Project Title	Quiz Game App
Technologies	MERN
Domain	Education
Project Level	Hard

Table

Table of Contents

1. Problem Statement:

Design a Quiz Game Application using the MERN stack

1.1. Overview of MERN Stack and Quiz Game Application

With the JavaScript stack MERN, full-stack web applications may be launched more rapidly and simply. The four technologies MongoDB, Express, React, and Node.js make up the MERN Stack. The development process is intended to be streamlined and made simpler.

These four powerful technologies each offer programmers a comprehensive working environment and significantly facilitate the development of web applications.

Quiz game is an application from where a user can join and can give test to judge his skills. It is one of the most popular and used application currently in between the students.

1.1.1 Expected features of the Quiz Application for the Admin

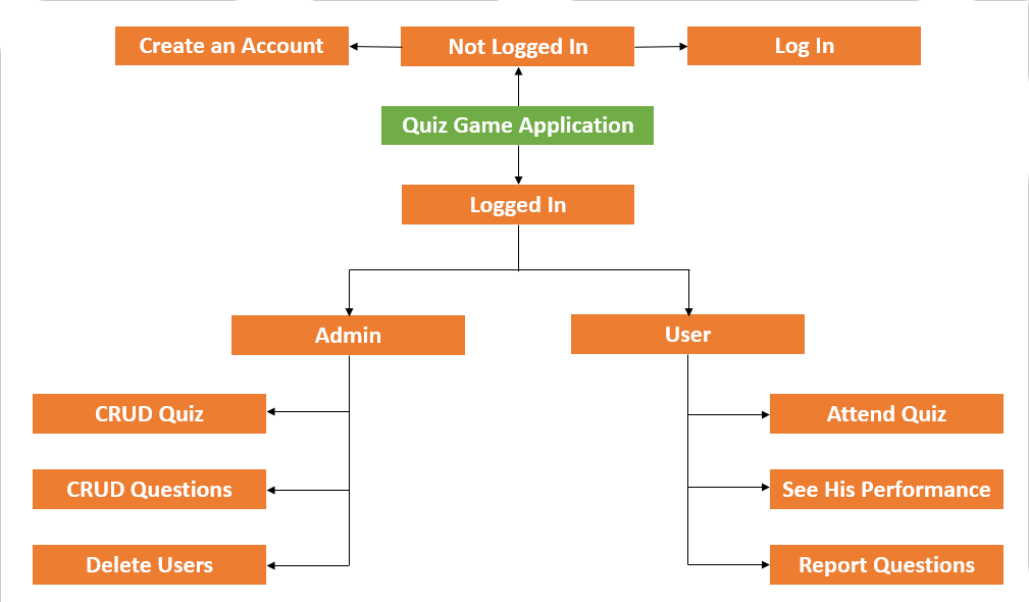
This is list of the minimum required features to which you have to incorporate in your application-

a. First thing should be the authentication system which will be needed to create an account or to log in (if account already exists).
b. Admin can add new quizzes, questions, fix answers, remove or add any user and can also track the progress of any of the user.
c. An admin dashboard will be required so that the admin can perform his job from there directly.
d. Dashboard should have total users, quizzes in which the admin can track them individually.

1.1.2 Expected features of the Quiz Application for the User

a. First thing should be the authentication system which will be needed to create an account or to log in (if account already exists).
b. A user can select any quiz and can attend it.
c. The report of his quiz should be shown at the time of quiz end using the graphs and text to visualize it in a better way.
d. User dashboard should be there so that the user can track his progress by seeing over his report graph.
e. Every time the questions of the quiz should shuffle so that the repetition can be stopped.

Low Level user flow



1.2. Project Objective

• **Quiz Game App:** A simple and easy to use platform so that the user can use to enhance his knowledge there by participating in the quizzes.

• **User-friendliness:** Platform should be easy to use so that everyone can use it and track their progress directly there.

1.3. Scope of The Project

1. There is a huge user base for the quiz application because everyone wants to improve their skills to become more skillful.

2. This project will give you a brief idea of using the MERN stack to solve a real-world problem.

1.4. Functional and Non-Functional Requirements: -

1.4.1. Functional Requirements

1. **User Registration:** The user must be able to register for the application through an Email, Username, and Password. On Opening the application, users must be able to register themselves or they can directly log in if they have an account already.

2. **CRUD on Quiz, Question and Users:** Admin can perform CRUD over any of these data over on the platform.

3. **Admin Dashboard:** An admin should be able to see all the users over his platform. He should be able to see their progress and participation.

4. **User Dashboard:** The user should be able to see all his progress and participation on his dashboard.

1.4.2. Non-Functional Requirements

1. **Privacy:** Only admin can perform CRUD operation on quizzes, questions and user.

2. **Robustness:** In case the user's system crashes, a backup of their progress report should be there to get it back.

3. **Performance:** The application must be lightweight and must loads the homepage and quizzes instantly.

1.5. Use Case Table

Authentication System	Register, Login, Logout	User and Admin
User Dashboard	Display User's details and progress	User and Admin
Admin Dashboard	Display All User's List, quizzes	Admin

Table 1. Use Case

2. Project Evaluation Metrics:

2.1. Code:

- You are supposed to write code in a modular fashion
- Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.

- Portable: It works the same in every environment (operating system).
- You have to maintain your code on GitHub.
- You have to keep your GitHub repo public so that anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

2.2. Database:

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

2.3. API Details or User Interface:

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

2.4. Deployment:

Implementation of reverse proxy, load balancing, and security group is mandatory for deployed applications.

2.5. Solutions Design:

You have to submit complete solution design strategies in High-level Document (HLD), Low-level Document (LLD), and Wireframe documents**.**

2.6. System Architecture:

You have to submit a system architecture design in your wireframe document and architecture document.

2.7. Optimization of solutions:

Try to optimize your solution on code level, architecture level, and mention all of these things in your final submission.

Mention your test cases for your project.

3. Submission requirements:

3.1. High-level Document:

You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link: HLD Document Link

3.2. Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: LLD Document Link

3.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: Architecture sample link

3.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: Wireframe Document Link

3.5. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

3.6. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link: DPR sample link

3.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link as per the given demo.

3.8. The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.