**AI Candidate Assessment System**

**FYP– I REPORT**

**BS(CS) Fall 2024**

21K-3451 Ashesh Kumar

21K-4589 Abdul Wasay

21K-4926 Fahad Ahmed



**Supervisor: Ms.Nida Munawar**

**Department of Computer Science**

**FAST- National University of Computer &**

**Emerging Sciences, Karachi**

**Contents**

Introduction……………………………………………………………………….........1

[Related Work…………………………………………………………………………...2](#_Toc12530)

[Methodology ……………………………………………………………………………3](#_Toc12531)

Testing & Results……………………………………………………………………….4

System Diagram………………………………………………………………………..5

Goals for FYP-II………………………………………………………………………...6

Conclusion………………………………………………………………………………7

References……………………………………………………………………………...8

## 1. Introduction

The recruitment process is a critical component of organizational success, yet traditional methods often fall short in meeting the demands of today’s competitive job market. Manual CV screening, test creation, and candidate evaluations are time-consuming, inconsistent, and susceptible to human biases, resulting in inefficiencies and delays. The need for scalable, objective, and automated solutions has become more pressing as organizations face growing applicant volumes and increasing pressure to make data-driven hiring decisions.

The AI Candidate Assessment System is developed to address these challenges by leveraging state-of-the-art Artificial Intelligence (AI) technologies. The system integrates SBERT for precise CV analysis, enabling the extraction of relevant skills and qualifications, and GPT for generating personalized, skill-specific tests tailored to job requirements. By automating critical HR processes, the system streamlines workflows, reduces manual effort, and ensures fairer and faster candidate evaluations.

Phase 1 of the project focuses on implementing the CV classification module, where resumes are analyzed, classified, and matched with job descriptions based on semantic and contextual relevance. This phase lays the foundation for subsequent functionalities, such as test generation and automated performance evaluations, creating a comprehensive end-to-end solution for recruitment automation. The AI Candidate Assessment System aims to transform hiring practices, making them more efficient, scalable, and equitable.

# 2. Related Work

SRS/SDS soft copy provided

# 3. Methodology

# Goals:

# Focus on developing the core components, including database schema, initial frontend development, and AI integration for CV parsing.

* **Requirement Gathering and Analysis:**
  + Conducted sessions with team members and the supervisor to gather system requirements.
  + Drafted the **Software Requirement Specification (SRS)** document, outlining system features and objectives.
  + Identified the gaps in existing AI-driven recruitment platforms to define the project’s unique goals.
* System Design:
  + Designed the **system architecture** and created detailed mockups using tools like Figma.
  + Developed the **Software Design Specification (SDS)**, detailing technical specifications, data flows, and backend architecture.
  + Created **Entity-Relationship Diagrams (ERD)** and **sequence diagrams** for key workflows such as CV parsing and test generation.
* Database Schema Development:
  + Implemented a **MongoDB database schema**, focusing on scalability and flexibility.
  + Finalized schema entities such as User\_Details, Candidate, HR, Job\_Description, and Report.
* Prototype for AI-Powered CV Parsing:
  + Researched AI technologies like **SBERT** and implemented a prototype for **CV classification**.
  + Tested the CV classification functionality and refined the implementation based on feedback.

# 4. Testing and Results

## 1. Testing Approach

## a) Database Schema Validation

## Objective: Verify the correctness of the database schema and its ability to store and retrieve data as per the project requirements.

## Focus Areas:

## Data structure for entities like User\_Details, Candidate, HR, and Job\_Description.

## Relationships and constraints between entities (e.g., one-to-many and many-to-one relationships).

## Tools Used: MongoDB Compass for schema validation and sample data insertion.

## b) CV Parsing Prototype

## Objective: Test the functionality of the CV parsing prototype to ensure it extracts relevant information accurately.

## Focus Areas:

## Extracting key details like skills, experience, and qualifications using the SBERT model.

## Handling resumes in various formats (e.g., PDF, DOCX).

## Tools Used: Python-based implementation of SBERT for CV analysis.

## c) Frontend Mockup Validation

## Objective: Validate the usability and completeness of the designed frontend mockups.

## Focus Areas:

## Reviewing the visual design and user workflows for candidate registration, CV upload, and HR review.

## Ensuring the mockups covered all intended system functionalities.

## Tools Used: Figma for mockup creation and validation.

## d) Workflow Validation

## Objective: Ensure that the proposed workflows align with project objectives and technical feasibility.

## Focus Areas:

## Workflow for candidate registration, CV upload, and automated analysis.

## HR workflows for viewing parsed data and managing candidates.

## Tools Used: Sequence diagrams and flowcharts created in PlantUML and draw.io.

## 2. Test Cases and Results

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Description | Expected Outcome | Result |
| Database Schema Validation | Validate relationships and constraints in MongoDB. | Relationships are correctly enforced. | Pass |
| Sample Data Insertion | Insert mock data for User\_Details and Candidate. | Data is stored and retrieved as expected. | Pass |
| CV Parsing Prototype | Parse a sample CV in PDF format. | Extract key skills and qualifications. | Pass |
| CV Parsing Error Handling | Parse unsupported file formats (e.g., images). | Display an error message. | Pass |
| CV Classification: Skill Match | Analyze CV for skills matching a job description. | Identify skills and assign matching score. | Pass |
| CV Classification: Invalid Input | Provide corrupted CV for analysis. | System flags the CV as invalid. | Pass |
| CV Classification: Formatting | Parse CVs with various layouts (e.g., tabular, paragraph). | Extract consistent information. | Pass |
| Frontend Mockup Validation | Review Figma designs for completeness. | Mockups cover all intended functionalities. | Pass |
| Workflow Validation | Validate sequence diagrams for candidate workflows. | Workflow aligns with system objectives. | Pass |

## 3. Results and Analysis

## a) Database Schema

## The MongoDB schema was validated with mock data, ensuring relationships between entities like User\_Details, Candidate, and HR were correctly implemented.

## Sample queries demonstrated the schema’s ability to handle expected data loads.

## b) CV Parsing Prototype

## The SBERT-based CV parsing prototype achieved an accuracy of 90% in extracting key details from structured resumes.

## Identified limitations in handling unstructured or poorly formatted resumes, which will be addressed in FYP-2.

## c) Frontend Mockups

## The Figma mockups were reviewed and approved by team members and the supervisor.

## The designs provided a clear representation of the user workflows, such as candidate registration, CV upload, and HR review processes.

## d) Workflow Validation

## Sequence diagrams and flowcharts validated the logical flow of data and user actions.

## Feedback from the team led to minor adjustments in the workflows, improving system clarity and feasibility.

## 4. Summary

## The testing phase of FYP-1 successfully validated the foundational aspects of the system:

## The database schema and CV parsing prototype were tested and met the initial requirements.

## The frontend mockups provided a clear visual representation of user workflows.

## Logical workflows were validated through diagrams and team reviews.

## The results of FYP-1 form a solid base for FYP-2, where the focus will shift to implementing advanced features like test generation, evaluation, and deployment of the complete system.

## 5. Snippets of Testing and Results:

A close-up of a document

Description automatically generated

A white paper with text on it

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## 5. System Diagram

## 

## 6. Goals for FYP-II

**FYP-2 focuses on building upon the foundation of FYP-1 to deliver a fully functional, scalable, and deployable AI Candidate Assessment System. The primary goals are:**

1. **Integration of AI Models**
   * Enhance the CV parsing functionality with an optimized **SBERT model** for improved skill extraction and candidate profiling.
   * Implement **GPT models** to generate tailored skill-based tests for candidates and evaluate their responses automatically.
2. **Development of Advanced Features**
   * Automate test generation based on job descriptions and candidate skills.
   * Create a robust module for test evaluation and generate detailed performance reports.
3. **Full Stack Implementation**
   * Develop a dynamic and responsive frontend using **React**, connecting seamlessly with the backend APIs.
   * Finalize and refine backend workflows in **Node.js** and integrate them with the **MongoDB** database.
4. **Scalability and Deployment**
   * Deploy the system on a cloud platform (e.g., AWS or Heroku) for real-world usability.
   * Ensure scalability to handle multiple concurrent users.
5. **Testing and Optimization**
   * Conduct comprehensive testing, including **unit testing**, **integration testing**, and **user acceptance testing** (UAT).
   * Optimize performance to meet response time benchmarks.
6. **User Experience and Feedback**
   * Enhance UI/UX to ensure ease of use for candidates and HR users.
   * Incorporate feedback from real-world testing to improve system functionality.
7. **Documentation and Presentation**
   * Prepare technical documentation, including a user manual and deployment guide.
   * Develop a comprehensive presentation for project evaluation.

## 7. Conclusion

The successful completion of FYP-1 established a strong foundation for the AI Candidate Assessment System by addressing the project's critical initial components. The database schema was validated to ensure data integrity and effective relationships among key entities. The CV parsing prototype demonstrated the system's capability to extract essential details from resumes accurately, achieving a high accuracy rate. The frontend mockups provided a clear and functional representation of user workflows, while the workflow validation confirmed the system's feasibility and alignment with project goals.

Through rigorous testing and validation, the foundational components were proven to meet the project's requirements and objectives. These achievements set the stage for FYP-2, which will focus on implementing advanced functionalities, including AI-driven test generation and evaluation, full-stack development, and deployment. By building upon the work completed in FYP-1, the project remains on track to deliver an innovative, scalable, and efficient solution for AI-powered candidate assessment.

## 8. References

 Hugging Face, "SBERT Model Documentation," [Online]. Available: <https://huggingface.co/docs/>. [Accessed: Dec. 12, 2024].

 Hugging Face, "SBERT Model Documentation," [Online]. Available: <https://huggingface.co/sentence-transformers>. [Accessed: Dec. 12, 2024].

 A. Deshmukh, A.Raut, "Applying BERT-Based NLP for Automated Resume Screening and Candidate Ranking," *Springer*, 2024. [Online]. Available: [https://www.researchgate.net/publication/378829570\_Applying\_SBERT-Based\_NLP\_for\_Automated\_Resume\_Screening\_and\_Candidate\_Ranking](https://www.researchgate.net/publication/378829570_Applying_BERT-Based_NLP_for_Automated_Resume_Screening_and_Candidate_Ranking). [Accessed: Dec. 12, 2024].

 A. V. Vaswani et al., "Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks," *NAACL 2021*, 2021. [Online]. Available: <https://arxiv.org/abs/2010.08240>. [Accessed: Dec. 12, 2024].

 Grenze International Journal of Engineering and Technology, "Resume Parsing using Natural Language Processing," *GRENZE*, vol. 9, no. 1, 2023. [Online]. Available: <https://thegrenze.com/pages/servej.php?fn=587.pdf&name=Resume%20Parsing%20using%20Natural%20Language%20Processing&id=1418&association=GRENZE&journal=GIJET&year=2023&volume=9&issue=1>. [Accessed: Dec. 12, 2024].