

Interview Task: AI-Powered Interview Preparation Platform with Firebase Authentication and JWT Integration

Task Overview

Develop an **AI-driven interview preparation platform** where users can:

1. **Practice Mock Interviews** – Users can select a domain (Tech, HR, Finance, etc.) and get AI-generated questions.
2. **Receive AI Feedback** – The AI evaluates their responses and suggests improvements.
3. **Track Performance Metrics** – Users receive a **scorecard with strengths and weaknesses**.
4. **User Authentication** – Secure access using Firebase Authentication and JWT.
5. **Admin Dashboard** – Admins can track user progress and popular interview topics (optional).

This project integrates **AI-based real-time feedback, Firebase authentication, JWT security, and interactive analytics** to enhance **interview preparation**.

Requirements

Backend (Node.js + Firebase + OpenAI API)

1. **Firestore Setup:**
 - Initialize Firebase Admin SDK for backend operations.
 - Set up Firebase Authentication for user registration and login.
 - Use Firebase Firestore (or Realtime Database) for storing:
 - **User Sessions** (interview attempts, scores, feedback).
 - **Question Bank** (pre-defined and AI-generated questions).
2. **AI-Powered Interview Simulation:**
 - Generate **domain-specific questions** dynamically using **OpenAI API (GPT-4)**.
 - Evaluate **text-based** or **voice-based** responses.
 - Provide **structured feedback** based on clarity, confidence, relevance, and correctness.

3. Authentication:

- Implement user signup and login using Firebase Authentication.
- Generate and manage JWT tokens for secure API requests.

4. API Endpoints:

- **POST /signup**: Register a new user with Firebase Authentication.
- **POST /login**: Authenticate users and issue JWT tokens.
- **POST /interview/start**: Allow users to start a mock interview session.
- **POST /interview/answer**: Accept user answers for AI evaluation.
- **GET /interview/results/:id**: Fetch AI-generated feedback and scoring.
- **GET /interview/history**: Retrieve past interview attempts for authenticated users.
- **GET /admin/insights**: Allow admins to analyze interview trends (optional).

5. JWT Integration:

- Middleware to verify JWT tokens for **secure API access**.
 - Users should **only access their own interview sessions**.
 - Admins should have **aggregated insights without accessing individual user answers**.
-

Frontend (React.js or any frontend framework)

1. User Authentication:

- **Signup Form**: Register new users.
- **Login Form**: Authenticate users.
- **Token Storage**: Securely store JWT tokens in **localStorage** or cookies.

2. Mock Interview UI:

- **Select Interview Domain** (Tech, HR, Marketing, etc.).
- **Receive Questions in Real-Time** (via API).
- **Answer by Typing or Speaking** (voice-based support optional).
- **Submit Answers for AI Review**.

3. Performance Dashboard:

- **Show Scores & Feedback**: Users can view AI-generated insights.
- **Weakness Analysis**: Highlight areas for improvement.
- **Past Performance Tracking**: Allow users to see progress over time.

4. Admin Interface (Optional):

- Provide an **admin dashboard** to track popular topics and user performance trends.

5. **Authorization:**

- Ensure JWT tokens are included in API requests for security.
 - Restrict users to managing their own interview attempts.
-

Bonus (Optional Enhancements)

1. **Advanced Features:**

- **Real-Time Speech Analysis:** Use **Whisper API** to evaluate speech clarity.
- **Job-Specific Mode:** Customize mock interviews based on **company roles**.
- **AI Resume Review Integration:** Provide instant resume feedback before interviews.
- **Video Mock Interviews:** Allow users to record responses for self-evaluation.

2. **User Experience Enhancements:**

- **Dark Mode UI:** Improve readability for users.
 - **Gamification:** Award badges for milestones (e.g., 10 mock interviews completed).
 - **Community Feature:** Allow users to discuss common interview challenges.
-

Expected Deliverables

1. **GitHub Repository:**

- A repository with:
 - Full source code for **backend and frontend**.
 - A **README.md** file with setup instructions.

2. **Live URL:**

- A deployed version of the platform (using Vercel, Netlify, or Firebase Hosting).

3. **Functionality:**

- Ensure JWT token verification for secure interview management.
-

Important Notes

- Ensure **user privacy and data security** for mock interviews.
- Users should **only access their own interviews and AI feedback**.
- **Admins should have access** to aggregate interview analytics but not individual responses.
- Store JWT tokens securely and handle authentication errors properly.