iNeuron

Project Title	Auto Recruit AI
Technologies	Large Language Model
Domain	Human Resource
Project Difficulties level	Advance

Problem Statement:

Problem Statement: Traditional recruitment systems face several challenges that hinder the efficiency and effectiveness of the hiring process. The existing methods are time-consuming, prone to screening hassles, yield inconsistent results, and are often ineffective. To address these issues and revolutionise the recruitment experience, a new solution called AutoRecruit AI has been proposed. AutoRecruit AI aims to transform recruitment by placing candidates at the center of its process and leveraging a breakthrough fine tuning the LLM model. The identified challenges and their corresponding solutions are as follows:

Challenges with Traditional Recruitment System:

Time-Consuming:
Existing recruitment processes are time-consuming, leading to delays in filling crucial positions.

Screening Hassles:
The screening process is often cumbersome, making it difficult to efficiently filter through a large number of applicants.

Inconsistent Results:

Traditional methods may yield inconsistent and unreliable results, impacting the quality of hires.

Ineffective Methods:

Outdated and ineffective recruitment methods fail to identify the best-suited candidates for specific roles.

Objective:

The primary objective of the AutoRecruit AI project is to revolutionise the recruitment process by introducing a cutting-edge, personalised, and efficient system that addresses the inherent challenges of traditional hiring methods. The project aims to achieve the following specific goals:

Enhance Efficiency:

Streamline the recruitment process to significantly reduce the time required for candidate sourcing, screening, and selection.

Improve Candidate Experience:

Prioritise the candidate experience by implementing personalised engagement strategies, ensuring a positive and user-friendly interaction throughout the recruitment journey.

Ensure Accuracy and Consistency:

Implement a fine tuning of LLM model to enhance the accuracy of candidate evaluations and eliminate inconsistencies in the screening process.

Optimise Resource Utilisation:

Minimise the costs associated with recruitment by creating a cost-effective system that efficiently utilises resources while maintaining high-quality standards in candidate selection.

Facilitate Informed Decision-Making:

Provide employers with comprehensive candidate scoring and summaries, enabling them to make informed decisions quickly and effectively.

Maximize Candidate Fit:

Improve the likelihood of selecting candidates who are the best fit for specific roles by leveraging advanced algorithms and personalised engagement.

Dataset:

You must collect your dataset for this project and based on that, you must design your solution and create a repo for the dataset. You can use the below dataset for this project or you can use the list of resume pdfs and parse it to make dataset.

https://drive.google.com/file/d/1iI-gzZgKINfdzBSWaTFLu3Ab2xMWOtlS/view?usp=sharing

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 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 MongoDB for this project if needed.

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:

- 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 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 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 detail on LinkedIn and submit that post link in your dashboard in your respective field.

Demo Link

[Linkedin post sample link:](#)