PRESENTED BY:
FAROOQ AHMAD KHAN
REEMA KAMAL
SYED MANAN.

AI POWERED HR APPLICATION

ABOUT

TThe AI-powered job portal represents a sophisticated platform engineered to enhance the recruitment ecosystem by seamlessly connecting job seekers and hiring managers through advanced artificial intelligence capabilities, driven by the Mistral Large Language Model (LLM). This system integrates a comprehensive suite of functional features designed to optimize the job-matching process. For candidates, the platform facilitates user authentication, enabling registration, login, and profile management. It employs AI-powered CV processing to extract critical data—such as experience, education, skills, location, and languages—from uploaded CVs. Candidates receive tailored AI-generated job insights, including an assessment of their job suitability and specific recommendations for CV improvement, such as adding missing qualifications or refining skill descriptions. An innovative AI-based CV enhancement feature further refines and restructures CVs to maximize visibility and appeal to prospective employers. For hiring managers, the portal provides a dynamic dashboard that leverages AI to rank candidates based on education, experience, and skills, supplemented by detailed reasoning and summaries for each ranking to expedite informed decision-making. The job application process allows candidates to apply effortlessly, with the AI system ranking applicants according to job relevance. To ensure broad accessibility, the platform incorporates full English and German language support via the Google Translate API. Deployment on Microsoft Azure, coupled with a CI/CD pipeline automated through GitHub Actions, guarantees efficient updates, rigorous testing, and operational reliability.

The platform adheres to stringent non-functional requirements to deliver a robust and user-friendly experience. Scalability is achieved through Azure App Service, which supports thousands of simultaneous users without compromising performance. Security is prioritized with encrypted authentication and secure handling of sensitive user data, ensuring compliance with privacy standards. Performance optimization ensures swift AI responses, rapid job searches, and efficient filtering, minimizing latency for end users. Usability is enhanced by a minimalistic, intuitive interface that simplifies navigation for both candidates and hiring managers. The system maintains a 99.9% uptime, leveraging Azure's high-availability cloud infrastructure, while the CI/CD pipeline streamlines development and deployment processes. Localization, powered by AI-driven English-German language switching, broadens the platform's reach. The technical architecture comprises a responsive frontend built with HTML, CSS, and JavaScript; a Flask-based Python backend; and a flexible database layer starting with SQLite, scalable to Azure SQL/PostgreSQL. The Mistral LLM drives AI processing, while Google Translate enables seamless translation. Azure services—including App Service for hosting, SQL/PostgreSQL for data storage, and Azure Storage for CV management—combined with GitHub Actions for CI/CD automation, form a resilient, scalable, and secure foundation for the platform's operations.

THE INTERNAL TEAM

Muhmmad Farooq Khan

AI engineer

Syed Manan

UI/UX designer

Reema Kamal

Technical writing

CHALLANGES :

The development of the AI-powered hiring platform presented a series of technical and design challenges that required innovative solutions to ensure the system met its functional and non-functional requirements. One of the primary hurdles in the AI engineering phase was the integration and optimization of the ColBERT and SBERT models for candidate matching. These models, while highly effective for semantic search and text embedding, demanded significant computational resources, posing a challenge in achieving real-time performance for ranking candidates from the FAISS database. Balancing accuracy with response time necessitated extensive experimentation with model configurations and indexing strategies, as initial iterations struggled to process large datasets within acceptable latency thresholds. Additionally, the resume summarization process using the Mistral 7B and T5 models encountered difficulties in consistently extracting structured data from diverse resume formats (PDF, DOCX). Variability in document layouts, inconsistent terminology, and incomplete candidate information led to occasional inaccuracies in skill and experience extraction, requiring the team to implement robust preprocessing pipelines and fallback mechanisms to enhance reliability.

From a system design perspective, ensuring scalability and seamless interaction between the Flask framework backend and the frontend (index.html) introduced further complexities. The API needed to handle concurrent requests from multiple hiring managers and candidates, yet early load testing revealed bottlenecks in database queries and AI inference calls, necessitating optimization of the FAISS indexing process and the adoption of asynchronous task handling. Security posed another critical challenge, as the platform required secure authentication (username/password with optional OAuth) and protection of sensitive candidate data. Implementing encryption and ensuring compliance with data privacy standards added layers of complexity to the design, particularly under tight development timelines. Usability also emerged as a concern during the design phase; creating an intuitive interface for hiring managers to apply filters (e.g., location priority, skill match) and for candidates to enhance their profiles demanded iterative user testing, as initial designs risked overwhelming users with technical options. Finally, the optional admin workflow introduced challenges in providing flexible yet controlled access to AI ranking criteria and performance logs, requiring a delicate balance between configurability and system stability to prevent unintended disruptions.

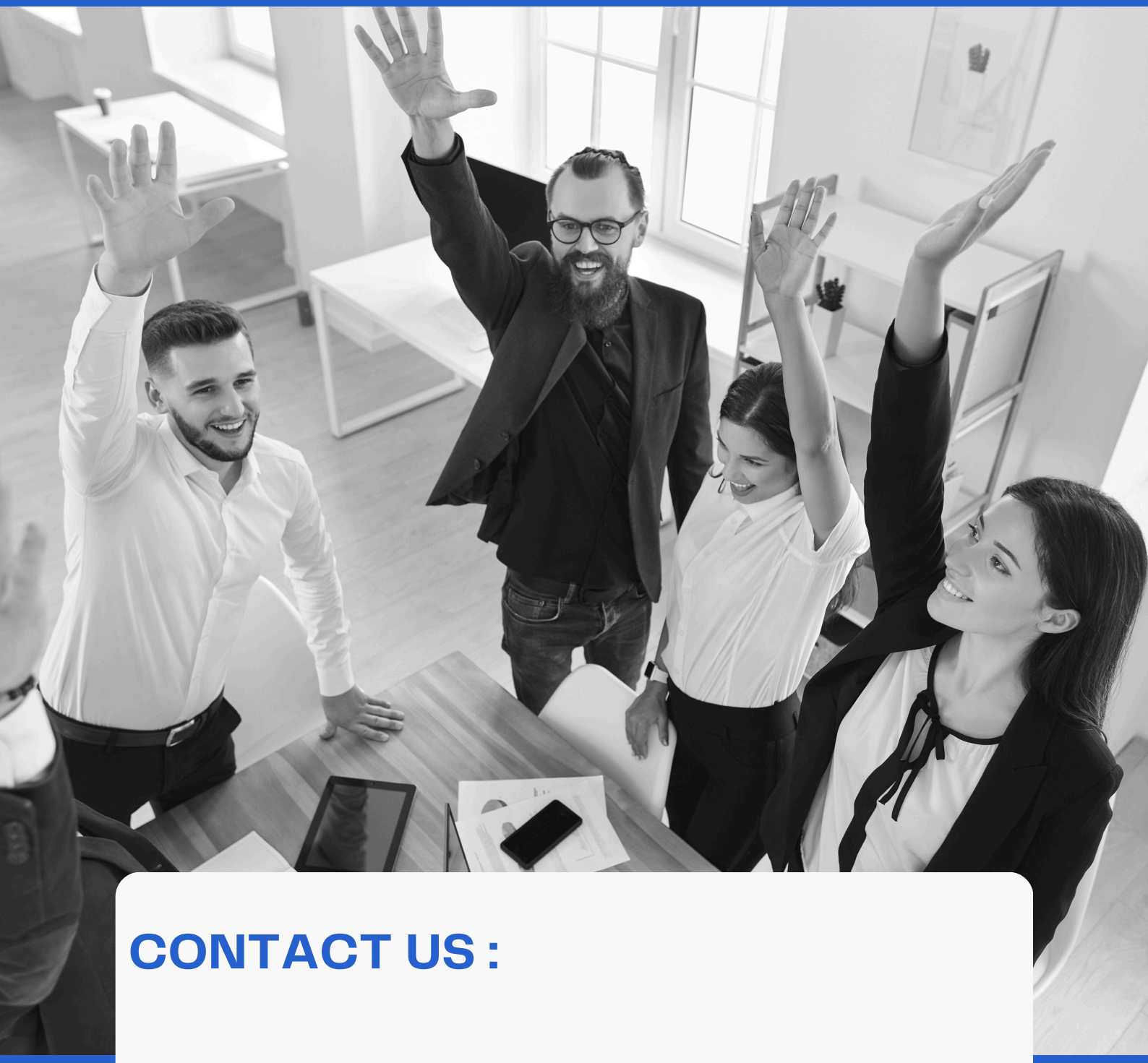
Despite these obstacles, the team addressed each challenge through a combination of technical ingenuity, rigorous testing, and user-focused design adjustments. These efforts ultimately resulted in a robust, efficient, and user-friendly hiring platform capable of meeting the diverse needs of its stakeholders.

TECHNICAL ARCHITECTURE

The technical architecture of the AI-powered hiring platform is designed to deliver a scalable, efficient, and user-centric solution that seamlessly integrates frontend interfaces, backend processing, and advanced AI models to facilitate job-candidate matching. The system is structured as a modular, layered architecture comprising the presentation layer, application layer, AI processing layer, and data storage layer, all orchestrated to meet the platform's functional and non-functional requirements.
At the presentation layer, the platform leverages a web-based interface anchored by index.html, serving as the entry point for hiring managers, candidates, and optional admin users. This frontend is built using modern HTML, CSS, and JavaScript frameworks to ensure a responsive and intuitive user experience. For hiring managers, the dashboard provides a job search panel with input fields for job titles, skills, and locations, alongside filter options (e.g., Location Priority, Best Skill Match, Availability). Candidates interact with a profile creation and job-matching interface, while admins access a management dashboard for database and AI oversight. The frontend communicates with the backend via RESTful API calls, ensuring smooth data exchange and real-time updates.

The application layer is powered by a Flask-based framework, acting as the central orchestrator of the platform's workflows. Implemented in Python, Flask handles incoming requests from the frontend, such as login authentication (username/password with optional OAuth), candidate searches, and profile updates. It routes these requests to the appropriate backend services and returns processed results, such as AI-ranked candidate lists or job matches. The framework is designed for scalability, utilizing asynchronous task processing to manage high concurrency and integrating middleware for security features like token-based authentication and input validation.

AI processing layer forms the core of the platform's intelligent functionality, leveraging state-of-the-art models for candidate matching and resume summarization. For hiring managers' candidate searches, the system employs ColBERT and SBERT models to generate semantic embeddings of job requirements and candidate profiles. These embeddings are indexed and searched using a FAISS database, enabling efficient similarity matching based on skills, experience, and location. The AI ranks candidates by applying weighted filters, configurable via the admin interface, and delivers results to the Flask framework. For candidates, resume uploads (PDF/DOCX) are processed by Mistral 7B or T5 models, which extract structured data (skills, experience, location) and generate concise summaries. This layer is optimized for performance, with precomputed embeddings and batch processing to minimize latency, and is deployed on a cloud infrastructure (e.g., AWS or GCP) with GPU support for model inference



CONTACT US :

03151582072
03316353349
03175256498

reemach496@gmail.com