Контейнеризація проєкту

# Підготовка проекту

## Структура проекту:

site\_project

├── add\_products.py

├── api

│ ├── \_\_init\_\_.py

│ ├── shop.py

│ └── users.py

├── app.py

├── create\_db.py

├── delete\_products.py

├── docker-compose.yaml

├── Dockerfile

├── instance

│ └── shop\_website.db

├── models

│ ├── \_\_init\_\_.py

│ ├── models.py

├── requirements.txt

├── resources

│ ├── css

│ │ ├── about-us.css

│ │ ├── base.css

│ │ ├── header.css

│ │ ├── register.css

│ │ └── shop.css

│ ├── images

│ │ ├── about-us.webp

│ │ ├── bottle.webp

│ │ ├── box.webp

│ │ ├── calendar.webp

│ │ ├── cube.webp

│ │ ├── icon-logo.png

│ │ ├── people-1.webp

│ │ ├── people-2.webp

│ │ ├── protein.webp

│ │ ├── simple-box.webp

│ │ ├── sweets1.webp

│ │ └── video.mp4

│ └── js

│ ├── adaptive\_navbar.js

│ ├── flash\_messages.js

│ └── validate\_checkout.js

├── routes

│ ├── admin.py

│ ├── \_\_init\_\_.py

│ ├── shop.py

│ └── users.py

├── templates

│ ├── about.html

│ ├── admin.html

│ ├── base.html

│ ├── cart.html

│ ├── feedback.html

│ ├── home.html

│ ├── login.html

│ ├── order\_details.html

│ └── register.html

└── utils

├── api\_login.py

├── \_\_init\_\_.py

## Налаштування Docker

### Dockerfile

FROM python:3.9-slim

WORKDIR /app

COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

COPY . .

EXPOSE 5000

CMD ["sh", "-c", "python create\_db.py && python add\_products.py && python app.py "]

### docker-compose.yml

services:

web:

build: . # Використовує існуючий Dockerfile

ports:

- "5000:5000" # Мапінг портів

volumes:

- data:/app/data # Для бази даних

env\_file:

- .env # Зчитування змінних середовища з файлу

restart: unless-stopped # Автоматичний перезапуск при збої

volumes:

data: # Створюж Volume для бази даних

## Налаштування бази даних

Не було модифіковано

# Тестування

## Збірка та запуск контейнера

Збірка образу

$ docker compose up –build

Результат:

[+] Building 1.8s (11/11) FINISHED

…

✔ Network site\_project\_default Created 0.2s

✔ Container site\_project-web-1 Created 0.1s

Запуск контейнера

$ docker compose start

Перевірка статусу

$ docker ps

CONTAINER ID IMAGE PORTS STATUS

6ffba646c20e site\_project-web 0.0.0.0:5000->5000/tcp, :::5000->5000/tcp Up 6 seconds

## Тестування endpoints

* Endpoint: /api/products
* Метод: GET
* Результат: 200 OK
* [Скріншот результату]

…

# Проблеми та їх вирішення

## Проблема з БД

Проблема: При створенні контейнера не запускалися скрипти які додавали товари та створювали бд

Рішення: Додано додатковий скрипт до Dockerfile

CMD ["sh", "-c", "python create\_db.py && python add\_products.py && python app.py "]

# Висновки

У ході виконання лабораторної роботи:

1. Успішно контейнеризовано Flask-додаток

2. Налаштовано постійне зберігання даних через Docker Volumes

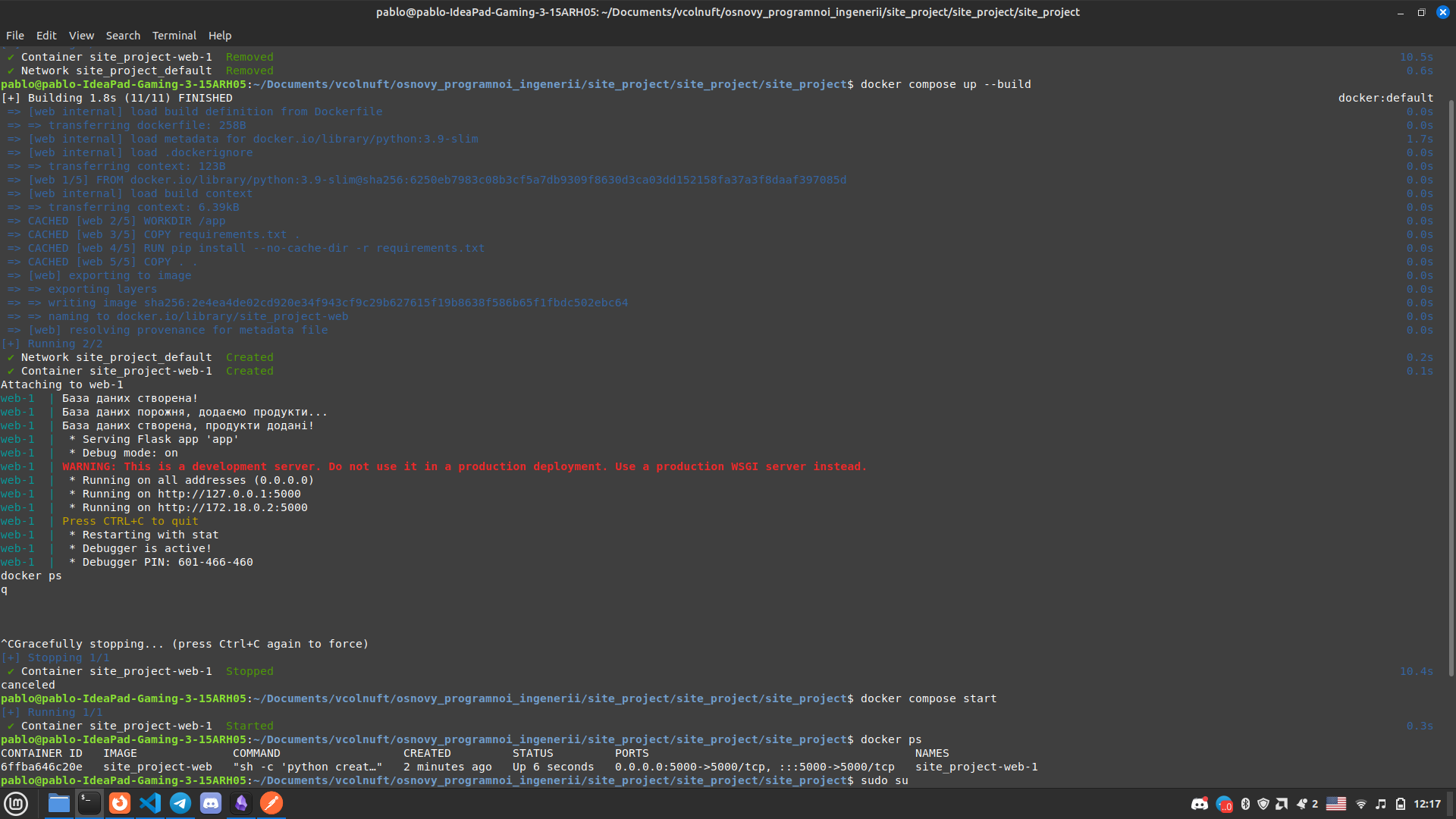
3. Реалізовано безпечну роботу з БД

4. Отримано практичні навички роботи з Docker

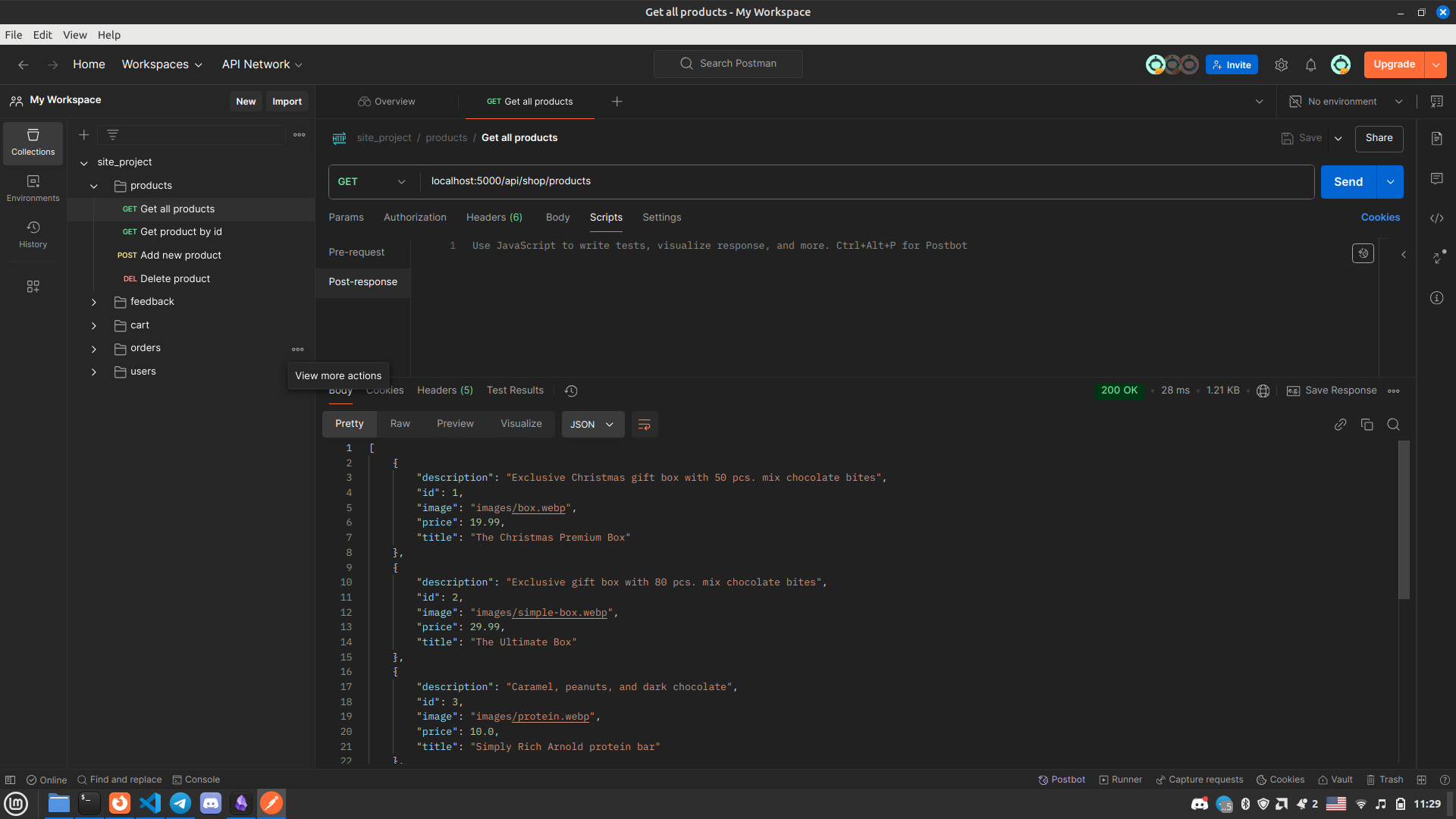
# Додатки

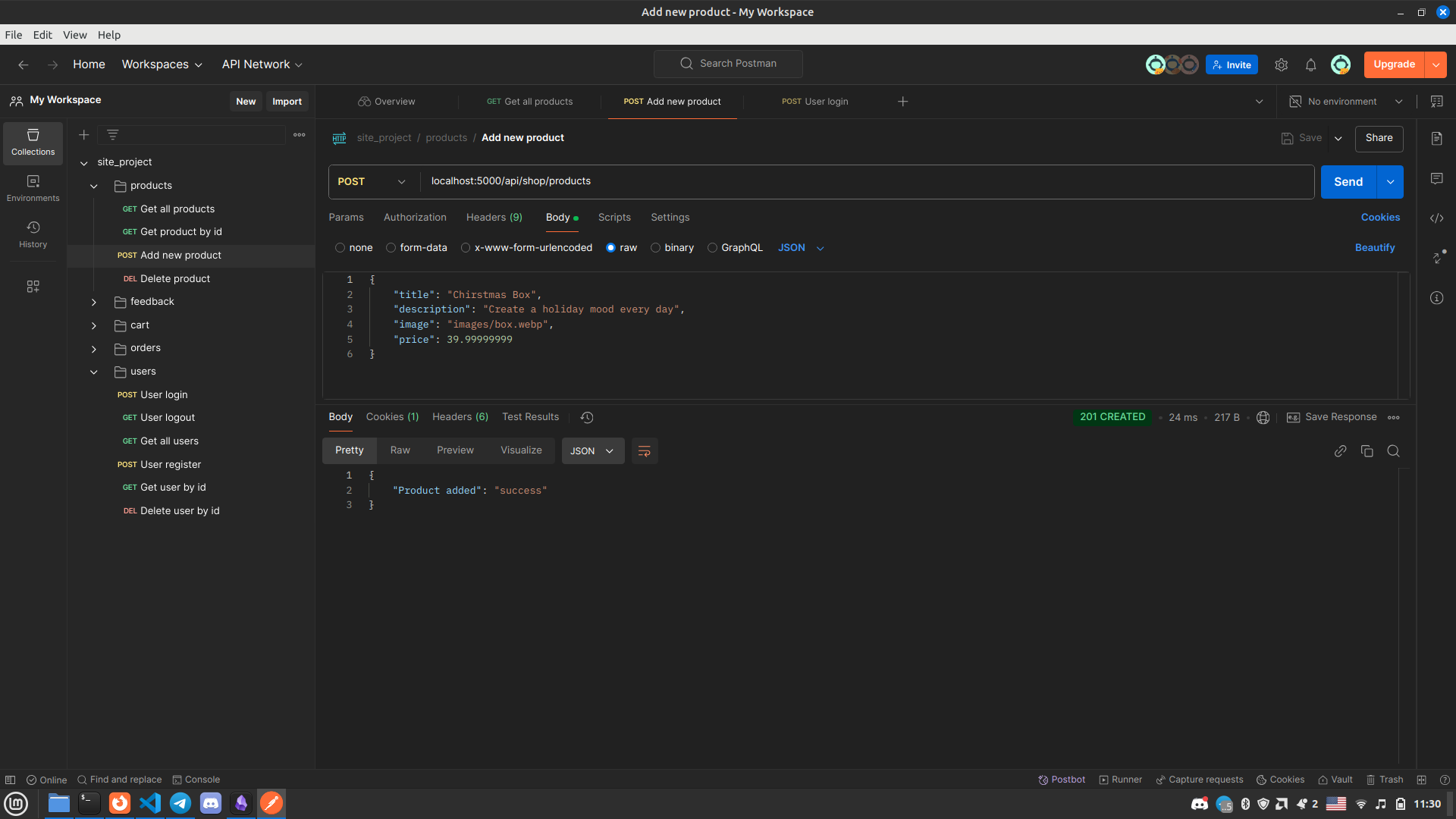
## Скріншоти роботи

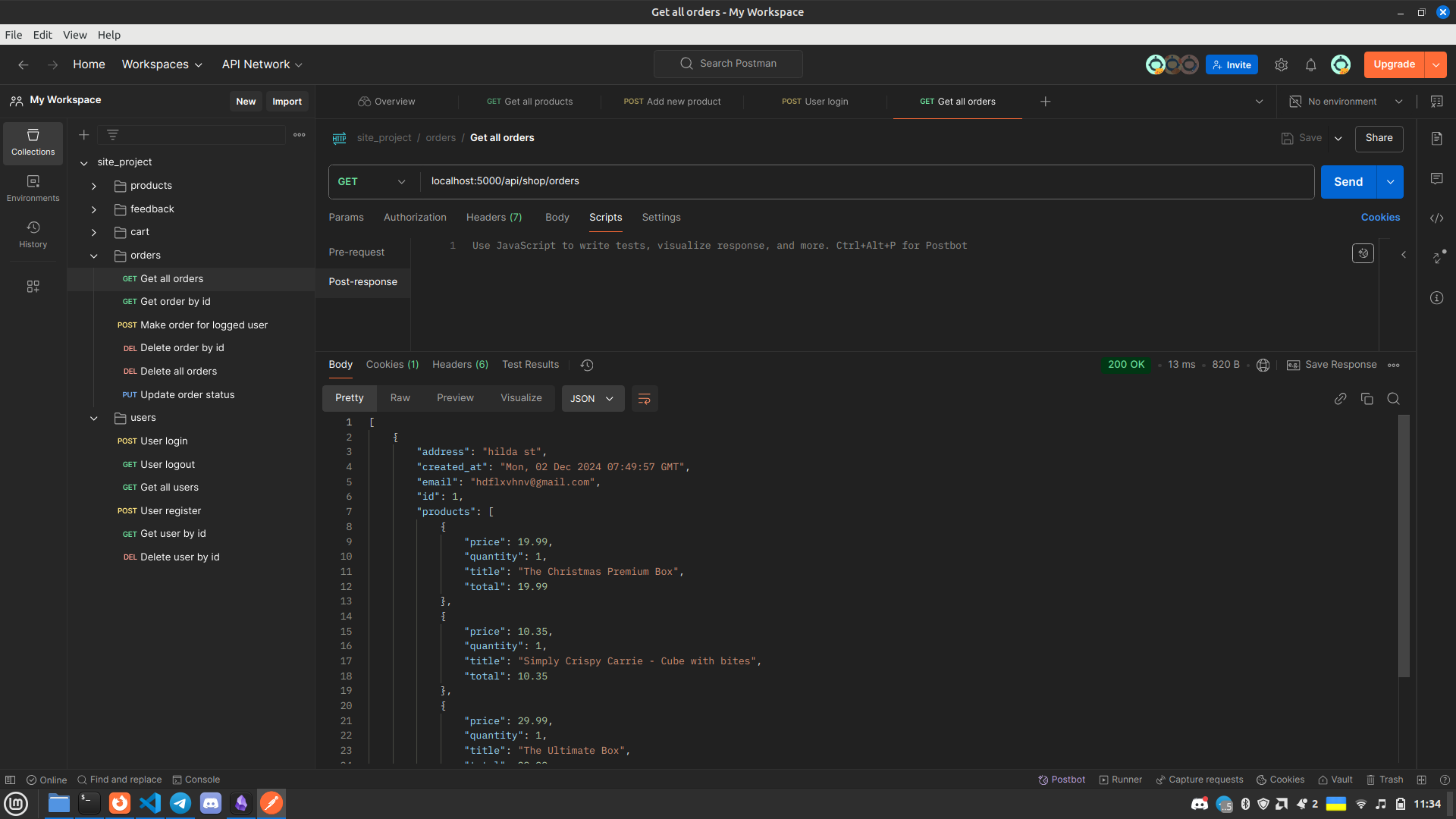
Процес збірки Docker образу

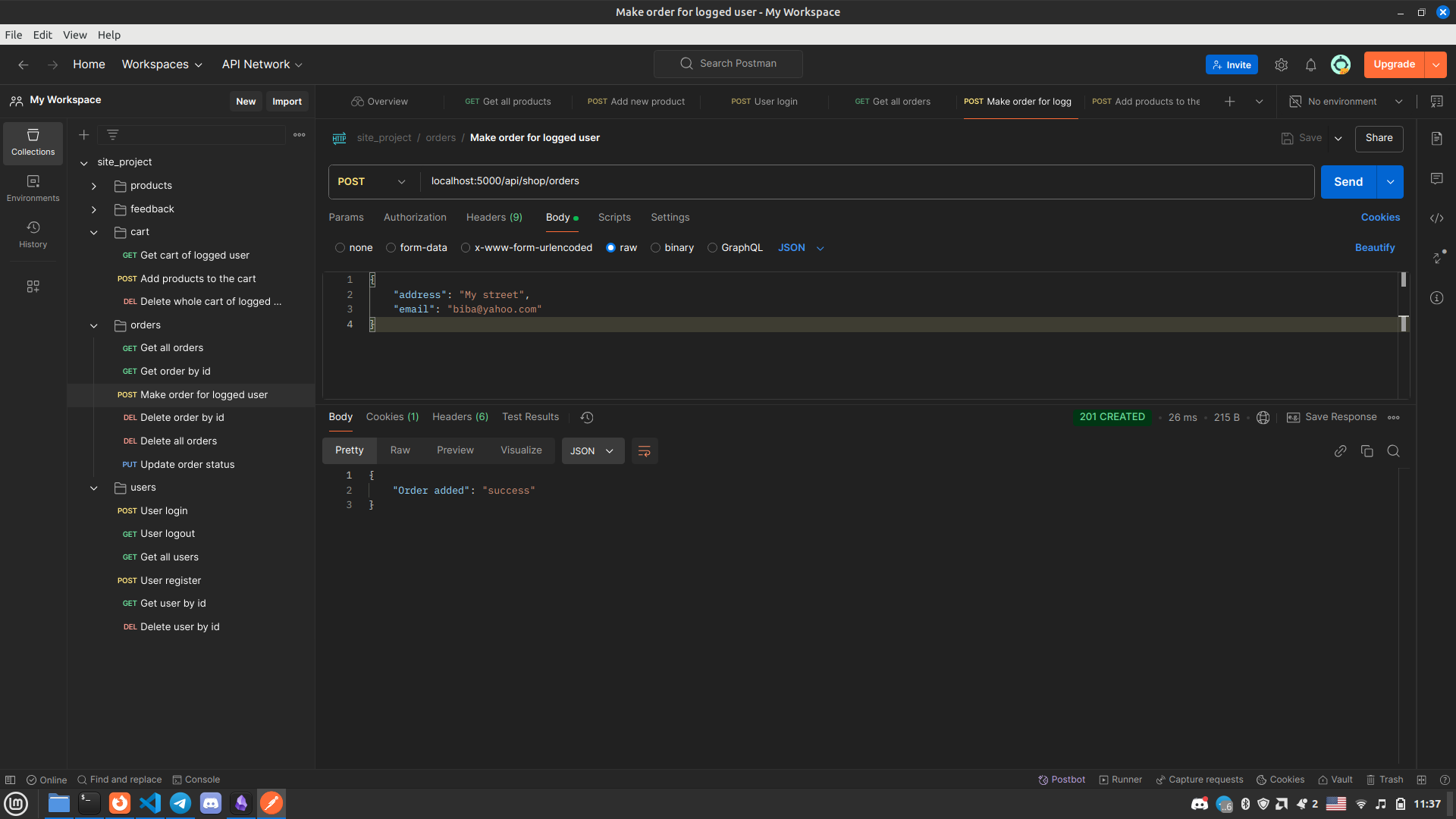


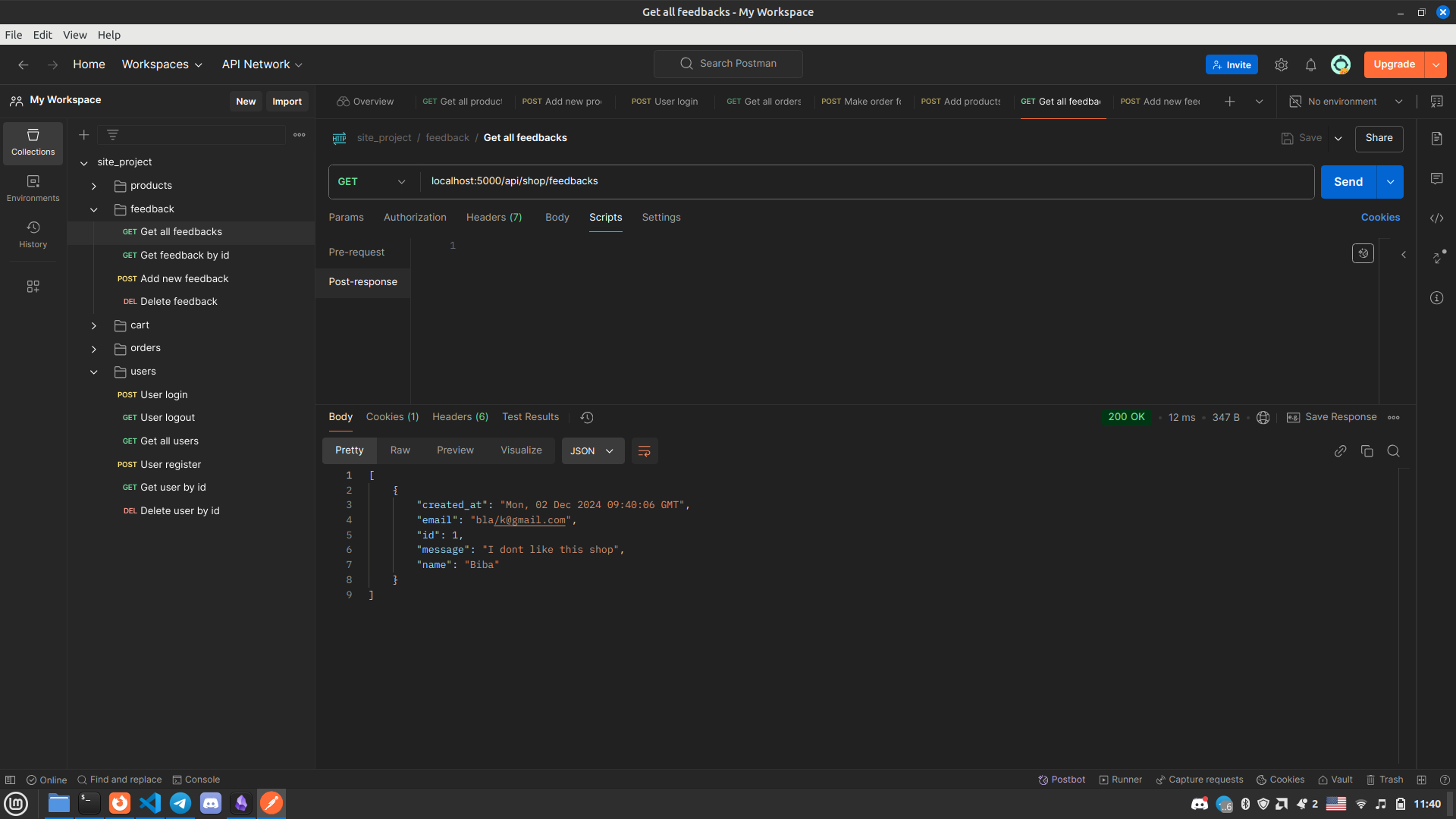
Тестування API endpoints

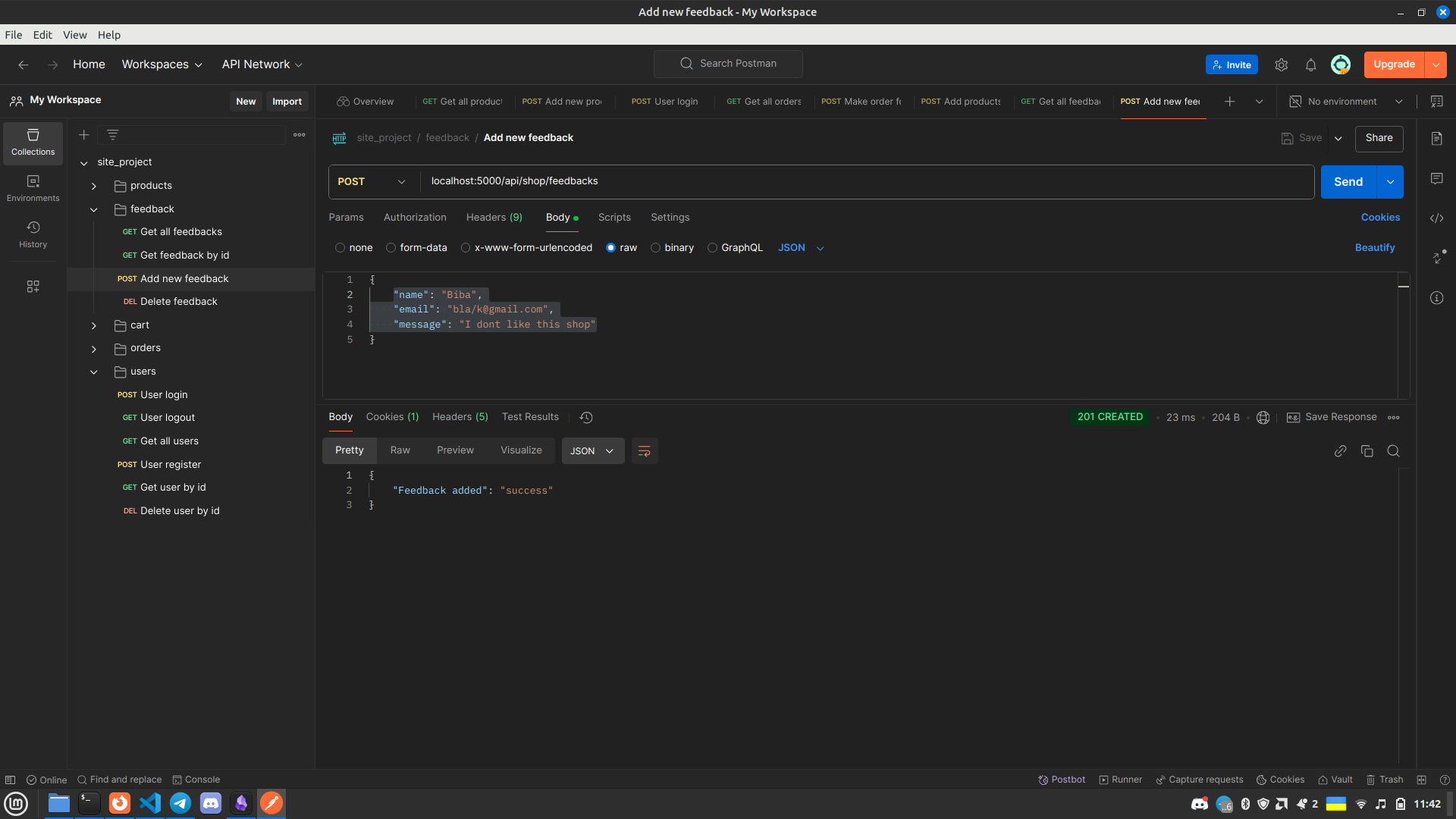


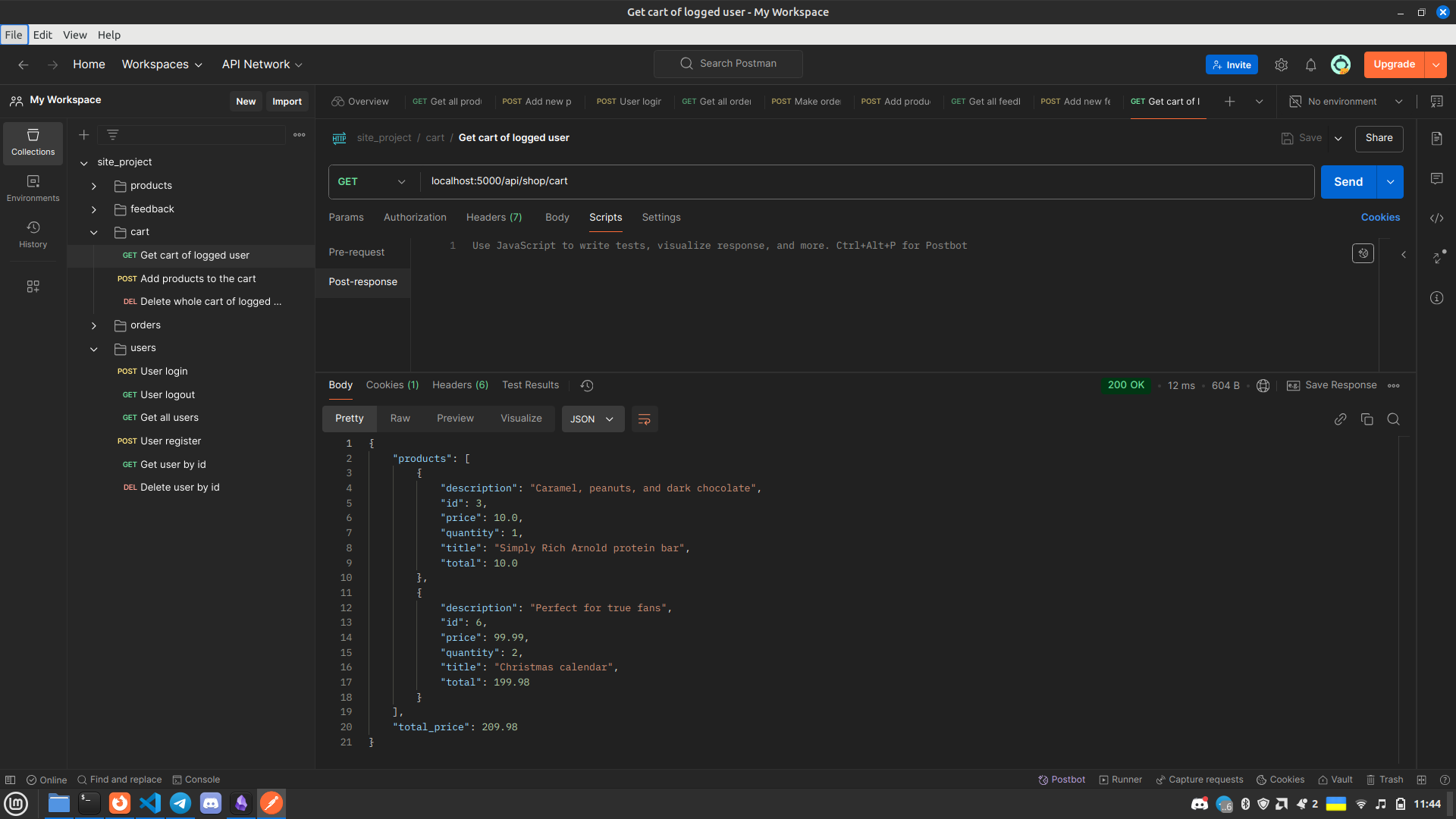


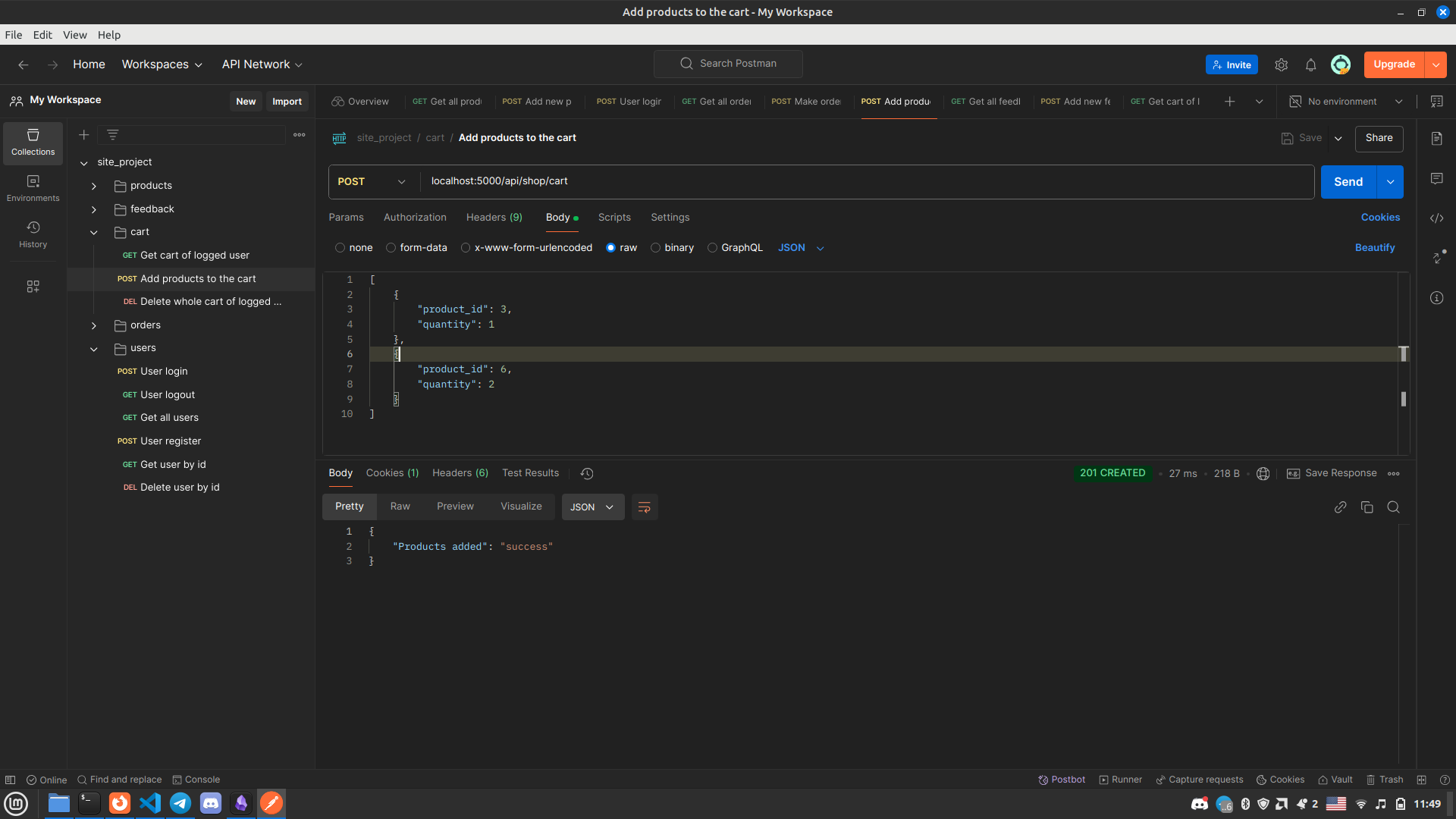


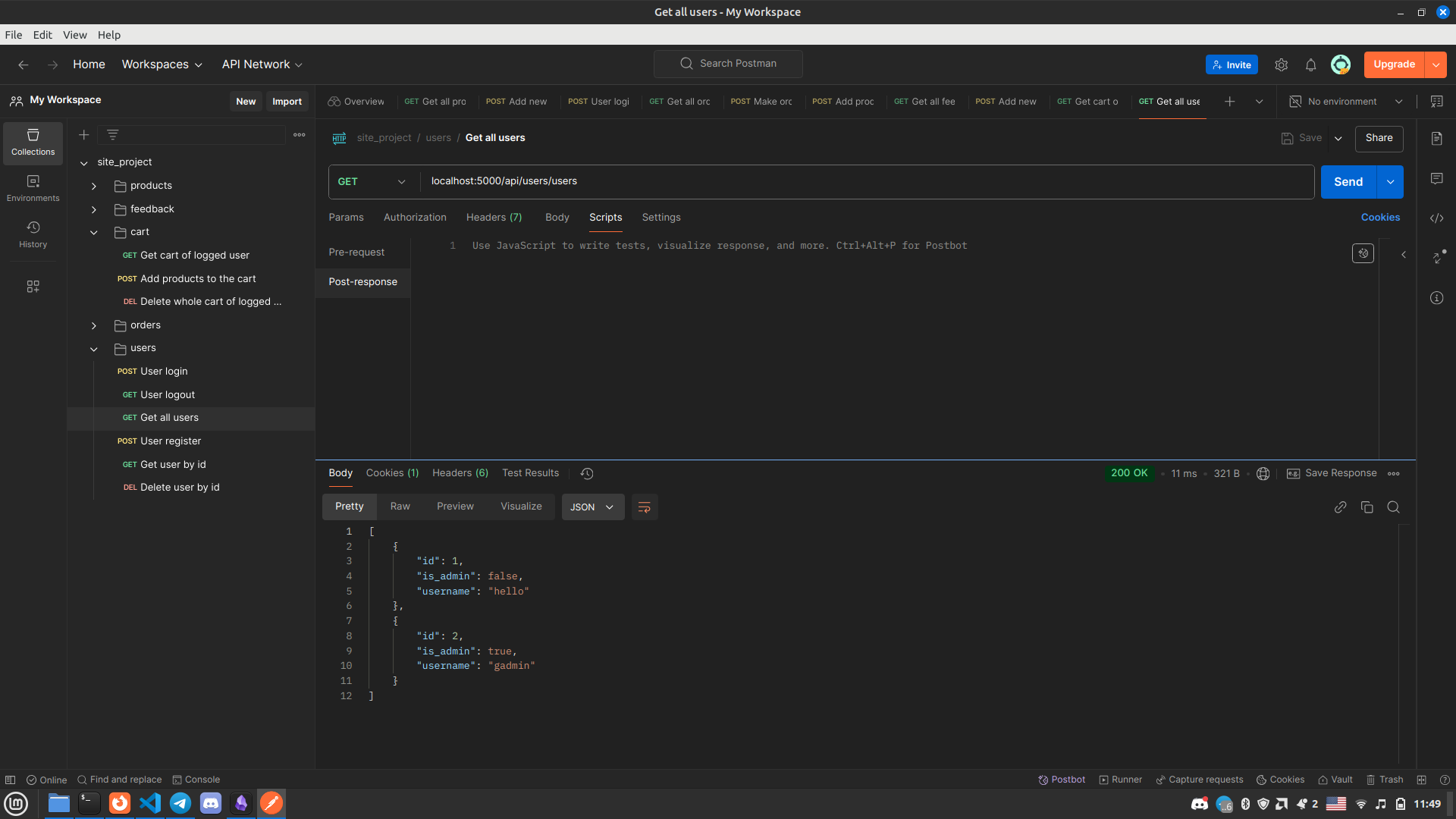


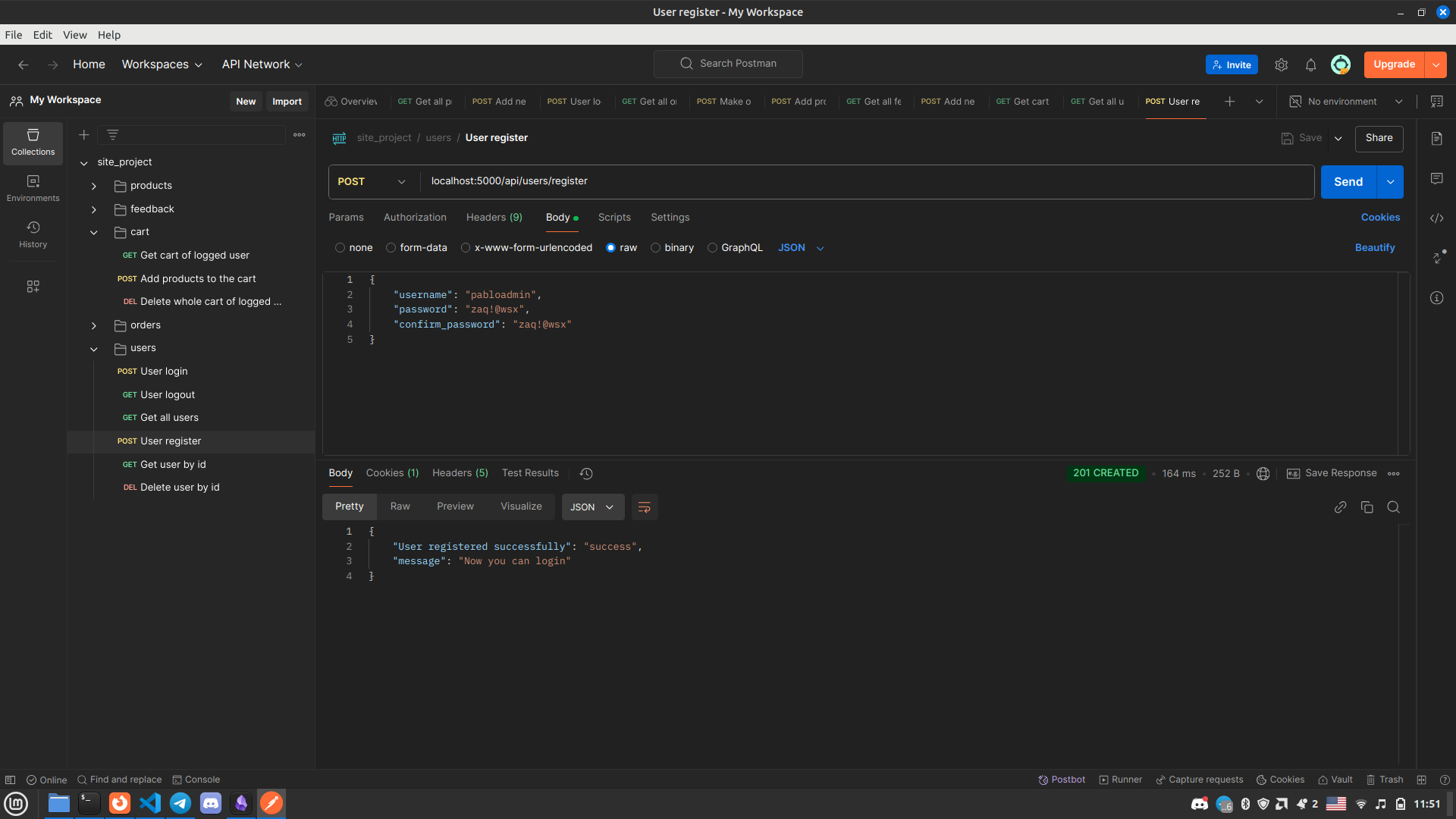


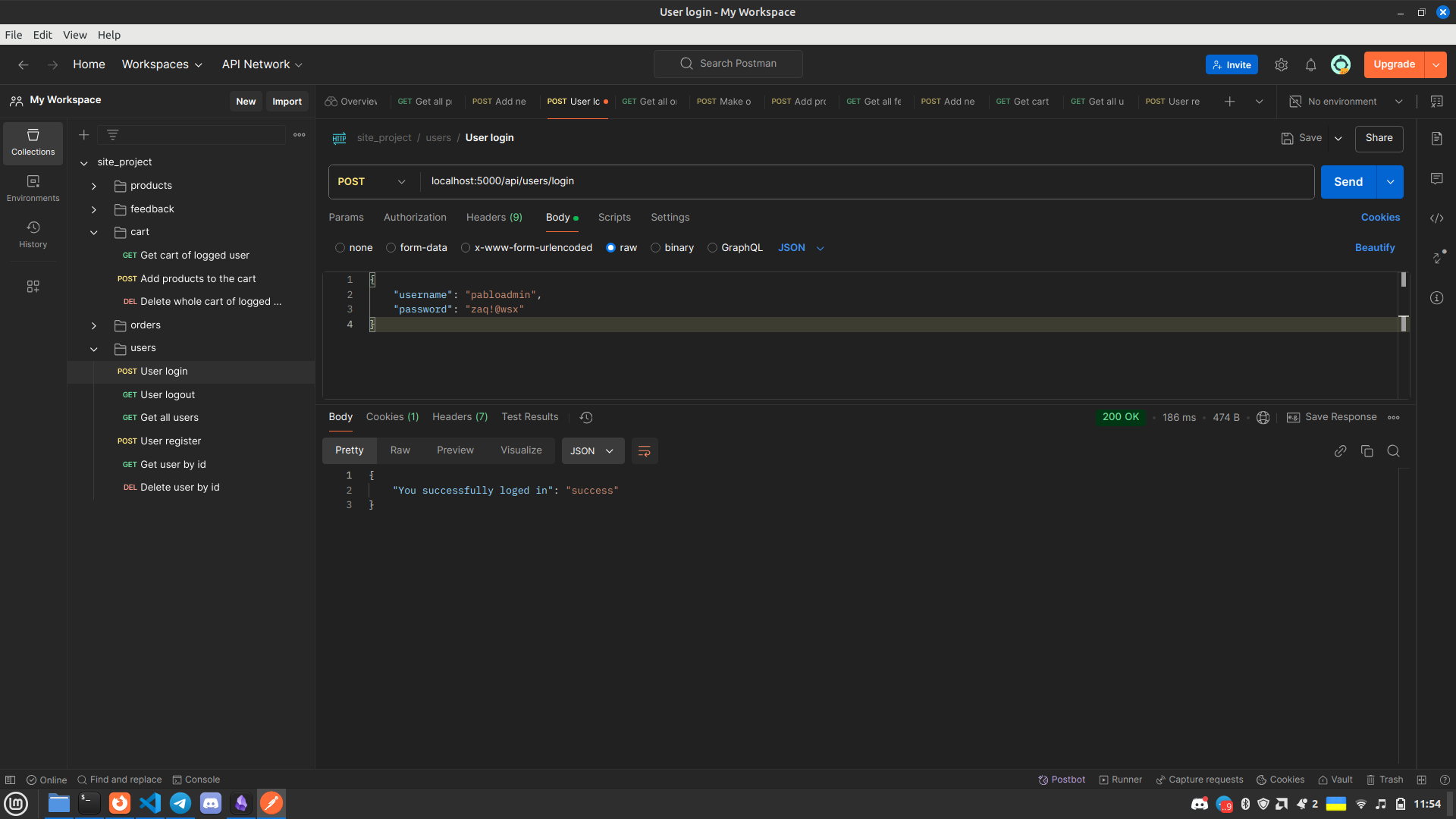


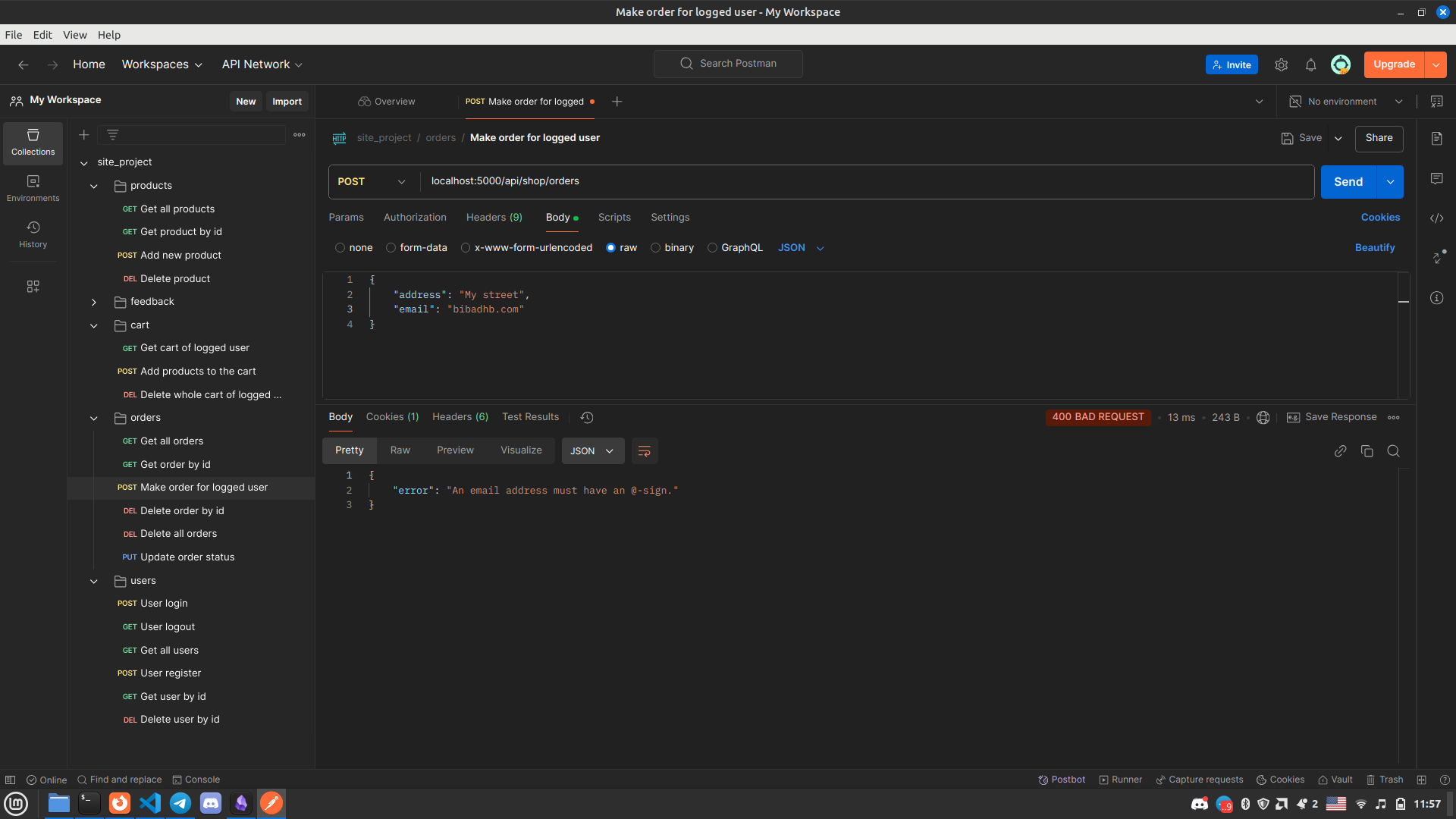


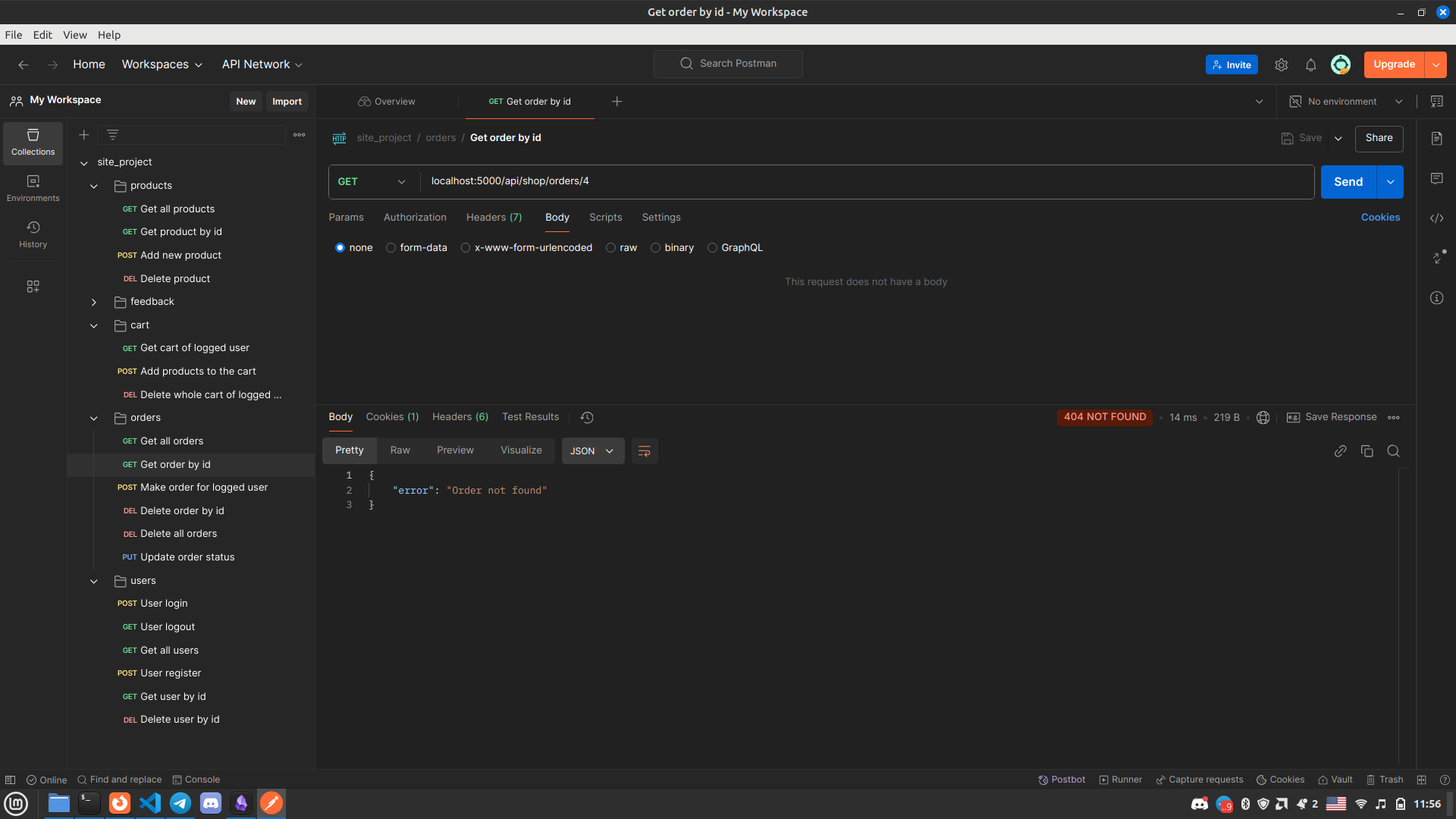












Перевірка даних у Docker Volume

