

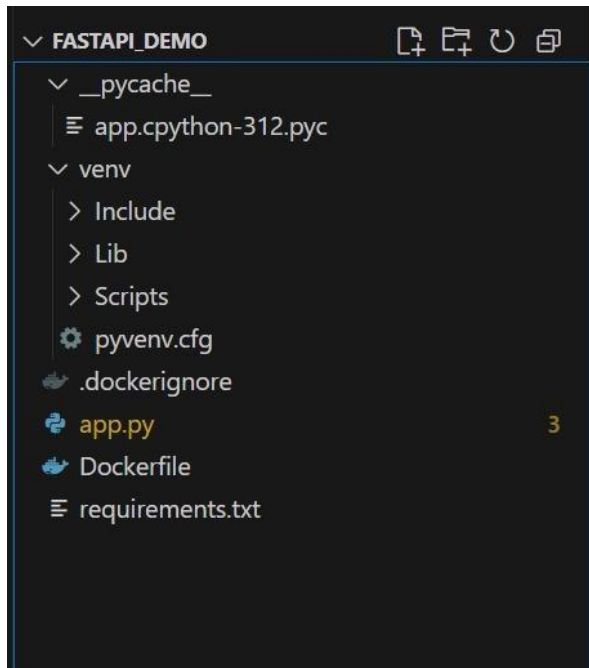
ADVANCED PYTHON PROGRAMMING

API using FastAPI and Docker

Name:Sathish V

Reg. NO:22MID0160

Project Structure:



Application Code - app.py

```
from fastapi import FastAPI, HTTPException, Request
from fastapi.middleware.cors import CORSMiddleware
from mangum import Mangum
import random
```

```
app = FastAPI(title="E-Commerce API", version="1.0")
```

```
# Enable CORS for frontend integration
```

```
app.add_middleware(
```

```
CORSMiddleware,
```

```
    allow_origins=["*"],
```

```
    allow_credentials=True,  
    allow_methods=["*"],  
    allow_headers=["*"],  
)
```

```
# --- API Endpoint ---
```

```
@app.post("/discount_vendor_info")
```

```
async def add_to_cart(request: Request):
```

```
    try:
```

```
        data = await request.json()
```

```
    print(" Received data:", data)
```

```
        product_id =
```

```
        data.get("product_id")    name =
```

```
        data.get("name")    price =
```

```
        data.get("price")
```

```
    if not all([product_id, name, price]):
```

```
        raise HTTPException(status_code=400, detail="Missing product details")
```

```
    # Generate random discount and vendor    discount = random.choice([0, 5, 10,  
15, 20])    vendor = random.choice(["TechWorld", "ShopEase", "ElectroMart",  
"MegaDeals"])
```

```
    if discount > 0:
```

```
        message = f"{discount}% discount available from {vendor} for {name}."
```

```
    else:
```

```
        message = f"No current discounts for {name}."
```

```
        return {"message": message}

    except Exception as e:
        raise HTTPException(status_code=500, detail=f"Processing failed: {str(e)}")

# --- AWS Lambda handler ---
lambda_handler = Mangum(app)
```

Docker Configuration

Dockerfile

```
FROM public.ecr.aws/lambda/python:3.11

WORKDIR /var/task

# Install system dependencies
RUN yum install -y gcc make rust cargo && \
python3.11 -m pip install --upgrade pip setuptools wheel

# Install dependencies
COPY requirements.txt .
RUN python3.11 -m pip install --no-cache-dir -r requirements.txt

# Copy source code
COPY . .

# Set entry point for AWS Lambda
CMD ["ECommerce_API.lambda_handler"]
```

.dockerignore

Ignore virtual environment

venv/

.env/

ENV/

.venv/

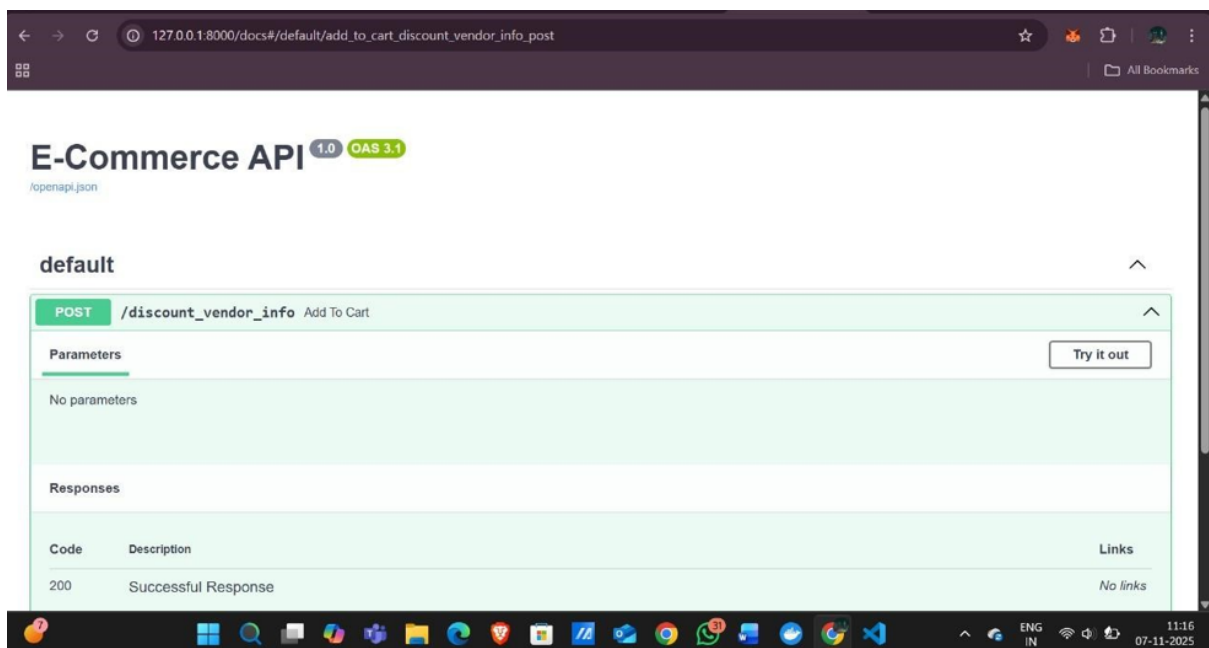
Ignore Python cache and compiled files

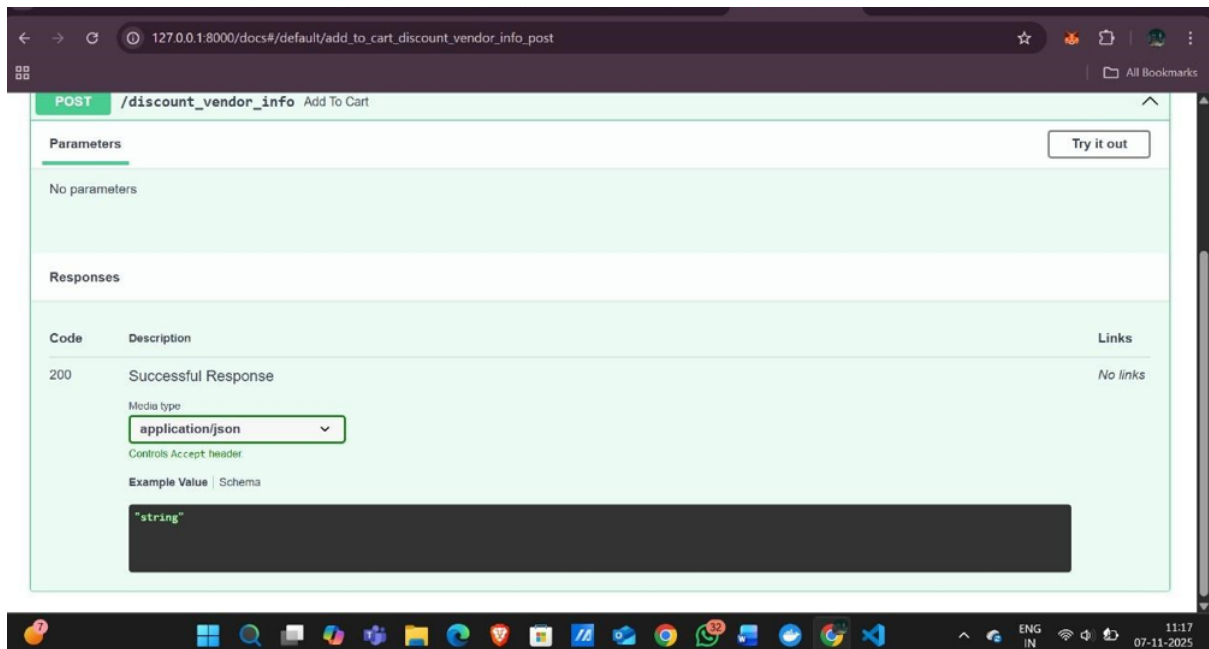
__pycache__/

Code Execution Commands:

- `python -m venv venv`
- `env\Scripts\activate`
- `pip install -r requirements.txt`
- `python -m uvicorn app:app --reload`

Result:





```
(.venv) matlab@sjt318scope012:~/cloud$ curl -X 'POST' \
'http://127.0.0.1:8000/discount_vendor_info' \
-H 'accept: application/json' \
-d '{ "product_id": 1, "name": "Product Name", "price": 499 }'
{"message": "5% discount available from ElectroMart for Product Name."}
```

Building and Running Docker Image :

```
(.venv) matlab@sjt318scope012:~/cloud$ docker build -t ecommerce_api .
(.venv) matlab@sjt318scope012:~/cloud$ sudo docker build -t ecommerce_api .
```

```
(.venv) matlab@sjt318scope012:~/cloud$ sudo docker login -u sathishveera012

Info → A Personal Access Token (PAT) can be used instead.
To create a PAT, visit https://app.docker.com/settings

Password:

WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
```

```
(.venv) matlab@sjt318scope012:~/cloud$ sudo docker tag ecommerce_api sathishveera012/ecommerce_api:latest
(.venv) matlab@sjt318scope012:~/cloud$ sudo docker push sathishveera012/ecommerce_api:latest
The push refers to repository [docker.io/sathishveera012/ecommerce_api]
f0ba7bfa9e5c: Pushed
30f2d01d697c: Pushed
5d09df60fd49: Pushed
b0ed68d1696c: Pushed
589beca3ab4a: Pushed
0d42791a23de: Pushed
7c5a65f222db: Pushed
dbc26eb5da09: Pushed
a4d6a2c814a7: Pushed
186096cd2e83: Pushed
latest: digest: sha256:cff154a40379d43b85da55e858300eb90b0d5fc945517259adea5f4c6f6cee4b size: 2418
```

Docker Container:

