

MONOLITHIC ARCHITECTURE

LAB – 1

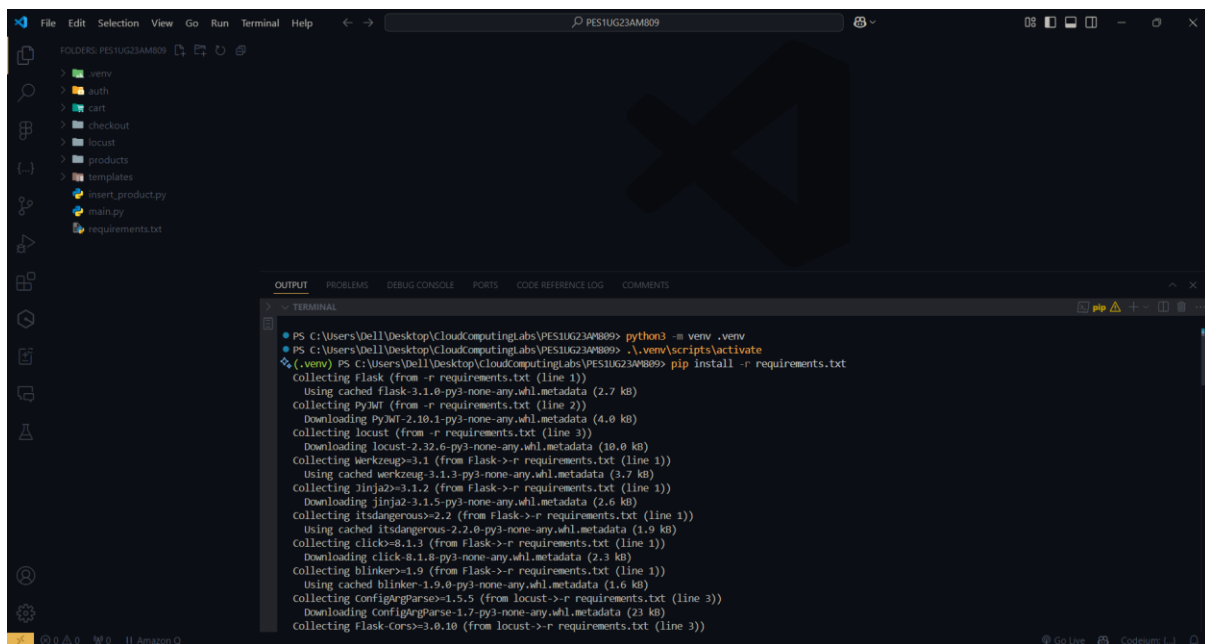
STEP- 1

Creating the directory my SRN PES1UG23AM809 and Unzipping the monolithic architecture folder.

STEP – 2 and 3

Now we will run the code

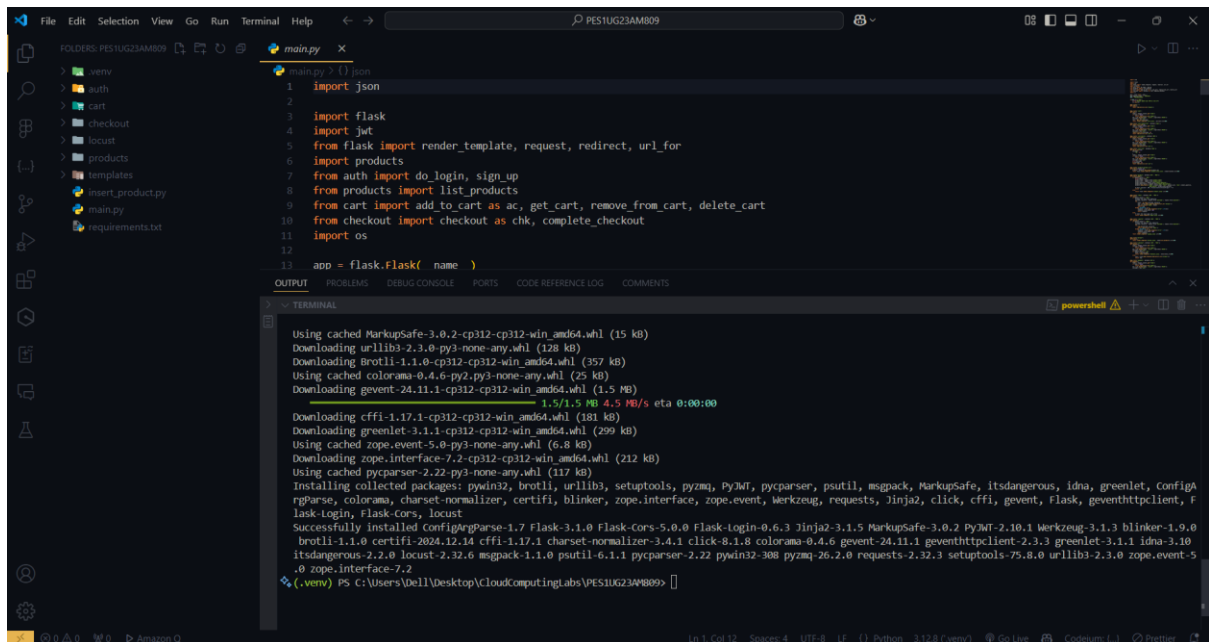
- We will setup python virtual environment so that there is no collision between module versions
- In windows



```
File Edit Selection View Go Run Terminal Help
PES1UG23AM809

FOLDERS: PES1UG23AM809
> .venv
> auth
> cart
> checkout
> locust
> products
> templates
> insert_product.py
> main.py
> requirements.txt

OUTPUT PROBLEMS DEBUG CONSOLE PORTS CODE REFERENCE LOG COMMENTS
TERMINAL
PS C:\Users\De11\Desktop\CloudComputingLabs\PES1UG23AM809> python3 -m venv .venv
(.venv) PS C:\Users\De11\Desktop\CloudComputingLabs\PES1UG23AM809> .\.venv\scripts\activate
(.venv) PS C:\Users\De11\Desktop\CloudComputingLabs\PES1UG23AM809> pip install -r requirements.txt
Collecting Flask (from -r requirements.txt (line 1))
  Using cached flask-3.1.0-py3-none-any.whl.metadata (2.7 kB)
Collecting PyJWT (from -r requirements.txt (line 2))
  Downloading PyJWT-2.10.1-py3-none-any.whl.metadata (4.0 kB)
Collecting locust (from -r requirements.txt (line 3))
  Downloading locust-2.32.6-py3-none-any.whl.metadata (10.0 kB)
Collecting Werkzeug>=3.1 (from Flask->-r requirements.txt (line 1))
  Using cached werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Collecting Jinja2>=3.1.2 (from Flask->-r requirements.txt (line 1))
  Downloading Jinja2-3.1.5-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.2 (from Flask->-r requirements.txt (line 1))
  Using cached itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from Flask->-r requirements.txt (line 1))
  Downloading click-8.1.8-py3-none-any.whl.metadata (2.3 kB)
Collecting blinker>=1.9 (from Flask->-r requirements.txt (line 1))
  Using cached blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting ConfigArgparse>=1.5.5 (from locust->-r requirements.txt (line 3))
  Downloading ConfigArgparse-1.7-py3-none-any.whl.metadata (23 kB)
Collecting Flask-Cors>=3.0.10 (from locust->-r requirements.txt (line 3))
```



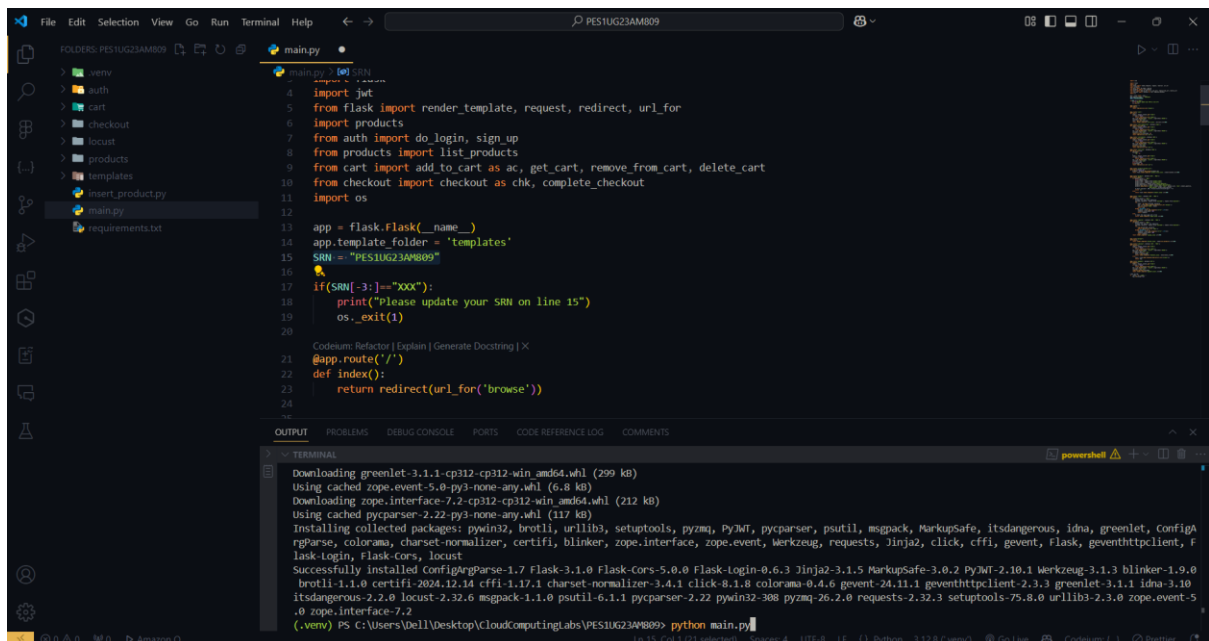
```
1 import json
2
3 import flask
4 import jwt
5 from flask import render_template, request, redirect, url_for
6 import products
7 from auth import do_login, sign_up
8 from products import list_products
9 from cart import add_to_cart as ac, get_cart, remove_from_cart, delete_cart
10 from checkout import checkout as chk, complete_checkout
11 import os
12
13 app = flask.Flask(__name__)
```

OUTPUT

```
Using cached MarkupSafe-3.0.2-cp312-cp312-win_amd64.whl (15 kB)
Downloading urllib3-2.3.0-py3-none-any.whl (128 kB)
Downloading Brotli-1.1.0-cp312-cp312-win_amd64.whl (357 kB)
Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Downloading gevent-24.11.1-cp312-cp312-win_amd64.whl (1.5 MB)
1.5/1.5 MB 4.5 MB/s eta 0:00:00
Downloading cffi-1.17.1-cp312-cp312-win_amd64.whl (181 kB)
Downloading greenlet-3.1.1-cp312-cp312-win_amd64.whl (299 kB)
Using cached zope.event-5.0-py3-none-any.whl (6.8 kB)
Downloading zope.interface-7.2-cp312-cp312-win_amd64.whl (212 kB)
Using cached pycparser-2.22-py3-none-any.whl (117 kB)
Installing collected packages: pywin32, brotli, urllib3, setuptools, pyzmq, PyJWT, pycparser, psutil, msgpack, MarkupSafe, itsdangerous, idna, greenlet, ConfigArgparse, colorama, charset-normalizer, certifi, blinker, zope.interface, zope.event, Werkzeug, requests, Jinja2, click, cffi, gevent, Flask, geventhttpclient, Flask-Login, Flask-Cors, locust
Successfully installed ConfigArgparse-1.7 Flask-3.1.0 Flask-Cors-5.0.0 Flask-Login-0.6.3 Jinja2-3.1.5 MarkupSafe-3.0.2 PyJWT-2.10.1 Werkzeug-3.1.3 blinker-1.9.0 brotli-1.1.0 certifi-2024.12.14 cffi-1.17.1 charset-normalizer-3.4.1 click-8.1.8 colorama-0.4.6 gevent-24.11.1 geventhttpclient-2.3.3 greenlet-3.1.1 idna-3.10 itsdangerous-2.2.0 locust-2.32.6 msgpack-1.1.0 psutil-6.1.1 pycparser-2.22 pywin32-308 pyzmq-26.2.0 requests-2.32.3 setuptools-75.8.0 urllib3-2.3.0 zope.event-5.0 zope.interface-7.2
(.venv) PS C:\Users\Dell\Desktop\CloudComputingLabs\PES1UG23AM809>
```

STEP – 4

In main.py I have updated the SRN, on line number 15



```
4 import jwt
5 from flask import render_template, request, redirect, url_for
6 import products
7 from auth import do_login, sign_up
8 from products import list_products
9 from cart import add_to_cart as ac, get_cart, remove_from_cart, delete_cart
10 from checkout import checkout as chk, complete_checkout
11 import os
12
13 app = flask.Flask(__name__)
14 app.template_folder = 'templates'
15 SRN = "PES1UG23AM809"
16
17 if (SRN[-3:]=="xxx"):
18     print("Please update your SRN on line 15")
19     os._exit(1)
20
21 @app.route('/')
22 def index():
23     return redirect(url_for("browse"))
24
```

OUTPUT

```
Downloading greenlet-3.1.1-cp312-cp312-win_amd64.whl (299 kB)
Using cached zope.event-5.0-py3-none-any.whl (6.8 kB)
Downloading zope.interface-7.2-cp312-cp312-win_amd64.whl (212 kB)
Using cached pycparser-2.22-py3-none-any.whl (117 kB)
Installing collected packages: pywin32, brotli, urllib3, setuptools, pyzmq, PyJWT, pycparser, psutil, msgpack, MarkupSafe, itsdangerous, idna, greenlet, ConfigArgparse, colorama, charset-normalizer, certifi, blinker, zope.interface, zope.event, Werkzeug, requests, Jinja2, click, cffi, gevent, Flask, geventhttpclient, Flask-Login, Flask-Cors, locust
Successfully installed ConfigArgparse-1.7 Flask-3.1.0 Flask-Cors-5.0.0 Flask-Login-0.6.3 Jinja2-3.1.5 MarkupSafe-3.0.2 PyJWT-2.10.1 Werkzeug-3.1.3 blinker-1.9.0 brotli-1.1.0 certifi-2024.12.14 cffi-1.17.1 charset-normalizer-3.4.1 click-8.1.8 colorama-0.4.6 gevent-24.11.1 geventhttpclient-2.3.3 greenlet-3.1.1 idna-3.10 itsdangerous-2.2.0 locust-2.32.6 msgpack-1.1.0 psutil-6.1.1 pycparser-2.22 pywin32-308 pyzmq-26.2.0 requests-2.32.3 setuptools-75.8.0 urllib3-2.3.0 zope.event-5.0 zope.interface-7.2
(.venv) PS C:\Users\Dell\Desktop\CloudComputingLabs\PES1UG23AM809> python main.py
```

STEP – 5

Visit the site (note if localhost didn't work try with 127.0.0.1)

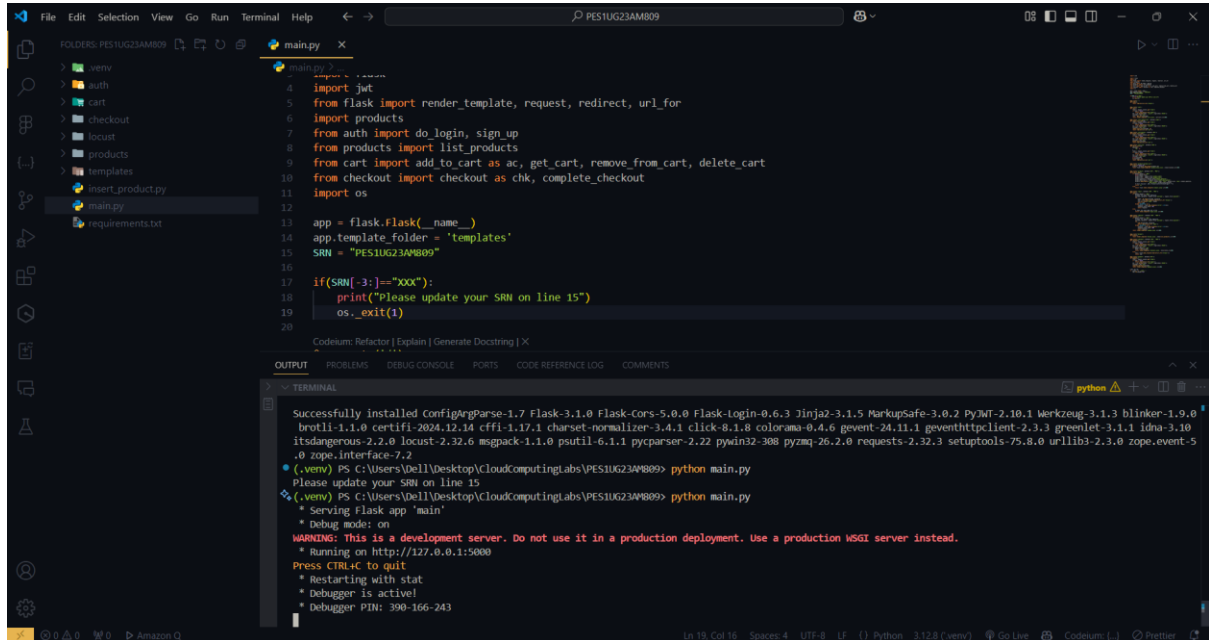
a. <http://localhost:5000/register> to register a user

b. <http://localhost:5000/login> to login

c. <http://localhost:5000/browse> to browse for stuff

d. When you are done adding to the cart, you can check out.

<http://localhost:5000/cart>

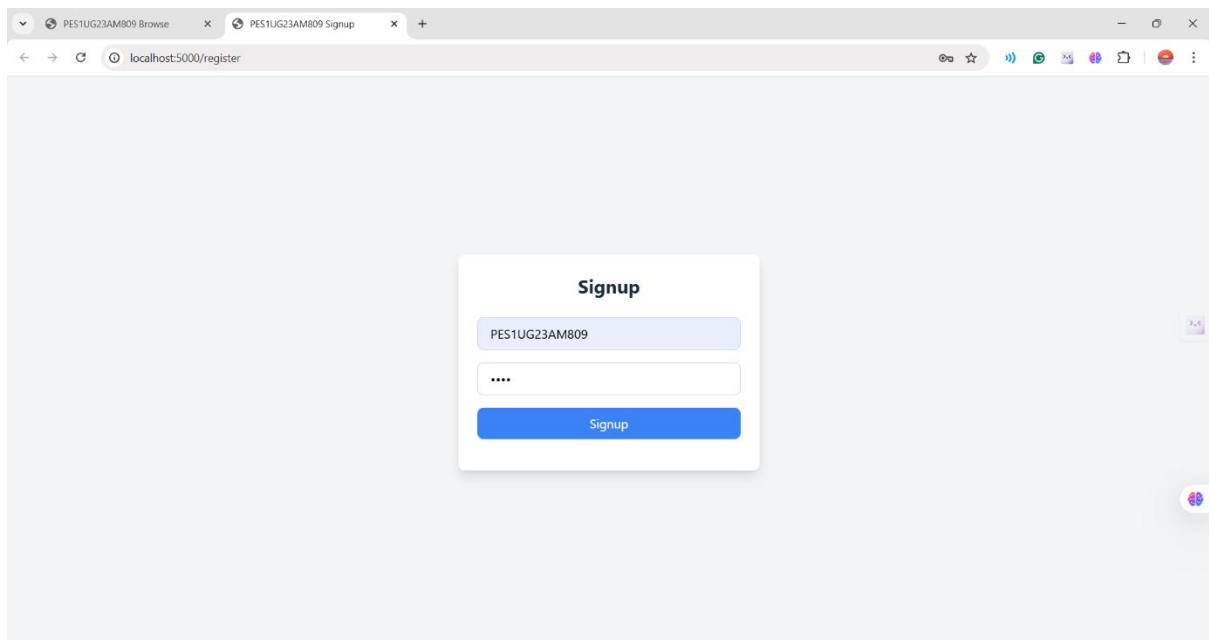


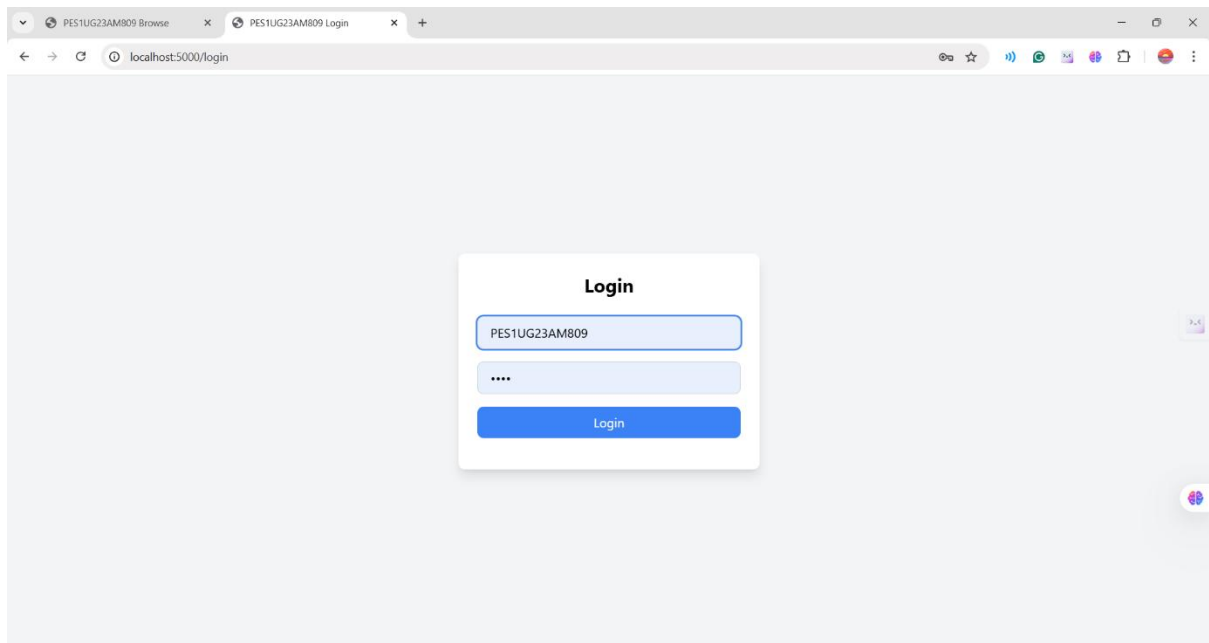
The screenshot shows a VS Code editor with a file explorer on the left displaying a project structure for 'PES1UG23AM809'. The main editor window shows the 'main.py' file with the following code:

```
1 import os
2 import json
3 import jwt
4 from flask import render_template, request, redirect, url_for
5 from flask import Flask
6 import products
7 from auth import do_login, sign_up
8 from products import list_products
9 from cart import add_to_cart as ac, get_cart, remove_from_cart, delete_cart
10 from checkout import checkout as chk, complete_checkout
11 import os
12
13 app = flask.Flask(__name__)
14 app.template_folder = 'templates'
15 SRN = "PES1UG23AM809"
16
17 if(SRN[-3:]!="xxx"):
18     print("Please update your SRN on line 15")
19     os._exit(1)
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

The terminal output shows the successful installation of various dependencies and the execution of the Flask application:

```
Successfully installed ConfigArgParse-1.7 Flask-3.1.0 Flask-Cors-5.0.0 Flask-Login-0.6.3 Jinja2-3.1.5 MarkupSafe-3.0.2 PyJWT-2.10.1 Werkzeug-3.1.3 blinker-1.9.0
brotli-1.1.0 certifi-2024.12.14 cffi-1.17.1 charset-normalizer-3.4.1 click-8.1.8 colorama-0.4.6 gevent-24.11.1 geventhttpclient-2.3.3 greenlet-3.1.1 idna-3.10
itsdangerous-2.2.0 locust-2.32.6 msgpack-1.1.0 psutil-6.1.1 pycparser-2.22 pywin32-308 pyzmq-26.2.0 requests-2.32.3 setuptools-75.8.0 urllib3-2.3.0 zope.event-5
.zope.interface-7.2
(.venv) PS C:\Users\Deell\Desktop\CloudComputingLabs\PES1UG23AM809> python main.py
Please update your SRN on line 15
(.venv) PS C:\Users\Deell\Desktop\CloudComputingLabs\PES1UG23AM809> python main.py
* Serving Flask app 'main'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 390-166-243
```

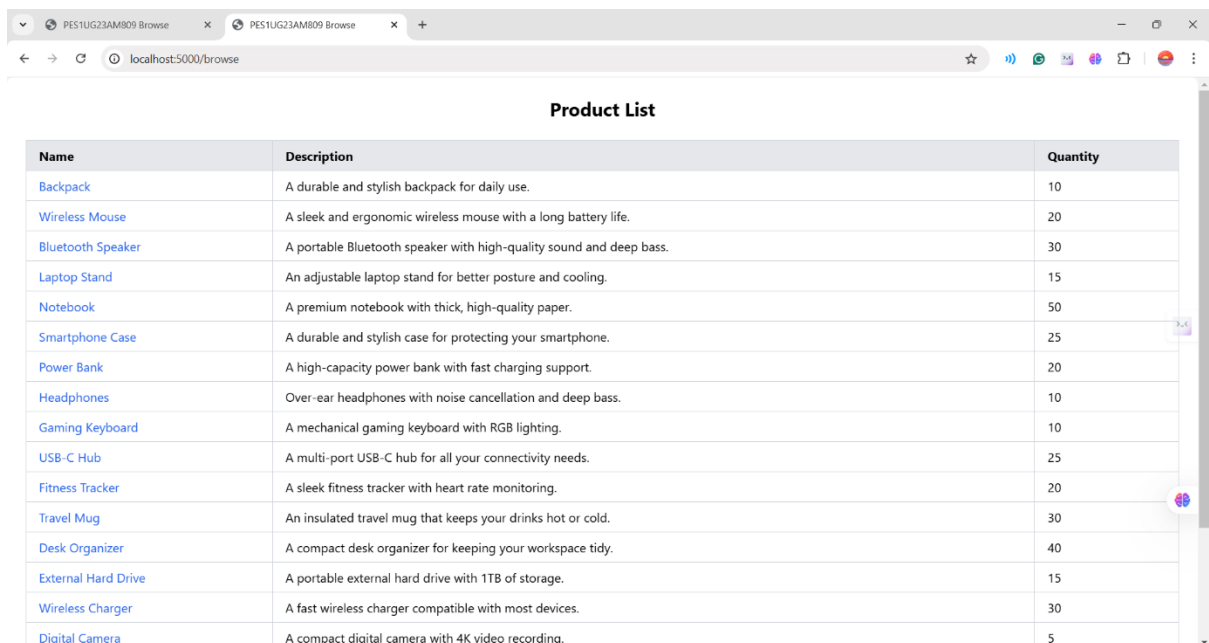




Browser tabs: PES1UG23AM809 Browse, PES1UG23AM809 Login

Address bar: localhost:5000/login

Login



Browser tabs: PES1UG23AM809 Browse, PES1UG23AM809 Browse

Address bar: localhost:5000/browse

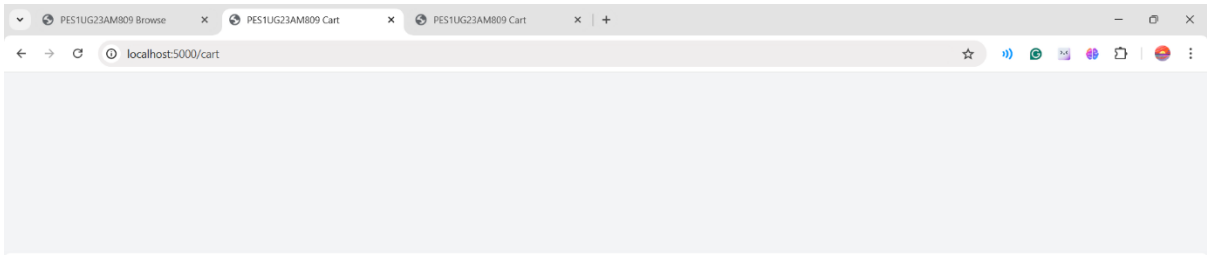
Product List

Name	Description	Quantity
Backpack	A durable and stylish backpack for daily use.	10
Wireless Mouse	A sleek and ergonomic wireless mouse with a long battery life.	20
Bluetooth Speaker	A portable Bluetooth speaker with high-quality sound and deep bass.	30
Laptop Stand	An adjustable laptop stand for better posture and cooling.	15
Notebook	A premium notebook with thick, high-quality paper.	50
Smartphone Case	A durable and stylish case for protecting your smartphone.	25
Power Bank	A high-capacity power bank with fast charging support.	20
Headphones	Over-ear headphones with noise cancellation and deep bass.	10
Gaming Keyboard	A mechanical gaming keyboard with RGB lighting.	10
USB-C Hub	A multi-port USB-C hub for all your connectivity needs.	25
Fitness Tracker	A sleek fitness tracker with heart rate monitoring.	20
Travel Mug	An insulated travel mug that keeps your drinks hot or cold.	30
Desk Organizer	A compact desk organizer for keeping your workspace tidy.	40
External Hard Drive	A portable external hard drive with 1TB of storage.	15
Wireless Charger	A fast wireless charger compatible with most devices.	30
Digital Camera	A compact digital camera with 4K video recording.	5

STEP – 6

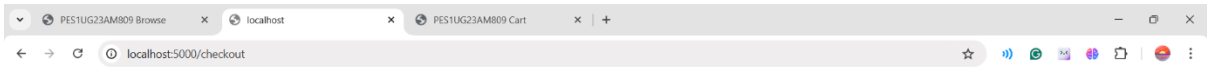
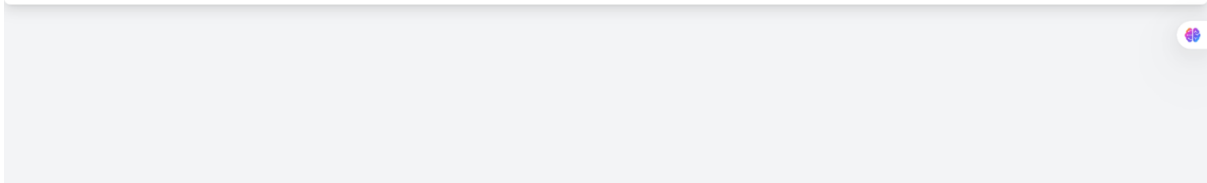
When you are done adding to the cart, you can checkout.

<http://localhost:5000/cart>



Cart

Product	Price	
Notebook	50.0	<button>Remove</button>
<div><button>Checkout</button><button>Clear Cart</button><button>Refresh</button><button>Back to Products</button></div>		



This site can't be reached

localhost refused to connect.

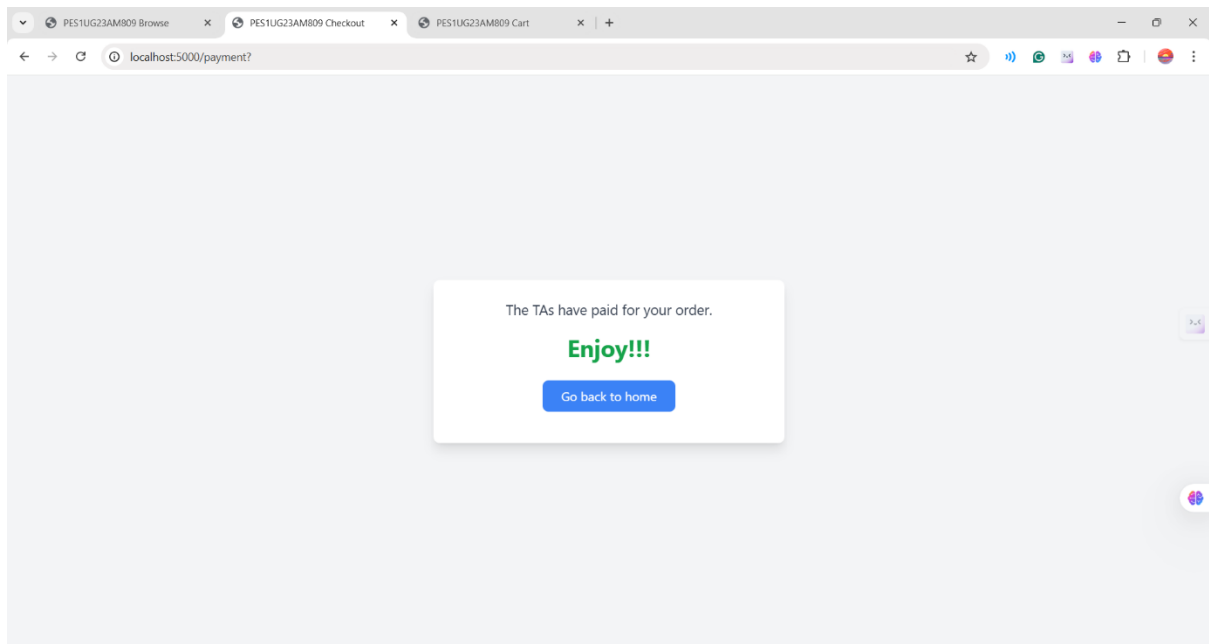
Try:

- Checking the connection
- [Checking the proxy and the firewall](#)

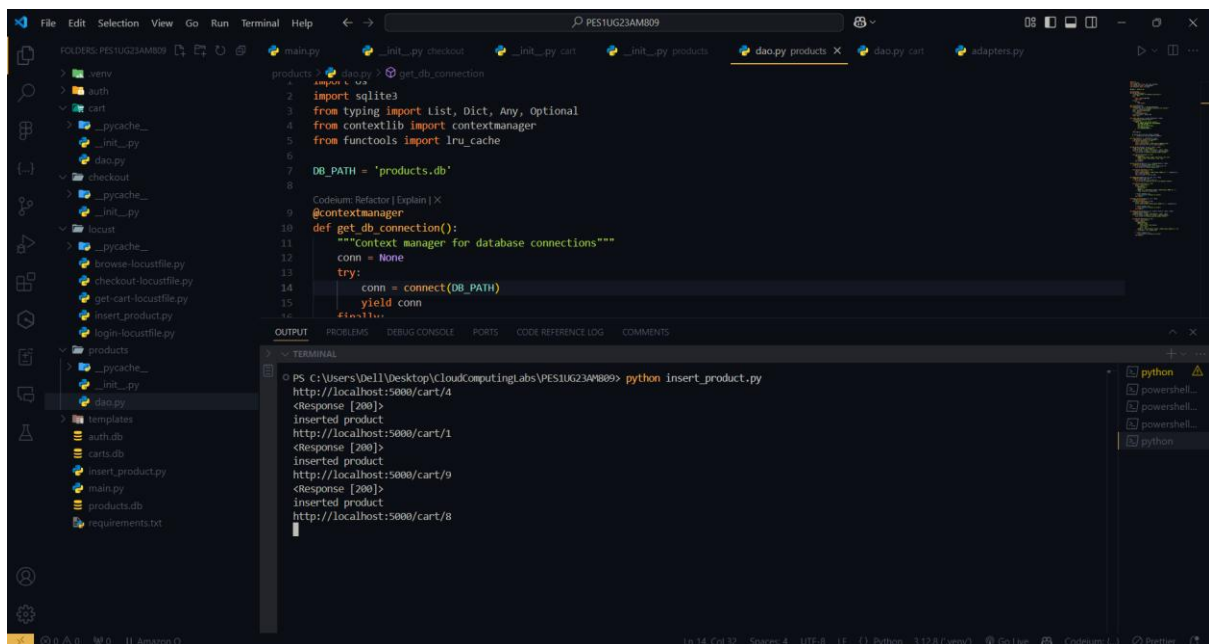
ERR_CONNECTION_REFUSED

Reload

Details



Locust - A load-testing tool, where you can define a test and run it on your endpoints.



The screenshot shows a PyCharm IDE with a project named PES1UG23AM809. The left sidebar displays the project structure, including folders like .venv, auth, cart, _pycache_, dao.py, checkout, _pycache_, _init_.py, locust, _pycache_, browse-locustfile.py, checkout-locustfile.py, get-cart-locustfile.py, insert_product.py, login-locustfile.py, products, _pycache_, dao.py, templates, auth.db, carts.db, insert_product.py, main.py, products.db, and requirements.txt. The main editor window shows the code for products.py, which includes imports for sqlite3, typing, contextlib, and functools, and a function get_db_connection() that uses a context manager to connect to a database. The terminal window at the bottom shows the execution of a locust stress test, including the command locust -f checkout-locustfile.py and the output of the test, which shows the starting web interface at http://localhost:8889 and the ramping to 1 user at a rate of 1.00 per second. The terminal also displays a traceback error for a connection issue.

```
products.py
1 import sqlite3
2 from typing import List, Dict, Any, Optional
3 from contextlib import contextmanager
4 from functools import lru_cache
5
6 DB_PATH = 'products.db'
7
8
9 @contextmanager
10 def get_db_connection():
11     """Context manager for database connections"""
12     conn = None
13     try:
14         conn = connect(DB_PATH)
15         yield conn
16     finally:
```

Terminal Output:

```
PS C:\Users\De11\Desktop\CloudComputingLabs\PES1UG23AM809> cd locust
PS C:\Users\De11\Desktop\CloudComputingLabs\PES1UG23AM809\locust> locust -f checkout-locustfile.py
[2025-01-20 15:34:26,293] DESKTOP-KNDQ6LUN/INFO/locust.main: Starting locust 2.32.6
[2025-01-20 15:34:26,294] DESKTOP-KNDQ6LUN/INFO/locust.main: Starting web interface at http://localhost:8889 (accepting connections from all network interfaces)
[2025-01-20 15:37:12,494] DESKTOP-KNDQ6LUN/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
Traceback (most recent call last):
  File "C:\Users\De11\Desktop\CloudComputingLabs\PES1UG23AM809\.venv\Lib\site-packages\urllib3\connection.py", line 198, in _new_conn
    sock = connection.create_connection(
            ~~~~~
  File "C:\Users\De11\Desktop\CloudComputingLabs\PES1UG23AM809\.venv\Lib\site-packages\urllib3\util\connection.py", line 85, in create_connection
    raise err
  File "C:\Users\De11\Desktop\CloudComputingLabs\PES1UG23AM809\.venv\Lib\site-packages\urllib3\util\connection.py", line 73, in create_connection
    sock.connect(sa)
  File "C:\Users\De11\Desktop\CloudComputingLabs\PES1UG23AM809\.venv\Lib\site-packages\eventlet\socketcommon.py", line 586, in connect
    self._internal_connect(address)
  File "C:\Users\De11\Desktop\CloudComputingLabs\PES1UG23AM809\.venv\Lib\site-packages\eventlet\socketcommon.py", line 630, in _internal_connect
```

The screenshot displays a development environment with three main components:

- Code Editor:** Shows a Python script named `adapters.py` with a `HTTPAdapter` class. The `send` method includes error handling for `ProxyError`, `SSLError`, and `ConnectionError`.
- Terminal:** Shows the output of a Locust test. It includes a traceback for a `KeyboardInterrupt`, a summary of test results, and response time percentiles.
- Web Browser:** Displays the Locust web interface at `localhost:8089`. The interface shows the test is running with 1 user and 0.5 RPS.

Terminal Output:

```
Traceback (most recent call last):
  File "C:\Users\De11\Desktop\CloudComputingLabs\PES1UG23AM809\.venv\Lib\site-packages\gevent\_ffi\loop.py", line 270, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument
KeyboardInterrupt
2025-01-20 15:42:36.956 DESKTOP-KNDQ6LN/INFO/locust.main: Shutting down (exit code 0)
[2025-01-20 15:42:36.956]
Type      Name      # reqs      # fails      Avg      Min      Max      Med      req/s      failures/s
-----
GET /checkout 37      0(0.00%)    2044     2018     2064     2018     0.46     0.00
Aggregated 37      0(0.00%)    2044     2018     2064     2018     0.46     0.00

Response time percentiles (approximated)
Type      Name      50%      66%      75%      80%      90%      95%      98%      99.0%  99.99%  100% # reqs
-----
GET /checkout 2000    2000    2100    2100    2100    2100    2100    2100    2100    2100    37
Aggregated 2000    2000    2100    2100    2100    2100    2100    2100    2100    2100    37

PS C:\Users\De11\Desktop\CloudComputingLabs\PES1UG23AM809\locust>
```

Locust Web Interface:

STATISTICS	CHARTS	FAILURES	EXCEPTIONS	CURRENT RATIO	DOWNLOAD DATA	LOGS
<div>LOCUST</div> <div>HOST: http://localhost:5000 STATUS: RUNNING USERS: 1 RPS: 0.5 FAILURES: 0%</div> <div>EDIT STOP RESET</div>						

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/checkout	12	0	2019.8	2100	2100	2045.29	2020	2064	690	0.5	0
Aggregated		12	0	2019.8	2100	2100	2045.29	2020	2064	690	0.5	0

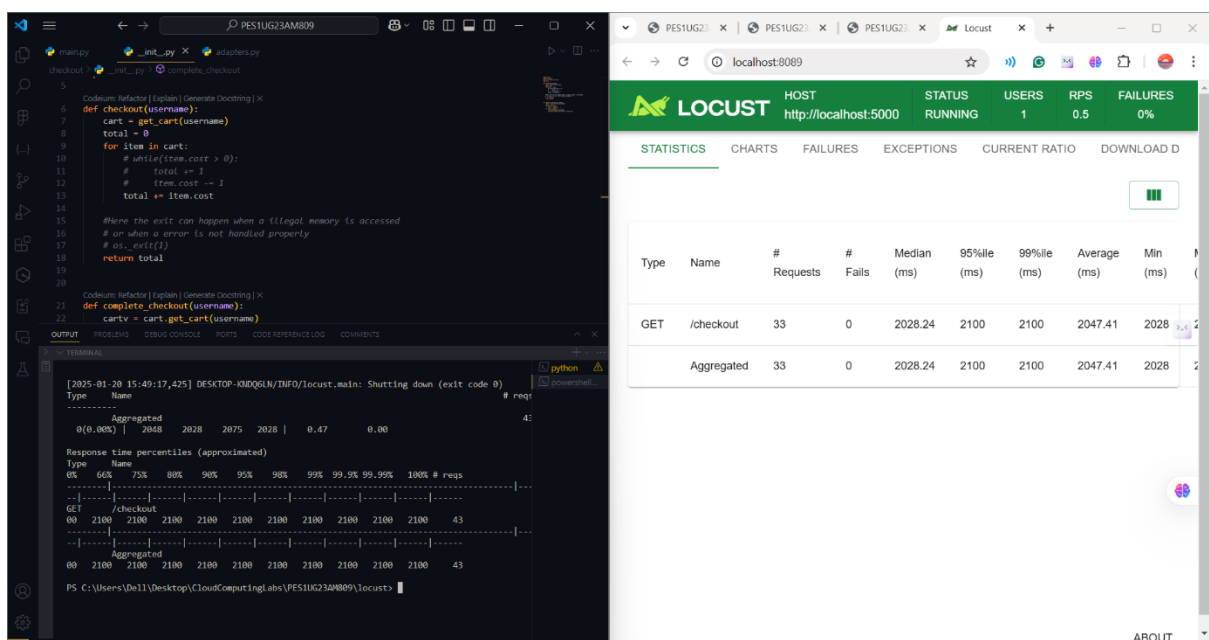
Optimise the route /checkout

The screenshot shows the VS Code editor with the file `checkout.py` open. The code defines a `checkout` function that calculates the total cost of items in a cart and returns it. The terminal window shows the command `locust -f checkout-locustfile.py` being executed, and the output displays the Locust web interface starting at `http://localhost:8089`. The terminal also shows the aggregated statistics for the `/checkout` route.

```
def checkout(username):
    cart = get_cart(username)
    total = 0
    for item in cart:
        while(item.cost > 0):
            total += 1
            item.cost -= 1
    # Here the exit can happen when a illegal memory is accessed
    # or when a error is not handled properly
    os._exit(1)
    return total

def complete_checkout(username):
    # ...
```

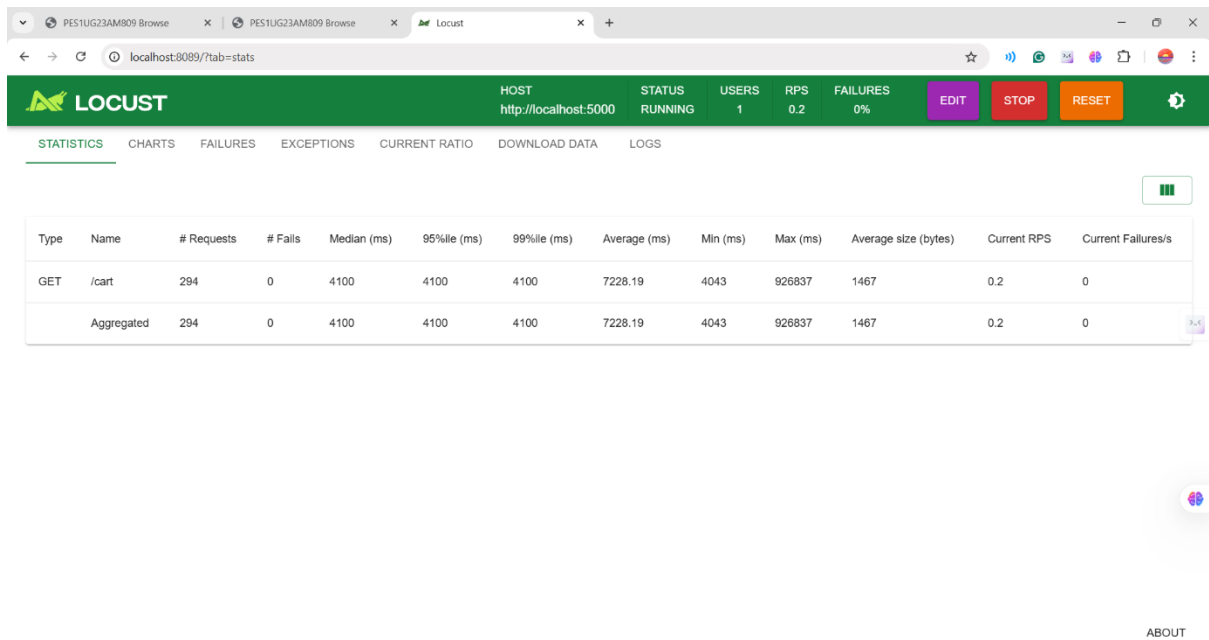
```
PS C:\Users\Del1\Desktop\CloudComputing\labs\PES1UG23AM809\locust> locust -f checkout-locustfile.py
[2025-01-20 15:47:19,355] DESKTOP-KNDQ6LN/INFO/locust.main: Starting Locust 2.32.6
[2025-01-20 15:47:19,356] DESKTOP-KNDQ6LN/INFO/locust.main: Starting web interface at http://localhost:8089 (accepting connections from all network interfaces)
[2025-01-20 15:47:44,174] DESKTOP-KNDQ6LN/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2025-01-20 15:47:48,327] DESKTOP-KNDQ6LN/INFO/locust.runners: All users spawned: ("checkout": 1) (1 total users)
Traceback (most recent call last):
-----
Aggregated
0.47 0.00
```



STEP - 7

Optimizing the browse and cart locust files from the cart/ and products/ folder.

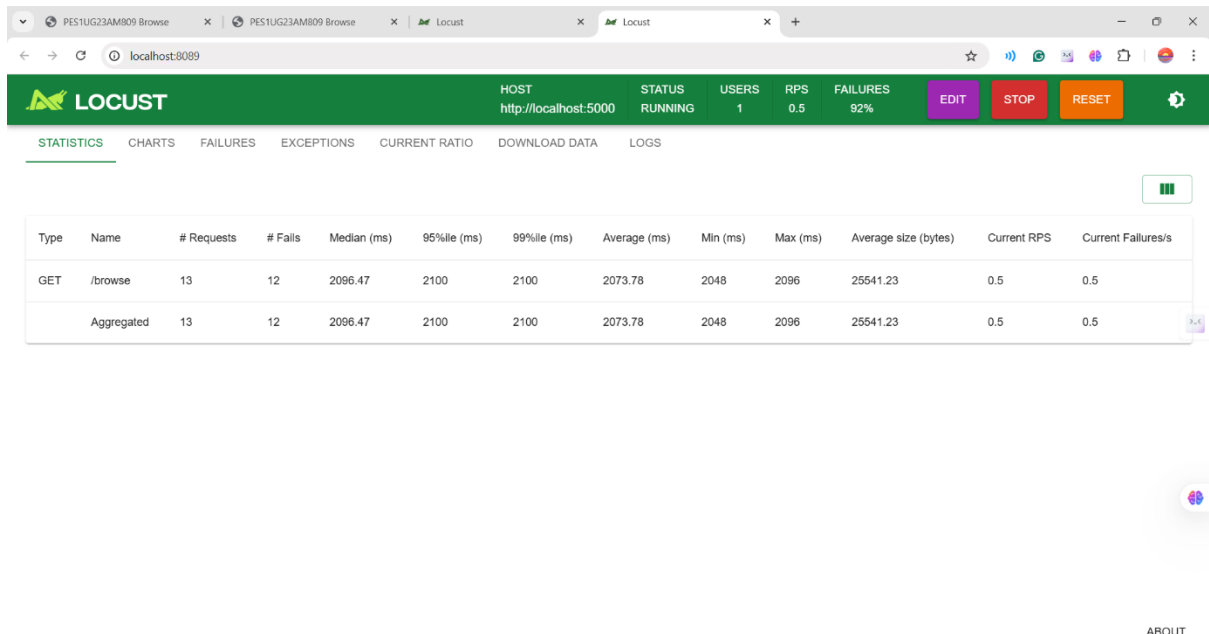
/cart/___init___py



The screenshot shows a code editor with the __init__.py file for the cart endpoint. The file contains imports for sqlalchemy, typing, contextlib, and functools, and defines a DB_PATH variable and a get_db_connection function. The terminal output shows the Locust test results, including the start of the web interface, the ramping of users, and the final performance statistics for the /cart endpoint.

```
PS C:\Users\Dell\Desktop\CloudComputingLabs\PES1UG23AM809> cd locust
PS C:\Users\Dell\Desktop\CloudComputingLabs\PES1UG23AM809> cd locust
PS C:\Users\Dell\Desktop\CloudComputingLabs\PES1UG23AM809> locust -f get-cart-locustfile.py
[2025-01-20 16:34:52,059] DESKTOP-KNDQ6LNU/INFO/locust.main: Starting web interface at http://localhost:8889 (accepting connections from all network interfaces)
[2025-01-20 16:35:25,721] DESKTOP-KNDQ6LNU/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2025-01-20 16:35:29,648] DESKTOP-KNDQ6LNU/INFO/locust.runners: All users spawned: {"add_to_cart": 1} (1 total users)
Traceback (most recent call last):
  File "c:\Users\Dell\Desktop\CloudComputingLabs\PES1UG23AM809\.venv\Lib\site-packages\event_ffli\loop.py", line 270, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument
KeyboardInterrupt
2025-01-20 17:11:41:26Z
[2025-01-20 17:11:26,958] DESKTOP-KNDQ6LNU/INFO/locust.main: Shutting down (exit code 0)
Type      Name      # reqs      # fails | Avg  Min  Max  Med | req/s  failures/s
Response time percentiles (approximated)
Type      Name      -----
GET        /cart      4100      4100 | 4100 4100 4100 4100 | 4100 927000 927000 927000 | 300
Aggregated 4100      4100 | 4100 4100 4100 4100 | 4100 927000 927000 927000 | 300
```

/products/__init__.py



The screenshot shows a code editor with the file products/__init__.py open. The code defines a context manager for database connections. Below the code, the Locust runner output is displayed in a terminal window.

```
PS C:\Users\Dell\Desktop\CloudComputingLabs\PES1UG23AM809\locust> locust -f browse-locustfile.py
[2025-01-20 17:12:32,438] DESKTOP-KHQ6LUN/INFO/locust-main: Starting Locust 2.32.6
[2025-01-20 17:12:32,439] DESKTOP-KHQ6LUN/INFO/locust-main: Starting web interface at http://localhost:8089 (accepting connections from all network interfaces)
[2025-01-20 17:12:50,947] DESKTOP-KHQ6LUN/INFO/locust-runners: Ramping to 1 users at a rate of 1.00 per second
[2025-01-20 17:12:50,948] DESKTOP-KHQ6LUN/INFO/locust-runners: All users spawned: ("browse": 1) (1 total users)
Traceback (most recent call last):
  File "C:\Users\Dell\Desktop\CloudComputingLabs\PES1UG23AM809\.venv\lib\site-packages\gevent\ffil\loop.py", line 270, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument
KeyboardInterrupt
[2025-01-20 17:13:34,113] DESKTOP-KHQ6LUN/INFO/locust-main: Shutting down (exit code 1)

Type      Name      # reqs      # fails      Avg      Min      Max      Med      req/s      failures/s
-----
GET      /browse      20      19(95.00%)      2071      2045      2096      2096      0.48      0.46
Aggregated
-----
GET      /browse      20      19(95.00%)      2071      2045      2096      2096      0.48      0.46

Response time percentiles (approximated)
Type      Name      50%      60%      75%      80%      90%      95%      98%      99%      99.9%      100% # reqs
-----
GET      /browse      2100      2100      2100      2100      2100      2100      2100      2100      2100      2100      20
```

STEP – 8

GitHub Link to the Optimized code and folder of the Monolithic Architecture

Link :

[git@github.com:tgo460/PES1UG23AM809](https://github.com/tgo460/PES1UG23AM809) CC MONOLITHIC ARC LA B3 .git