Online Quiz Portal Project WriteUp

1. Start by creating a new Spring Boot project. Use Spring Initialize or your preferred IDE to set up the basic project structure.

2. Add the necessary dependencies in the project's pom.xml file. Include dependencies for Spring Web and any other required libraries.

3. Create the entity classes that represent the core entities in your application. This includes classes like User, Quiz, Question, Answer, and Response. Add the necessary fields, getters, setters, and constructors to these classes.

4. Create the repository interfaces that will handle the database operations for each entity. Use Spring Data JPA to simplify the data access layer.

5. Implement the service classes that define the business logic for your application. These classes will use the repository interfaces to interact with the database and perform operations like creating quizzes, retrieving questions, validating answers, and calculating scores.

6. Create the controller classes that handle the HTTP requests and define the API endpoints for your application. Use annotations like @RestController and @RequestMapping to map the URLs to the corresponding controller methods.

7. Implement the registration and login functionality. Create an API endpoint for user registration, where users can provide their details such as name, email, and password. Implement a login API endpoint where users can authenticate themselves using their credentials.

8. Implement the admin login functionality. Create a separate API endpoint for admin login, where admins can authenticate themselves using their credentials.

9. Implement the CRUD operations for quizzes and questions. Create API endpoints that allow admins to create, read, update, and delete quizzes and questions. These endpoints should be protected and only accessible to authenticated admins.

10. Implement the API endpoints to view quizzes and questions. Create API endpoints that allow users to view available quizzes and questions. These endpoints should be accessible to authenticated users.

11. Implement the functionality to attend a quiz. Create an API endpoint where authenticated users can select a quiz, retrieve the questions for that quiz, and submit their answers.

12. Implement the functionality to check the quiz result. Create an API endpoint where users can view their quiz results. This endpoint should calculate the user's score based on the submitted answers and provide feedback on the correctness of each answer.

13. Test your application using a tool like Postman. Send requests to the API endpoints and verify that the expected responses are received.

14. Optionally, you can enhance the user interface of your application by developing a frontend using technologies like HTML, CSS, and JavaScript. This frontend can interact with the backend API endpoints to provide a more interactive and user-friendly experience.

15. Deploy your application to a web server or a cloud platform of your choice, making it accessible to users over the internet.