

A Project Report

on

QUIZORIA

By

Shreya Chaudhari

Rahul Shinde

MCA – I, SEM – I

2025-26

To

Savitribai Phule Pune University, Pune

In Partial Fulfillment of the Degree of
Master in Computer Application (M. C. A.)

Under The Guidance Of
Prof. Aishwarya Pawar

Suryadatta Group of Institutes, Pune
Suryadatta Institute of Business Management And Technology (SIBMT)

Date:-

CERTIFICATE

This is to certify that Ms. Shreya Chaudhari, Rahul Shinde has successfully completed project work entitled “**Quizoria**” in partial fulfillment of MCA – I Semester-I program for the year A.Y. 2025-26. She have worked under our guidance and direction.

Prof. Aishwarya Pawar
Project Guide

Dr.Manisha Kumbhar
PROFESSOR & DIRECTOR

Examiner 1

Examiner 2

Date :

Place :

Acknowledgment

We are the student of MCA first year. Here by we express our thanks to our project guide for allowing us to do the project on Quizoria. This project work has been the most exciting part of our learning experience which would be an asset for our future carrier. We would especially like to thank our guide and mentor Prof. Aishwarya Pawar, who constantly guided us in developing, pushing us to search for more answers to her numerous questions. Also I would like to thank Dr. Rupali Dahake & Prof. Apurva Patil, project coordinators for their support. As a building block of MCA Department, I thank Dr. Manisha Kumbhar, PROFESSOR & DIRECTOR MCA Department for her continuous support and help. We are grateful to many classmates who contributed their suggestions. Their hard work and examples push us to limits of our capability and encourage us daily.

Thank You

Student Name : Shreya Chaudhari
Rahul Shinde

Index

Chapter	Page number
CHAPTER 1: INTRODUCTION	
1.1 Existing System	
1.2 Need for System	
1.3 Operating Environment Hardware and Software	
CHAPTER 2: PROPOSED SYSTEM	
2.1 Proposed System (Introduction of system)	
2.2 Module specifications (Scope)	
2.3 Objectives of System	
CHAPTER 3: ANALYSIS & DESIGN	
3.1 Use Case Diagrams	
3.2 Activity Diagram	
3.3 Class Diagram	
3.4 Module Hierarchy Diagram	
3.5 Table specifications (Database design)	
3.6 Data dictionary	
CHAPTER 4: USER MANUAL	
4.1 User Interface Screens (Input)	
4.2 Output Screens with data	
4.3 Data Reports	
4.5 Sample program code	
4.6 Limitations and Bibliography	

CHAPTER 1: INTRODUCTION

Introduction

In today's fast-paced digital era, learning programming languages has become more than just an academic requirement—it is a vital skill for students, professionals, and anyone aspiring to thrive in the technology-driven world. With the growing demand for web development, mobile applications, and software solutions, individuals often struggle to find platforms that make learning both effective and enjoyable. This is where **Quizoria**, our innovative online quiz application, steps in as a powerful tool designed to transform the way people engage with programming concepts. Quizoria is not just another quiz platform; it is a thoughtfully crafted system that combines education, entertainment, and technology to create an immersive learning experience.

The primary aim of Quizoria is to provide users with a convenient, accessible, and interactive environment where they can test their knowledge of programming languages such as HTML, CSS, Java, and Android. Unlike traditional learning methods that rely heavily on textbooks or lengthy lectures, Quizoria introduces a dynamic approach by offering multiple quiz formats, including multiple-choice questions, true or false challenges, and open-ended questions. This variety ensures that learners of all levels—whether beginners or advanced coders—can find quizzes tailored to their needs and learning styles. By integrating real-time scoring, leaderboards, and gamification elements, the application motivates users to push their boundaries while enjoying the process of learning.

One of the most significant needs addressed by Quizoria is accessibility. Students and professionals often face time constraints, and the ability to practice and learn anytime, anywhere is invaluable. Quizoria provides 24/7 availability, allowing users to engage with quizzes at their convenience. Moreover, the system goes beyond simple scoring by offering detailed feedback and explanations for each answer, enabling learners to understand their mistakes and focus on areas that require improvement. This immediate feedback loop is crucial for building confidence and ensuring steady progress.

From a technical perspective, Quizoria is built on a robust stack of web technologies, including HTML, CSS, JavaScript for the frontend, and Java with MySQL for the backend. This ensures a secure, scalable, and reliable system capable of handling multiple users simultaneously. Features such as secure user authentication, personalized profiles, and progress tracking make the platform not only user-friendly but also highly practical for long-term learning. The inclusion of leaderboards and badges further enhances engagement, fostering a sense of healthy competition among peers.

Ultimately, Quizoria is more than just a project, it is a vision to make programming education accessible, engaging, and effective for everyone. Whether you are a student preparing for exams, a professional seeking to upskill, or simply a curious learner exploring the world of coding, Quizoria offers a platform that bridges the gap between knowledge and application. By combining convenience, variety, and innovation, Quizoria stands as a modern solution to the challenges of learning programming in the real world.

1.1 Existing System

Our application is designed to test your knowledge, challenge your thinking, and provide entertainment. Our application is designed to provide an immersive and engaging quiz experience. Users can participate in various quiz formats, including:

Multiple-choice questions

True or False

Open-ended questions.

The Application Provide Real-time scoring and leaderboard rankings. Provide an interactive and engaging platform for learning programming concepts and languages . Offer a comprehensive range of quizzes covering various programming languages, like HTML, CSS, PHP, ANDROID. Make programming learning fun, easy, and accessible to learners of all levels and backgrounds

1.2 Need for System-

2. Instead of passively reading notes, quizzes force students to recall and apply concepts. This strengthens memory and builds confidence.
3. Students often face MCQs in university exams or competitive tests. Practicing through online quizzes mirrors that format.
4. Unlike traditional assignments, quizzes give immediate results. Students know exactly where they went wrong and can correct themselves quickly.
5. Quizzes are short, bite-sized, and less intimidating than full coding projects. They reduce exam anxiety by breaking learning into manageable chunks.
6. Many apps allow leaderboards or challenges among classmates, which motivates students to study harder in a fun way.
7. Whether on a phone, tablet, or laptop, students can practice anywhere—even during commutes or breaks.
8. Companies often begin with online coding quizzes or MCQs to filter candidates. Practicing on quiz apps simulates that environment.
9. Busy adults can't always commit to long lectures. A 10-minute quiz fits into their schedule and still teaches something valuable.
10. Even non-tech people (like business analysts or researchers) can benefit by learning basic programming concepts to automate tasks.

1.2Operating Environment

Hardware Requirements (Server Hardware)

Component	Minimum Requirement	Recommended for Better Performance
Processor	Modern multi-core processor (e.g., Intel Xeon, AMD Ryzen)	—
RAM	2 GB	4 GB or more
Storage	At least 20 GB for web files and database	SSD preferred; scalable based on traffic and data

Software Requirements (Server-Side Software)

Component	Type	Requirement / Version
Operating System	Server-Side	Linux-based OS (Ubuntu, CentOS) or Windows Server
Application Server	Server-Side	Apache Tomcat / Jetty / WildFly
Java Development Kit	Server-Side	JDK Version 11 or higher (LTS recommended, e.g., JDK 17)
Database	Server-Side	MySQL 5.7 or higher (MariaDB/PostgreSQL as alternatives)
Database Management Tool	Client-Side	MySQL Workbench / pgAdmin

CHAPTER 2 : PROPOSED SYSTEM

2.1 Proposed System (Introduction of System)

The proposed system, Quizoria, is envisioned as a comprehensive and interactive web-based application designed to make learning programming languages engaging, accessible, and effective. At its core, the system focuses on delivering a structured quiz experience that not only tests knowledge but also enhances understanding through immediate feedback and gamified participation. The first major component of the system is Quiz Creation and Management, which empowers administrators with a dedicated panel to create, update, and delete quizzes. This module supports multiple question formats such as multiple choice, true/false, and fill-in-the-blank, ensuring flexibility and variety in assessment. Questions can be organized into categories and difficulty levels, allowing learners to progress gradually from basic to advanced concepts.

The second component, Quiz Participation, emphasizes user experience by offering a simple and intuitive interface where users can browse quizzes by category or difficulty level. To maintain engagement and simulate real-world exam conditions, a built-in timer is included for individual questions or entire quizzes. Once a quiz is completed, the Score Calculation and Feedback module automatically evaluates responses and displays the total score, correct answers, and detailed explanations. This immediate feedback loop helps learners identify mistakes and strengthens their conceptual clarity.

Finally, the system integrates Leaderboards and Gamification to foster motivation and healthy competition. Real-time leaderboards highlight top performers, while badges, rewards, and milestones encourage continuous participation and skill improvement. Together, these features transform Quizoria into more than just a quiz platform—it becomes a dynamic learning environment that blends education with entertainment, making programming practice enjoyable, personalized, and impactful for students and professionals alike.

2.2 Module Specifications (Scope)

1. Admin Module

The admin module provides complete control over the quiz platform. It includes a secure panel where administrators can create, update, and delete quizzes. Different question formats such as multiple choice, true/false, and fill-in-the-blank are supported. Quizzes can be organized by categories and difficulty levels to ensure structured learning. In addition, the admin can manage user accounts, monitor activity, and generate reports or analytics to evaluate performance and system usage. This module ensures scalability, flexibility, and smooth functioning of the overall system.

2. User Module

The user module is designed to give learners a personalized and convenient experience. It allows users to register and log in securely, maintaining their own profile with quiz history and performance records. Users can easily browse quizzes by category or difficulty level, making it simple to select quizzes that match their learning goals. The module also tracks progress over time, helping users identify strengths and weaknesses. With 24/7 accessibility, users can

practice anytime, while leaderboards and badges add motivation and encourage healthy competition.

3. Quiz Module

The quiz module is the core of the system where users actively participate in quizzes. It provides a user-friendly interface with built-in timers for individual questions or entire quizzes, simulating exam conditions. After completion, scores are automatically calculated and displayed instantly, along with correct answers and detailed explanations to help learners improve. This module also integrates gamification features such as rewards, milestones, and real-time leaderboards, making the learning process interactive and enjoyable. By combining assessment, feedback, and motivation, the quiz module ensures that learning programming concepts becomes engaging and effective.

2.3 Scope of the System

1. Add More Topics

The system can be expanded to include a wider range of programming languages, frameworks, and emerging technologies. This will allow learners from diverse backgrounds to access quizzes that match their academic or professional interests, making the platform more inclusive and valuable.

2. Personalized Feedback and Analytics

Future development can focus on providing detailed, personalized feedback for each user. Analytics such as accuracy rates, time taken per question, and topic-wise performance will help learners identify their strengths and weaknesses, guiding them toward targeted improvement.

3. Adaptive Difficulty Levels

The system can introduce adaptive quizzes that automatically adjust difficulty based on user performance. Beginners will receive simpler questions to build confidence, while advanced learners will be challenged with complex scenarios, ensuring a tailored learning experience.

4. Integration with Learning Resources

Beyond quizzes, the platform can link users to tutorials, documentation, or video lectures related to the topics they struggle with. This scope ensures that the system not only tests knowledge but also supports continuous learning.

5. Mobile Application Development

To increase accessibility, the system can be extended into a mobile app version. This will allow users to practice quizzes on the go, making learning more flexible and convenient across devices.

6. Gamification Enhancements

Additional gamification features such as streaks, daily challenges, and achievement badges can be introduced. These elements will motivate users to engage regularly, turning learning into an enjoyable and rewarding activity.

7. Collaboration and Community Features

The system can evolve to include discussion forums, peer challenges, and group quizzes. By fostering collaboration, learners can share knowledge, compete in teams, and build a supportive learning community around programming.

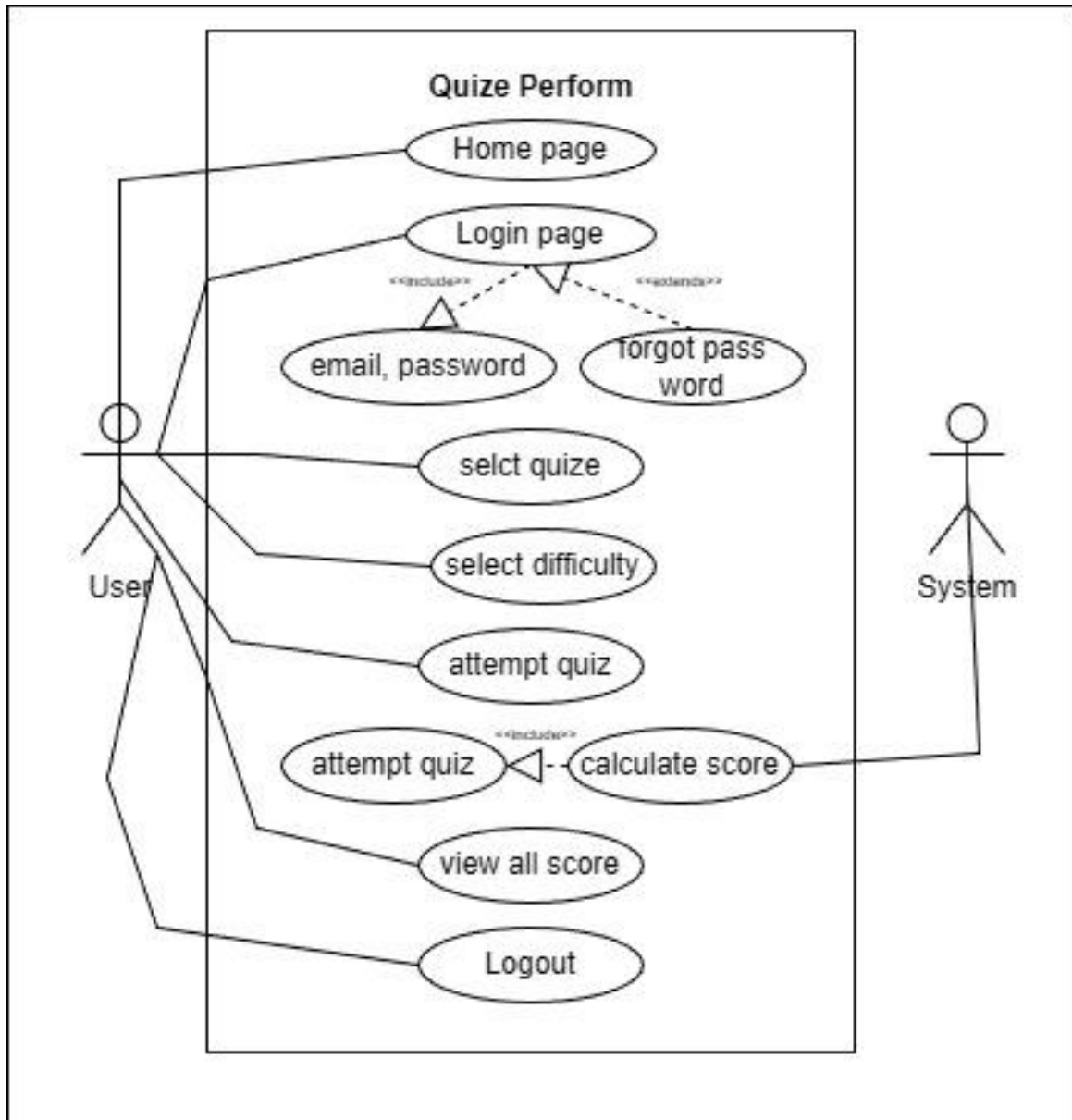
2.4 Objectives of the System

1. TO provide a user-friendly platform where learners can easily register, log in, and access programming quizzes anytime, anywhere.
2. TO create a secure environment with proper authentication so that users' profiles, quiz history, and performance data are safely maintained.
3. TO design an efficient quiz management system that allows administrators to create, update, and organize quizzes by category, topic, and difficulty level.
4. TO deliver immediate and detailed feedback after quiz completion, including correct answers, explanations, and performance insights, to help users learn from mistakes.
5. TO implement gamification features such as leaderboards, badges, and rewards to motivate learners and encourage continuous participation.
6. TO ensure scalability of the system by supporting multiple programming languages, frameworks, and technologies, thereby catering to a wider audience.
7. TO provide personalized analytics and progress tracking so that users can identify strengths, weaknesses, and areas for improvement over time.

CHAPTER 3: ANALYSIS & DESIGN

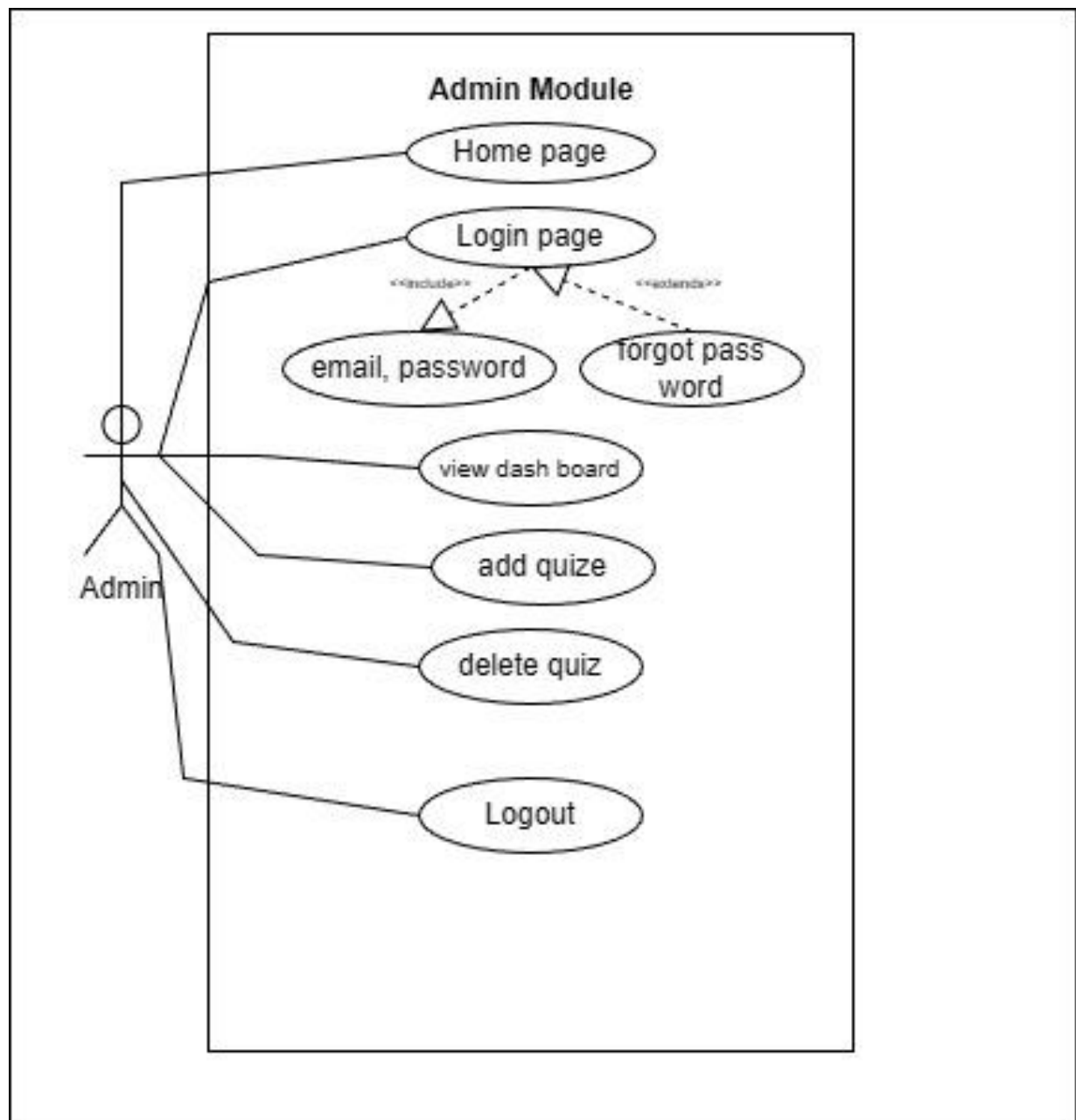
3.1 Use Case Diagrams

Quiz perform module



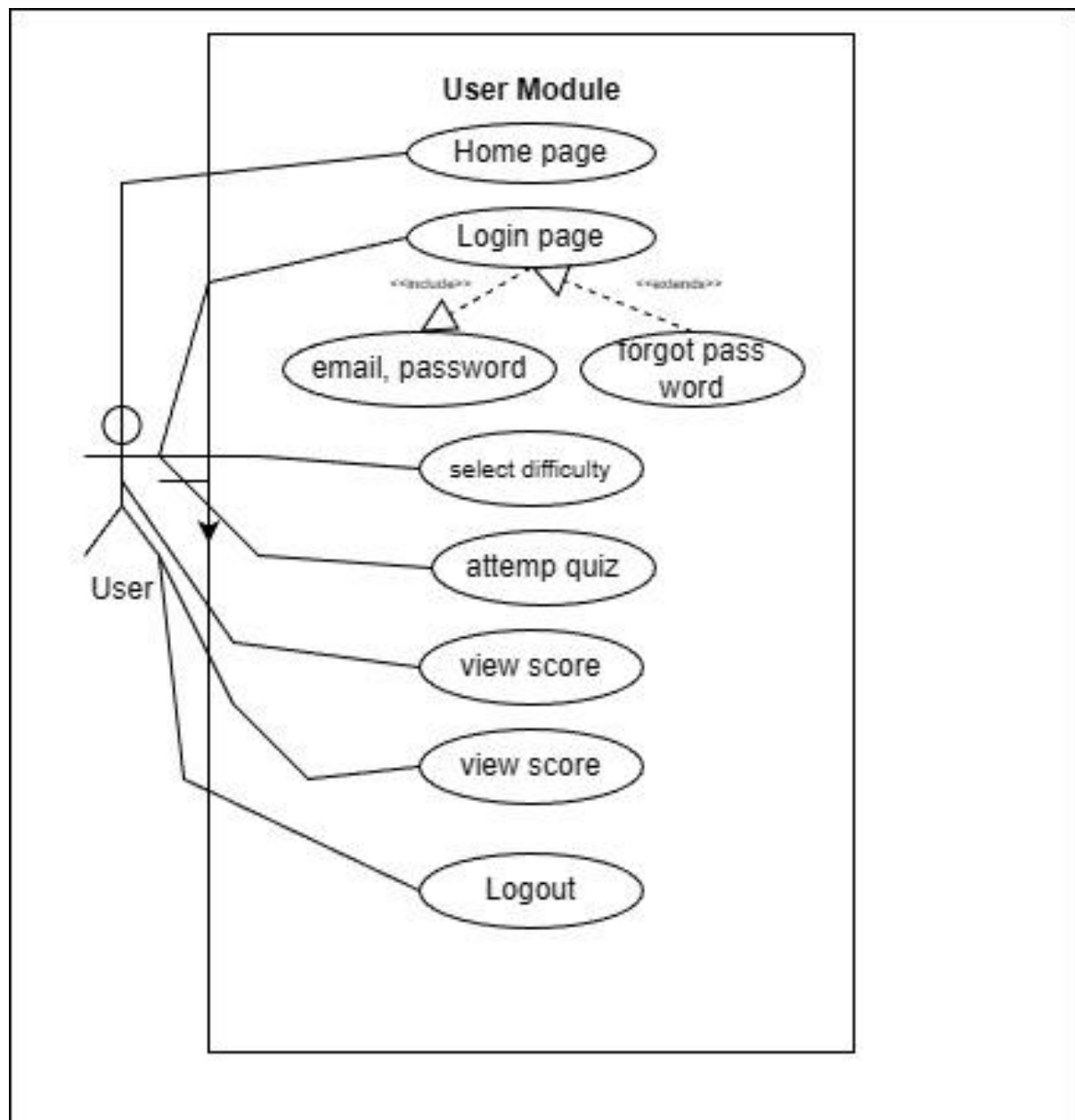
Field / Use Case	Description
Home Page	Entry point of the application where users can navigate to other features.
Login Page	Allows users to log in using email and password.
Forgot Password	Optional extension to recover or reset password if login fails.
Select Quiz	Enables users to choose a quiz topic or category.
Select Difficulty	Lets users pick the quiz level (Easy, Medium, Hard).
Attempt Quiz	Users answer questions in the selected quiz.
Calculate Score	System calculates the user's score after quiz submission.
View All Score	Displays the user's past quiz scores and performance.
Logout	Ends the user session and exits the system securely.

Admin module



Field / Use Case	Description
Home Page	Entry screen for the admin to access system features.
Login Page	Allows admin to log in using credentials.
Email, Password	Required fields included in the login process.
Forgot Password	Optional extension to recover admin credentials.
View Dashboard	Displays system overview and admin controls.
Add Quiz	Enables admin to create and upload new quizzes.
Delete Quiz	Allows admin to remove existing quizzes from the system.
Logout	Ends the admin session securely and exits the system.

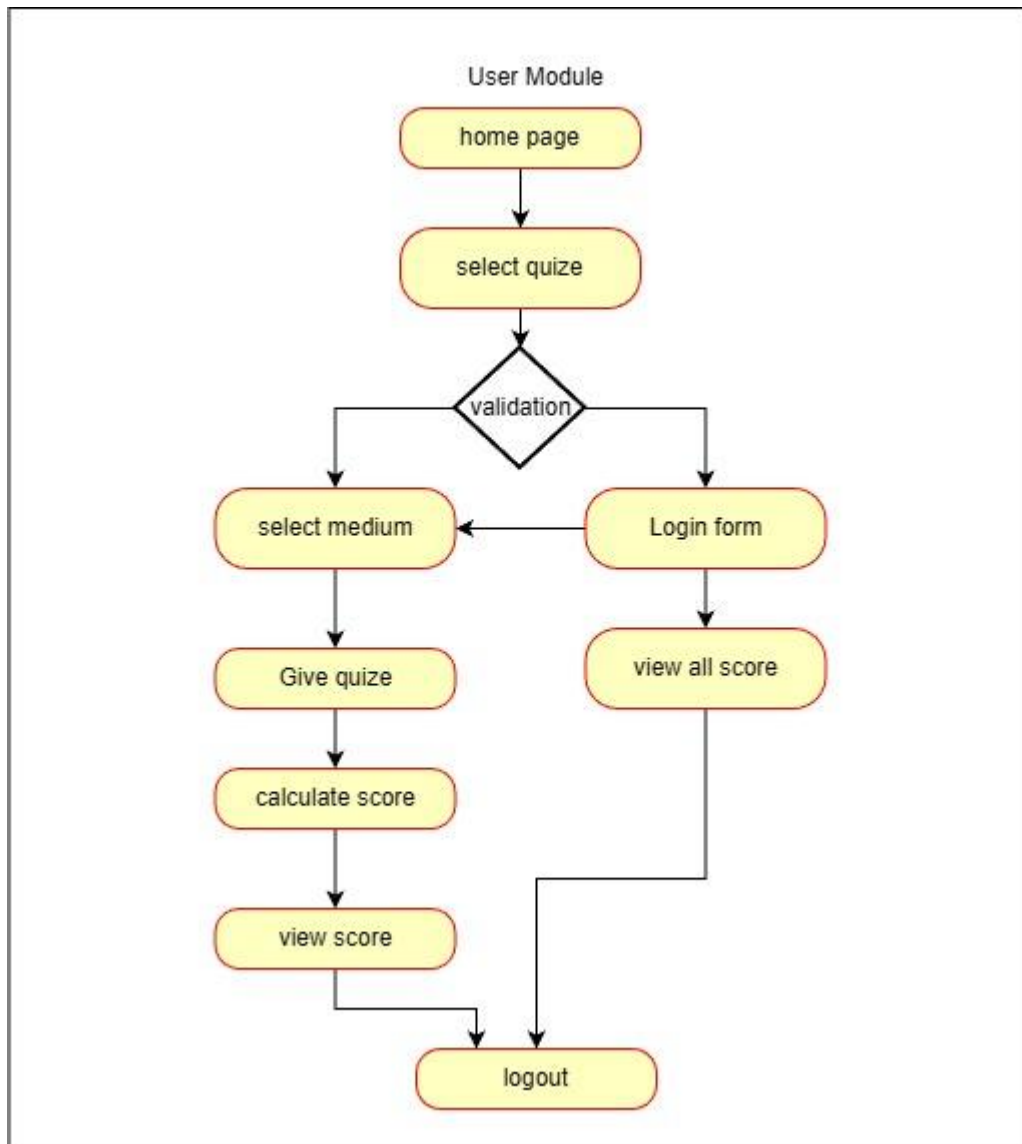
User module



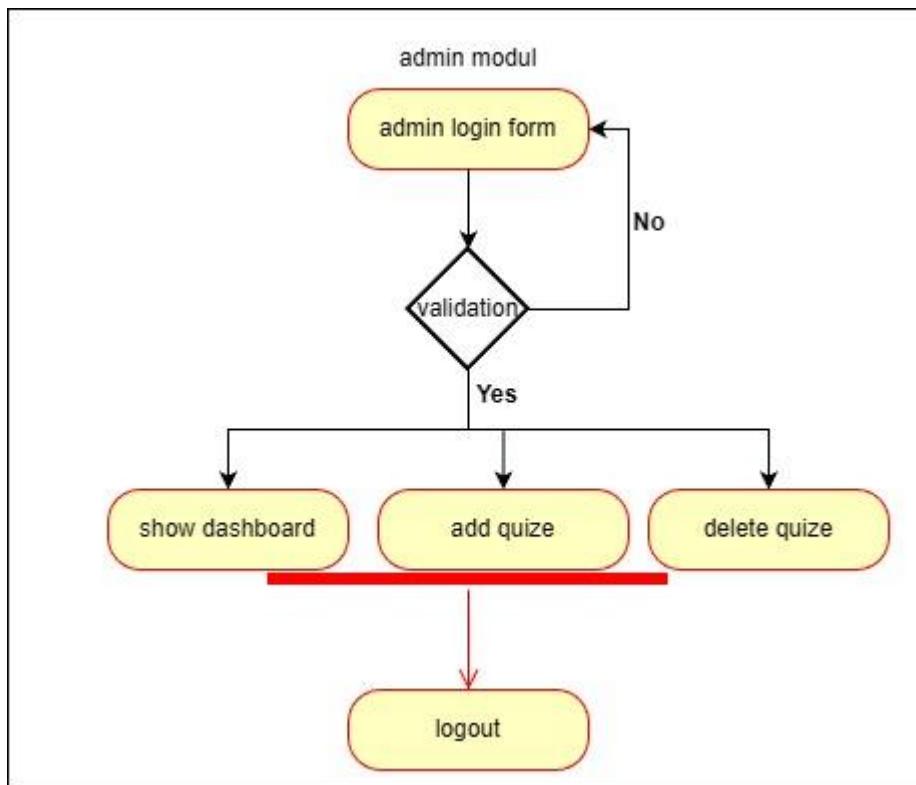
Field / Use Case	Description
Home Page	Entry point for users to access the quiz platform.
Login Page	Allows users to log in using their credentials.
Email, Password	Required fields included in the login process.
Forgot Password	Optional feature to recover or reset login credentials.
Select Difficulty	Enables users to choose the quiz level (e.g., Easy, Medium, Hard).
Attempt Quiz	Users participate in the quiz by answering questions.
View Score	Displays the user's performance and scores after quiz completion.
Logout	Ends the user session and exits the system securely.

3.2 Activity Diagram

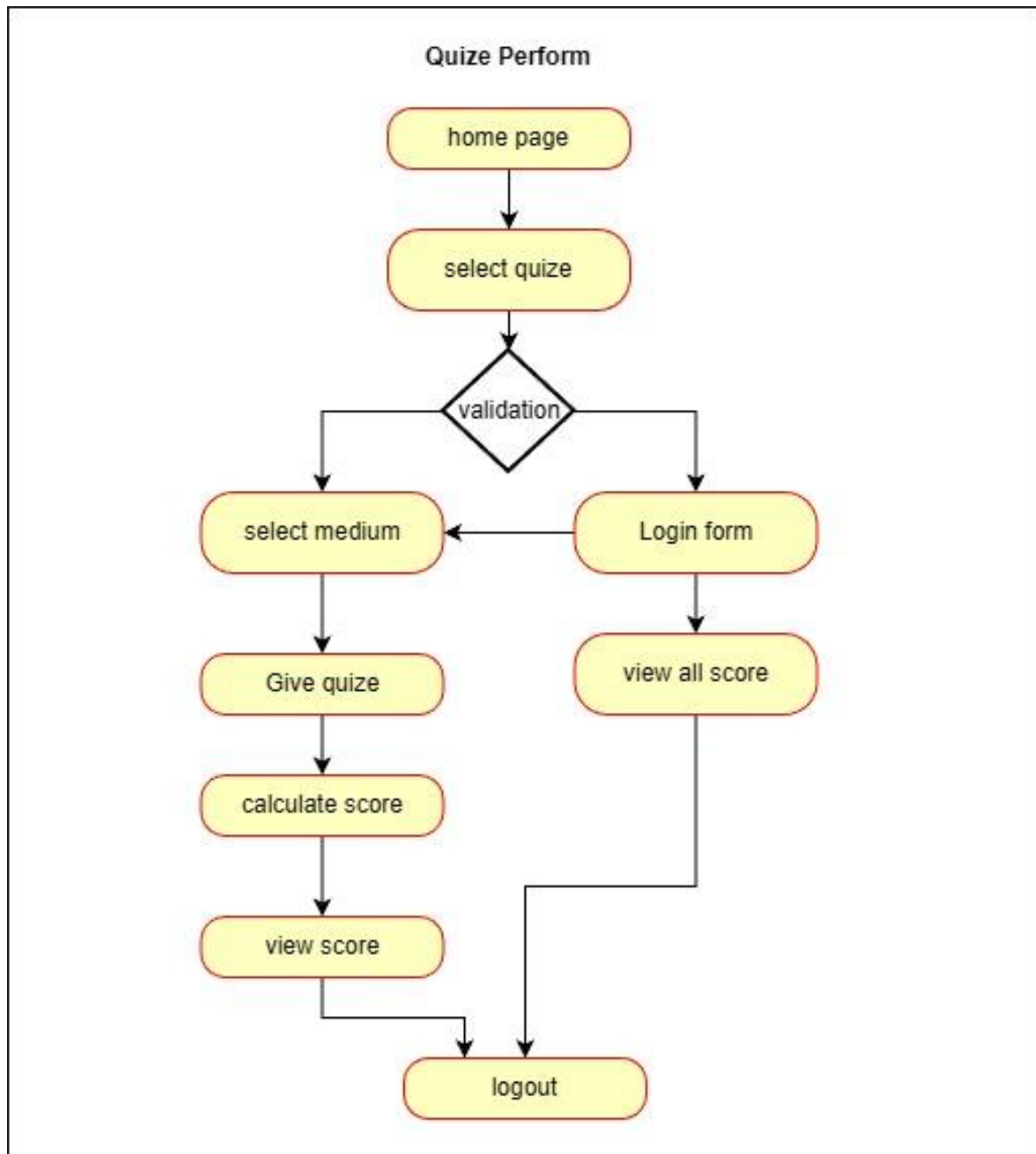
User module



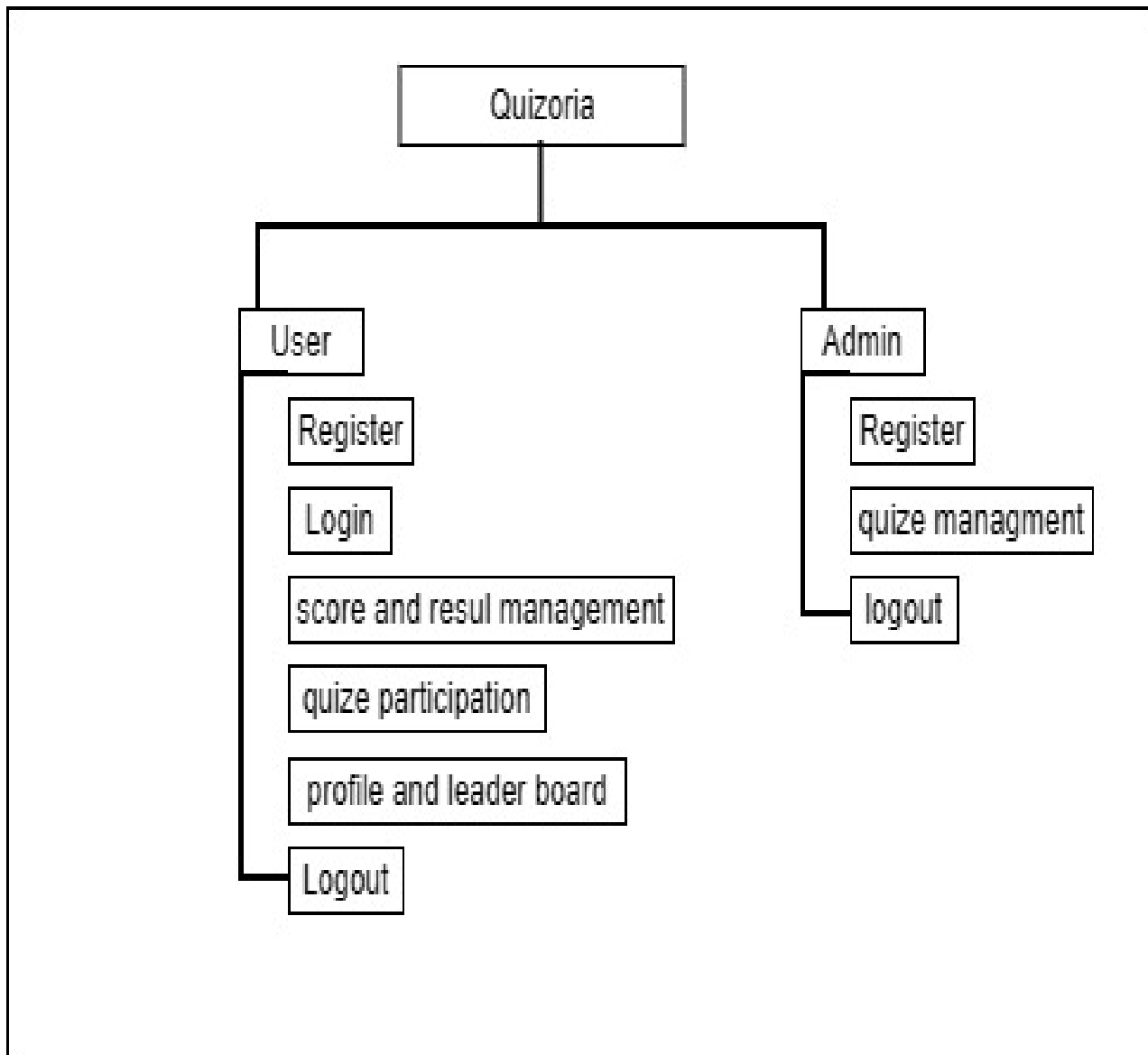
Admin module



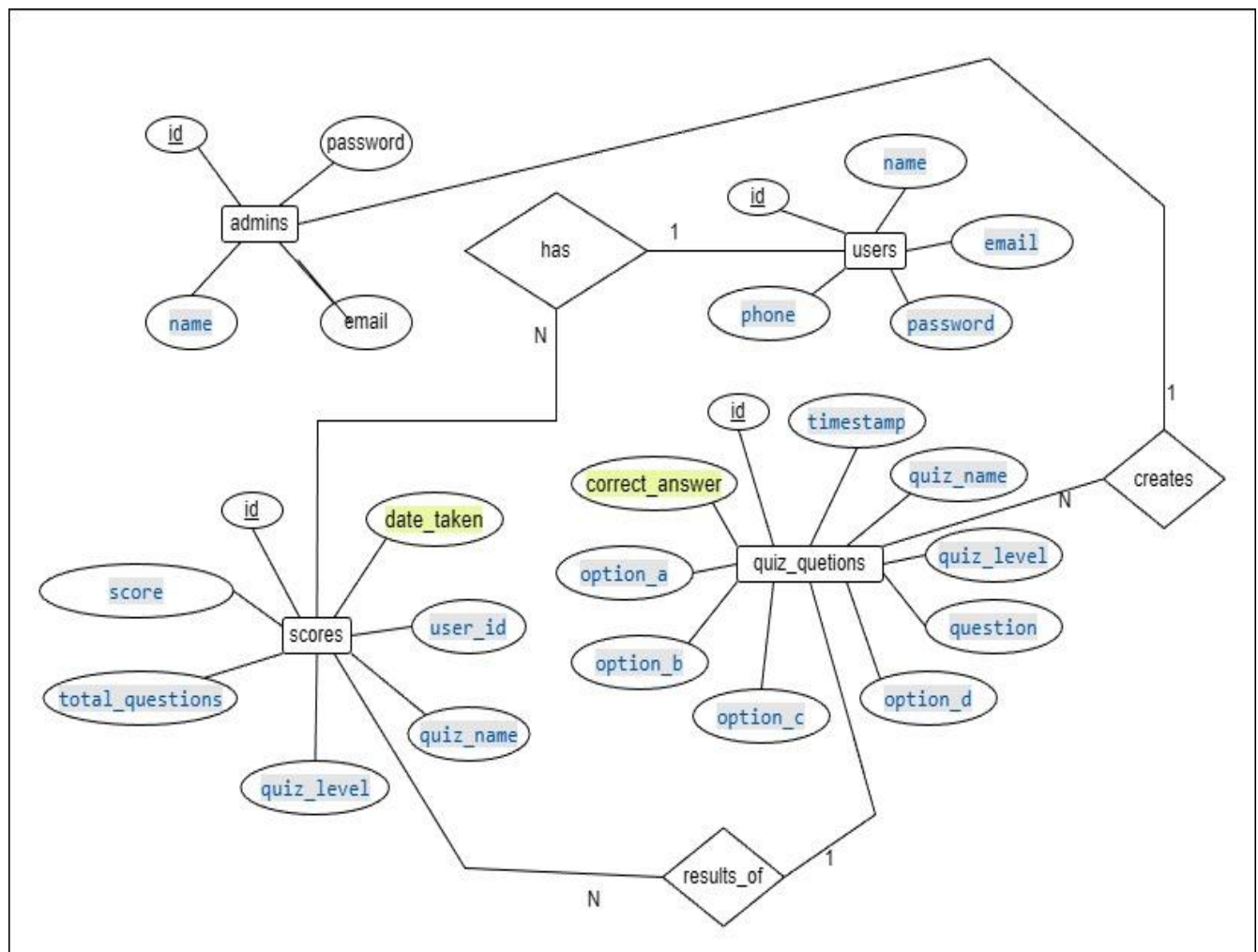
Quiz perform module



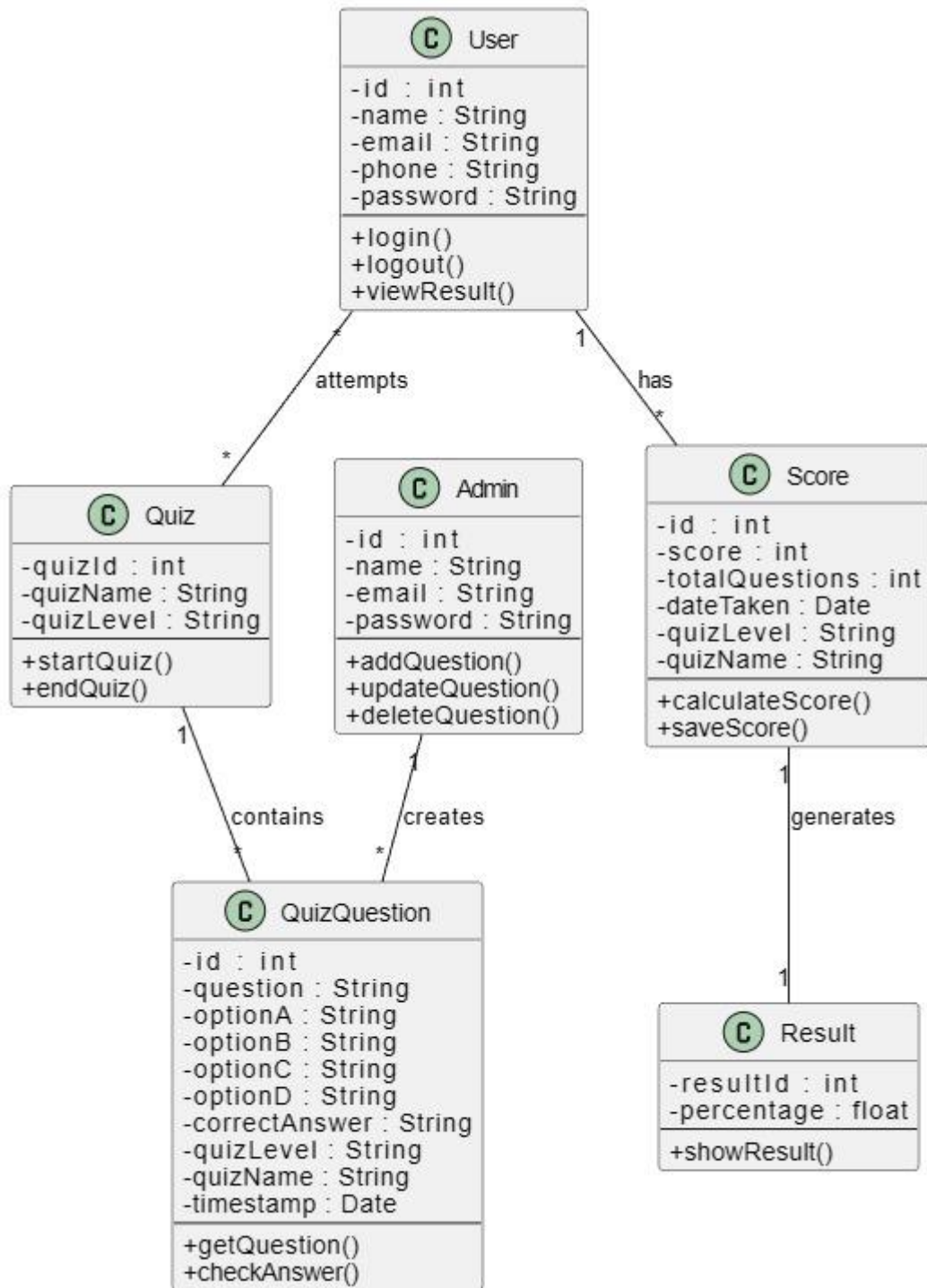
3.3 Module Hierarchy Diagram



3.4ERD



3.5 Class Diagram



3.6 Table specifications (Database design)

1) Table Name: Admin

Description: Information about admin

Primary Key: Admin_id

Field Name	Data Types	Size	Constraint	Description
Admin_id	INT	10	Primary Key	Generate Admin Id
Admin name	VARCHAR	10	Not Null	Used to save admin name
Adminpassword	VARCHAR	10	Not Null	Used to save admin password
adminmailid	VARCHAR	10	Not Null	Used to store admin mail id
Is_active	INT	10	Not Null	Used to check status of admin
Creation date	TIMESTAMP	—	Not Null	Used to save the login date of admin
Updation date	TIMESTAMP	—	Not Null	Used to save the updating date

2) Table Name: Forget_password

Description: Stores info of password reset

Primary Key: Pass_id

Field Name	Data Types	Size	Constraint	Description
Pass_id	INT	10	Primary Key	Generate password id of admin
Current password	VARCHAR	10	Not Null	Used to save current password of admin
Change password	VARCHAR	10	Not Null	Used to save changed password of admin

3) Table Name: Category

Description: Stores news categories

Primary Key: Id

Field Name	Data Types	Size	Constraint	Description
Id	INT	10	Primary Key	Enter category news id
Category name	VARCHAR	50	Not Null	Enter the main category of news
Description	VARCHAR	255	Not Null	Enter the description of the category
Posting date	DATE	—	Not Null	Generate posting date of category
Updation date	DATE	—	Not Null	Generate updating date of category
Is_active	VARCHAR	10	Not Null	Display news status

4) Table Name: Pages

Description: Stores static pages information

Primary Key: Id

Field Name	Data Types	Size	Constraint	Description
Id	INT	10	Primary Key	Enter id of the handling pages
Page name	VARCHAR	50	Not Null	Enter page name
Page title	VARCHAR	100	Not Null	Enter page title
Description	VARCHAR	255	Not Null	Enter the whole description about pages
Posting date	DATE	—	Not Null	Display posting date of the page
Updation date	DATE	—	Not Null	Display updating date of the page

5) Table Name: Posts

Description: Stores news posts

Primary Key: Id

Field Name	Data Types	Size	Constraint	Description
Id	INT	10	Primary Key	Enter post id
Posting title	VARCHAR	100	Not Null	Enter the title of the post
Categoryid	INT	10	Foreign Key	Select the category id of the news
Post details	VARCHAR	255	Not Null	Enter whole news post details
Subcategoryid	INT	10	Foreign Key	Generate sub category id
Posting date	DATE	—	Not Null	Generate posting date update
Updation date	DATE	—	Not Null	Generate update date
Posturl	VARCHAR	255	Not Null	Generate post url

6) Table Name: Subcategory

Description: Stores subcategories of news

Primary Key: Subcategoryid

Field Name	Data Types	Size	Constraint	Description
Subcategoryid	INT	10	Primary Key	Enter sub category id
Categoryid	INT	10	Foreign Key	Generate category id
Subcategory	VARCHAR	50	Not Null	Enter sub category name
Post details	VARCHAR	255	Not Null	Enter post details
Sub description	VARCHAR	255	Not Null	Enter sub description
Posting date	DATE	—	Not Null	Generate posting date
Is active	VARCHAR	10	Not Null	Generate active status

Data Dictionary

The Data Dictionary provides a structured overview of all database tables, their fields, data types, and constraints. It defines how information such as admin details, categories, pages, posts, and subcategories is stored and managed. This ensures clarity, consistency, and proper relationships across the database for smooth functioning of the project.

Sr No	Field Name	Description	Data Type	Constraint	Table Name
1	Admin_id	Enter admin id	INT	Primary Key	Admin
	Admin name	Enter admin name	VARCHAR	Not Null	
	Admin password	Enter admin password	VARCHAR	Not Null	
	adminmailid	Enter admin email id	VARCHAR	Not Null	
	Is_active	Showing active or not	INT	Not Null	
	Creation date	Display admin login creation date	TIMESTAMP	Not Null	
	Updating date	Display updating date of admin	TIMESTAMP	Not Null	
2.	Id	Enter category id	INT	Primary Key	Category
	Category name	Enter category name	VARCHAR	Not Null	
	Description	Enter description of category	VARCHAR	Not Null	
	Posting date	Generate posting date	DATE	Not Null	
	Updation date	Generate updating date	DATE	Not Null	
	Is_active	Display category status	VARCHAR	Not Null	
4	Id	Enter page id	INT	Primary Key	Pages
	Page name	Enter page name	VARCHAR	Not Null	
	Page title	Enter page title	VARCHAR	Not Null	
	Description	Enter description of page	VARCHAR	Not Null	
	Posting date	Display posting date	DATE	Not Null	
	Updation date	Display updating date	DATE	Not Null	
5	Id	Enter post id	INT	Primary Key	Posts
	Posting title	Enter post title	VARCHAR	Not Null	
	Categoryid	Select category id	INT	Foreign Key	
	Post details	Enter post details	VARCHAR	Not Null	
	Subcategoryid	Generate sub category id	INT	Foreign Key	
	Posting date	Generate posting date	DATE	Not Null	
	Updation date	Generate updating date	DATE	Not Null	

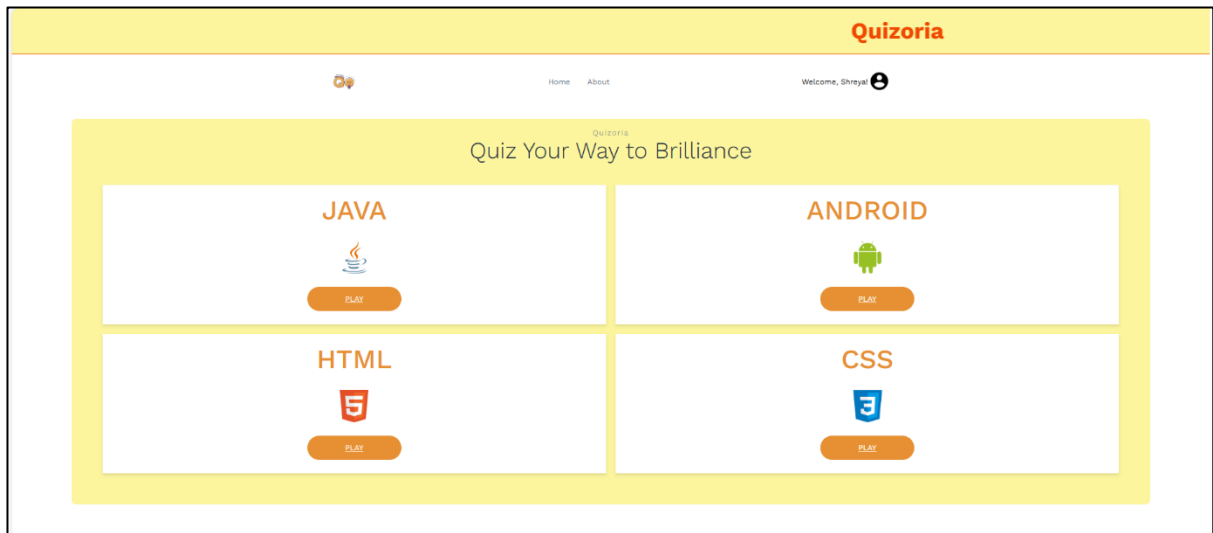
	Posturl	Generate post url	VARCHAR	Not Null	
6	Subcategoryid	Enter sub category id	INT	Primary Key	Subcategory
	Categoryid	Generate category id	INT	Foreign Key	
	Subcategory	Enter sub category name	VARCHAR	Not Null	
	Post details	Enter post details	VARCHAR	Not Null	

CHAPTER 4 : USER MANUAL

4.1 User Interface Screens (Input)

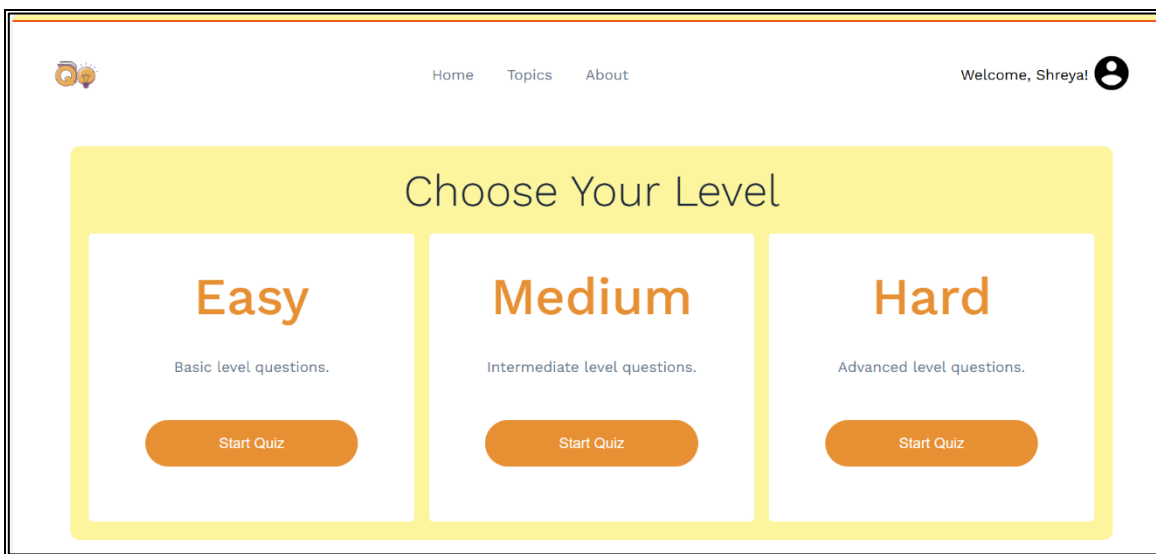
1. Landing Page

The landing page of Quizoria displays a vibrant interface with quick access to quiz categories like Java and Android, encouraging users to start learning instantly through clearly labeled "PLAY" buttons. It sets an engaging tone with the tagline Quiz Your Way to Brilliance.



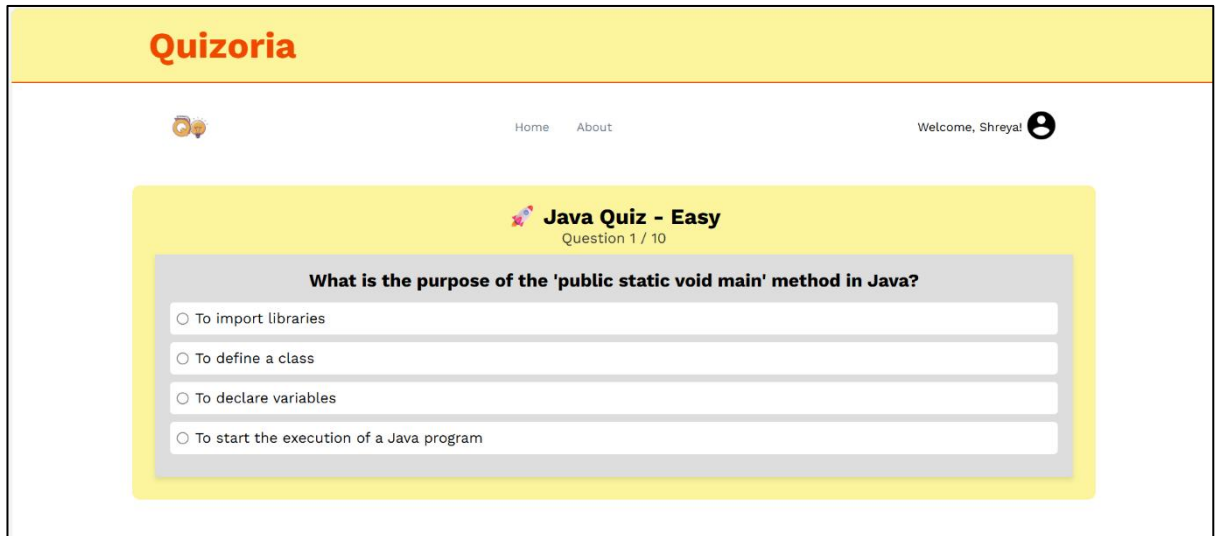
2. Level page

The level selection page allows users to choose their quiz difficulty—Easy, Medium, or Hard each with a brief description and a “Start Quiz” button. It offers a personalized experience and guides users based on their confidence and skill level.



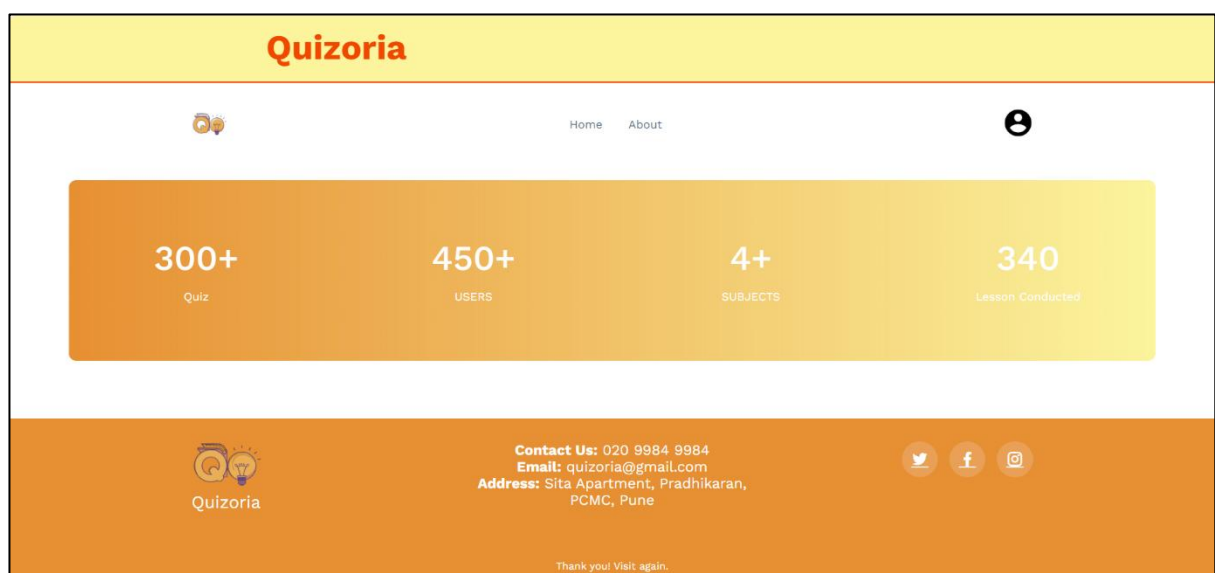
3. Quiz page

The quiz page presents an interactive interface where users answer multiple-choice questions with instant feedback. It highlights correct answers, tracks progress, and includes navigation options like “Next” to move through the quiz smoothly.



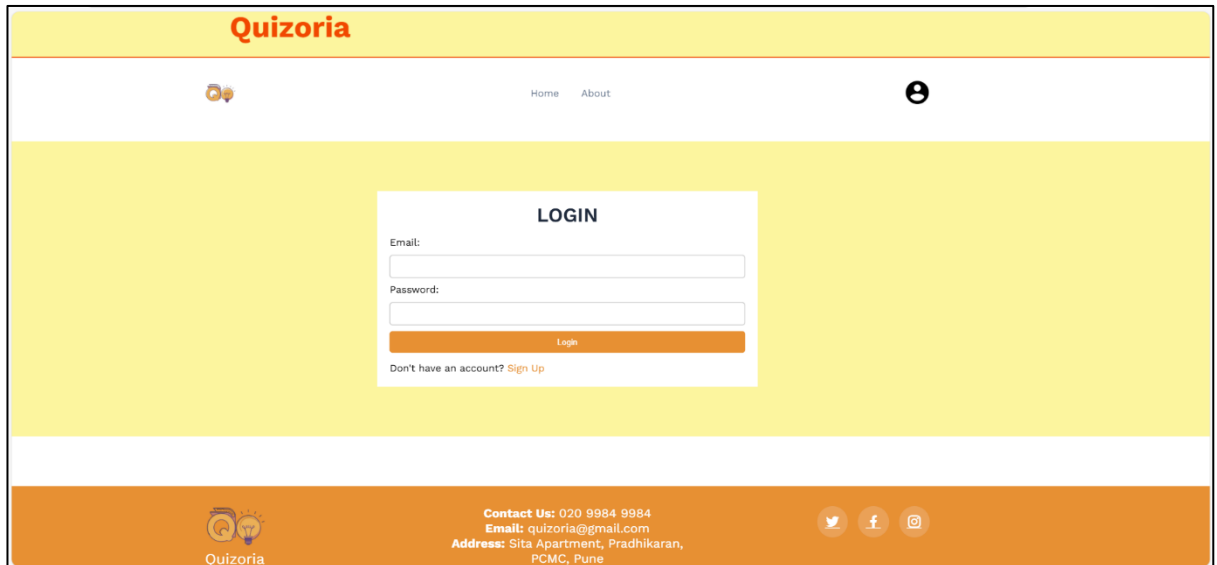
4. About page

The About page of Quizoria highlights key platform statistics such as total quizzes, users, subjects, and lessons conducted. It also provides contact details and a warm thank-you message, reinforcing trust and user engagement.



5. Login page

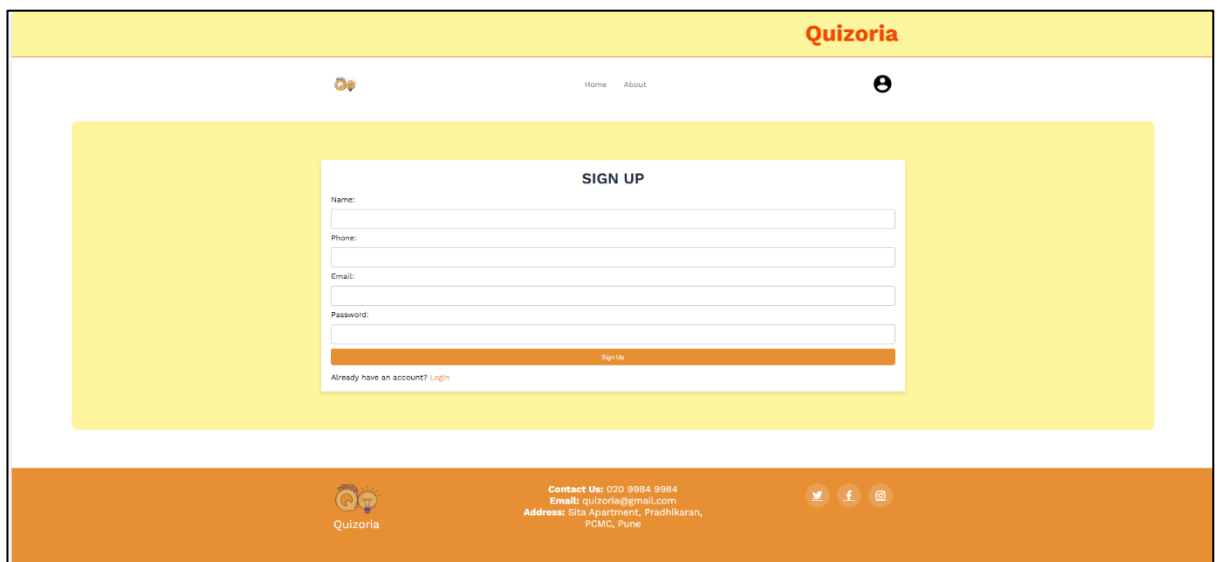
The login page offers a clean and user-friendly interface where users can securely enter their email and password to access the Quizoria platform. It also provides a quick link for new users to sign up and join the system.



The screenshot displays the Quizoria login page. At the top, a yellow header contains the 'Quizoria' logo. Below this, a white navigation bar features a Quizoria icon, 'Home' and 'About' links, and a user profile icon. The main content area has a yellow background with a white 'LOGIN' form in the center. The form includes 'Email:' and 'Paseword:' labels, corresponding input fields, and an orange 'Login' button. A link for 'Don't have an account? Sign Up' is positioned below the button. The footer is orange and contains the Quizoria logo, contact information (020 9984 9984, quizoria@gmail.com, Sita Apartment, Pradhikaran, PCMC, Pune), and social media icons for Twitter, Facebook, and Instagram.

6. Sign up page

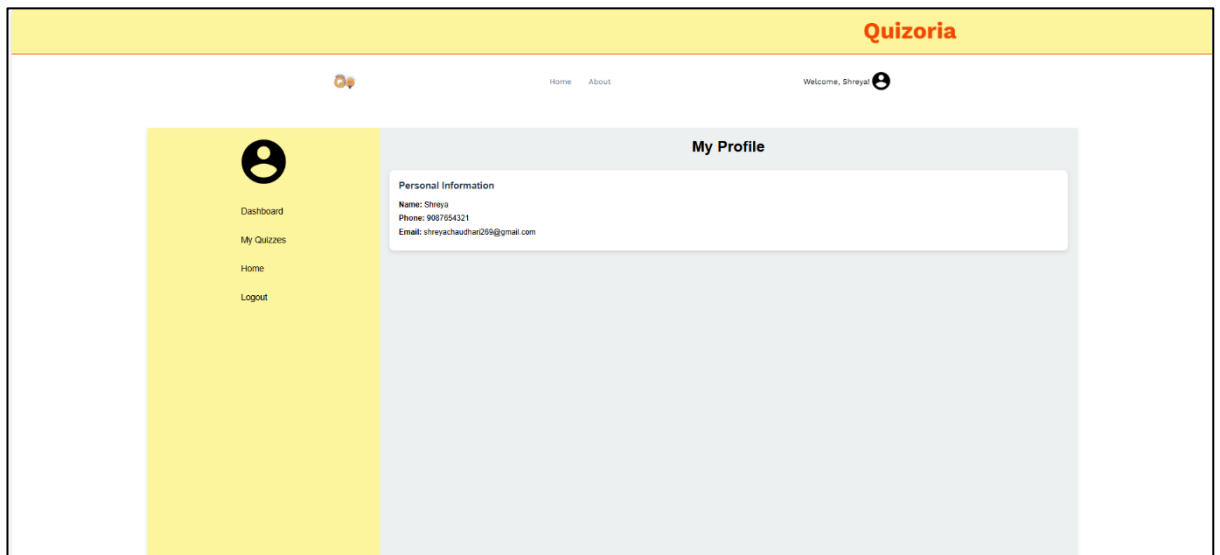
The sign-up page allows new users to create an account by entering their name, phone number, email, and password. It features a clean layout with easy navigation and a prompt for existing users to log in.



The screenshot displays the Quizoria sign-up page. It features a yellow header with the 'Quizoria' logo and a white navigation bar with a Quizoria icon, 'Home' and 'About' links, and a user profile icon. The main content area has a yellow background with a white 'SIGN UP' form in the center. The form includes 'Name:', 'Phone:', 'Email:', and 'Paseword:' labels, corresponding input fields, and an orange 'Sign Up' button. A link for 'Already have an account? Login' is positioned below the button. The footer is orange and contains the Quizoria logo, contact information (020 9984 9984, quizoria@gmail.com, Sita Apartment, Pradhikaran, PCMC, Pune), and social media icons for Twitter, Facebook, and Instagram.

7. Profile page

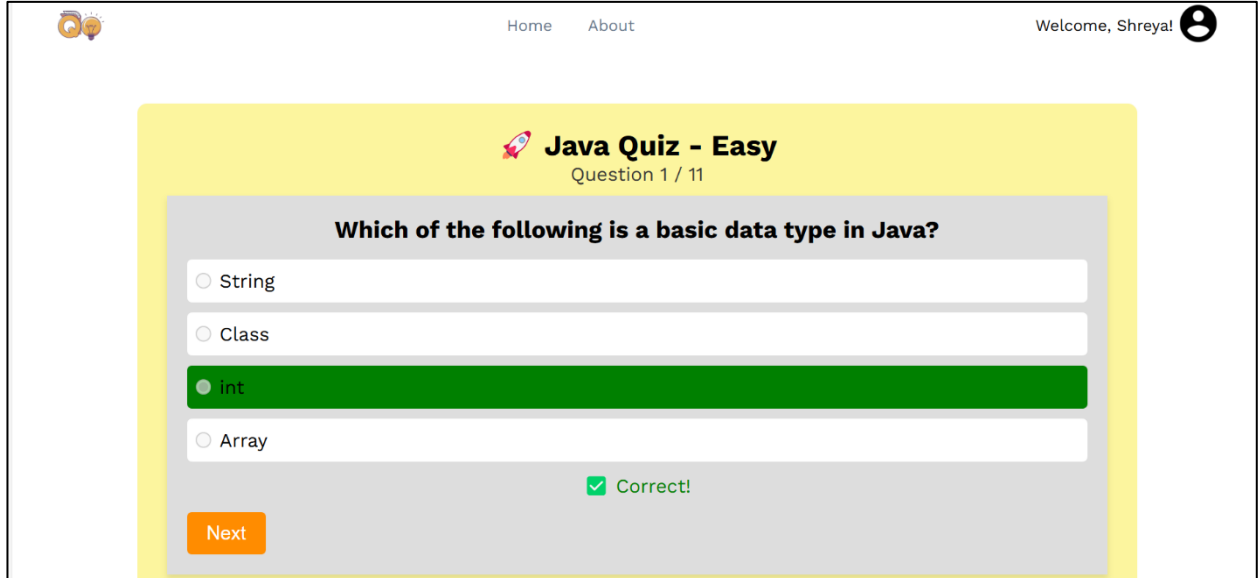
The profile page displays personalized user information including name, role, and email, along with sidebar navigation for accessing quizzes and dashboard features. It offers a clean layout and a welcoming interface for managing user activity.



4.2 Output Screens with data

1.Quiz page

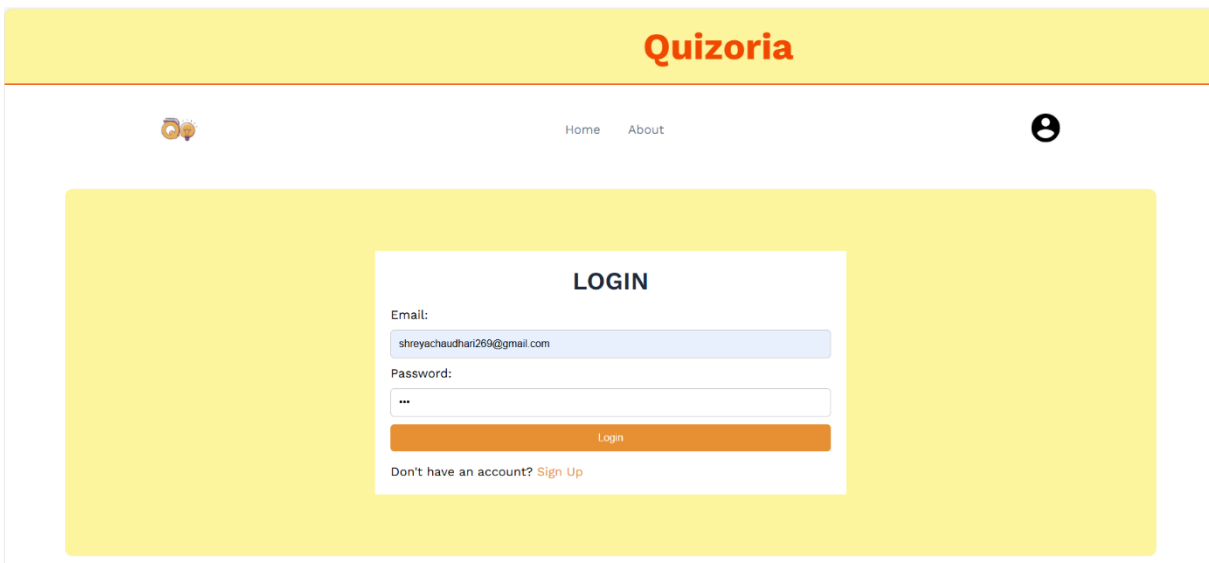
The quiz output screen displays Question 1 of 11 from the "Java Quiz - Easy" set, asking about basic data types in Java. The user selects "int" as the correct answer, which is confirmed with a green checkmark and "Correct!" message, along with a "Next" button to proceed.



The screenshot shows a web application interface for a quiz. At the top, there's a navigation bar with "Home" and "About" links, and a user greeting "Welcome, Shreya!" with a user icon. The main content area has a yellow background with the title "Java Quiz - Easy" and "Question 1 / 11". The question is "Which of the following is a basic data type in Java?". There are four radio button options: "String", "Class", "int" (which is selected and highlighted in green), and "Array". Below the options, there is a green checkmark and the text "Correct!". At the bottom left of the question area, there is an orange "Next" button.

2.Login page

The login output screen shows a clean form with prefilled email “shreyachaudhari269@gmail.com” and a password field, plus a prominent Login button. It includes Home/About navigation, a user icon, and a “Don’t have an account? Sign Up” link for quick onboarding.





The screenshot shows a web application interface for a login page. At the top, there's a yellow header with the "Quizoria" logo in orange. Below the header, there's a navigation bar with "Home" and "About" links, and a user icon. The main content area has a yellow background. In the center, there is a white login form with the title "LOGIN". The form has two input fields: "Email:" (prefilled with "shreyachaudhari269@gmail.com") and "Password:" (with a masked password "..."). Below the password field is an orange "Login" button. At the bottom of the form, there is a link "Don't have an account? Sign Up".

3. Sign up page

The sign-up output screen displays a filled registration form with user details like name “siya shinde,” phone number, email, and password. It includes a “Sign Up” button, navigation links, and footer contact info, offering a complete onboarding interface for new users.

Quizoria

Home About

SIGN UP


Name:

Phone:




Email:

Password:

Already have an account? [Login](#)



Contact Us: 020 9984 9984
Email: quizoria@gmail.com
Address: Sita Apartment, Pradhikaran,
PCMC, Pune



4.3 Report

1. Quiz questions report-

The quiz questions report provides a structured overview of the quiz content stored in the system's database. It includes a set of 10 easy-level questions focused on CSS (Cascading Style Sheets), each tagged with a timestamp and categorized under the "CSS" quiz name. Every question is accompanied by two multiple-choice options, labeled as option A and option B, designed to test basic understanding of CSS properties and usage. This report helps administrators verify the accuracy, relevance, and formatting of quiz entries. It also serves as a reference for future updates or additions to the quiz bank.

Database: quizzoria, Table: quiz_questions, Purpose: Dumping data

id	timestamp	quiz_name	quiz_level	question	option_a	option_b
1	2025-03-24 22:30:31	CSS	Easy	What does CSS stand for?	Cascading Style Sheets	Creative Style Sheets
2	2025-03-24 22:30:31	CSS	Easy	Which CSS property is used to set the color of text?	color	background-color
3	2025-03-24 22:30:31	CSS	Easy	What is the purpose of the "margin" property in CSS?	Set the padding of an element	Set the border of an element
4	2025-03-24 22:30:31	CSS	Easy	Which CSS property is used to set the font size of text?	font-size	font-style
5	2025-03-24 22:30:31	CSS	Easy	What is the purpose of the "background-image" property in CSS?	Set the background color of an element	Set the background image of an element
6	2025-03-24 22:30:31	CSS	Easy	Which CSS property is used to set the width of an element?	width	height
7	2025-03-24 22:30:31	CSS	Easy	What is the purpose of the "text-align" property in CSS?	Set the alignment of text	Set the color of text
8	2025-03-24 22:30:31	CSS	Easy	Which CSS property is used to set the border of an element?	border	margin
9	2025-03-24 22:30:31	CSS	Easy	What is the purpose of the "padding" property in CSS?	Set the space around an element	Set the space inside an element
10	2025-03-24 22:30:31	CSS	Easy	Which CSS property is used to set the height of an element?	width	height

2. Scores report

The scores report provides a detailed record of user performance across various quizzes in the system. Each entry captures the date and time of the quiz attempt, the user ID, quiz topic (such as Java, HTML, Android, or CSS), difficulty level, score obtained, and total number of questions. This data helps track individual progress, compare performance across topics, and analyze trends over time. It also supports administrators in identifying active users, evaluating quiz effectiveness, and planning future improvements. Overall, the report serves as a valuable tool for monitoring learning outcomes and engagement.

Database: quizzoria, Table: scores, Purpose: Dumping data

id	date taken	user id	quiz name	quiz level	score	total questions
11	2025-03-24 22:41:55	1	Java	Easy	2	10
12	2025-03-24 22:51:32	1	Java	Medium	1	10
13	2025-03-24 23:20:33	4	Java	Easy	2	10
14	2025-03-24 23:50:00	1	Java	Hard	0	10
15	2025-03-24 23:50:06	6	HTML	Easy	2	10
16	2025-03-24 23:54:31	6	HTML	Easy	4	10
17	2025-03-24 23:56:54	6	HTML	Easy	2	10
18	2025-03-25 12:17:26	7	Java	Easy	6	11
19	2025-03-26 11:42:47	4	Java	Medium	2	10
20	2025-04-01 11:38:51	8	Java	Easy	3	11
21	2025-10-31 11:40:16	1	Java	Easy	2	11
22	2025-11-04 04:17:40	9	Android	Easy	2	10
23	2025-11-04 22:48:22	9	Java	Medium	1	10
24	2025-11-04 22:51:14	9	Html	Medium	1	10
25	2025-12-05 01:30:32	10	Android	Medium	5	10
26	2025-12-05 04:26:57	10	Java	Hard	5	10
27	2025-12-05 04:28:08	10	Java	Medium	1	10
28	2025-12-06 13:01:07	11	Css	Medium	1	10
29	2025-12-06 15:20:29	9	Android	Medium	3	10
30	2025-12-06 15:36:59	9	Java	Easy	5	11
31	2025-12-06 15:46:56	9	Java	Medium	6	10
32	2025-12-06 22:10:06	9	Java	Medium	0	10
33	2025-12-06 22:52:57	9	Java	Medium	3	10
34	2025-12-06 22:56:16	9	Java	Medium	3	10
35	2025-12-09 00:21:55	9	Java	Medium	0	10
36	2025-12-09 00:22:52	9	Java	Easy	2	10
37	2025-12-09 00:23:36	9	Android	Medium	0	10
38	2025-12-10 17:54:55	9	Android	Medium	2	10
39	2025-12-13 16:21:45	9	CSS	Medium	2	10
40	2025-12-15 12:35:58	9	Java	Easy	2	10
41	2025-12-19 11:49:06	9	Java	Easy	1	10

3. Admin report

The admin report provides a summary of the registered administrators who manage the quiz system. It includes essential details such as admin ID, name, email, and login credentials. This data helps track who has access to the backend and ensures accountability in quiz creation, user management, and system maintenance. The report is useful for verifying authorized personnel and maintaining transparency in administrative operations. It also supports future audits and role-based access control within the platform.

Database: quizzoria, Table: admins, Purpose: Dumping data

id	name	email	password
1	Om Thite	omthite245@gmail.com	ommthite
2	Shreya Chaudhari	shreyachaudhari269@gmail.com	shreyachaudhari269

4. User report

The user report provides a comprehensive overview of all registered users within the Quizoria system. It includes key details such as user ID, name, email address, phone number, and password. This data helps administrators manage user accounts, monitor participation, and ensure secure access to quizzes. The report also supports tracking user engagement and identifying duplicate or inactive entries. It plays a vital role in maintaining the integrity and organization of the platform's user base.

Database: quizoria, Table: users, Purpose: Dumping data

id	name	email	phone	password
1	Om Thite	omthite245@gmail.com	9011010903	123
3	admin	om@hd.j	9087654321	123
4	Shreya	shreyachaudhari269@gmail.com	9087654321	123
5	Shravani Kulkarni	shravanikulkarni@gmail.com	09087654321	shrav
6	siya shinde	siya@gmail.com	90875347889	123
7	Shravni Kulkarni	shravanikulkarni08@gmail.com	9975505327	12345678
8	saara	saara29@gmail.com	09834039719	Saara@123
9	tejas	rshinde64470@gmail.com	8459102568	Rahul@123
10	rahul shinde	Rahul@gmail.com	0892930480928	Rahul@123
11	rahul shinde	rshinde64471@gmail.com	8459102568	Rahul@2577

4.5 Sample program code

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Admin Dashboard</title>
  <link rel="stylesheet" href="/home.css" />
</head>

<body>

<header th:replace="~{admin/header :: header}"></header>

<div class="admin-page">
  <div class="abovecontainer">
    <div class="admin-container">

      <!-- Sidebar -->
      <div class="sidebar">
        <nav>
          <ul>
```

```

        <li><a href="#" onclick="showSection('quizzes')">Quizzes</a></li>
        <li><a href="#" onclick="showSection('add-question')">Add
Question</a></li>
        <li><a href="#" onclick="showSection('view-questions')">View    &
Delete</a></li>
        <li><a href="/admin/logout">Logout</a></li>
    </ul>
</nav>
</div>

```

```
<!-- CONTENT -->
```

```
<div class="admin-content">
```

```
    <div class="admin-header">
```

```
        <h1>Admin Dashboard</h1>
```

```
    </div>
```

```
<!-- QUIZ COUNTS -->
```

```
<div class="quizzes active" id="quizzes">
```

```
    <h3>Quizzes & Question Counts</h3>
```

```
    <table>
```

```
        <thead>
```

```
            <tr>
```

```
                <th>Quiz Name</th>
```

```
                <th>Level</th>
```

```
                <th>Total</th>
```

```
            </tr>
```

```
        </thead>
```

```
        <tbody>
```

```
            <tr th:each="q : ${quizCounts}">
```

```
                <td th:text="${q[0]}"></td>
```

```
                <td th:text="${q[1]}"></td>
```

```
                <td th:text="${q[2]}"></td>
```

```
            </tr>
```

```
        </tbody>
```

```
    </table>
```

```
</div>
```

```
<!-- ADD QUESTION -->
```

```
<div class="add-question" id="add-question">
```

```
    <h3>Add New Question</h3>
```

```
    <p th:if="${message}" style="color:green" th:text="${message}"></p>
```

```
    <form method="post" action="/admin/add-question">
```

```
        <label>Quiz Name:</label>
```

```
        <select name="quiz_name" required>
```

```
            <option value="">Select Quiz</option>
```



```

        <option>Java</option>
        <option>Android</option>
        <option>CSS</option>
        <option>HTML</option>
    </select>

    <label>Level:</label>
    <select name="quiz_level" required>
        <option value="">Select Level</option>
        <option>Easy</option>
        <option>Medium</option>
        <option>Hard</option>
    </select>

    <label>Question:</label>
    <textarea name="question" required></textarea>

    <label>Option A:</label>
    <input type="text" name="option_a" required>

    <label>Option B:</label>
    <input type="text" name="option_b" required>

    <label>Option C:</label>
    <input type="text" name="option_c" required>

    <label>Option D:</label>
    <input type="text" name="option_d" required>

    <label>Correct Answer:</label>
    <select name="correct_answer" required>
        <option value="">Select</option>
        <option value="option_a">A</option>
        <option value="option_b">B</option>
        <option value="option_c">C</option>
        <option value="option_d">D</option>
    </select>

    <button type="submit">Add</button>
</form>
</div>

<!-- VIEW / DELETE QUESTIONS -->
<div class="view-questions" id="view-questions">
    <h3>View & Delete Questions</h3>

    <form method="get" action="/admin/view">
        <label>Quiz:</label>
        <select name="quiz">
            <option value="">Select</option>

```

```
    <option>Java</option>
    <option>CSS</option>
    <option>HTML</option>
    <option>Android</option>
</select>
```

```
<label>Level:</label>
<select name="level">
    <option value="">Level</option>
    <option>Easy</option>
    <option>Medium</option>
    <option>Hard</option>
</select>
```

```
<button type="submit">Filter</button>
</form>
```

```
<div class="question-cards">
    <div class="question-card" th:each="q : ${questions}">
        <p><b>Question:</b> <span th:text="${q.question}"></span></p>
        <p>A: <span th:text="${q.optionA}"></span></p>
        <p>B: <span th:text="${q.optionB}"></span></p>
        <p>C: <span th:text="${q.optionC}"></span></p>
        <p>D: <span th:text="${q.optionD}"></span></p>
        <p><b>Answer:</b> <span th:text="${q.correctAnswer}"></span></p>

        <button onclick="deleteQuestion([[${q.id}]], this)">Delete</button>
    </div>
</div>
```

```
</div>
</div>
</div>
</div>
```

```
<footer th:replace="~{admin/footer :: footer}"></footer>
```

```
<script>
    function showSection(id) {
        document.querySelectorAll('.quizzes, .add-question, .view-questions')
            .forEach(s => s.classList.remove('active'));

        document.getElementById(id).classList.add('active');
    }

    function deleteQuestion(id, btn) {
        fetch(/admin/delete/${id}, { method: "DELETE" })
    }
</script>
```

```
        .then(res => res.text())
        .then(msg => {
            alert(msg);
            btn.parentElement.remove();
        });
    }
</script>

</body>
</html>
```

4.6 Limitations and Bibliography

Limitations of the System

1. The system currently supports only a limited number of programming topics (HTML, CSS, Java, Android), which restricts its appeal to a broader audience of learners.
2. Quiz levels are confined to Easy, Medium, and Hard, which may not provide sufficient depth for advanced learners or enough guidance for complete beginners.
3. Personalized feedback is not fully implemented, meaning users may not receive detailed insights into their mistakes or tailored suggestions for improvement.
4. The system depends on a stable internet connection, making it inaccessible to users in areas with poor connectivity.
5. Mobile optimization is limited, as the platform is primarily web-based and may not deliver the best experience on smaller devices.
6. The current design lacks integration with external learning resources such as tutorials, documentation, or video lectures, which could support users beyond quizzes.
7. Scalability is restricted, as the backend may face challenges handling very high traffic or large volumes of quiz data without further optimization.
8. The gamification features are basic, with only leaderboards and badges; advanced features like streaks, daily challenges, or peer competitions are not yet available.

Future Enhancements of the System

1. Expand Topic Coverage

Introduce additional programming languages and technologies such as Python, C++, SQL, React, and Node.js to attract a wider range of learners and professionals.
2. Introduce Granular Difficulty Levels

Add sub-levels like Beginner, Intermediate, Advanced, and Expert to provide more precise guidance and challenge for users at different stages of learning.
3. Implement Personalized Feedback Engine

Develop a feedback system that analyzes user responses and offers tailored suggestions, explanations, and learning paths based on individual performance.
4. Enable Offline Quiz Access

Integrate offline functionality allowing users to download quizzes and attempt them without internet access, syncing results once reconnected.

5. Launch Mobile App Version

Design and deploy a mobile application optimized for Android and iOS to improve accessibility and user experience on smartphones and tablets.

6. Integrate External Learning Resources

Link quizzes to relevant tutorials, documentation, and video lectures from trusted platforms to support deeper learning beyond quiz attempts.

7. Optimize Backend for Scalability

Upgrade the backend architecture using load balancing, caching, and asynchronous processing to handle high traffic and large datasets efficiently.

8. Enhance Gamification Features

Add advanced gamification elements like daily challenges, streak rewards, peer battles, and seasonal competitions to boost engagement and retention.

BIBLIOGRAPHY

LINKS REFERED: -

For HTML, CSS and JavaScript: www.w3school.com

For Login Page: <https://w3codepen.com/html-css-login-form-page/>

HTML and CSS: Design and Built Websites by Jon Duckett