**Project Structure**

AI\_Project/

├── app.py # Flask server and Load data into ChromaDB

├── requirements.txt # Python dependencies

├── templates/

│ └── index.html # Frontend UI

├── static/

│ └── style.css # design logic

├── terraform/

│ ├── main.tf # Terraform config for EC2 and volume attachment

│ ├── variables.tf

│ └── outputs.tf

**| |\_\_** providers.tf

* Build AWS infrastructure using terraform
* After ec2 instance is up Ssh into your ec2 instance and do the below actions

**Mount the volume:**

lsblk /to check the volume name

create path where you will load everything and will mount volume

**Create the mount point (if it doesn't exist yet):**

sudo file -s /dev/nvme1n1

sudo mkfs.ext4 /dev/nvme1n1

mkdir -p /mnt/data

sudo mount /dev/nvme1n1 /mnt/data

mount /dev/nvme1n1 /mnt/data

**Verify it's mounted:**

df -h /mnt/data

**Now run below commands:**

1. python3 -m venv venv
2. source venv/bin/activate
3. yum update -y
4. yum install -y git python3
5. git clone https://github.com/Oendrella-ti/AI\_Project.git
6. mkdir -p /mnt/data/tmp\_pip
7. mkdir -p /mnt/data/AI\_Project/libs
8. pip install --upgrade pip

TMPDIR=/mnt/data/tmp\_pip pip install \

--no-cache-dir \

--target=/mnt/data/AI\_Project/libs \

sentence-transformers chromadb flask

1. pip install --no-cache-dir sentence-transformers chromadb flask

[Note: these can be added in terraform code as well]

* Build the application:

cd /mnt/data/AI\_Project

nohup python3 app.py > flask.log 2>&1 &

tail the flask.log and look for any error. If service is up the open the url.

http://{ec2-public-ip}:5000

Note : I have taken port 5000. You can take any available port