Project Specification Report - RecruitAssistant

Deniz Öztürk - 22102126
Emir Ensar Sevil - 22201926
Yüksel Barkın Baydar - 22201939
Mehmet Anıl Yeşil - 22102614
Cankutay Dündar - 22103284

1. Introduction

In today's highly competitive job market, candidates are expected to demonstrate not only strong technical and communication skills but also the ability to adapt their applications to rapidly evolving industry standards. Traditional career preparation tools often lack personalization and fail to address the full scope of the hiring journey. RecruitAssistant aims to bridge this gap by providing an AI-powered, end-to-end career guidance platform that supports users from CV creation to final interview performance. By leveraging modern machine learning, NLP-driven evaluation, and real-time interaction capabilities based on recent advances in transformer architectures and speech technologies [1], [3], [15], [16], the project aims to deliver a tailored and scalable solution that enhances user readiness and improves overall hiring outcomes.

1.1 Description

RecruitAssistant is an AI-powered, end-to-end career assistance platform designed to guide users through the entire lifecycle of the hiring process, from CV generation to successfully completing final interview rounds. The platform acts as a personalized career companion, offering structured, data-driven support tailored to each user's background, skill set, and career goals.

At its core, RecruitAssistant provides intelligent tools such as job-specific CV and cover letter generation, dynamic skills-gap analysis, and personalized learning paths that help users understand what employers expect in their target roles. Users can engage in mock interview simulations—both HR-style behavioral interviews and technical interviews—powered by advanced large language models capable of understanding, evaluating, and scoring responses based on industry standards [1], [15]. Combined with real-time speech-to-text, and feedback on clarity, tone, and structure, the system offers a deeply realistic interview environment that benefits from recent progress in self-supervised speech representation learning [3], [16].

The platform also includes pre-interview preparation modules, such as customizable quizzes, domain-specific question sets, scenario-based problem solving, and automated feedback that highlights strengths and weaknesses. These interactive assessments help users build confidence and cover their weaknesses before high-stakes interviews.

From a technical perspective, RecruitAssistant integrates modern machine learning pipelines, NLP transformers, and evaluation models to deliver accurate, context-aware guidance [1], [3], [4], [15]. The system architecture supports modular components, making it easy to update or replace models, add new job domains, or incorporate advanced analytics. Its backend is designed for high scalability, supporting large numbers of simultaneous users, asynchronous processing of long interview sessions, and efficient data storage for personalized progress tracking, following well-established architectural principles for network-based software systems [2], [14].

The platform is engineered to be fully cross-platform, allowing deployment on web browsers. Its flexible API infrastructure enables integration with external job platforms.

Overall, RecruitAssistant delivers a comprehensive, adaptive, and user-centric solution that assists its users through their recruitment journey. By combining advanced AI, intuitive

design, and scalable engineering, the system helps users not only prepare effectively but also continuously improve as they progress toward their career goals.

1.2 High Level System Architecture & Components of Proposed Solution

The system will consist of the following major components:

1. Frontend Interface

- Web and mobile interface for user interaction
- Modules for quiz-taking, mock interview UI, and CV customization, and their analysis
- Provides dashboards, progress tracking, and feedback summaries
- A user-friendly, responsive interface to deliver a smooth experience across all devices, ensuring that users can easily access tools, track their progress, and interact with AI-driven features without technical complexity, following modern web-application design principles and REST-based client–server interaction patterns [2], [14].

2. Backend Service Layer

- RESTful API endpoints for managing user requests [2], [14]
- Business logic for CV generation, interview question generation, quiz evaluation, and result computation
- Secure authentication and session handling with secure development guidelines and best practices [7], [11], [14]
- A scheduler that manages available mock interview times, reminders, and automated session summaries

3. AI/ML Engine

- **NLP Models:** CV generation, job description parsing, interview question generation, evaluation feedback, leveraging transformer-based architectures and modern NLP libraries [1], [4], [15]
- **Speech-to-Text Models:** For real-time mock interview transcription using state-of-the-art speech representation models [3], [16]
- **Evaluation Models:** Scoring candidate answers, identifying weak topics, and generating personalized study content, while incorporating fairness-aware ML practices [4]
- Sentiment and tone analysis models that evaluate user confidence, clarity, and communication style during interviews

4. Database & Storage

- User profiles, quiz results, CV versions, mock interview recordings/transcripts
- Scalable storage for logs, metadata, and learning materials

- Analytics tables storing performance metrics to track user improvement over time, designed to support long-term monitoring and analysis of user progress

5. Integration Layer

- Optionally connects to external APIs such as job boards, LinkedIn-style profile imports, or external datasets, making use of well-structured REST and web API practices [2], [14]
- Model hosting services for deploying NLP and STT models at scale [14], [15], [16]

1.3 Constraints

1.3.1 Implementation Constraints

- Limited computational resources for real-time model inference (e.g., STT or large NLP models) [1], [3], [16]
- Dependency on open-source libraries (Hugging Face, ONNX Runtime, vector DBs) [4], [15]
- Need for cross-platform compatibility (browsers, screen sizes, mobile devices)
- Availability of high-quality datasets for fine-tuning (CVs, interview transcripts, quiz banks) [4]
- Performance constraints for interactive features such as mock interviews and feedback

1.3.2 Economic Constraints

- Budget limitations regarding cloud GPUs, storage, and hosting [13], [14]
- Preference for open-source or self-hosted models to avoid high API billing [4], [15]
- Costs associated with dataset preparation, annotation, or fine-tuning [4], [13]
- Minimizing operational expenses while supporting growing user activity through scalable cloud architectures [13], [14]

1.3.3 Ethical Constraints

- Ensuring fairness in scoring interviews and quizzes [4]
- Avoiding bias in CV generation or job suggestions [4]
- Respecting user privacy and sensitive career information [5], [6], [7]
- Avoiding unethical automation that manipulates hiring processes
- Compliance with local and international data protection laws (GDPR, KVKK) [5], [6]

1.4 Professional and Ethical Issues

Professional and ethical issues relevant to this project include:

- **Data Privacy & Security:** Protecting user resumes, interview recordings, and personal details, in accordance with GDPR, KVKK, and information security standards [5], [6], [7].
- **Transparency:** Users must be informed how their data is processed and how model results are produced [4], [5].
- **Fairness & Bias Mitigation:** Preventing discrimination based on gender, age, nationality, or background, and applying fairness-aware ML principles [4].
- **Responsibility:** Providing accurate and non-misleading career guidance, avoiding overconfident or harmful recommendations [4].
- **Accountability:** Logging system operations and model decisions for audits when needed [7], [8], [9].
- **User Safety:** Avoiding harmful, discriminatory, or inappropriate interview questions or feedback, consistent with responsible AI and secure development principles [4], [11].

1.5 Standards

The following engineering and documentation standards will be used:

- **IEEE 830 – Software Requirements Specification standard** for structuring and documenting system requirements [8].
- **UML 2.5.1** – For modeling system diagrams (class diagrams, activity diagrams, sequence diagrams) [10].
- **IEEE 1016 – Software Design Documentation** for organizing and presenting the design artifacts [9].
- **ISO/IEC 27001 – Information security management** for guiding data protection, access control, and risk management measures [7].
- **OWASP Guidelines** – For secure development and authentication practices in the web application [11].
- **REST API Design Best Practices** – For endpoint structuring, pagination, error handling, and authentication [2], [14].

---

2. Design Requirements

2.1 Functional Requirements

- The system shall generate job-specific CVs using NLP-based templates [1], [15].
- The system shall provide mock interview sessions using generated or predefined question sets.
- The system shall transcribe user answers in real-time using speech-to-text models [3], [16].
- The system shall score interview answers and provide personalized feedback [1], [4].
- The system shall generate pre-interview quizzes tailored to job roles and user weaknesses.

- The system shall store user performance metrics and track progress over time.
- The system shall allow users to edit, export, and download CVs.
- The system shall authenticate users securely and maintain sessions with proper access controls [7], [11], [14].
- The system shall log user actions and maintain history for analytics.
- The system shall allow administrators or mentors to view analytics (if applicable).

2.2 Non-Functional Requirements

2.2.1 Usability

- UI must be intuitive and easy to navigate.
- The mock interview interface must clearly present questions, timers, and recording indicators.
- CV editor may allow direct modification and preview after generation.

2.2.2 Reliability

- All user data must be backed up periodically.
- Mock interview sessions must handle network interruptions gracefully.

2.2.3 Performance

- STT transcription latency must remain below defined thresholds for interactive use [3], [16].
- Quiz evaluation must respond immediately after submission.

2.2.4 Supportability

- Codebase must follow modular architecture [2], [8], [9].
- Logging, monitoring, and debug tools should be implemented [7], [14].
- Documentation must be maintained for APIs, models, and development setup [8], [9].

2.2.5 Scalability

- Backend services must support increasing numbers of concurrent users [13], [14].
- STT and NLP inference modules should scale horizontally (e.g., container-based or cloud auto-scaling setups) [3], [14], [15].
- Database must support sharding or replication if needed.

---

3. Feasibility Discussions

3.1 Market & Competitive Analysis

There is a strong and growing market for AI-assisted job preparation tools, supported by recent labor market and skills reports [12], [13]. Competitors include platforms offering:

- AI-powered resume builders
- Interview practice tools
- Career coaching materials

However, most existing systems provide only one of these components. RecruitAssistant differentiates itself by offering:

- All-in-one hiring preparation pipeline
- Real-time STT mock interviews [3], [16]
- Personalized quizzes based on weaknesses [1], [4], [15]
- Automatic job-specific CV generation [1], [15]

This integrated approach provides a significant competitive advantage both in usability and practicality [12], [13].

3.2 Academic Analysis

From an academic perspective, the project intersects multiple fields:

- **Natural Language Processing:** CV generation, question generation, answer scoring, based on transformer-based models and modern NLP frameworks [1], [15].
- **Machine Learning:** Personalized quiz recommendation, performance analysis, and fairness-aware evaluation [4].
- **Speech Processing:** Live transcription for mock interviews utilizing self-supervised speech models [3], [16].
- **Software Engineering:** Scalable backend design, API modeling, user-centered interface design, following established standards and architectural principles [2], [8], [9], [10], [14].
- **Data Ethics:** Bias detection, fairness methods, responsible AI usage, and regulatory compliance [4], [5], [6], [7].

The system demonstrates real-world applicability of concepts learned in machine learning, algorithms, databases, and software engineering and aligns with contemporary practices in AI-powered systems design [1]–[4], [7], [8], [14].

4. Glossary

- **CV:** Curriculum Vitae (Resume)
- **NLP:** Natural Language Processing
- **STT:** Speech-to-Text
- **API:** Application Programming Interface
- **ML:** Machine Learning
- **Frontend:** User-facing application layer
- **Backend:** Server-side logic and data processing layer

5. References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). *Attention is All You Need.* Advances in Neural Information Processing Systems.

[2] Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures.* Doctoral Dissertation, University of California, Irvine.

[3] Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations.* NeurIPS.

[4] Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness and Machine Learning.* fairmlbook.org.

[5] European Union. (2016). *General Data Protection Regulation (GDPR).* Official Journal of the European Union.

[6] Türkiye Cumhuriyeti. (2016). *6698 Sayılı Kişisel Verilerin Korunması Kanunu (KVKK).* Resmî Gazete.

[7] International Organization for Standardization. (2022). *ISO/IEC 27001: Information Security Management Systems – Requirements.*

[8] IEEE. (1998). *IEEE Std 830-1998: Recommended Practice for Software Requirements Specifications.*

[9] IEEE. (2009). *IEEE Std 1016-2009: Recommended Practice for Software Design Documentation.*

[10] OMG – Object Management Group. (2017). *Unified Modeling Language (UML) Version 2.5.1 Specification.*

[11] OWASP Foundation. (2021). *OWASP Top Ten – The Ten Most Critical Web Application Security Risks.*

[12] LinkedIn. (2024). *Jobs on the Rise Report.* Available: https://www.linkedin.com

[13] Gartner Research. (2023). *AI Adoption and Market Forecast Report.*

[14] Amazon Web Services. (2022). *AWS Well-Architected Framework.* Available: https://aws.amazon.com/architecture/well-architected/

[15] Hugging Face. (2024). *Transformers Documentation.* Available: https://huggingface.co/docs/transformers

[16] OpenAI. (2023). *Whisper Speech-to-Text Model – Technical Report.* Available: https://openai.com/research/whisper

[17] Bruegge, B., & Dutoit, A. (2004). *Object-Oriented Software Engineering Using UML, Patterns, and Java.* 2nd ed., Prentice Hall. (Ders dokümanındaki [1] kaynağıyla uyumludur.)

[18] CATME Smarter Teamwork. Available: https://catme.org/

[19] TEAMMATES – Online Peer Feedback System. Available: https://teammatesv4.appspot.com/

[20] Wikipedia. *List of Collaborative Software.* Available: https://en.wikipedia.org/wiki/List_of_collaborative_software

[21] Wikipedia. *Comparison of Project Management Software.* Available: https://en.wikipedia.org/wiki/Comparison_of_project_management_software