

Лабораторна робота №5

Тема: Робота з базою даних MySQL та з бібліотекою PDO у мові PHP

Мета роботи: ознайомитися з бібліотекою PDO для роботи з базами даних у мові PHP

Хід роботи

Завдання 1:

Для відкриття **phpmyadmin**, на **wampserver** (Рис.1). За замовчанням ім'я користувача **root**, пароль залишіть пустим.



1. Створіть MySQL базу даних **lab5** і в ній таблицю **users**, яка зберігатиме інформацію про користувачів (від 10 до 15 полів).

2. На головній сторінці розмістіть форму для введення логіну та пароля і посилання на сторінку реєстрації.

3. Реалізуйте форму реєстрації користувачів та внесення даних до таблиці **users**. При цьому потрібно перевіряти коректність заповнення полів та заборонити повторну реєстрацію користувачів.

4. При заповненні форми входу на сайт, потрібно перевірити, чи є відповідний запис у таблиці **users**. Якщо є, то записати у сесію ознаку того, що користувач успішно аутентифікований і надалі для нього не відображати форму входу на сайт та форму реєстрації.

5. Для користувача, який успішно пройшов аутентифікацію потрібно на головній сторінці відобразити вітання та надати можливість зміни своїх даних, які

					ІПТР.420001.123-ЗЛ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Мартинюк Ю.І.				Звіт з лабораторної роботи		Лім.	Арк.
Перевір.	Ковтун В.В.							1
Керівник							ФІКТ, гр. ІПЗ-23-3	
Н. контр.								
Затверд.								

він ввів при реєстрації. При зміні даних вони повинні змінюватися і в таблиці **users**.

6. Реалізуйте можливість виходу користувача.

7. Реалізуйте можливість видалення власного профілю користувачем.

У цьому завданні я розробила веб-систему, яка дозволяє користувачам реєструватися, входити в систему, редагувати свій профіль та видаляти свій обліковий запис. Основна мета полягала у створенні функціонального та відносно безпечного механізму керування користувачами з використанням PHP та бази даних MySQL.

Основні виконані частини:

- 1. Конфігурація підключення до бази даних:** Я створила файл `db_config.php`, який містить необхідні налаштування для підключення до бази даних MySQL. Це включає ім'я хоста, ім'я бази даних, ім'я користувача та пароль. Для забезпечення коректної роботи з україномовною символікою я також налаштувала кодування з'єднання на UTF-8:

```
// db_config.php
$pdo->exec("SET NAMES utf8"); // Встановлює кодування з'єднання на UTF-8
```

- 2. Реалізація процесу реєстрації:** На сторінці `register.php` я розробила форму для введення реєстраційних даних. Одним з ключових аспектів було забезпечення безпеки збереження паролів. Для цього я використала функцію `password_hash()`:

```
// register.php
$hashed_password = password_hash($password, PASSWORD_DEFAULT);
// ...
$stmt_insert->bindParam(':password', $hashed_password);
```

					ІПТР.420001.123-ЗЛ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

Це дозволило мені зберігати в базі даних не сам пароль, а його безпечний хеш. Також я реалізувала перевірку на унікальність логіна та email, щоб запобігти дублюванню облікових записів.

3. **Реалізація процесу входу:** На сторінці login.php я обробила введені користувачем логін та пароль. Для перевірки пароля я використала функцію password_verify():

```
// login.php
if ($user && password_verify($password, $user['password'])) {
    // ... аутентифікація успішна
}
```

Ця функція безпечно порівнює введений пароль з хешем, що зберігається в базі даних. У разі успішної аутентифікації я створюю сесію користувача для відстеження його авторизованого стану.

4. **Розробка функціоналу редагування профілю:** На сторінці profile.php авторизовані користувачі можуть переглядати та змінювати свої особисті дані. Для оновлення даних у базі я використовувала підготовлені SQL-запити з прив'язкою параметрів, що є важливим для захисту від SQL-ін'єкцій:

```
// profile.php
$stmt_update = $pdo->prepare("UPDATE users SET first_name = :first_name, ... WHERE id = :id");
$stmt_update->bindParam(':first_name', $first_name);
// ... інші bindParam ...
$stmt_update->bindParam(':id', $user_id);
```

5. **Реалізація функції видалення профілю:** На сторінці delete_profile.php користувач може видалити свій обліковий запис. Перед виконанням видалення я передбачила підтвердження цієї дії. Сам процес видалення також здійснюється за допомогою підготовленого SQL-запиту:

```
// delete_profile.php
```

					ІПТР.420001.123-ЗЛ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

```
$stmt_delete = $pdo->prepare("DELETE FROM users WHERE id = :id");  
$stmt_delete->bindParam(':id', $user_id);
```

Після успішного видалення я знищую сесію користувача.

6. **Забезпечення безпеки:** Протягом розробки я приділяла особливу увагу безпеці. Окрім хешування паролів та використання підготовлених запитів для запобігання SQL-ін'єкціям, я також використовувала сесії для керування авторизацією та перевіряла авторизацію на кожній сторінці, доступ до якої повинен бути обмежений для неавторизованих користувачів.

Вхід на сайт

Логін:

Пароль:

Увійти

Зареєструватися

Реєстрація

Логін:

Yulia188

Пароль:

Email:

martinukulia311@gmail.com

Ім'я:

Юлія

Прізвище:

Мартинюк

Дата народження:

06.05.2025

Стать:

Жінка

Місто:

Житомир

Країна:

Україна

Телефон:

0664985612

Зареєструватися

Назад на головну

Вітаємо, користувачу!

Редагувати профіль

Вийти

Реєстрація

Реєстрація успішна! Тепер ви можете увійти.

					ІРТР.420001.123-ЗЛ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

Редагування профілю

Ім'я:

Юлія

Прізвище:

Мартинюк

Дата народження:

06.05.2025



Стать:

Жінка



Місто:

Житомир

Країна:

Україна

Телефон:

0664985612

Зберегти зміни

[Назад на головну](#)

[Видалити профіль](#)

[Вийти](#)

Видалення профілю

Ви впевнені, що хочете видалити свій профіль? Ця дія незворотня.

Видалити

Скасувати

Ваш профіль було успішно видалено.

Вхід на сайт

Логін:

Пароль:

Увійти

[Зареєструватися](#)

Вхід на сайт

Користувача з таким логіном не існує.

Логін:

Пароль:

Увійти

[Зареєструватися](#)

Рис.1-9 Результат виконання програми

					ІРТР.420001.123-ЗЛ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2:

1. У розділі **Користувачі** додайте нового користувача **homeuser** з усіма правами.

2. Перевірте конфігураційний файл **php.ini** (знаходиться в корені директорії з версією php, яку Ви обрали в Вашому проекті, наприклад **C:\OSPanel\modules\php\php-7.1**). Розширення **php_pdo_mysql.dll** має бути підключене (перед записом **extension=php_pdo_mysql.dll** НЕ повинна стояти крапка з комою).

3. Створіть папку **Lab5**. У ній файл **index.php**.

4. Підключіться до бази даних:

```
$pdo = new PDO('mysql:host=localhost;dbname=slim_db;charset=utf8','root','');
```

5. Для обробки виключень заключіть код в блок **try...catch()**. Опишіть в даному блоці дії, які необхідно виконувати при виникненні помилок. Наприклад:

```
try {  
    $pdo = new  
PDO('mysql:host=localhost;dbname=slim_db;charset=utf8','root','');  
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
}  
catch(PDOException $e) {  
    echo $e->getMessage();  
}
```

6. До бази даних, створеної в попередньому завданні додайте ще одну таблицю з назвою **tov**:

- a. У поле Ім'я ввести **tov**;
- b. У поле Поля – число **5**;
- c. Кнопка **Ок**.

7. Додайте 10 записів до таблиці **tov**. Тематику товарів придумайте самостійно.

8. Виведіть на екран всі дані з таблиці **tov**. Приклад коду:

```
$sql = "SELECT * FROM tov";  
$result = $pdo->query($sql);  
  
foreach($result as $row) {
```

					ІПТР.420001.123-ЗЛ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

echo "<p>".$row['id']." | ".$row['username'];
}
або
$sql = "SELECT * FROM tov";
$result = $pdo->query($sql);

while($row = $result->fetch()) {
    echo "<p>".$row['id']." | ".$row['username'];
}

```

9. Оформіть виведення полів з таблиці **tov** у вигляді таблиці або блоків.
Наприклад:

Створення бази даних

Номер	Назва	Ціна	Кількість	Примітка

Рис. 2

10. Розмістіть 2 кнопки **Додати запис** і **Вилучити запис** і текстове поле для вказівки **№ запису**, що віддаляється.

11. Створіть файл **insert.php**, у якому розмістити форму для введення нових даних у таблицю. Наприклад, для виконання запитів типу **INSERT** та **UPDATE** потрібно, необхідно викликати метод **exec()** та в якості першого параметру передати запит **SQL**, який виконується.

```

$sql = 'INSERT INTO `tov` (`kkk`, `bbb`) VALUES ("newskkk","newbbb");
$affected_rows = $pdo->exec($sql);
echo $affected_rows;
або
if ($result = $db->prepare("INSERT INTO `users`(`name`,`email`,`age`)
VALUES (?, ?, ?)"))
{
    $name = "Andrew";
    $email = "andrew@hisemail.com";

```

```

$age = 22;
$data = array ( $name, $email, $age );
$result->execute($data);
$id = $db->lastinsertid();
echo $id; // 4
}

```

12. Створіть файл **delete.php**, у якому опишіть код для видалення запису за заданим номером. Приклад для поновлення запису (аналогічний видаленню):

```

if ($result = $db->prepare("UPDATE `users` SET `age`=:age WHERE `name`=:name"))
{
    $age = 25;
    $name = "Alex";
    $result->bindParam(":age",$age,PDO::PARAM_INT);
    $result->bindParam(":name",$name,PDO::PARAM_STR);
    $result->execute();
}

```

13. Експортуйте створену базу даних до файлу та скопіюйте до папки **Lab6**.

13. Експортуйте створену базу даних до файлу та скопіюйте до папки **Lab6**.

Додаткові методичні рекомендації до виконання роботи:

Структура бази даних **TOVARS** може бути наступною:

1. Таблиця **tovar**, містить облікові записи товарів

№	Назва поля	Опис	Тип
1	id	Поле-Лічильник	INT
2	name	Назва товару	VARCHAR (20)
3	cost	Вартість	INT
4	kol	Кількість товару	INT
5	date	Дата реалізації	DATE

Приклад записів:

id	name	cost	kol	date
1	Хліб столовий	24	100	25.03.18
2	Хліб житній	20	50	27.03.18

У цьому завданні я розробила веб-систему, яка дозволяє переглядати список товарів з бази даних slim_db, а також додавати та видаляти окремі записи. Основна мета полягала у створенні простого та функціонального інтерфейсу для управління даними про товари.

Основні виконані частини:

1. **Підключення до бази даних та відображення списку товарів:** На головній сторінці index.php я реалізувала підключення до бази даних slim_db за допомогою PDO. Для забезпечення коректного відображення даних я виконала запит SELECT * FROM tov та вивела результати у вигляді HTML-таблиці:

```
// index.php
$pdo = new PDO('mysql:host=localhost;dbname=slim_db;charset=utf8','root','');
// ...
$sql = "SELECT * FROM tov";
$result = $pdo->query($sql);
// ... виведення даних у таблицю ...
```

2. **Реалізація функціоналу додавання нового товару:** Для додавання нових товарів я створила сторінку insert.php з HTML-формою. Обробка даних з цієї форми відбувається у файлі insert_process.php. Для безпечного додавання даних до бази я використала підготовлений SQL-запит з прив'язкою параметрів:

```
// insert_process.php
$sql = "INSERT INTO tov (name, cost, kol, note) VALUES (:name, :cost, :kol, :note)";
$stmt = $pdo->prepare($sql);
$stmt->bindParam(':name', $name);
$stmt->bindParam(':cost', $cost, PDO::PARAM_INT);
// ...
$stmt->execute();
```

Це дозволило мені запобігти SQL-ін'єкціям при додаванні нових записів.

					ІПТР.420001.123-3Л	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Реалізація функціоналу видалення товару: На головній сторінці index.php я передбачила форму для введення ID товару, який потрібно видалити. Обробка цього запиту відбувається у файлі delete_process.php (який у коді називається delete.php). Для безпечного видалення запису я також використала підготовлений SQL-запит з прив'язкою параметра ID:

```
// delete_process.php
$sql = "DELETE FROM tov WHERE id = :id";
$stmt = $pdo->prepare($sql);
$stmt->bindParam(':id', $idToDelete, PDO::PARAM_INT);
$stmt->execute();
```

Перед виконанням запиту на видалення я також перевіряла, чи є введений ID числовим, для уникнення потенційних помилок.

4. Візуальне оформлення: Для покращення користувацького досвіду я створила файл style2.css, який містить базові стилі для оформлення сторінок та елементів, таких як таблиці, форми та посилання.

Підключення до бази даних slim_db успішне!

Список товарів

Номер	Назва	Ціна	Кількість	Примітка
14	Навушники	1021	13	Білі, з шумопоглинанням
2	Навушники бездротові	799	15	Чорні, з шумопоглинанням
3	Кава мелена 250г	120	30	Арабіка, свіжа обсмажка
4	Книга "Володар кілець	350	10	Тверда обкладинка
5	Блокнот у клітинку	45	100	A5, 96 сторінок
6	Зарядний пристрій US	180	25	2 порти, швидка зарядка
7	Ручка кулькова	15	200	Синя паста
8	Пляшка для води	150	40	750 мл, нержавіюча сталь
9	Шоколад молочний	35	80	100г, з горіхами
10	Пазл "Карта світу"	280	12	1000 елементів

Керування записами

Додати запис

Введіть ID запису для видалення:

Вилучити запис

Додати новий товар

Назва товару:

Хліб

Ціна:

30

Кількість:

15

Примітка:

житній

Додати товар

Назад до списку товарів

10	Пазл "Карта світу"	280	12	1000 елементів
15	Хліб	30	15	житній

Керування записами

Додати запис

Введіть ID запису для видалення:

15

Вилучити запис

Запис з ID 15 успішно видалено.

Назад до списку товарів

Рис.10-14 Результат виконання програми

Завдання 3:

- Створіть базу даних MySQL за допомогою PhpMyAdmin і назвіть її "company_db".
- Створіть таблицю "employees" з наступними полями: id (INT, AUTO_INCREMENT, PRIMARY KEY), name (VARCHAR), position (VARCHAR), і salary (DECIMAL).
- Вставте кілька записів в таблицю "employees" через PhpMyAdmin.
- Створіть PHP-сторінку для виведення всіх записів з таблиці "employees".
- Реалізуйте можливість додавання нового працівника через веб-форму на PHP-сторінці.
- Створіть функціонал для редагування існуючого запису працівника (зміна імені, посади або заробітної плати).
- Реалізуйте можливість видалення запису про працівника з бази даних.
- Створіть PHP-скрипт, який виводить статистику, наприклад, середню заробітну плату всіх працівників чи кількість працівників на кожній посаді.

У цьому завданні я розробила веб-систему для управління інформацією про працівників компанії, яка дозволяє переглядати їх список, додавати нових працівників, редагувати існуючі записи та видаляти їх. Також реалізовано базову статистику щодо заробітних плат. Основна мета полягала у створенні простого та функціонального інструменту для управління даними про персонал.

Основні виконані частини:

1. **Відображення списку працівників:** На головній сторінці index.php я реалізувала підключення до бази даних company_db та вибірку всіх записів з таблиці employees. Отримані дані відображаються у вигляді HTML-таблиці з можливістю перегляду імені, посади та зарплати кожного працівника, а також посиланнями на дії редагування та видалення:

```
// index.php
$conn = new PDO("mysql:host=$servername;dbname=$dbname;charset=utf8", $username,
$password);
// ...
$stmt = $conn->prepare("SELECT id, name, position, salary FROM employees");
$stmt->execute();
$result = $stmt->fetchAll(PDO::FETCH_ASSOC);
// ... виведення даних у таблицю з посиланнями на редагування та видалення ...
```

2. **Додавання нового працівника:** Для додавання нового працівника я створила сторінку add_employee.php з відповідною HTML-формою. Обробка даних з цієї форми відбувається у файлі process_add_employee.php. Для безпечного додавання даних до бази я використала підготовлений SQL-запит з прив'язкою параметрів:

```
// process_add_employee.php
$stmt = $conn->prepare("INSERT INTO employees (name, position, salary) VALUES (:name,
:position, :salary)");
$stmt->bindParam(':name', $name);
```

					ІРТР.420001.123-ЗЛ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

```
$stmt->bindParam(':position', $position);
$stmt->bindParam(':salary', $salary);
$stmt->execute();
```

3. **Редагування інформації про працівника:** Функціонал редагування реалізовано на сторінці `edit_employee.php`. Спочатку за допомогою GET-параметра `id` отримуються дані працівника, які заповнюють форму редагування. Після внесення змін та відправки форми дані обробляються у файлі `process_edit_employee.php`, де за допомогою підготовленого SQL-запиту `UPDATE` оновлюється відповідний запис у базі даних:

```
// process_edit_employee.php
$stmt = $conn->prepare("UPDATE employees SET name = :name, position = :position, salary = :salary WHERE id = :id");
$stmt->bindParam(':id', $id);
$stmt->bindParam(':name', $name);
// ...
$stmt->execute();
```

4. **Видалення працівника:** Видалення працівника здійснюється за допомогою GET-запиту до файлу `delete_employee.php`, в якому передається ID працівника для видалення. У цьому файлі виконується підготовлений SQL-запит `DELETE` для видалення відповідного запису з бази даних:

```
// delete_employee.php
$stmt = $conn->prepare("DELETE FROM employees WHERE id = :id");
$stmt->bindParam(':id', $id);
$stmt->execute();
```

Перед виконанням запиту я також перевіряю, чи є переданий ID коректним числовим значенням.

5. **Відображення статистики:** Для аналізу даних про працівників я створила сторінку `statistics.php`. На цій сторінці реалізовано розрахунок

					ІПТР.420001.123-ЗЛ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

та відображення середньої заробітної плати всіх працівників, а також виводиться кількість працівників на кожній посаді за допомогою SQL-запитів AVG() та COUNT() з GROUP BY:

```
// statistics.php
$stmt = $conn->query("SELECT AVG(salary) AS average_salary FROM employees");
// ...
$stmt = $conn->query("SELECT position, COUNT(*) AS count FROM employees GROUP BY position");
// ... виведення результатів ...
```

6. **Візуальне оформлення:** Для забезпечення базового візуального стилю всіх сторінок системи я використала файл style3.css, який містить CSS-правила для форматування тексту, таблиць, форм та посилань.

Список працівників

Додати нового працівника

Переглянути статистику

ID	Ім'я	Посада	Зарплата	Дії
1	Іван Петренко	Менеджер з продажу	16000.00	Редагувати Видалити
4	Марія Сидоренко	Бухгалтер	11000.00	Редагувати Видалити
5	Олег Коваленко	Програміст	18000.00	Редагувати Видалити
6	Анна Шевченко	Маркетолог	13500.00	Редагувати Видалити
7	Дмитро Гончарук	Аналітик даних	16000.00	Редагувати Видалити

Додати нового працівника

Ім'я:

Юлія

Посада:

Програміст

Зарплата:

22000.00

Додати працівника

[Назад до списку працівників](#)

					ІПТР.420001.123-ЗЛ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

Список працівників

Додати нового працівника

Переглянути статистику

ID	Ім'я	Посада	Зарплата	Дії
1	Іван Петренко	Менеджер з продажу	16000.00	Редагувати Видалити
10	Юлія	Програміст	22000.00	Редагувати Видалити
4	Марія Сидоренко	Бухгалтер	11000.00	Редагувати Видалити
5	Олег Коваленко	Програміст	18000.00	Редагувати Видалити
6	Анна Шевченко	Маркетолог	13500.00	Редагувати Видалити
7	Дмитро Гончарук	Аналітик даних	16000.00	Редагувати Видалити

Редагувати працівника

Ім'я:

Юлія

Посада:

Програміст

Зарплата:

25000.00

Зберегти зміни

[Назад до списку працівників](#)

10	Юлія	Програміст	25000.00	Редагувати Видалити
----	------	------------	----------	---

Рис.15-19 Результат виконання програми

Висновок: На сьогоднішньому занятті я ознайомила з бібліотекою PDO для роботи з базами даних у мові PHP.