

DESIGN AND DEVELOPMENT OF PERSONALIZED AI MOCK INTERVIEW

A

MAJOR PROJECT - II REPORT

Submitted in partial fulfillment of the requirements.

for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

GROUP NO. 36

Saurabh Vishwakarma	0187CS211154
Shubham Yadav	0187CS223D03
Vishal Sisodiya	0187CS211184
Satyam Rajak	0187CS211151

Under the guidance of

Prof. Deepti Jain

(Assistant Professor)



**Department of Computer Science & Engineering
Sagar Institute of Science & Technology (SISTec), Bhopal (M.P)**

**Approved by AICTE, New Delhi & Govt. of M.P.
Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P)**

June-2025

Sagar Institute of Science & Technology (SISTec), Bhopal (M.P)
Department of Computer Science & Engineering



CERTIFICATE

We hereby certify that the work which is being presented in the B.Tech. Major Project-II Report entitled **DESIGN AND DEVELOPMENT OF PERSONALIZED AI MOCK INTERVIEW**, in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology**, submitted to the Department of **Computer Science & Engineering**, Sagar Institute of Science & Technology (SISTec), Bhopal (M.P.) is an authentic record of our own work carried out during the period from Jan-2025 to Jun-2025 under the supervision of **Prof. Deepti Jain**.

The content presented in this project has not been submitted by us for the award of any other degree elsewhere.

Saurabh Vishwakarma

0187CS211154

Shubham Yadav

0187CS223D03

Vishal Sisodiya

0187CS211184

Satyam Rajak

0187CS211151

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Date:

Prof. Deepti Jain
Project Guide

Dr. Amit Kumar Mishra
HOD, CSE

Dr. D.K. Rajoriya
Principal

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr. D. K. Rajoriya, Principal, SISTec and Dr. Swati Saxena, Vice Principal SISTec** Gandhi Nagar, Bhopal for giving us an opportunity to undertake this project.

We also take this opportunity to express a deep sense of gratitude to **Dr. Amit Kumar Mishra, HOD, Department of Computer Science & Engineering** for his kindhearted support

We extend our sincere and heartfelt thanks to our guide, **Prof. Deepti Jain**, for providing us with the right guidance and advice at crucial junctures and for showing us the right way.

We are thankful to the **Project Coordinator, Prof. Deepti Jain**, who devoted her precious time in giving us the information about various aspects and gave support and guidance at every point of time.

We would like to thank all those people who helped us directly or indirectly to complete our project whenever we found ourself in any issue.

TABLE OF CONTENTS

TITLE	PAGE NO.
Abstract	i
List Of Figures	ii
List Of Abbreviations	iii
Chapter 1 Introduction	1
1.1 About Project	1
1.2 Project Objectives	1
1.3 Functionality	2
1.4 Interface	2
1.5 Design and Implementation Constraints	3
1.6 Assumptions and Dependencies	3
Chapter 2 Software and Hardware Requirements	4
Chapter 3 Problem Description	7
Chapter 4 Literature Survey	10
Chapter 5 Software Requirements Specification	12
5.1 Functional Requirements	12
5.2 Non-Functional Requirements	13
Chapter 6 Software Design	14
6.1 Use Case Diagram	15
6.2 Data Flow Design	16
Chapter 7 AI ML Module	17
Chapter 8 Coding	21
8.1 Overview	21
8.2 Home Page	22
Chapter 9 Results and Output Screens	39
9.1 Login & Sign Up Page	39
9.2 Landing Page & Footer Page	40
9.3 Dashboard Page & Add New Interview Page	41
9.4 Interview Page	42
9.5 Feedback Page & Previous Mock Interview Page	43
9.6 Database	44
Chapter 10 Conclusion and Future Work	45

10.1	Conclusion	45
10.2	Future Work	45
References		
Project Summary		
Appendix-1: Glossary of Terms		

ABSTRACT

The "**Personalised AI Mock Interview**" is an intelligent and interactive web application developed to assist job seekers in enhancing their interview preparedness through AI-driven mock interviews. Leveraging the capabilities of **Gemini AI**, the platform generates customized technical interview questions tailored to the user's selected job role, preferred technology stack, and level of experience. Upon logging in, users are navigated to a dashboard where they input their details through a guided form. Based on this input, the application dynamically generates **5-6 personalized interview questions**, which are presented sequentially with **camera accessibility** for a realistic interview setting. Users verbally respond to each question, and their responses are recorded and securely stored in the database. Once the session concludes, the Gemini AI model evaluates the recorded responses, providing comprehensive feedback that includes performance analysis, correct answers, improvement suggestions, and an **overall performance rating**. This personalized feedback loop enables users to pinpoint their strengths and weaknesses, polish their answering strategies, and boost confidence for real-world interviews. This project aims to bridge the gap between theoretical preparation and practical experience by offering an immersive and adaptive mock interview environment powered by artificial intelligence.

LIST OF ABBREVIATIONS

ACRONYM	FULL FORM
AI	Artificial Intelligence
UI	User Interface
IDE	Integrated Development Environment
API	Application Programming Interface
JWT	JSON Web Token
JUnit	Java Unit Testing
NLP	Natural Language Processing
DB	Data Base
OCR	Optical Character Recognition
TTS	Text To Speech
UX	User Experience

LIST OF FIGURES

FIGURE. NO.	TITLE	PAGE NO.
6.1	Use Case Diagram	15
6.2	Data Flow Diagram	16
9.1	Login & Sign Up	39
9.2	Landing Page & Footer Page	40
9.3	Dashboard & Add New Interview Page	41
9.4	Interview Page	42
9.5	Feedback& Previous Mock Interview Page	43
9.6	Database	44

Chapter 1

Introduction

CHAPTER-1

INTRODUCTION

1.1 ABOUT PROJECT

This project focuses on building a web application that assists users in preparing for job interviews by offering a personalized AI-powered mock interview experience. The application combines AI-generated interview questions with additional functionalities such as a performance dashboard, answer evaluation, and feedback system. The primary objective is to support users in practicing interview scenarios, analyzing their responses, and boosting their confidence through personalized insights and guidance.

Key functionalities include:

- 1.1.1 AI-Generated Interview Questions:** Allows users to receive personalized technical questions based on job role, selected technology, and years of experience, helping them prepare effectively for targeted interviews.
- 1.1.2 Camera-Based Response Recording:** Enables real-time recording of user answers via camera access to simulate a realistic interview environment.
- 1.1.3 AI-Based Answer Evaluation:** The Gemini AI model evaluates user responses, highlights strengths, and provides suggestions for improving weak areas.
- 1.1.4 Performance Dashboard:** Offers a centralized interface where users can view their session history, feedback, and ratings.
- 1.1.5 Fruit Nutrition Tracker:** Provides an aggregated score based on the user's responses, offering insight into readiness and improvement areas

1.2 PROJECT OBJECTIVE

The primary objectives of this project are to develop a comprehensive and user-friendly AI mock interview application, promote personalized skill assessment, and offer tools to assist in mastering technical interviews..

- 1.2.1 Personalized Question Generation:** Help users receive job-specific and skill-based questions using AI, ensuring targeted preparation.

1.2.2 Realistic Interview Practice: Provide users with a simulated interview environment using camera-based response capture for realistic practice.

1.2.3 Feedback And Suggestions: Offer constructive feedback and improvement suggestions after each session to help users polish their interview skills.

1.2.4 Performance Insights: Present a detailed performance analysis with metrics and personalized suggestions through an intuitive interface.

1.3 FUNCTIONALITY

The following point covers the functionality of the project:

1.3.1 AI Question Generation: Uses Gemini AI to generate context-based questions aligned with job roles, technologies, and experience level.

1.3.2 Camera-Based Recording: Captures user responses using the device's camera for natural and realistic answer delivery.

1.3.3 AI Evaluation Engine: Evaluates the answers using Gemini AI and returns personalized feedback and improvement tips.

1.3.4 Dashboard View: A centralized view showing previous mock interview sessions, question history, feedback, and ratings.

1.3.5 User Performance Analytics: Displays strengths and weaknesses in a visual format, helping users identify focus areas.

1.4 INTERFACE

The interface is designed to be user-friendly, intuitive, and visually appealing, providing real-time feedback and seamless navigation between features.

1.4.1 Interview Interface: A live question panel with camera view and answer capture functionality during the mock session.

1.4.2 Feedback Display: A clean and organized display of feedback for each question, showing model answers and improvement suggestions.

1.4.3 Dashboard Interface: Summary of user sessions, ratings, and progress over time in an easy-to-read layout.

1.4.4 Question Form Interface: Simple input fields for users to enter job role, technology, and experience to receive tailored questions.

1.4.5 Rating And Report View: Displays overall performance rating and session reports, helping users monitor improvement.

1.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

The design and implementation constraints of the projects are:

1.5.1 AI Evaluation Accuracy: Requires precise interpretation and analysis of spoken responses, depending on audio clarity and pronunciation.

1.5.2 AI Question Generation Dependency: Requires up-to-date and relevant question patterns for different domains and technologies

1.5.3 User Privacy And Data Security: The system must protect recorded videos, user profiles, and performance data, ensuring full privacy and compliance.

These constraints highlight factors to consider in ensuring reliability and accuracy.

1.6 ASSUMPTIONS AND DEPENDENCIES

The following are the assumptions and dependencies of the project:

1.6.1 Reliable AI Model Output: Assumes the Gemini AI model will accurately generate questions and evaluate answers based on relevant input.

1.6.2 Stable Camera And Audio Input: Relies on functional and clear audio/video hardware for accurate response capture.

Chapter 2

Software and

Hardware Requirements

CHAPTER-2

SOFTWARE & HARDWARE REQUIREMENTS

2.1 INTRODUCTION

This chapter outlines the essential software and hardware requirements for developing and deploying the AI-Powered Mock Interview application. These components are chosen to ensure high performance, accurate AI processing, real-time feedback, and an intuitive user experience. The software stack includes development tools, AI models, databases, and libraries necessary for building and deploying intelligent features. Hardware specifications are optimized for smooth processing of voice recognition, AI responses, and interview simulations across various devices.

2.2 SOFTWARE REQUIREMENTS

The software requirements for the AI-Powered Mock Interview application comprise robust development tools, AI integration frameworks, audio processing libraries, and reliable databases. These technologies enable seamless development and ensure secure, responsive, and intelligent interaction between users and the system.

2.2.1 Development Environment

Visual Studio: It is the official Integrated Development Environment (IDE) for Web development, offering a comprehensive platform for building, testing, and debugging applications. It provides tools for efficient coding, including a powerful code editor, real-time suggestions, and easy project management. Visual Studio also features a built-in live server, allowing developers to test the website across various devices.

Version Control System: Git, managed on platforms like GitHub, is used for source code versioning, team collaboration, and continuous integration. It ensures consistent code management and streamlined deployment workflows through features like pull requests and issue tracking.

2.2.2 Programming Languages

Javascript / Typescript: JavaScript is the primary language for frontend and backend development, while TypeScript enhances code reliability through static typing and better tooling support. TypeScript is especially beneficial for larger codebases and collaborative

2.2.3 Libraries And Frameworks

Gemini AI API: Integrates advanced natural language processing (NLP) models to generate contextually relevant interview questions, evaluate answers, and provide personalized feedback

Speech-To-Text API (Google/Whisper AI) : Converts user speech into text during interviews. This API ensures accurate transcription, enabling the AI model to evaluate spoken responses effectively.

Tailwind Css / Shadcn UI: Used to build a responsive, user-friendly, and modern UI with consistent design components for a smooth user experience.

Drizzle Orm + Neon Database: Drizzle ORM manages schema-safe database queries while Neon provides scalable Postgres hosting. Together, they ensure secure storage and efficient retrieval of resumes, questions, and feedback logs.

Jest & Cypress: Jest is used for unit testing, verifying functionality of individual components. Cypress enables end-to-end testing of the complete interview flow, including question generation, audio input, and feedback delivery.

2.3 HARDWARE REQUIREMENTS

The hardware requirements are designed to ensure smooth functioning of resource-intensive features like real-time AI feedback and audio transcription. While lightweight for end users, developers require more capable systems for local model testing and frontend/backend builds.

2.3.1 Smartphone Requirements

Operating System: Android OS version 7.0 (Nougat) or higher is recommended. This is necessary to support advanced functionalities such as image recognition.

processor: A processor with a minimum speed of 1.8 GHz and above for optimal perform.

RAM: At least 3GB RAM for mobile and 4GB for desktops/laptops to support smooth operation of real-time features and web interactions.

Microphone: Built-in or external microphone for capturing clear voice input during mock interviews.

Internet Connectivity: A stable internet connection is required for interacting with AI APIs

and saving data to the cloud. Offline access may be supported for stored interview history and user preferences.

Camera: A built-in rear camera with at least 8MP resolution is required to capture clear during mock interviews.

2.3.2 REQUIREMENTS AT DEVELOPER SIDE

Computer (PC OR MAC): Minimum Intel i5 (or equivalent), multi-core CPU. Required to run development tools like Next.js server, Node.js backend, and frontend builds.

RAM: 8GB minimum, 16GB recommended for running VSCode/Android Emulator, Node.js, browser instances, and AI testing environments simultaneously.

Storage Space: At least 50GB of free storage for packages, builds, databases, and project.

Testing Devices: Physical smartphones or tablets with built-in microphones are recommended for real-world testing of the voice input and UI responsiveness.

Display: Full HD (1080p) or higher recommended for optimal development and UI/UX.

Chapter 3

Problem Description

CHAPTER-3

PROBLEM DESCRIPTION

3.1 OVERVIEW

In today's competitive job market, job seekers face numerous challenges when preparing for technical interviews. The pressure to perform well, coupled with the uncertainty of the types of questions that may be asked, can create significant stress and hinder a candidate's ability to effectively showcase their skills. Additionally, many candidates struggle with self-evaluation, often finding it difficult to gauge their strengths and weaknesses in real-time. Traditional methods of interview preparation, such as studying from books, practicing with friends, or using generic question banks, often lack interactivity and fail to provide personalized feedback.

To address these challenges, our AI-powered Mock Interview application offers an intelligent, interactive, and structured approach to interview preparation. The application utilizes advanced AI models to generate personalized technical questions based on the user's resume and desired job role. Additionally, it incorporates speech recognition to simulate real interview scenarios, allowing users to respond verbally and receive real-time feedback on their answers. This ensures that candidates not only refine their technical knowledge but also improve their communication and problem-solving skills under pressure.

3.1.1 THE CHALLENGES OF TRADITIONAL INTERVIEW PREPARATION

Traditional interview preparation methods, such as reading books, using generic coding platforms, or attending mock interviews with peers, come with several limitations. One of the major drawbacks is the lack of real-time feedback. Many candidates struggle to identify their weak areas because they do not receive instant, personalized insights on their performance. This often leads to ineffective preparation, as candidates may focus on the wrong topics or fail to improve in key areas.

Another challenge is the limited availability of realistic interview scenarios. Practicing alone or with peers does not always simulate the high-pressure environment of a real job interview. Additionally, traditional methods fail to evaluate a candidate's spoken responses effectively, making it difficult for users to assess their communication skills, confidence, and ability to

articulate their thoughts clearly.

Furthermore, many job seekers face difficulty in finding high-quality, role-specific technical questions that align with their job aspirations. Most online resources provide generic question sets, which may not adequately prepare candidates for specific roles or industries. This lack of personalization makes it challenging to tailor preparation strategies to an individual's unique strengths and weaknesses.

These challenges highlight the need for an advanced interview preparation tool that provides real-time, AI-driven feedback, generates personalized technical questions, and simulates a realistic interview experience. Our AI-powered Mock Interview application addresses these gaps by offering an intelligent, interactive, and structured approach to interview readiness.

3.1.2 THE IMPORTANCE OF REAL-TIME FEEDBACK AND PERSONALIZED INTERVIEW SIMULATION

Effective interview preparation requires real-time feedback and tailored question generation to ensure that candidates can identify and improve upon their weaknesses. Traditional methods often lack immediacy, requiring users to wait for external feedback from mentors or peers. Without instant insights, candidates may continue making the same mistakes, reducing their chances of improvement.

Our AI-powered Mock Interview application leverages real-time feedback to provide users with instant evaluations of their technical responses, coding solutions, and verbal answers. By integrating advanced AI models, the system can assess answer accuracy, coding efficiency, and communication skills, offering constructive feedback that helps users refine their responses.

Moreover, personalization is a crucial aspect of effective interview preparation. Generic question banks do not cater to individual learning needs, leading to inefficient study plans. Our application solves this issue by analyzing the user's resume, job role, and skill set to generate targeted interview questions. This ensures that candidates practice questions relevant to their desired job positions, maximizing their chances of success.

3.2 KEY FEATURES AND FUNCTIONALITY

RESUME BASED QUESTION GENERATION:

The AI scans and analyzes the user's resume to generate personalized technical questions. Questions are tailored to the user's job role and experience, ensuring relevant and targeted practice.

REAL-TIME FEEDBACK AND PERFORMANCE ANALYSIS

Users receive instant feedback on their answers, highlighting strengths and areas for improvement.

AI-driven analysis evaluates coding efficiency, accuracy, and problem-solving approach.

VOICE-BASED INTERVIEW SIMULATION:

Integrated speech recognition allows users to practice verbal responses.

The AI assesses fluency, coherence, confidence, and articulation, providing constructive feedback.

INTERACTIVE DASHBOARD:

Users can track their progress, monitor performance trends, and review past interview.

Visual insights and analytics help candidates focus on weak areas and improve.

MOCK INTERVIEW SCHEDULING

Users can schedule AI-driven mock interviews based on their preferred job roles and skill.

The system provides a structured, time-bound interview experience to simulate real world.

MULTI-DOMAIN COVERAGE:

The platform supports a wide range of technical domains, including software development.

Chapter 4

Literature Survey

CHAPTER-4

LITERATURE SURVEY

The evolution of AI-powered web applications has significantly transformed how individuals prepare for job interviews, offering intelligent, interactive, and personalized practice opportunities. This section explores existing research, technologies, and methodologies in mock interview systems and AI-based preparation tools, identifying their strengths and limitations to highlight how the **AI-Powered Mock Interview Web Application** enhances and innovates upon them.[1]

The journey of digital interview preparation has evolved from static question banks and basic aptitude tests to dynamic, AI-integrated, and user-adaptive platforms. Early tools primarily offered generic multiple-choice questions or text-based mock interviews with limited personalization and feedback mechanisms. With advancements in Artificial Intelligence and Natural Language Processing (NLP), modern platforms have started incorporating intelligent features like speech recognition, real-time evaluation, and context-aware questioning. However, many of these platforms either lack interactivity, require premium access, or are not tailored to specific user profiles based on their resumes or skill sets.[2]

The **AI-Powered Mock Interview Web Application** addresses these limitations by providing a fully interactive, AI-enhanced environment that generates personalized technical questions from uploaded resumes. It evaluates user responses, offers real-time feedback, and simulates real interview scenarios—all within a cost-effective, web-based solution built using cutting-edge technologies like Next.js, Drizzle ORM, and Neon. Neon is a serverless and scalable cloud-hosted PostgreSQL database designed for modern development environments. It supports autoscaling, instant branching, and bottomless storage, making it an excellent choice for handling dynamic and growing datasets.[3]

In this project, Neon provides efficient storage and retrieval mechanisms for managing user resumes, question sets generated by AI, and corresponding evaluation metrics. Its serverless architecture reduces operational overhead and allows seamless integration with the Next.js backend via Drizzle ORM.

Neon is a serverless and scalable cloud-hosted PostgreSQL database designed for modern development environments. It supports autoscaling, instant branching, and bottomless storage, making it an excellent choice for handling dynamic and growing datasets. The core AI functionality of the application is powered by **Gemini AI** or similar Large Language Models (LLMs). These models analyze uploaded resumes to generate personalized technical interview questions.[4]

Leveraging the transformer architecture and extensive training data, LLMs offer high accuracy in language comprehension and generation. Gemini APIs provide an accessible interface for integrating this intelligence into the application, transforming static resumes into interactive and personalized interview sessions. Tailwind CSS is a utility-first CSS framework that allows developers to rapidly build modern, responsive user interfaces directly in HTML and JSX. It offers complete customization without writing custom CSS, supporting design consistency and rapid prototyping.^[5] For this application, Tailwind CSS is used to create an elegant and user-friendly dashboard for resume upload, question viewing, and interactive feedback.^[5]

To simulate realistic interview conditions, the system may incorporate browser-native or third-party **Speech-to-Text APIs** like the Web Speech API. These APIs convert spoken user responses into text in real-time. This textual data can then be analyzed by the LLM for evaluating fluency, correctness, and technical depth—enhancing the realism and effectiveness of the mock interview experience. Recent studies and industry trends highlight the growing role of AI in recruitment and learning tools. AI-powered mock interview systems provide scalable, personalized, and low-stress environments for job seekers to practice and improve. They reduce preparation anxiety, improve communication skills, and offer real-time insights, making them highly effective compared to traditional static preparation methods.^[6]

The AI-Powered Mock Interview Web Application combines modern web development technologies—Next.js, Drizzle ORM, Neon DB, Tailwind CSS—with AI advancements like Gemini LLMs and Speech APIs to deliver a dynamic, realistic, and scalable solution for job interview preparation. This integration of cutting-edge tools enables a highly interactive and educational experience for users, preparing them more effectively for real-world technical interviews. AI-powered mock interview systems provide scalable, personalized, and low-stress environments for job seekers to practice and improve. They reduce preparation anxiety, improve communication skills, and offer real-time insights, making them highly effective compared to traditional static preparation methods.^[7]

Chapter 5

Software Requirements

Specification

CHAPTER-5

SOFTWARE REQUIREMENTS SPECIFICATION

5.1 FUNCTIONAL REQUIREMENTS

User Registration And Authentication : The application must allow users to securely register and log in using email/password. Authentication should ensure secure access to personalized mock interview sessions, resume history, and AI-generated feedback.

Resume Upload And Parsing: Users should be able to upload their resumes in PDF format. The system must parse the resume and extract structured data such as skills, work experience, education, and projects to tailor interview questions using AI.

AI-Powered Question Generation: The system must generate personalized technical and behavioral questions based on the user's resume and desired job role using Gemini AI. It should support domains like Software Development, Data Science, and more.

Voice-Based Interaction: Users should be able to answer questions via voice. The app must capture, transcribe, and analyze voice responses in real time to simulate an actual interview experience. Text-based answering can also be supported as a fallback.

Interview History Dashboard : The application must provide a dashboard where users can view their previous mock interviews, AI-generated feedback, improvement suggestions, and track their progress over time.

Question Bank And Re-Attempt Option : Users should be able to re-attempt previously asked questions and access a curated bank of common interview questions relevant to their field and job role.

Multi-Domain And Difficulty Support: The application should allow users to select their preferred domain (e.g., Frontend, Backend, Data Analytics) and difficulty level (Easy, Medium, Hard) before starting a mock session.

5.2 NON-FUNCTIONAL REQUIREMENTS

Usability: The app must have a modern, clean, and user-friendly interface. Navigation should be intuitive, ensuring users can easily upload resumes, participate in mock interviews, and review.

Performance: The application should offer fast response times, with smooth transitions between screens and features. It should work efficiently even when handling high amounts of data, such as product details or health metrics over time.

Security: User data, including resumes and feedback, must be securely stored and encrypted. Secure APIs and authentication mechanisms must be in place to protect personal and sensitive information.

Scalability: The backend must support increasing numbers of users and interviews, ensuring stable performance during simultaneous access and high traffic.

Reliability: The application should have minimal downtime. It must gracefully handle interruptions during voice input or AI processing and ensure interview data is not lost.

Compatibility: The app should be compatible across a wide range of devices and operating systems, including Android and iOS, ensuring broad accessibility. It should work seamlessly on smartphones and tablets of various sizes and resolutions.

Maintenance And Support: Regular maintenance should be provided to fix bugs, introduce new features, and improve app performance based on user feedback. An in-app support system should be available for users to get help with any issues they encounter.

Backup And Data Recovery: The app should have data backup mechanisms in place, ensuring that user data can be recovered in case of system failures.

Chapter 6

Software Design

CHAPTER-6

SOFTWARE DESIGN

6.1 SOFTWARE DESIGN

Software Design is a critical phase in software development that involves planning and defining the system's architecture, components, user interfaces, and data structures. It acts as a blueprint for the development process, ensuring that the software meets both functional requirements (what the software should do) and non-functional requirements (how the software should perform). This stage focuses on creating a robust, scalable, and maintainable system, guiding the development team in implementing a solution that aligns with user needs, business goals, and technical constraints.

6.1.1 Actors

User: The user is the primary end-user of the mobile app, interacting with features like browsing content, tracking health metrics, and managing personal settings. They have access to personalized functionalities based on permissions.

Admin: The admin manages the backend of the system, including monitoring uploaded resumes, managing AI model configurations, updating question banks, resolving technical issues, and ensuring a smooth and secure user experience.

6.1.2 Use Cases

Upload Resume : The user uploads their resume in PDF format. The system parses the file and extracts key data points such as skills, work experience, education, and projects to feed into the AI model.

Generate AI-Based Interview Questions: Based on the uploaded resume and selected domain, the system uses Gemini AI to generate customized technical and behavioral questions relevant to the user's profile.

Mock Interview (Voice/Text Interaction): Users answer questions either via voice or text. The system captures and analyzes these responses in real-time, mimicking an actual interview experience.

Real-Time Feedback and Scoring: After the interview session, the AI provides personalized feedback on communication skills, technical knowledge, confidence level, and content structure, helping users understand areas of improvement

Dashboard Overview: The home/dashboard screen displays recent activity, mock interview stats, performance trends, upcoming recommendations, and easy access to past feedback and reports.

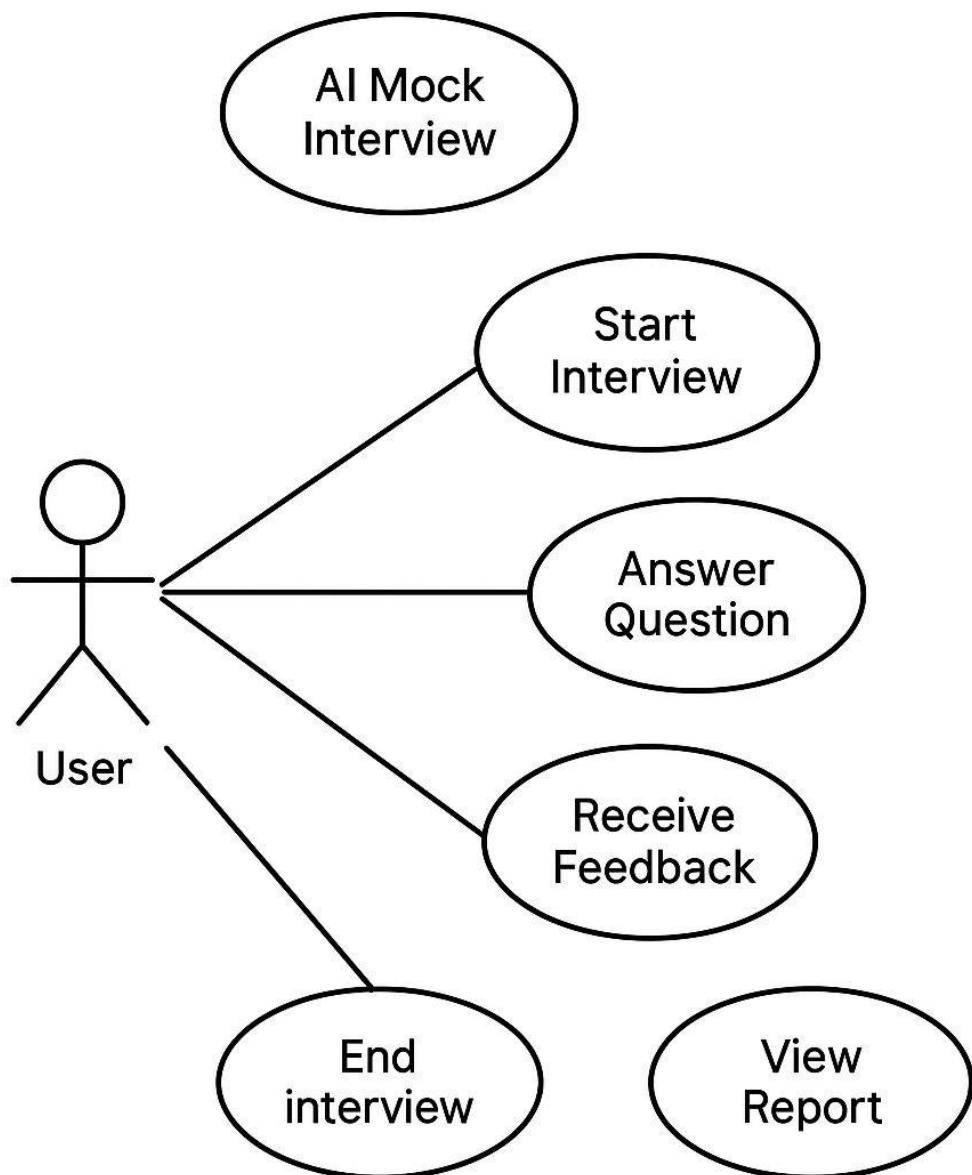


Figure 6.1: Use Case Diagram

6.2 DATA FLOW DIAGRAM

A **Data Flow Diagram (DFD)** is a graphical representation of the flow of data within a system. It illustrates how input data is processed, stored, and transformed into output data. DFDs help visualize the movement and transformation of information within a system, highlighting how different components interact with each other.

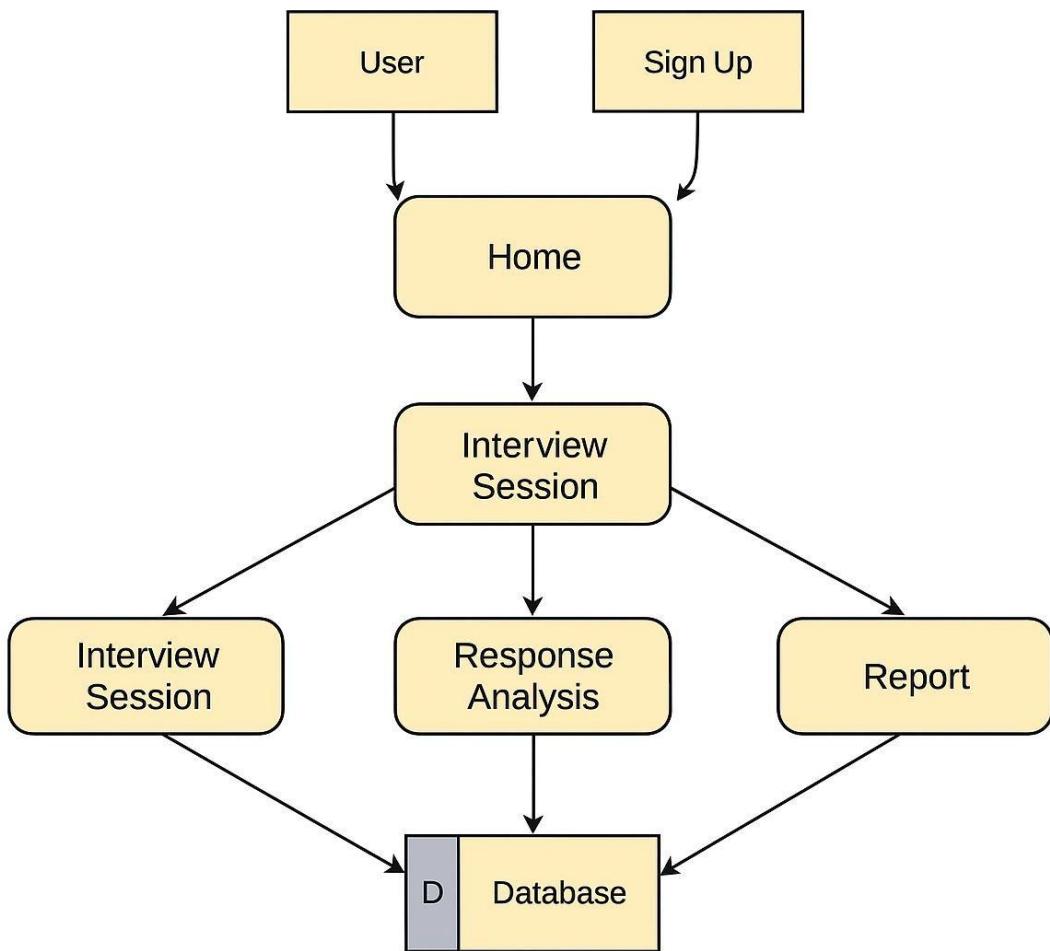


Figure 6.2: Dataflow Diagram

Chapter 7

AJML Module

Chapter-7

AI ML Module

7.1 INTRODUCTION

In the context of web development for the AI Mock Interview App, a module refers to a self-contained part of the project that handles specific functionalities such as resume parsing, question generation, audio recording, or feedback delivery. Using modules helps organize code, streamline development, and enable independent testing and deployment of key features. This chapter dives into the role of web modules in building scalable, maintainable, and feature-rich AI-based mock interview systems.

7.1.1 What is an AI ML module?

An AI ML module is a collection of source code, assets, and configurations that accomplish a distinct purpose in the app. For the AI Mock Interview App, modules help separate concerns such as UI handling, speech processing, AI integration, and user management. Each module can be developed, built, and tested individually, which improves collaboration and code maintenance. Modern frameworks like Next.js and React support modularization through folder structures, shared components, and service layers.

7.1.2 Types Of AI ML Modules

1. Core App Module

This is the primary module for your web application.

Contains the application logic, routes, and global components file.

Handles routing, layout, and shared state.

2. Library Module

Houses reusable components like AI question generators, audio processors, and utility functions.

Outputs an Gemini Archive (.AAR) file.

Ideal for shared logic like NLP-based question generation and resume parsing.

3. Dynamic Feature Module

Enables code-splitting and on-demand loading of features using dynamic imports.

Reduces initial load time for better performance.

Reduces the initial download size of the app.

Ideal for features that are not required on initial load (e.g., feedback analysis).

4. JavaScript/Next.js Module

Contains backend logic like Gemini API calls, answer evaluation algorithms, and data formatting utilities.

Independent of Website UI framework.

Outputs a NextJS Archive (.JAR) file.

7.1.3 Structure Of A Web Module

A Web module contains the following key components:

/src/: Contains all React or Next.js source code files.

src/main/res: Stores public assets like image and static files and animations.

/components/: Reusable UI components.

/pages/ or /routes/: Defines application routing.

/api/: Contains backend logic (in Next.js or Express).

/utils/: Utility functions and shared services.

package.json: Manages dependencies and scripts.

next.config.js or vite.config.js: Configuration for builds and plugins.

7.1.4 Advantages Of Using Modules

Code Reusability: Library modules can be reused in multiple projects.

Scalability: Modularization makes it easier to scale projects by separating concerns.

Parallel Development: Teams can work on different modules simultaneously without conflicts.

Faster Build Times: Gradle builds smaller, independent modules instead of project.

Testing and Debugging: Modules can be individually tested and debugged.

7.1.5 Creating A Module In Web Project

1. Steps to Create a Module:

Open the AI Mock Interview project.

Navigate to **File > New > Module**.

Choose the appropriate module type (e.g., Android Library).

Set the module name (e.g., speech_evaluator, resume_parser).

Click **Finish** to create the module.

2. Setting Module Dependencies:

In your app/build.grade : implementation project(':resume_parser')

7.1.6 Best Practices For Modularization

Feature-Based Modules: Separate modules for Resume Upload, Question Generation, and Feedback.

Domain-Specific Modules: Divide logic and data processing for clean code structure.

Avoid Tight Coupling: Minimize dependencies between modules to maintain flexibility.

Use Dependency Injection: Use Hilt to manage shared services like API clients or data sources.

7.1.7 Modularizing A Project

Imagine a shopping app with features like user authentication, product browsing, and cart management. This can be modularized as follows:

app Module: Handles app-level configurations and navigation.

auth Module: Manages login, registration, and user sessions.

resume_parser : Upload and analyze resumes.

question_generator : Uses AI (Gemini API) to generate personalized questions.

speech_evaluator : Records and analyzes voice answers.

feedback_engine: Provides real-time interview feedback.

network : API calls and Firebase interactions.

This modular approach enables independent development and testing of each feature.

7.1.8 Testing And Debugging Modules

Unit Testing: Test resume parsing, AI-generated questions, and scoring algorithms.

Instrumentation Testing: Validate modules interacting with Android components (e.g., microphone, camera).

Gradle Tasks: Use commands like `./gradlew :speech_evaluator:test` for specific builds.

7.1.9 Challenges In Modularization

Setup Complexity: Requires careful configuration of inter-module dependencies.

Build Time: Over-modularization may lead to overhead during builds.

Communication: Sharing data (like user responses) across modules can be tricky without proper architecture (e.g., using clean architecture or shared ViewModels).

Chapter 8

Coding

CHAPTER-8

CODING

8.1 OVERVIEW

The coding phase focuses on the development process of the AI-Powered Mock Interview Web Application, built using **Next.js**, **JavaScript**, **Drizzle ORM**, **Neon Database**, and **Gemini AI**. This chapter outlines how each technology was leveraged to implement core functionalities and ensure a smooth, efficient, and scalable development process.

Next.js served as the primary framework, providing a powerful React-based environment with server-side rendering, routing, and performance optimization. It enabled the development of a modern, responsive, and SEO-friendly user interface.

JavaScript was used throughout the application to implement dynamic behavior, logic control, and frontend-backend interactions. Tailwind CSS was used for building a sleek and responsive UI.

Drizzle ORM facilitated structured interaction with the **Neon database**, allowing developers to manage schema definitions, perform CRUD operations, and maintain type safety in a developer-friendly manner.

The **Gemini API** (via Google AI) was integrated to automatically generate personalized technical questions from uploaded resumes. This AI component plays a key role in simulating real interview environments.

8.1.1 Future Scope Of Coding

The future of coding in the context of AI-powered platforms is vast and rapidly evolving. Technologies like **natural language processing (NLP)**, **voice recognition**, and **machine learning models** will continue to shape interactive systems.

8.2 HOME PAGE

```

"use client";
import { Button } from "@/components/ui/button";
import { db } from "@/utils/db";
import { MockInterview } from "@/utils/schema";
import { eq } from "drizzle-orm";
import { Lightbulb, WebcamIcon } from "lucide-react";
import Link from "next/link";
import React, { useEffect, useState } from "react";
import Webcam from "react-webcam";

function Interview({ params }) {
  const [interviewData, setInterviewData] = useState();
  const [webCamEnabled, setWebCamEnabled] = useState(false);
  useEffect(() => {
    // console.log(params.interviewId);
    GetInterviewDetails();
  }, []);
}

/**
 * Used to Get Interview Details by MockId/Interview Id
 */
const GetInterviewDetails = async () => {
  const result = await db
    .select()
    .from(MockInterview)
    .where(eq(MockInterview.mockId, params.interviewId));

  // console.log(result);
  setInterviewData(result[0]);
};

return (
  <div className="my-10">
    <h2 className="font-bold text-2xl">Let's Get Started</h2>
    <div className="grid grid-cols-1 md:grid-cols-2 gap-10">
      <div className="flex flex-col my-5 gap-5">
        <div className="flex flex-col gap-5 p-5 rounded-lg border">
          <h2 className="text-lg">
            <strong>Job Position/Role : &nbsp;</strong>
            {interviewData?.jobPosition}
          </h2>
          <h2 className="text-lg">
            <strong>Job Description/Tech Stack: &nbsp;</strong>
            {interviewData?.jobDesc}{" "}
          </h2>
          <h2 className="text-lg">
            <strong>Years of Experience: &nbsp;</strong>{" "}
            {interviewData?.jobExperience}
          </h2>
        </div>
      </div>
    <div className="p-5 rounded-lg border-yellow-300 bg-yellow-100">

```

```

<h2 className="flex gap-2 items-center text-yellow-700"><Lightbulb/><strong>Information</strong></h2>
<h2 className="mt-3 ">{process.env.NEXT_PUBLIC_INFORMATION}</h2>
</div>
</div>

<div className="flex flex-col items-center justify-center">
  {webCamEnabled ? (
    <Webcam
      onUserMedia={() => setWebCamEnabled(true)}
      onUserMediaError={() => setWebCamEnabled(false)}
      mirrored={true}
      style={{ height: 300, width: 300 }}
    />
  ) : (
    <>
      <WebcamIcon className="h-72 w-full p-20 my-7 bg-secondary border rounded-lg" />
      <Button onClick={() => setWebCamEnabled(true)} variant="ghost" className="border">
        Enable Web Cam and Microphone
      </Button>
    </>
  )}
</div>
</div>
<div className="flex justify-end items-end">
  <Link href={`/dashboard/interview/${params.interviewId + '/start'}`}>
    <Button>Start Interview</Button>
  </Link>
</div>
</div>
);
}

```

export default Interview;

```

"use client"
import { UserButton } from '@clerk/nextjs'
import Image from 'next/image'
import { usePathname } from 'next/navigation'
import React, { useEffect } from 'react'

function Header() {

  const path = usePathname();
  useEffect(()=>{
    console.log(path);

  })

  return (
    <div className='flex p-4 items-center justify-between bg-secondary shadow-sm'>
      <Image src={'/interview_logo.svg'} width={90} height={90} alt='logo' />

```

```

<ul className='hidden md:flex gap-6'>
    <li className={`hover:text-primary hover:font-bold translate-all cursor-pointer
    ${path=='/dashboard' && 'text-primary font-bold'}`}>Dashboard</li>

    <li className={`hover:text-primary hover:font-bold translate-all cursor-pointer
    ${path=='/dashboard/question' && 'text-primary font-bold'}`}>Question</li>

    <li className={`hover:text-primary hover:font-bold translate-all cursor-pointer
    ${path=='/dashboard/upgrade' && 'text-primary font-bold'}`}>Feedback</li>

    <li className={`hover:text-primary hover:font-bold translate-all cursor-pointer
    ${path=='/dashboard/how' && 'text-primary font-bold'}`}>About Us</li>
</ul>
<UserButton/>
</div>
)
}

```

export default Header

```

"use client";
import React, { useState } from "react";
import {
    Dialog,
    DialogContent,
    DialogDescription,
    DialogHeader,
    DialogTitle,
    DialogTrigger,
} from "@/components/ui/dialog";
import { Button } from "@/components/ui/button";
import { Input } from "@/components/ui/input";
import { Textarea } from "@/components/ui/textarea";
import { chatSession } from "@/utils/GeminiAIModal";
import { LoaderCircle } from "lucide-react";
import { db } from "@/utils/db";
import { MockInterview } from "@/utils/schema";
import { v4 as uuidv4 } from "uuid";
import { useUser } from "@clerk/nextjs";
import moment from "moment";
import { useRouter } from "next/navigation";
function AddNewInterview() {
    const [openDialog, setOpenDialog] = useState(false);
    const [jobPosition, setJobPosition] = useState();
    const [jobDesc, setJobDesc] = useState();
    const [jobExperience, setJobExperience] = useState();
    const [loading, setLoading] = useState(false);
    const [jsonResponse, setJsonResponse] = useState([]);
    const { user } = useUser();
    const router = useRouter();

    const onSubmit = async (e) => {
        setLoading(true);
        e.preventDefault();

```

```

console.log(jobPosition, jobDesc, jobExperience);

const InputPrompt =
  "Job Position: " +
  jobPosition +
  ", Job Description: " +
  jobDesc +
  ", Years of Experience: " +
  jobExperience +
  ", Depends on this information please give me " +
process.env.NEXT_PUBLIC_INTERVIEW_QUESTION_COUNT +
  " Interview question with Answered in Json Format, Give Question and Answered as field in JSON";

const result = await chatSession.sendMessage(InputPrompt);
const MockJsonResp = result.response
  .text()
  .replace("```json", "")
  .replace("```", "");
console.log(JSON.parse(MockJsonResp));

setJsonResponse(MockJsonResp);

if (MockJsonResp) {
  const resp = await db
    .insert(MockInterview)
    .values({
      mockId: uuidv4(),
      jsonMockResp: MockJsonResp,
      jobPosition: jobPosition,
      jobDesc: jobDesc,
      jobExperience: jobExperience,
      createdBy: user?.primaryEmailAddress?.emailAddress,
      createdAt: moment().format("DD-MM-YYYY"),
    })
    .returning({ mockId: MockInterview.mockId });
}

console.log("Inserted Id:", resp);
if (resp) {
  setOpenDialog(false);
  router.push("./dashboard/interview/" + resp[0]?.mockId);
}
} else {
  console.log("Error");
}
 setLoading(false);
};

return (
<div>
  <div
    onClick={() => setOpenDialog(true)}
    className="p-10 border rounded-lg bg-secondary hover:scale-105 hover:shadow-md cursor-pointer transition-all"
  >
    <h2 className="text-lg text-center">+ Add New</h2>
  </div>

```

```

<Dialog open={openDialog}>
  <DialogContent className="max-w-2xl">
    <DialogHeader>
      <DialogTitle className="text-2xl">
        Tell us more about your job interviewing
      </DialogTitle>
      <DialogDescription>
        <form action="" onSubmit={onSubmit}>
          <div>
            <h2>
              Add Details about your position/role, Job description and
              years of experience
            </h2>

            <div className="mt-7 my-3">
              <label>Job Role/Job Position</label>
              <Input
                placeholder="Ex. Full Stack Developer"
                required
                onChange={(event) => setJobPosition(event.target.value)}
              />
            </div>

            <div className="my-3">
              <label>Job Description/ Tech Stack (In Short) </label>
              <Textarea
                placeholder="Ex. React, Angular, NodeJS, MySQL"
                required
                onChange={(event) => setJobDesc(event.target.value)}
              />
            </div>

            <div className="my-3">
              <label>Years of experience </label>
              <Input
                type="number"
                placeholder="Ex. 2"
                max="50"
                required
                onChange={(event) => setJobExperience(event.target.value)}
              />
            </div>
          </div>

          <div className="flex gap-5 justify-end">
            <Button
              type="button"
              variant="ghost"
              onClick={() => setOpenDialog(false)}
            >
              Cancel
            </Button>
            <Button type="submit" disabled={loading}>
              {loading ? (

```

```

    </>
    <LoaderCircle className="animate-spin" />
    'Generating From AI'
  </>
) : (
  "Start Interview"
)
</Button>
</div>
</form>
</DialogDescription>
</DialogHeader>
</DialogContent>
</Dialog>
</div>
);
}

export default AddNewInterview;

import { Button } from '@/components/ui/button';
// import Link from 'next/link'
import { useRouter } from 'next/navigation'
import React from 'react'

function InterviewItemCard({interview}) {
  const router = useRouter();
  const onStart = () =>{
    router.push('/dashboard/interview/'+interview?.mockId)
  }
  const onFeedback = () =>{
    router.push('/dashboard/interview/'+interview?.mockId+"/feedback")
  }
  return (
    <div className='border shadow-sm rounded-lg p-3'>
      <h2 className='font-bold text-primary'>{interview?.jobPosition}</h2>
      <h2 className='text-sm text-gray-600'>{interview.jobExperience} Years of Experience</h2>
      <h2 className='text-xs text-gray-400'>Created At: {interview.createdAt}</h2>
      <div className='flex justify-between mt-2 gap-5'>
        <Button size="sm" variant="outline" className="w-full"
          onClick={onFeedback}>Feedback</Button>
        <Button size="sm" className='w-full' onClick={onStart}>Start</Button>
      </div>
    </div>
  )
}

export default InterviewItemCard

```

```

"use client"
import { db } from '@/utils/db';
import { MockInterview } from '@/utils/schema';
import { useUser } from '@clerk/nextjs'
import { desc, eq } from 'drizzle-orm';

```

```

import React, { useEffect, useState } from 'react'
import InterviewItemCard from './InterviewItemCard';

function InterviewList() {
  const {user} = useUser();
  const [interviewList, setInterviewList] = useState([]);
  useEffect(()=>{
    user && GetInterviewList();
  },[user])
  const GetInterviewList=async()=>{
    const result = await db.select()
      .from(MockInterview)
      .where(eq(MockInterview.createdBy, user?.primaryEmailAddress?.emailAddress))
      .orderBy(desc(MockInterview.id))

    // console.log(result);
    setInterviewList(result);
  }
  return (
    <div>
      <h2 className='font-medium text-xl mt-4'>Previous Mock Interview</h2>
      <div className='grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-5 my-3'>
        {interviewList && interviewList.map((interview, index)=>(
          <InterviewItemCard
            interview={interview}
            key={index}/>
        )))
      </div>
    </div>
  )
}

export default InterviewList

```

//FEEDBACK PAGE

```

"use client";

import { db } from "@/utils/db";
import { UserAnswer } from "@/utils/schema";
import { eq } from "drizzle-orm";
import React, { useEffect, useState } from "react";
import { Collapsible, CollapsibleContent, CollapsibleTrigger, } from "@/components/ui/collapsible";
import { ChevronsUpDown } from "lucide-react";
import { Button } from "@/components/ui/button";
import { useRouter } from "next/navigation";

function Feedback({ params }) {
  const interviewId = React.use(params).interviewId; // Unwrap params using React.use()
  const router = useRouter();

```

```

const [feedbackList, setFeedbackList] = useState([]);
const [overallRating, setOverallRating] = useState(0);
useEffect(() => {
  GetFeedback();
}, []);

```

```

const GetFeedback = async () => {
  try {
    const result = await db
      .select()
      .from(UserAnswer)
      .where(eq(UserAnswer.mockIdRef, interviewId))
      .orderBy(UserAnswer.id);
    console.log(result);
    setFeedbackList(result);
    calculateOverallRating(result);
  } catch (error) {
    console.error("Error fetching feedback:", error);
  }
};

```

```

const calculateOverallRating = (feedbacks) => {
  if (feedbacks.length > 0) {
    const totalRating = feedbacks.reduce((acc, item) => acc + (parseFloat(item.rating) || 0), 0);
    const averageRating = totalRating / feedbacks.length;
    const scaledRating = Math.max(Math.max((averageRating / 5) * 10, 1), 10); // Ensuring rating between 1 and 10
    setOverallRating(scaledRating.toFixed(1)); // Keeping one decimal place
  }
};

```

```

return (
<div className="p-10">
  {feedbackList?.length === 0 ?
    <h2 className="font-bold text-xl text-gray-500">No Interview Feedback Record Found</h2> :
    <h2 className="text-primary text-lg my-3">
      Your overall interview rating: <strong>{overallRating}</strong>
    </h2>
    <h2 className="text-sm text-gray-500">
      Find below interview questions with correct answers, your answers, and
      feedback for improvement.
    </h2>
  }
  {feedbackList &&
    feedbackList.map((item, index) => (
      <Collapsible key={index} className="mt-7">
        <CollapsibleTrigger className="p-2 bg-secondary rounded-lg my-2 text-left flex justify-between gap-7 w-full">
          {item.question} <ChevronsUpDown className="h-4" />
        </CollapsibleTrigger>
        <CollapsibleContent>
          <div className="flex flex-col gap-2">
            <h2 className="text-red-500 p-2 border rounded-lg">
              <strong>Rating:</strong> {item.rating}
            </h2>

```

```

<h2 className="p-2 border rounded-lg bg-red-50 text-sm text-red-900"><strong>Your Answer:</strong>{item.userAns}</h2>
<h2 className="p-2 border rounded-lg bg-green-50 text-sm text-green-900"><strong>Correct Answer:</strong>{item.correctAns}</h2>
<h2 className="p-2 border rounded-lg bg-blue-50 text-sm text-primary"><strong>Feedback:</strong>{item.feedback}</h2>
</div>
</CollapsibleContent>
</Collapsible>

))}

</>

<Button onClick={()=>router.replace('/dashboard')}>
    Go Home
</Button>
</div>
);
}

```

export default Feedback;

```

"use client";
import React from "react";
import { ArrowRight, CheckCircle2, Globe2, Users2, Zap } from "lucide-react";
import Header from "./dashboard/_components/Header";
import Link from "next/link";
import { useRouter } from "next/navigation";
import {
    SignedIn,
    SignedOut,
    SignInButton,
    SignUpButton,
    UserButton,
} from "@clerk/nextjs";

function Home() {
    const router = useRouter();
    return (
        <div className="min-h-screen bg-white">
            /* Hero Section */
            <header className="bg-gradient-to-r from-indigo-600 to-violet-600 text-white">
                <nav className="container mx-auto px-6 py-4 flex items-center justify-between">
                    <div className="flex items-center space-x-2 text-xl font-bold">
                        <Globe2 className="h-8 w-8" />
                        <span>AI Interview</span>
                    </div>
            
```

{/* User actions */}

<SignedIn>

<UserButton />

</SignedIn>

<SignedOut>

```

<div className="hidden md:flex items-center space-x-5">
  <SignInButton>
    <button className="bg-white text-indigo-600 px-6 py-2 rounded-full font-medium hover:bg-indigo-50 transition-colors">
      Sign In
    </button>
  </SignInButton>
  <SignUpButton>
    <button variant={'outline'} className="bg-white text-indigo-600 px-6 py-2 rounded-full font-medium hover:bg-indigo-50 transition-colors">
      Sign Up
    </button>
  </SignUpButton>
</div>
</SignedOut>
</nav>

<div className="container mx-auto px-6 py-20">
  <div className="max-w-3xl">
    <h1 className="text-5xl font-bold mb-6">
      Master Your Technical Interviews with AI
    </h1>
    <p className="text-xl mb-8 text-indigo-100">
      Practice with our AI interviewer and get real-time feedback.
      Perfect your responses and land your dream job in tech.
    </p>
    <div className="flex flex-col sm:flex-row gap-4">
      <button onClick={()=>router.replace('/dashboard')} className="bg-white text-indigo-600 px-8 py-4 rounded-full font-medium hover:bg-indigo-50 transition-colors flex items-center justify-center">
        Start Practicing Now
        <ArrowRight className="ml-2 h-5 w-5" />
      </button>
      <button className="border-2 border-white text-white px-8 py-4 rounded-full font-medium hover:bg-white/10 transition-colors">
        Watch Demo
      </button>
    </div>
  </div>
</div>
</header>
/* <Header/> */
/* Features Section */
<section className="py-20 bg-gray-50">
  <div className="container mx-auto px-6">
    <h2 className="text-3xl font-bold text-center mb-16 text-gray-800">
      Why Choose AI Interview?
    </h2>
    <div className="grid md:grid-cols-3 gap-12">
      <FeatureCard
        icon={<Zap className="h-8 w-8 text-indigo-600" />}
        title="AI-Powered Practice"
        description="Get realistic interview experience with our advanced AI interviewer that adapts to your responses."
      />
      <FeatureCard>
    </div>
  </div>
</section>

```

```

icon={<CheckCircle2 className="h-8 w-8 text-indigo-600" />}
title="Real-time Feedback"
description="Receive instant feedback on your answers, communication skills, and technical accuracy."
/>
<FeatureCard
  icon={<Users2 className="h-8 w-8 text-indigo-600" />}
  title="Industry Standards"
  description="Practice with questions based on real interviews from top tech companies."
/>
</div>
</div>
</section>

/* Social Proof Section */
<section className="py-20">
  <div className="container mx-auto px-6">
    <div className="text-center mb-12">
      <h2 className="text-3xl font-bold mb-4 text-gray-800">
        Trusted by Developers Worldwide
      </h2>
      <p className="text-gray-600 max-w-2xl mx-auto">
        Join thousands of developers who have improved their interview
        skills and landed jobs at top tech companies.
      </p>
    </div>
    <div className="grid md:grid-cols-3 gap-8">
      <TestimonialCard
        image="https://images.unsplash.com/photo-1494790108377-be9c29b29330?auto=format&fit=crop&w=150"
        name="Sarah Chen"
        role="Software Engineer at Google"
        quote="GiveInterview helped me prepare effectively for my technical interviews. The AI feedback
        was invaluable."
      />
      <TestimonialCard
        image="https://images.unsplash.com/photo-150700321169-0a1dd7228f2d?auto=format&fit=crop&w=150"
        name="Michael Rodriguez"
        role="Senior Developer at Microsoft"
        quote="The realistic interview scenarios and detailed feedback helped me gain confidence in my
        abilities."
      />
      <TestimonialCard
        image="https://images.unsplash.com/photo-1438761681033-6461ffad8d80?auto=format&fit=crop&w=150"
        name="Emily Johnson"
        role="Full Stack Developer at Meta"
        quote="I credit my successful career transition to the practice I got through GiveInterview."
      />
    </div>
  </div>
</section>

/* CTA Section */

```

```

<section className="bg-indigo-600 py-20">
  <div className="container mx-auto px-6 text-center">
    <h2 className="text-3xl font-bold mb-8 text-white">
      Ready to Ace Your Next Technical Interview?
    </h2>
    <button className="bg-white text-indigo-600 px-8 py-4 rounded-full font-medium hover:bg-indigo-50 transition-colors inline-flex items-center">
      Get Started Free
      <ArrowRight className="ml-2 h-5 w-5" />
    </button>
  </div>
</section>

/* Footer */

<footer className="bg-gray-900 text-gray-300 py-12">
  <div className="container mx-auto px-6">
    <div className="grid md:grid-cols-4 gap-8">
      <div>
        <div className="flex items-center space-x-2 text-xl font-bold text-white mb-4">
          <Globe2 className="h-8 w-8" />
          <span>AI Interview</span>
        </div>
        <p className="text-sm">
          Empowering developers to succeed in technical interviews through
          AI-powered practice.
        </p>
      </div>
      <div>
        <h3 className="font-semibold text-white mb-4">Product</h3>
        <ul className="space-y-2">
          <li>
            <a href="#" className="hover:text-white">
              Features
            </a>
          </li>
          <li>
            <a href="#" className="hover:text-white">
              Pricing
            </a>
          </li>
          <li>
            <a href="#" className="hover:text-white">
              Enterprise
            </a>
          </li>
        </ul>
      </div>
      <div>
        <h3 className="font-semibold text-white mb-4">Resources</h3>
        <ul className="space-y-2">
          <li>
            <a href="#" className="hover:text-white">
              Blog
            </a>
          </li>
        </ul>
      </div>
    </div>
  </div>
</footer>

```

```

<li>
  <a href="#" className="hover:text-white">
    Documentation
  </a>
</li>
<li>
  <a href="#" className="hover:text-white">
    Community
  </a>
</li>
</ul>
</div>
<div>
  <h3 className="font-semibold text-white mb-4">Company</h3>
  <ul className="space-y-2">
    <li>
      <a href="#" className="hover:text-white">
        About
      </a>
    </li>
    <li>
      <a href="#" className="hover:text-white">
        Careers
      </a>
    </li>
    <li>
      <a href="#" className="hover:text-white">
        Contact
      </a>
    </li>
  </ul>
</div>
</div>
<div className="border-t border-gray-800 mt-12 pt-8 text-sm text-center">
  <p>&copy; 2024 AI Interview. All rights reserved.</p>
</div>
</div>
</footer>
</div>
);
}

function FeatureCard({ icon, title, description }) {
  return (
    <div className="bg-white p-8 rounded-xl shadow-sm hover:shadow-md transition-shadow">
      <div className="mb-4">{icon}</div>
      <h3 className="text-xl font-semibold mb-3 text-gray-800">{title}</h3>
      <p className="text-gray-600">{description}</p>
    </div>
  );
}

function TestimonialCard({ image, name, role, quote }) {
  return (
    <div className="bg-white p-8 rounded-xl shadow-sm hover:shadow-md transition-shadow">

```

```

<div className="flex items-center mb-4">
  <img src={image} alt={name} className="w-12 h-12 rounded-full mr-4" />
  <div>
    <h4 className="font-semibold text-gray-800">{name}</h4>
    <p className="text-sm text-gray-600">{role}</p>
  </div>
</div>
<p className="text-gray-700 italic">"{quote}"</p>
</div>
);
}

```

```
export default Home;
```

```

import { serial, text, varchar } from "drizzle-orm/pg-core";
import { pgTable } from "drizzle-orm/pg-core";

```

```

export const MockInterview = pgTable("mockInterview", {
  id: serial("id").primaryKey(),
  jsonMockResp: text("jsonMockResp").notNull(),
  jobPosition: varchar("jobPosition").notNull(),
  jobDesc: varchar("jobDesc").notNull(),
  jobExperience: varchar("jobExperience").notNull(),
  createdBy: varchar("createdBy").notNull(),
  createdAt: varchar("createdAt"),
  mockId: varchar("mockId").notNull(),
});

```

```

export const UserAnswer = pgTable('userAnswer', {
  id: serial("id").primaryKey(),
  mockIdRef: varchar("mockId").notNull(),
  question: varchar('question'),
  correctAns: text('correctAns'),
  userAns: text('userAns'),
  feedback: text("feedback"),
  rating: varchar("rating"),
  userEmail: varchar("userEmail"),
  createdAt: varchar("createdAt"),
});

```

```

const {
  GoogleGenerativeAI,
  HarmCategory,
  HarmBlockThreshold,
} = require("@google/generative-ai");

```

```

const apiKey = process.env.NEXT_PUBLIC_GEMINI_API_KEY;
const genAI = new GoogleGenerativeAI(apiKey);

```

```

const model = genAI.getGenerativeModel({
  model: "gemini-1.5-flash",
});

```

```
const generationConfig = {  
    temperature: 1,  
    topP: 0.95,  
    topK: 40,  
    maxOutputTokens: 8192,  
    responseMimeType: "text/plain",  
};  
  
export const chatSession = model.startChat({  
    generationConfig,  
});
```

```

import { SignIn } from "@clerk/nextjs";

export default function Page() {
  return (
    <section className="bg-white">
      <div className="lg:grid lg:min-h-screen lg:grid-cols-12">
        <section className="relative flex h-32 items-end bg-gray-900 lg:col-span-5 lg:h-full xl:col-span-6">
          

          <div className="hidden lg:relative lg:block lg:p-12">
            <a className="block text-white" href="#">
              <span className="sr-only">Home</span>
              <svg
                className="h-8 sm:h-10"
                viewBox="0 0 28 24"
                fill="none"
                xmlns="http://www.w3.org/2000/svg"
              >
                <path
                  d="M0.41 10.3847C1.14777 7.4194 2.85643 4.7861 5.2639 2.90424C7.6714 1.02234 10.6393 0
13.695 0C16.7507 0 19.7186 1.02234 22.1261 2.90424C24.5336 4.7861 26.2422 7.4194 26.98
10.3847H25.78C23.7557 10.3549 21.7729 10.9599 20.11 12.1147C20.014 12.1842 19.9138 12.2477 19.81
12.3047H19.67C19.5662 12.2477 19.466 12.1842 19.37 12.1147C17.6924 10.9866 15.7166 10.3841 13.695
10.3841C11.6734 10.3841 9.6976 10.9866 8.02 12.1147C7.924 12.1842 7.8238 12.2477 7.72
12.3047H7.58C7.4762 12.2477 7.376 12.1842 7.28 12.1147C5.6171 10.9599 3.6343 10.3549 1.61
10.3847H0.41ZM23.62 16.6547C24.236 16.175 24.9995 15.924 25.78
15.9447H27.39V12.7347H25.78C24.4052 12.7181 23.0619 13.146 21.95 13.9547C21.3243 14.416 20.5674
14.6649 19.79 14.6649C19.0126 14.6649 18.2557 14.416 17.63 13.9547C16.4899 13.1611 15.1341 12.7356
13.745 12.7356C12.3559 12.7356 11.0001 13.1611 9.86 13.9547C9.2343 14.416 8.4774 14.6649 7.7
14.6649C6.9226 14.6649 6.1657 14.416 5.54 13.9547C4.4144 13.1356 3.0518 12.7072 1.66
12.7347H0V15.9447H1.61C2.39051 15.924 3.154 16.175 3.77 16.6547C4.908 17.4489 6.2623 17.8747
7.65 17.8747C9.0377 17.8747 10.392 17.4489 11.53 16.6547C12.1468 16.1765 12.9097 15.9257 13.69
15.9447C14.4708 15.9223 15.2348 16.1735 15.85 16.6547C16.9901 17.4484 18.3459 17.8738 19.735
17.8738C21.1241 17.8738 22.4799 17.4484 23.62 16.6547ZM23.62 22.3947C24.236 21.915 24.9995
21.664 25.78 21.6847H27.39V18.4747H25.78C24.4052 18.4581 23.0619 18.886 21.95 19.6947C21.3243
20.156 20.5674 20.4049 19.79 20.4049C19.0126 20.4049 18.2557 20.156 17.63 19.6947C16.4899 18.9011
15.1341 18.4757 13.745 18.4757C12.3559 18.4757 11.0001 18.9011 9.86 19.6947C9.2343 20.156 8.4774
20.4049 7.7 20.4049C6.9226 20.4049 6.1657 20.156 5.54 19.6947C4.4144 18.8757 3.0518 18.4472 1.66
18.4747H0V21.6847H1.61C2.39051 21.664 3.154 21.915 3.77 22.3947C4.908 23.1889 6.2623 23.6147
7.65 23.6147C9.0377 23.6147 10.392 23.1889 11.53 22.3947C12.1468 21.9165 12.9097 21.6657 13.69
21.6847C14.4708 21.6623 15.2348 21.9135 15.85 22.3947C16.9901 23.1884 18.3459 23.6138 19.735
23.6138C21.1241 23.6138 22.4799 23.1884 23.62 22.3947Z"
            fill="currentColor"
          />
        </svg>
      </a>
    </div>
  
```

```
<h2 className="mt-6 text-2xl font-bold text-white sm:text-3xl md:text-4xl">
  Welcome to AI Interview Mockup 🎉
</h2>
```

```
<p className="mt-4 leading-relaxed text-white">
  Lorem, ipsum dolor sit amet consectetur adipisicing elit. Eligendi
  nam dolorum aliquam, quibusdam aperiam voluptatum.
</p>
</div>
</section>
```

```
<main className="flex items-center justify-center px-8 py-8 sm:px-12 lg:col-span-7 lg:px-16 lg:py-12 xl:col-span-6">
  <div className="max-w-xl lg:max-w-3xl">
    <div className="relative -mt-16 block lg:hidden">
      <a
        className="inline-flex size-16 items-center justify-center rounded-full bg-white text-blue-600 sm:size-20"
        href="#"
      >
        <span className="sr-only">Home</span>
        <svg
          className="h-8 sm:h-10"
          viewBox="0 0 28 24"
          fill="none"
          xmlns="http://www.w3.org/2000/svg"
        >
          <path
            d="M0.41 10.3847C1.14777 7.4194 2.85643 4.7861 5.2639 2.90424C7.6714 1.02234 10.6393 0
            13.695 0C16.7507 0 19.7186 1.02234 22.1261 2.90424C24.5336 4.7861 26.2422 7.4194 26.98
            10.3847H25.78C23.7557 10.3549 21.7729 10.9599 20.11 12.1147C20.014 12.1842 19.9138 12.2477 19.81
            12.3047H19.67C19.5662 12.2477 19.466 12.1842 19.37 12.1147C17.6924 10.9866 15.7166 10.3841 13.695
            10.3841C11.6734 10.3841 9.6976 10.9866 8.02 12.1147C7.924 12.1842 7.8238 12.2477 7.72
            12.3047H7.58C7.4762 12.2477 7.376 12.1842 7.28 12.1147C5.6171 10.9599 3.6343 10.3549 1.61
            10.3847H0.41ZM23.62 16.6547C24.236 16.175 24.9995 15.924 25.78
            15.9447H27.39V12.7347H25.78C24.4052 12.7181 23.0619 13.146 21.95 13.9547C21.3243 14.416 20.5674
            14.6649 19.79 14.6649C19.0126 14.6649 18.2557 14.416 17.63 13.9547C16.4899 13.1611 15.1341 12.7356
            13.745 12.7356C12.3559 12.7356 11.0001 13.1611 9.86 13.9547C9.2343 14.416 8.4774 14.6649 7.7
            14.6649C6.9226 14.6649 6.1657 14.416 5.54 13.9547C4.4144 13.1356 3.0518 12.7072 1.66
            12.7347H0V15.9447H1.61C2.39051 15.924 3.154 16.175 3.77 16.6547C4.908 17.4489 6.2623 17.8747
            7.65 17.8747C9.0377 17.8747 10.392 17.4489 11.53 16.6547C12.1468 16.1765 12.9097 15.9257 13.69
            15.9447C14.4708 15.9223 15.2348 16.1735 15.85 16.6547C16.9901 17.4484 18.3459 17.8738 19.735
            17.8738C21.1241 17.8738 22.4799 17.4484 23.62 16.6547ZM23.62 22.3947C24.236 21.915 24.9995
            21.664 25.78 21.6847H27.39V18.4747H25.78C24.4052 18.4581 23.0619 18.886 21.95 19.6947C21.3243
            20.156 20.5674 20.4049 19.79 20.4049C19.0126 20.4049 18.2557 20.156 17.63 19.6947C16.4899 18.9011
            15.1341 18.4757 13.745 18.4757C12.3559 18.4757 11.0001 18.9011 9.86 19.6947C9.2343 20.156 8.4774
            20.4049 7.7 20.4049C6.9226 20.4049 6.1657 20.156 5.54 19.6947C4.4144 18.8757 3.0518 18.4472 1.66
            18.4747H0V21.6847H1.61C2.39051 21.664 3.154 21.915 3.77 22.3947C4.908 23.1889 6.2623 23.6147
            7.65 23.6147C9.0377 23.6147 10.392 23.1889 11.53 22.3947C12.1468 21.9165 12.9097 21.6657 13.69
            21.6847C14.4708 21.6623 15.2348 21.9135 15.85 22.3947C16.9901 23.1884 18.3459 23.6138 19.735
            23.6138C21.1241 23.6138 22.4799 23.1884 23.62 22.3947Z"
              fill="currentColor"
            />
          </svg>
        </a>
```

Chapter 9

Result and Output

Screens

CHAPTER-9

RESULT AND OUTPUT SCREEN

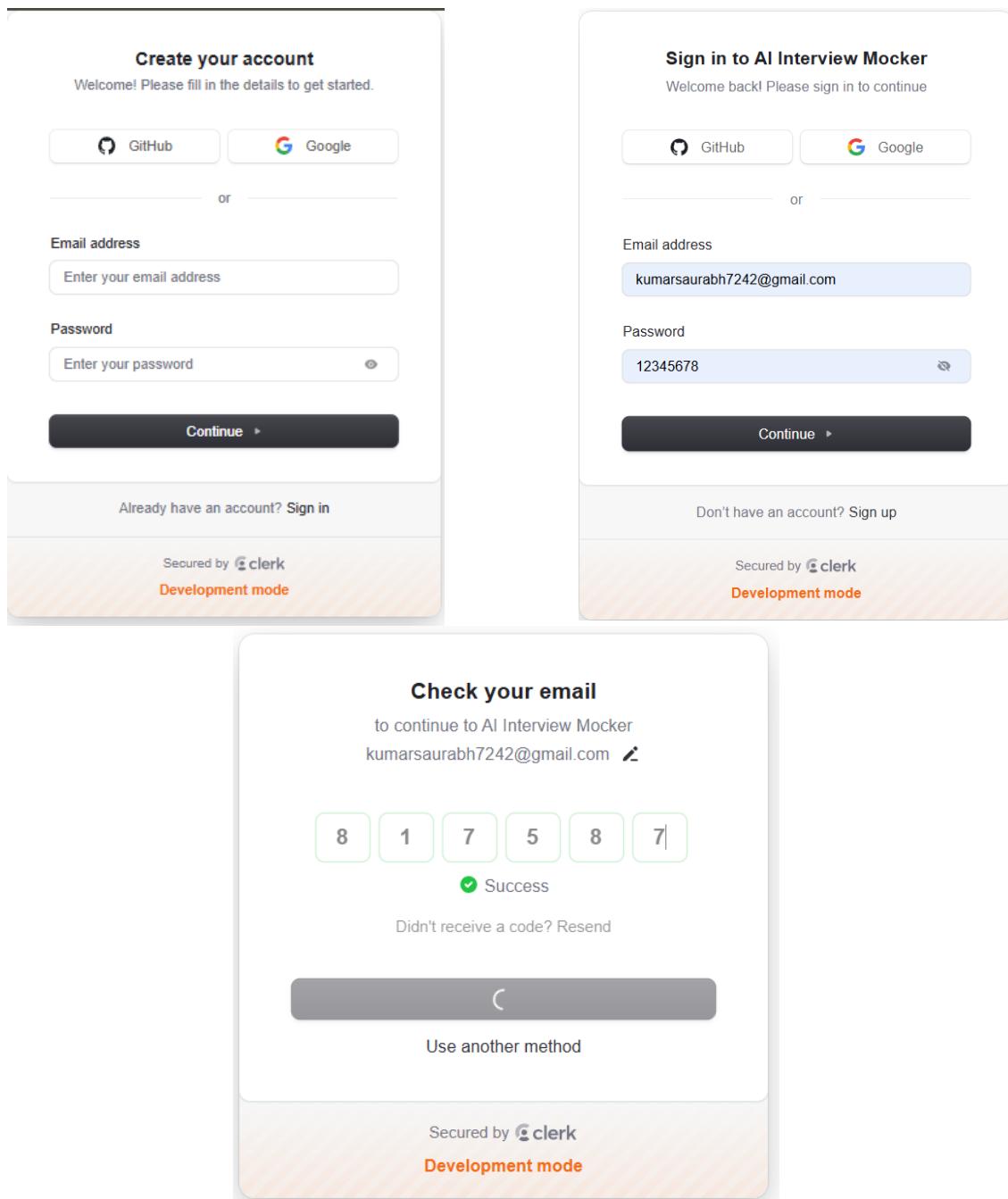
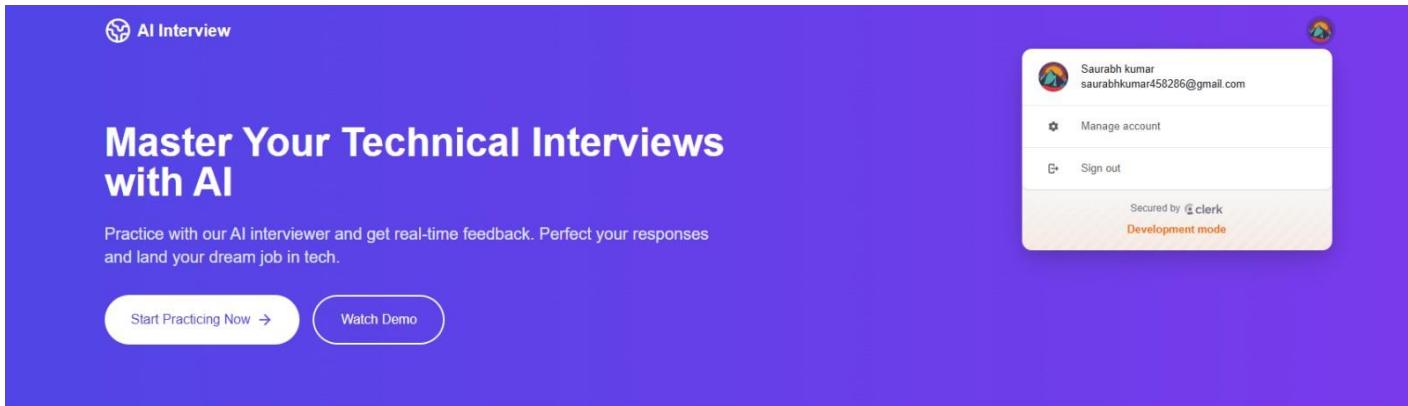


Figure 9.1: Login & Sign Up Page



Master Your Technical Interviews with AI

Practice with our AI interviewer and get real-time feedback. Perfect your responses and land your dream job in tech.

[Start Practicing Now →](#) [Watch Demo](#)

Why Choose AI Interview?

AI-Powered Practice

Get realistic interview experience with our advanced AI interviewer that adapts to your responses.

Real-time Feedback

Receive instant feedback on your answers, communication skills, and technical accuracy.

Industry Standards

Practice with questions based on real interviews from top tech companies.

Trusted by Developers Worldwide

Join thousands of developers who have improved their interview skills and landed jobs at top tech companies.

Sarah Chen
Software Engineer at Google

"GiveInterview helped me prepare effectively for my technical interviews. The AI feedback was invaluable."

Michael Rodriguez
Senior Developer at Microsoft

"The realistic interview scenarios and detailed feedback helped me gain confidence in my abilities."

Emily Johnson
Full Stack Developer at Meta

"I credit my successful career transition to the practice I got through GiveInterview."

Ready to Ace Your Next Technical Interview?

[Get Started Free →](#)

Empowering developers to succeed in technical interviews through AI-powered practice.

Product

- Features
- Pricing
- Enterprise

Resources

- Blog
- Documentation
- Community

Company

- About
- Careers
- Contact

© 2024 AI Interview. All rights reserved.

Figure 9.2: Landing Page & Footer Page

The screenshot displays the AI Interview application interface. At the top, there is a navigation bar with links for Dashboard, Question, Feedback, and About Us. On the right side, a user profile is shown with a purple circular icon containing a white letter 'S', the name "Science And Technology", and the email "kumarsaurabh7242@gmail.com". Below the profile are options for "Manage account" and "Sign out". A note at the bottom right states "Secured by clerk" and "Development mode".

Dashboard Section:

- Header:** Dashborad
- Text:** Create and Start Your AI Mockup Interview
- Buttons:** + Add New

Previous Mock Interview Section:

Interview Type	Experience	Created At	Action Buttons
JavaScript Developer	0 Years of Experience	Created At: 16-02-2025	Feedback Start
java developer	0 Years of Experience	Created At: 16-02-2025	Feedback Start
Java Developer	0 Years of Experience	Created At: 26-01-2025	Feedback Start
frontend developer	0 Years of Experience	Created At: 26-01-2025	Feedback Start
frontend	0 Years of Experience	Created At: 20-01-2025	Feedback Start

Add New Interview Dialog:

Title: Tell us more about your job interviewing

Text: Add Details about your position/role, Job description and years of experience

Form Fields:

- Job Role/Job Position: Frontend Developer
- Job Description/ Tech Stack (In Short): React, html, javascript, css
- Years of experience: 1

Buttons:

- Cancel
- 'Generating From AI'
- Start

Figure 9.3: Dashboard & Add New Interview Page

 AI INTERVIEW

Dashboard Question Feedback About Us 

Let's Get Started

Job Position/Role : Frontend Developer

Job Description/Tech Stack: React, html,javascript, css

Years of Experience: 1



 **Information**

Enable Video Web Cam and Microphone to Start your AI Generated Mock Interview. It has 5 question which you can answer and at the last you will get the report on the basis of your answer. NOTE: We never record your video, Web cam access you can disable at any time if you want

Start Interview



 AI INTERVIEW

Dashboard Question Feedback About Us 

Question #1
Question #2
Question #3
Question #4

Question #5

 **Note**

Enable Video Web Cam and Microphone to Start your AI Generated Mock Interview. It has 5 question which you can answer and at the last you will get the report on the basis of your answer. NOTE: We never record your video, Web cam access you can disable at any time if you want



Record Answer
User answer recorded successfully

Previous Question
Next Question

Figure 9.4: Interview Page

The screenshot displays two pages from the AI Interview platform. The top section is the 'Feedback Page' showing a summary of the interview rating and specific questions with user responses and feedback. The bottom section is the 'Previous Mock Interview Page' listing various developer roles with their experience levels and creation dates.

Feedback Page Summary:

- Your overall interview rating: **06/10**
- Find below interview questions with correct answers, your answers, and feedback for improvement.
- Describe your experience working with React.js. What are some of your favorite React features and why?
- Rating: 3**
- Your Answer:** I have been actively working with realization several personal and academic project some example include a blog application Notes app and react with documentation websit in this project I have handled both front and you it is an an integration with battery API using tools like exihilo DB and firebase a sum of the favourite react features like react hope JavaScript XML component based architecture virtual domain performance Optimisation reactive
- Correct Answer:** I've worked with React for approximately one year; building [mention type of projects, e.g., single-page applications, interactive components, etc.] My favorite features include its component-based architecture, which promotes reusability and maintainability. I also appreciate JSX for its intuitive syntax that bridges HTML and JavaScript, making component creation cleaner and more readable. Finally, React's virtual DOM significantly improves performance by minimizing direct manipulations of the actual DOM.
- Feedback:** The answer lacks clarity and professionalism. Rewrite focusing on specific React features (e.g., hooks, JSX, component lifecycle) and quantifiable achievements. Replace vague phrases like 'handled both front and you it is an an integration' with concrete examples. Improve grammar and spelling significantly.
- How do you handle asynchronous operations in React? Explain your approach to fetching data from an API.
- Rating: 1**
- Your Answer:** in react state management refer to how data or a state is a store excessed and updated across component in a application react has built in local state the using a status like you just stayed which is perfect for component level estate but as an approach specifically with many component sharing data managing state become Complex so tool that have used for the stain management like contact API in react state management refer to how data or a state is a store excessed and updated across component in a application react has built in local state the using a status like you just stayed which is perfect for component level estate but as an approach specifically with many component sharing data managing state become Complex so tool that have used for the stain management like contact API just stand reductionin react state management refer to how data or state is stored exist and update updated occurs component is an application react has it on react state management refer to how data or a state is a store excessed and updated across component in a application react has built in local state the using a status like you just stayed which is perfect for component level estate but as an approach specifically with many component sharing data managing state become Complex so tool that have used for the stain management like contact API just stand reductionin react state management refer to how data or state is stored exist and update updated occurs component is an application react has it on local state using you just state which is perfect for the component level toast that have used for a state management like context API just and or rearden react state management refer to how data or a state is a store excessed and updated across component in a application react has built in local state the using a status like you just stayed which is perfect for component level estate but as an approach specifically with many component sharing data

Previous Mock Interview Page:

Role	Experience	Created At	Action Buttons
python backend	0 Years of Experience	Created At: 03-02-2025	Feedback Start
java developer	4 Years of Experience	Created At: 03-02-2025	Feedback Start
Java Developer	5 Years of Experience	Created At: 03-02-2025	Feedback Start
react developer	0 Years of Experience	Created At: 30-01-2025	Feedback Start
Java Developer	0 Years of Experience	Created At: 30-01-2025	Feedback Start
frontend developer	0 Years of Experience	Created At: 24-01-2025	Feedback Start
Frontend developer	0 Years of Experience	Created At: 23-01-2025	Feedback Start
junior developer	0 Years of Experience	Created At: 21-01-2025	Feedback Start
Junior developer	0 Years of Experience	Created At: 21-01-2025	Feedback Start
junior developer	0 Years of Experience	Created At: 21-01-2025	Feedback Start
backend developer	0 Years of Experience	Created At: 19-01-2025	Feedback Start
frontend developer	0 Years of Experience	Created At: 18-01-2025	Feedback Start
frontend	0 Years of Experience	Created At: 17-01-2025	Feedback Start
frontend developer	0 Years of Experience	Created At: 16-01-2025	Feedback Start
frontend developer	1 Years of Experience	Created At: 16-01-2025	Feedback Start

Figure 9.5: Feedback Page & Previous Mock Interview Page

	id	serial	jsonMockResp	text	jobPosition	varchar	jobDesc	varchar	jobExperience	varchar	createdBy	varchar	createdAt	varchar	mockId	varchar
	1		[{ "Question": "Descri..."	Full stack developer	react, nodejs		1				saurabhkumar458286@gmail...	15-01-2025		e982e3b7-7f13-4e3d-a87...		
	2		[{ "Question": "Descri..."	Full Stack Developer	nodejs, javascript, my...		4				saurabhkumar458286@gmail...	15-01-2025		c5758fc-9f10-4367-94e...		
	3		[{ "Question": "Expla..."	java developer	java, springboot		1				saurabhkumar458286@gmail...	15-01-2025		60431030-a874-4372-93b...		
	4		[{ "Question": "Descri..."	full stack	java		1				saurabhkumar458286@gmail...	15-01-2025		bfc43fd3-1acd-4103-8c6...		
	5		[{ "Question": "You've..."	javascript	html,css		1				saurabhkumar458286@gmail...	16-01-2025		65baa7f6-58c3-4c1a-871...		
	6		[{ "Question": "Descri..."	Full stack developer	react, angular		3				saurabhkumar458286@gmail...	16-01-2025		a04e28b1-cae4-4972-891...		
	7		[{ "Question": "Descri..."	frontend	react		1				saurabhkumar458286@gmail...	16-01-2025		6c380bcf-0fe-42c2-b63...		
	8		[{ "Question": "Descri..."	frontend developer	html, css		1				saurabhkumar458286@gmail...	16-01-2025		806652f0-426e-474d-b27...		
	9		[{ "Question": "Expla..."	frontend developer	html,css		0				saurabhkumar458286@gmail...	16-01-2025		46062578-10a4-429f-ae5...		
	10		[{ "Question": "Expla..."	frontend	html,css		0				saurabhkumar458286@gmail...	17-01-2025		696e2c16-6ae0-44e6-8cf...		
	11		[{ "Question": "You ha..."	frontend developer	react, html, css		0				saurabhkumar458286@gmail...	18-01-2025		d866edae-8c00-499b-b66...		
	12		[{ "Question": "Expla..."	backend developer	nodejs, javascript		0				saurabhkumar458286@gmail...	19-01-2025		74b26111-93c2-464d-ba0...		
	13		[{ "Question": "Expla..."	frontend	html,css		0				kumarsaurabh7242@gmail...	20-01-2025		d0ec8669-20e0-4eff-a94...		
	14		[{ "Question": "Expla..."	junior developer	c++		0				saurabhkumar458286@gmail...	21-01-2025		61f8a988-8550-4abf-927...		
	15		[{ "Question": "You ha..."	Junior developer	c++		0				saurabhkumar458286@gmail...	21-01-2025		5dc36547-def9-48a7-9e2...		
	16		[{ "Question": "Descri..."	junior developer	html, css		0				saurabhkumar458286@gmail...	21-01-2025		bea59f6d-b854-44b4-81c...		
	17		[{ "Question": "Expla..."	Frontend developer	html,css,javascript,re...		0				saurabhkumar458286@gmail...	23-01-2025		a5588996-5ef3-44c4-819...		
	18		[{ "Question": "Expla..."	frontend developer	html,css,javascript		0				saurabhkumar458286@gmail...	24-01-2025		af0976ef-f962-4a84-83d...		
	19		[{ "Question": "Expla..."	frontend developer	html,css,javascript		0				kumarsaurabh7242@gmail...	26-01-2025		018bbde8-69f5-4a94-b0a...		
	20		[{ "Question": "Expla..."	Java Developer	java, springboot		0				kumarsaurabh7242@gmail...	26-01-2025		6128f34a-db62-455d-abf...		
	21		[{ "Question": "You ha..."	c++ developer	c++		0				kumarsaurabh8286@gmail...	30-01-2025		4e4ada8e-8128-4fc6-814...		
	22		[{ "Question": "You ha..."	full stack developer	react, html		0				kumarsaurabh8286@gmail...	30-01-2025		35394b5b-f5fe-4cd2-89c...		

	id	serial	mockId	varchar	question	varchar	correctAns	text	userAns	text	feedback	text	rating	varchar	userEmail	varchar	createdAt	varchar
	56		e2d800e2-0094-4286-b22...	Explain the difference...	In JavaScript: * `var` -	The main difference be...		The answer is confusin...		2		kumarsaurabh8286@gmail...	27-03-2025					
	55		84bd2f2c-dcb5-4197-884...	What is an event liste...	An event listener is a...	an event listener in J...		The answer is incomple...		2		kumarsaurabh7242@gmail...	16-02-2025					
	54		84bd2f2c-dcb5-4197-884...	Describe the concept o...	A closure is a functio...	a closer in JavaScript...		The answer is largely -		3		kumarsaurabh7242@gmail...	16-02-2025					
	53		84bd2f2c-dcb5-4197-884...	What are JavaScript da...	JavaScript has several...	JavaScript has 8 prima...		The answer is good, co...		4		kumarsaurabh7242@gmail...	16-02-2025					
	52		84bd2f2c-dcb5-4197-884...	Explain the difference...	Both are comparison op...	double equal is actual...		The answer correctly i...		3		kumarsaurabh7242@gmail...	16-02-2025					
	51		84bd2f2c-dcb5-4197-884...	What is JavaScript, an...	JavaScript is a progra...	JavaScript is a progra...		The answer is good, co...		4		kumarsaurabh7242@gmail...	16-02-2025					
	50		84bd2f2c-dcb5-4197-884...	What is an event liste...	An event listener is a...	event listener basical...		The answer is too brie...		2		kumarsaurabh7242@gmail...	16-02-2025					
	49		84bd2f2c-dcb5-4197-884...	Describe the concept o...	A closure is a functio...	closures are the prop...		The answer is too vagu...		2		kumarsaurabh7242@gmail...	16-02-2025					
	48		84bd2f2c-dcb5-4197-884...	What are JavaScript da...	JavaScript has several...	there are wages data t...		The answer is incomple...		2		kumarsaurabh7242@gmail...	16-02-2025					
	47		84bd2f2c-dcb5-4197-884...	Explain the difference...	Both are comparison op...	the slide difference b...		The answer is partiall...		3		kumarsaurabh7242@gmail...	16-02-2025					
	46		84bd2f2c-dcb5-4197-884...	What is JavaScript, an...	JavaScript is a progra...	JavaScript is a client...		The answer is repetiti...		2		kumarsaurabh7242@gmail...	16-02-2025					
	45		2d710355-9425-4b09-bc2...	What is an exception i...	An exception is an eve...	see if there is a chan...		The answer gives a bas...		3		kumarsaurabh7242@gmail...	16-02-2025					
	44		2d710355-9425-4b09-bc2...	What are Java access m...	Java access modifiers ..	access modifier basica...		The answer is vague an...		2		kumarsaurabh7242@gmail...	16-02-2025					
	43		2d710355-9425-4b09-bc2...	What is object-orientate...	Object-oriented progra...	object oriented progra...		The answer is partiall...		2		kumarsaurabh7242@gmail...	16-02-2025					
	42		2d710355-9425-4b09-bc2...	What is the difference...	Both 'ArrayList' and '...	double equal operator ...		The answer is complete...		1		kumarsaurabh7242@gmail...	16-02-2025					
	41		666a4bab-e7b6-4140-b13...	Explain the difference...	Lists and tuples are b...	there is no difference...		This answer is complet...		1		kumarsaurabh8286@gmail...	04-02-2025					
	40		666a4bab-e7b6-4140-b13...	What is your understan...	I understand Python as...	just a sample test cas...		This answer is unclear...		2		kumarsaurabh8286@gmail...	04-02-2025					
	39		666a4bab-e7b6-4140-b13...	Let's say you encounte...	My approach to debuggi...	let's say you encounte...		The answer is vague. I...		2		kumarsaurabh8286@gmail...	04-02-2025					
	38		666a4bab-e7b6-4140-b13...	What are some common P...	I'm familiar with seve...	what are the some comm...		The answer directly st...		1		kumarsaurabh8286@gmail...	04-02-2025					
	37		666a4bab-e7b6-4140-b13...	Describe your experien...	While I haven't used G...	describe your experien...		The answer shows a ver...		2		kumarsaurabh8286@gmail...	04-02-2025					
	36		666a4bab-e7b6-4140-b13...	Explain the difference...	Lists and tuples are b...	the least and tuples a...		The answer is nonsensic...		1		kumarsaurabh8286@gmail...	04-02-2025					
	35		666a4bab-e7b6-4140-b13...	What is your understan...	I understand Python as...	it is just a sample to...		This response is compl...		1		kumarsaurabh8286@gmail...	04-02-2025					
	34		29a28298-f0b4-48e6-86d...	What are some best pra...	I adhere to several be...	what are the some best...		This answer is complet...		1		saurabhkumar458286@gmail...	03-02-2025					
	33		29a28298-f0b4-48e6-86d...	Describe a time you ha...	In one project, we enc...	describe a time you ha...		This response is nonse...		1		saurabhkumar458286@gmail...	03-02-2025					
	32		29a28298-f0b4-48e6-86d...	How do you handle exce...	I handle exceptions in...	so any I never handle ...		This answer is complet...		1		saurabhkumar458286@gmail...	03-02-2025					

Figure 9.6: Database

Chapter 10

Conclusion and

Future Work

CHAPTER-10

CONCLUSION AND FUTURE WORK

10.1 CONCLUSION

In conclusion, the development of the **AI-Powered Mock Interview App** represents a significant advancement in the way individuals prepare for technical job interviews. By combining cutting-edge technologies like AI-based question generation, resume parsing, voice input analysis, and real-time feedback, the app delivers an immersive and personalized interview practice experience.

This project was built with a strong focus on solving real-world problems faced by job seekers—such as lack of personalized preparation, limited access to mock interviews, and insufficient feedback. Through rigorous planning, modular architecture, and integration of tools like Gemini AI and Firebase, we created an application that is not only user-centric but also scalable and adaptable.

The modular nature of the application ensures ease of future expansion while maintaining clean and manageable code. From uploading resumes and generating custom technical questions to recording answers and evaluating them in real-time, every feature is designed to simulate a realistic interview environment.

Overall, the **AI Mock Interview App** is not just a preparation tool—it's an intelligent, interactive platform that empowers users to build confidence, improve performance, and take meaningful steps toward their career goals. With continuous improvements and feature expansions, this app has the potential to become a go-to companion for job seekers worldwide.

10.2 FUTURE WORK

1. Advanced AI-Based Evaluation

Integrate Natural Language Processing (NLP) and speech emotion recognition for deeper answer evaluation. Provide scoring on multiple criteria such as tone, fluency, technical depth, and confidence.

2. Gemini-Driven Interview Simulations

Simulate multi-round interviews with increasing difficulty levels.

Expand Gemini API usage for dynamic follow-up questions and context-aware responses.

3. Speech-To-Text Accuracy Improvements

Integrate advanced models for better real-time voice-to-text conversion.

Provide real-time hints and corrections during the interview.

4. Dashboard For Performance Tracking

Enable simulations for HR, Technical, and Managerial rounds.

Create scenario-based behavioral question simulations.

REFERENCES

- [1] **VS Code Documentation:** <https://code.visualstudio.com/docs> : Official documentation for Visual Studio Code, used for building and managing the AI Interview application.
- [2] **Drizzle ORM Documentation:** <https://orm.drizzle.team/> : Guide and references for Drizzle ORM, used for handling relational database operations within the project.
- [3] **Gemini API (Google AI):** <https://ai.google.dev/> : Source for integrating Gemini AI to generate personalized technical interview questions.
- [4] **Node.js Documentation:** <https://nodejs.org/en/docs/> Backend runtime environment documentation used for server-side logic and API handling.
- [5] **Next.js Documentation:** <https://nextjs.org/docs> Framework documentation for building the frontend of the web application.
- [6] **Web Speech API (SpeechRecognition):** Reference used for implementing voice-based input and real-time speech-to-text features.
- [7] **YouTube Tutorials:** <https://www.youtube.com> Learning platform for various development topics including Next.js, Drizzle ORM, and Web Speech API.

PROJECT SUMMARY

About Project

Title of the project	Design and Development of Personalized AI-Powered Mock Interview Web Application
Semester	8 th
Members	4
Team Leader	Saurabh Vishwakarma
Describe role of every member in the project	Shubham Yadav – User Interface & AI Integration Saurabh Vishwakarma – Drizzle ORM & Resume Upload Functionality & DataBase Management. Satyam Rajak – User Interface & DataBase Management. Vishal Sisodiya – Deployment & Speech Recognition Integration
What is the motivation for selecting this project?	With the rising competition in job markets, many candidates struggle with interview preparation. The motivation behind this project is to build a smart platform that enables users to practice technical interviews by uploading their resumes and receiving AI-generated questions, real-time feedback, and speech analysis. The goal is to help users improve confidence, fluency, and accuracy before facing real interviews.
Project Type (Desktop Application, Web Application, Mobile App, Web)	Web Application (Responsive for both desktop and mobile)

Tools & Technologies

Programming language used	JavaScript (Next.js), SQL
Compiler used (with version)	Node.js (v18+)
IDE used	Visual Studio Code

(with version)	
Front End Technologies (With version, wherever Applicable)	Next.js, Tailwind CSS, ShadCN UI
Back End Technologies (With version, wherever applicable)	Node.js, Express.js
Database used (with version)	Neon with Drizzle ORM

Software Design & Coding

Is the prototype of the software developed?	Yes
SDLC model followed (Waterfall, Agile, Spiral etc.)	Big Bang Model
Why is the above SDLC model followed?	The Big Bang model is ideal for small, straightforward projects with clearly defined scope. It requires minimal planning, as all resources and efforts are focused directly on development and implementation, without incremental phases. This approach works best when requirements are stable and easily understood, minimizing complexity and ensuring rapid completion.
Justify that the SDLC model mentioned above is followed in the project.	The project adheres to the SDLC model by following systematic phases: requirement gathering to understand user needs, system design to create the app's architecture, implementation to develop the features, testing to ensure functionality and reliability, deployment for user access, and maintenance for ongoing improvements.
Software Design approach followed (Functional or Object-oriented)	Object Oriented
Name the diagrams developed (According to the Design approach followed)	Use Case Diagram, Data Flow Diagram

In case Object Oriented approach is followed, which of the OOPS principles are covered in design?	Encapsulation, Inheritance
No. of Tiers (example 3-tier)	2-tier
Total no. of front-end pages	22
Total no. of tables in database	4
Database in which Normal Form?	3NF
Are the entries in the database encrypted?	Yes
Front end validations applied (Yes / No)	Yes
Session management done (in case of web applications)	
Is application browser compatible (in case of web applications)	
Exception handling done (Yes / No)	Yes
Commenting done in code (Yes / No)	Yes
Naming convention followed (Yes / No)	Yes
What difficulties faced during deployment of the project?	
Total no. Of Use-cases	1
Given titles of Use-cases	Use Case Diagram

Project Requirements

MVC architecture followed (Yes / No)	No
If yes, write the name of MVC architecture followed (MVC-1, MVC-2)	N/A
Design Pattern used (Yes / No)	No
If yes, write the name of Design Pattern used	N/A
Interface type (CLI / GUI)	GUI
No. of Actors	2
Name of Actors	User, Admin
Total no. of Functional Requirements	8
List few important non-Functional Requirements	Performance, Security, Compatibility

Testing

Which testing is performed? (Manual or Automation)	Manual
Is Beta testing done for this project?	Yes

**Write project narrative covering
above mentioned points**

The AI-Powered Mock Interview Assistant is an innovative solution aimed at transforming interview preparation. Users upload their resumes, and AI (e.g., Gemini API) generates tailored technical questions. The system offers a real-time practice environment with voice recognition, answer recording, and performance evaluation. Built using modern technologies like Next.js, Drizzle ORM, and Neon DB, the platform ensures scalability, speed, and security. The GUI is clean and user-friendly, and extensive manual testing ensured a smooth user experience. With two main actors (User & Admin), the app supports dynamic interaction and content updates without authentication.

Saurabh Vishwakarma	0187CS211154	Guide Signature Prof. Deepti Jain
Shubham Yadav	0187CS223D03	
Vishal Sisodiya	0187CS211184	
Satyam Rajak	0187CS211151	

APPENDIX 1**GLOSSARY OF TERM**

A	
AI	The simulation of human intelligence processes by machines, especially computer systems. In this project, AI (like Gemini API) is used to generate technical interview questions based on resume content.
API	The A set of functions and protocols that allow different software components to communicate. Gemini API is integrated in this project for generating questions.
D	
Drizzle ORM	A lightweight Object Relational Mapper used to interact with SQL databases in a type-safe and schema-defined way. It is used with Neon database in this project.
G	
GUI	A visual interface that allows users to interact with the application through graphical elements like buttons and forms instead of code or command lines.
I	
IDE	Software that provides tools like code editor, debugger, and terminal in one place to streamline development. Visual Studio Code (VS Code) is the IDE used in this project.