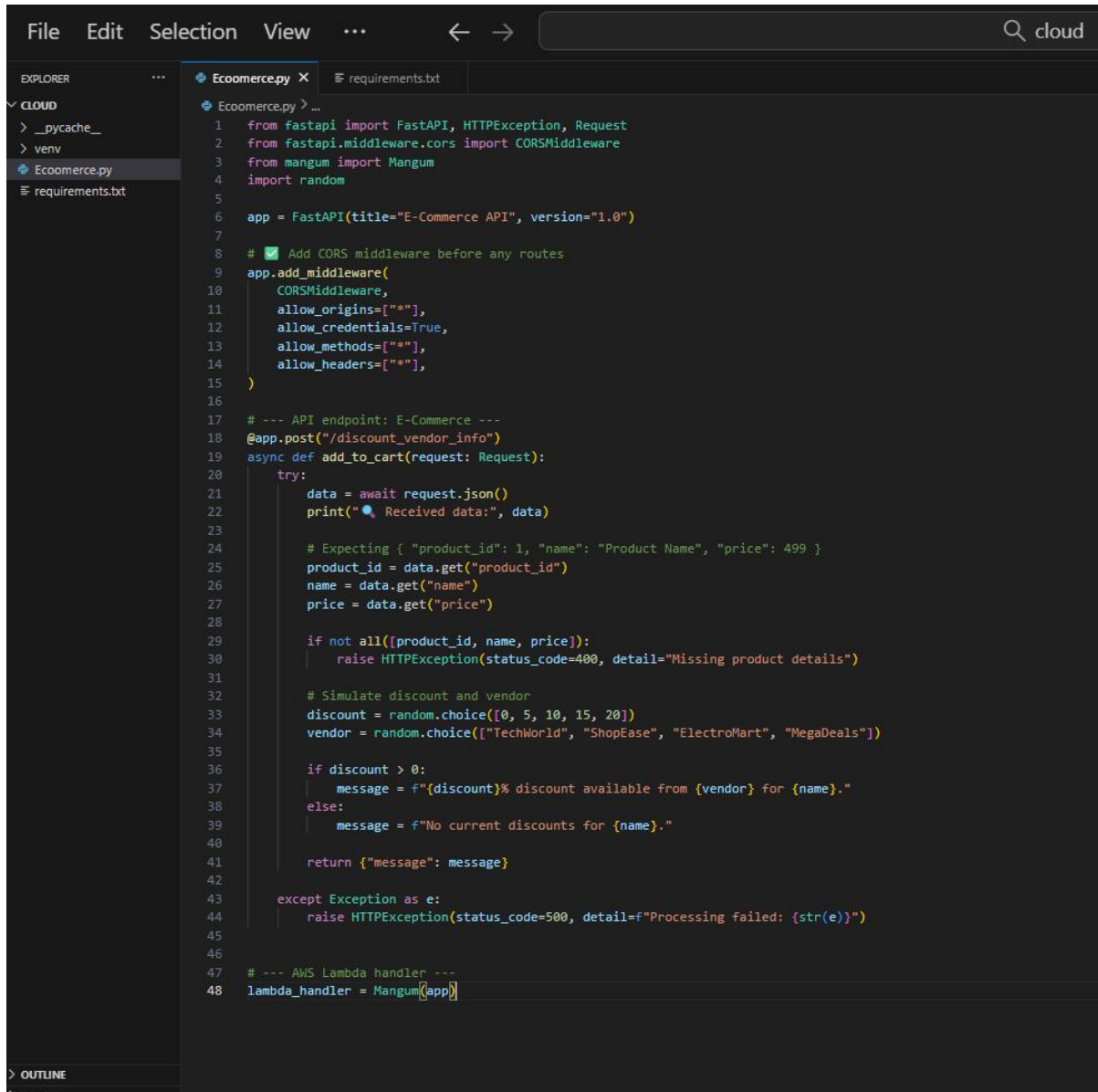


Name:S.Jewel Reddy

Reg No:22MID0161

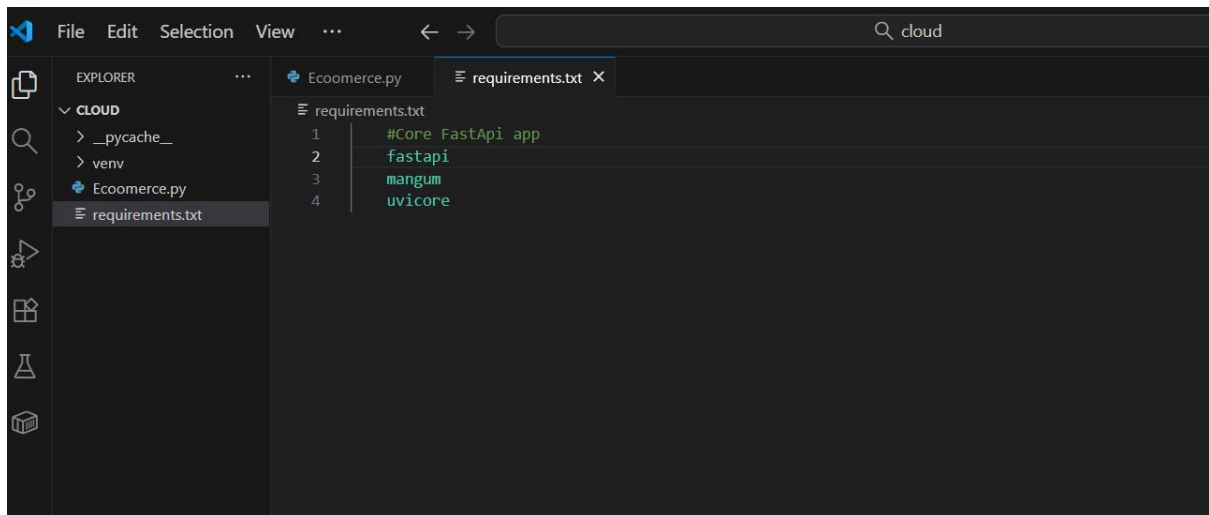
Cloud Demo:

Ecommerce file

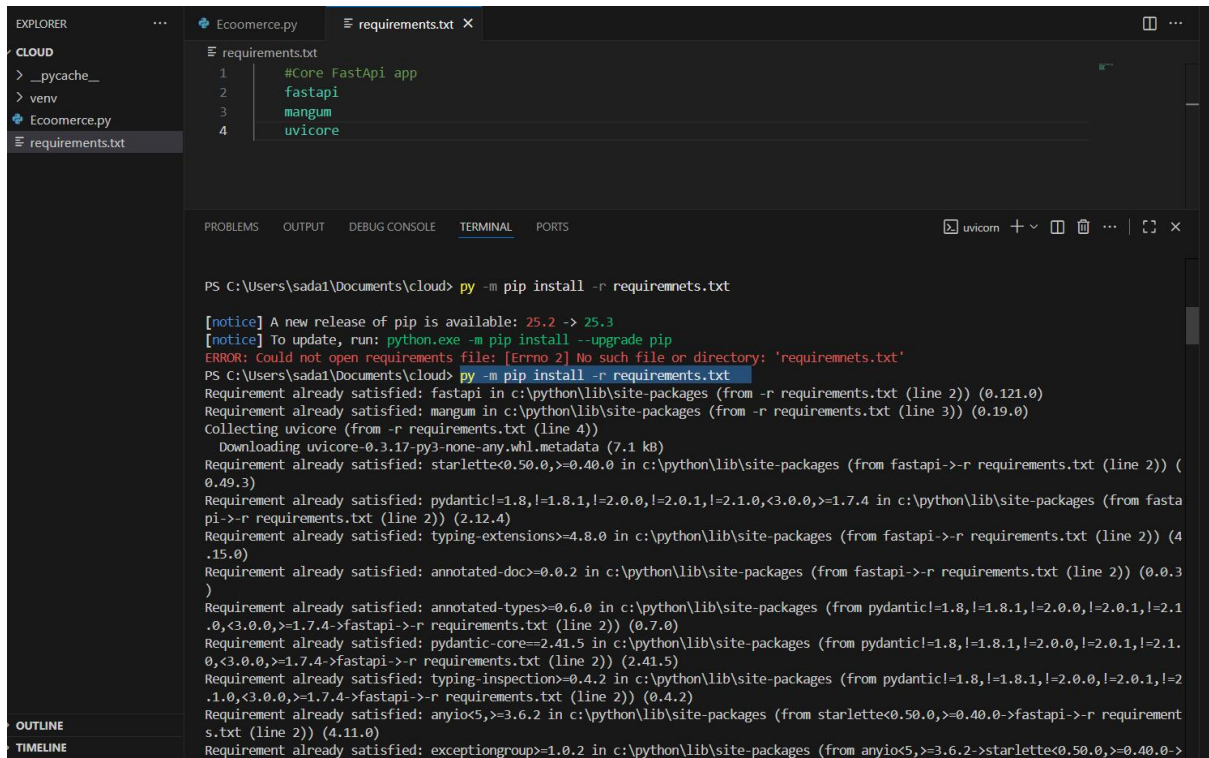


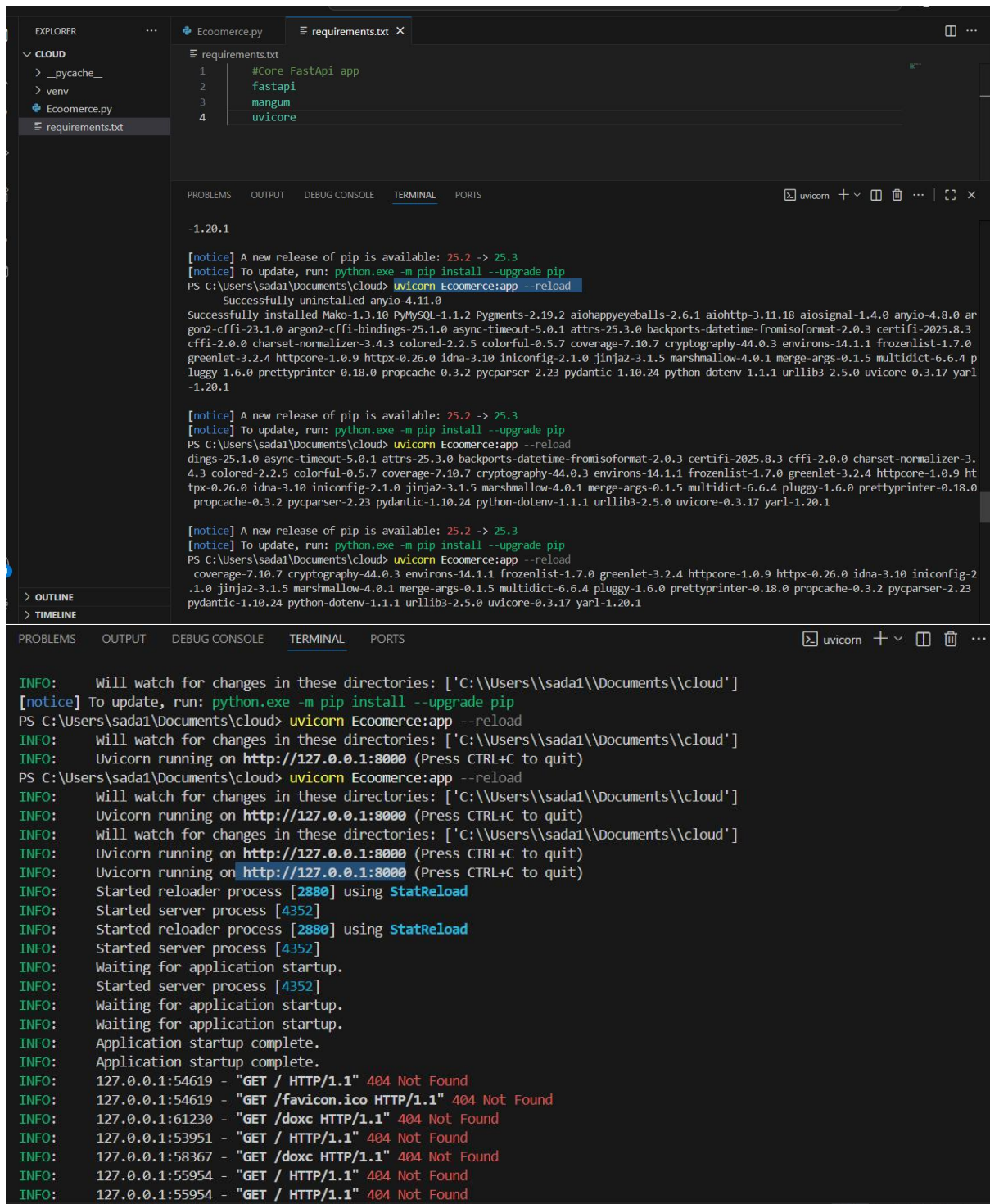
```
File Edit Selection View ... < > cloud
EXPLORER
  CLOUD
    > __pycache__
    > venv
    Ecommerce.py
    requirements.txt
Ecommerce.py > ...
requirements.txt
1 from fastapi import FastAPI, HTTPException, Request
2 from fastapi.middleware.cors import CORSMiddleware
3 from mangum import Mangum
4 import random
5
6 app = FastAPI(title="E-Commerce API", version="1.0")
7
8 # Add CORS middleware before any routes
9 app.add_middleware(
10     CORSMiddleware,
11     allow_origins=["*"],
12     allow_credentials=True,
13     allow_methods=["*"],
14     allow_headers=["*"],
15 )
16
17 # --- API endpoint: E-Commerce ---
18 @app.post("/discount_vendor_info")
19 async def add_to_cart(request: Request):
20     try:
21         data = await request.json()
22         print("Received data:", data)
23
24         # Expecting { "product_id": 1, "name": "Product Name", "price": 499 }
25         product_id = data.get("product_id")
26         name = data.get("name")
27         price = data.get("price")
28
29         if not all([product_id, name, price]):
30             raise HTTPException(status_code=400, detail="Missing product details")
31
32         # Simulate discount and vendor
33         discount = random.choice([0, 5, 10, 15, 20])
34         vendor = random.choice(["TechWorld", "ShopEase", "ElectroMart", "MegaDeals"])
35
36         if discount > 0:
37             message = f"{discount}% discount available from {vendor} for {name}."
38         else:
39             message = f"No current discounts for {name}."
40
41         return {"message": message}
42
43     except Exception as e:
44         raise HTTPException(status_code=500, detail=f"Processing failed: {str(e)}")
45
46
47 # --- AWS Lambda handler ---
48 lambda_handler = Mangum(app)
```

Requirements file:



Importing required files





```
requirements.txt
1 #Core FastApi app
2 fastapi
3 mangum
4 uvicorn

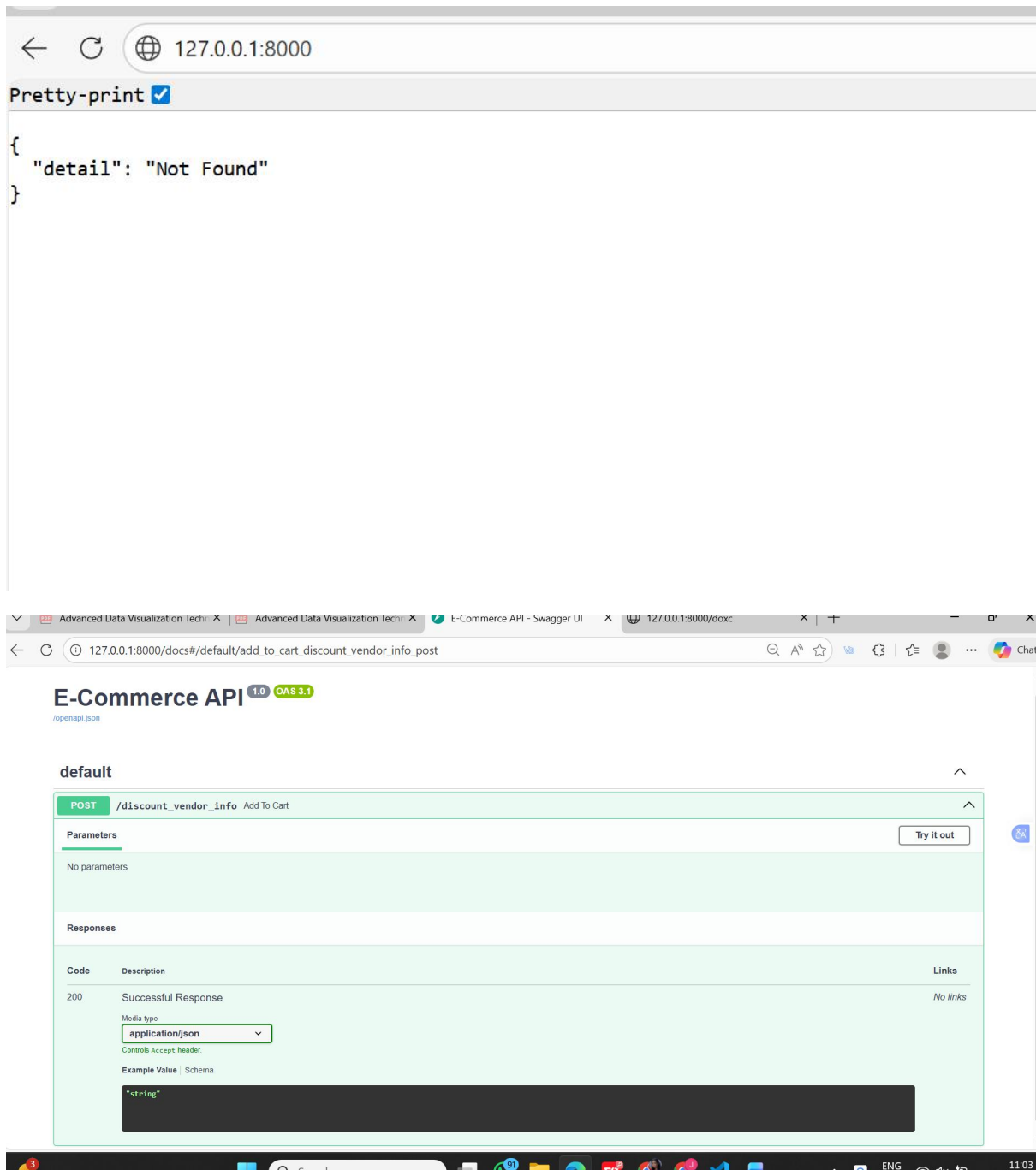
-1.20.1

[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\sada1\Documents\cloud> uvicorn Ecoommerce:app --reload
Successfully uninstalled anyio-4.11.0
Successfully installed Mako-1.3.10 PyMySQL-1.1.2 Pygments-2.19.2 aiohappyeyeballs-2.6.1 aiohttp-3.11.18 aiosignal-1.4.0 anyio-4.8.0 ar
gon2-cffi-23.1.0 argon2-cffi-bindings-25.1.0 async-timeout-5.0.1 attrs-25.3.0 backports-datetime-fromisoformat-2.0.3 certifi-2025.8.3
cffi-2.0.0 charset-normalizer-3.4.3 colored-2.2.5 colorlog-6.9.0 coverage-7.10.7 cryptography-44.0.3 environs-14.1.1 frozenlist-1.7.0
greenlet-3.2.4 httpcore-1.0.9 httpx-0.26.0 idna-3.10 iniconfig-2.1.0 jinja2-3.1.5 marshmallow-4.0.1 merge-args-0.1.5 multidict-6.6.4 p
luggy-1.6.0 prettyprinter-0.18.0 propcache-0.3.2 pycparser-2.23 pydantic-1.10.24 python-dotenv-1.1.1 urllib3-2.5.0 uvicorn-0.3.17 yar
l-1.20.1

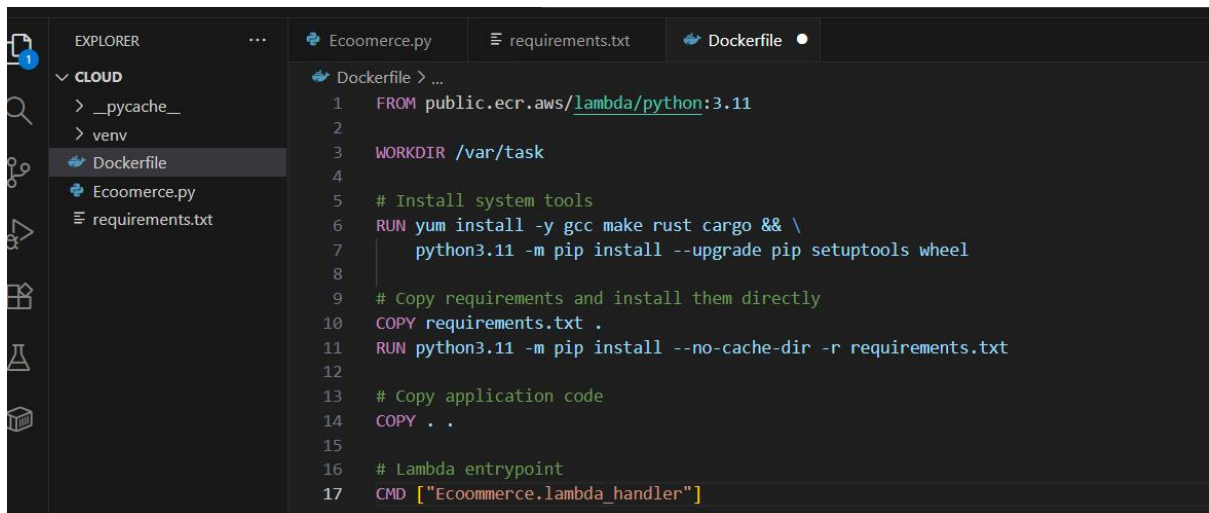
[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\sada1\Documents\cloud> uvicorn Ecoommerce:app --reload
dings-25.1.0 async-timeout-5.0.1 attrs-25.3.0 backports-datetime-fromisoformat-2.0.3 certifi-2025.8.3 cffi-2.0.0 charset-normalizer-3.
4.3 colored-2.2.5 colorlog-6.9.0 coverage-7.10.7 cryptography-44.0.3 environs-14.1.1 frozenlist-1.7.0 greenlet-3.2.4 httpcore-1.0.9 ht
tpx-0.26.0 idna-3.10 iniconfig-2.1.0 jinja2-3.1.5 marshmallow-4.0.1 merge-args-0.1.5 multidict-6.6.4 pluggy-1.6.0 prettyprinter-0.18.0
propcache-0.3.2 pycparser-2.23 pydantic-1.10.24 python-dotenv-1.1.1 urllib3-2.5.0 uvicorn-0.3.17 yarl-1.20.1

[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\sada1\Documents\cloud> uvicorn Ecoommerce:app --reload
coverage-7.10.7 cryptography-44.0.3 environs-14.1.1 frozenlist-1.7.0 greenlet-3.2.4 httpcore-1.0.9 ht
tpx-0.26.0 idna-3.10 iniconfig-2.1.0 jinja2-3.1.5 marshmallow-4.0.1 merge-args-0.1.5 multidict-6.6.4 pluggy-1.6.0 prettyprinter-0.18.0
propcache-0.3.2 pycparser-2.23 pydantic-1.10.24 python-dotenv-1.1.1 urllib3-2.5.0 uvicorn-0.3.17 yarl-1.20.1

INFO: Will watch for changes in these directories: ['C:\\Users\\sada1\\Documents\\cloud']
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\sada1\Documents\cloud> uvicorn Ecoommerce:app --reload
INFO: Will watch for changes in these directories: ['C:\\Users\\sada1\\Documents\\cloud']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
PS C:\Users\sada1\Documents\cloud> uvicorn Ecoommerce:app --reload
INFO: Will watch for changes in these directories: ['C:\\Users\\sada1\\Documents\\cloud']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Will watch for changes in these directories: ['C:\\Users\\sada1\\Documents\\cloud']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [2880] using StatReload
INFO: Started server process [4352]
INFO: Started reloader process [2880] using StatReload
INFO: Started server process [4352]
INFO: Waiting for application startup.
INFO: Started server process [4352]
INFO: Waiting for application startup.
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Application startup complete.
INFO: 127.0.0.1:54619 - "GET / HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:54619 - "GET /favicon.ico HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:61230 - "GET /doxc HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:53951 - "GET / HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:58367 - "GET /doxc HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:55954 - "GET / HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:55954 - "GET / HTTP/1.1" 404 Not Found
```

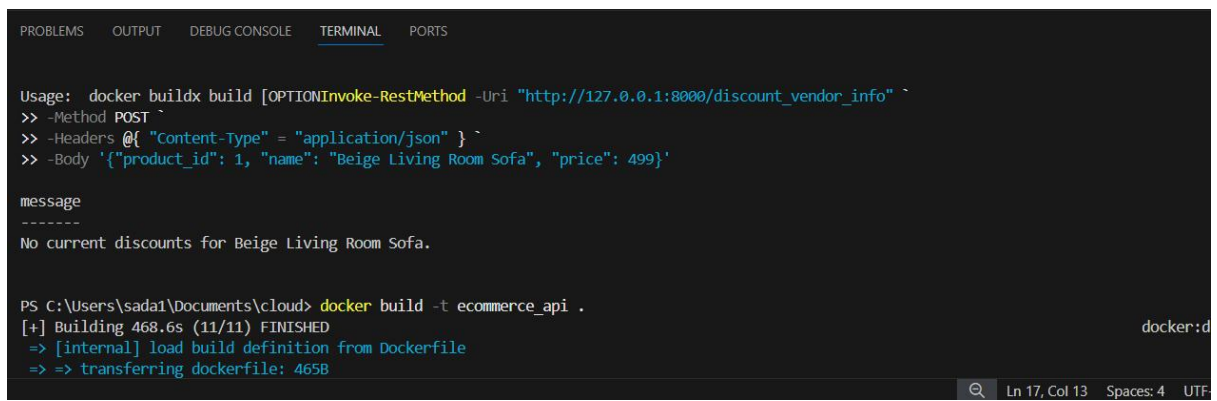


Creating the Docker file and building Docker image



The screenshot shows the VS Code Explorer sidebar on the left with the 'CLOUD' section expanded. The 'Dockerfile' file is selected. The main editor displays the content of the Dockerfile, which is a multi-stage build for an AWS Lambda function using Python 3.11. The Dockerfile includes instructions for installing system tools, requirements, and the application code, and sets the Lambda entrypoint.

```
Dockerfile > ...
1 FROM public.ecr.aws/lambda/python:3.11
2
3 WORKDIR /var/task
4
5 # Install system tools
6 RUN yum install -y gcc make rust cargo && \
7     python3.11 -m pip install --upgrade pip setuptools wheel
8
9 # Copy requirements and install them directly
10 COPY requirements.txt .
11 RUN python3.11 -m pip install --no-cache-dir -r requirements.txt
12
13 # Copy application code
14 COPY . .
15
16 # Lambda entrypoint
17 CMD ["Ecommerce.lambda_handler"]
```



The screenshot shows the VS Code Terminal with the 'TERMINAL' tab selected. It displays a REST client request and its response, followed by a Docker build command and its output.

```
Usage: docker buildx build [OPTION]...-RestMethod -Uri "http://127.0.0.1:8000/discount_vendor_info" ^
>> -Method POST ^
>> -Headers @{"Content-Type" = "application/json" } ^
>> -Body '{"product_id": 1, "name": "Beige Living Room Sofa", "price": 499}'

message
-----
No current discounts for Beige Living Room Sofa.

PS C:\Users\sada1\Documents\cloud> docker build -t ecommerce_api .
[+] Building 468.6s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 465B
```