**Online Quiz Portal Using REST APIs**

DESCRIPTION

To create an online quiz portal with multiple REST APIs where users can browse different quizzes, attempt them, and find their scores and standings.

**Description:**

This project, which aims to create an online quiz application, is based on different REST APIs through which users can log in and take quizzes. After finishing the quiz, they can find their standing and scores based on their accurate responses.

There are two types of users on this quiz portal:

Admin user

Participants

**Sprint:**

The project is planned to be completed in 1 sprint.

Planning:

1)Creating the flow of the application

2)Writing java, springboot code .

3)Testing codes in postmant .

4) Pushing code to GitHub.

5)Document writing

**IDE:** Eclipse

**Programming Language:** Java

**Server:**Tomcat

**Backend:** Spring boot

**Tool:**Postmant, MySql Workbench

**GithubLin**k:

**Concepts Used:**

**Java:** Java basics, Java Object Oriented Programming, Data Structure, List, ArrayList, Algorithms, Conditional Statements,loop.

**Database**: MySql, Table, Entity

**Spring Technology**: Spring Core, Spring web, Spring MVC, Model, Service, Repository, Controller, Spring boot web, JPA, Autowiring, Mapping

**Security:** Spring Seccurity, Authentication,UserDetailsService

**Rest API:** Rest APIs, Http method, web service

**Process:**

**Project Cretion:**

1)Creted a spring boot project named ‘quiztest’.

2) Selected all dependencies .

3) Selected spring security dependency.

4)Update project

5) Configured all required properties in application.properties.

**Security creation in com.abc package:**

1)Created User.java

2)Created UserRepository.java

3) Created MyUserDetailsImp.java

4)Created MyUserDetailsService.java

5) Created AuthConfig.java for authentication purpose.

**Creation of all required model for quiz purpose:**

1)Created Question.java to store Question in database.

2)Created Quiz.java

3)Created Panswer.java

4)Created ParticipantDetails.java

5)Created CheckAnswer.java

6)Created QuestionToParticipant.java

7)Created QuizToParticipant.java

8)Created Statistics.java

**Creation of all required Repository:**

1)Created QuestionRepository.java

2)Created QuizRepository.java

3)Created ParticipantDetailsRepo.java

**Creation of all required controller class for quiz purpose**

1)Created QuestionController.java

2)Created QuizController.java

3)Created CreationController.java

4)Created ParticipantController.java

**Output:**

RESTful Web API to perform CRUD operations using Spring Boot and MySQL database.

The admin user has a API to access the admin portal, which requires authentication with the admin username and password.

The admin user can update the profile details and change the password after login.

The admin user can add questions using admin/question API.

The admin user can create a quiz by entering quizid and selecting questions with questionid.

The admin user can obtain statistics on total quizzes, questions, and users by using their APIs.

Participant user can explore various quizzes created by the admin.

Participant user can create their profile using new user registration.

Participant user can take the quiz and try to answer the questions.

Participant will get a response after giving the quiz where they can check if the given answers are correct or not.

**Pushing to git repository:**

* Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

* Initialize your repository using the following command:

git init

* Add all the files to your git repository using the following command:

git add .

* Commit the changes using the following command:

git commit . -m “Changes have been committed.”

* Push the files to the folder you initially created using the following command:

git push -u origin master

**Conclusion:**

Admin can login by username and passwoed and can create questions, quiz. Participant user can appear in quiz by register and login.

This application can by further developed by adding features web views.