Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Title of the Project

Quiz Portal

Abstract of the project

The Quiz Portal is a web-based application designed to simplify the management and conduct of online quizzes and examinations. Developed using Spring Boot for the backend and Angular for the frontend, the system allows administrators to create quizzes, manage questions, and view results, while enabling students to register, take quizzes, and review their performance. This project aims to provide a seamless, secure, and interactive testing environment, supporting multiple question types, timed assessments, and comprehensive result analysis. It is designed with scalability and performance in mind, leveraging RESTful APIs and MySQL for efficient data management.

Keywords

Generic Keywords :

Integration, Databases, Programming

Specific Technology Keywords :

Spring Boot, Angular, MySQL

Project Type keywords :

UI/UX Interface, API Integration, Testing, Security

Functional components of the project

1. User Registration and Authentication Enables students and administrators to securely register, log in, and manage their accounts using Spring Security with JWT-based authentication.

2. Quiz Creation and Management Allows admins to create, update, and delete quizzes, define time limits, and manage question banks.

3. Question Bank Management Supports adding, editing, and categorizing multiple types of questions (MCQs, true/false, short answer) for quizzes.

4. Quiz Participation Students can browse available quizzes, attempt them within the specified time, and submit their responses.

5. Admin Dashboard Provides administrative users with tools to view quiz statistics, manage student records, and monitor overall platform activity.

6. Role-Based Access Control Distinguishes between student and admin users, enforcing appropriate permission levels across the platform.

7. Secure API Communication The Angular frontend securely interacts with the Spring Boot REST API using JWT tokens for authentication and data protection.

8. Result Management Automates the calculation and display of quiz results, including score analysis, percentage, and rank generation.

9. Data Visualization (Dashboard) Includes graphical representation of quiz statistics for better insights and performance tracking.

10. Security and Data Protection Implements secure communication, encrypted data storage, and proper access control to protect sensitive user data.

Steps to start-off the project:

- Gain a Solid Understanding of the Required Technologies.
- Research Quiz Management Needs and User Workflows.
- Define User Roles (Students and Administrators).
- Design a User-Friendly and Intuitive Interface.
- Maintain Consistent and Professional UI/UX

Software Requirements:

Windows 10	Not applicable
MySQL Database Server	Not applicable
Spring Boot for Backend	Not applicable
Angular for Frontend	Not applicable

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Software requirements

No.	Milestone Name	Milestone Description	Timeline	Remarks
1	Requirements Specification	Define quiz system requirements (quiz creation, question bank, results, roles). Document them.	1-3	Consider enhancements like real-time scoring, live quizzes, and leaderboard support.
2	Technology familiarization	Understand Spring Boot, Angular, REST APIs, MySQL.	4-5	Focus on setting up sample apps, running the project locally, and reviewing architecture.
3	Database connection	Design models for Quizzes, Questions, Students, Results.	5-7	Ensure model relationships and validations are properly set.
4	High-level Detailed Design	Flowcharts for login, quiz management, result calculation.	7-9	Cover all user interactions including student and admin roles.
5	Implementation front-end	Create login page, quiz list UI, and navigation.	10-12	Start writing basic test cases and frontend component testing.
6	Front-End and Database Integration	Implement API connections, CRUD operations for quiz data.	12-13	Ensure Angular frontend connects properly with Spring Boot REST APIs.
7	Integration Testing	Test functionality: login, take quiz, result calculation	13-14	Use JUnit, Angular unit tests, and manual testing for flows.
8	Final Review	Final checks, documentation, and project demo.	15-16	Make sure deployment guide, user manual, and any gaps in features are addressed.

- Spring Boot Documentation
- Angular Official Docs
- MySQL Documentation
- W3Schools