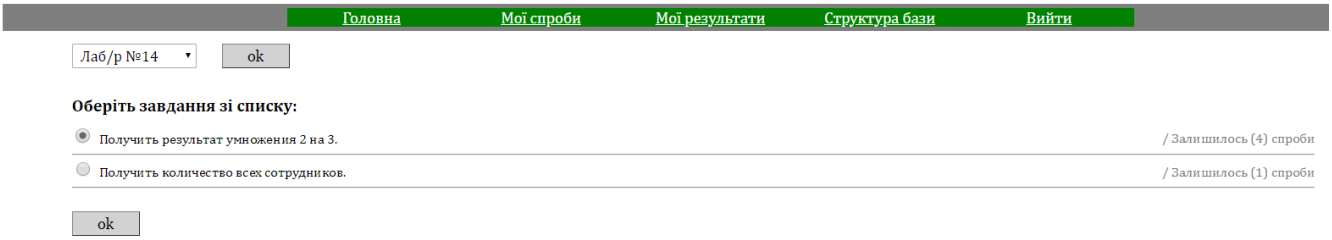


Рисунок 3.21 – Демонстрація головної сторінки студента

Після того, як студент вибере відповідне завдання, йому необхідно написати запит згідно з завданням та натиснути кнопку «Відправити відповідь». Якщо студент не знає відповіді на дане завдання, то має можливість обрати інше, натиснувши кнопку «Обрати інше завдання» (рис.3.22).

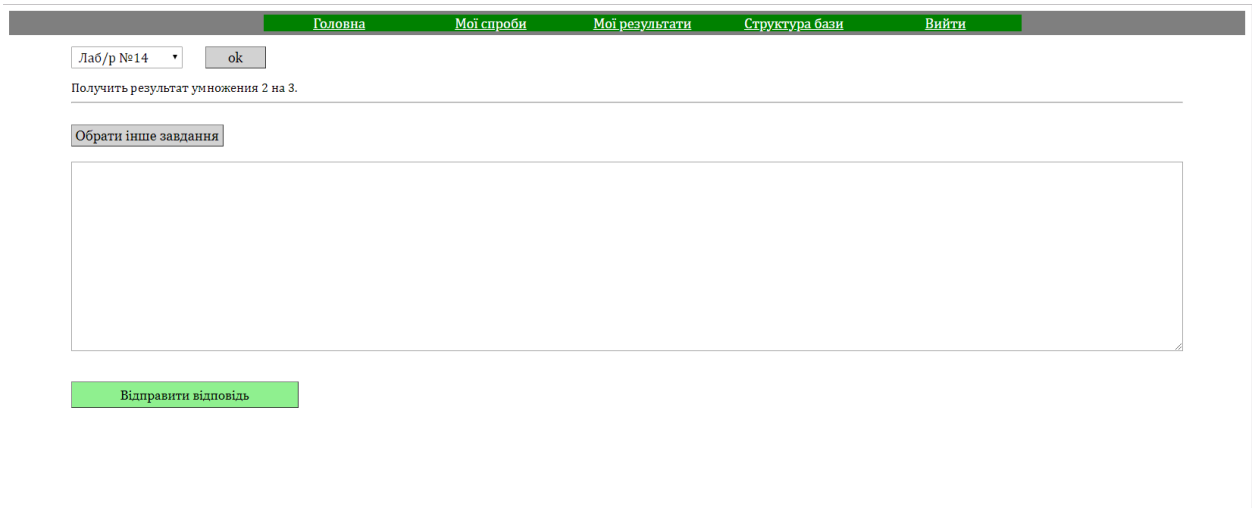


Рисунок 3.22 – Демонстрація форми виконання завдання

У випадку виконання студентом невірного запиту, він отримає відповідне повідомлення про виникнення помилки. У такому випадку студент має можливість повторно виконати запит, якщо це передбачено наявною кількістю спроб на виконання завдання (рис.3.23).

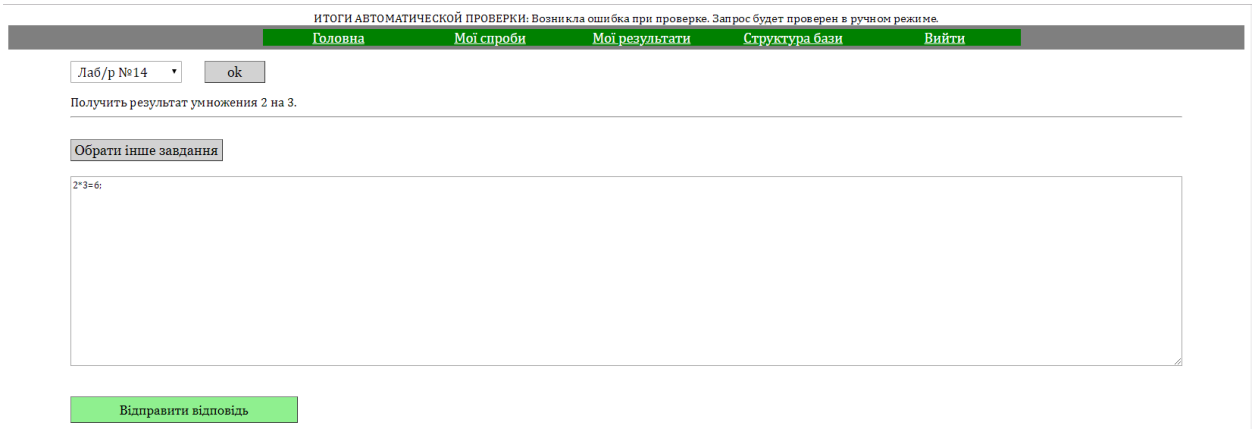


Рисунок 3.23 – Повідомлення про помилкове виконання завдання

Студент має можливість переглянути результати всіх своїх спроб виконання завдання на сторінці головного меню «Мої спроби» (рис.3.24).

Рисунок 3.24 – Демонстрація роботи сторінки «Мої спроби»

На сторінці «Мої результати» студент може відслідковувати свою успішність за конкретну роботу (рис.3.25).

Рисунок 3.25 – Демонстрація роботи сторінки «Мої результати»

Потрібна інформація про базу даних, відповідно до якої студентами виконуються завдання, розміщена на сторінці «Структура бази» (рис.3.26).

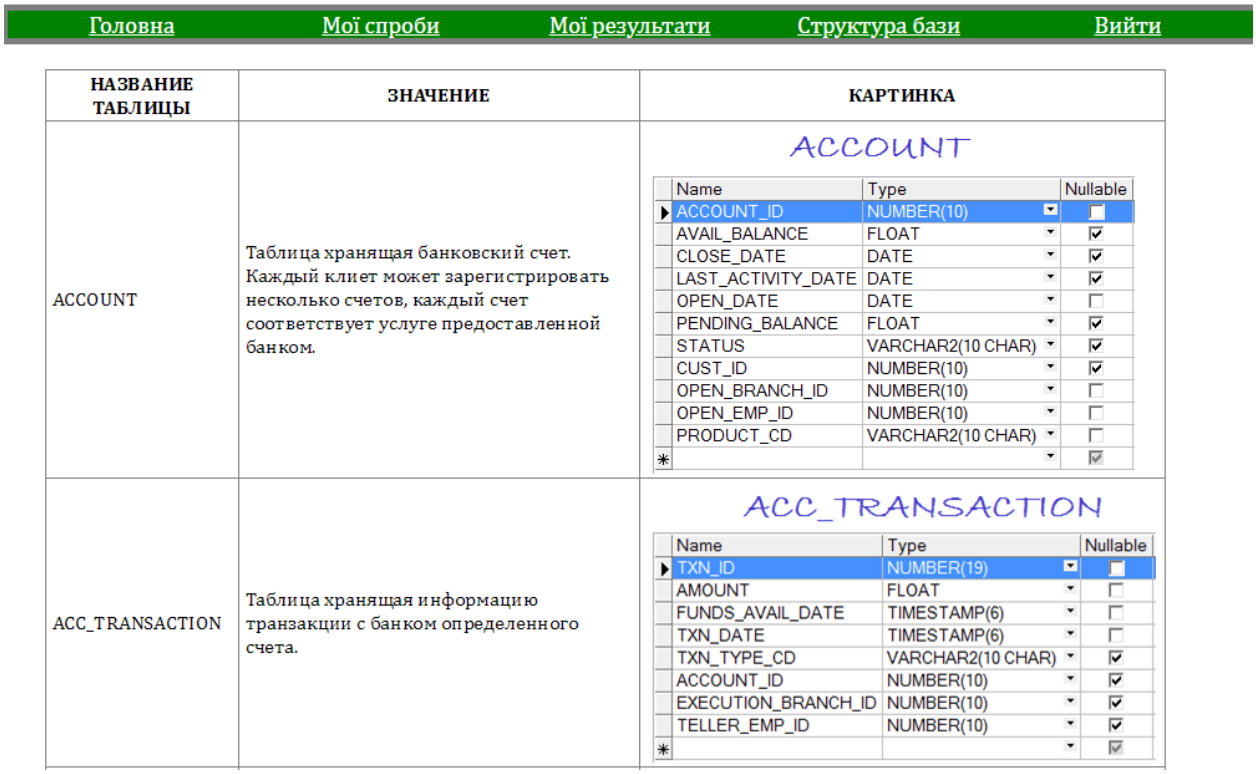


Рисунок 3.26 – Демонстрація роботи сторінки «Структура бази»

Для виконання виходу з особистого кабінету потрібно натиснути кнопку «Вихід» (рис.3.27).

Рисунок 3.27 – Кнопка «Вийти»

3.4.2 Інструкція користування додатком для викладача

Після авторизації викладач потрапляє на головну сторінку, де знаходиться адміністративна панель, а також таблиці, що містять в собі інформацію про роботи та завдання. Для того, щоб додати нову роботу для виконання викладачу необхідно заповнити форму з відповідною інформацією про неї (дата початку, дата кінця) та натиснути кнопку «Додати» (рис.3.28).

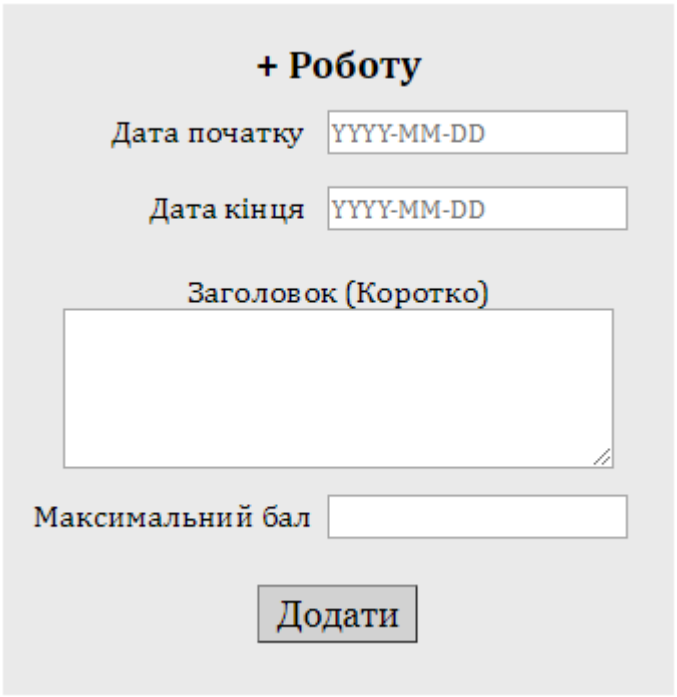


Рисунок 3.28 – Форма додавання роботи

Після створення нової роботи, її вміст додається до таблиці «Роботи» (рис.3.29).

Роботи:

Номер роботи	Дата початку	Дата кінця	Заголовок	MAX бал
1	2020-05-01	2020-05-31	Л/р 10	6
2	2019-01-03	2019-01-14	Л/р 9	7
3	2019-01-15	2019-01-30	Л/р 11	5
4	2019-02-01	2019-02-10	Л/р 12	5.4
6	2019-02-15	2019-02-23	ОДЗ	25
7	2020-01-01	2020-01-07	Лаб/р №13	10
8	2020-01-08	2020-01-19	Лаб/р №14	8
9	2020-01-09	2020-02-01	Лаб.р №15	10

Рисунок 3.29 – Вміст таблиці робіт

Для додавання нового завдання до обраної роботи необхідно ввести номер роботи, у форму «Текст завдання» необхідно внести назву завдання, а у форму «Правильна відповідь» занести еталонний запит без «;», вказавши

кількість зірок за завдання та максимальну кількість спроб на виконання даного завдання (рис.3.30).

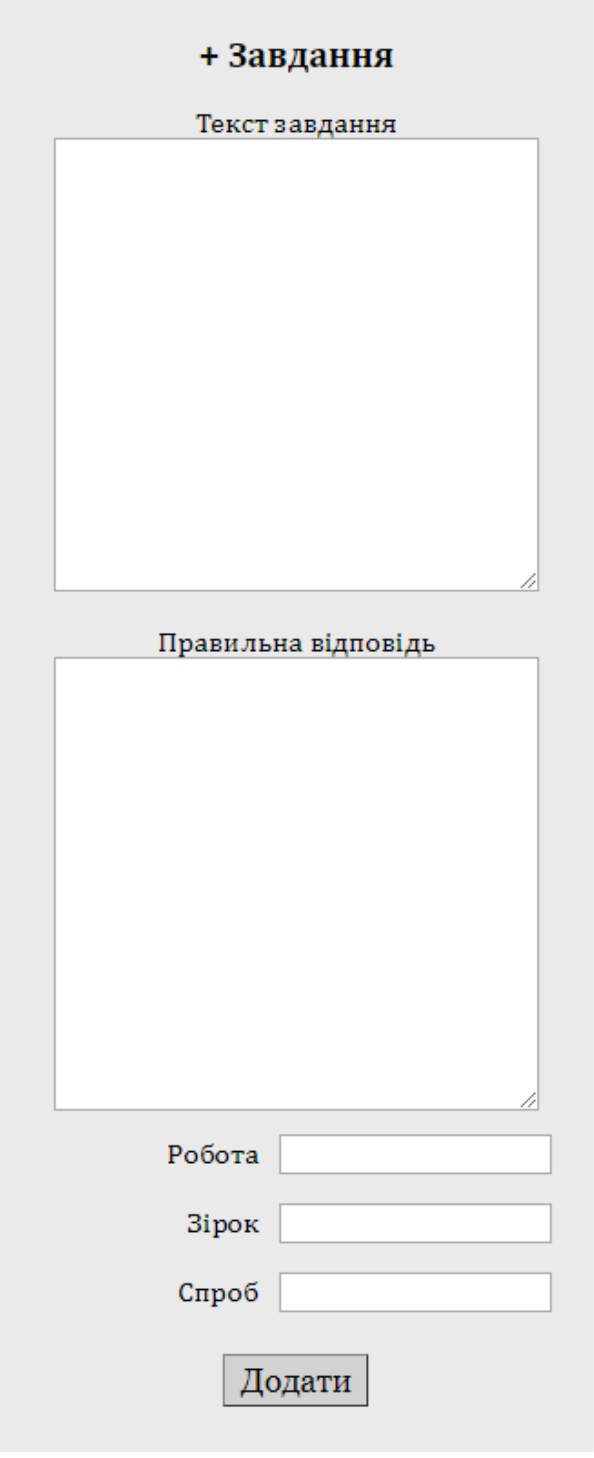


Рисунок 3.30 – Форма додавання завдання

Після додавання завдання, вся необхідна інформація зберігається в таблиці «Завдання» (рис.3.31).

Завдання:

Номер запиту	Текст завдання	Номер роботи	Складність	MAX спроб
1	Напишите SQL запрос, чтобы напечатать имя отдельного работника у которого поле START_DATE между первым числом марта 2004 года и последним числом августа 2004 года.	9	(3) ★★★	2
2	Напишите SQL запрос, чтобы найти сотрудника, чья зарплата равна или превышает 20000	9	(1) ★	3
3	Найти имя руководителя и подчиненного.	9	(2) ★★	3

Рисунок 3.31 – Вміст таблиці робіт

Для того, щоб змінити зміст обраної роботи, необхідно ввести номер даної роботи у відповідне поле та натиснути кнопку «Змінити», після цього відредагувати роботу та натиснути кнопку «Зберегти зміни» (рис.3.32).

Змінити роботу

Номер 1

Змінити

Дата початку 2020-05-01

Дата кінця 2020-05-31

Заголовок (Коротко)

Л/р 10

Максимальний бал 6

Зберегти зміни

Рисунок 3.32 – Форма редагування роботи

Для того, щоб змінити зміст обраного завдання, необхідно ввести номер даного завдання у відповідне поле та натиснути кнопку «Змінити», після цього відредагувати завдання та натиснути кнопку «Зберегти зміни» (рис.3.33).

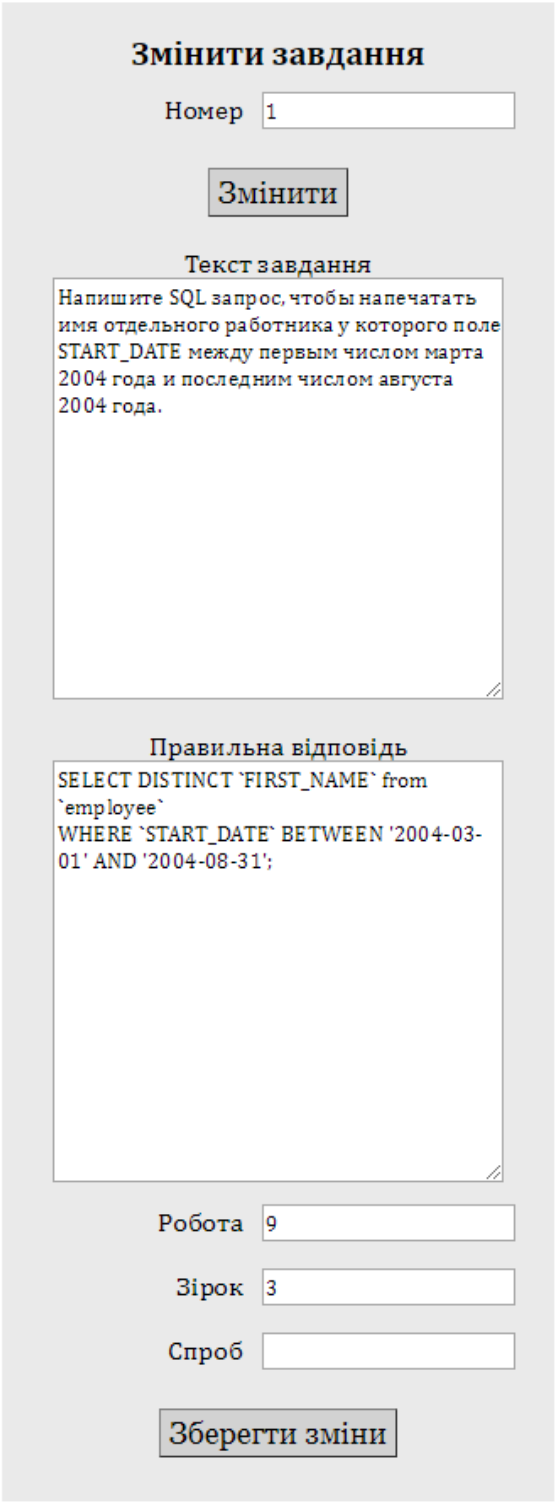


Рисунок 3.33 – Форма редагування завдання

Для видалення роботи необхідно ввести її номер у відповідне поле та натиснути кнопку «Видалити» (рис.3.34).

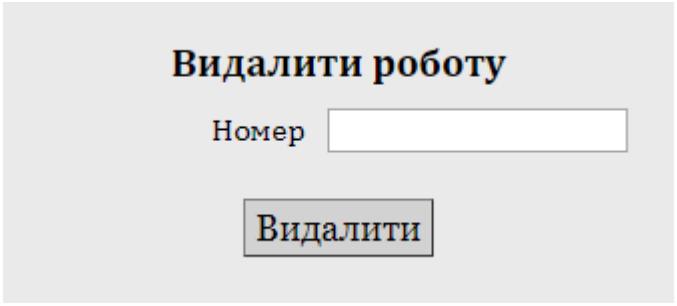


Рисунок 3.34 – Форма видалення роботи

Для видалення завдання необхідно ввести її номер у відповідне поле та натиснути кнопку «Видалити» (рис.3.35).

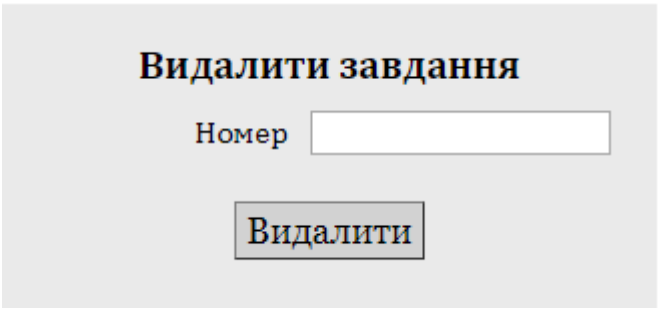


Рисунок 3.35 – Форма видалення роботи

На сторінці «Нові користувачі» викладач має можливість підтвердити реєстрацію студента натиснувши кнопку «Ок» (рис.3.36).

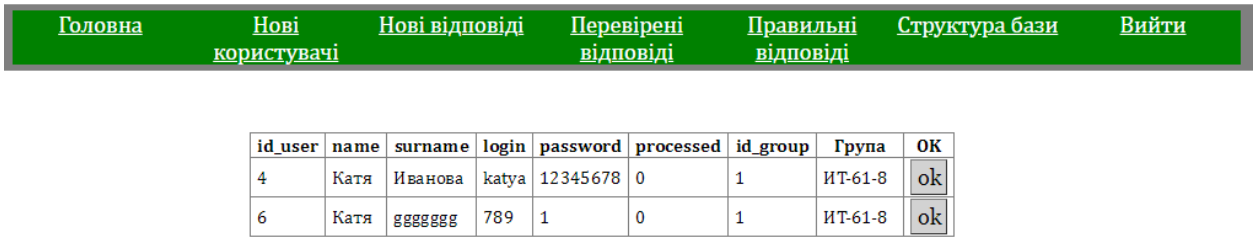


Рисунок 3.36 – Демонстрація роботи сторінки «Нові користувачі»

На сторінці «Нові відповіді» викладачу надається перелік відповідей студентів, які можна оцінити вручну, ввівши у відповідну комірку оцінку. Також викладач може залишити особистий коментар до відповіді студента, для збереження змін необхідно натиснути кнопку «Ок». (рис.3.37).

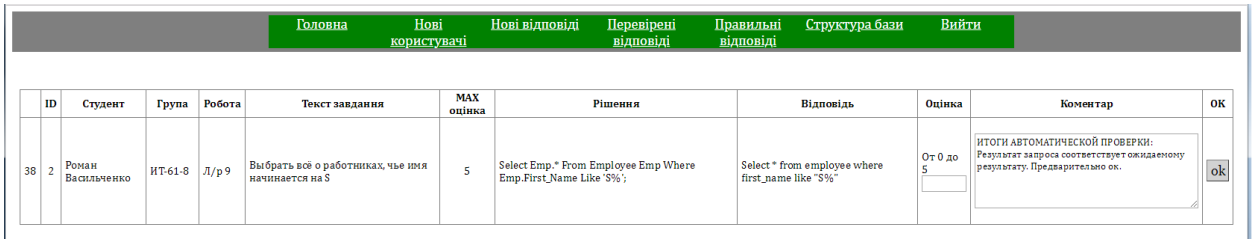


Рисунок 3.37 – Демонстрація роботи сторінки «Нові відповіді»

На сторінці «Перевірені відповіді» зберігаються всі роботи студентів. Також викладач має можливість за допомогою кнопки «Експорт оцінок в Excel» виконати виведення даних до файлу Excel відповідно до обраного року. Після чого буде створено файл з форматом .xls (рис.3.38).

Рисунок 3.38 – Демонстрація роботи сторінки «Перевірені відповіді»

На сторінці «Правильні відповіді» зберігається відповідна інформація, що включає в себе текст завдання та еталонний запит (рис.3.39) .

Рисунок 3.39 – Демонстрація роботи сторінки «Правильні відповіді»

Викладачу так як і студенту надається можливість переглядати структуру бази даних та здійснити вихід з особистого кабінету (рис.3.26 – 3.27) .

ВИСНОВКИ

У ході розробки дипломного проекту було реалізовано web-додаток, що надає можливості автоматизованої перевірки sql-запитів, перевірки студентом кількості набраних балів, обмеження кількості спроб на виконання обраного завдання, також обмежує кількість спроб на їх виконання. Крім того, було розроблено адміністративну панель, за допомогою якої, викладач просто та зручно завантажує нові завдання.

Всі поставлені завдання були вирішені в процесі розробки web-додатку

Аналіз предметної області дозволив визначити функціональні вимоги та особливості реалізації додатку.

Під час проектування даної роботи були розроблені діаграми бізнес-процесів в нотації IDEF0, діаграми варіантів використання, моделі даних. Розроблені шаблони головних сторінок для студента та викладачів. Реалізовані у представленому web-додатку.

Використання розробленого web-додатку дозволить зменшити зусилля викладача та часові затрати для коректної перевірки сформованих студентом запитів до бази даних.

Розроблений додаток впроваджений в роботу секції ІТП кафедри КН СумДУ, що підтверджується актом впровадження.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1 5 сайтов для оттачивания навыков написания SQL-запросов [Електронний ресурс]. – Режим доступу: https://proglib.io/p/sql-practice-sites.

2 Практическое владение языком SQL [Електронний ресурс]. – Режим доступу: http://sql-ex.ru/.

3 Postgre SQL Exercises [Електронний ресурс]. – Режим доступу: https://www.pgexercises.com/.

4 Introduction to SQL [Електронний ресурс]. – Режим доступу: https://sqlbolt.com/.

5 Ивасенко А. Г. Управление проектами : учебное пособие для студентов./Ивасенко А. Г. – Ростов н/Д. : Феникс , 2015. – 330 с.

6 Комп’ютерне моделювання [Електронний ресурс]. – Режим доступу: https://sites.google.com/.

7 Про методології IDEF0 [Електронний ресурс]. – Режим доступу: https://beasthackerz.ru/.

8 Розробка uml діаграми варіантів використання [Електронний ресурс]. – Режим доступу: https://studfile.net/preview/5200239/page:6/

9 SQL [Електронний ресурс] – режим доступу: http://progopedia.ru/language/sql/

10 Что такое DFD [Електронний ресурс] – режим доступу: https://habr.com/ru/company/trinion/blog/340064/

11 Діаграма варіантів використання [Електронний ресурс] – режим доступу: https://studopedia.ru/

12 Диаграмма компонентов (component diagram) [Електронний ресурс] – режим доступу: http://khpi-iip.mipk.kharkiv.edu/

13 Was ist ein Entity-Relationship-Diagramm? [Електронний ресурс] – режим доступу: https://bitly.su/jWaiTl

14 PHP MySQL Database [Електронний ресурс] – режим доступу: https://bitly.su/jiKXll.

15 Шварц, Б. MySQL по максимуму / Б. Шварц, П. Зайцев, В. Ткаченко. – СПб: Питер, 2018. – 864 с.

16 Кузнецов, М. Самоучитель PHP7 / М. Кузнецов, И. Симдянов. – СПб: БХВ-Петербург, 2018. – 448 с.

Додаток А

ТЕХНІЧНЕ ЗАВДАННЯ

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 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

Для підтримки web-системи й експлуатації web-інтерфейсу системи керування сайтом від персоналу не повинно вимагатися спеціальних технічних навичок, знання технологій або програмних продуктів, за винятком 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

Інформація повинна зберігатися в базі даних. Інформація щодо успішності може бути поміщена до Microsoft Excel.

2.1.4 Вимоги до розмежування доступу

Інформація, розташовувана у web-додатку, не є загальнодоступною.

Користувачів сайту можна розділити на такі групи відповідно до прав доступу:

1. Адміністратор
2. Студенти

Адміністратор може виконувати такі дії:

– Видаляти, змінювати та додавати нові завдання лабораторних робіт;

– Перевіряти роботи вручну;

– Погоджувати реєстрацію студента на сайті;

– Вносити еталонний запит, відповідь якого буде порівнюватися з результатом запитів студентів

– Погоджувати авторизацію викладачів;

Студенти можуть виконувати такі дії:

– Відслідковують успішність в особистому кабінеті,

– Надсилають листа викладачеві;

– Вибирати номер лабораторної роботи та завдання яке хоче виконати

– Переглядати свої спроби які попередньо здійснив.

2.2 Вимоги до функцій, виконуваних сайтом

2.2.1 Основні вимоги

2.2.1.1 Структура сайту

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.

2.2.1.2 Навігація

Інтерфейс web-системи повинен бути інтуїтивно зрозумілий і мати простий функціонал. Система повинна забезпечувати навігацію по всіх доступних користувачеві ресурсам і відображати відповідну інформацію. Посилання на сторінки повинні мати заголовок.

Система повинна забезпечувати навігацію по всіх доступних користувачеві ресурсам і відображати відповідну інформацію. Для навігації повинна використовуватися система контент-меню. Меню повинне являти собою текстовий блок (список гіперпосилань) у лівій колонці або у верхній частині сторінки (залежно від затвердженого дизайну).

2.2.1.3 Наповнення сайту (контент)

Сторінки всіх розділів сайту повинні формуватися програмним шляхом на підставі інформації з бази даних на сервері.

Редагування вмісту розділів повинно здійснюватися за допомогою адміністраторського web-інтерфейсу, який повинен передбачати можливість редагування інформаційного вмісту сторінок сайту без застосування спеціальних навичок програмування. Наповнення інформацією повинне проводитися з використанням шаблонів сторінок сайту.

2.2.1.4 Система навігації (карта сайту)

Взаємозв'язок між розділами й підрозділами сайту (карта сайту) представлено на рис. А.1-А.2.

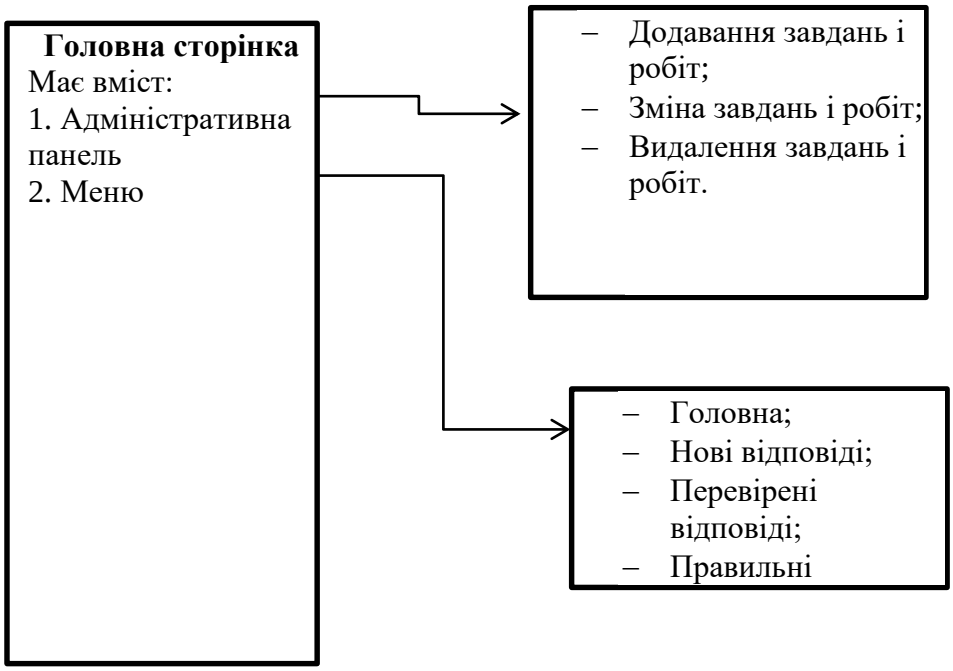


Рисунок А.1– Карта сайту для адміністратора

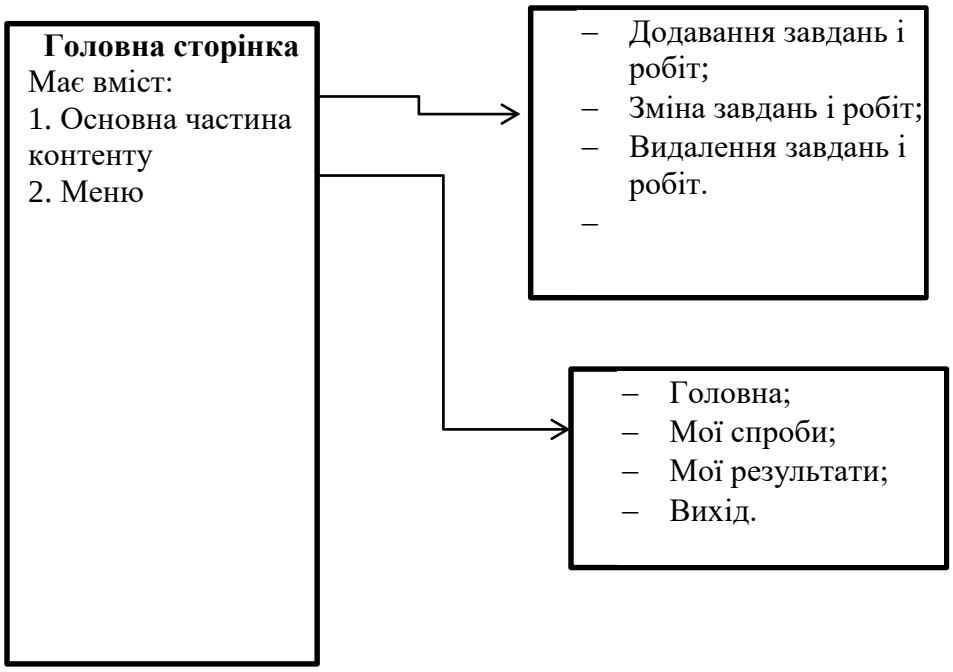


Рисунок А.2– Карта сайту для студента

2.2.2 Вимоги до функціональних можливостей

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.

2.2.2.1 Функціональні можливості розділів

На головній сторінці будуть представлені такі елементи:

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

2.2.2.2 Загальні вимоги

Стиль web-додатку можна описати як сучасний, діловий, що має лаконічний і зручний вигляд.

Оформлення не повинне защемляти інформативність: хоча на сайті не повинно бути досить багато графіки, він повинен бути зручний користувачам у плані навігації й цікавий для багаторазового відвідування.

Розташування елементів на головній сторінці сайту схематично показано на рис. А.3-А.4.

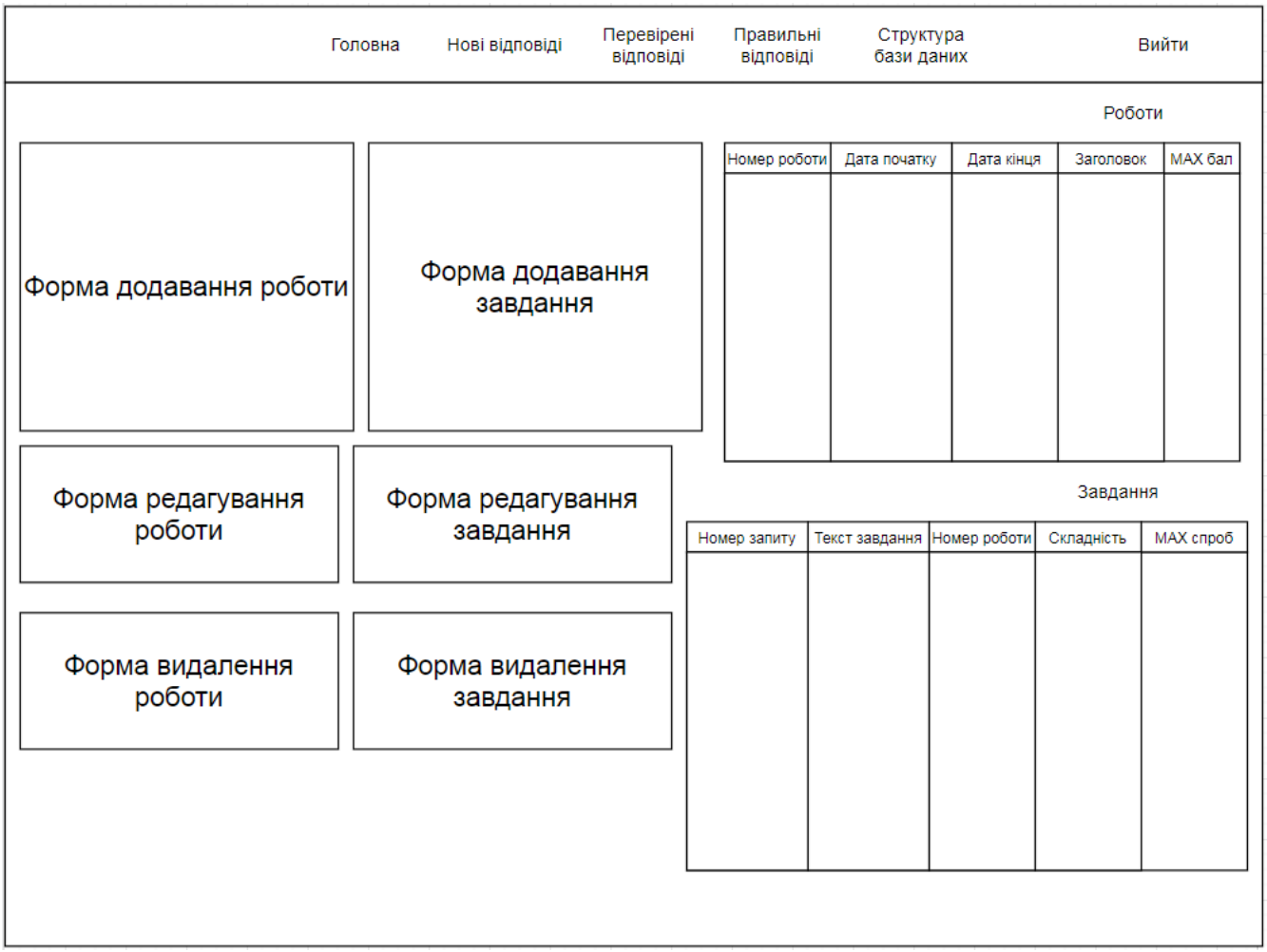


Рисунок А.3 – Головна сторінка адміністратора

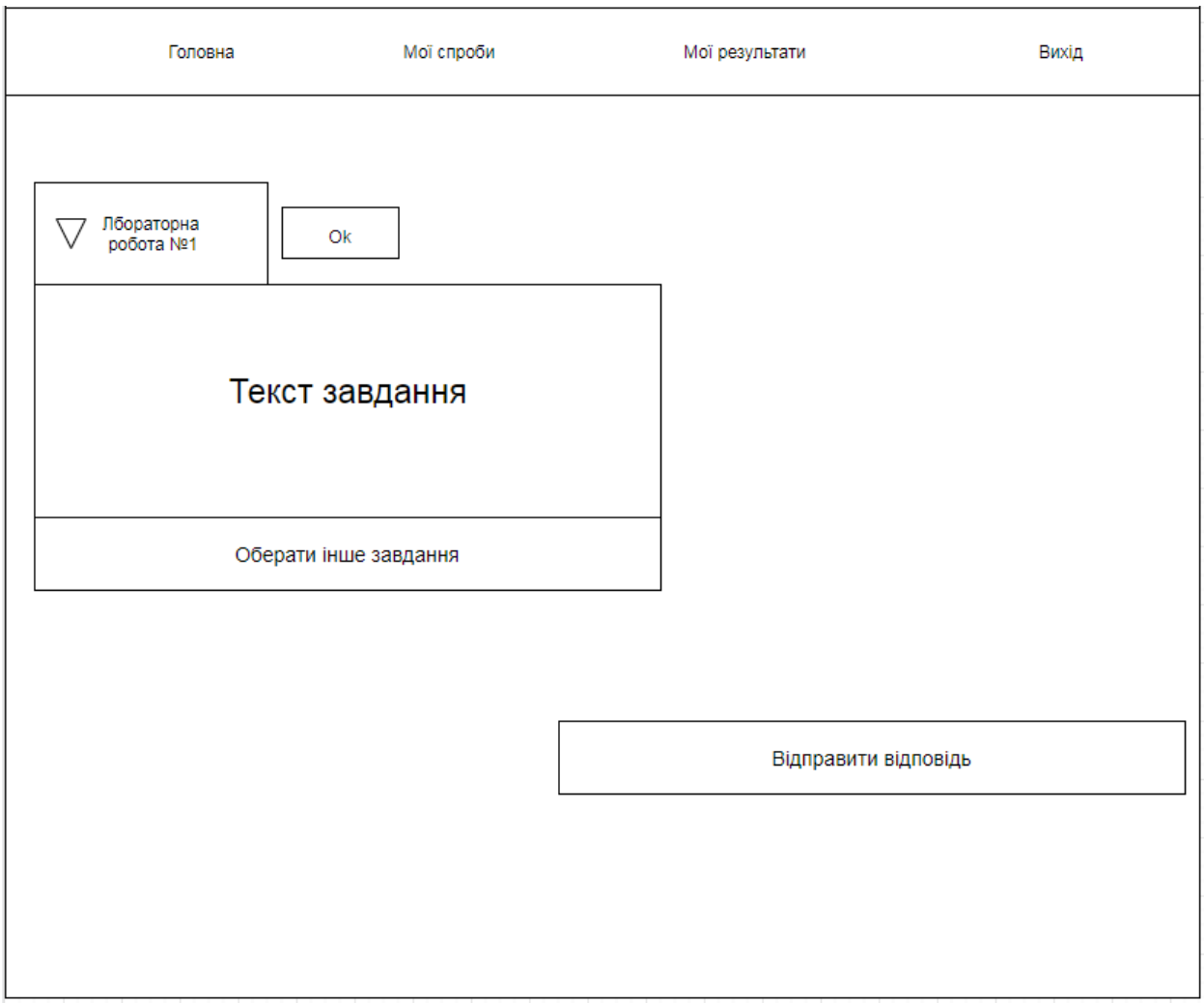


Рисунок А.4 – Головна сторінка студента

2.2.2.3 Типові навігаційні й інформаційні елементи

– Основне меню
– Основне поле контенту

2.2.2.4 Основне меню

Основне меню повинне розташовуватися у верхній частині вікна і містити посилання на сторінки сайту.

2.2.2.5 Основне поле контенту

Основне поле контенту повинне розташовуватися на головній сторінці. В даному полі відображається інформація с завданнями до лабораторної роботи.

2.2.1 Вимоги до інформаційного забезпечення

Реалізація відбувається за допомогою:

- PHP;
- HTML;
- JS;
- MySQL;
- CSS.

2.2.2 Вимоги до лінгвістичного забезпечення

Інтерфейс web-системи повинен бути виконаний українською мовою.

2.2.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Наявний доступ до Інтернету
- Веб-браузер: Firefox 3.1 (і вище), Google Chrome, Internet Explorer 7.0 (і вище).

2.2.4 Вимоги до апаратного забезпечення

Апаратне забезпечення клієнтської частини повинне забезпечувати підтримку програмного забезпечення клієнтської частини, зазначеного в п. 2.2.3

3 Склад і зміст робіт

Докладний опис етапів роботи розробки web-додатку наведено в табл. 1.

Таблиця А.1 – Етапи створення сайту

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Розробка каркасу: Проектування розмітки та наповнення сайту	2 день
2	Авторизація: Розроблення та реалізація блоку Авторизації на web-сайті	2 день
3	База даних: Розробка бази даних, встановлення зв’язків між таблицями, та наповнення їх.	2 день
4	Головний вивід (основна контентна частина): Реалізація виводу структурованих даних у центральній частині сайту по натисканню на кнопки розділів головного меню	3 дні
5	Завершення роботи: Проведення стилістичних виправлень веб-сайту, перевірка (тестування) реалізованого функціоналу	2 день
	Загальна тривалість робіт (з урахуванням резервного строку на налагодження й виправлення помилок) і строк закінчення проекту	11 днів

4 Вимоги до складу й змісту робіт із введення сайту в експлуатацію

Для створення умов функціонування, при яких гарантується відповідність створюваного сайту вимогам сьогодення ТЗ і можливість його ефективної роботи, в організації Замовника повинен бути проведений певний комплекс заходів.

Для переносу сайту на хостинг необхідно, щоб параметри хостинга відповідали вимогам, зазначеним у ТЗ. На хостинг переноситься програма (сайт), зверстаний шаблон дизайну й структура й наповнення бази даних з подальшою їх доробкою.

Додаток Б

Планування робіт

Деталізація мети web-додатку методом SMART.

Мета проекту: створити програмний продукт, який дозволить автоматизувати роботу викладача та заощадити час і сили. Мета є досяжною, знання які були здобуті на попередніх курсах з WEB, допоможуть реалізувати завдання. Проект буде виконано вчасно, що підтверджується календарним планом проекту.

Таблиця Б.1 – Деталізація мети web-додатку методом SMART

Specific (конкретна)	Візуалізувати модель сайту виконання лабораторних робіт з дисципліни «баз даних та знань».
Measurable (вимірювана)	Створити якісний web-додаток, використовуючи наявні ресурси у обсязі, що не виходять за межі можливостей.
Achievable (досяжна)	Створити web-додаток, використовуючи доступні 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 структури робіт.

WBS (Work Breakdown Structure) – структура проекту, що розбиває складну задачу на безліч простих, тим самим структурує виконання роботи і дає точний розрахунок на виконання часу та грошових витрат. WBS - діаграма даного проекту представлена на рисунку Б.1.

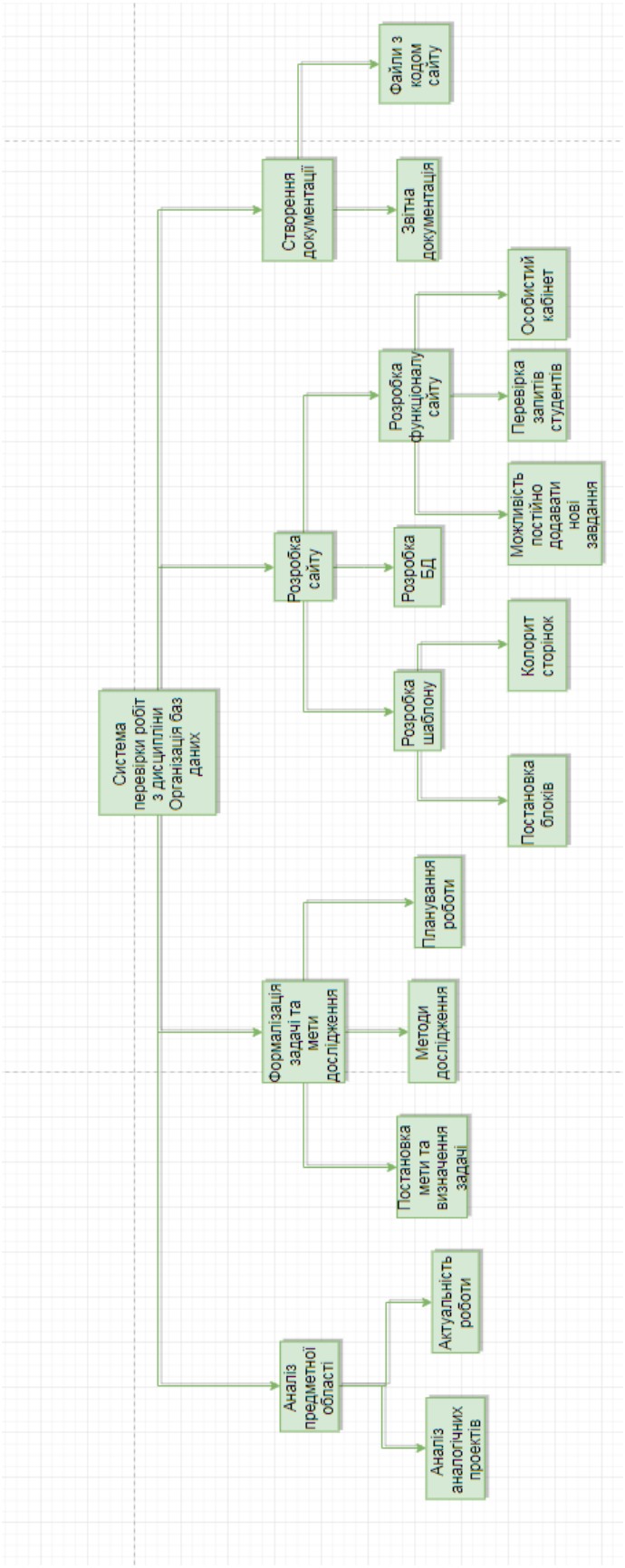


Рисунок Б.1. – WBS-діаграма

Планування структури організації, для впровадження готового проекту. Аналіз ризиків.

При розробці будь-якого проекту необхідно передбачити всі можливі ризики, які можуть трапитись та провести необхідні заходи для їх запобігання або вирішення.

Для даного проекту було визначено такі можливі ризики:

1. Часті зміни вимог до проекту
2. Помилки в документації
3. Невірно розрахований термін виконання
4. Бюджет
5. Несправність ПК
6. Несправність ПО
7. Неточність створеного проекту

У кожному проекті виділені ризики оцінюють за шкалою ймовірності їх виникнення та величини нанесених збитків. [3].

Таблиця Б.2 – Шкала оцінювання ймовірності виникнення та величини витрат

Оцінка	Ймовірність виникнення	Величина втрат
1	Слабоймовірна	Мінімальна
2	Малоймовірна	Низька
3	Ймовірна	Середня
4	Досить ймовірна	Висока
5	Майже можлива	Максимальна

Класифікуємо ризики згідно таблиці Б.2 та розрахуємо індекс ризику за формулою $R = P_q \times I_q$, де

R – індекс ризику (бали);

P_q – ймовірність виникнення ризиків відповідно до класифікації (бали);

I_q – величина втрат відповідно до класифікації ризику (бали).

Таблиця Б.3 – Оцінка ймовірності виникнення, величини витрат та індексу ризику

№	Ризики	Ймовірність виникнення	Величина ризику	Індекс ризику
1	Часті зміни вимог до проекту	4	4	16
2	Помилки в документації	3	4	12
3	Невірно розрахований термін виконання	2	2	4
4	Бюджет	2	1	2
5	Несправність ПК	3	5	15
6	Несправність ПО	5	3	15
7	Неточність створеного проекту	3	3	9

На підставі отриманого значення індексу ризики класифікують: за ступенем впливу (таблиця Б.4) та рівнем ризику (таблиця Б.5).

Таблиця Б.4 – Шкала оцінювання за ступенем впливу

№	Назва	Межі	Ризики, які входять
1	Ігноруючі	1≤R≤4	3. Невірно розрахований термін виконання 4. Бюджет
2	Незначні	5≤R≤8	Відсутні
3	Помірні	9≤R≤10	7. Неточність створеного проекту
4	Істотні	12≤R≤16	1. Часті зміни вимог до проекту 2. Помилки в документації 5.Несправність ПК 6. Несправність ПО
5	Критичні	20≤R≤25	Відсутні

Таблиця Б.5 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	1≤R≤4	3,4
2	Виправдані	5≤R≤10	7
3	Неприпустимі	12≤R≤25	1,2,5,6

Отже, необхідно завчасно передбачати усі можливі варіанти для уникнення будь-яких ризиків та по можливості не допускати їх виникнення.

Додаток В

ФАЙЛИ КОДУ РЕАЛІЗАЦІЇ

Файл коду admin.php

```
$_SESSION['coment'] = "";
?>
<html>
<head>
<link rel='stylesheet' href='style.css'>
<meta charset='utf-8'>
</head>
<body>
<?php
$uid = $_SESSION['user_id'];
if($uid==1){
		$db=mysqli_connect("localhost","root","","query_testing");
		if(!$db) exit();

		//МЕНЮ +
		echo"<div id='you_container'><ul id='you'>";
		echo"<li><a href='index.php'>Головна</a></li>";
		echo"<li><a href='new_users.php'>Нові користувачі</a></li>";
		echo"<li><a href='new_answer.php'>Нові відповіді</a></li>";
		echo"<li><a href='student_answer.php'>Перевірені відповіді</a></li>";
		echo"<li><a href='answer.php'>Правильні відповіді</a></li>";
		echo"<li><a href='describe_db_note.php'>Структура бази</a></li>";
		echo"<li><a href='index.php?link=exit'>Вийти</a></li>";
		echo"</ul></div>";
		//МЕНЮ -


		echo"<div id='content_admin'>";
		echo"<div><div class='grey_block'>";
		echo"<h2 class='pcenter'>+ Роботу</h2>";

		//ФОРМА ДОБАВЛЕНИЯ НОВОЙ РАБОТЫ +
		echo"<form method='post' action = 'admin_processor.php'>";
		echo"Дата початку <input type='text' name='date_start' placeholder='YYYY-MM-DD' pattern='[0-
9]{4}-[0-9]{1,2}-[0-9]{1,2}'><br>";
		echo"Дата кінця <input type='text' name='deadline' placeholder='YYYY-MM-DD' pattern='[0-9]{4}-[0-
9]{1,2}-[0-9]{1,2}'><br>";
		echo"<p class='pcenter'>Заголовок (Коротко)<textarea name='title'
style='height:80px;'></textarea></p>";
		echo"Максимальний бал<input type='text' name='max_mark'><br>";
		echo"<p class='pcenter'><input type='submit' value='Додати' name='add_task'></p>";
		echo"</form></div><br>";
		//ФОРМА ДОБАВЛЕНИЯ НОВОЙ РАБОТЫ -


		//ФОРМА ИЗМЕНЕНИЯ РАБОТЫ +
		echo"<div class='grey_block'><h2 class='pcenter'>Змінити роботу</h2><form method='post'>";
		$id_task=$_POST['update_id_task1'];
		echo"Номер <input type='text' name='update_id_task1' list='list_tasks' value='".$id_task."'><br>";
		echo"<p class='pcenter'><input type='submit' value='Змінити' name='update_task'></p>";
		//echo"<p class='pcenter'><input type='submit' value='Видалити' name='delete_task'></p>";
		if(isset($_POST['update_task'])){
		    $query4=mysqli_query($db,"SELECT * FROM `tasks` where id_task=".$id_task);
		    while ($myrow4=mysqli_fetch_array($query4, MYSQLI_ASSOC)){
```

```
                    echo"Дата початку <input type='text' name='update_date_start' placeholder='YYYY-MM-
DD' pattern='[0-9]{4}-[0-9]{1,2}-[0-9]{1,2}' value='".$myrow4['date_start']."'><br>";
                    echo"Дата кінця <input type='text' name='update_deadline' placeholder='YYYY-MM-DD'
pattern='[0-9]{4}-[0-9]{1,2}-[0-9]{1,2}' value='".$myrow4['deadline']."'><br>";
                    echo"<p class='pcenter'>Заголовок (Коротко)<textarea name='update_title'
style='height:80px;'>".$myrow4['title']."</textarea></p>";
                    echo"Максимальний бал<input type='text' name='update_max_mark'
value='".$myrow4['max_mark']."'><br>";
                    echo"<p class='pcenter'><input type='submit' value='Зберегти зміни' name='save_task'
formaction = 'admin_processor.php'></p>";
                }
            }
            echo"</form></div><br>";
            //ФОРМА ИЗМЕНЕНИЯ РАБОТЫ -


            //ФОРМА УДАЛЕНИЯ РАБОТЫ +
            echo"<div class='grey_block'><h2 class='pcenter'>Видалити роботу</h2><form method='post'>";
            $id_task=$_POST['delete_id_task'];
            echo"Номер <input type='text' name='delete_id_task' list='list_tasks' value='".$id_task."'><br>";
            echo"<p class='pcenter'><input type='submit' value='Видалити' name='delete_task'></p>";
            if(isset($_POST['delete_task'])){
                echo"<script>
                var txt = 'false';
                var isConfirm = confirm('Подтверждаете удаление работы? Вместе с ней удалятся все
вложенные задания, их эталонные запросы и решения студентов.');
                if(isConfirm == true){ txt = 'true';}
                var str = 'admin_processor.php?action=delete_task&id_task=".$id_task."&isConfirm='+txt;
                location.href = str;
                </script>";
            }
            echo"</form></div>";
            //ФОРМА УДАЛЕНИЯ РАБОТЫ -
            echo"</div>";


            //ФОРМА ДОБАВЛЕНИЯ НОВОГО ЗАДАНИЯ +
            echo"<div><div class='grey_block'>";
            echo"<h2 class='pcenter'>+ Завдання</h2>";
            echo"<form method='post' action = 'admin_processor.php'>";
            echo"<p class='pcenter'>Текст завдання <textarea name='text_exercise'></textarea></p>";
            echo"<p class='pcenter'>Правильна відповідь <textarea name='true_solution'></textarea></p>";
            echo"Робота <input type='text' name='id_task' list='list_tasks'><br>";
            echo"Зірок <input type='text' name='stars'><br>";
            echo"Спроб <input type='text' name='attempts'><br>";
            echo"<p class='pcenter'><input type='submit' value='Додати' name='add_query'></p>";
            echo"</form></div><br>";
            //ФОРМА ДОБАВЛЕНИЯ НОВОГО ЗАДАНИЯ -


            //ФОРМА ИЗМЕНЕНИЯ ЗАДАНИЯ +
            echo"<div class='grey_block'><h2 class='pcenter'>Змінити завдання</h2><form method='post'>";
            $id_query=$_POST['update_id_query'];
            echo"Номер <input type='text' name='update_id_query' list='list_queries'
value='".$id_query."'><br>";
            echo"<p class='pcenter'><input type='submit' value='Змінити' name='update_query'></p>";
            //echo"<p class='pcenter'><input type='submit' value='Видалити' name='delete_query'></p>";
            if(isset($_POST['update_query'])){
                $query3=mysqli_query($db,"SELECT q.*,s.true_solution FROM `queries` q left join
`Solution_by_teacher` s on(s.id_query=$id_query) where q.id_query=$id_query");
                while ($myrow3=mysqli_fetch_array($query3, MYSQLI_ASSOC)){
```

```
                    echo"<p class='pcenter'>Текст завдання <textarea name='update_text_exercise'>".$myrow3['text_exercise']."</textarea></p>";
                    echo"<p class='pcenter'>Правильна відповідь <textarea name='update_true_solution'>".$myrow3['true_solution']."</textarea></p>";
                    echo"Робота <input type='text' name='update_id_task2' list='list_tasks' value='".$myrow3['id_task']."'><br>";
                    echo"Зірок <input type='text' name='update_stars' value='".$myrow3['stars']."'><br>";
                    echo"Спроб <input type='text' name='update_attempts'><br>";
                }
                echo"<p class='pcenter'><input type='submit' value='Зберегти зміни' name='save_query' formaction = 'admin_processor.php'></p>";
            }
            echo"</form></div><br>";
            //ФОРМА ИЗМЕНЕНИЯ ЗАДАНИЯ -


            //ФОРМА УДАЛЕНИЯ ЗАДАНИЯ +
            echo"<div class='grey_block'><h2 class='pcenter'>Видалити завдання</h2><form method='post'>";
            $id_query=$_POST['delete_id_query'];
            echo"Номер <input type='text' name='delete_id_query' list='list_queries' value='".$id_query."'><br>";
            echo"<p class='pcenter'><input type='submit' value='Видалити' name='delete_query'></p>";
            if(isset($_POST['delete_query'])){
                echo"<script>
                var txt = 'false';
                var isConfirm = confirm('Подтверждаете удаление задания? Вместе с ним удалятся все эталонные запросы и решения студентов.');
                if(isConfirm == true){ txt = 'true';}
                var str = 'admin_processor.php?action=delete_query&id_query=".$id_query."&isConfirm='+txt;
                location.href = str;
                </script>";
            }
            echo"</form></div>";
            echo"</div>";
            //ФОРМА УДАЛЕНИЯ ЗАДАНИЯ -


            //ТАБЛИЦА С РАБОТАМИ +
            echo"<div><div class='grey_block'><h2 class='pcenter'>Роботи:</h2>";
            echo"<table class='result_table' style = 'max-width: 50vw'>";
            $for_add_admin = array();
            $p=1;
            $query=mysqli_query($db,"SELECT * FROM `tasks`");
                while ($myrow=mysqli_fetch_array($query, MYSQLI_ASSOC)){
                    if ($p==1) {
                        echo"<tr>";

                        foreach($myrow as $in => $buf){
                            if($in=='id_task'){ $col="Номер роботи";}
                            if($in=='deadline'){ $col="Дата кінця";}
                            if($in=='date_start'){ $col="Дата початку";}
                            if($in=='title'){ $col="Заголовок";}
                            if($in=='max_mark'){ $col="MAX бал";}
                            echo"<th>".$col."</th>";

                        }
                        echo"</tr>";
                    }
                    echo"<tr>";
                    foreach($myrow as $in=>$buf) {
                        if($in=='id_task'){
                            $for_add_admin[] = $buf;
```

```
                }
                echo"<td>".$buf."</td>";
            }
            echo"</tr>";
            $p++;
}
echo "</table></div><br><div class='grey_block'><h2 class='pcenter'>Завдання:</h2>";
//ТАБЛИЦА С РАБОТАМИ +


//ТАБЛИЦА С ЗАДАНИЯМИ +
echo"<table class='result_table' style = 'max-width: 50vw'>";
$p=1;
$for_add_admin1 = array();
$query=mysqli_query($db,"SELECT * FROM `queries`");
    while ($myrow=mysqli_fetch_array($query, MYSQLI_ASSOC)){
            if ($p==1) {
                echo"<tr>";

                foreach($myrow as $in => $buf){
                    if($in=='id_task'){ $col="Номер роботи";}
                    if($in=='text_exercise'){ $col="Текст завдання";}
                    if($in=='id_query'){ $col="Номер запиту";}
                    if($in=='stars'){ $col="Складність";}
                    if($in=='attempts'){ $col="MAX спроб";}

                    echo"<th>".$col."</th>";

                }
                echo"</tr>";
            }
            echo"<tr>";
            foreach($myrow as $in=>$buf) {
                if($in=='id_query'){
                    $for_add_admin1[] = $buf;
                }
                if($in=='stars'){
                    echo"<td style='max-width:330px'>(".$buf.")<br>";
                    for($i = 0; $i < $buf; $i++){
                        if($buf > 4){
                            echo"<br>...";
                            break;
                        }
                        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 ЗАДАНИЯМИ -


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>
```

Файл коду index.php

```
<?php
session_start();
if($_SERVER['REQUEST_URI']=='/bd_kurs/index.php?link=exit'){
                    session_unset();
                    echo"<script>document.location.href = 'index.php'</script>";
}
function form_str($start){
                $test = stripslashes($start);
                if($test!=$start){
                    echo"Не використовуйте \ при введенні даних. Це службовий символ.";
                    return -1;
                }
                else{
                    $next = addslashes($start);
                    $next = str_replace(" ", "\ ", $next);
                    $next = str_replace("<", "\<", $next);
                    $next = str_replace(">", "\>", $next);
                    $next = str_replace("--", "\-", $next);
                    $next = str_replace(";", "\;", $next);
                    $next = str_replace(":", "\:", $next);
                    $next = str_replace("(", "\(", $next);
                    $next = str_replace(")", "\)", $next);
                    $next = str_replace(".", "\.", $next);
                    $finish = $next;
                    return $finish;
}}
function reform_str($start){
                $finish = stripslashes($start);
                return $finish;
}
function nvl($start){
                if($start==""){
                    $finish = 'null';
                }
                else{
                    $finish = $start;
                }
                return $finish;
}
if(isset($_POST['sub_user']) || isset($_POST['sub_reg']) || isset($_POST['add_sub'])){
                $db=mysqli_connect("localhost","root","","query_testing");
                if(!$db){ exit();}
}
if(isset($_POST['sub_user'])){
                $login = form_str($_POST['login']);
                $password = form_str($_POST['password']);
```

```
		$query=mysqli_query($db,"SELECT * FROM `users` WHERE `login` = '$login' and `password` = '$password' and `processed`");
		while($myrow = mysqli_fetch_array($query, MYSQLI_ASSOC)){
			$_SESSION['user_id'] = $myrow['id_user'];
		}
		if($_SESSION['user_id']==""){
			echo "Користувача з такими логіном і паролем не знайдено.";
}}
if(isset($_POST['sub_reg'])){
		$login = form_str($_POST['login']);
		$password = form_str($_POST['password']);
		$password1 = form_str($_POST['password1']);
		$name = form_str($_POST['name']);
		$surname = form_str($_POST['surname']);
		$num_group = form_str($_POST['num_group']);
		$id_group = null;
		$query_group=mysqli_query($db,"SELECT max(id_group) as id_group FROM `groups` WHERE `num_group` = '$num_group'");
		while($myrow_group = mysqli_fetch_array($query_group, MYSQLI_ASSOC)){
			$id_group = $myrow_group["id_group"];
		}
		if($name!=-1 & $login!=-1 & $name!=-1){
			if($password==$password1){
				if($password!="" & $login!="" & $name!=""){
					$str="insert into users(name, login, password, surname, id_group, processed) values ('$name','$login','$password', '$surname', $id_group, false)";
					if($str!=$_SESSION['last_add']){
						$query=mysqli_query($db,$str);
						$_SESSION['last_add']=$str;
						if($query){
							echo"Запит принято, очікуйте схвалення реєстрації від адміністратора";
						}
						else{
							echo"Виникла помилка";
						}
					}
				}
				else{
					echo"Сервер отримав порожні параметри. Спробуйте ще раз.";
			}}
			else{
				echo"Введені паролі не співпадають!";
}}}
?>
<html>
<head>
<link rel='stylesheet' href='style.css'>
<meta charset='utf-8'>
</head>
<body>
<?php
$uid = $_SESSION['user_id'];
if($uid==""){
		$group_list = $_SESSION['group_list'];
		$j = 0;
		if($group_list == null){
			$db=mysqli_connect("localhost","root","","query_testing");
			if(!$db) exit();
			$query_group=mysqli_query($db,"SELECT distinct num_group FROM `groups`");
			while($myrow_group = mysqli_fetch_array($query_group, MYSQLI_ASSOC)){
				$group_list[$j] = $myrow_group["num_group"];
```

```
                    $j++;
                }
                $_SESSION['group_list'] = $group_list;
            }
            echo"<datalist id = 'group_list'>";
            for($i = 0; $i < sizeof($group_list); $i++){
                echo"<option value = '".$group_list[$i]."'></option>";
            }
            echo"</datalist>";

            if($_SERVER['REQUEST_URI']!='/bd_kurs/index.php' &
$_SERVER['REQUEST_URI']!='/bd_kurs/index.php?link=registration'){
                echo "<script>document.location.href = 'index.php'</script>";
            }
            else{
                if($_SERVER['REQUEST_URI']=='/bd_kurs/index.php'){
                    echo"<form method='post' id='autorization'>";
                    echo"<h2>Авторизація</h2>";
                    echo"<p>Логін <input type='text' name='login' required></p>";
                    echo"<p>Пароль <input type='password' name='password' required></p>";
                    echo"<h2><input type='submit' name='sub_user' value='Далі'></h2>";
                    echo"<a href='index.php?link=registration'>Реєстрація</a>";
                    echo"</form>";
                }
                else{
                    if($_SERVER['REQUEST_URI']=='/bd_kurs/index.php?link=registration'){
                        echo"<form method='post' id='registration'>";
                        echo"<h2>Реєстрація</h2>";
                        echo"<p>Ім`я<input type='text' name='name' required></p>";
                        echo"<p>Прізвище<input type='text' name='surname' required></p>";
                        echo"<p>Група<input type='text' name='num_group' required list =
'group_list'></p>";
                        echo"<p>Логін<input type='text' name='login' required></p>";
                        echo"<p>Пароль<input type='password' name='password' id='psw'
required></p>";
                        echo"<p>Повторіть пароль <input type='password' name='password1' id='psw1'
required></p>";
                        echo"<h2><input type='submit' name='sub_reg' value='Далі'></h2>";
                        echo"<a href='index.php'>Авторизація</a>";
                        echo"</form>";
                    }
                    else{
                        echo"<script>document.location.href = 'index.php'</script>";
}}}}
else{
            $db=mysqli_connect("localhost","root","","query_testing");
            if(!$db)
                exit();
            if($uid==1){
                echo"<script>document.location.href='admin.php'</script>";
            }
            $id_task=$_SESSION['id_task'];
            $id_query=$_SESSION['id_query'];
            $coment = $_SESSION['coment'];
            if($coment != null & $coment != ""){
                echo $coment;
                $_SESSION['coment'] = "";
            }
            echo"<div id='you_container'><ul id='you'>";
            echo"<li><a href='index.php'>Головна</a></li>";
            echo"<li><a href='try.php'>Мої спроби</a></li>";
            echo"<li><a href='result.php'>Мої результати</a></li>";
```

```
            echo"<li><a href='describe_db_note.php'>Структура бази</a></li>";
            echo"<li><a href='index.php?link=exit'>Вийти</a></li>";
            echo"</ul></div>";
            echo"<div style='min-height:100vh;width:100%'>";
            $tasks_for_select=mysqli_query($db,"select * from Tasks");
            echo"<form method='post' id='content' action = 'user_processor.php'>";
            echo"<fieldset>";
            if($id_task==""){
                echo"<h2>Оберіть роботу зі списку:</h2>";
            }
            echo"<select name='tasks' id='tasks'><option value=''></option>";

            while ($row_tasks_for_select=mysqli_fetch_array($tasks_for_select, MYSQLI_ASSOC)){
                echo "<option value='".$row_tasks_for_select['id_task']."'";
                if($id_task==$row_tasks_for_select['id_task']){
                        echo" selected";
                }
                echo">".$row_tasks_for_select['title']."</option>";
            }
            echo"</select><input type='submit' name='choose_task' value='ok' style='margin-
left:30px'></fieldset>";
            if($id_task>0 & ($id_query==null || $id_query=="")){
                $queries_for_select=mysqli_query($db,"select q.id_query, q.text_exercise,
                q.attempts-count(s.id_solution) as attempts_limit from Queries q
                left join solution_by_student s on(s.id_query = q.id_query and s.id_user = ".$uid.")
                where q.id_task=$id_task group by q.id_query, q.text_exercise");
                echo"<fieldset class='list'>";
                echo"<h2>Оберіть завдання зі списку:</h2>";
                while ($row_queries_for_select=mysqli_fetch_array($queries_for_select, MYSQLI_ASSOC)){
                        if($row_queries_for_select['attempts_limit']>0){
                                echo "<div><input type='radio' name='query'
value='".$row_queries_for_select['id_query']."'>".$row_queries_for_select['text_exercise']."
                                <span style = 'float:right; color:grey;'>/ Залишилось
(".$row_queries_for_select['attempts_limit'].") спроби</span></div><hr>";
                        }
                        else{
                                echo "<div><input type='radio' disabled name='query'
value='".$row_queries_for_select['id_query']."'>".$row_queries_for_select['text_exercise']."
                                <span style = 'float:right; color:grey;'>/ Кількість спроб
вичерпано</span></div><hr>";
                        }
                }
                echo"<br><input type='submit' name='choose_query' value='ok'></fieldset>";
            }
            if($id_task>0 & ($id_query!=null || $id_query!="")){
                $queries_for_select=mysqli_query($db,"select q.id_query, q.text_exercise,
                q.attempts-count(s.id_solution) as attempts_limit from Queries q
                left join solution_by_student s on(s.id_query = q.id_query and s.id_user = ".$uid.")
                where q.id_query=$id_query group by q.id_query, q.text_exercise");

                echo"<fieldset>";
                $row_queries_for_select=mysqli_fetch_array($queries_for_select, MYSQLI_ASSOC);
                if($row_queries_for_select['text_exercise']==""){
                        $_SESSION['id_query']='';
                        echo"<script>document.location.href=document.location.href</script>";
                }
                echo $row_queries_for_select['text_exercise']."<hr>";
                echo"<br><input type='submit' name='other_query' value='Обрати інше
завдання'></fieldset>";

                if($row_queries_for_select['attempts_limit']<1){
                        echo"<textarea name='text_query' disabled></textarea>";
```

```
                        echo"<p class='alltoleft'><input type='submit' name='add_sub' value='Відправити відповідь' id='add_sub' disabled></p>";
                        echo"<script>alert('Кількість спроб вичерпано');</script>";
                        $_SESSION['id_query']='';
                        //echo"<script>document.location.href=document.location.href</script>";
                    }
                    else{
                        echo"<textarea name='text_query'></textarea>";
                        echo"<p class='alltoleft'><input type='submit' name='add_sub' value='Відправити відповідь' id='add_sub'></p>";
                    }
                }
                echo"</form></div>";
}
?>
</body></html>
```

Файл коду admin_processor.php

```
<?php
session_start();
$_SESSION['coment'] = "";

$uid = $_SESSION['user_id'];
if($uid==1){
                if(isset($_POST['add_task']) || isset($_POST['add_query']) || isset($_POST['save_query']) || isset($_POST['save_task'])
                    ||$_GET['action']=='delete_task'||$_GET['action']=='delete_query'){
                    $db=mysqli_connect("localhost","root","","query_testing");
                    if(!$db){ exit();}
                }
                if(isset($_POST['add_task'])){
                    $max_mark = nvl(preg_replace('/[^0-9.]/', '', $_POST['max_mark']));
                    $title = nvl(form_str($_POST['title']));
                    $deadline = nvl(preg_replace('/[^0-9\-]/', '', $_POST['deadline']));
                    $date_start = nvl(preg_replace('/[^0-9\-]/', '', $_POST['date_start']));
                    $str="insert into `tasks`(max_mark,title,deadline,date_start) values($max_mark,'$title','$deadline','$date_start')";
                    //if($str!=$_SESSION['last_add']){
                        $query=mysqli_query($db,$str);
                    //      $_SESSION['last_add']=$str;
                    //}
                }
                if(isset($_POST['save_task'])){
                    $max_mark = nvl(preg_replace('/[^0-9.]/', '', $_POST['update_max_mark']));
                    $title = nvl(form_str($_POST['update_title']));
                    $deadline = nvl(preg_replace('/[^0-9\-]/', '', $_POST['update_deadline']));
                    $date_start = nvl(preg_replace('/[^0-9\-]/', '', $_POST['update_date_start']));
                    $id_task = nvl(preg_replace('/[^0-9]/', '', $_POST['update_id_task1']));
                    $str="update `tasks` set max_mark=$max_mark,title='$title',deadline='$deadline',date_start='$date_start' where id_task=$id_task";
                    //if($str!=$_SESSION['last_add']){
                        $query=mysqli_query($db,$str);
                    //      $_SESSION['last_add']=$str;
                    //}
                }
                if(isset($_POST['add_query'])){
                    $id_task = nvl(preg_replace('/[^0-9]/', '', $_POST['id_task']));
                    $text_exercise = nvl(form_str($_POST['text_exercise']));
                    $true_solution = nvl(form_str($_POST['true_solution']));
                    $true_solution = str_replace(";","",$true_solution);
                    $stars = $_POST['stars'];
```

```
                $attempts = $_POST['attempts'];

                if($stars==""){ $stars = 0;}
                if($attempts==""){ $attempts = 0;}

                $id_query = 1;
                $str1=mysqli_query($db,"SELECT max(id_query) id_query from `queries`");
                if($myrow1=mysqli_fetch_array($str1, MYSQLI_ASSOC)){
                        $id_query = $myrow1['id_query']+1;
                }
                $str="insert into `queries`(id_query,id_task,text_exercise,stars,attempts)
values($id_query,$id_task,'$text_exercise',$stars,$attempts)";
                $str1="insert into `Solution_by_teacher`(id_task,true_solution,id_query)
values($id_task,'$true_solution',$id_query)";
                $str2=$str."".$str1;
                //if($str2!=$_SESSION['last_add']){
                        $query=mysqli_query($db,$str);
                        $query1=mysqli_query($db,$str1);
                //      $_SESSION['last_add']=$str2;
                //}
            }
            if(isset($_POST['save_query'])){
                $id_query = nvl(preg_replace('/[^0-9]/', '', $_POST['update_id_query']));
                $id_task = nvl(preg_replace('/[^0-9]/', '', $_POST['update_id_task2']));
                $true_solution = nvl(form_str($_POST['update_true_solution']));
                $text_exercise = nvl(form_str($_POST['update_text_exercise']));
                $stars = $_POST['update_stars'];
                $attempts = $_POST['update_attempts'];

                if($stars==""){ $stars = 0;}
                if($attempts==""){ $attempts = 0;}

                $str="update `queries` set id_task = $id_task, text_exercise = '$text_exercise', stars = $stars, attempts
= $attempts where id_query = $id_query";
                $str1="update `Solution_by_teacher` set id_task = $id_task, true_solution = '$true_solution' where
id_query = $id_query";
                $str2=$str.";".$str1;
                //if($str2!=$_SESSION['last_add']){
                        $query=mysqli_query($db,$str);
                        $query1=mysqli_query($db,$str1);
                //      $_SESSION['last_add']=$str2;
                //}
            }
            if($_GET['action']=='delete_task' & $_GET['isConfirm']=='true'){
                $id_task = $_GET['id_task'];
                $str="DELETE FROM `Tasks` where `id_task` = ".$id_task;
                //if($str!=$_SESSION['last_add']){
                        $query=mysqli_query($db,$str);
                //      $_SESSION['last_add']=$str;
                //}
            }
            if($_GET['action']=='delete_query' & $_GET['isConfirm']=='true'){
                $id_query = $_GET['id_query'];
                $str="DELETE FROM `Queries` where `id_query` = ".$id_query;
                //if($str!=$_SESSION['last_add']){
                        $query=mysqli_query($db,$str);
                //      $_SESSION['last_add']=$str;
                //}
            }
}

function form_str($start){
```

```
                $test = stripslashes($start);
                if($test!=$start){
                    echo"Не використовуйте \ при введенні даних. Це службовий символ.";
                    return -1;
                }
                else{
                    $next = addslashes($start);
                    $next = str_replace(" ", "\ ", $next);
                    $next = str_replace("<", "\<", $next);
                    $next = str_replace(">", "\>", $next);
                    $next = str_replace("--", "\-", $next);
                    $next = str_replace(";", "\;", $next);
                    $next = str_replace(":", "\:", $next);
                    $next = str_replace("(", "\(", $next);
                    $next = str_replace(")", "\)", $next);
                    $next = str_replace(".", "\.", $next);
                    $finish = $next;
                    return $finish;
                }
}

function reform_str($start){
                $finish = stripslashes($start);
                return $finish;
}
function nvl($start){
                if($start==""){
                    $finish = 'null';
                }
                else{
                    $finish = $start;
                }
                return $finish;
}
if (@$_SERVER['HTTP_REFERER'] != null) {
   header("Location:".$_SERVER['HTTP_REFERER']);
}

?>
```