**1. Project Title and Scope**

**Project Title: Interview Preparation Guide (MERN Stack)**

**Project Scope:**

This project aims to create a **dynamic, role-based interview preparation platform** that provides users with structured learning paths, quizzes, mock interviews, and progress tracking. It will help students and professionals systematically prepare for job interviews in various technology roles.

**User Requirements:**

**1. Students/Users**

* Sign up and log in to access personalized content.
* Select a career roadmap (Frontend, Backend, Full Stack, etc.).
* Access topic-wise interview questions and practice problems.
* Take quizzes and mock interviews.
* Track learning progress and completed tasks.

**2. Admins**

* Manage roadmaps (add, update, delete steps).
* Upload and manage interview questions & quizzes.
* Monitor user progress and performance.

**Module Specifications:**

**1. User Authentication Module**

* Signup/Login (JWT authentication).
* Role-based access (Admin, Student).

**2. Roadmap Module**

* learning paths based on technology roles.

**3. Question Bank & Practice Problems**

* Organized topic-wise (DSA, CS Fundamentals, etc.).
* Solutions and explanations included.

**4. Quizzes & Assessments**

* Multiple-choice quizzes with instant scoring.
* Performance tracking for users.

**5. Aptitude Module:**

Aptitude practice for users

**6. Mock Interview Module**

* HR-based interview question bank.
* Basic chatbot for text-based Q&A.

**7. Progress Tracking Module**

* User dashboard to track roadmap and quiz progress.

**8. Admin Panel**

* Add/update roadmaps, questions, and quizzes.

**Technologies to be Used:**

**Frontend:**

* **React.js** (Component-based UI).
* **CSS** (Styling).

**Backend:**

* **Node.js & Express.js** (Server-side API).
* **MongoDB (Mongoose)** (Database for dynamic content storage).

**Other Tools & Libraries:**

* **JWT & bcrypt.js** (Authentication & security).
* **Axios** (API calls).

**Data Dictionary**

**1. User Collection (users)**

Goods user details and authentication data.

| **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- |
| User | \_id | ObjectId | Primary Key, Auto-generated |
|  | name | String | Required |
|  | email | String | Unique, Required |
|  | password | String | Required, Hashed |
|  | role | String | Enum: admin, student |
|  | roadmapProgress | Array | Stores completed roadmap steps |
|  | quizScores | Array | Stores user's quiz results |
|  | createdAt | Date | Auto-generated Timestamp |
|  | updatedAt | Date | Auto-updated on modification |

**2. Roadmap Collection (roadmaps)**

Stores different learning paths based on technology roles.

| **Sr. No.** | **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- | --- |
| 1 | Roadmap | \_id | ObjectId | Primary Key, Auto-generated |
| 2 |  | role | String | Required (e.g., Frontend Developer) |
| 3 |  | title | String | Required |
| 4 |  | description | String | Optional |
| 5 |  | steps | Array | Ordered list of learning steps |
| 6 |  | resources | Array | Links to articles, videos, etc. |

**3. Question Bank Collection (questions)**

Stores practice questions for different topics.

| **Sr. No.** | **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- | --- |
| 1 | Question | \_id | ObjectId | Primary Key, Auto-generated |
| 2 |  | topic | String | Required (e.g., Data Structures) |
| 3 |  | difficulty | String | Enum: Easy, Medium, Hard |
| 4 |  | questionText | String | Required |
| 5 |  | options | Array | Contains multiple-choice options |
| 6 |  | correctAnswer | String | Required |
| 7 |  | explanation | String | Optional |

**4. Quiz Collection (quizzes)**

Stores quiz-related data for assessments.

| **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- |
| Quiz | \_id | ObjectId | Primary Key, Auto-generated |
| Quiz | title | String | Required |
| Quiz | questions | Array | List of question IDs |
| Quiz | timeLimit | Number | Time limit in minutes |
| Quiz | createdAt | Date | Auto-generated Timestamp |

**5. Mock Interview Collection (mock\_interviews)**

Stores mock interview data.

| **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- |
| Mock Interview | \_id | ObjectId | Primary Key, Auto-generated |
|  | category | String | Enum: HR Interview, Technical Round |
|  | questions | Array | List of question IDs |
|  | feedback | String | Automated or Admin-provided feedback |

**6. Admin Collection (admins)**

Stores admin details and access levels.

| **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- |
| Admin | \_id | ObjectId | Primary Key, Auto-generated |
|  | name | String | Required |
|  | email | String | Unique, Required |
|  | password | String | Required, Hashed |
|  | role | String | Enum: superadmin, moderator |

**7. Progress Tracking Collection (progress)**

Stores users' progress data.

| **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- |
| Progress | \_id | ObjectId | Primary Key, Auto-generated |
|  | userId | ObjectId | Foreign Key (References Users) |
|  | roadmapId | ObjectId | Foreign Key (References Roadmaps) |
|  | completedSteps | Array | List of completed roadmap steps |
|  | quizScores | Array | Stores quiz results |