**Звіт лабораторної роботи 11**

**Тема**: Робота з промісами та запитами до API

**Мета**: навчитися працювати з об'єктом Promise, методами .then(), .catch(); навчитися використовувати fetch() для отримання даних із зовнішнього API; засвоїти базові принципи обробки помилок у асинхронному програмуванні.

**КОНТРОЛЬНІ ПИТАННЯ**

1. Що таке Promise у JavaScript? Які існують стани у Promise?

2. Поясніть механізм роботи методів .then(), .catch() та .finally().

3. Як обробляти паралельні API-запити за допомогою Promise.all() та Promise.allSettled()?

**Відповіді:**

1) **Promise** – це об'єкт для роботи з асинхронними операціями. Він має 3 стани:

* pending (очікування)
* fulfilled (успішно виконано)
* rejected (відхилено)

2) **Методи Promise**:

* .then(res => {...}) – виконується при успіху.
* .catch(err => {...}) – обробляє помилки.
* .finally(() => {...}) – виконується завжди, незалежно від результату.

3) **Promise.all() vs Promise.allSettled()**:

* Promise.all([...]) – очікує всі проміси, якщо хоч один відхилений – все падає.
* Promise.allSettled([...]) – чекає всі проміси і повертає їхні результати незалежно від успіху чи помилки.

Висновок: На лабораторній роботі було виконана робота з об'єктом Promise, методами .then(), .catch(); використано fetch() для отримання даних із зовнішнього API; засвоїно базові принципи обробки помилок у асинхронному програмуванні.