Project Title	**Fine-tune LLMs and build your application**
Technologies	Large Language Model
Domain	Generative AI
Project Difficulties level	Advance

Problem Statement:

In the rapidly evolving landscape of natural language processing, there is a growing demand for accessible and efficient solutions that enable developers and businesses to harness the capabilities of Large Language Models (LLMs) for creating bespoke applications. The challenge lies in the absence of a unified platform that seamlessly integrates leading technologies such as Langchain, Hugging Face, LLama2, and Mistral 7b, to facilitate the fine-tuning of LLMs for diverseuse cases.

Developers currently face hurdles in adapting state-of-the-art language models to their specific requirements, limiting the potential for innovation and customization in applications. This project seeks to address these challenges by establishing a comprehensive framework that not only streamlines the fine-tuning process but also provides a user-friendly interface for building applications on top of these tailored language models.

Key issues to address include:

1. **Integration Complexity:** The integration of Langchain, Hugging Face, LLama2, and Mistral 7bposes challenges in terms of compatibility and interoperability. The project needs to tackle thecomplexity associated with seamlessly incorporating these technologies into a cohesive platform.
2. **Fine-tuning Accessibility:** Developers currently lack a straightforward method for fine-tuning LLMs to suit specific application needs. The project must devise user-friendly tools and documentation to democratize the fine-tuning process and make it accessible to a broader audience.
3. **Application Development Barriers:** The absence of a unified platform for building applications on top of fine-tuned language models hinders developers from efficiently.

translating their ideas into functional products. The project needs to provide a robust infrastructure that simplifies application development and deployment.

4. **Performance Optimization:** Optimizing the performance of fine-tuned language models on different hardware configurations and deployment environments is a critical aspect that demands attention. The project should address these challenges to ensure the seamless execution of applications across varied scenarios.
5. **Community Collaboration:** Encouraging collaboration and knowledge sharing among developers using the platform is vital for fostering a vibrant community. Implementing features such as forums, documentation updates, and collaborative spaces will be essential in building a supportive ecosystem around the project. By addressing these challenges, the "Fine-tune LLMs & Build Your Application" project aims to empower developers and businesses to unlock the full potential of LLMs, fostering innovation and customization in natural language processing applications.

Dataset:

Dataset Repository for "Fine-tune LLMs & Build Your Application" is OpenAssistant/oasst1. Curate a diverse dataset specific to the Indian continent, addressing linguistic and cultural nuances. The dataset repository, including train and validation sets, is accessible at:

- Train Dataset
- Validation Dataset

Project Evaluation metrics:

Code:

- You are supposed to write a code in a modular fashion
- Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system)
- You have to maintain your code on GitHub.
- You have to keep your GitHub repo public so that anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards: https://www.python.org/dev/peps/pep-0008/

Database:

- You are supposed to use a given dataset for this project which is a MongoDB and MySQL database.
- https://astra.dev/ineuron

Cloud:

- You can use any cloud platform for this entire solution hosting like AWS, Azure or GCP

API Details or User Interface:

- You must expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

Logging:

- Logging is a must for every action performed by your code. Use the python logginglibrary for this.

Ops Pipeline:

- If possible, you can try to use AI ops pipeline for project delivery Ex. DVC, MLflow, Sage maker, Azure machine learning studio, Jenkins, Circle CI, Azure DevOps , TFX, Travis CI.

Deployment:

- You can host your model in the cloud platform, edge devices, or maybe local, but with a proper justification of your system design.

Solutions Design:

- You must submit complete solution design strategies in HLD and LLD document.

System Architecture:

- You must submit a system architecture design in your wireframe document and architecture document.

Latency for model response:

- You must measure the response time of your model for a particular input of a dataset.

Optimization of solutions:

- Try to optimize your solution on code level, architecture level and mention all of these things in your final submission.
- Mention your test cases for your project.

Submission requirements:

High-level Document:

You must create a high-level document design for your project. You can reference the HLD form below the link.

Sample link:

HLD Document Link

Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the below link.

Sample link

LLD Document Link

Architecture: You must create an Architecture document design for your project; you can refer to the Architecture from the below link.

Sample link

Architecture sample link

Wireframe: You must create a Wireframe document design for your project; refer to the Wireframe from the below link.

Demo link

Wireframe Document Link

Project code:

You must submit your code GitHub repo in your dashboard when the final submissionof your project.

Demo link

Project code sample link :

Detail project report:

You must create a detailed project report and submit that document as per the given sample.

Demo link

DPR sample link

Project demo video:

You must record a project demo video for at least 5 Minutes and submit that link as per the given demo.

Demo link

Project sample link :

The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.

Demo link

Linkedin post sample link :