**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**



**ІКНІ**

Кафедра **ПЗ**

ЗВІТ

До лабораторної роботи № 3

**На тему:** *“* *РНР як засіб написання сценаріїв”*

**З дисципліни:** *“Програмування в Інтернет”*

**Лектор:**

асистент каф. ПЗ

Степанов Д. С.

**Виконав:**

ст. гр. ПЗ-22

Гамела Б. А.

**Перевірила:**

старша викладачка каф. ПЗ

Грицай О. Д.

«\_\_\_»  \_\_\_\_\_\_\_\_\_  2025 р.

∑ = \_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_

Львів – 2025

**Тема роботи:** РНР як засіб написання сценаріїв.

**Мета роботи:** Засвоїти елементи програмування серверної частини застосувань мовою РНР.

# Теоретичні відомості

PHP (Препроцесор Гіпертексту) – це мова сценаріїв загального призначення з відкритим кодом, призначена переважно для веб-розробки. Вона вбудовується в HTML-код і виконується на сервері, генеруючи динамічні веб-сторінки. Синтаксис PHP подібний до C, Java та Perl, що робить його легким для вивчення.

**Основні характеристики:**

* **Серверна обробка**: PHP-код виконується на сервері, а браузер отримує готовий HTML.
* **Інтерпретатор**: PHP транслює код у байт-код, що забезпечує швидке виконання без створення виконавчого файла.
* **Переваги**:
  + Не вимагає явного керування пам’яттю чи оголошення типів змінних.
  + Спрощує відлагодження завдяки повідомленням інтерпретатора.
  + Інтегрується з базами даних (MySQL, PostgreSQL тощо) та підтримує роботу з файлами, cookie, PDF, зображеннями.
  + Безкоштовний, кросплатформений (Linux, Windows тощо).
* **Використання**: PHP-код вбудовується в HTML за допомогою тегів <?php ?>, підтримує оператори print(), echo(), include(), require() для виведення даних та підключення файлів.

**Приклади**:

* Вбудований код: <?php print("Hello, World"); ?> виводить текст у HTML.
* Самостійний код: підрахунок відвідувачів з використанням файлу count.dat.
* Відповіді сервера: JSON-формати для помилок ({status: false, error: {...}}) чи успішних операцій ({status: true, id: 1}).

PHP широко застосовується для створення форм, обробки даних та взаємодії з базами даних у веб-додатках.

# Формулювання завдання

1. Розробити web-сторінку згідно [макета](https://cacoo.com/diagrams/ZvVhYS3UpG5PdbBy/EDE3A) (wireframe).

2. Сторінка повинна відповідати наступним [вимогам](https://docs.google.com/document/d/15B3YB6pn-m4WZZtV4-iVfSlNV4h-sJA7Tcr-TV-rNzI/edit?usp=sharing).

# Результати виконання (спільні 3-4 лабораторна)

Репозиторій проекту на github.com: [PI-Labs](https://github.com/thething1001/PI-Labs)

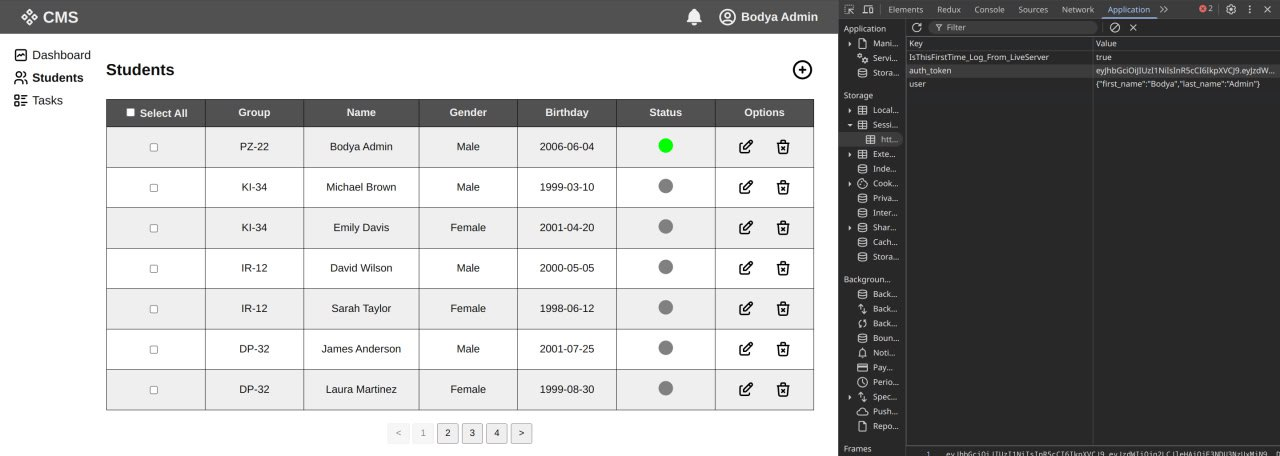
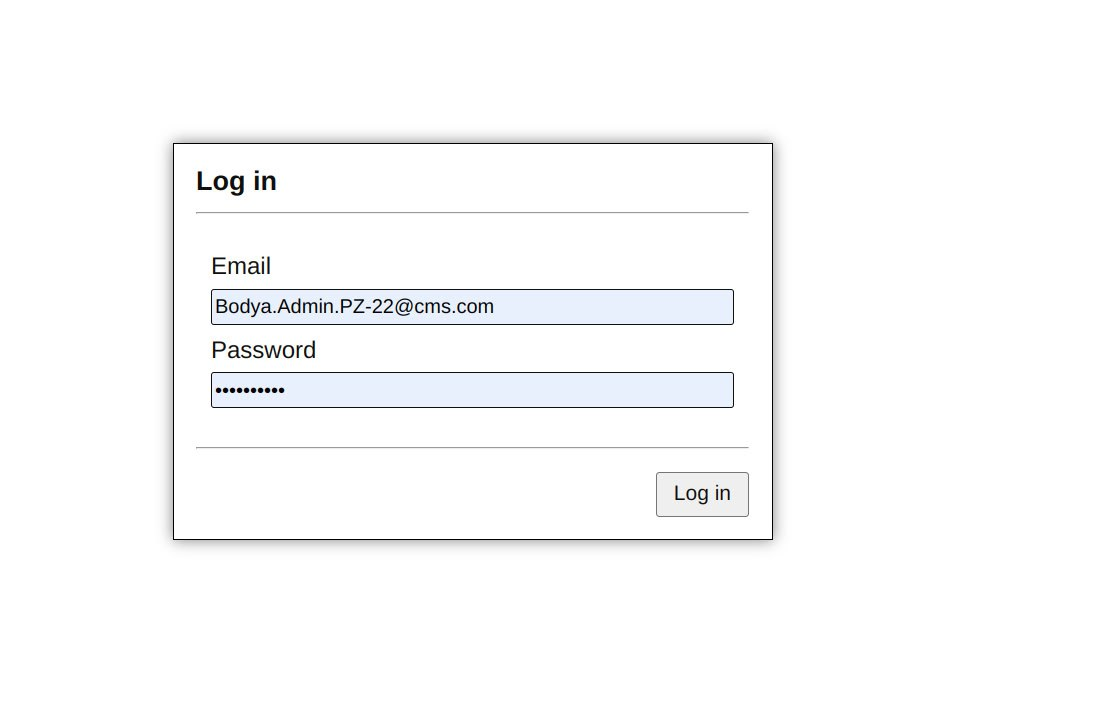


Рис. 1. Авторизація (JWT)

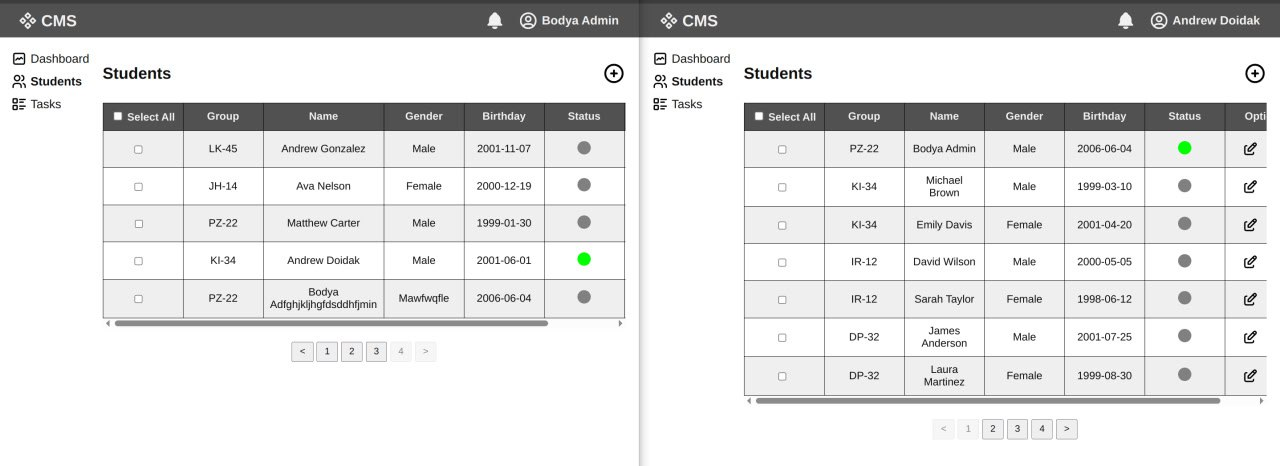


Рис. 2. Відображення студентів та статусу

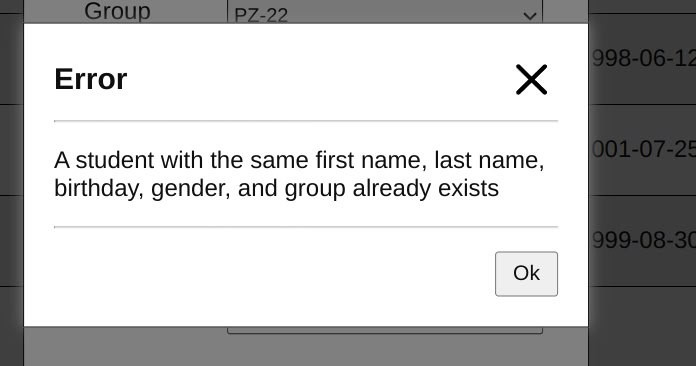


Рис. 3. Повідомлення про помилки з backend’у

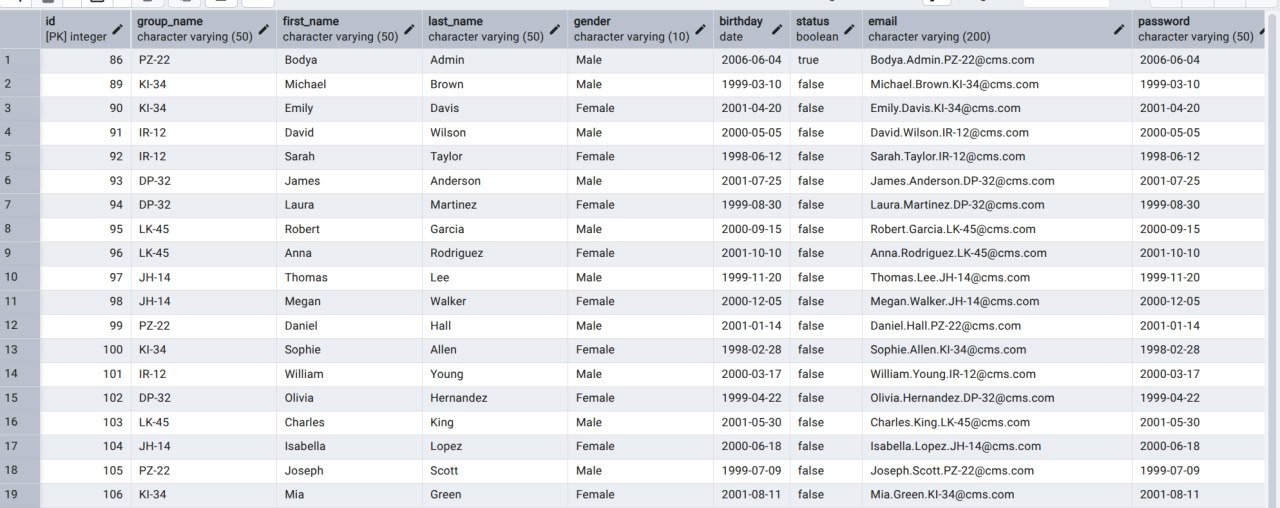


Рис. 4. База даних

# Висновки (спільні 3-4 лабораторна)

Під час виконання лабораторних робіт №3 та №4 я поглибив знання про серверну веб-розробку з використанням PHP та взаємодію з базами даних PostgreSQL. Було розроблено веб-сторінку з формою для додавання, редагування та видалення даних студентів, що взаємодіє з сервером через JSON-відповіді. Серверна частина, побудована на Apache, використовувала PDO для обробки запитів до PostgreSQL, валідації даних та повернення результатів у JSON-форматі (успішні операції або помилки). Для авторизації застосовано JWT. На клієнтській стороні використано JavaScript для обробки подій, відображення помилок та оновлення таблиці студентів.

Робота розвинула практичні навички створення динамічних веб-додатків, поглибила розуміння серверної логіки, роботи з PostgreSQL через PDO, авторизації за допомогою JWT та інтеграції клієнт-серверних технологій, що сприятиме подальшому розвитку компетенцій у веб-розробці.