“QUIZ PORTAL”

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

in Department of Information Technology

Project Mentor:
Dr.Vipin Jain
Associate Professor

Submitted By :
Archi Patidar
21ESKIT020

Department of Information Technology
Swami Keshvanand Institute of Technology, M & G, Jaipur
Rajasthan Technical University, Kota
Session 2024-2025

Swami Keshvanand Institute of Technology,
Management & Gramothan, Jaipur
Department of Information Technology

CERTIFICATE

This is to certify that Ms Archi Patidar, a student of B.Tech(Information Technology) 8th semester has submitted his/her Project Report entitled ”Quiz Portal” under my guidance.

Mentor	**Coordinator**
Dr. Vipin Jain	Priyanka Yadav
Associate Professor	Assistant Prof.

DECLARATION

We hereby declare that the report of the project entitled ”Quiz Portal” is a record of an original work done by us at Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur under the mentorship of ”Dr.Vipin Jain”(Dept. of Information Technology) and coordination of ”Ms. Priyanka Yadav” (Dept.of Information Technology). This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology (B.Tech) in the Department of Information Technology.It has not been submitted anywhere else, under any other program to the best of our knowledge and belief.

Team Members	**Signature**
Archi Patidar 21ESKIT020	

Acknowledgement

A project of such a vast coverage cannot be realized without help from numerous sources and people in the organization.We take this opportunity to express our gratitude to all those who have been helping us in making this project successful.

We are highly indebted to our faculty mentor Dr.Vipin Jain .He has been a guide, motivator source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We also thank our project coordinator Ms. Priyanka Yadav for her co-operation, encouragement, valuable suggestions and critical remarks that galvanized our efforts in the right direction.

We would also like to convey our sincere thanks to Dr. Anil Chaudhary, HOD, Department of Information Technology, for facilitating, motivating and supporting us during each phase of development of the project.Also, we pay our sincere gratitude to all the Faculty Members of Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur and all our Colleagues for their co-operation and support.

Last but not least we would like to thank all those who have directly or indirectly helped and cooperated in accomplishing this project.
Team Members:
Archi Patidar
21ESKIT020

Table of Content

List of Figures

Chapter 1
Introduction

1.1 Problem Statement and Objective

Traditional systems are also vulnerable to technical issues, leading to delays and undermining exam fairness and accuracy. Moreover, they are susceptible to cheating, compromising the credibility of student assessments and the integrity of the system.

The Quizz Portal aims to overcome these limitations by offering an efficient, accessible, flexible, and cost-effective online platform. It enables students to take exams remotely, integrates features to minimize cheating, and includes robust mechanisms to handle technical glitches—ensuring a fair and reliable evaluation process.]The Quizz Portal project addresses the inefficiencies and limitations of traditional examination systems. Conventional exam methods are often costly, time-consuming, and dependent on physical infrastructure such as classrooms and examination halls—posing challenges for institutions with limited resources. Additionally, they lack flexibility, requiring students to be present at specific times and locations, which may conflict with other commitments.

Traditional systems are also vulnerable to technical issues, leading to delays and undermining exam fairness and accuracy. Moreover, they are susceptible to cheating, compromising the credibility of student assessments and the integrity of the system.

The Quizz Portal aims to overcome these limitations by offering an efficient, accessible, flexible, and cost-effective online platform. It enables students to take exams remotely, integrates features to minimize cheating, and includes robust mechanisms to handle technical glitches—ensuring a fair and reliable evaluation process.

1.2 Introduction to Project

An online quiz portal allows users to participate in quizzes remotely using an internet-enabled device, offering accessibility and flexibility. It consists of two main interfaces: one for students to take quizzes and another for administrators to create, manage, and monitor them. The system provides instant feedback, aiding in self-assessment and performance improvement. It is cost-effective and efficient, eliminating the need for physical infrastructure. However, to ensure fairness and reliability, the system must address challenges like cheating prevention and technical issues.

1.3 Scope of the Project

The traditional quiz system faces several inefficiencies and limitations that hinder its effectiveness in modern educational and training environments. Conventional quiz setups often require significant physical infrastructure such as classrooms or meeting spaces, making them costly and logistically challenging—especially for institutions or organizations with limited resources. Additionally, these systems tend to lack flexibility, requiring participants to be present at a specific time and location, which may not be feasible for learners with varying schedules or remote locations. Traditional quiz methods also risk delays due to manual evaluation processes and are susceptible to errors and inconsistencies. Furthermore, preventing cheating and ensuring the integrity of quiz results can be difficult in non-digital environments. Therefore, the problem statement for developing an online quiz portal is to overcome these drawbacks by offering a cost-effective, accessible, and flexible platform that allows users to participate in quizzes remotely, ensures fair evaluation, and incorporates mechanisms to minimize cheating and technical disruptions.

Chapter 2
Software Requirement Specification

2.1 Overall Description

2.1.1 Product Perspective

The online quiz portal is a web-based system that provides a digital platform for conducting quizzes. It aims to replace traditional quiz methods with a scalable, efficient, and flexible solution. The system is divided into two primary modules: the student module and the admin module. The frontend communicates with the backend through REST APIs. It is built using modern technologies such as Angular, Spring Boot, and MySQL.

2.1.1.1 System Interfaces

The system is designed to interact with various components, such as web browsers and databases. Students and admins will use a browser to access the system. The backend connects to a MySQL database to store and retrieve data. Angular will send HTTP requests to Spring Boot APIs for all interactions. The system does not depend on any third-party desktop application

2.1.1.2 User Interfaces

The user interface is built using Angular and Bootstrap to ensure responsiveness. Students can register, login, attempt quizzes, and view their results. Admins can manage quizzes, view reports, and handle student data. The interface is clean, intuitive, and accessible on both desktop and mobile devices. Proper error messages and validation are implemented for a smooth user experience.

2.1.1.3 Hardware Interfaces

The system works on any device with basic modern specifications. It supports desktops, laptops, and mobile devices with internet access. Minimum hardware includes a Pentium 4 processor, 2 GB RAM (4 GB recommended), and a display resolution of 1280x1024. Devices should support any modern browser. The system doesn't require special hardware peripherals.

2.1.1.4 Software Interfaces

The frontend uses Angular, HTML, CSS, Bootstrap, and TypeScript. The backend uses Spring Boot with Java, and connects to a MySQL database. Hibernate may be used for database interactions. The application uses REST APIs to allow data exchange between client and server. Additional development tools like Angular CLI and Maven are used during development.

2.1.1.5 Communications Interfaces

All communication between frontend and backend takes place over HTTP/HTTPS. JSON format is used for sending and receiving data between components. A stable internet connection is required to interact with the system. There is no need for Bluetooth or other local network configurations. WebSockets may be used in the future for real-time updates.

2.1.1.6 Memory Constraints

The system requires at least 2 GB of RAM to run smoothly, though 4 GB is recommended. Low memory may result in slow performance, especially during quiz attempts or data processing. The application should be optimized to minimize memory leaks. Browser memory and local storage are also used temporarily. Memory-intensive operations are handled on the server side.

2.1.1.7 Operations

Users will log in and access their dashboards based on their roles. Admins can create quizzes, manage users, and view reports. Students can take quizzes within a specified time and receive instant feedback. All operations are session-based with proper validation and error handling. The system is built for smooth and uninterrupted functioning during active quiz sessions.

2.1.1.8 Project Functions

The core functionalities include user authentication, quiz creation, quiz participation, result evaluation, and report generation. Admins have full control over quiz content and user data. Students can only access quizzes assigned to them. All actions are logged and stored securely. Features like random question order and timer-based control enhance security and usability.

2.1.1.9 User Characteristics

Students are expected to have basic knowledge of internet usage and browsing. They should be able to navigate web applications and input answers correctly. Admins are usually faculty or technical staff with quiz management responsibilities. They should be able to handle question uploads and monitor student activity. No advanced technical expertise is required.

2.1.1.10 Constraints

The system must work efficiently on different browsers and operating systems. Internet dependency is a major constraint; without it, quizzes cannot be attempted. Quiz attempts are strictly time-bound and cannot be extended arbitrarily. All user inputs must be validated to prevent injection attacks. Performance issues may arise on very low-end devices.

2.1.1.11 Assumption and Dependencies

It is assumed that users will have a working internet connection and a compatible browser. The backend depends on Java and Spring Boot being properly configured. MySQL should be installed and accessible for data storage. The system assumes server uptime and availability. Dependencies like Angular CLI and Node.js should be installed for frontend development.

Chapter 3
System Design Specification

3.1 System Architecture

A system Architecture is a useful presenting tool since it illustrates how the main parts of my system work together and communicate. It acts as a kind of system road map.

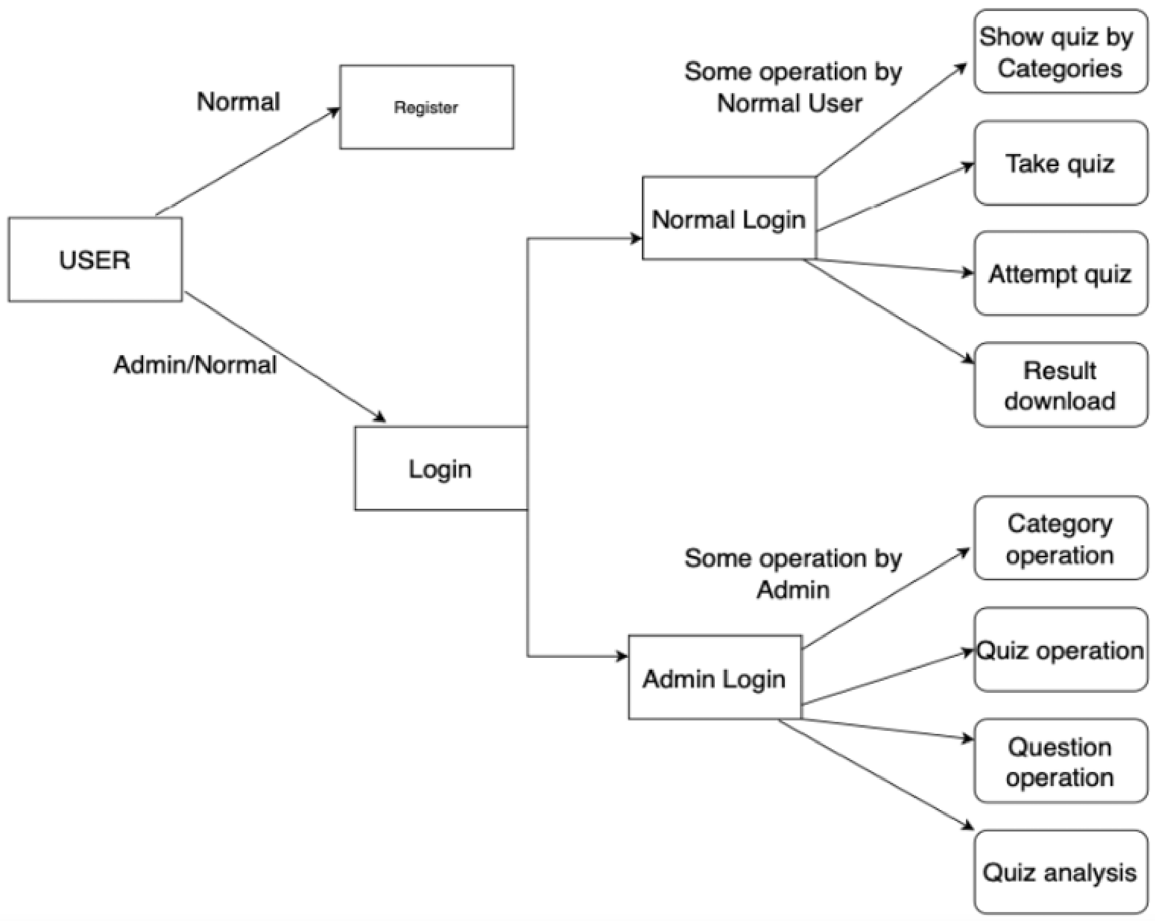


Figure 3.1: System Architecture

3.2 High Level Design Diagrams

3.2.1 Use Case Diagram

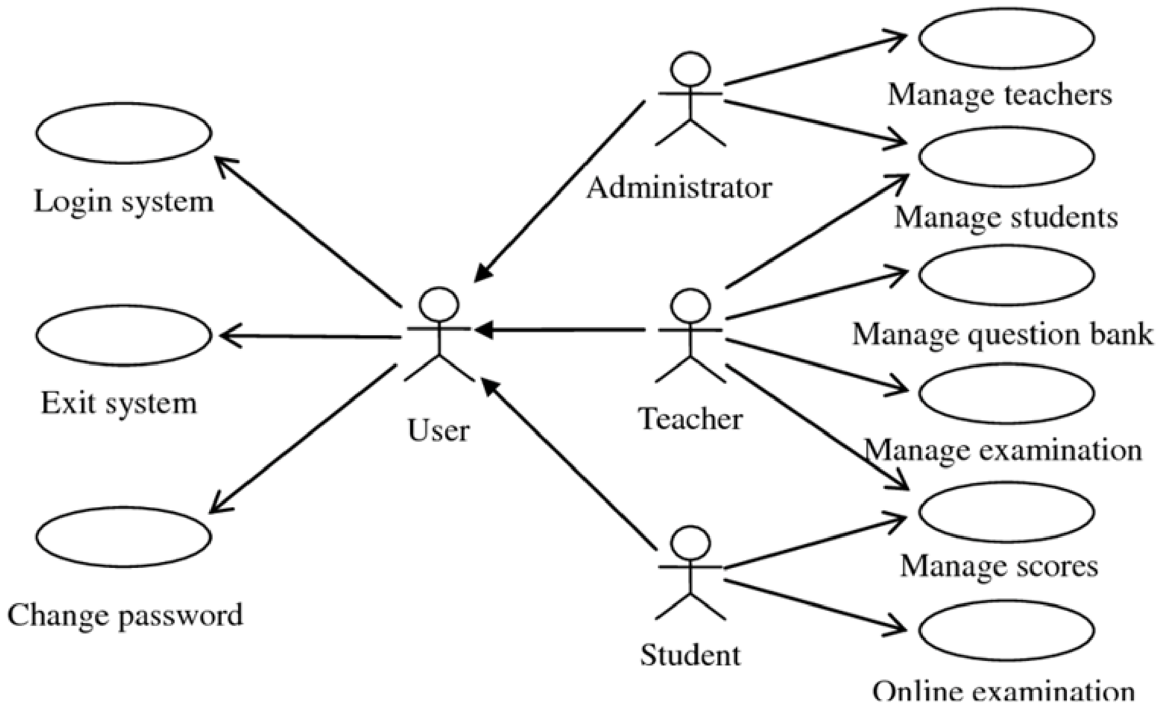


Figure 3.2: Use Case diagram

3.2.2 Activity Diagram

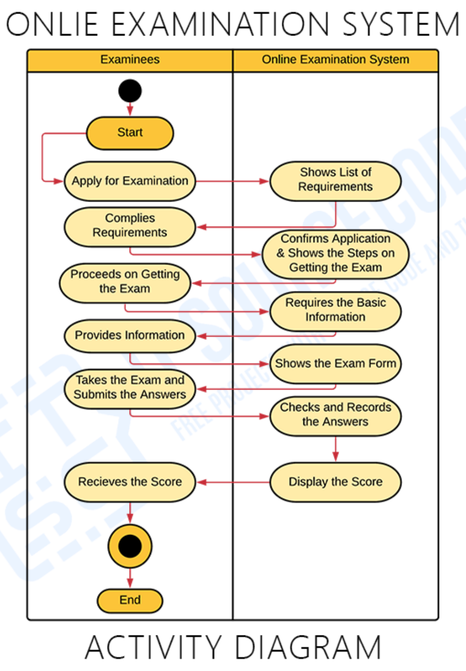

ONLIE EXAMINATION SYSTEM
Examinees
Online Examination System
Start
Apply for Examination
Shows List of Requirements
Complies Requirements
Confirms Application & Shows the Steps on Getting the Exam
Proceeds on Getting the Exam
Requires the Basic Information
Provides Information
Shows the Exam Form
Takes the Exam and Submits the Answers
Checks and Records the Answers
Recieves the Score
Display the Score
End
ACTIVITY DIAGRAM

Figure 3.3: Activity Diagram

3.2.3 Class Diagram

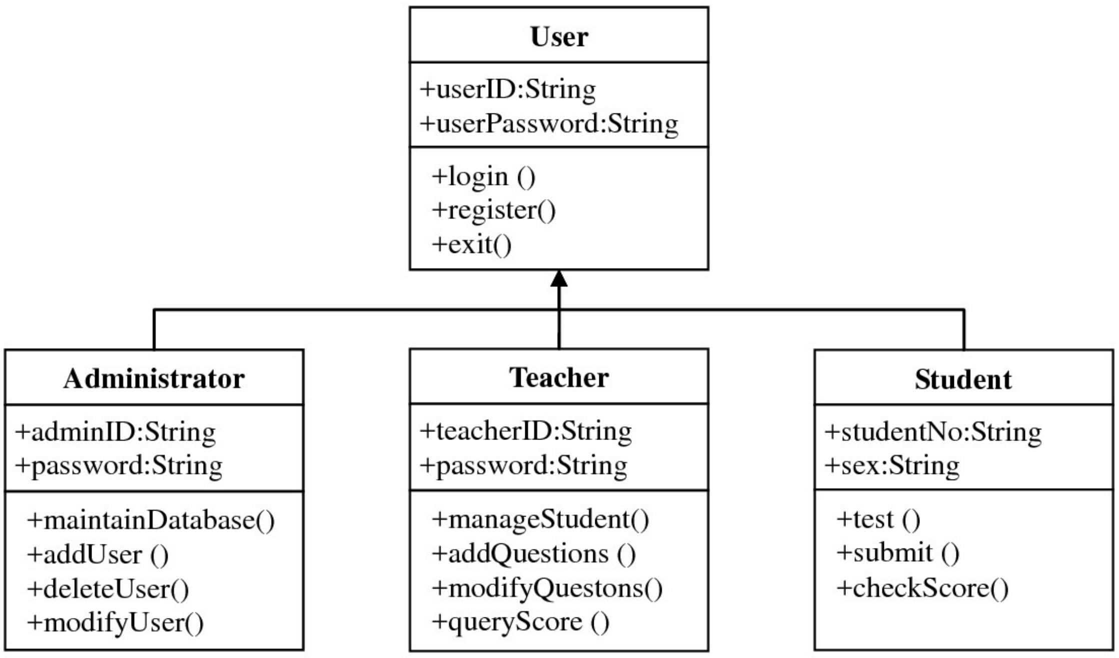


Figure 3.4: Class Diagram

Chapter 4
Implementation

4.1 Tools Used:Angular Material, type script html css, spring boot

Spring Boot: Spring Boot is a popular Java-based framework designed to simplify and accelerate the development of Spring applications. It offers a streamlined configuration process and includes many commonly used libraries, reducing the amount of boilerplate code required to get a project up and running. Spring Boot also includes features such as auto-configuration, which automatically configures the application based on its dependencies, and embedded servers, which allow for easy deployment of the application. These features make Spring Boot a popular choice for developing modern, scalable, and easily deployable web applications.

- Auto-configuration: Spring Boot includes a set of predefined configurations that can be automatically applied based on the dependencies included in the project.
- Embedded servers: Spring Boot includes embedded servers like Tomcat, Jetty, and Undertow that allow the application to be run as a standalone executable.
- Spring CLI: Spring Boot provides a command-line interface that allows developers to quickly create and run Spring applications.

- Actuator: Spring Boot Actuator provides a set of endpoints that can be used to monitor and manage the application at runtime.
- Dependency management: Spring Boot manages dependencies for the application, making it easier to manage and upgrade dependencies.
- Spring Data JPA: Spring Boot provides support for Spring Data JPA, which simplifies the development of data access layers in the application.

ng serve: ng serve is a command used in Angular development to start a local development server that compiles and serves an Angular application locally. It is a part of the Angular CLI (Command Line Interface) and is used during the development phase of an Angular application. when you run ng serve in your terminal, it will start the development server and 21 watch for any changes to your application files. When any changes are made, the application will automatically recompile and refresh in the browser. This allows developers to see the changes they make in real-time, without having to stop and restart the server manually. The ng serve command also provides additional options, such as specifying the port number to use, enabling hot module reloading, and setting up a proxy configuration for API requests. These options can be passed as command-line arguments to the ng serve command.

MVC (Model View Controller): The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects.

Model : The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

View : The View component is used for all the UI logic of the application. For example,the 22 Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

Controllers: Controllers act as an interface between Model and View components to pro- cess all the business logic and incoming requests, manipulate data using the Model com- ponent and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

MySQL: MySQL is an open-source relational database management system (RDBMS) that is widely used for building web applications. It is one of the most popular databases in use today and is often used in combination with the PHP programming language to build dynamic websites. MySQL is based on SQL (Structured Query Language) and uses tables, columns, and rows to store and manage data.

Angular Material: Button, form, card, and dialogue UI components are all pre-built and adaptable with the help of the Angular Material UI component library. It follows Google's Material Design guidelines and offers a consistent, modern look and feel for web applications. Angular Material also includes powerful features such as theming, accessibility, and internationalization support. It is built on top of the Angular framework and is designed to work seamlessly with it, providing a cohesive development experience. With Angular Material, developers can create responsive and mobile-friendly web applications quickly and easily, reducing the time and effort required to build a modern UI.

TypeScript: TypeScript is a statically-typed superset of JavaScript that adds optional static typing, classes, interfaces, and other advanced features to JavaScript. It is designed to improve the development experience and catch errors early in the development process. TypeScript code is compiled into plain JavaScript, making it compatible with all browsers and environments that support JavaScript. Some of the key features of TypeScript include: It is open-source software, especially for web applications and APIs. It is a fast and minimalist framework that helps make web applications fastly, easily, and with minimum lines of code.

Pom.XML File

- The Pom.XML file contains all the related dependencies which are used in the project.
- The most important part of this file is that it automatically downloads and configures all the jars and the suitable versions which are used to deploy or to connect the project. Just you need to add all the dependencies and plugins in the Pom.XML file.

Spring-boot-starter-data-jpa: Spring Boot Starter Data JPA is a popular starter package that enables developers to quickly set up and use the Java Persistence API (JPA) with the Spring framework. JPA is a specification that defines a standard way to map Java objects to relational databases, and Spring Boot provides a simplified and consistent way to configure JPA in your application. The starter package includes all the necessary dependencies, configuration, and auto-configuration to get started with JPA quickly and easily. With Spring Boot Starter Data JPA, developers can use annotations to define entities and relationships between them, perform CRUD operations, and write queries without writing SQL. The package also provides features like pagination, sorting, and caching to enhance performance. Spring Boot Starter Data JPA integrates seamlessly with other Spring Boot starters and is widely used in building modern web applications and microservices. All the database-related operations such as connecting projects with databases or defining Local port numbers when we are building our project and storing the data in the database. All the above figure shows the user role and the relation of the user, default and parameterized constructor getter and setter methods that anyone can't access or manipulate the data directly, and the access authority of the data of the different fields.

REST API for Exam Porta : REST (Representational State Transfer) API is a type of web service that uses HTTP protocol to exchange data between client and server. RESTful APIs are designed to be scalable, stateless, and cacheable, and they typically use standard HTTP methods like GET, POST, PUT, and DELETE to interact with resources. REST APIs can be used to build complex applications by providing a flexible and standardized way of accessing data across multiple platforms and devices. They have become a popular choice for building modern web applications and services.

- To upload the user-related data such as username, password, First Name, Last Name, email, and phone number and it also defines the role of the user he/she is ADMIN or NORMAL user.
- This first API is used to access the data of a person with the username and the other one for the same but it will return the details with the user.
- It is auto-wiring the other class by creating the child of that class so that it can be used in this particular class according to the need.

Token : A token generator in an API is a tool that generates a unique access token for a client or user to authenticate and authorize access to the API's resources. The access token typically contains a randomly generated string of characters that represents the client's identity and access privileges. The token generator issues the token upon successful authentication and validates the token during subsequent requests to ensure the user has the necessary permissions to access the requested resource. Token-based authentication is commonly used in modern API architectures and can enhance security by eliminating the need to transmit sensitive information like passwords over the network. It will generate the token whenever any user will log in to the system and it will save it in the database.

- If the password and username are not right then it will return the message invalid credentials.
- The above API will reflect the details of the current user who has accessed the portal.
- Throughout the one login period it will be there on logout it will automatically delete from the database.
- If the both username and password are not correct it will return the message user not found It will help to show the profile of the user and he or she can update or change the details.
- This will generate and validate the token of the login user.

Login and the request :

- If the token has login and the request portal is not null then it will return the string from the 7th index.
- If the token is invalid or expired it will reflect the message accordingly or if it is not started with Bearer it will also reflect the message.

Chapter 5
Centering System Testing

The designed quiz portal has been tested based on the following test parameters to ensure it functions efficiently and meets user expectations.

5.1 Functionality Testing

In testing the functionality of the quiz portal, the following features were verified:

1. Links

   (a) Internal Links: All internal navigation links within the portal—such as Dashboard, Start Quiz, View Results, and Admin Panel—were tested individually. Each link correctly directed the user to the intended section or page after providing necessary inputs, like login credentials or quiz selection.

   (b) External Links: Currently, the portal does not have external links. However, in future enhancements, external resources such as online learning material, student profiles, or certification pages may be integrated.

   (c) Broken Links: All links within the portal were verified and no broken links were found. Each link redirected successfully to the appropriate page without generating errors.

2. Forms

    (a) Error Messages for Incorrect Inputs: The portal provides real-time error messages when invalid or missing data is entered in forms. For instance, during login, entering incorrect credentials such as a wrong password prompts an appropriate error message. Similarly, during quiz creation or registration, incorrect formats (e.g., invalid email or date) trigger instant feedback.

    (b) Optional and Mandatory Fields: Mandatory fields in forms are clearly marked with a red asterisk (*). If a user tries to submit a form without filling in these fields, specific error messages are displayed. For example, while registering a new quiz or adding a user, not entering the quiz title or email triggers relevant validation warnings.

3. Database Testing ensured that all form submissions are successfully connected to the backend database. Quiz records, student responses, scores, and user profiles were stored and retrieved accurately.

5.2 Performance Testing

The performance of the quiz portal was evaluated under different load conditions:

- Load Time: The portal was tested for response time during peak usage (e.g., when multiple users attempted quizzes simultaneously).

The system maintained acceptable performance with minimal delays.

- Concurrent Users: Simulated tests with multiple concurrent users (students and admins) were performed. The portal successfully handled concurrent access without crashing or slowing down significantly.
- Quiz Submission Time: Timed quizzes were tested to ensure the timer runs accurately and submissions are captured instantly when time expires or the student submits manually.

5.3 Usability Testing

The quiz portal was tested for its ease of use and user experience:

- User Interface: The interface was found to be intuitive and user-friendly for both students and administrators. Navigation is straightforward with clearly labeled buttons and instructions.
- Accessibility: The design is responsive and works across various devices (desktop, tablet, and mobile), ensuring accessibility for a wide range of users.
- User Feedback: Test users provided feedback on the ease of taking quizzes, viewing results, and accessing features. Adjustments were made based on suggestions to improve clarity and reduce confusion.
- Help and Support: Tooltips and error messages guide users throughout the portal, reducing the need for external support.

Chapter 6
Project Screen Shots

6.0.1 Welcom Page

This is the home page of the web application. The user will be redirected to this page after login.

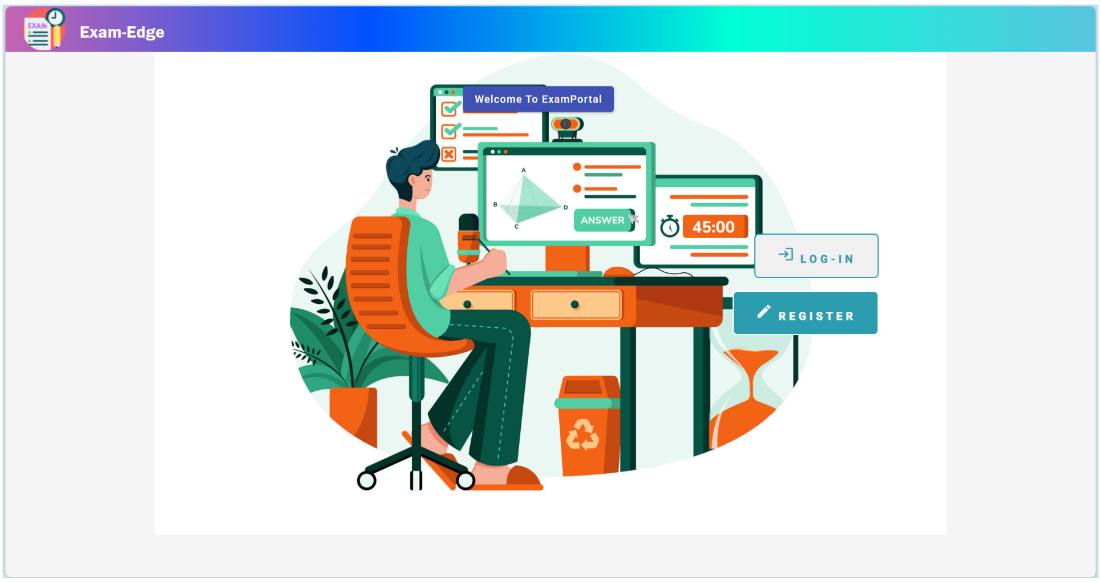


Figure 6.1: Welcome Page

6.0.2 Registration Page

To register for the new user.

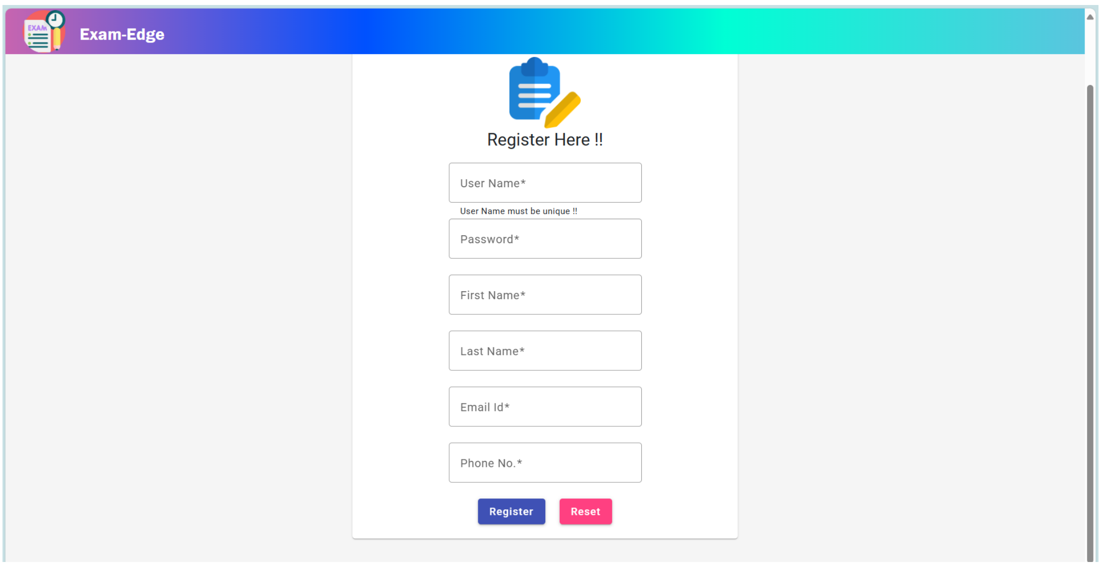


Figure 6.2: Registration Page

6.0.3 Successful Registered

It will pop up a bar and it will show the message and the unique user id generated.

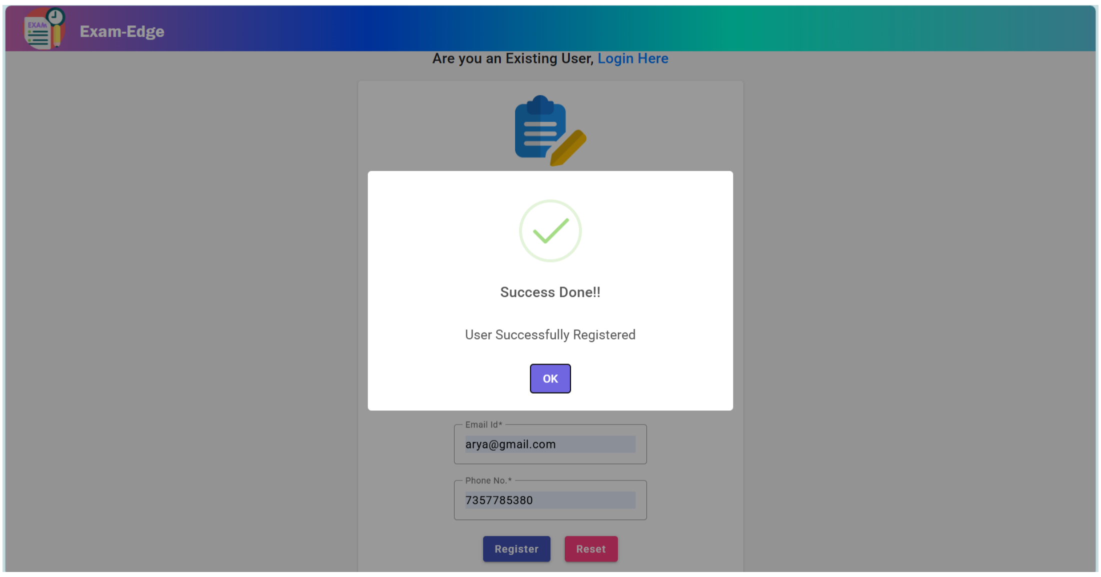


Figure 6.3: Successful Registered

6.0.4 Login Page

This is the login page.

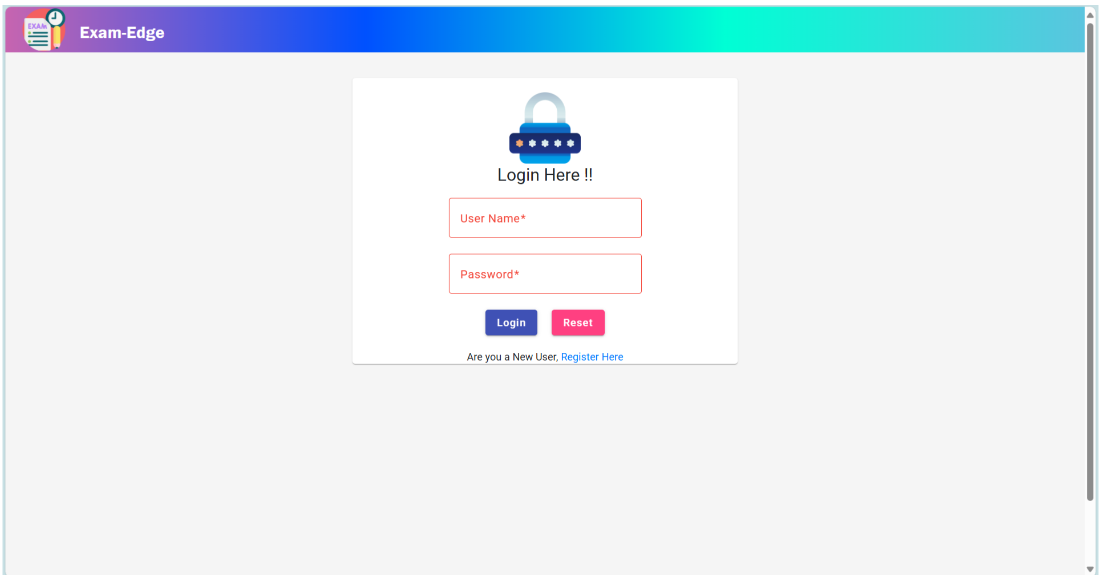


Figure 6.4: Login Page

6.0.5 Admin Dashboard

After login as Admin you will be reflected to this page where what a user can do it will be shown there.

Users can conduct quizzes by adding different quizzes and categories.

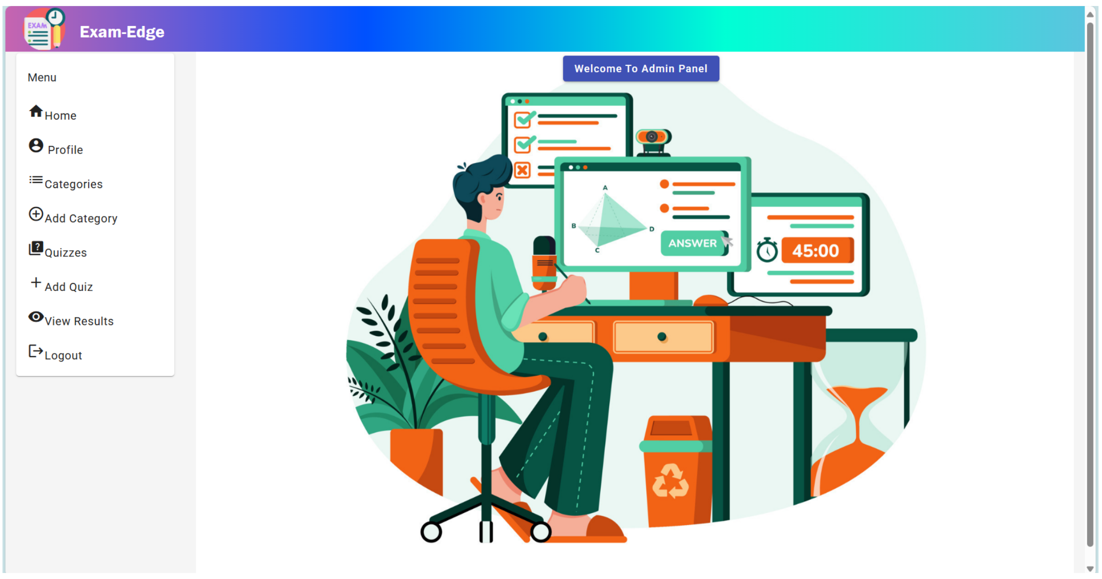
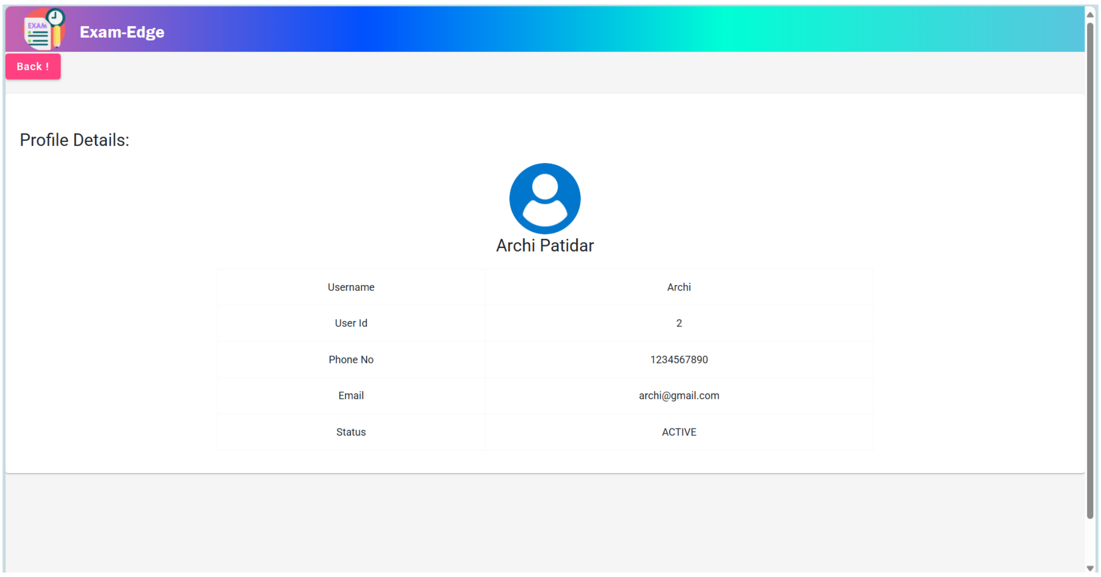


Figure 6.5: Admin Dashboard

6.0.6 Profile Detail

It is showing the profile details where the user can see their own registered name and email or mobile no.



Profile Details:

Archi Patidar

Username	Archi
User Id	2
Phone No	1234567890
Email	archi@gmail.com
Status	ACTIVE

Figure 6.6: Profile Detail

6.0.7 Add New Category

This is showing all the categories which are present and the admin can add more new categories.

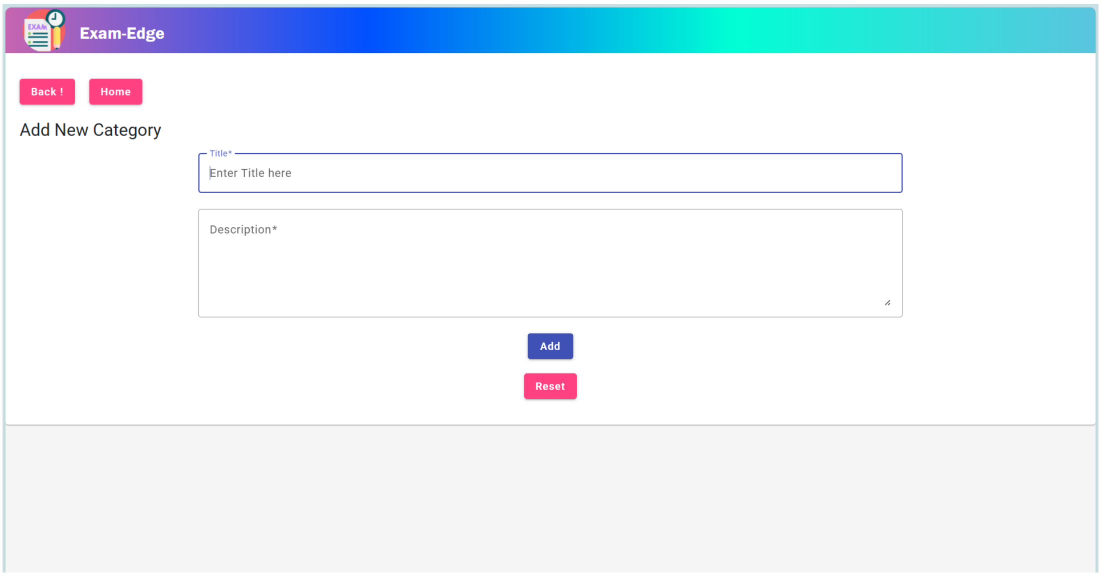
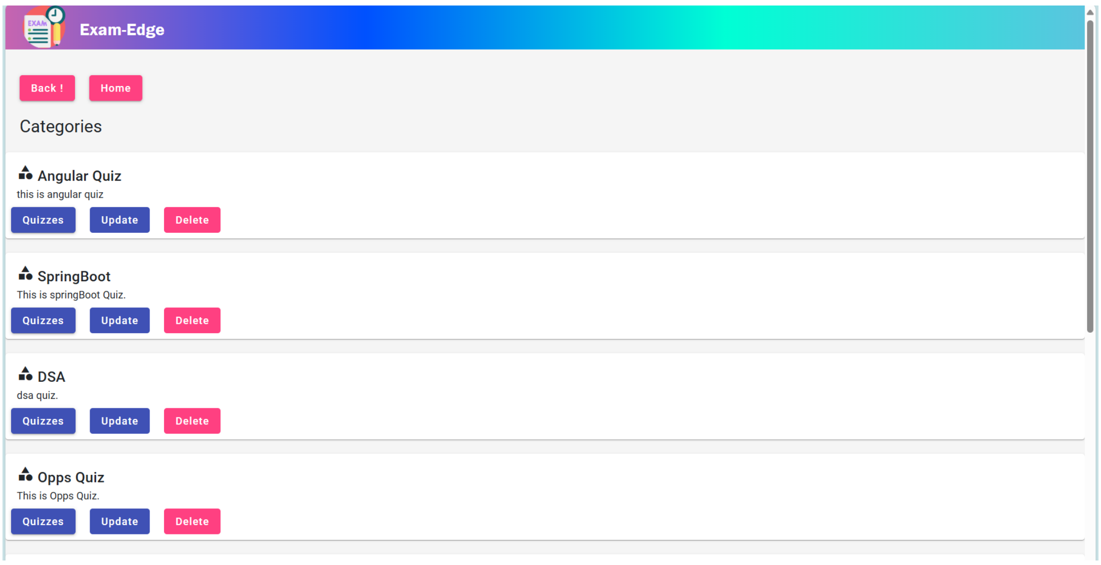


Figure 6.7: Add New Category

6.0.8 Show All Category



Categories

Angular Quiz
this is angular quiz
Quizzes Update Delete

SpringBoot
This is springBoot Quiz.
Quizzes Update Delete

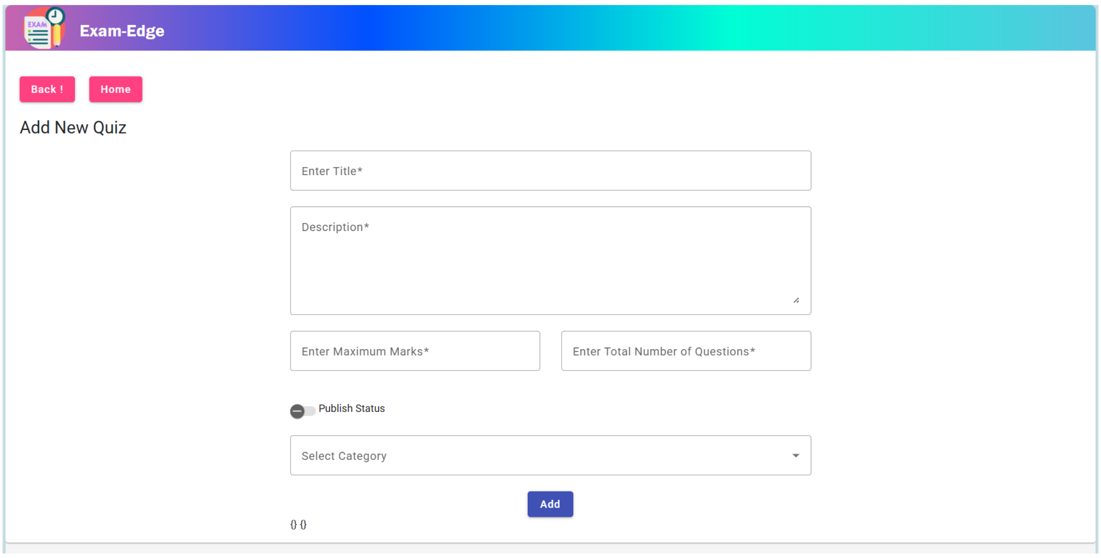
DSA
dsa quiz.
Quizzes Update Delete

Opps Quiz
This is Opps Quiz.
Quizzes Update Delete

Figure 6.8: Show All Category

6.0.9 Add Quiz

Adding new quizzes to the quiz section.



Add New Quiz

Enter Title*

Description*

Enter Maximum Marks*

Enter Total Number of Questions*

Publish Status

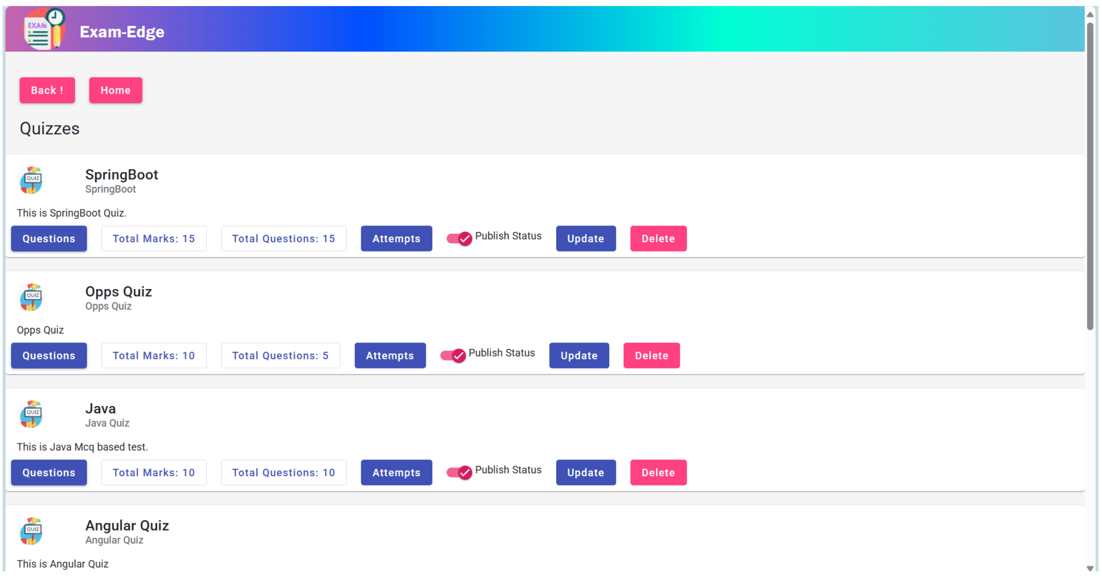
Select Category

0 0

Figure 6.9: Add Quiz

6.0.10 Show All Quiz

These are the quizzes.


Exam-Edge

Back ! Home

Quizzes

SpringBoot
SpringBoot
This is SpringBoot Quiz.
Questions | Total Marks: 15 | Total Questions: 15 | Attempts | Publish Status | Update | Delete

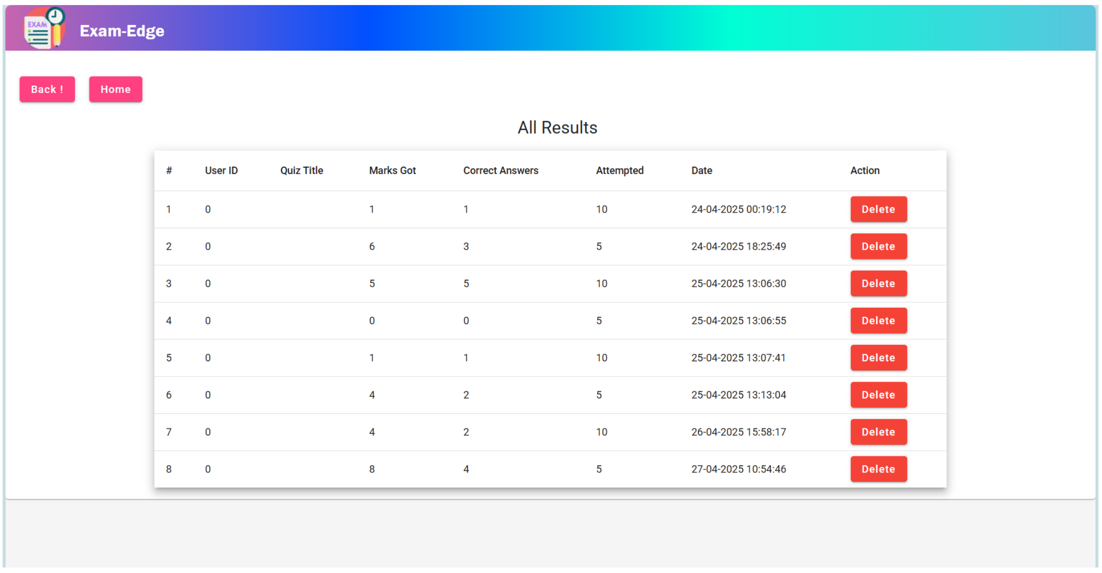
Opps Quiz
Opps Quiz
Opps Quiz
Questions | Total Marks: 10 | Total Questions: 5 | Attempts | Publish Status | Update | Delete

Java
Java Quiz
This is Java Mcq based test.
Questions | Total Marks: 10 | Total Questions: 10 | Attempts | Publish Status | Update | Delete

Angular Quiz
Angular Quiz
This is Angular Quiz

Figure 6.10: Show All Quiz

6.0.11 View Result


Exam-Edge

Back ! Home

All Results

#	User ID	Quiz Title	Marks Got	Correct Answers	Attempted	Date	Action
1	0		1	1	10	24-04-2025 00:19:12	Delete
2	0		6	3	5	24-04-2025 18:25:49	Delete
3	0		5	5	10	25-04-2025 13:06:30	Delete
4	0		0	0	5	25-04-2025 13:06:55	Delete
5	0		1	1	10	25-04-2025 13:07:41	Delete
6	0		4	2	5	25-04-2025 13:13:04	Delete
7	0		4	2	10	26-04-2025 15:58:17	Delete
8	0		8	4	5	27-04-2025 10:54:46	Delete

Figure 6.11: View Result

6.0.12 User Dashboard

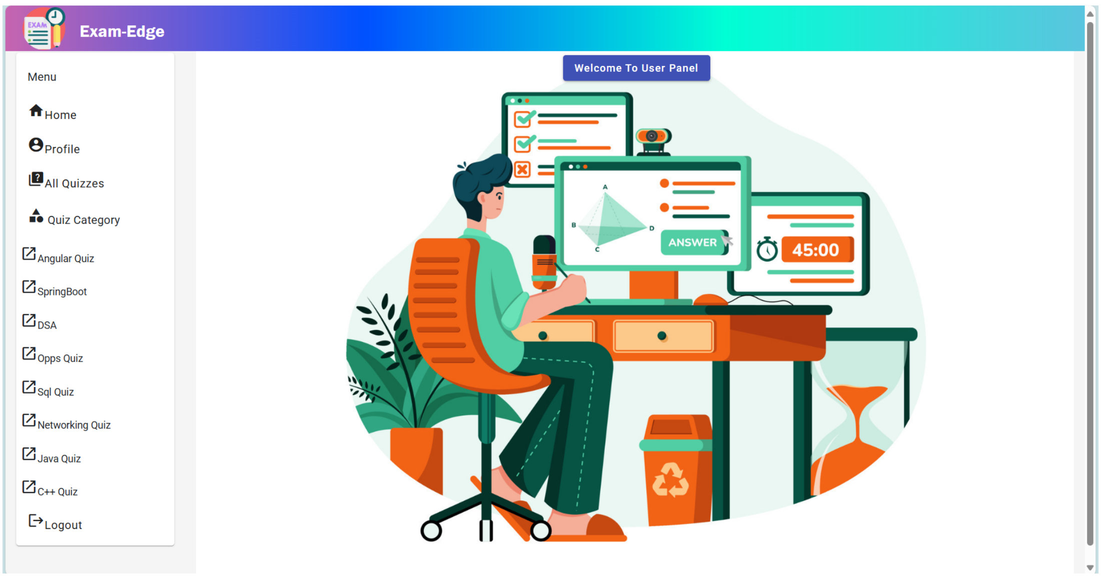
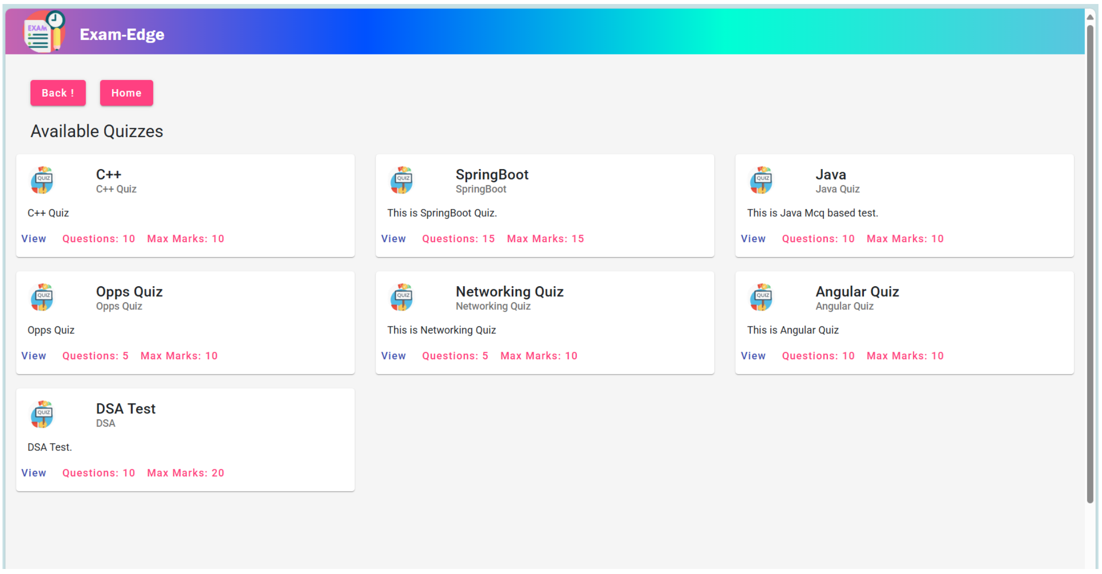


Figure 6.12: User Dashboard

6.0.13 Available Quizzes

These are the available quizzes normal users or students can see and attempt.



Available Quizzes

C++
C++ Quiz
C++ Quiz
View Questions: 10 Max Marks: 10

SpringBoot
SpringBoot
This is SpringBoot Quiz.
View Questions: 15 Max Marks: 15

Java
Java Quiz
This is Java Mcq based test.
View Questions: 10 Max Marks: 10

Opps Quiz
Opps Quiz
Opps Quiz
View Questions: 5 Max Marks: 10

Networking Quiz
Networking Quiz
This is Networking Quiz
View Questions: 5 Max Marks: 10

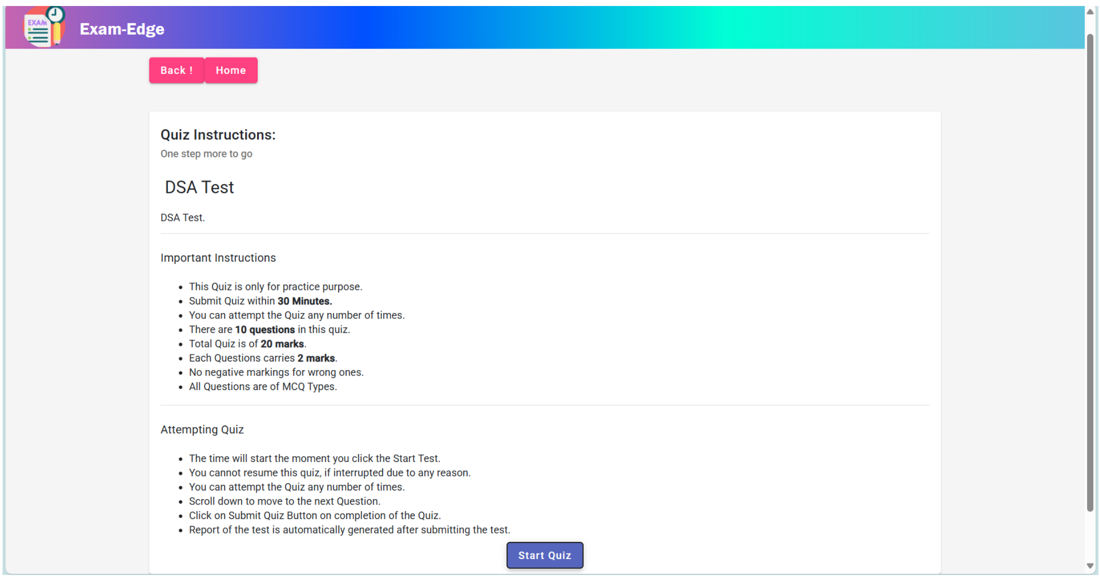
Angular Quiz
Angular Quiz
This is Angular Quiz
View Questions: 10 Max Marks: 10

DSA Test
DSA
DSA Test.
View Questions: 10 Max Marks: 20

Figure 6.13: Available Quizzes

6.0.14 Instructions Page

Instructions page containing the number of questions, time, and marks.


Exam-Edge

Back ! Home

Quiz Instructions:

One step more to go

DSA Test

DSA Test.

Important Instructions

- This Quiz is only for practice purpose.
- Submit Quiz within **30 Minutes.**
- You can attempt the Quiz any number of times.
- There are **10 questions** in this quiz.
- Total Quiz is of **20 marks**.
- Each Questions carries **2 marks**.
- No negative markings for wrong ones.
- All Questions are of MCQ Types.

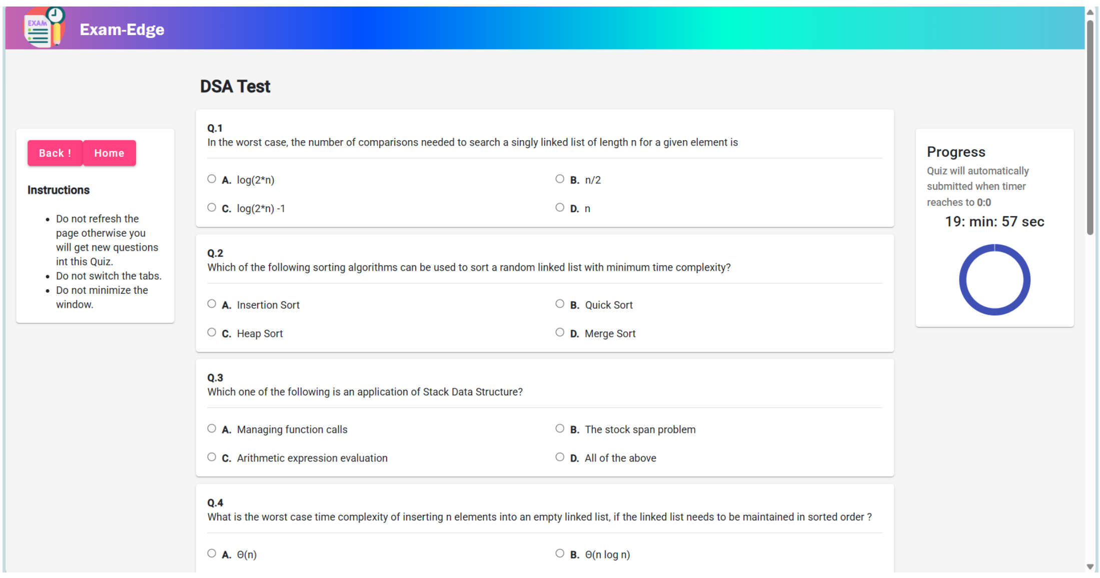
Attempting Quiz

- The time will start the moment you click the Start Test.
- You cannot resume this quiz, if interrupted due to any reason.
- You can attempt the Quiz any number of times.
- Scroll down to move to the next Question.
- Click on Submit Quiz Button on completion of the Quiz.
- Report of the test is automatically generated after submitting the test.

Start Quiz

Figure 6.14: Instructions Page

6.0.15 Attempt Quiz


Exam-Edge

DSA Test

Back ! Home

Instructions

- Do not refresh the page otherwise you will get new questions int this Quiz.
- Do not switch the tabs.
- Do not minimize the window.

Q.1
In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is

○ **A.** log(2*n) ○ **B.** n/2
○ **C.** log(2*n) -1 ○ **D.** n

Q.2
Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity?

○ **A.** Insertion Sort ○ **B.** Quick Sort
○ **C.** Heap Sort ○ **D.** Merge Sort

Q.3
Which one of the following is an application of Stack Data Structure?

○ **A.** Managing function calls ○ **B.** The stock span problem
○ **C.** Arithmetic expression evaluation ○ **D.** All of the above

Q.4
What is the worst case time complexity of inserting n elements into an empty linked list, if the linked list needs to be maintained in sorted order ?

○ **A.** Θ(n) ○ **B.** Θ(n log n)

Progress

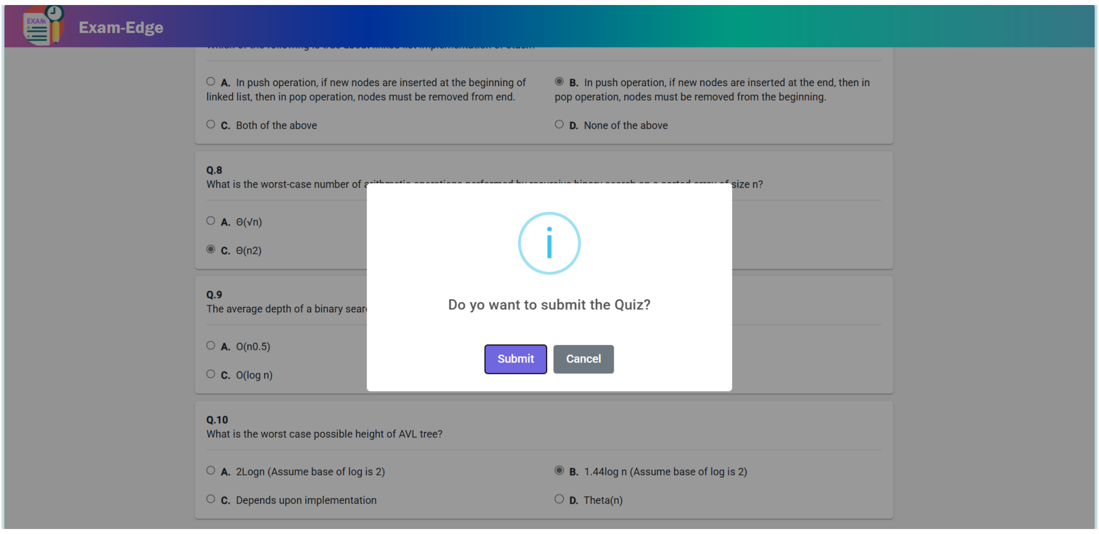
Quiz will automatically submitted when timer reaches to **0:0**

19: min: 57 sec

Figure 6.15: Attempt Quiz

6.0.16 Submit Quiz

It is showing the pop-up that you want to submit the quiz.



( ) **A.** In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.
[✓ **B.** In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from the beginning.]
( ) **C.** Both of the above
( ) **D.** None of the above

Q.8
What is the worst-case number of size n?

( ) **A.** Θ(√n)
[✓ **C.** Θ(n2)]

Q.9
The average depth of a binary sear

( ) **A.** O(n0.5)
( ) **C.** O(log n)

Do yo want to submit the Quiz?

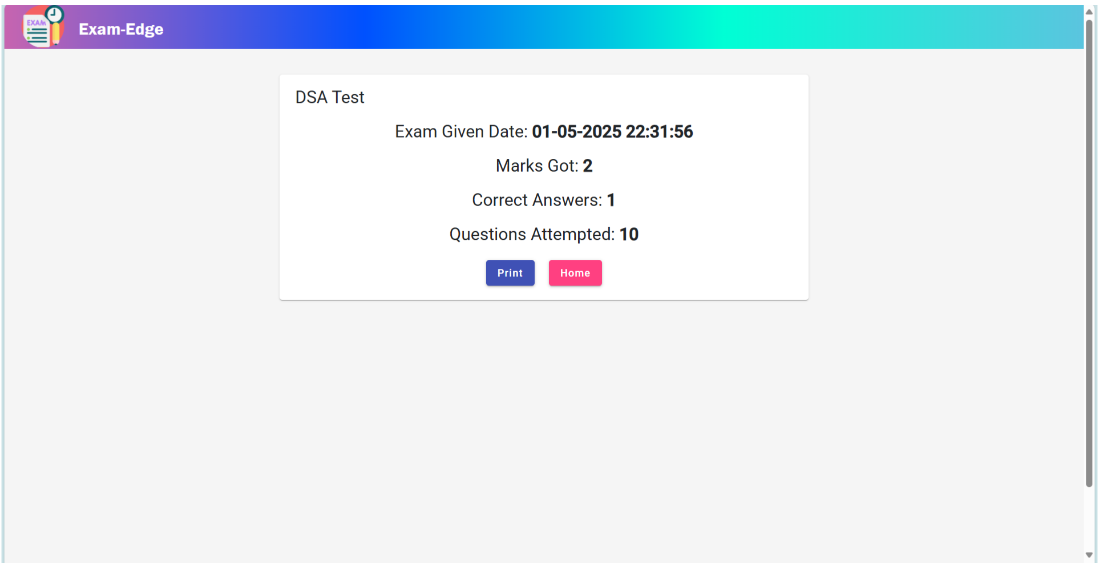
Q.10
What is the worst case possible height of AVL tree?

( ) **A.** 2Logn (Assume base of log is 2)
[✓ **B.** 1.44log n (Assume base of log is 2)]
( ) **C.** Depends upon implementation
( ) **D.** Theta(n)

Figure 6.16: Submit Quiz

6.0.17 Marks Screen

Users can see the marks just after attempting the quiz.



DSA Test

Exam Given Date: **01-05-2025 22:31:56**

Marks Got: **2**

Correct Answers: **1**

Questions Attempted: **10**

Figure 6.17: Marks Screen

Chapter 7
Project Summary and Conclusions

7.1 Conclusion

A quiz portal enhances accessibility to education and training by allowing users to take quizzes from any location at any time, leading to increased participation and higher completion rates. It reduces human errors in grading by automating quiz creation, administration, and evaluation, resulting in more accurate and reliable assessments. This automation also saves time and resources, which can be redirected to other educational or organizational activities. The portal can incorporate robust security features such as data encryption, anti-cheating mechanisms, and biometric authentication to ensure the integrity of quiz results. Instant feedback provided after each quiz helps learners quickly identify areas for improvement and adapt their study strategies, ultimately leading to better learning outcomes.

The quiz portal can be effectively used by educational institutions such as schools, colleges, and universities to conduct assessments, track student performance, and offer personalized feedback. It is also valuable in corporate training to evaluate employee knowledge and monitor progress. Certification bodies can use the system to conduct secure exams and issue credentials, while recruitment agencies and companies can assess job applicants' skills through custom quizzes.

Chapter 8
Future Scope

- Mobile Compatibility Develop a dedicated mobile app or optimize the existing quiz portal to be fully responsive on mobile devices. This ensures users can easily take quizzes on smartphones or tablets, increasing accessibility and convenience.
- Advanced Analytics Integrate advanced analytics using machine learning and predictive modeling to analyze quiz performance. This can help identify patterns, assess individual learning progress, and offer personalized recommendations to improve student outcomes.
- Collaborative Quizzes Introduce collaborative quiz modes where students can form groups and solve quizzes together in real-time. This fosters teamwork, encourages peer learning, and enhances the overall quiz experience.
- LMS Integration Integrate the quiz portal with popular Learning Management Systems (LMS) like Moodle, Blackboard, or Canvas. This streamlines quiz creation, scheduling, grading, and reporting within a unified learning environment.

References

[1] *1.https://v7.material.angular.io/components/categories*

[2] *2.https://angular.dev/overview*

[3] *3.https://www.geeksforgeeks.org/spring-boot/*