AI MOCK INTERVIEW

A PROJECT REPORT
for
Mini-Project 2 (ID201B)
Session (2024-25)

Submitted by

Tanishka Gupta (202410116100218)
Shipra Upadhyay (202410116100196)
Vanshika Garg (202410116100235)
Tripti Rajpoot (202410116100225)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Ms. Shruti Aggarwal
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(MAY 2025)

CERTIFICATE

Certified that **Tanishka Gupta 202410116100218, Shipra Upadhyay 202410116100196**, **Vanshika Garg 202410116100235** and **Tripti Rajput 202410116100225** have carried out the project work having “**Title of Report AI mock Interview**” (**Mini Project-II, ID201B**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Ms. Shruti Aggarwal
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Akash Rajak
Dean
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

AI MOCK INTERVIEW

Tanishka Gupta, Shipra Upadhyay, Vanshika Garg, Tripti Rajput

ABSTRACT

The AI Mock Interview Web Application is a web-based, smart platform that assists job applicants in improving their interview performance using real-time simulations powered by AI. With the highly competitive nature of the job market today, there is a pressing need to be well prepared for interviews to secure a job. The conventional mock interview process can be expensive and time-consuming since it involves human mentors. The project **utilizes Artificial Intelligence (AI)** and **Natural Language Processing (NLP)** to develop an interactive, scalable solution for practicing interviews.

The site creates **dynamic**, **profession-specific interview questions** based on the user's role and level of experience. Users can answer in text or voice, and AI-based assessment offers immediate feedback on factors like clarity, confidence, technical correctness, and aptness. The system also features performance monitoring, analytics, and customized improvement recommendations, enabling users to hone their answers with repeated practice sessions.

The most important goals of this project are automated generation of interview questions, response **assessment based on AI**, and **adaptive learning processes** to make users more prepared. The app is multi-industry-compatible, so it can be used by students, new graduates, and professionals.

Through the offering of an **affordable, cost-efficient, and AI-driven** alternative to conventional mock interviews, this platform equips users with the skillset and confidence needed to succeed in actual job interviews, thus driving their chances of getting hired.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms. Shruti Aggarwal, Dr. Neelam Rawat and Dr. Amit Kr. Gupta** for their guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Akash Rajak, Professor and Dean, Department of Computer Applications**, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Tanishka Gupta

Shipra Upadhyay

Vanshika Garg

Tripti Rajput

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Background on AI Mock Interview

With the current competitive and dynamic nature of the job market, landing a job involves more than mere academic credentials. Employers are now considering candidates more on the basis of communication skills, confidence, problem-solving skills, and cultural alignment—usually tested through the process of job interviews. Yet, numerous candidates experience fear and do not have the organized guidance to enable them to prepare well for such interviews.

This void has been filled by AI-based mock interview platforms that are capable of replicating actual interview situations, offering instant feedback, and enabling users to continuously refine their performance. These platforms leverage developments in Artificial Intelligence (AI) and Natural Language Processing (NLP) to assess user responses—either verbal or textual—and offer personalized improvement suggestions.

The **AI Mock Interview** Web Application is designed with the purpose of assisting students, job applicants, and professionals with interview practice in a virtual setting. Through dynamically generated role-specific interview questions and intelligent analysis of user answers, the system replicates the actions of an actual interviewer. It renders unnecessary costly training or manual mock interviewing by providing an automated, real-time, and scalable method.

By AI algorithms like GPT-4 or Gemini, the system analyses key dimensions of a user's performance such as clarity, confidence, relevance, and technical accuracy. By integrating analytics and feedback systems, users are able to track their improvement and better prepare themselves for actual interviews under real-world conditions.

1.2 Scope of the Project

The **AI Mock Interview Web Application** is conceptualized as an extensive digital solution enabling users to conduct actual interview sessions using the services of Artificial Intelligence. The scope of the project is to create a **user-friendly, scalable, and role-based web application that accommodates interactive interviews**, intelligent evaluations, and performance monitoring for users across various industries and experience levels.

Key Features within the Scope:

1. User Role Management

The platform accommodates different types of users:

- Job Seekers: Can take simulated interviews and get AI-based feedback.
- Students: Can take mock interviews based on academic-level job positions.
- Professionals: Can choose higher-level positions for more technical or managerial mock interviews.

2. Dynamic Question Generation

The process employs AI API (such as Gemini or GPT-4) to:

- Produce type-specific questions to match job classification and experience level.
- Modify questions dynamically through prior user answer patterns.

3. Verbal and Keyboard-Based Answers

Clients can:

- Type answer responses in real time.
- Record verbal answers to offer natural dialogue mimicking.

4. NLP-Based Feedback

The system:

- Measures confidence, clarity, tone, and relevance with the help of NLP.
- Gives graded feedback and ratings.

5. Analytics

- Allows users to compare historical performance.
- Shows improvement trends and suggestions graphically.

1.3 Purpose and Objectives

The objective of this project is to create and develop an intelligent, interactive **AI Mock Interview Web Application** that assists job applicants in practicing and improving their interview skills using AI-based simulations. This platform overcomes the drawbacks of conventional mock interviews—high costs, unavailability, and inconsistent feedback—by offering a scalable, customized, and automated solution based on **Artificial Intelligence (AI)** and **Natural Language Processing (NLP)**.

The project is designed to enable users with live practice, positive feedback, and tracking of their progress, hence boosting their confidence and preparedness for real job interviews. Through the simulation of realistic interview scenarios in different industries and levels of jobs, the application fosters inclusivity and flexibility for users with different professional aspirations.

Objectives of the Project:

1. Automated Interview Question Generation

- Create an AI system that can produce dynamic and job-related interview questions based on users' industries, job positions, and levels of experience.

2. Real-Time Response Analysis

- Use NLP to analyse user responses (voice or text) against parameters like clarity, relevance, confidence, tone, and technical correctness.

3. Personalized Feedback and Recommendations

- Give instant, actionable feedback on users' strengths and weaknesses.
- Provide improvement suggestions in communication, content quality, and overall performance.

4. User Progress Tracking and Analytics

- Integrate performance tracking features to track user development across several sessions.
- Utilize data visualization and scoring trends to enable users to recognize important areas for development.

5. Multi-Industry and Experience-Level Support

- Provide interview preparation support for different industries such as IT, finance, healthcare, engineering, and others.
- Create a platform that is accessible to students, new graduates, and experienced professionals.

1.4 Significance

The creation of an **AI Mock Interview Web Application** is extremely important in the context of the current competitive and dynamic job market. Since job interviews are essential portals to career opportunities, preparation is more important than ever. Nevertheless, conventional interview coaching techniques tend to fail because of reasons such as restricted accessibility, prohibitive costs, and variable quality of feedback. This project deals with these challenges by taking advantage of the potential of Artificial Intelligence (AI) and Natural Language Processing (NLP) to offer a solid, affordable, and smart solution.

Areas of Importance:

1. Improved Interview Readiness: The system provides a real-world and effective setting for job applicants to prepare through naturalistic, AI-simulated interviews. This makes it possible for users to become exposed to different patterns of questions and enhance their performance while responding under pressure.

2. Increased Accessibility and Inclusivity: In contrast to traditional mock interviews that necessitate human guides, this site operates 24/7 for anyone with access to the internet. This makes interview practice more accessible to students, graduates, and working professionals in urban and rural areas alike.

3. Objective and Real-Time Feedback: Through the application of NLP and AI-based algorithms, the system is capable of objectively judging user responses. It offers immediate feedback on verbal delivery, tone, relevance of content, and confidence levels, allowing users to hone their skills step by step.

4. Continuous Learning and Performance Monitoring: The app monitors user improvement over multiple sessions, providing analytics of growth points and frequent mistakes. This ongoing learning strategy means users are not only practicing, but enhancing.

5. Support of Multiple Industries and Experience Levels: Whether a user is preparing for an entry-level IT job or a senior role in healthcare management, the system adapts accordingly. This multi-industry support ensures relevance and applicability across various career paths.

6. Technological Innovation in Career Services: The project illustrates how new technologies may be leveraged to reshape conventional education and career development paradigms. It demonstrates the application of AI not only for automating tasks, but for customized, smart learning experiences.

By offering an effective alternative to traditional mock interviews, this project helps to bridge the gap between theoretical learning and practical preparation—eventually, boosting the probability of success at actual job interviews for a variety of users.

1.5 Overview of Methodology

The implementation of the **AI Mock Interview Web Application** involves a systematic methodology to achieve systematic implementation, from design and planning to implementation and testing. The methodology integrates best practices in software engineering and AI-specific development approaches to design a scalable, interactive, and intelligent mock interview system.

Step-by-Step Overview of Methodology:

1. Requirement Analysis

- Collect and evaluate requirements from prospective users like students, career seekers, and professionals.
- Determine required features like dynamic question generation, instant feedback, and performance monitoring.

2. System Design

- Develop the system architecture with a proper separation between the frontend (user interface), backend (server logic), and AI/NLP modules.
- Implement navigation and responsiveness on various devices.
- Specify workflows like user registration, role assignment, interview session flow, and feedback submission.

3. Technology Stack Choice

- Frontend: Next.js and Tailwind CSS for a responsive and dynamic interface.
- Backend: Next.js and Firebase to handle user sessions, data flow, and API interactions.
- Database: Firebase Firestore for handling interview data, user profiles, and session records.

4. AI-Driven Interview Simulation

- Allow users to choose job positions and industries to personalize the session.
- Dynamically create suitable questions using AI based on user choices.
- Permit text or voice responses, which are subsequently analysed with NLP for coherence, confidence, and technical correctness.

5. Real-Time Feedback and Scoring

- AI processes the response and gives immediate feedback on important parameters.
- Users are given customized recommendations for improvement in areas such as vocabulary usage, tone, technical content, and confidence.

6. Performance Tracking and Analytics

- Save each session's outcome and graph trends in the user's performance over time.
- Allow users to see strengths, determine repeating weaknesses, and redo sessions to enhance.

7. Testing and Validation

- Perform unit testing, integration testing, and user acceptance testing.
- Test the system's AI-provided feedback against expert-checked answers to verify quality and reliability.

8. Deployment and Maintenance

- Install the application on a cloud server with constant surveillance and upgrading.
- Maintain data privacy, user security, and scalability as user interest increases.

CHAPTER 2
FEASIBILITY STUDY

Prior to the development of any software system, it's crucial to evaluate if the project is feasible from different aspects. A feasibility study assists in establishing the viability of the project and ensuring that the suggested solution can be effectively designed, implemented, and maintained. This chapter analyses the **AI Mock Interview Web Application** based on technical, economic, operational, legal, and time-based feasibility.

The feasibility study aids the development team to know the limitations, resources, and possible risks of the project. It makes sure that the project is compatible with technological capability, cost, and organizational objectives, as well as legal and ethical standards.

2.1 Technical Feasibility

Technical feasibility is concerned with whether the existing technology stack, development tools, and capabilities are sufficient enough to develop and maintain the AI Mock Interview Web Application.

Key Considerations:

- **Suitability of Technology Stack:** The tech stack that is suggested—Next.js and Tailwind CSS for frontend development, Next.js with Firebase for backend computation is strong and popular. AI capabilities are driven by Google's Gemini API, and this facilitates easy integration of smart capabilities like question creation and NLP-driven analysis.
- **Resources Accessibility**: The software and platforms needed are open-source or commercially available. Cloud servers, contemporary web browsers, and development frameworks are all easily available.

- **Scalability and Performance:** With cloud hosting solutions, the application can be scaled according to user demand. The application of contemporary tools also guarantees high performance and minimal latency during real-time activities such as voice input processing through VAPI AI for instant feedback.
- **Ease of Integration:** The AI and NLP capabilities are API-based and modular, ensuring smooth integration with the core application. The APIs are capable of processing voice and text inputs, creating context-based questions, and assessing user responses.
- **Team Expertise:** The technical skills necessary to develop this project (web development, API integration, database schema, and fundamental AI functions) are widely possessed by today's developers. Therefore, the technical team is well equipped to effectively execute the development process.

Conclusion:

The project is technically feasible. Through existing and available modern technologies, and with an able development team, the AI Mock Interview app can be developed, deployed, and supported effectively.

2.2 Economical Feasibility

The economic feasibility looks at whether developing and maintaining the project is affordable based on the benefits that the project provides. In the **AI Mock Interview Web Application** case, there is high economic feasibility. It is cheap to develop, particularly with contemporary open-source technology, and the gains far surpass what it takes in terms of investments.

Key Considerations:

1. Development Costs

- The setup is based on universally available and free or inexpensive technologies, including React.js, Next.js, Tailwind CSS, and open-source APIs.
- AI services like Google's **Gemini API** can have usage-based charges, but tend to provide free tiers that are very generous for initial development or research work.

2. Hosting and Maintenance

- Hosting is possible using cloud platforms (such as Google Cloud, AWS, or Azure) with flexible pricing based on server usage.
- Maintenance is continuous server uptime, bug fixing, and periodic updates, which are low-cost and manageable.

3. Human Resources

- A small team with basic to intermediate web development and AI integration expertise can handle the entire system lifecycle.
- There is no requirement for costly or specialist AI engineers, so team expenses are low.

4. Scalability and User Reach

- One web-based application can support thousands of users without substantially raising costs, so it is very scalable.
- Once implemented, the platform can create long-term value with minimal monetary input.

5. Return on Investment (ROI)

- Saves customers money on expensive mock interview services or career consulting.
- Improves employment success rates, thus contributing indirectly to the financial security and employability of users.

Conclusion:

The AI Mock Interview system is viable economically. With negligible infrastructure investment and maximum usability potential, the system promises great dividends to both the user and developer in terms of a scalable and viable solution to preparing for interviews.

2.3 Operational Feasibility

Operational feasibility examines whether the system proposed addresses the problems identified to a great extent and to what extent it can be utilized effectively in a real-world setting. It considers whether the end users will embrace, utilize, and gain from the application when developed and implemented.

For the **AI Mock Interview Web Application**, operational feasibility is high, because the platform is designed to address a distinct and prevalent problem: the need for low-cost, low-friction, and effective interview preparation.

Key Considerations:

1. User Acceptance and Accessibility

- The platform is made to be straightforward, easy-to-use, and available to a wide audience comprising students, freshers, and professionals.
- Web-based interface reduces installation requirements, and one can utilize it on any device and from any location.

2. Enhanced Learning Outcomes

- Real-time feedback and dynamic simulation interviews offer instant feedback on performance.
- Customized recommendations drive ongoing improvement, making the app a worthwhile tool for learning.

3. Use Case Adaptability

- Accommodates a wide range of industries (e.g., IT, finance, healthcare) and professional roles.
- Both beginners and advanced professionals can use it by adjusting the difficulty level and question context.

4. Ease of Operation

- The users require only a stable internet connection and a modern browser to access the platform.
- Voice and text input provide flexibility, and performance tracking allows users to monitor progress.

5. Minimal Human Supervision Required

- The AI system is able to perform interviews, analyse responses, and give feedback without any need for manual interference.
- This keeps the operation burden low and makes the system scalable.

Conclusion:

The system is feasible from an operational perspective. It represents an effective, scalable, and user-oriented solution that improves interview preparation by enabling intelligent automation. Its ease of use and suitability to address the needs of the user ensures successful adoption and long-term sustainability.

2.4 Legal Feasibility

Legal feasibility is the process of determining if the suggested project adheres to all the relevant laws, regulations, and moral principles. Legal feasibility for the **AI Mock Interview Web Application** is critical so that user information is safe, intellectual property rights are upheld, and AI methods are open and responsible.

Key Considerations:

1. Data Privacy and Protection

- The system should adhere to data protection legislation like the General Data Protection Regulation (GDPR), particularly when dealing with user data from Europe.
- It should also adhere to India's Personal Data Protection Bill (once implemented) to maintain user privacy and safety of user data.
- Users should be made aware of how their data is stored, used, and collected, with terms and conditions and privacy policies clearly stated.

2. User Consent and Data Handling

- Explicit consent from users should be taken before storing responses or analysing performance.
- Adequate disclaimers should be given to ensure users that their data is not misused or shared.

3. Usage of AI and Open APIs

- AI services (e.g., Gemini API) are utilized in the project, and these have licensing and usage terms of their own. Care must be taken to ensure that their usage complies with the providers' terms.
- No reverse engineering, misuse, or resale of AI services for commercial purposes should take place without the right authorization.

4. Ownership of Content and Intellectual Property

- AI-generated interview questions and user answers could be under intellectual property concerns.
- The system must establish ownership rights on user-provided content and AI-generated content.

5. Non-Discriminatory Algorithms

- AI systems should be designed and tested to prevent gender, ethnic, language accent, or other sensitive factor-biased judgments.
- Proper ethical AI practices must be adopted to promote fairness and transparency.

Conclusion:

The AI Mock Interview Web Application is legally viable, as long as it includes proper privacy policies, ethical standards of AI usage, and adheres to local and global data protection legislation. By addressing these areas pre-emptively, the project can sidestep legal issues and gain user trust.

2.5 Timeline Feasibility

Timeline feasibility assesses if the project is feasible to be delivered within a reasonable and defined timeframe, given the scope, resources, and complexity. For the **AI Mock Interview Web Application**, the suggested timeline is realistic and well-structured, with opportunities for consistent progress throughout each phase of development.

The project duration is estimated to be 12 weeks, which is adequate for analysis, design, development, testing, and deployment.

Suggested Timeline Breakdown:

PHASE	TIME DURATION
Requirement Analysis	1 Weeks
Design Phase	2 Weeks
Development Phase	6 Weeks
Testing phase	2 Weeks
Deployment Phase	1 Weeks
Total Duration	12 Weeks

Supporting Factors:

- **Clear Task Division:** Each phase clearly has deliverables and can be performed in parallel by separate team members where feasible.
- **Moderate Complexity:** The utilization of APIs and pre-existing frameworks eliminates excessive workload and minimizes development.

Conclusion: The AI Mock Interview Web Application is within the suggested timeline and feasible. With effective planning and task delegation, the system can be developed and rolled out in full within 18 weeks, thus appropriate for academic, prototype, or startup-level implementation.

CHAPTER 3

PROJECT OBJECTIVE

For any software project, properly defining the objectives is an important step in guaranteeing successful development and deployment. The objectives give direction, lead the design and implementation stages, and assist in measuring success of the system upon completion. For the **AI Mock Interview Web Application**, the overarching goal is to provide an intelligent, scalable, and interactive system that helps individuals prepare for job interviews by leveraging Artificial Intelligence (AI) and Natural Language Processing (NLP)-powered realistic simulations.

The current chapter elucidates the large-scale vision of the project and discusses the important and specific goals driving its implementation.

3.1 Key Objectives

The **principal goals** of this project revolve around addressing actual life problems of job candidates in getting ready for interviews. Conventional mock interviews are in need of human mentors, and this may prove to be expensive, time-consuming, and challenging to scale. This project focuses on personalizing and automating the interview preparation with the aid of AI technology.

Core Objectives:

1. Emulate Real Interview Scenarios:

- Present users with profession-based questions and real-like interview flow.
- Imitate both technical and HR interviews in different industries.

2. Apply AI for Adaptive Interviewing:

- Dynamically create questions depending on user profile, job, and experience.
- Adapt complexity in real time according to the responses of the candidate.

3. Provide Instant, Personalized Feedback:

- Employ AI to evaluate verbal and written answers.
- Offer advice on clarity, relevance, confidence, and technical accuracy.

4. Monitor Progress and Performance:

- Provide metrics that demonstrate improvement over time.
- Direct users to improve their weak points with specific feedback.

5. Make It Scalable and Accessible:

- Create a web-based solution accessible to anyone who has an internet connection.
- Accommodate a large base of users from students and recent graduates to office workers.

3.2 Project Objectives

The project goals set the key objectives that dictate the development, functionality, and purpose of the **AI Mock Interview Web Application**. These goals are set to ensure the system provides an authentic, smart, and highly effective platform for interview practice. Every goal is intended to address certain issues that job seekers experience, including a lack of access to customized mock interviews, minimal feedback, and the requirement of scalable solutions.

1. To Create an Intelligent Mock Interview System

The main objective is to create an intelligent web application capable of mimicking real job interviews. The system should dynamically formulate interview questions depending on the chosen job category and experience level of the user. This system, unlike traditional question banks, has to change in real time, providing a different experience for every user. The simulation should be both technical and behavioural in style, replicating what candidates experience in real interviews.

2. For Integrating NLP and Artificial Intelligence

The app will make use of Artificial Intelligence and NLP to monitor the verbal and text responses from the user. The AI tool must be able to assess measures like clarity, coherence, grammar, confidence, voice tone, and technical correctness. This combination is key to the platform being able to provide automated, unprejudiced, and knowledgeable feedback—something lacking in customary mock interviews.

3. To Give Real-Time Feedback and Performance Assessment

One of the most distinctive features of this project is that it can provide real-time automated feedback. The system should rate the user after every question according to pre-set criteria and offer suggestions for improvement. This will make users adapt and rephrase their answers immediately, which will speed up their learning and make them more ready for real interviews.

4. To Create an Intuitive, Responsive User Interface

The platform will provide a smooth user experience. With React.js and contemporary UI/UX principles, the interface will be made clean, responsive, and intuitive. It should accommodate access from multiple devices—desktops, tablets, and smartphones—so that users can practice interviews anywhere, anytime.

5. To Accommodate Multi-Industry and Experience-Level Customization

The application needs to appeal to a broad set of users through customization by industry (e.g., IT, Finance, Healthcare, Engineering) and user type (e.g., student, fresher, experienced professional). This makes the platform useful and relevant to a broad set of users with diverse requirements.

6. To Track User Progress and Learning Trends

The system will need to remember session data and offer performance metrics to users. This entails tracking scores against previous scores, uncovering frequent errors, and underlining areas which require extra study. Users can be made to go back through past sessions to compare progress by encouraging continuous learning.

7. In Order to Have Security, Privacy, and Compliance

User information—such as responses, feedback, and personal details—needs to be stored and processed securely. The platform needs to comply with applicable data privacy legislation and best practices, providing users with transparency regarding how their data is processed.

8. To Build a Scalable and Maintainable Application

Lastly, the project should yield a system that is maintainable and scalable. Technologies such as Next.js, Tailwind CSS, Firebase, and cloud hosting like Vercel will enable long-term stability and future upgrades.

CHAPTER 4
HARDWARE AND SOFTWARE REQUIREMENTS

4.1 Hardware Requirements: Hardware specification is essential in order to assure smooth performance for development, testing, and deployment.

These details are classified below:

- **Development and Testing Environment**
    1. **Processor:** A Quad-Core processor (Intel i5 or AMD Ryzen 5) to be used for effective multitasking while developing.
    2. **RAM:** 8 GB of RAM as the minimum, but 16 GB for enhanced speed, particularly when multiple tools (IDE, browsers, and testing environments) are being used concurrently.
    3. **Storage:** 50 GB of available disk space, ideally on an SSD, for rapid data access and uninterrupted use.
    4. **Network:** A high-speed internet connection (at least 10 Mbps) is needed for real-time voice interactions and cloud access.
    5. **Audio Devices:** Good speakers and microphone are required for voice-based interview interactions.
- **Production Server Requirements**
    1. **Processor:** Multi-core processors (Intel Xeon or AMD EPYC) are advised for serving multiple users.
    2. **RAM:** 16 GB, scalable according to user needs.
    3. **Storage:** 100 GB SSD (expandable) for data storage and fast access.
    4. **Network:** High-speed internet (1 Gbps) for scalable, real-time communication.
    5. **Security:** SSL Certificates, Firewall Protection, and Cloud Infrastructure (AWS, Azure, or Google Cloud) for secure operation.

4.2 Software Requirements: The AI Mock Interview Platform's software stack is built for scalability, security, and effective development. It consists of:

- **Operating System:** Supports Windows 10/11, macOS, and Linux (Ubuntu 20.04+). This cross-platform compatibility provides ease for developers.
- **Development Tools**
    1. **IDE:** Visual Studio Code or WebStorm for code management with syntax highlighting along with inbuilt Git support.
    2. **Version Control:** Git for source code management, with GitHub as the collaborative development platform.
    3. **Deployment Platform:** Vercel for front-end deployment (Next.js) and Firebase Console for backend services.
- **Programming Languages**
    1. **JavaScript and TypeScript:** JavaScript for front-end development, while TypeScript ensures type safety for scalable projects.
- **Frameworks and Libraries**
    1. **Frontend:** Next.js (React-based) for dynamic user interface, Tailwind CSS for responsive design.
    2. **Backend:** Firebase for real-time database management and secure authentication.
    3. **Voice AI:** Vapi (Voice API) for real-time conversational voice interactions.
    4. **AI Module:** Gemini AI for customized interview question generation.
- **Database**
    1. **Firebase Firestore (NoSQL):** Highly scalable, real-time user data, interview question, and performance analytics database.
- **Security and Encryption**
    1. **SSL/TLS Encryption:** Facilitates secure data transfer.
    2. **Firebase Security Rules:** Manages user data access, ensuring privacy.
    3. **Data Encryption:** Encrypts sensitive user data and interview history.

CHAPTER 5
PROJECT FLOW

5.1 Problem Identification

With the current digital-first job market, the necessity for organized, role-based interview preparation has never been more important. Yet, conventional approaches like face-to-face mock interviews, coaching classes, and peer sessions have a number of drawbacks:

- **Accessibility Issues:** Not all individuals can afford experienced mentors or professional coaching services, especially students in remote or underserved communities.
- **Steep Price:** Custom coaching and HR advice is costly, not affordable for many job seekers.
- **Unchanging Learning Architectures:** All but a handful of existing web-based mock interviewing systems provide inflexible question lists that don't change based on user performance, profession, or level of experience.
- **Insufficient Live Feedback:** There is often little or no instantaneous, actionable feedback. Users never know what went wrong or how to correct their mistakes.
- **Limited User Roles:** Solutions tend to equal all users when they are not a fresher, an experience professional, or a career-switcher.

This project aims at creating a solution that addresses all these limitations with the introduction of an AI-driven mock interview web application that is inclusive, scalable, smart, and personalized to every user's need.

5.2 Requirement Analysis

An in-depth requirement analysis was done to make sure that the system is catering to users' needs as well as being technically feasible and easy to use. The requirements are categorized under functional and non-functional.

Functional Requirements:

The system is supposed to accommodate the following main functions:

- User Registration & Login: Users must be able to register, log in, and manage their profiles safely.
- Role Selection: When users register, they should choose their role—Job Seeker, Student, or Professional—to allow the interview flow to be customized accordingly.
- Industry & Job Type Selection: Users should be able to choose their industry (e.g., IT, Finance, Healthcare) and job type (e.g., Developer, Analyst, Manager).
- Dynamic Question Generation: The system should employ AI in generating questions from chosen job roles and experience levels.
- Input Modes: Users must have the option of responding to interview questions through voice or text.
- AI Evaluation: User responses must be evaluated by analysing them using NLP methods as a measure for clarity, tone, confidence level, and correctness.
- Progress Tracking: Users must see analytics reflecting the trends in performance, repeated errors, and skills to improve upon.

Non-Functional Requirements:

- Scalability: The system should support 100+ simultaneous users without slowing down.
- Responsiveness: The interface needs to adjust to various device types and screen sizes (mobile, tablet, desktop).
- Security: Secure login protocols like JWT (JSON Web Tokens) and encryption of data must be used.

- Performance: The system needs to react to user actions and provide feedback in 1–2 seconds for a delightful experience.
- Accuracy: Evaluations done by AI should have an accuracy of no less than 85% with respect to human evaluators.

5.3 Design Phase

The design phase concentrates on organizing the whole application flow—both technically and visually—to make sure that users enjoy a seamless, intuitive experience while preserving system integrity and performance.

A. User Interface Design:

- Built using React.js to enable a dynamic and responsive UI.
- Interfaces differ according to user roles to customize the experience.
- Important screens are:
- Login/Signup Page
- Role & Industry Selection
- Interview Dashboard
- Response Input Page
- Feedback & Analytics Page
- Utilization of icons, color-coded feedback, and charts to present insights in a visually appealing manner.

B. System Architecture:

- The architecture is of a client-server model, segregated into:
- Frontend (React.js): Manages user interface and interaction.
- Backend (Next.js + Firebase): Handles server-side logic, routing, and API interactions.
- Database (Firebase Firestore(NoSQL)): Saves user information, responses, scores, and feedback history.

- AI Engine (Gemini API and VAPI AI): Handles questions and analyses answers through API calls.

C. Database Modelling:

- Important entities: Users, Interview Sessions, Questions, Responses, and Feedback.
- Relations:
- A User may have multiple Interview Sessions.
- Every Interview Session is associated with several Questions and Responses.
- Data is normalized to provide quick querying and low redundancy.

D. Security Design:

- Passwords are hashed using hashing algorithms.
- Role-based access control (RBAC) provides users with access to only relevant data.
- Input validation avoids injection attacks and maintains data integrity.

5.4 Technology Selection

Selecting the appropriate technologies was instrumental to the success of the AI Mock Interview Web Application. The system needed to be reliable, responsive, scalable, and able to facilitate AI-processing in real-time. On these bases, the technology stack was chosen.

A. Frontend Technologies

- Next.js: Employed for constructing the user interface. It offers:
  - Component-based architecture for reusability.
  - Fast rendering with virtual DOM.
  - A responsive layout that works for mobile and desktop devices.
- Tailwind CSS: Facilitates rapid styling with utility-first CSS classes.

B. Backend Technologies

- Next.js: Robust JavaScript runtime environment for developing scalable backend services.
- Firebase: Firebase for real-time database management and secure authentication.

C. Database Management

- Firebase Firestore(NoSQL): Highly scalable, real-time user data, interview question, and performance analytics database. It is used to store:
  - User login credentials.
  - History of interview sessions.
  - AI feedback and analytics information.

D. AI/NLP Engine

- Gemini API by Google AI (or GPT-4 by OpenAI): The core NLP model responsible for:
  - Generating role-specific interview questions.

- Analysing user responses for clarity, coherence, tone, and relevance.

- Providing detailed, real-time feedback.

- The AI is integrated via RESTful API calls and supports dynamic prompt handling based on user selections.

E. Security & Encryption

- **SSL/TLS Encryption:** Facilitates secure data transfer.
- **Firebase Security Rules:** Manages user data access, ensuring privacy.
- **Data Encryption:** Encrypts sensitive user data and interview history.

F. Hosting and Deployment

- Vercel: For hosting the frontend (React-based Single Page Application).
- Firebase (optional): For real-time functionality and analytics if required in future releases.

The selected tech stack provides quick performance, security, and the capacity to add more features such as video analysis or real-time chat support in the future.

5.5 Development

Development of the AI Mock Interview Web App was carried out in iterative cycles to allow for ongoing testing and feedback. The primary steps of the development life cycle are as follows:

A. Phase 1 – Environment Setup

- Install local development environment with:
- IDE: Visual Studio Code or WebStorm for code management with syntax highlighting along with inbuilt Git support.
- Next.js (React-based) for dynamic user interface, Tailwind CSS for responsive design.
- Firebase for real-time database management and secure authentication.

B. Phase 2 – User Authentication & Role Management

- Developed registration and login pages with role-based redirection.
- Implemented JWT-based secure authentication.
- Established three primary roles:
  - Job Seeker
  - Student
  - Professional
- Each role was provided with customized dashboard access and interview options.

C. Phase 3 – Interview Session Integration

- Developed interfaces for:
  - Choosing industry and job role.
  - Starting mock interview session.
- Developed API endpoints to:
  - Call the AI model and dynamically create questions.
  - Receive and cache user answers (text or voice).
  - Retrieve AI feedback and show scores.

D. Phase 4 – AI Evaluation Integration

- Integrated Gemini or GPT-4 APIs to evaluate answers and provide:
  - Clarity, confidence, and relevance scores.
  - Improvement suggestions.
- Employed asynchronous API calls to keep the app responsive when waiting for results generated by AI.

E. Phase 5 – Feedback & Analytics

- Created a dashboard to display:
  - Session scores for individual sessions.
  - Graphical charts of performance trends.
  - Most common errors and tips for improvement.
- Made it possible to filter performance history by:
  - Date
  - Type of job
  - Score type (clarity, content, etc.)

F. Phase 6 – Responsive Design & UX Improvements

- Made mobile compatibility a guarantee.
- Included loading indicators, toast notifications, and error management.
- Incorporated Framer Motion animations for better transitions and interactions.

G. Phase 7 – Testing and Refinement

- Employed unit tests to guarantee frontend and backend module reliability.
- Manually tested every user flow.
- Optimized API performance and reduced response latency.

5.6 Testing and Validation

Testing is an important step in the software development life cycle. It verifies that the system works correctly, securely, and reliably in various scenarios and user roles. The AI Mock Interview Web Application went through various testing approaches, using manual and automated testing to confirm its behaviour from both functional and non-functional aspects.

A. Unit Testing

- Centred on standalone modules like:
  - User registration and login.
  - API endpoints for interview generation.
  - AI feedback and scoring evaluation.
- Tools utilized:
  - Jest for testing JavaScript.
  - Postman for testing backend APIs.

B. Integration Testing

- Verified the integration between:
  - Frontend elements and backend APIs.
  - AI response processing and user sessions.
- Verified that:
  - AI-generated questions were displayed correctly.
  - User-submitted responses were correctly analysed and saved.

C. Functional Testing

- Confirmed the system's functionality against the specified requirements:
  - Dynamic question generation.
  - Role-based navigation.
  - AI feedback integration.

  - Analytics dashboard display.
- All user flows (Student, Job Seeker, Professional) were separately tested to ensure that role-based features functioned as expected.

D. User Acceptance Testing (UAT)

Performed using real users (students and colleagues).

- Got good feedback on:
  - Ease of interface.
  - Realism in quality of AI-generated questions.
  - Utility of the feedback provided.
- Improved the following based on user feedback:
  - Improve handling of voice inputs.
  - Offering more descriptive improvement hints.

E. Validation of AI Feedback

- Compared AI feedback outcomes with expert ratings.
- Made sure that:
  - Scoring of "confidence" and "clarity" aligned with 85% of human evaluations.
  - Correctly identified content relevance, particularly for technical positions.
- Adjusted AI prompts and parameters to optimize model output.

F. Performance Testing

- Evaluated system behaviour under load:
  - Simulated multiple users creating sessions simultaneously.
  - Guaranteed backend API response time at <2s.
- Optimized database queries and front-end rendering to minimize lag.

G. Bug Tracking and Resolution

- Issues were tracked with Trello and GitHub Projects.
- Common bugs included:
  - Delay in rendering feedback.
  - AI returning overly generic advice.
- All critical bugs were fixed before deployment.

5.7 Authentication Flow:

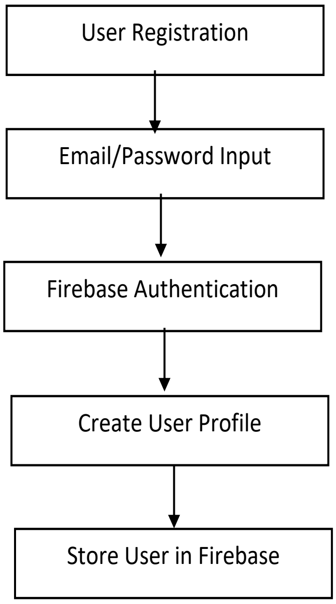


Fig. 5.7.1 Authentication Flow

Explanation of the Flow:

1. **User Registration:**

   The user starts the registration process, typically by giving their details.

2. **Email/Password Input:**

   The user is asked to input their email and password for account creation.

3. **Firebase Authentication:**
    - Firebase Authentication service is utilized to authenticate the user's credentials (email and password).
    - If the credentials are correct, the user is authenticated.
4. **Create User Profile:**
    - Upon successful authentication, a user profile is created.
    - The profile can hold extra user details (such as name, role, or preferences).
5. **Store User in Firebase:**
   The credentials and profile of the user are safely stored in Firebase, enabling them to log in subsequently.

How it relates to your AI Mock Interview (PrepWise) Project:

- You will employ this authentication process to enable the registration, login, and safe access to the mock interview system.
- Once you have registered the users, their profiles will be saved, and it will be simple to personalize the interview experience for every user.
- This authentication configuration ensures that user information is protected and only allows authenticated users to access the mock interview functionality.

5.8 Interview Workflow:

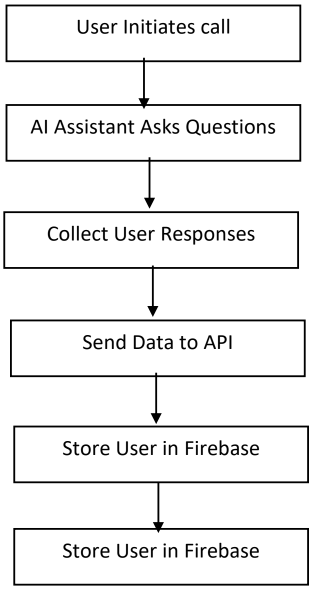


Fig. 5.8.1 Interview Workflow

Workflow Explanation:

1. **User Requests Call:**
   The user initiates the mock interview session, either voice or text-based.
2. **AI Assistant Questions:**
   The AI mock interview program starts by querying predefined or dynamically created interview questions.
   These are derived from the user's chosen interview type or skill category.
3. **Gather User Responses:**
   The responses from the interview questions of the user are gathered.
   They may be in text, audio, or video forms based on your implementation.
4. **Send Data to API:**
   The gathered responses are transmitted to an API for processing.
   Such an API may process the responses with Natural Language Processing (NLP) to generate feedback.

5. **Save User in Firebase (First Block):**
   The interview data of the user (questions and answers) is saved in Firebase for analysis and tracking.
   This keeps each user's history of performance saved.
6. **Save User in Firebase (Second Block - Duplicate):**
   This appears to be a duplicate in the process. You ideally save user data only once in Firebase.
   Or, equivalently, keeping user responses or analysis results after the interview.

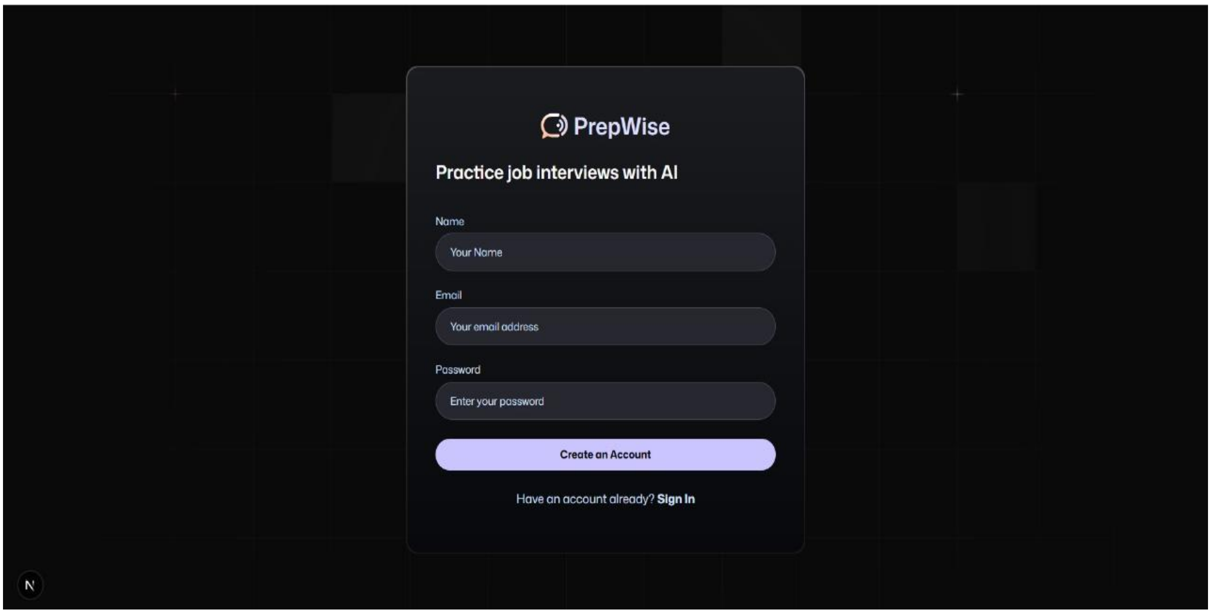
How it Relates to Your AI Mock Interview (PrepWise) Project:

- This process illustrates how your mock interview program will function in real-time.
- It addresses the entire process from initiating an interview to keeping user responses and results.
- You will have to apply API integration in order to process the responses of users (such as creating feedback or grading).
- Firebase will serve as your database for user profile management, interview history, and performance monitoring.

CHAPTER 6

PROJECT OUTCOME

6.1 Sign-Up:



PrepWise

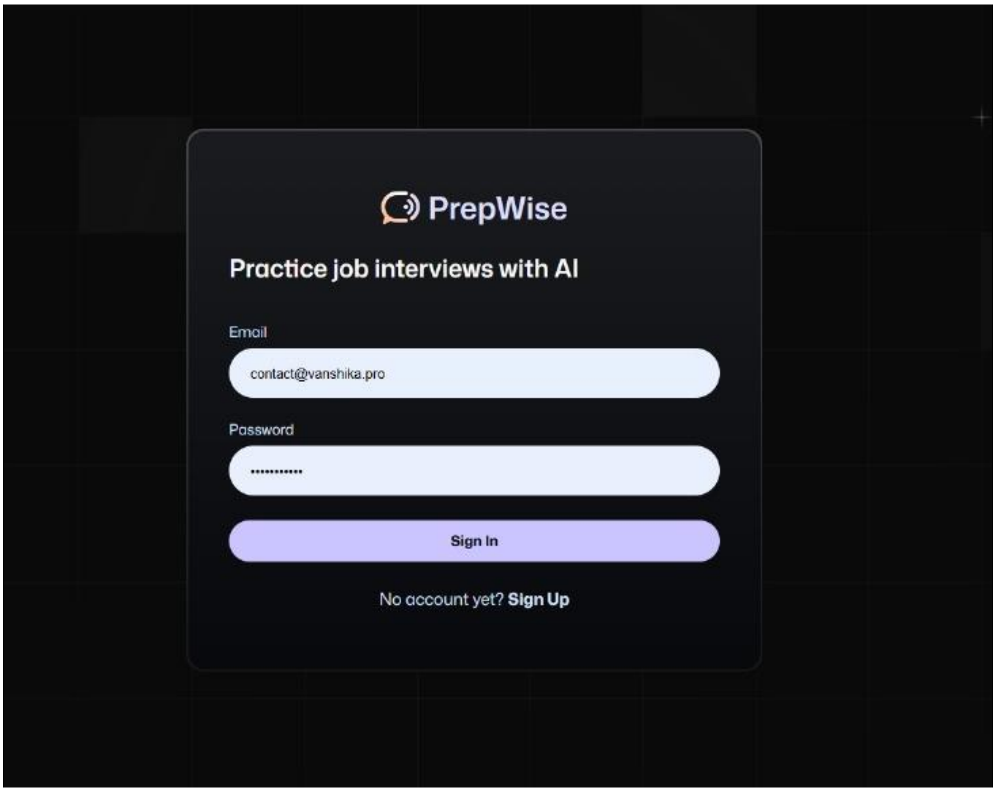
Practice job interviews with AI

Have an account already? Sign In

Fig. 6.1 Sign Up

- This is a **registration screen** of an AI-powered interview preparation platform called **PrepWise**.
- It contains form fields to accept the user's **Name, Email, and Password** in order to have a new account.
- There is a purple **"Create an Account"** button to save the form.
  At the bottom, it offers a link to **Sign In** in case the user already has an account.

6.2 Sign in:



PrepWise

Practice job interviews with AI

Email

contact@vanshika.pro

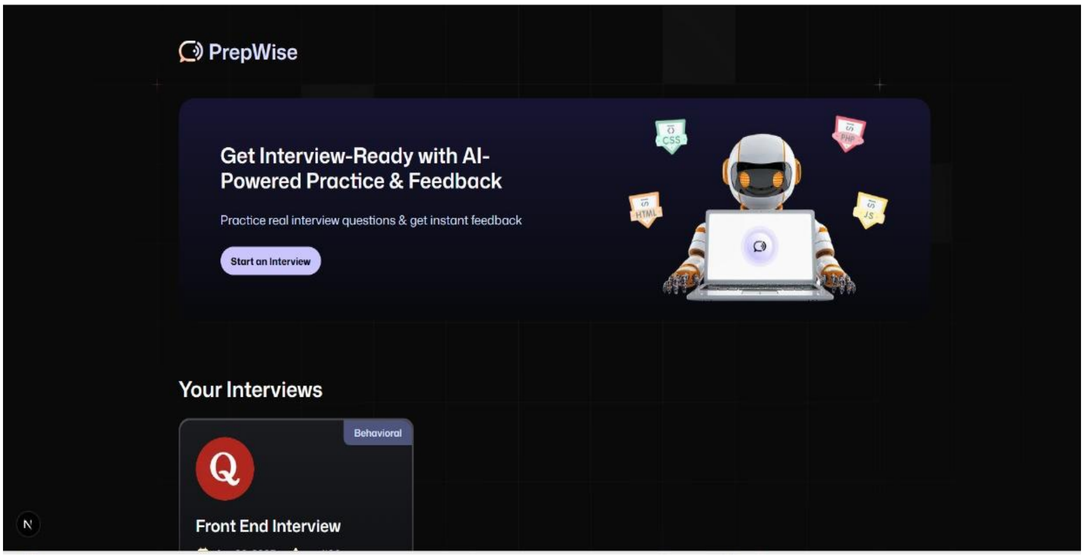
Password

•••••••••••

Fig. 6.2 Sign in

- This picture shows the **login screen** of PrepWise, an AI-based interview practice platform.
- It has spaces for users to input their **Email and Password** to log in to their account.
- A **"Sign In"** purple button enables users to sign into the system.
- Under the button, there is also a link for new users to Sign Up in case they don't yet have an account.

6.3 Home Page:



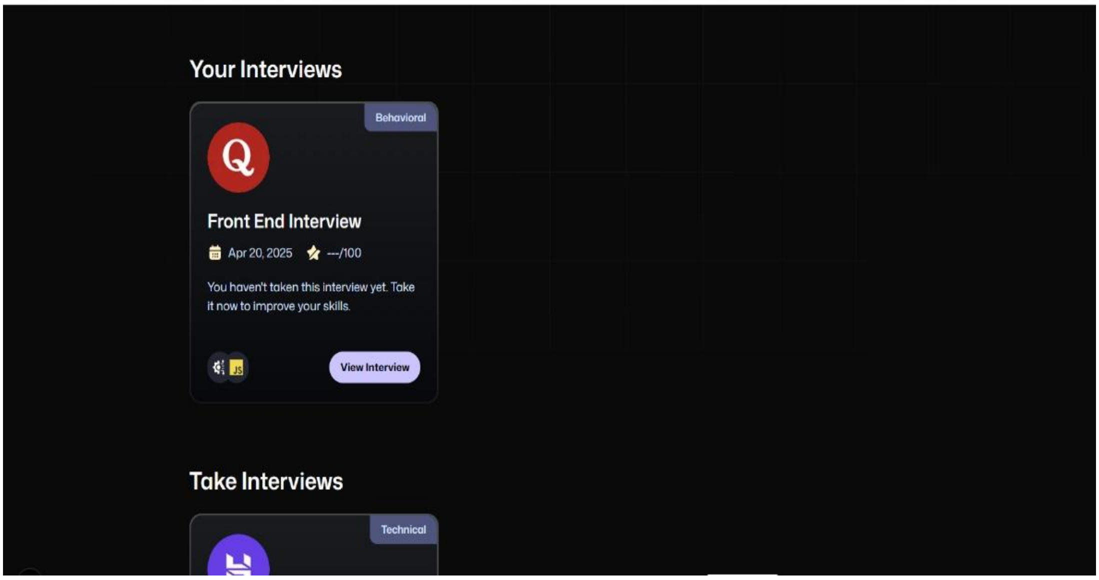
Get Interview-Ready with AI-Powered Practice & Feedback

Practice real interview questions & get instant feedback

Your Interviews

Front End Interview

Fig. 6.3.1 Home Page



Your Interviews

Front End Interview

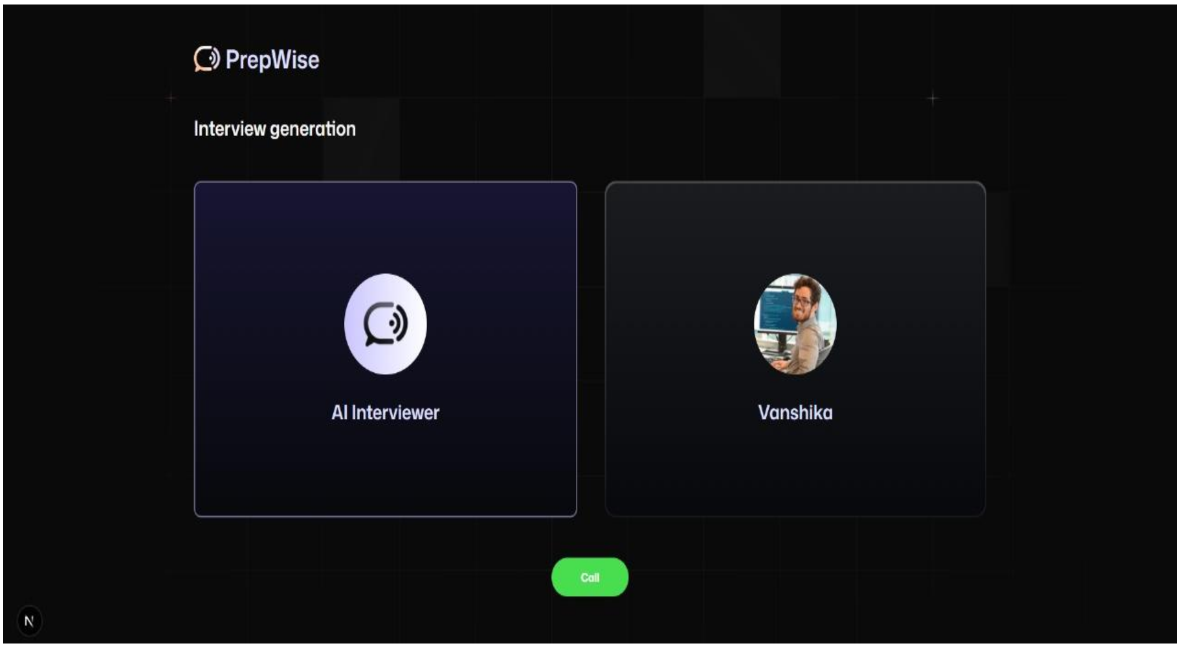
Apr 20, 2025 ---/100

You haven't taken this interview yet. Take it now to improve your skills.

Take Interviews

Fig. 6.3.2 Home Page

6.4 Interview generation screen:

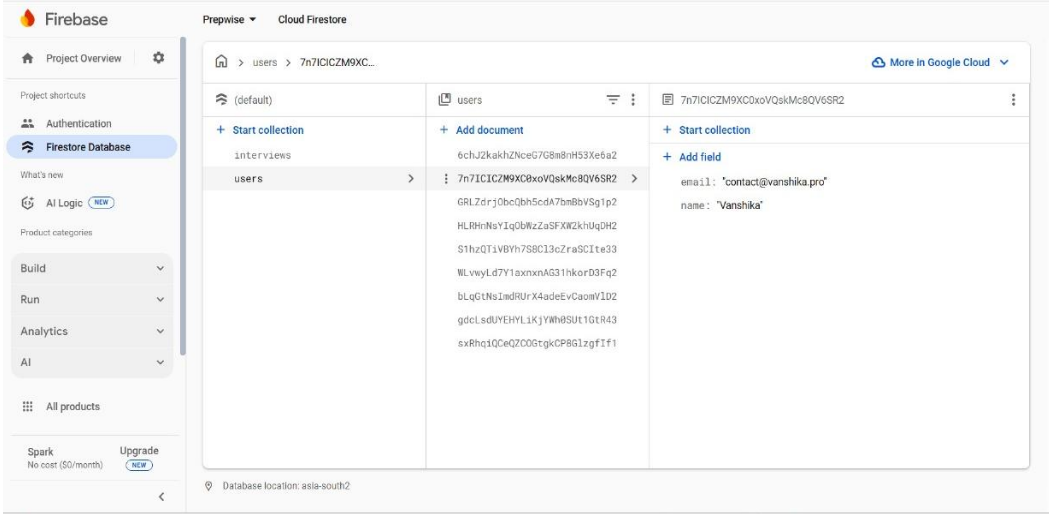

PrepWise

Interview generation

Fig. 6.4 Interview generation screen

- This indicates the interview generation screen on the PrepWise site.
- To the left is an **AI Interviewer** symbol to denote the virtual interviewer.
- At the bottom is a green "Call" button that initiates the simulated interview between the user and the AI.

6.5 Firebase Firestore Users



(default): interviews, users

users:

```
6chJ2kakhZNceG7G8m8nH53Xe6a2
7n7ICICZM9XC0xoVQskMc8QV6SR2
GRLZdrjObcQbh5cdA7bmBbVSg1p2
HLRHnNsYIqObWzZaSFXW2khUqDH2
S1hzQTiVBYh7S8Cl3cZraSCIte33
WLvwyLd7Y1axnxnAG31hkorD3Fq2
bLqGtNsImdRUrX4adeEvCaomVlD2
gdcLsdUYEHYLiKjYWh0SUt1GtR43
sxRhqiQCeQZCOGtgkCP8GlzgfIf1
```

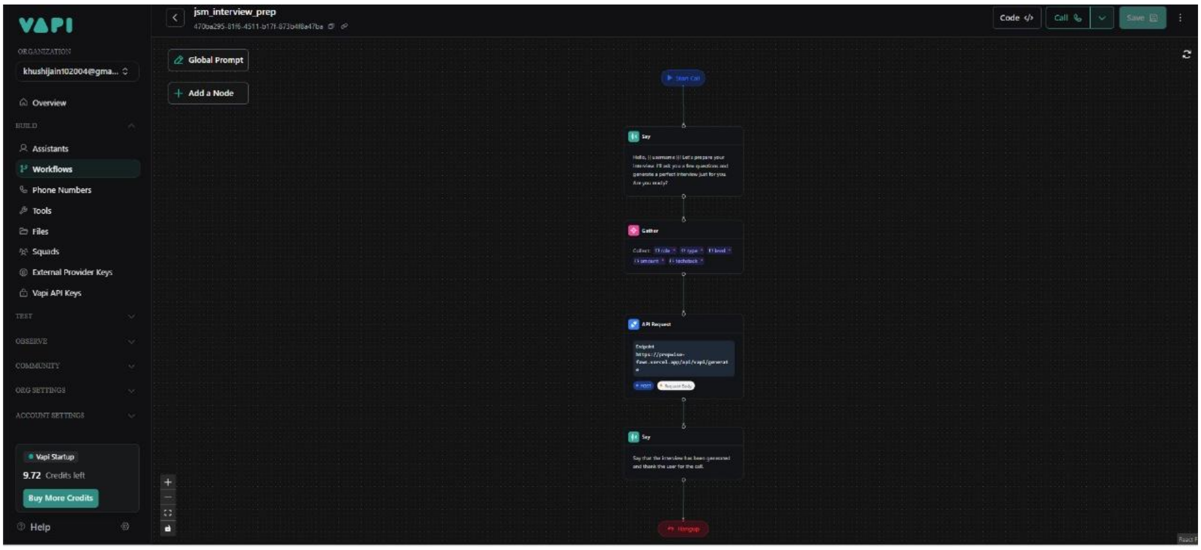
7n7ICICZM9XC0xoVQskMc8QV6SR2:

```
email: "contact@vanshika.pro"
name: "Vanshika"
```

Fig. 6.5 Firebase Firestore Users

- This Firebase Firestore page shows the users collection for your **AI Mock Interview** project. A specific user document (7n7ICICZM9XC...) contains fields like email and name.
- The user's name is "Vanshika" with the email contact@vanshika.pro.
- You can store more user data or subcollections (e.g., interviews) under each user document.

6.6 Vapi AI Voice Assistant



Hello, {{ username }}! Let's prepare your interview. I'll ask you a few questions and generate a perfect interview just for you. Are you ready?

Collect: {} role {} type {} level {} amount {} techstack

Endpoint https://prepwise-fawn.vercel.app/api/vapi/generate

POST Request Body

Say that the interview has been generated and thank the user for the call.

Fig. 6.5 Vapi AI

- This Vapi dashboard shows a workflow named jsm_interview_prep for your **ai_mock_interview project**.
- It starts a call, asks the user if they're ready, and gathers input like domain or level.
- Then, it sends an API request to generate interview questions.
- Finally, it confirms that the interview has been generated and ends the call.

6.7 Graph Firebase

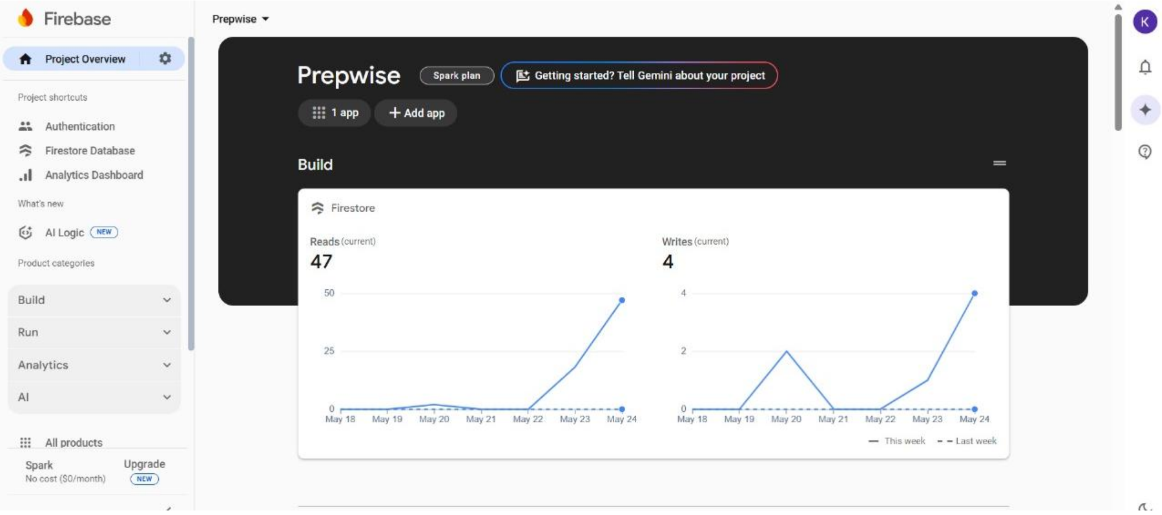
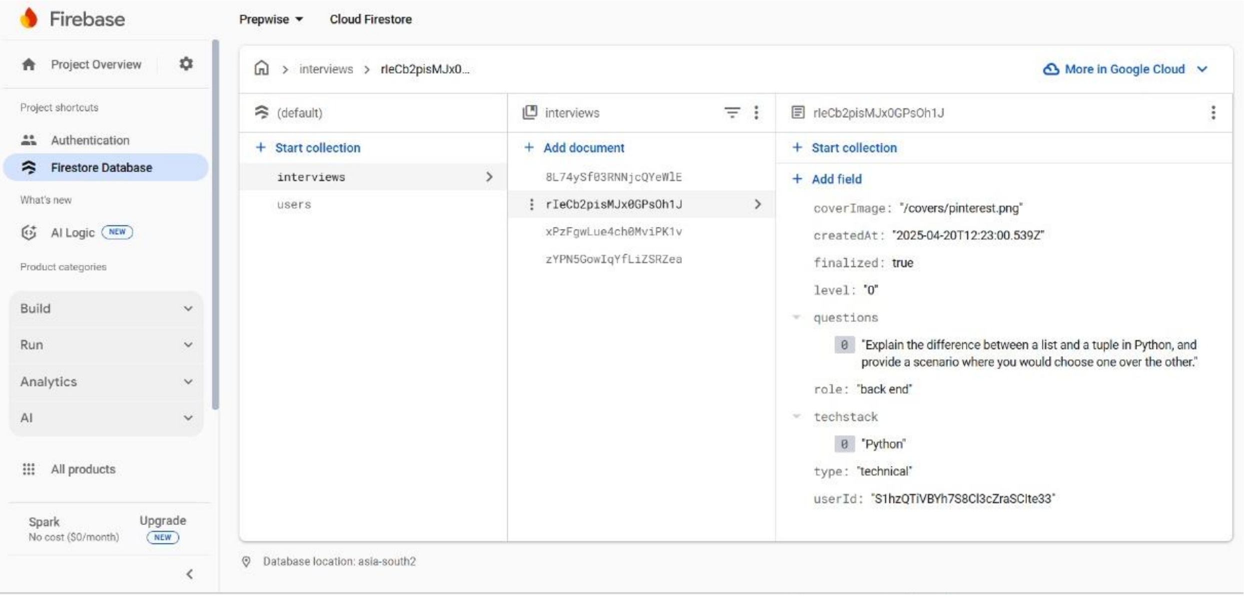


Fig. 6.5 Graph Firebase

- This Firebase dashboard shows the **AI Mock Interview** project activity for your **ai_mock_interview** system.
- It tracks Firestore usage with 47 reads and 4 writes, mostly on May 23–24.
- This indicates user interactions like storing and retrieving data.
- The project is currently on the Spark (free) plan.

6.8 Firebase Interview Database



interviews
users

8L74ySf03RNNjcQYeWlE
rIeCb2pisMJx0GPsOh1J
xPzFgwLue4ch0MviPK1v
zYPN5GowIqYfLiZSRZea

coverImage: "/covers/pinterest.png"
createdAt: "2025-04-20T12:23:00.539Z"
finalized: true
level: "0"
questions
0 "Explain the difference between a list and a tuple in Python, and provide a scenario where you would choose one over the other."
role: "back end"
techstack
0 "Python"
type: "technical"
userId: "S1hzQTiVBYh7S8Cl3cZraSCIte33"

Database location: asia-south2

Fig. 6.5 Firebase Interview Database

- This Firebase Firestore page shows interview data stored for your **AI Mock Interview** project.
- It contains fields like questions, role, techstack, type, and userID, indicating a technical backend interview.
- The document also logs metadata like createdAt, finalized, and level.
- This structure supports dynamically fetching and displaying interview sets per user.

6.9 Below are the key outcomes of the project:

1. **Development of AI-Based Mock Interview System**
   - A smart system was developed to simulate real interview scenarios using AI.
   - It can ask interview questions and analyse user responses.
2. **Realistic Interview Experience**
   - The system provides an experience similar to actual interviews.
   - It assists in making users less fearful and more confident via practice.
3. **Speech and Text Input Support**
   - The user can answer questions using voice or text.
   - This makes the system flexible and easy to use.
4. **Evaluation and Feedback Generation**
   - The system provides immediate feedback after every session.
   - It evaluates the user responses on correctness, confidence, and clarity.
5. **Suggested Personalized Improvement**
   - The application provides improvement areas such as communication, subject, or body language (in case camera support is activated).
   - This focuses users on the areas of improvement.
6. **Question Bank and Role-Based Interviews**
   - It has a broad set of questions based on job titles (e.g., Developer, Designer, Analyst).
   - Users can choose the interview type they wish to practice.

7. Enhances Communication and Interview Skills

- Practice on the system on a regular basis improves speaking skills and creates a good introduction.
- It also assists users in formatting their responses in the correct manner using the STAR method (Situation, Task, Action, Result).

8. Recording of Data and Tracking Progress

- The system is able to store performance data after every session.
- Users are able to monitor improvement over time.

9. Simple to Use and Accessible

- The interface is easy to use so that anyone can use it without technical expertise.
- It is usable on laptops and mobile phones.

10. Overall Impact

- The project effectively offers a smart, affordable, and accessible solution for interview preparation.
- It enables students and job seekers to prepare well for actual interviews confidently.

REFERENCES

Web & Tools:

- **Vercel Inc. (2024).** Vercel Documentation – Deployment and Hosting. Retrieved from https://vercel.com
- **Firebase (Google). (2024).** Firebase Documentation – Realtime Database, Authentication, and Security. Retrieved from https://console.firebase.google.com
- **GitHub Inc. (2024).** GitHub Documentation – Version Control and Collaboration. Retrieved from https://github.com
- **VAPI (Voice API). (2024).** Voice API Documentation – Real-time Voice Interaction. Retrieved from https://dashboard.vapi.ai/workflows
- **Gemini AI. (2024).** Gemini AI Documentation – Custom AI Model Integration. Retrieved from https://ai.google.dev/gemini-api/docs/api-key
- **W3Schools**. JavaScript Tutorial. [Online]. Available: https://www.w3schools.com/js/DEFAULT.asp.
- **MDN Web Docs.** (2024). Web APIs and JavaScript Reference. Retrieved from https://developer.mozilla.org/en-US/docs/Web/JavaScript

Books on AI, NLP, and ML:

- Jurafsky, D., & Martin, J. H. (2023). Speech and Language Processing (3rd ed.). Pearson.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- Russell, S., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach (4th ed.). Pearson.
- Tunstall, L., von Werra, L., & Wolf, T. (2022). Natural Language Processing with Transformers. O'Reilly Media.
- Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O’Reilly.
- Shedroff, N. (2019). Voicebot and Chatbot Design. Rosenfeld Media.

Books on Software Engineering & Project Management:

- Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.
- Pressman, R. S. (2019). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill.
- McConnell, S. (2004). Code Complete: A Practical Handbook of Software Construction (2nd ed.). Microsoft Press.
- Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley.

Books on Web, Firebase & UI Design:

- East, D. (2020). Full Stack Firebase. Leanpub.
- Duckett, J. (2011). HTML and CSS: Design and Build Websites. Wiley.
- Duckett, J. (2014). JavaScript and jQuery: Interactive Front-End Web Development. Wiley.
- Krug, S. (2014). Don’t Make Me Think, Revisited: A Common Sense Approach to Web Usability. New Riders.