Project Title

Project Documentation

1.Introduction

• Project title : Citizen Ai
• Team member : Rithika R
• Team member :Shalini S
• Team member : Shamyuktha D
• Team member :Sharmila M

2.project overview

• **Purpose :**

The purpose of a Sustainable Smart City Assistant is to empower cities and their residents to thrive in a more eco-conscious and connected urban environment. By leveraging AI and real-time data, the assistant helps optimize essential resources like energy, water, and waste, while also guiding sustainable behaviors among citizens through personalized tips and services. For city officials, it serves as a decision□making partner—offering clear insights, forecasting tools, and summarizations of complex policies to support strategic planning. Ultimately, this assistant bridges technology, governance, and community engagement to foster greener cities that are more efficient, inclusive, and resilient.

Architecture

The new architecture would be broken down into four main layers:

1. **Front-end (Streamlit)**:

   A user-facing web application built with Streamlit. Unlike Gradio, which is excellent for quick demos, Streamlit is better suited for building more polished, feature-rich dashboards and applications. It will handle the user interface for both the city analysis and citizen services functions, sending requests to the backend API.

2. **Back-end (FastAPI)**:

   A high-performance, asynchronous API built with FastAPI. This will be the central hub that receives requests from the front-end. It'll contain all the business logic, including the LLM integration, data processing, and communication with other services. FastAPI's built-in automatic documentation with OpenAPI (Swagger UI) is a huge advantage for development and testing.

3. **LLM Integration Layer**:

   This layer is managed by the FastAPI backend. It's responsible for interacting with the language model.

   - **IBM watsonx Granite**: This is the core LLM that powers the application. The backend will use the Hugging Face `transformers` library to load and run this model, similar to the original code. For production, you could consider using IBM's official `watsonx-python-sdk` for a more stable and enterprise-grade connection.

4. **Data & ML Layer**:

   This layer contains the services for data retrieval and specialized machine learning tasks.

   - **Vector Search (Pinecone)**: To improve the accuracy and relevance of the LLM's responses, especially for specific, up-to-date information, you can implement a **Retrieval-Augmented Generation (RAG)** system. A vector database like Pinecone would store embeddings of a knowledge base (e.g., city crime reports, government documents). When a user asks a question, the system retrieves relevant documents from Pinecone and provides them to the LLM as context, reducing "hallucinations."
   - **ML Modules (Forecasting & Anomaly Detection)**: These modules would be separate services within the backend. They could use pre-trained models or be trained on relevant data. For example, they could analyze historical crime data to **forecast** future trends or detect **anomalous** spikes in accidents. Common algorithms for these tasks include ARIMA for time series forecasting and Isolation Forest or One-Class SVM for anomaly detection.

Setup Instructions

Prerequisites

- Python 3.8+
- `pip` package installer
- Access to an IBM watsonx account (or Hugging Face hub for the open model)
- A Pinecone API key and environmen

Installation Process

1. Clone the Repository
2. Set up Virtual Environment
3. **Install Dependencies**: Create a `requirements.txt` file in the root directory and install everything

Folder Structure

A well-organized structure is crucial for a multi-layered application.

Running the Application

1. **Backend**: Navigate to the `backend` directory and run the FastAPI server using `uvicorn`
2. **Frontend**: Open a new terminal, navigate to the `frontend` directory, and run the Streamlit app streamlit run app.py .This will launch the Streamlit app in your web browser, which will make requests to the running FastAPI server.

API Documentation

FastAPI automatically generates interactive API documentation. Once the backend server is running, you can access the OpenAPI documentation at

Swagger UI: `http://127.0.0.1:8000/docs`

Redoc: `http://127.0.0.1:8000/redoc`

Authentication

For a production environment, you should implement user authentication to secure the API endpoints. A common and secure method with FastAPI is using **OAuth2 with JWT (JSON Web Tokens)**. This would involve:

1. A user logs in with a username/password, and the backend validates their credentials.
2. Upon successful validation, the backend generates and returns a JWT.
3. For subsequent requests, the user's front-end sends the JWT in the `Authorization` header.
4. FastAPI's security dependencies can then automatically validate this token for protected routes.

User Interface

The Streamlit front-end will mirror the original Gradio app's functionality but with a more polished look. It'll use a sidebar for navigation between the "City Analysis" and "Citizen Services" pages. Users can input data, see progress indicators, and view the AI-generated responses in styled text boxes.

Testing

- **Unit Tests**: Use `pytest` to test individual functions, such as the `generate_response` helper or the functions that interact with Pinecone.
- **Integration Tests**: Test the API endpoints to ensure they correctly handle requests and return the expected responses, including proper error handling for invalid inputs or authentication failures.

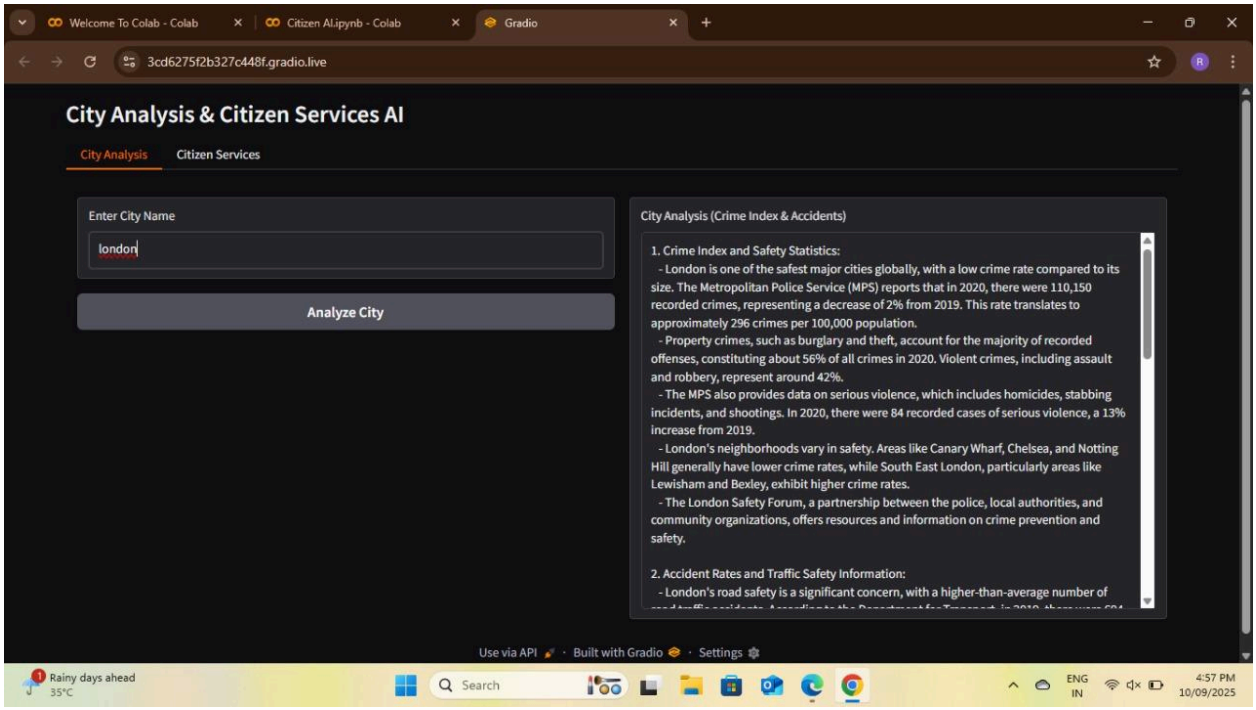
Known Issues

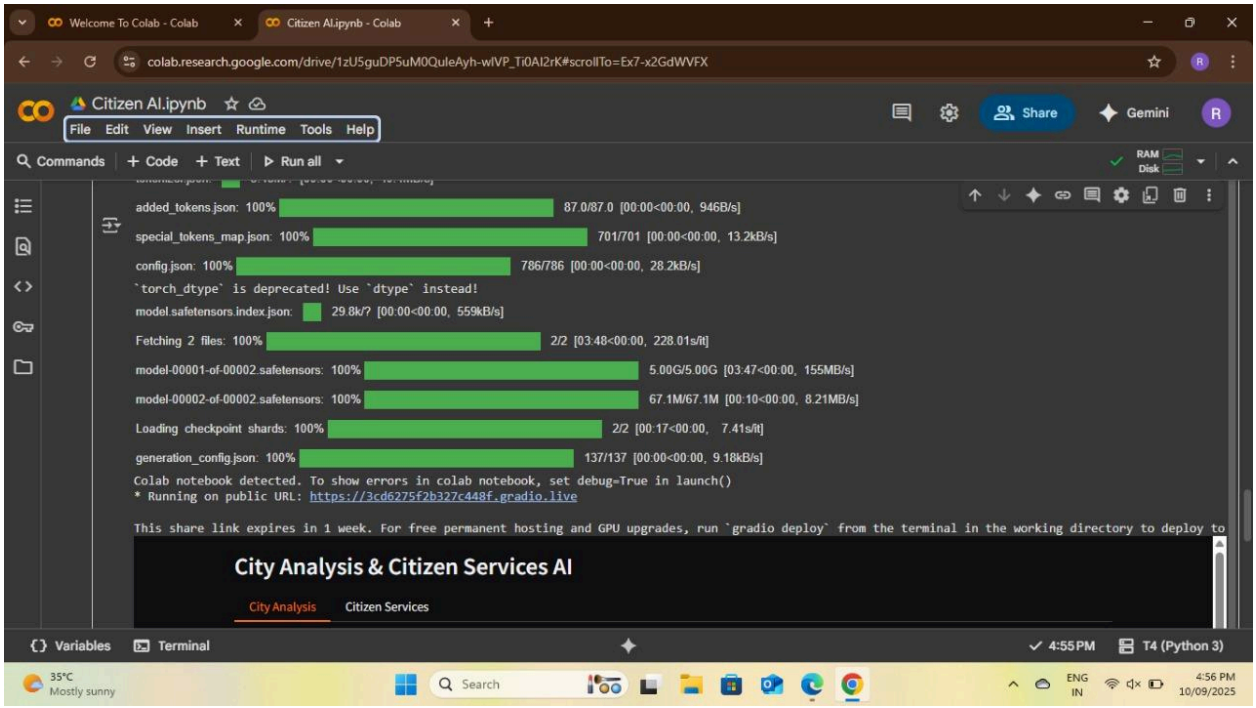
- **LLM Hallucinations**: Despite the RAG setup, the LLM may still generate factually incorrect information. This is an inherent risk of current LLMs.
- **Performance**: Loading a 2B-parameter model into memory can be slow and resource-intensive, especially on CPU. This can be mitigated with quantizing the model, or using GPUs.
- **API Latency**: The time taken for the LLM to generate a response can lead to high latency.

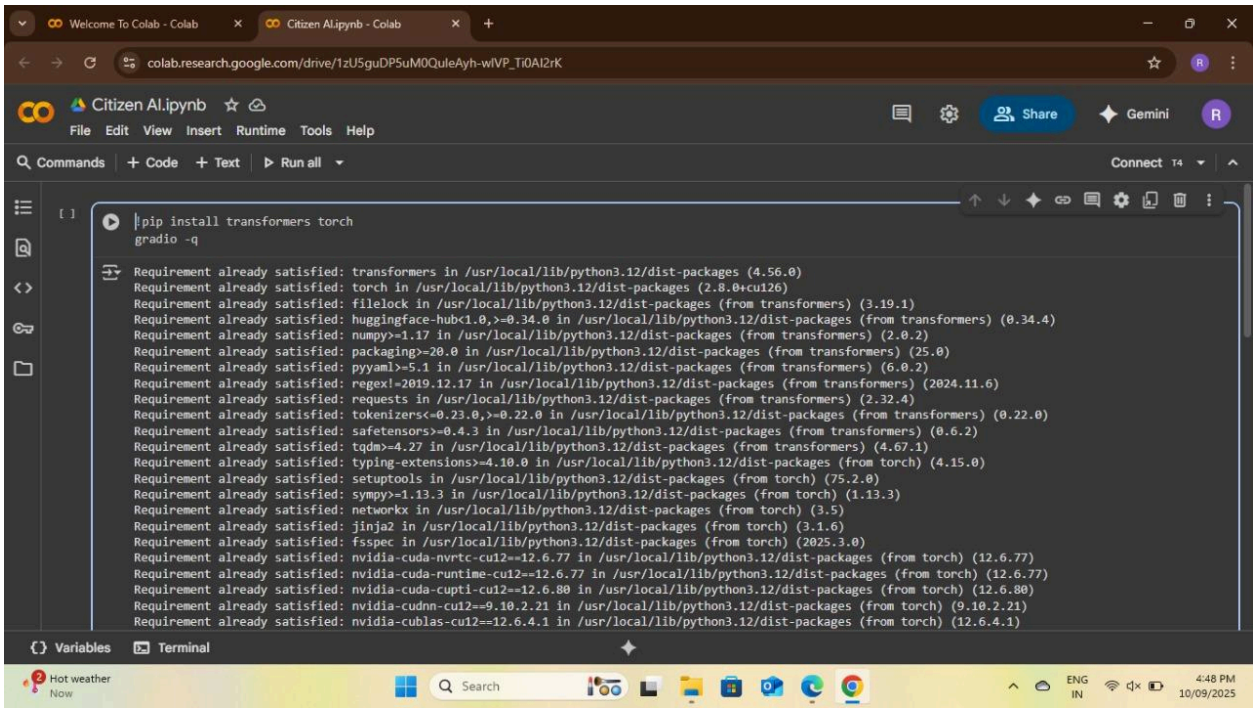
Future Enhancements

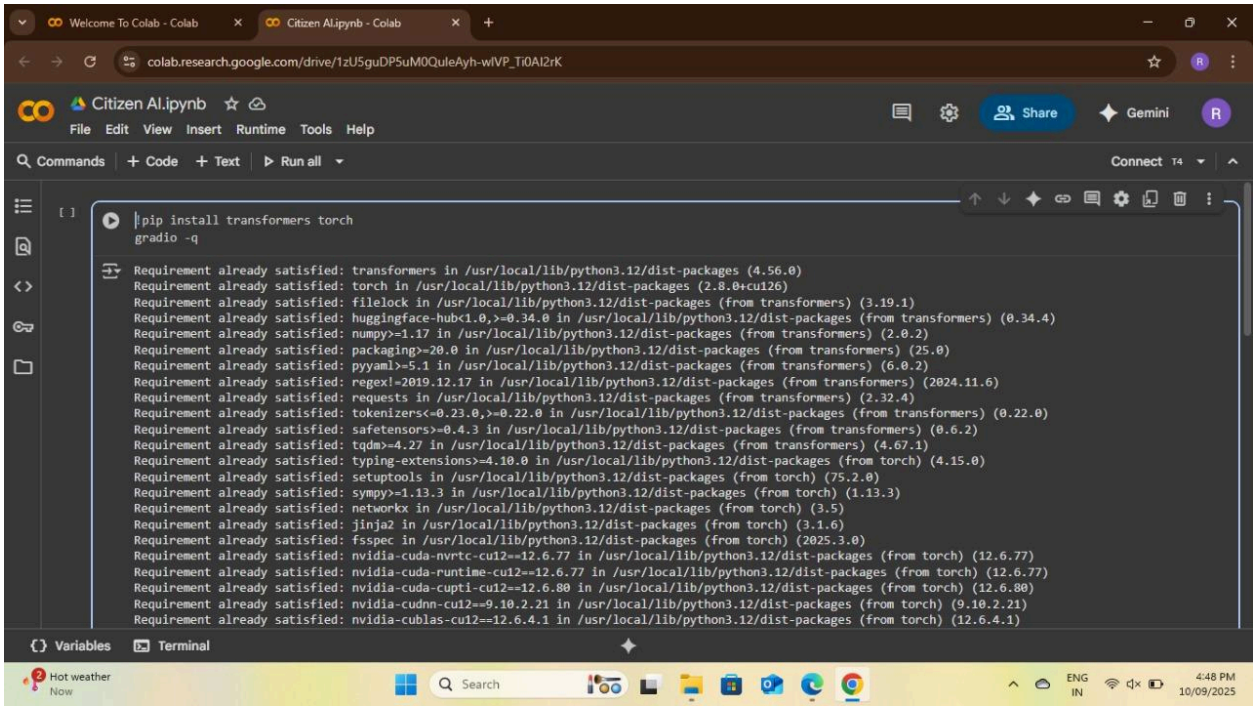
- **Real-time Data Integration**: Connect the backend to real-time data sources (e.g., public crime APIs, traffic data) to provide up-to-the-minute information.
- **Chat History**: Implement a chat feature that maintains context for a more conversational experience in the "Citizen Services" tab. This would involve a session management system.
- **User Feedback Loop**: Add a feature where users can rate the quality of the AI's response (e.g., thumbs up/down). This data can be used to fine-tune the model or improve the prompting strategy over time.
- **Advanced ML Modules**: Integrate more complex ML models, like time-series models for forecasting crime rates or deep learning models for detecting specific anomalies in traffic patterns.

OUTPUT




Welcome To Colab - Colab
Citizen AI.ipynb - Colab
colab.research.google.com/drive/1zU5guDP5uM0QuleAyh-wlVP_Ti0AI2rK#scrollTo=Ex7-x2GdWVFX
Citizen AI.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
RAM
Disk
added_tokens.json: 100% 87.0/87.0 [00:00<00:00, 946B/s]
special_tokens_map.json: 100% 701/701 [00:00<00:00, 13.2kB/s]
config.json: 100% 786/786 [00:00<00:00, 28.2kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json: 29.8k/? [00:00<00:00, 559kB/s]
Fetching 2 files: 100% 2/2 [03:48<00:00, 228.01s/it]
model-00001-of-00002.safetensors: 100% 5.00G/5.00G [03:47<00:00, 155MB/s]
model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:10<00:00, 8.21MB/s]
Loading checkpoint shards: 100% 2/2 [00:17<00:00, 7.41s/it]
generation_config.json: 100% 137/137 [00:00<00:00, 9.18kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://3cd6275f2b327c448f.gradio.live
This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to
City Analysis & Citizen Services AI
City Analysis
Citizen Services
Variables
Terminal
4:55PM
T4 (Python 3)
35°C
Mostly sunny
Search
ENG
IN
4:56 PM
10/09/2025


Welcome To Colab - Colab
Citizen AI.ipynb - Colab
colab.research.google.com/drive/1zU5guDP5uM0QuleAyh-wlVP_Ti0AI2rK
Citizen AI.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
Connect T4
pip install transformers torch
gradio -q
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.56.0)
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.8.0+cu126)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from transformers) (3.19.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.34.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.34.4)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from transformers) (2.32.4)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.22.0)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.6.2)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch) (4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from torch) (3.5)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch) (3.1.6)
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from torch) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch) (9.10.2.21)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.4.1)
Variables
Terminal
Hot weather
Now
Search
ENG
IN
4:48 PM
10/09/2025


Welcome To Colab - Colab
Citizen AI.ipynb - Colab
colab.research.google.com/drive/1zU5guDP5uM0QuleAyh-wlVP_Ti0AI2rK
Citizen AI.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Run all
Connect T4
[ ]
pip install transformers torch
gradio -q
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.56.0)
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.8.0+cu126)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from transformers) (3.19.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.34.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.34.4)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from transformers) (2.32.4)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.22.0)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.6.2)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch) (4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from torch) (3.5)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch) (3.1.6)
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from torch) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch) (9.10.2.21)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.4.1)
Variables
Terminal
Hot weather
Now
Search
ENG
IN
4:48 PM
10/09/2025