Project Title	**AI-Powered Data Analysis System**
Technologies	Large Language Model
Domain	Generative AI
Project Difficulties level	Advance

Problem Statement:

Traditional data analysis methods often struggle to handle large volumes of data efficiently. Moreover, deriving meaningful insights from diverse data sources in real time poses a significant challenge. Businesses and researchers face hurdles in making data-driven decisions promptly due to the time-consuming nature of manual analysis. The project addresses these issues by automating the analysis process using AI.

To address this problem statement one can use large language models (LLMs) and Python. programming language, relational databases like SQL or no SQL databases like MongoDB for storage, machine learning algorithms for prediction and clustering of the data, visualization tools like Matplotlib, Seaborn, Plotly for visualization of insights, and NLTK or SpaCy to process the text data. Then deploy the project in any cloud like AWS, Azure, or Google Cloud.

Objectives:

1. **Automate Data Analysis**: Develop an intelligent system capable of automatically analyzing large and diverse datasets without the need for extensive manual intervention.
2. **Insight Generation**: Implement advanced algorithms and machine learning models to extract meaningful and actionable insights from the data, enabling users to make informed decisions promptly.
3. **Enhance Decision-Making**: Empower businesses, researchers, and users across various domains by providing them with accurate, real-time insights derived from analyzed data, enabling better decision-making.
4. **Ease of Use**: Design an intuitive user interface that simplifies the interaction with the system, allowing users of varying technical expertise to access and benefit from its analytical capabilities.

5. **Customization and Adaptability**: Enable users to customize analyses, create tailored dashboards, and adapt the system to their specific data requirements and analytical needs.
6. **Real-Time Processing**: Ensure the system's ability to process and deliver insights in real-time, facilitating swift responses to changing data patterns or business conditions.

7. **Data Security and Compliance**: Implement robust security measures to protect sensitive data, adhering to industry standards and compliance regulations to maintain data integrity and confidentiality.
8. **Scalability and Performance**: Build a system that can efficiently handle large-scale datasets while maintaining high performance and accuracy, regardless of the volume or complexity of the data.

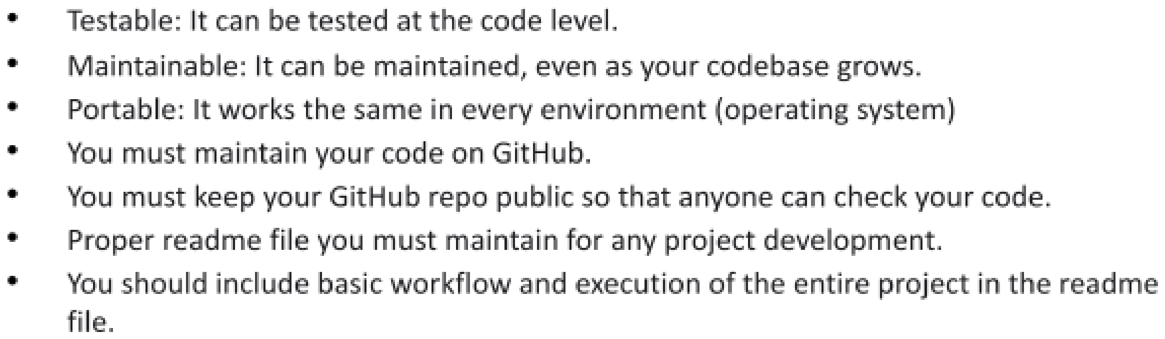
Dataset:

- Use multiple datasets of your own for this project.
  Sample: sample data

Project Evaluation metrics:

Code:

- You are supposed to write 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 must maintain your code on GitHub.
- You must keep your GitHub repo public so that anyone can check your code.
- Proper readme file you must 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 can use any Vector or NOSQL Database for this Project.

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.

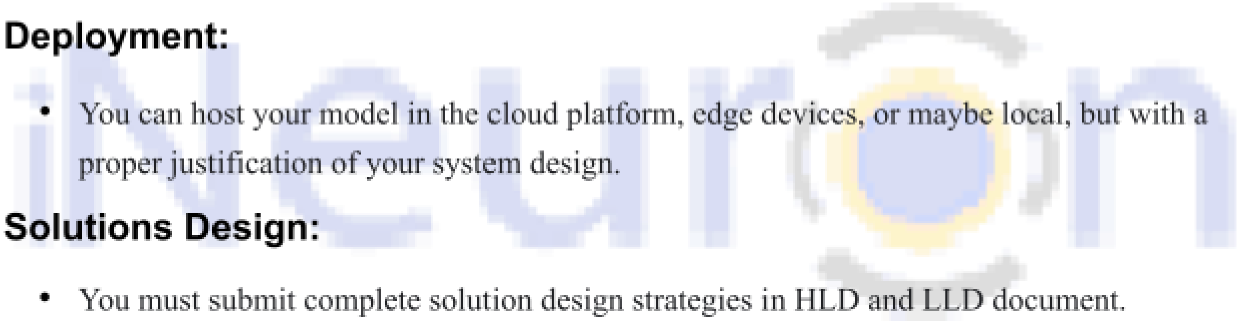
Logging:

- Logging is a must for every action performed by your code. Use the Python logging library for this.

Ops Pipeline:

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

System Architecture:



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.

- 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 the code level, and architecture level, and mention all of these things in your final submission.
- Mention your test cases for your project.

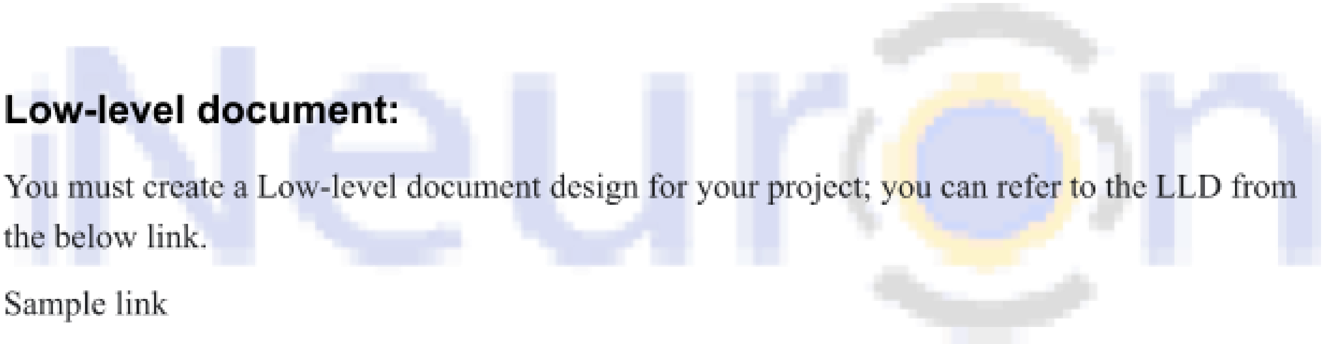
Submission requirements:

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 must 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 submission of 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 must post your project details on LinkedIn and submit that post link in your dashboard in your respective field.

Demo link

Linkedin post sample link :