Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп’ютерних технологій

ЛАБОРАТОРНА РОБОТА № 6
з курсу “Веб програмування на стороні сервера”
“Створення сервісу інвентаризації ”

Виконала:
Ст. ФЕІ-23
Гвоздевич Вікторія
Перевірив:
асистент Чмихало Олександр
Сергійович

Львів 2025

Мета роботи

● Робота з різними типами HTTP запитів
● Закріпити знання з використання Express.js для створення Веб-сервісів
● Використання Postman для тестування сервісу
● Створити документацію для веб сервісу з використанням Swagger

Хід роботи:

Підготовка:

● Створено новий репозиторій backend-course-2025-6.

● Налаштовано ім’я користувача у форматі Ваше Ім’я bc2025-6 та електронну пошту для комітів за допомогою команди git config.

● Створено файл package.json у кореневій теці та додано головний файл програми; обидва файли закомічено.

● Локально встановлено пакет Commander.js для роботи з аргументами командного рядка. ○ Створено файл .gitignore та додано тека node_modules до виключень Git. ○ Підключено модуль Commander.js у програмі та реалізовано читання аргументів командного рядка.

● Локально встановлено пакет superagent.

● Додано пакет nodemon як залежність для розробки.

● Закомічено оновлений package.json, package-lock.json та головний файл програми.

Частина 1 - параметри командного рядка та Веб сервер

● Програма налаштована на приймання параметрів командного рядка за допомогою Commander.js: ○ -h, --host — обов’язковий параметр, що задає адресу сервера ○ -p, --port — обов’язковий параметр, що задає порт сервера ○ -c, --cache — обов’язковий параметр, що задає шлях до директорії для кешування файлів. Якщо вказано шлях до неіснуючої директорії, програма автоматично створює її під час запуску.

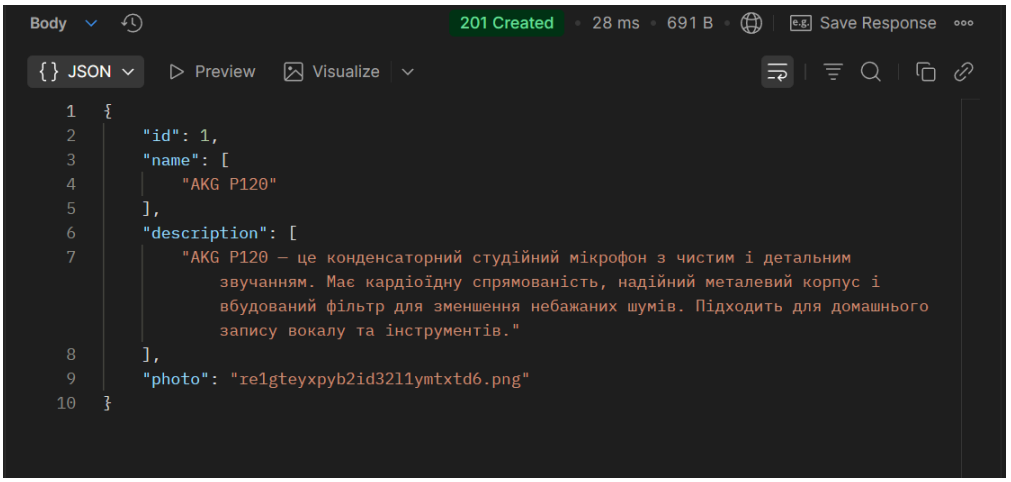
● Реалізовано перевірку: якщо будь-який обов’язковий параметр не передано — виводиться відповідна помилка.

● За допомогою модуля http запущено веб-сервер; значення параметрів --host та --port передаються у метод http.Server.listen().

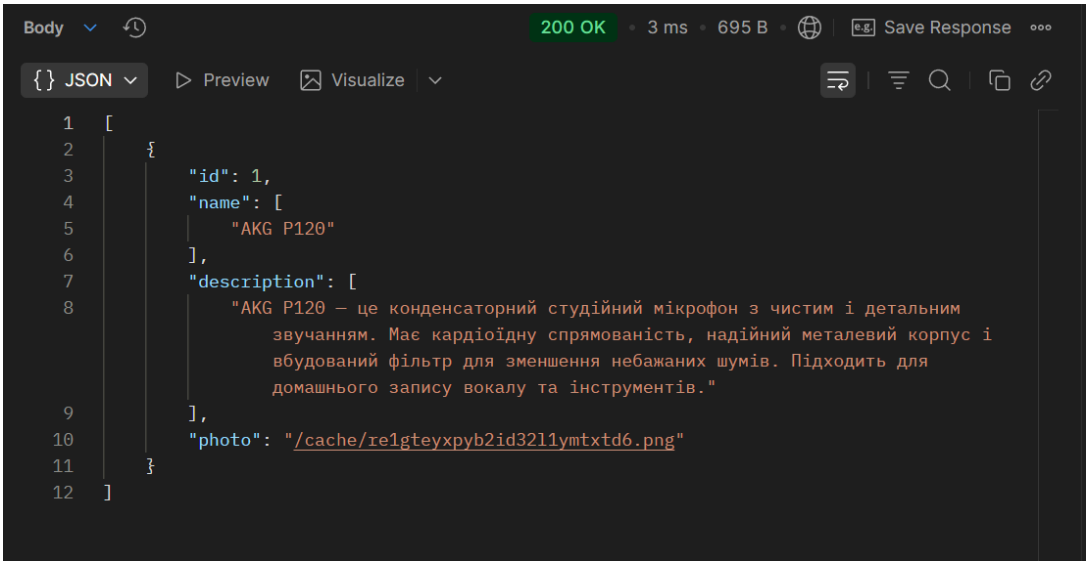
● Для автоматичного перезапуску сервера при зміні файлів сервер запускається через nodemon, встановлений як залежність лише для розробки.

Частина 2 - Реалізація WebAPI сервісу

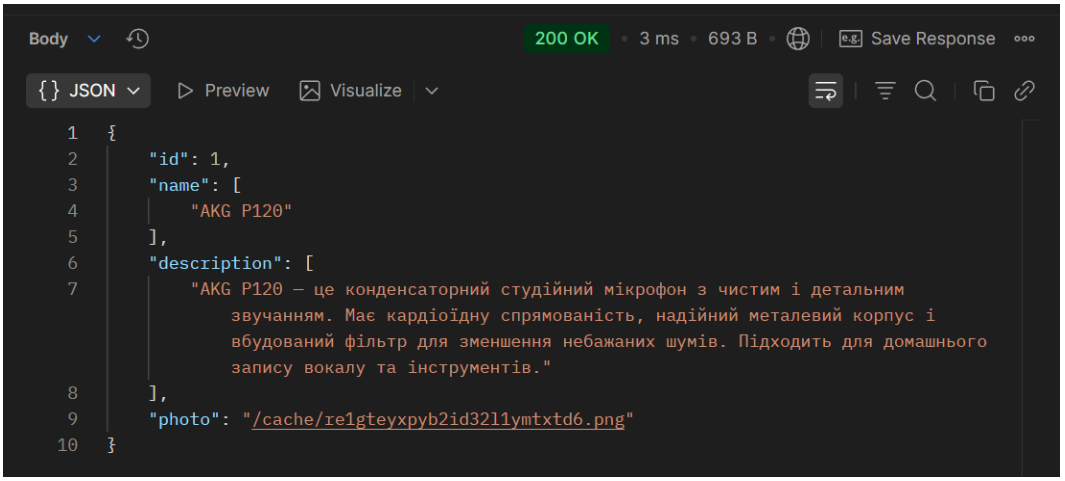
Реалізовано POST /register Обробляється multipart form data за допомогою formidable Приймає поля inventory_name description photo Якщо ім’я не передано сервер повертає 400 Bad Request У разі успіху повертається 201 Created



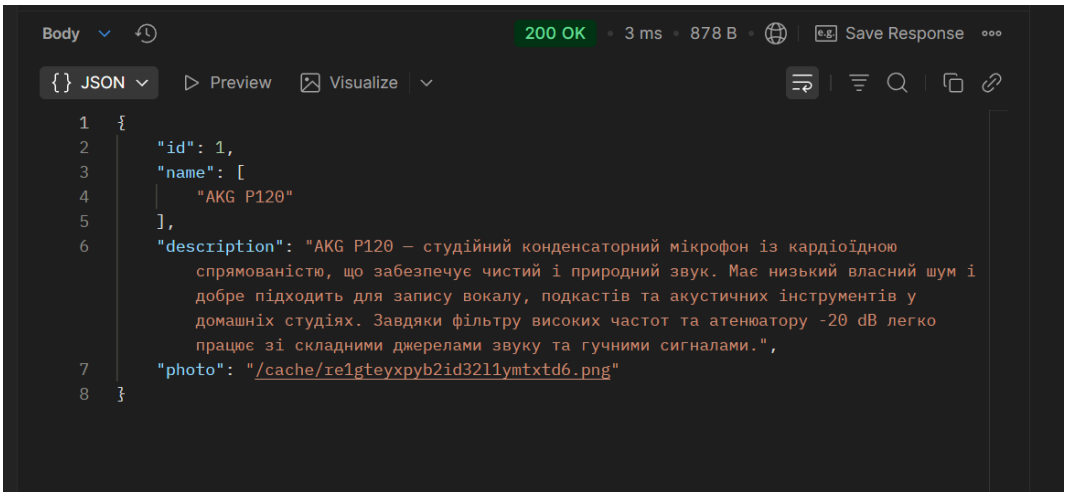
Реалізовано GET /inventory Повертає список усіх інвентаризованих речей у форматі JSON Кожен елемент містить ID опис інформацію та посилання на фото



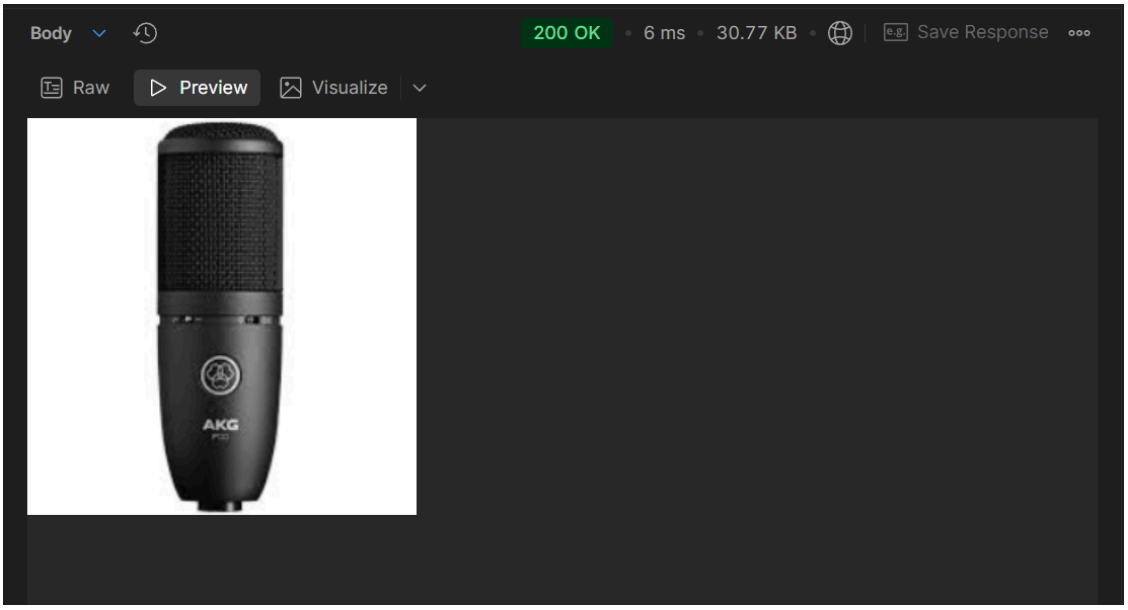
Реалізовано GET /inventory/ Повертає інформацію про конкретну річ Якщо річ не знайдена повертається 404 Not Found



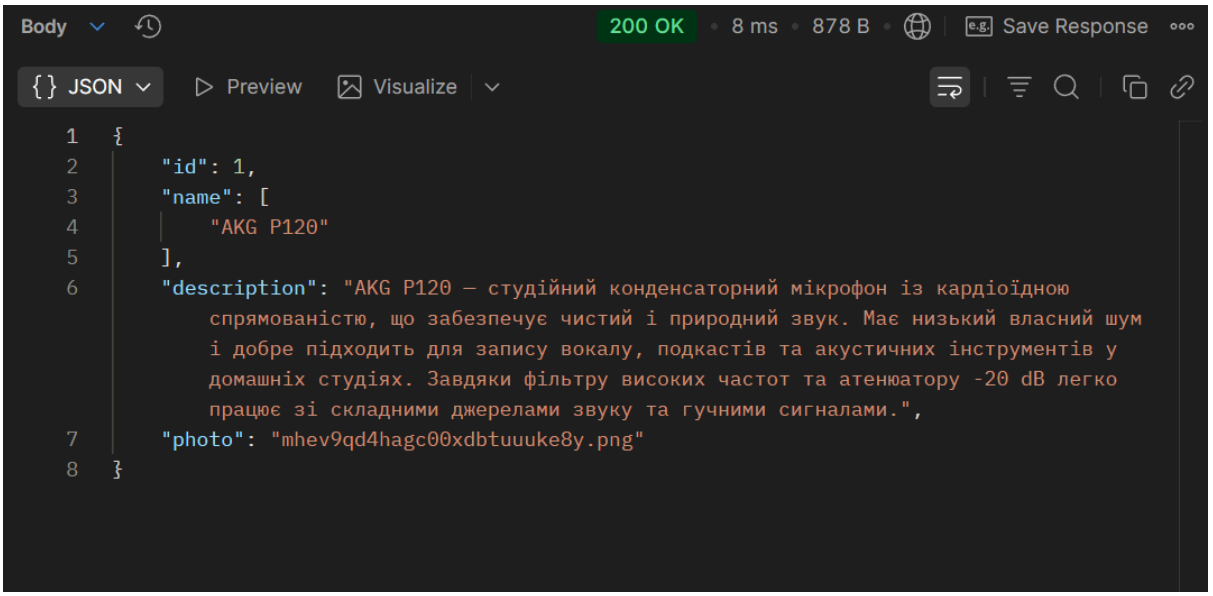
Реалізовано PUT /inventory/ Приймає JSON з новим ім’ям або описом Оновлює відповідні дані Якщо річ не існує повертається 404 Not Found



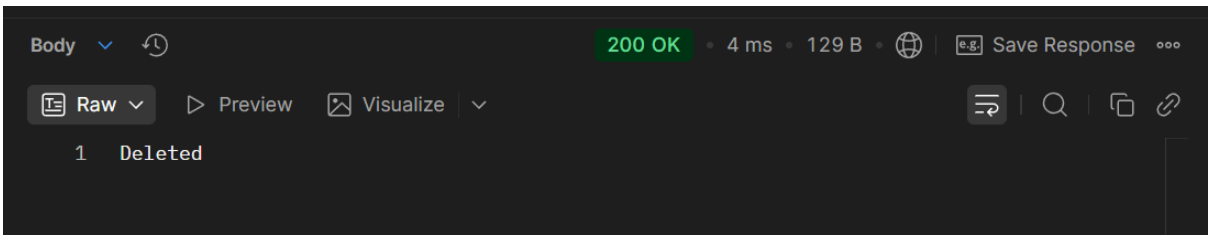
Реалізовано GET /inventory//photo Повертає зображення речі Додається заголовок Content-Type image/jpeg Якщо фото або річ не існує повертається 404 Not Found



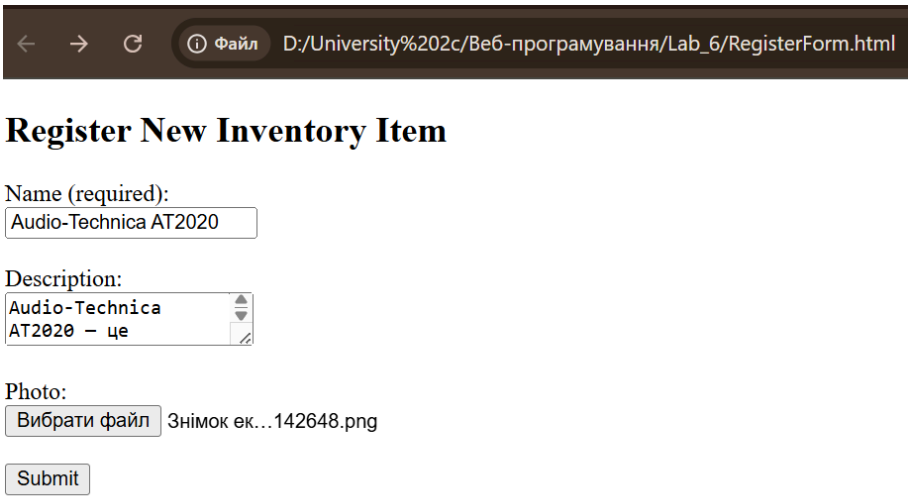
Реалізовано PUT /inventory//photo Приймає multipart form data з фото Оновлює файл фото у кеші Якщо річ не існує повертається 404 Not Found

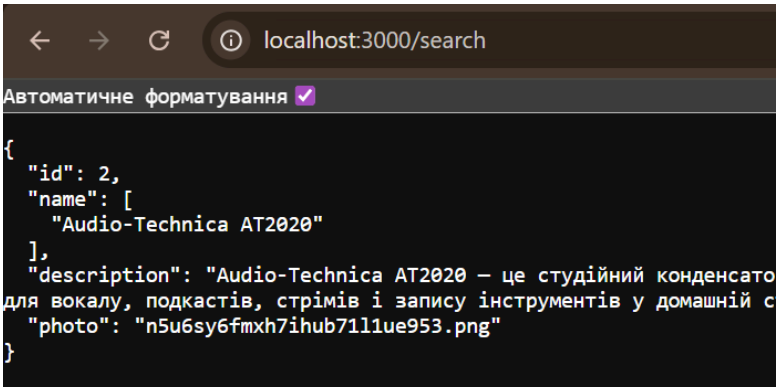


Реалізовано DELETE /inventory/ Видаляє річ зі списку Якщо ID неправильний повертається 404 Not Found

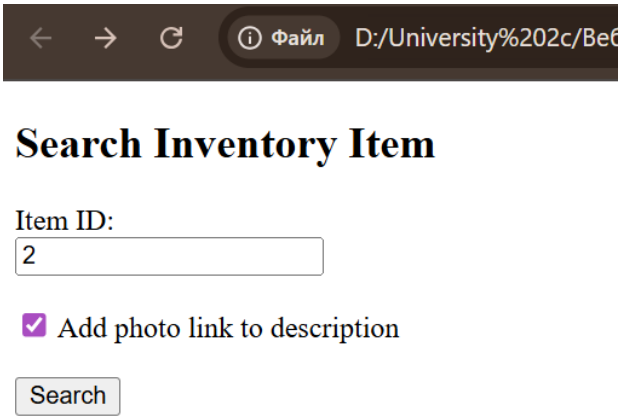


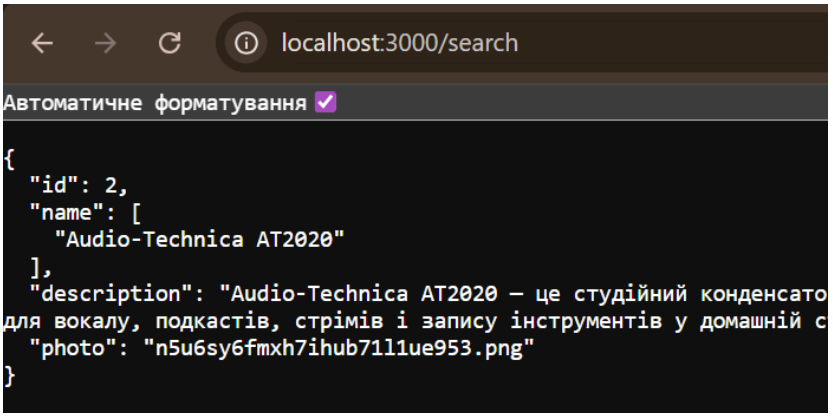
Реалізовано GET /RegisterForm.html та GET /SearchForm.html HTML форма реєстрації пристрою HTML форма пошуку за ID з прапорцем has_photo





Реалізовано POST /search Приймає дані у форматі x www form urlencoded Обробляє id та has_photo Повертає знайдений елемент або 404 Not Found Якщо прапорець увімкнений додає посилання на фото до опису

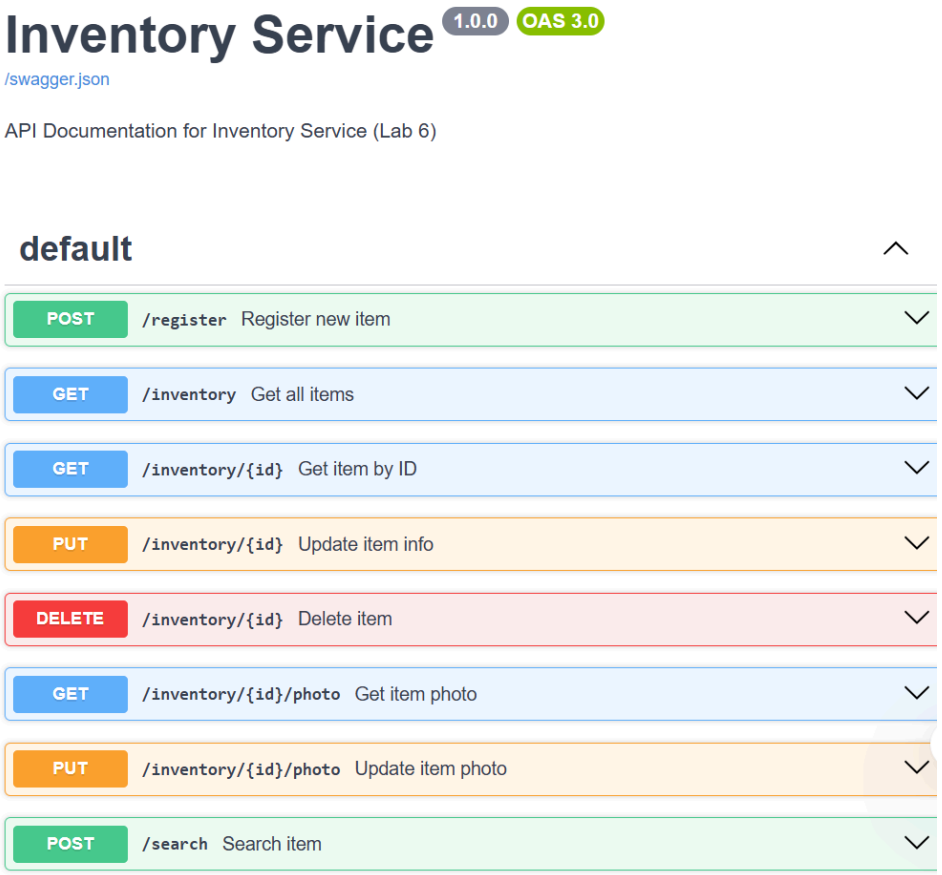




Додано обробку неправильних методів Для будь якого іншого методу повертається 405 Method Not Allowed

Частина 3 - тестування та документування сервісу

Додано Swagger документацію (/docs) Створено ендпоінт /swagger.json Створено інтерфейс Swagger UI на /docs



Docker підтримка Створено Dockerfile Перевірено запуск сервера всередині контейнера

Після реалізації кожного ендпоінта було виконано тестування та зроблено окремий коміт

Висновок: У ході виконання лабораторної роботи №6 було створено веб-сервіс інвентаризації з підтримкою різних типів HTTP-запитів, HTML-форм, пошуку, роботи з файлами та документації Swagger. Сервіс було повністю протестовано за допомогою Postman та запущено у Docker.

Під час виконання виникло дві основні проблеми. Перша — помилковий порядок методів у сервері, через що деякі маршрути перекривали інші (наприклад, /inventory/:id перехоплював запити до /inventory/:id/photo). Після впорядкування маршрутів сервер почав працювати коректно. Друга — Docker не запускався через помилку імпорту, оскільки Node.js у контейнері не розпізнавав деякі модулі. Проблему було вирішено шляхом додавання параметра "type": "module" у package.json, після чого контейнер успішно запустився.

Лабораторна робота виконана повністю, усі вимоги реалізовано.