**Internship Assignment Report: Candidate Recommendation Web Application**

1. Introduction

The aim of this project was to develop a web application to assist employers in identifying the most suitable candidates for a job role from a pool of resumes. The web application allows users to upload resumes, specify the job role and its description, and then generates recommendations based on the content of the resumes.

2. Problem Statement

Employers often face challenges in efficiently sifting through numerous resumes to find the ideal candidates for a job role. The goal was to streamline this process by leveraging technology to automate resume analysis and recommendation generation.

3. Approach and Implementation

3.1 Frontend

- Technology Stack: React.js was chosen for the frontend development along with react-dropzone for file uploads and Firebase storage for file hosting.

- User Interface: A user-friendly interface was designed to facilitate resume uploads, selection of resumes for analysis, and input of job role details.

- Global State Management: Utilized React's useContext hook for managing global states related to files, URLs, and results.

- Libraries Used: Employed notistack for error handling, react-icons for icons, and Material UI for UI components.

- Performance Optimization: Consideration given to optimizing bundle size and performance by potentially replacing notistack and react-icons with Material UI alternatives.

3.2 Backend

- Technology Stack: Django REST Framework (DRF) was employed for building the backend API.

- API Endpoint: Created an API endpoint to receive job role details and a list of uploaded resumes for processing.

- File Processing: Used the PYPDF2 library to extract text data from PDF resumes.

- Integration: Integrated django-cors-headers to handle Cross-Origin Resource Sharing (CORS) issues.

- Error Handling: Utilized DRF's exception handler to manage errors during API processing.

3.3 OpenAI API Integration

- Model Selection: Utilized the 'gpt-4-0125-preview' model from OpenAI for optimized responses.

- Structured Prompts: Employed structured prompts for scoring and detail retrieval to ensure accurate responses.

- Error Handling: Implemented error handling mechanisms for handling errors and mathematical operations.

3.4 Scoring Algorithm and Hyperparameters

- Scoring Method: Developed a custom scoring method dividing resumes into sections (Education, Projects, Past Experience, etc.) and assigning maximum scores.

- Normalization: Normalized scores and aggregated them to compute a final score out of 100.

- Hyperparameters: Defined hyperparameters based on the scoring method and normalization process.

4. Evaluation Results

- Evaluation Process: Conducted evaluations to assess the performance of the web application.

- Metrics: Evaluated performance based on response time, accuracy of recommendations, and user feedback.

- Challenges: Faced challenges such as long response times due to API calls and potential errors in mathematical operations.

- Mitigation: Addressed challenges through optimization efforts and error handling mechanisms.

5. Insights Gained

- Project Reflection: Reflecting on the project experience, insights were gained into the complexities of resume analysis and recommendation generation.

- Lessons Learned: Learned valuable lessons regarding frontend-backend integration, API development, and handling third-party APIs.

- Future Enhancements: Identified potential areas for improvement such as performance optimization, persistence of data using local storage, and refining the scoring algorithm.

6. Conclusion

In conclusion, the development of the candidate recommendation web application provided valuable insights into the challenges and opportunities of automating the hiring process. Through the implementation of frontend and backend technologies, integration of third-party APIs, and design of custom scoring algorithms, the project aimed to address the needs of employers in efficiently identifying suitable candidates. While challenges were encountered during the development and evaluation process, the project serves as a foundation for future enhancements and improvements in automating the recruitment process.

---

This report provides a comprehensive overview of the project, detailing the approach, implementation, evaluation results, insights gained, and concluding remarks. Further details such as code snippets, diagrams, and performance metrics can be included to supplement the report as needed.