Project-Github-Link: https://github.com/sunnysavita10/ecomm-prod-assistant

These are all the commands that you need to run on your command prompt

1. Write Python in your terminal
2. If you have Python, then no need to install it
3. uv --version
4. If you are not able to get the version
5. Pip install uv
6. import shutil
7. print(shutil.which("uv"))
8.
9. 6. Uv init <my-project-name>
10. 7. uv pip list
11.
12. 8. uv python list
13. uv venv env --python cpython-3.10.18-windows-x86_64-none
14. uv venv <your-env-namne> --python <your-python-version>
15. Note: Please use either 3.10, 3.11, or 3.12
16. **Command Prompt (CMD)**
    .\<your-env-nanme>\Scripts\activate.bat
17. **Git Bash ya WSL terminal, or MAC Terminal:**
    a. source <your-env-nanme>/Scripts/activate
    b.
18. If your git is asking for a login to publish the repo, execute the command below

ECR_REGISTRY=<account-id>.dkr.ecr.<aws_region>.amazonaws.com

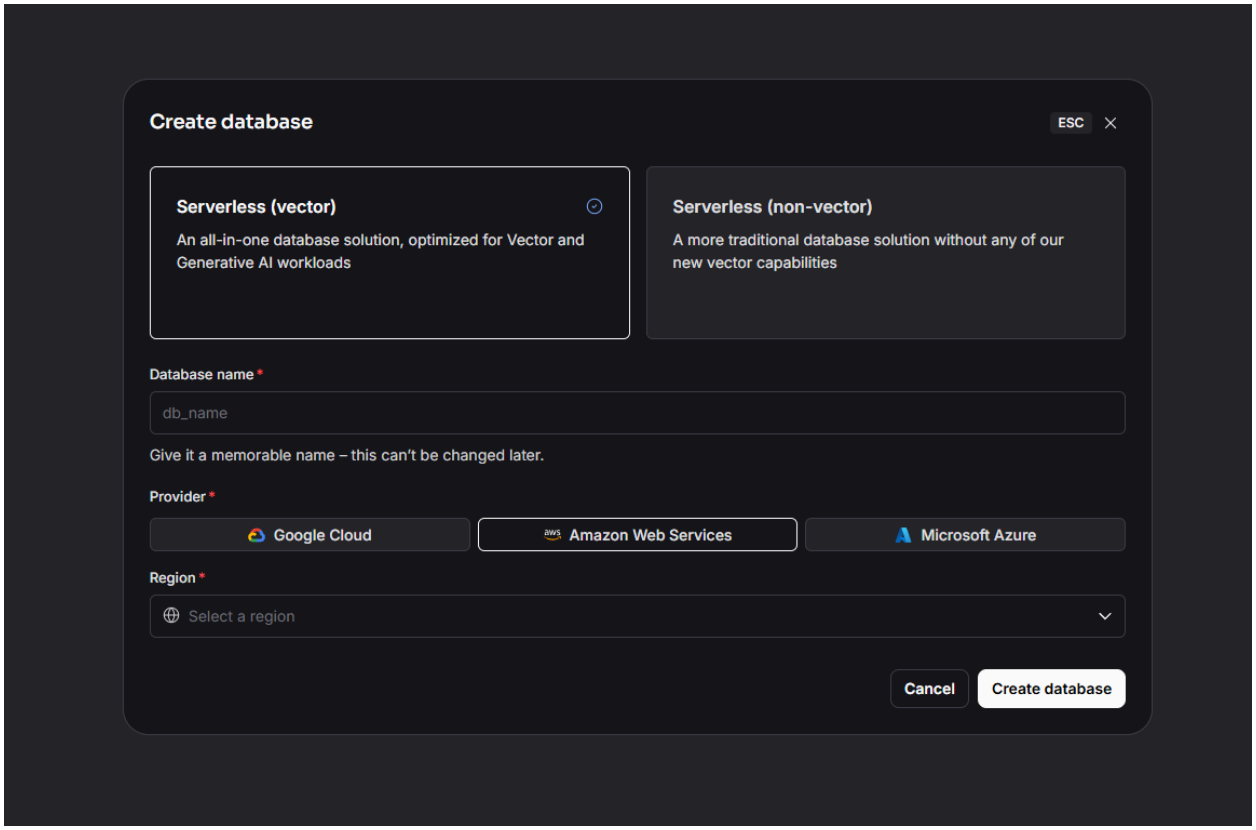
    c. git config --global user.name "Your Name"
    d. git config --global user.email "your-email@example.com"
19. UV add <package_name>
20. Uv add -r requirements.txt
21. Streamlit run <give your streamlit python filename>
22. Install the live server extension in VS Code for testing the HTML

For accessing the DataStax, here is a link:
https://accounts.datastax.com/session-service/v1/login

Vectordb Comparison:
https://superlinked.com/vector-db-comparison

Once you log in to the DataStax Vector page, you will get the following page

ECR_REGISTRY=<account-id>.dkr.ecr.<aws_region>.amazonaws.com



ECR_REGISTRY=<account-id>.dkr.ecr.<aws_region>.amazonaws.com

For running the streamlit UI, the command is:

streamlit run <file_path_of_streamlit_python_file>

For installing your prod_assistant as a package use the .toml file

For install the package through the toml file here is a command

Uv pip install -e .

Or mention -e . in th requirements.txt and run the command

uv pip install -r requirements.txt

(NOTE: Same thing we can do with the setup.py file and we have already done it in the previous project)

Command for executing the fastapi:
uvicorn prod_assistant.router.main:app --reload --port 8000

Command for running the streamlit app
Stream run <your_file_name.py>

ECR_REGISTRY=<account-id>.dkr.ecr.<aws_region>.amazonaws.com

Step to the run the application:

1. First run the mcp server:

D:\complete_content_new\llmops-batch\ecomm-prod-assistant\prod_assistant\mcp_servers\product_search_server.py

2. If you want to test your application you can in two ways

First: with client.py file
Second: from agentic workflow

Note: use the latest workflow:
D:\complete_content_new\llmops-batch\ecomm-prod-assistant\prod_assistant\workflow\agentic_workflow_with_mcp_websearch.py

Note: please use your system path not mine

3. Now after testing run the application from api and test it via ui your application will be running on this url

http://127.0.0.1:8000/
uvicorn prod_assistant.router.main:app --reload --port 8000

ECR_REGISTRY=<account-id>.dkr.ecr.<aws_region>.amazonaws.com

docker ps # running containers check karne ke liye
docker stop <container_id>
docker rm <container_id>
docker images # images list check karne ke liye
docker rmi <image_id>

Build Docker Image
Use this command: docker build -t prod-assistant .

Run Docker Container
docker run -d -p 8080:8080 --name <container_custon_name>
<give image name which you have created using dockerfile>

Use this command:
docker run -d -p 8000:8000 --name product-assistant prod-assistant

${{ secrets.AWS_ACCESS_KEY_ID }}
${{ secrets.AWS_SECRET_ACCESS_KEY }}
${{ secrets.AWS_REGION }}
${{ secrets.ECR_REGISTRY }}
${{ secrets.ECR_REPOSITORY }}
${{ secrets.EKS_CLUSTER_NAME }}
${{ secrets.GROQ_API_KEY }}
${{ secrets.GOOGLE_API_KEY }}
${{ secrets.ASTRA_DB_API_ENDPOINT }}
${{ secrets.ASTRA_DB_APPLICATION_TOKEN }}
${{ secrets.ASTRA_DB_KEYSPACE }}

Keep the scerates without the double quote

ECR_REGISTRY=<account-id>.dkr.ecr.<aws_region>.amazonaws.com

Link for downloading the aws CLI:
https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

```
C:\Users\Sunny>doskey /history
aws
aws configure
aws eks update-kubeconfig --name product-assistant-cluster-latest --region us-west-1
kubectl get nodes
kubectl get svc -o wide
kubectl describe svc product-assistant-service
kubectl get pods -o wide
kubectl exec -it product-assistant-776b47db47-tp4jb -- curl http://localhost:8000
kubectl logs product-assistant-776b47db47-tp4jb
doskey /history

C:\Users\Sunny>

C:\Users\Sunny>
```

Once deployment is done then after for getting all the details through your CLI you need to execute some important commands

Aws eks update-kubeconfig –name <eks-cluster-name> –region <write_aws_region>

Kubectl get nodes
Kubectl get svc -o wide
aws
aws configure

aws eks update-kubeconfig --name product-assistant-cluster-latest --region us-west-1

kubectl get nodes
kubectl get svc -o wide
kubectl describe svc product-assistant-service
kubectl get pods -o wide

ECR_REGISTRY=<account-id>.dkr.ecr.<aws_region>.amazonaws.com

kubectl logs <write_your_pod_id>

doskey /history

ECR_REGISTRY=<account-id>.dkr.ecr.<aws_region>.amazonaws.com