**MODELS.py CODE**

import sqlite3

def init\_db():

    conn = sqlite3.connect('database.db')

    cursor = conn.cursor()

    # Create the tables

    cursor.execute('''

    CREATE TABLE IF NOT EXISTS flavors (

        id INTEGER PRIMARY KEY,

        name TEXT NOT NULL,

        is\_seasonal BOOLEAN NOT NULL

    )''')

    cursor.execute('''

    CREATE TABLE IF NOT EXISTS inventory (

        id INTEGER PRIMARY KEY,

        ingredient TEXT NOT NULL,

        quantity INTEGER NOT NULL

    )''')

    cursor.execute('''

    CREATE TABLE IF NOT EXISTS allergens (

        id INTEGER PRIMARY KEY,

        name TEXT NOT NULL

    )''')

    cursor.execute('''

    CREATE TABLE IF NOT EXISTS cart (

        id INTEGER PRIMARY KEY,

        flavor\_id INTEGER NOT NULL,

        quantity INTEGER NOT NULL,

        FOREIGN KEY(flavor\_id) REFERENCES flavors(id)

    )''')

    conn.commit()

    conn.close()

if \_\_name\_\_ == "\_\_main\_\_":

    init\_db()

**app.py code**

from flask import Flask, request, jsonify

import sqlite3

app = Flask(\_\_name\_\_)

def connect\_db():

    return sqlite3.connect('database.db')

@app.route('/flavors', methods=['GET', 'POST'])

def manage\_flavors():

    conn = connect\_db()

    cursor = conn.cursor()

    if request.method == 'POST':

        data = request.get\_json()

        name = data.get('name')

        is\_seasonal = data.get('is\_seasonal', False)

        cursor.execute("INSERT INTO flavors (name, is\_seasonal) VALUES (?, ?)", (name, is\_seasonal))

        conn.commit()

        return jsonify({"message": "Flavor added successfully"}), 201

    elif request.method == 'GET':

        cursor.execute("SELECT \* FROM flavors")

        flavors = cursor.fetchall()

        return jsonify(flavors)

@app.route('/cart', methods=['POST'])

def add\_to\_cart():

    conn = connect\_db()

    cursor = conn.cursor()

    data = request.get\_json()

    flavor\_id = data.get('flavor\_id')

    quantity = data.get('quantity')

    cursor.execute("INSERT INTO cart (flavor\_id, quantity) VALUES (?, ?)", (flavor\_id, quantity))

    conn.commit()

    return jsonify({"message": "Item added to cart"}), 201

@app.route('/allergens', methods=['POST'])

def add\_allergen():

    conn = connect\_db()

    cursor = conn.cursor()

    data = request.get\_json()

    name = data.get('name')

    cursor.execute("INSERT INTO allergens (name) VALUES (?)", (name,))

    conn.commit()

    return jsonify({"message": "Allergen added successfully"}), 201

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

**Dockerfile**

FROM python:3.9-slim

WORKDIR /app

COPY . .

RUN pip install flask

CMD ["python", "app.py"]

**COMMANDS**

1. Python models.py
2. Python app.py
3. Build and run container  
    docker build -t ice-cream-parlor .

    docker run -p 5000:5000 ice-cream-parlor

1. Test the application

Add a Flavor:  Invoke-RestMethod -Uri http://127.0.0.1:5000/flavors -Method Post -Headers @{ "Content-Type"     ="application/json" } -Body '{"name": "Vanilla", "is\_seasonal": false}'

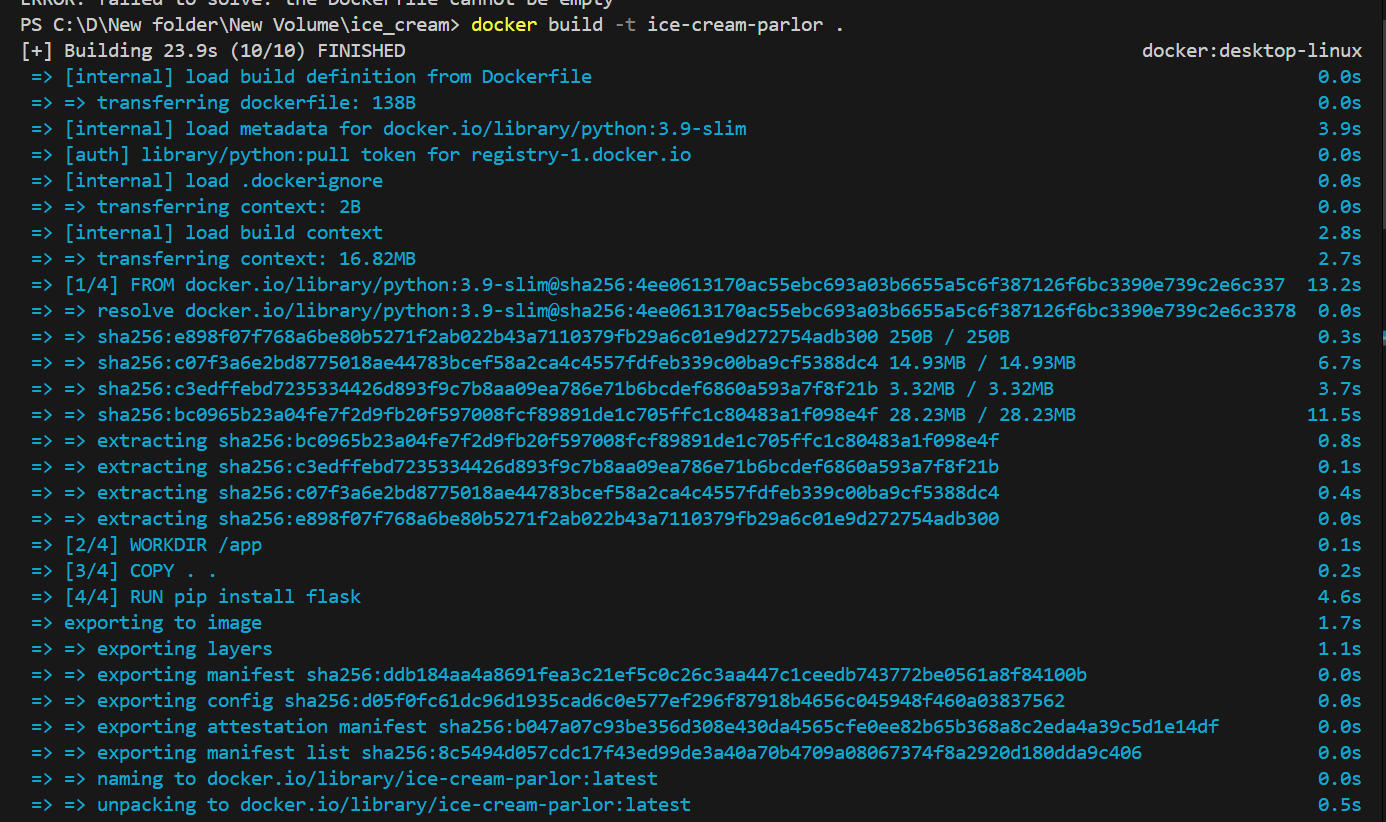
    View Flavors: curl http://127.0.0.1:5000/flavors

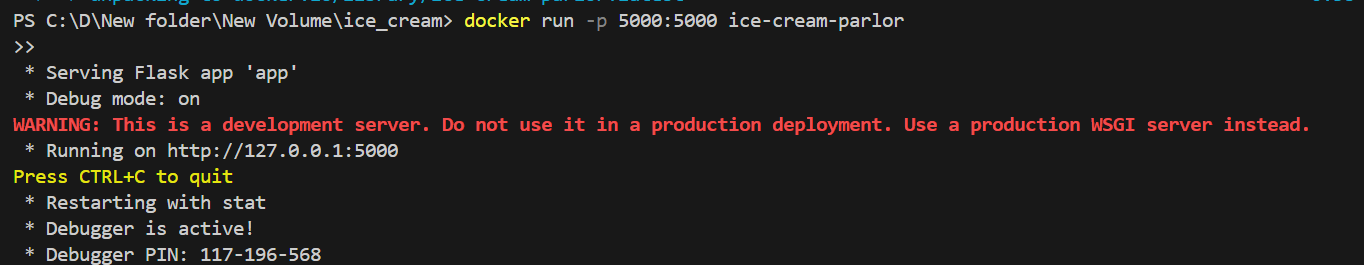
    Add to Cart: Invoke-RestMethod -Uri http://127.0.0.1:5000/cart -Method Post - Headers @{ "Content-Type" = "application/json" } -Body '{"flavor\_id": 1, "quantity": 2}'

    Add an Allergen: Invoke-WebRequest -Uri http://127.0.0.1:5000/allergens -Method Post -Headers @{ "Content-Type" = "application/json" } -Body '{"name": "Peanuts"}'

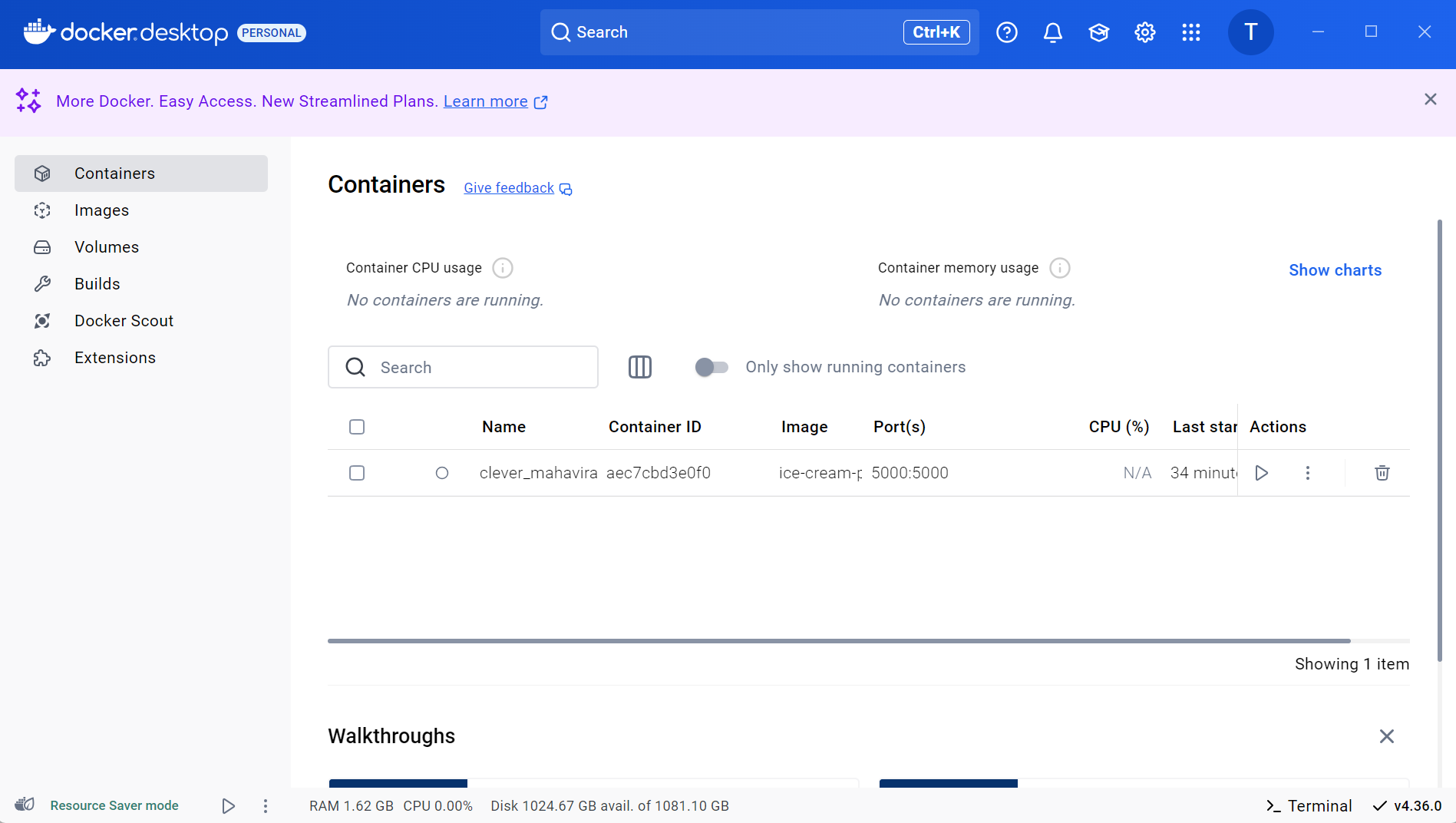
**Screen Shot**

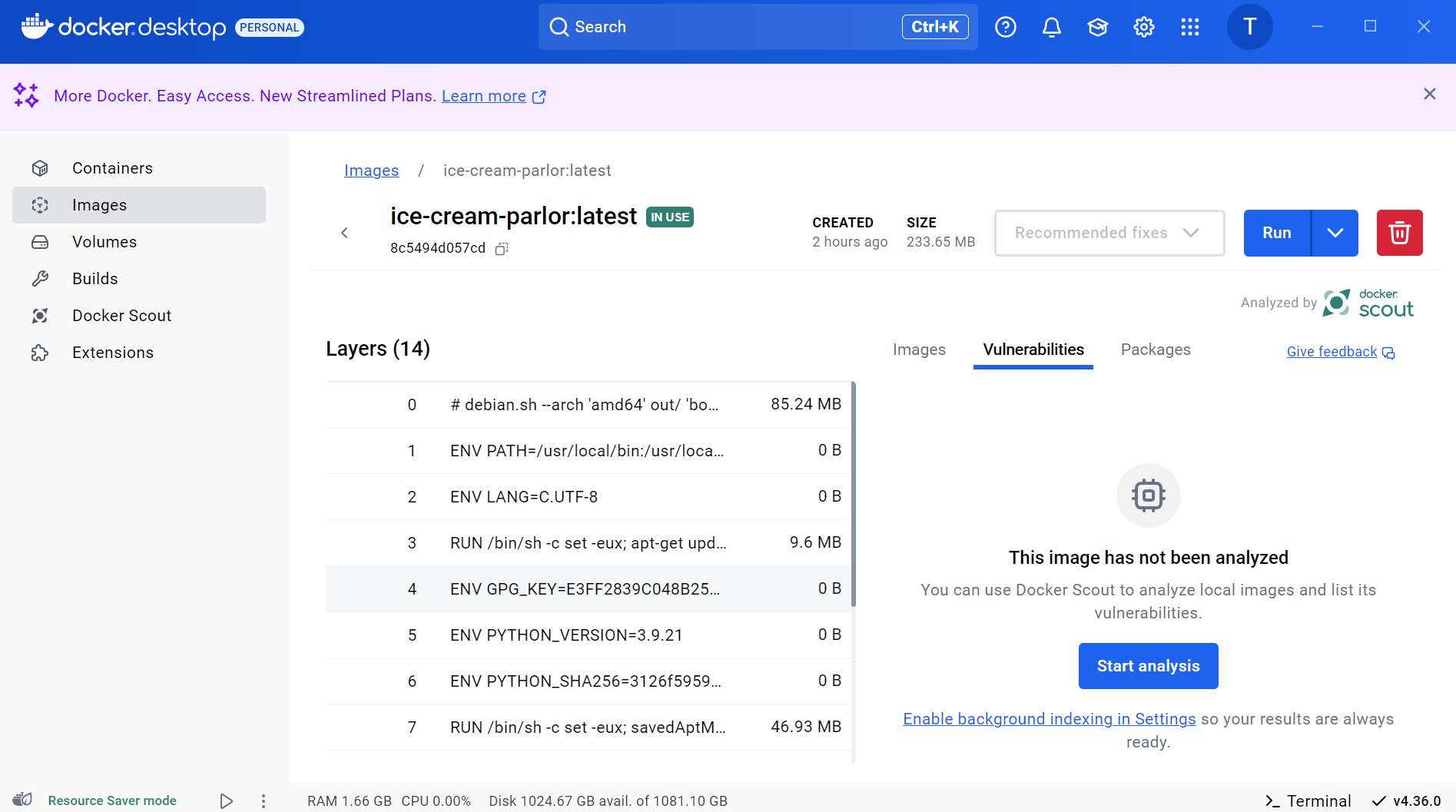
**Build and run container**

****

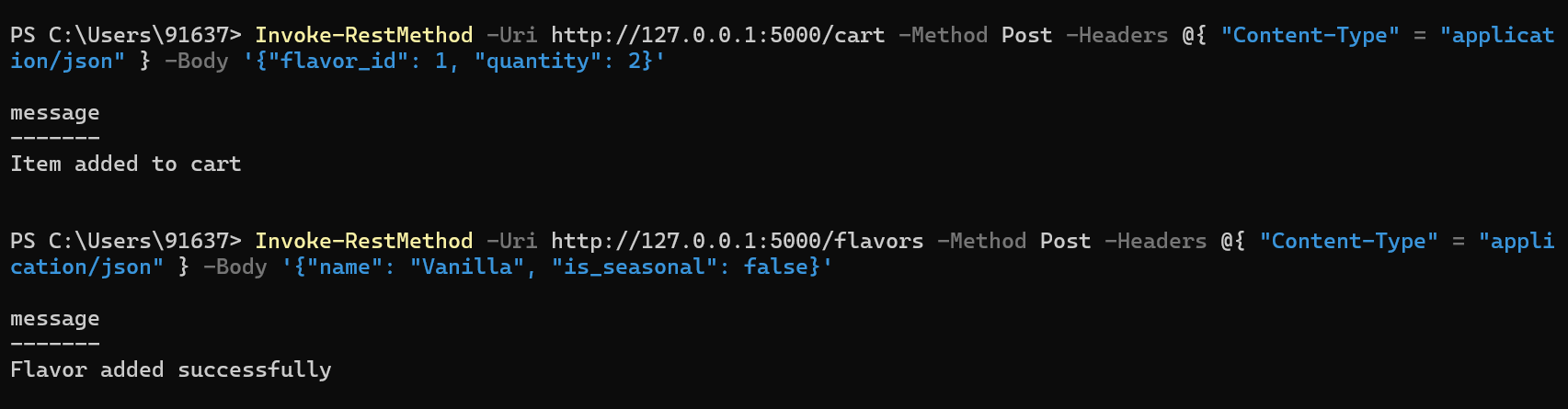
****

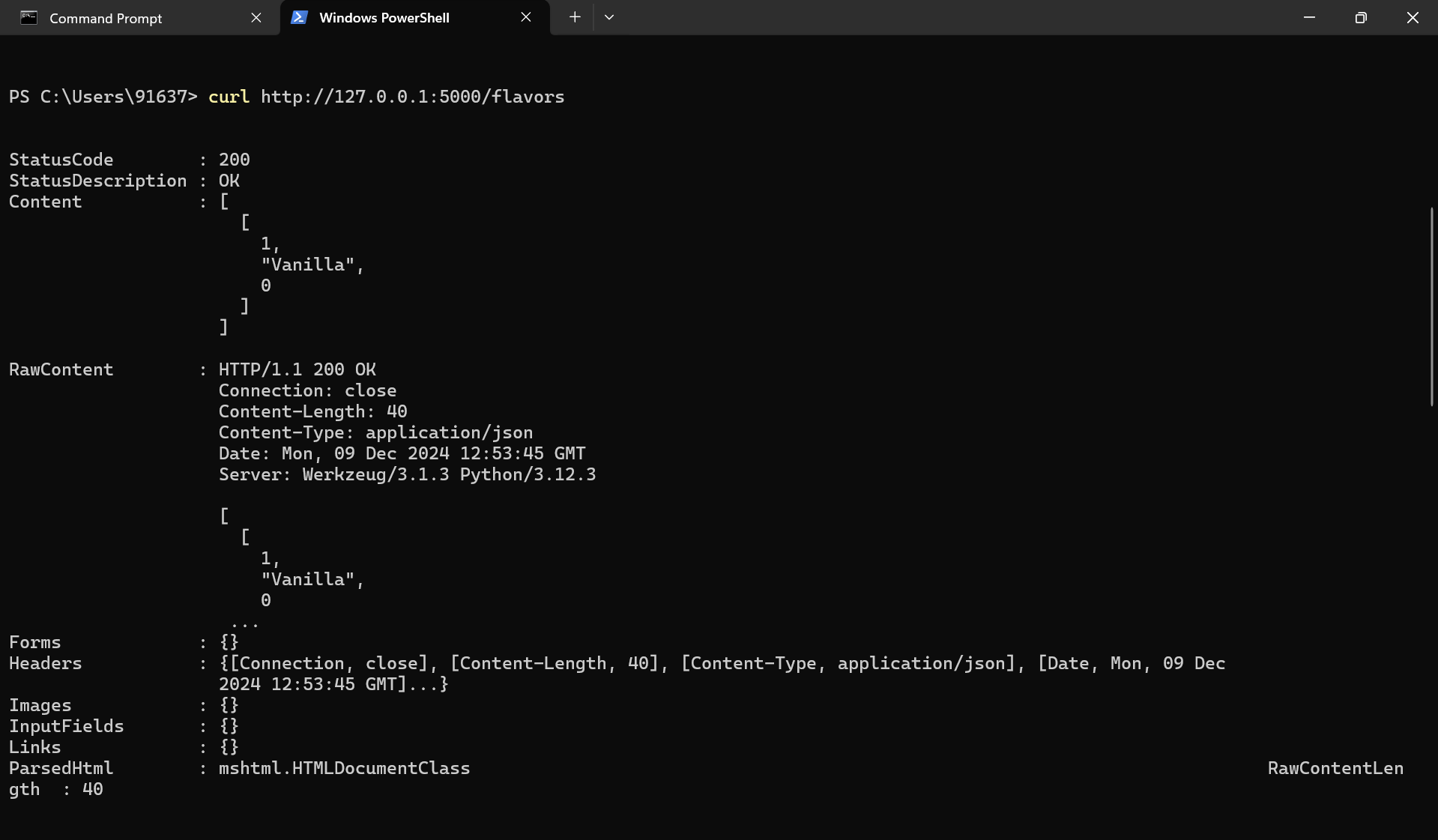
**Docker Setup**

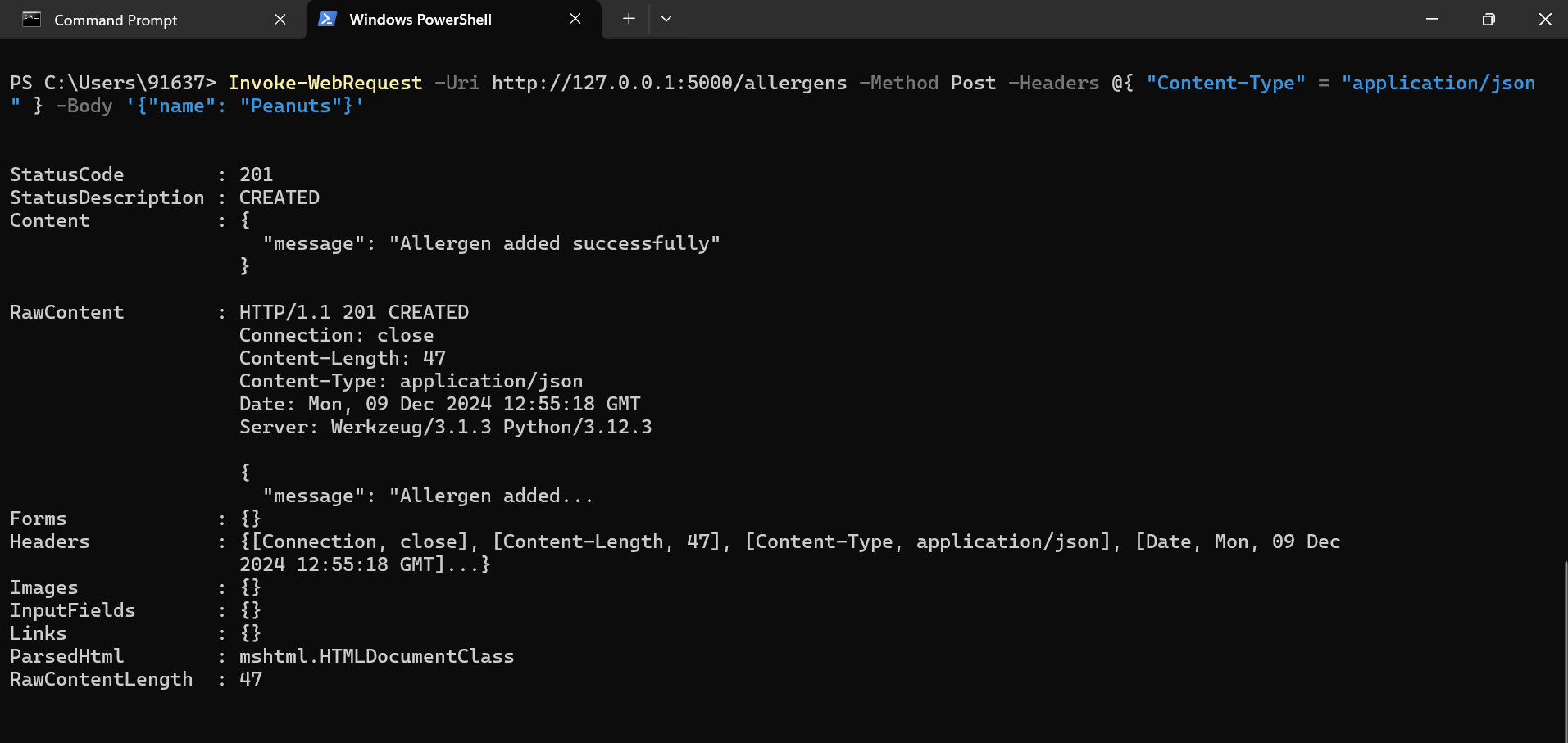


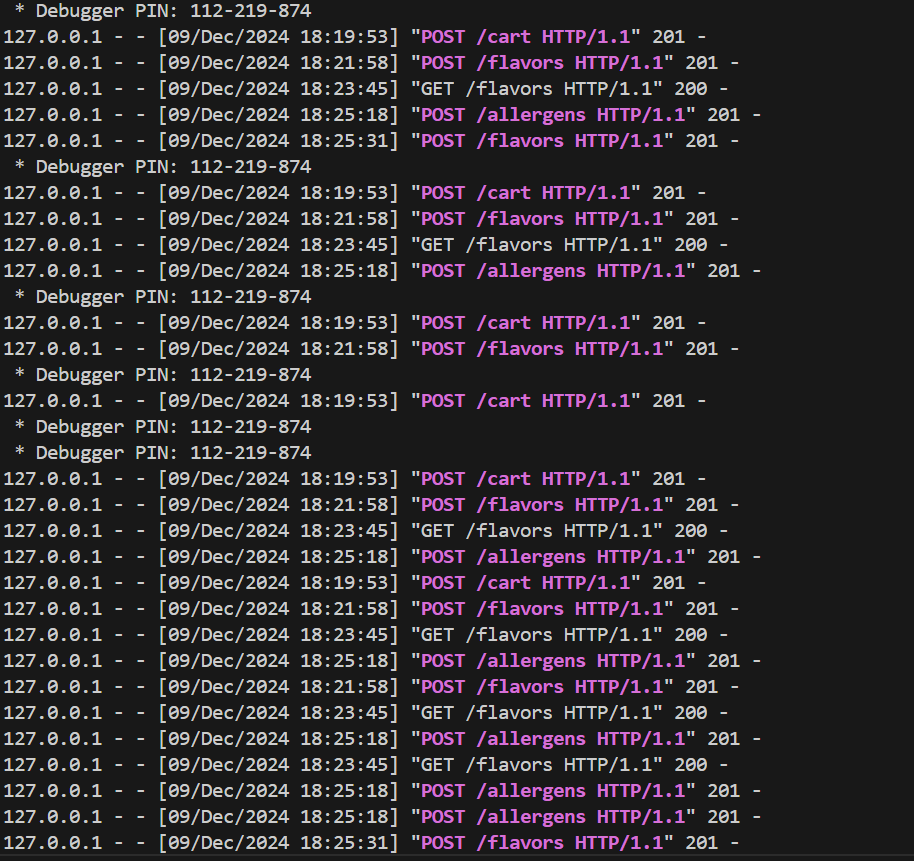


**Test the application**

****

****

****

****