Software Requirement Specification

Generate AI Mock Interview Platform

1. Patil Rajesh Shantaram

2. Patil Yogesh Vinod

3. Patil Nilesh Someshawar

4. Jadhav Shashikant Pravin

INDEX

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for the AI-Powered Mock Interview Platform. This system is designed to provide a platform that connects users, AI modules, evaluators, and administrators to facilitate efficient and realistic interview preparation experiences.

1.2 Product Scope

The AI-Powered Mock Interview Platform is a web-based application that allows users to simulate interviews, answer AI-generated questions, and receive performance feedback. The system also enables AI modules to evaluate responses, administrators to manage platform users and content, and provides an interactive environment for practicing interviews in a realistic and automated manner.

Key Features:

- **User Authentication**: Secure login and access for different user roles (Candidate, Evaluator/AI Module, Admin).
- **Interview Generation**: AI generates questions based on role, domain, and difficulty level.
- **Response Recording**: Users can answer via audio/video, which is recorded for evaluation.
- **Performance Analysis**: Real-time feedback and scoring based on answers and communication skills.
- **Interview Management**: Admins can view user performance, manage question sets, and oversee platform operations.
- **Feedback Dashboard:** Visual insights, correct answers, and improvement tips are shown post-interview.

1.3 Overview

This document provides a detailed description of the AI-Powered Mock Interview Platform's functionality, user roles, system architecture, and requirements. It outlines the project's objectives, features, and constraints.

2. Overall Description

2.1 Product Perspective

The AI-Powered Mock Interview Platform is a web-based application that provides a platform for realistic interview simulations. It integrates various components to facilitate interaction between candidates, AI modules, evaluators, and administrators.

2.2 Product Functions

- **User Registration & Authentication**: Secure login/logout system for all user roles.
- **Interview Management**: Generate mock interviews based on user-selected roles, domains, and difficulty.
- **Response Handling**: Record user answers via audio/video and store them securely
- **Evaluation Module**: AI analyzes responses, provides scores, and generates feedback.
- **Feedback Delivery**: Display performance reports, improvement areas, and correct answers post-interview.
-

2.3 User Classes and Characteristics

- **Candidates**: Users who can generate mock interviews, answer questions, receive feedback, and track progress.
- **AI Evaluators**: Users who manage candidate accounts, review platform activity, and configure system settings.
- **Admins**: Individuals who can view assigned orders, update delivery status, and confirm deliveries.
- **Evaluators (Optional):** Human reviewers who can assess responses and provide additional feedback if enabled.

2.4 Operating Environment

- Web-based application accessible via standard web browsers (e.g.,

Chrome, Firefox, Safari).

- Database Management System: PosgresSQL.

- Web/Application Server: Localhost.

- Operating System: Windows, macOS, or Linux.

2.5 Design and Implementation Constraints

- The system must be secure and protect user data.

- The system should be scalable to handle a large number of users and Interviews.

- The system should provide a user-friendly interface.

- The system should ensure high availability and reliability.

3. Specific Requirements

3.1 Functional Requirements

- Users (Candidates, AI Modules, and Admins) should be able to register, log in, and authenticate securely.

- Candidates should be able to select interview roles, start interviews, answer questions via audio/video, and receive performance feedback.

- The system should generate role-specific and difficulty-based interview questions using AI.

- The AI module should evaluate candidate responses, generate scores, highlight strengths and weaknesses, and provide improvement tips.

- Admins should be able to manage users, configure question banks, monitor interview logs, and adjust platform settings.

- The system should support secure access to premium features and paid evaluations.

- The system should provide real-time feedback and notifications after each interview round.

3.2 Non-Functional Requirements

- **Performance**: The system should handle multiple concurrent users and Interview with minimal delay.

- **Usability**: The user interface should be intuitive and easy to navigate for all user roles.

- **Security**: User data and transactions should be protected with secure authentication, authorization, and encryption.

- **Availability**: The system should have minimal downtime and high availability.

- **Scalability**: The system should be able to handle increasing numbers of users, Interviews.

- **Reliability**: The system should function consistently and accurately.

3.3 Hardware Interface

- Standard web browser.
- Internet connection.
- Server to host the application and database.

3.4 Software Interface

- Frontend: Next JS , React JS.
- Backend: Node JS.
- Database: PosgresSQL.
- Web Server: Localhost.

4. Selection of Relevant Process Model

4.1 Chosen Model: Agile Process Model

The Agile Software Development Model is suitable for this project. Agile allows for incremental development and continuous feedback, making it easier to refine features based on user needs.

4.2 Justification for Agile Model

Agile allows for iterative development, flexibility, and continuous feedback, which is important for incorporating changes and improvements throughout the development process.

Conclusion

The Agile Model will help ensure the development of a robust, user-friendly, and efficient AI Mock Interview Platform that meets the needs of all stakeholders.