PUCIT
Punjab University College of Information Technology

Revaluator

Version 1.0

.

TABLE OF CONTENTS

1. Introduction

In an era defined by rapid technological advancements, the recruitment process is at a crossroads. The need for an efficient, data-driven, and unbiased approach to identifying the right talent has never been more critical. Our project, the AI-Recruiter, is a three-tier architecture web-based system designed to revolutionize how organizations hire. By automating CV screening, generating tailored tests using a Large Language Model (LLM), and prioritizing data security, AI-Recruiter streamlines the recruitment journey. It offers a solution that enhances efficiency, reduces bias, and enables recruiters to make more informed decisions, ultimately redefining how businesses identify and select top-tier candidates in an increasingly competitive job market.

1.1 Project Title

- Revaluator - The Ai Recruiter

1.2 Project Overview Statement

Our project, Revaluator, transforms the recruitment process by harnessing the power of a three-tier architecture web-based platform. At its core, our system revolves around the implementation of a **CV filtering system**, driven by a **Large Language Model (LLM)**. Our primary objectives are to empower recruiters with an efficient tool that revolutionizes candidate selection and evaluation processes.

Key Features:

1. CV Uploading and Skill Extraction: We provide recruiters with a user-friendly interface to effortlessly upload CVs. Our system then extracts the skills and qualifications from these CVs, streamlining the initial candidate assessment.

2. CV Rating System: Our platform employs sophisticated algorithms to rate the uploaded CVs, enabling recruiters to quickly identify the most promising candidates.

3. Language Model (LLM) Testing: One of the standout features of our AI-Recruiter is the integration of a trained Language Model. This model generates customized tests based on the skills and qualifications extracted from the CVs. These tests provide valuable insights into a candidate's proficiency, enhancing the evaluation process.

4. Custom Test Creation: Recruiters have the flexibility to create custom tests using our intuitive test editor. This feature allows for tailored assessments to evaluate candidates' suitability for specific roles.

5. Candidate Test Distribution: Our platform simplifies the test distribution process. Recruiters can effortlessly send test links to candidates, ensuring a seamless testing experience.

6. Automated Evaluation: After candidates complete the tests, our system automates the evaluation process, providing recruiters with clear and objective results to inform their hiring decisions.

Our project places a strong emphasis on data security, scalability, and user experience. We are committed to ensuring that our system meets the highest standards of data protection, can adapt to varying demands, and offers a seamless experience for both recruiters and candidates.

In summary, AI-Recruiter aims to bridge the gap between job seekers and recruiters, redefining how organizations identify and select top talent. By simplifying and optimizing the recruitment journey, we aim to make the process more efficient, accurate, and user-friendly. Our versatile solution is poised to revolutionize the way companies approach candidate evaluation, ultimately leading to better hiring decisions and enhanced workforce quality.

1.3 Project Overview Statement Template

Project Title: Revaluator - The Ai Recruiter

Group Leader: BCSF20M009-Hashir Ahmad

Project Members:

Name	Roll no. #	Email Address	Signature
Hashir Ahmad	BCSF20M009	BCSF20M009@PUCIT.EDU.PK	
Muhammad Huzaifa	BCSF20M024	BCSF20M024@PUCIT.EDU.PK	
Muhammad Shazil	BCSF20M030	BCSF20M030@PUCIT.EDU.PK	
Asim Ali	BCSF20M036	BCSF20M036@PUCIT.EDU.PK	
Muhammad Qadeer	BCSF20M037	BCSF20M037@PUCIT.EDU.PK	

Project Goal:
The core objectives of our final project revolve around implementing a CV filtering system using a trained Language Model (LLM). We aim to provide recruiters with an efficient tool for enhancing candidate selection processes, with initial and skill proficiency tests playing a pivotal role in its functionality and utility.

Objectives:

Sr.#	
1	Develop a system that automates the initial screening of CVs, saving recruiters time and effort in the candidate selection process.
2	Implement a feature that extracts key skills and qualifications from CVs, providing recruiters with valuable insights into candidate profiles.
3	Create an algorithm to objectively rate CVs, enabling recruiters to quickly identify high-potential candidates.
4	Integrate a trained Language Model(LLM) to generate customized tests based on candidate skills and qualifications extracted from CVs.
5	Provide a user-friendly test editor that allows recruiters to create custom assessments tailored to specific job requirements.
6	Develop a system for recruiters to easily send test links to candidates, ensuring a seamless testing experience.
7	Design the system to be scalable, accommodating the evolving needs of organizations as they grow and adapt to changing recruitment criteria.
8	Implement automated evaluation mechanisms to provide recruiters with objective results from candidate tests.

Project Success criteria:
The project's success is contingent on the accurate and efficient CV filtering enabled by automated screening and skills assessments.

Assumptions, Risks and Obstacles:
CVs are assumed to be well-documented and readable.
Potential obstacles may arise if CVs lack clarity or readability, impacting the accuracy of the filtering process.

Type of project: ❑Research ■ Development

Target End users:
1- Company's HR

Development Technology: ■ Object Oriented ❑Structured

Platform: ■ Web-based ❑Distributed
❑Desktop based ❑Setup Configurations
❑Other_____

Suggested Project Supervisor: Dr.Madiha Khalid

Approved By:

Date:

1.4 Project Goals & Objectives

Goal

The core objectives of our final project revolve around implementing a CV filtering system using a trained Language Model (LLM). We aim to provide recruiters with an efficient tool for enhancing candidate selection processes, with initial and skill proficiency tests playing a pivotal role in its functionality and utility.

Objectives

- Automated CV Screening
- Skills Extraction
- CV Rating System
- Language Model (LLM) Training
- Custom Test Creation
- Candidate Test Distribution
- Automated Evaluation
- Data Security and Scalability
- User-Friendly Interface
- Streamlined Recruitment

1.5 High-level system components

The AI-Recruiter project comprises several high-level system components that collectively power our innovative recruitment platform. These components include the User Interface for user interaction, CV Processing for automated screening, Language Model (LLM) integration for test generation, a Test Editor for customization, Communication and Notification management, robust Database and Data Storage, Security measures, Scalability and Performance optimization, Reporting and Analytics tools, and seamless Integration with external systems. Together, these components create a comprehensive and efficient solution for recruiters and candidates in the recruitment process.

Our necessary system components are:

- User Interface (UI)
- CV Processing Module
- Language Model (LLM) Integration
- Test Generation Module
- Test Editor
- Test Distribution
- Test Taken
- Communication and Notification System
- Database and Data Storage
- Security and Access Control
- Scalability and Performance Optimization
- Reporting and Analytics
- Integration and Deployment

1.6 List of optional functional units

- Feedback module.
- Payment

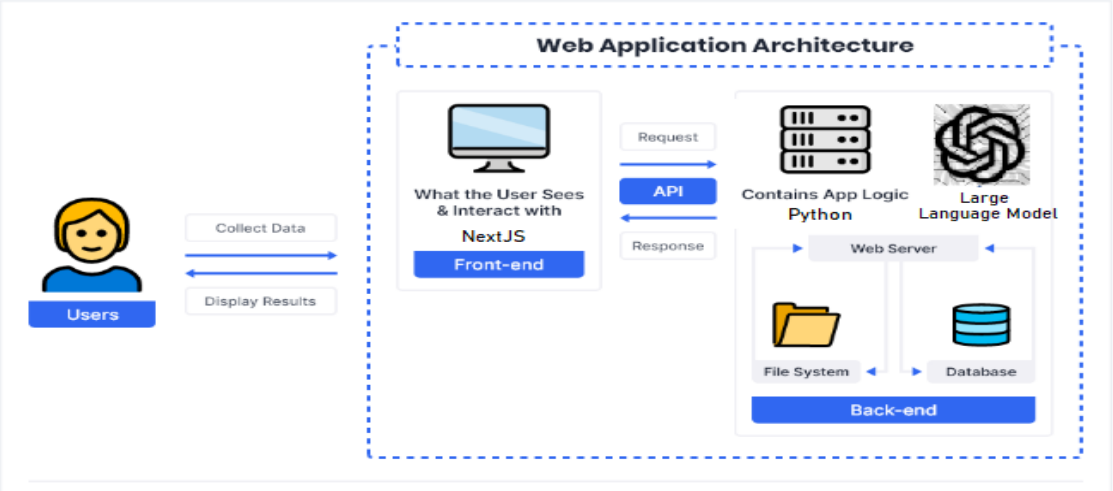
1.7 Exclusions

There will be no exclusions in our project.

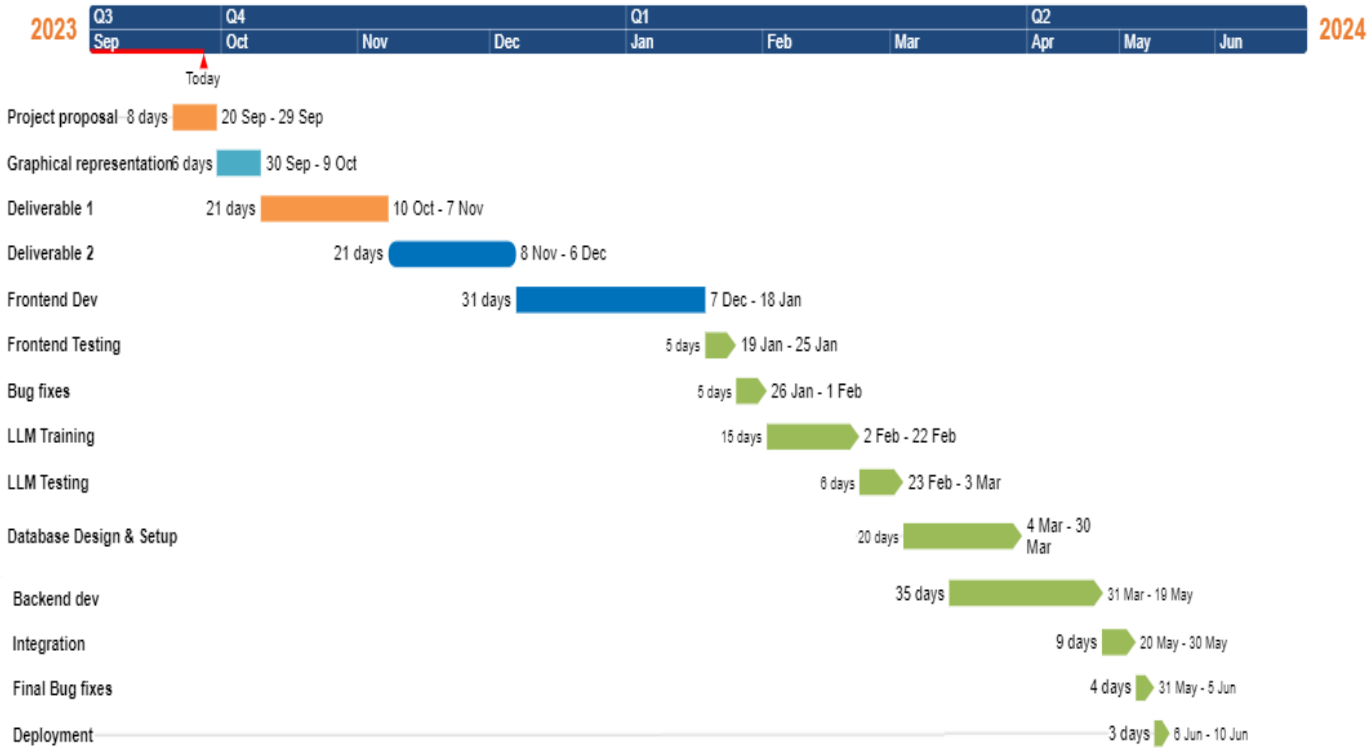
1.8 Application Architecture

We will follow 3-tier Architecture.

- Next.Js (Front End)
- Python (Back End)
- MongoDB/Postgres(Databas



1.9 Gantt chart



1.10 Hardware and Software Specification

Client-Side Hardware Requirements:

- Mobile phone or desktop.
- Minimum 512 MB RAM.

Client-Side Software Requirements

- Windows 7/8/10/11, MAC OS, Linux.
- Browser (Chrome , Opera , Firefox , Apple Safari)

1.11 Tools and technologies used with reasoning

Tools:

- **VS Code**
  We leverage Visual Studio Code as our primary code editor, for backend and frontend development, it is light-weight, extensible, free, open source and cross-platform.

- **Git and GitHub**
  Git version control combined with GitHub repository hosting enable seamless collaboration and code management within our development team.

- **Jupyter Notebook**
  Jupyter Notebook provides an interactive environment for data analysis and experimentation, aiding in the development of our machine learning models.

- **MS Project**
  Microsoft Project helps us efficiently manage project timelines, allocate resources, and track progress, ensuring our project stays on schedule.

- **Figma/ Canva**
  Figma/Canva serves as a collaborative design platform, enabling our team to create and refine user interfaces that enhance the user experience of our CV filtering system.

- **PgAdmin and DBeaver**

  These database management tools, PgAdmin for PostgreSQL and DBeaver for various databases, assist in database administration and schema design.

Technologies:

- **Next.js**

  We employ Next.js to build a high-performance, server-rendered React application, delivering a responsive and engaging front-end for our CV filtering system.

- **Language Model (LLM)**

  LLM plays a central role in our project by generating customized tests based on candidate skills and qualifications extracted from CVs. It enables personalized assessments and enhances the evaluation process.

- **Python**

  Python serves as the primary programming language for our project, facilitating machine learning model development, data processing, and scripting tasks.

- **MongoDB/PostgreSQL**

  Depending on the project requirements, we use MongoDB for flexible, NoSQL data storage or PostgreSQL for structured, relational data storage, ensuring data integrity and retrieval efficiency.

- **Prisma ORM**

  Prisma ORM streamlines database access and management, allowing us to work with databases using a type-safe and intuitive API, improving the overall robustness of our system.