# Лабораторна робота 2: Розробка RESTful API

**Мета:** Ознайомитися з основами розробки RESTful API, навчитися створювати та інтегрувати RESTful сервіси у корпоративний веб-додаток.

#### **Теоретичні відомості:**

**RESTful API** – це архітектурний стиль веб-сервісів, який ґрунтується на протоколі HTTP і відповідає принципам архітектури REST (Representational State Transfer). Основними компонентами RESTful API є ресурси, які ідентифікуються унікальними URI (Uniform Resource Identifier), і методи HTTP (GET, POST, PUT, DELETE), які використовуються для взаємодії з цими ресурсами.

**Основні принципи RESTful API:**

1. **Унікальний ідентифікатор ресурсу:** Кожен ресурс має унікальний URI.
2. **Стандартизовані методи:** Для роботи з ресурсами використовуються стандартні методи HTTP.
3. **Безстанова взаємодія:** Кожен запит до сервера є незалежним, і сервер не зберігає стан між запитами.
4. **Клієнт-серверна архітектура:** Клієнт і сервер розділені, і взаємодіють через стандартний інтерфейс.
5. **Кешування:** Відповіді можуть бути кешовані для зменшення навантаження на сервер.

#### **Завдання до виконання:**

1. **Ознайомлення з RESTful API:**
   * Розглянути принципи роботи RESTful API та їх відмінності від інших архітектур веб-сервісів.
2. **Розробка серверної частини RESTful API:**
   * Вибрати мову програмування та відповідний фреймворк для реалізації API. Наприклад, Node.js з Express або NestJS, Python з FastAPI aбо Flask, .NET з ASP.NET Core.
   * Створити серверний додаток, що реалізує CRUD-операції (створення, читання, оновлення, видалення) над певним ресурсом (наприклад, користувачі, товари, замовлення).
   * Виконати валідацію вхідних даних та обробку можливих помилок.
3. **Тестування RESTful API:**
   * Використати інструменти для тестування API, такі як Postman або cURL.
   * Створити та виконати тести для перевірки правильності роботи CRUD-операцій API.

#### **Контрольні питання:**

1. Що таке RESTful API? В яких випадках його доцільно використовувати?
2. Які основні принципи лежать в основі архітектури REST?
3. Як реалізувати CRUD-операції у RESTful API?
4. Які інструменти використовуються для тестування API?
5. Які особливості маршрутизації у обраному Вами фреймворку?
6. Як використовувати дані з запиту?
7. Як повернути відповідь?

#### **Вимоги до звіту:**

У звіті повинні бути наведені:

* Титульний лист
* Мета роботи.
* Код реалізованого API.
* Результати тестування API.
* Висновки за результатами виконаної роботи.