

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**JNANASANGAMA, BELGAVI-590018**



**DATABASE MANAGEMENT SYSTEM MINI PROJECT  
(BCS403)**

on

**“QUIZ MANAGEMENT SYSTEM”**

Submitted in partial fulfilment of the requirements for the 4<sup>th</sup> Semester  
**INFORMATION SCIENCE AND ENGINEERING**

**Submitted by**

**SHIRISHA MP                      1BI23IS112**

**SPOORTHY HR                    1BI23IS120**

**Under the guidance of**

**Dr. M. SHILPA**

Associate Professor

Dept. of ISE,

BIT, Bangalore.



**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING  
BANGALORE INSTITUTE OF TECHNOLOGY**

**K. R. Road, V.V. Puram, Bengaluru-560004**

**2024-2025**

**BANGALORE INSTITUTE OF TECHNOLOGY**

K.R. Road, V.V. Puram, Bengaluru -560004

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the implementation of **DBMS MINI PROJECT (BCS403)** entitled “**QUIZ MANAGEMENT SYSTEM**” has been successfully completed by **SHIRISHA MP (1BI23IS112)** and **SPOORTHY HR (1BI23IS120)** of IVth semester B.E. for the partial fulfilment of the requirements for the Bachelor's degree in Information Science & Engineering of the **Visvesvaraya Technological University** during the academic year 2024-2025.

**Course Co-Ordinator:**

**Dr . M.SHILPA**  
Associate Professor  
Dept. of ISE, BIT.

**Signature with date**

---

---

## ABSTRACT

The Quiz Management System is an interactive web application designed to facilitate the creation, management, and participation in quizzes. It allows users to sign up and log in to access various features. Admin users can create and manage quizzes, each consisting of 10 questions with multiple-choice options, ensuring a structured and dynamic quiz experience. Users can participate in pre-existing quizzes or those created by themselves, answering questions in real-time. The system also features scoring functionality to evaluate performance after completing a quiz.

The backend of the system is powered by Node.js, with routes handling user authentication and quiz management. The database is built on SQL, storing user data, quizzes, and results efficiently. The frontend offers a user-friendly interface using HTML, CSS, and JavaScript, featuring styled buttons, background images, and a smooth user experience. This system aims to provide an engaging platform for learning and testing knowledge across various subjects, making it accessible to a wide audience.

# CONTENTS

	<b>Page No.</b>
<b>Abstract</b>	<b>i</b>
<b>CHAPTER 1</b>	
1.1 Overview	1
1.2 Problem Statement	1-2
1.3 Objectives	2
1.4 Features of Proposed System	2-3
<b>CHAPTER 2 BACK-END DESIGN</b>	
2.1 JavaScript	4
2.2 Conceptual Backend Design	4-5
2.3 Logical Database Design	5-6
2.4 Normalisation	6-10
<b>CHAPTER 3 FRONT-END DESIGN</b>	
3.1 HTML	11
3.2 CSS	11
3.3 JavaScript	11
3.4 System Requirements	12
3.5 Connectivity	12
<b>CHAPTER 4 IMPLEMENTATION</b>	
4.1 MySQL	12
<b>CHAPTER 5 TESTING</b>	16-17

<b>CHAPTER 6</b>	<b>SNAPSHOTS</b>	18-20
<b>CHAPTER 7</b>	<b>APPLICATIONS</b>	21
<b>CHAPTER 8</b>	<b>CONCLUSION</b>	22
23		
	<b>REFERENCES</b>	

## LIST OF FIGURES

<b>Figure No.</b>	<b>Figure Name</b>	<b>Page No.</b>
2.1	ER Diagram	05
2.2	Flow Chart	06
6.1	Home Page	18
	Sign up Page	18
6.2	Login page	19
6.3	Dashboard	19
6.4	Create quiz	20
6.5	Result	20
6.6		

## LIST OF TABLES

<b>Table No.</b>	<b>Table Name</b>	<b>Page No.</b>
2.1	Quiz Attempts	07
2.2	User Information	08
2.3	User Information	08
2.4	User Information	08
2.5	Quiz Management	09
2.6	Responses	09
2.7	Attempts Table	10
5.1	Testing Validation	16



## **ABBREVIATIONS**

1. ER            - Entity Relationship
2. HTML       - Hyper Text Markup Language
3. JS           - Java Script
4. CSS          - Cascading Style Sheets



## CHAPTER 1

# INTRODUCTION

### 1.1 Overview

The **Quiz Management System** is a web-based application designed to simplify and enhance the process of conducting quizzes online. It serves both quiz creators—such as educators, trainers, or administrators—and quiz participants, like students or learners. The system provides a secure user authentication module where individuals can sign up and log in to access different functionalities based on their role. Once logged in, users can either create new quizzes or participate in existing ones.

For quiz creation, the platform allows users to input a quiz title, select a subject, and enter 10 multiple-choice questions, each with four options and one correct answer. This data is stored securely in a structured SQL database, making it easy to retrieve, edit, or delete quizzes as needed. Participants, on the other hand, can browse and select from a list of available quizzes based on their interest or subject area. The system then presents the quiz in a user-friendly format, displaying one question at a time with radio button options for answers.

Upon completion of the quiz, the system automatically evaluates the responses and displays the user's score. This immediate feedback helps users assess their performance and identify areas for improvement. Results can also be stored for later review. The frontend of the system is developed using HTML, CSS, and JavaScript, offering a clean and responsive user interface with visually appealing elements like styled buttons and background images.

On the backend, Node.js with Express.js handles server-side logic, including user authentication, quiz management, and result processing. The application connects to a MySQL database that maintains user data, quiz content, and scoring records in an organized schema. Overall, the Quiz Management System is a reliable, scalable, and interactive solution for conducting online assessments, suitable for educational institutions, training platforms, or individual educators.

### 1.2 Problem Statement

In educational and training environments, conducting and managing quizzes manually is often time-consuming, inefficient, and prone to errors. Traditional methods of quiz administration—such as paper-based tests or unstructured digital formats—lack automation, centralized control. Educators face difficulties in creating, organizing, and grading quizzes, while learners have limited opportunities for interactive, self-paced assessment.

Additionally, there is often no proper record-keeping or accessibility for reviewing past performance.

There is a clear need for a digital solution that simplifies quiz creation, enables seamless participation, and provides immediate results, all within a user-friendly interface. Such a system should support secure login, efficient question management, and reliable result computation and storage. Therefore, this project proposes the development of a **Quiz Management System**—a full-stack web application that addresses these challenges by offering a centralized, automated, and accessible platform for quiz-based assessments.

### 1.3 Objectives

Here are the **objectives** of the **Quiz Management System** listed in clear points:

1. To design and develop an interactive web-based platform for managing quizzes.
2. To provide secure user authentication including sign-up and login features.
3. Enable quiz creators to add quizzes with 10 multiple-choice questions and one correct answer per question.
4. To allow users to play available quizzes based on subjects or titles.
5. To automate quiz evaluation and display scores immediately after completion.
6. To store user information, quiz data, and results in a structured SQL database.
7. To build a responsive and user-friendly frontend using HTML, CSS, and JavaScript.
8. To implement backend logic using Node.js and Express.js for handling routes and server operations.
9. To allow users to view and track their past quiz attempts and scores.
10. To create a scalable system that can be expanded with advanced features in the future.

### 1.4 Features of Proposed System

Here are the **features of the proposed Quiz Management System** summarized in 6 key points:

1. **User Authentication** – Secure login and registration system to manage user access and roles (admin or participant).

2. **Quiz Creation** – Allows admins to create quizzes with 10 multiple-choice questions, each with four options and one correct answer.
3. **Quiz Participation**–Users select available quizzes by subject and participate interactively.
4. **Automated Scoring** – The system automatically evaluates answers and displays the final score immediately after submission.
5. **Database Integration** – A structured SQL database is used to store user details, quizzes, and results for reliable data management.

## CHAPTER 2

### BACK-END DESIGN

#### 2.1 JavaScript

##### 2.1.1 User Authentication Module

Handles user registration and login functionality.

- **Signup Endpoint (/signup)**  
Allows new users to register by providing a unique username and password.
  - Checks for duplicate usernames.
  - Stores valid users in an in-memory array.
- **Login Endpoint (/login)**  
Authenticates users by matching submitted credentials with stored users.
  - Returns success message if credentials match.
  - Sends error if credentials are invalid.

#### 2.2 Conceptual Backend Design

The backend design for this Quiz Management System using Node.js and Express.js includes the following key elements:

- **Users**  
Stored in an array with username and password fields.
- **Quizzes**  
Represented as an object (can be extended) to hold quiz data per user or subject.  
Placeholder for future quiz creation and retrieval logic.

## 2.3 Server Configuration

- Libraries Used:
  - express: To create HTTP server and routes.
  - cors: To handle Cross-Origin Resource Sharing.
- Middleware Setup:
  - express.json(): Parses incoming JSON requests.
  - cors(): Allows communication with frontend on different origins.

## ER DIAGRAM FOR QUIZ MANAGEMENT SYSTEM

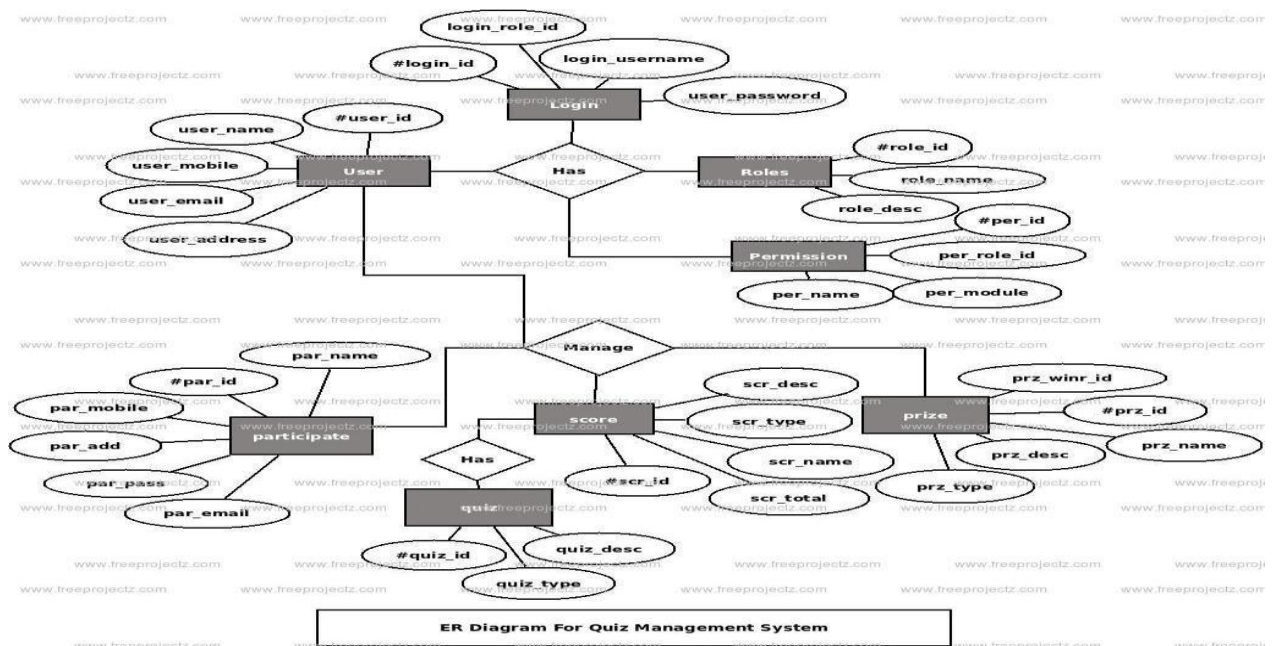


Figure 2.1 ER Diagram

## 2.3 Logical Database Design

The Logical Database Design for the Quiz Management System (QMS) organizes the data in a way that ensures efficiency and data integrity while supporting the core functionalities of the system. The system consists of several interrelated tables that manage user data, quizzes, questions, attempts, responses, and notifications. The relationships between these tables are defined using foreign key constraints to ensure that the data across these entities remains consistent and interconnected.

## QUIZ MANAGEMENT SYSTEM

The user\_info diagram as shown in Figure.2.1 serves as the core entity for managing user details, including information such as UserID, Username, Name, Gender, Password, and Avatar. This table stores the personal information of all users, such as students and admins. The UserID is the primary key, and it plays a critical role in linking the user to other tables within the system. Each user can have multiple interactions with the system, including creating quizzes, attempting quizzes, and receiving notifications. Thus, the user\_info table forms the foundation for tracking user activities throughout the system.

The Figure.2.1 also explains about the quiz\_info that stores the details of each quiz created in the system. This includes attributes such as QuizID, Title, Subject, Description, CreatedBy, DateCreated, TimeLimit, and TotalQuestions. The CreatedBy field references the UserID from the user\_info table, establishing a one-to-many relationship between the user\_info and quiz\_info tables. This means that one user (typically an admin or teacher) can create multiple quizzes.



Additionally, Figure.2.2 can be referred for flow chart, which helps in better understanding of the flow of the management system. The quiz\_info table is central to linking quizzes to the various questions that are part of each quiz.

## FLOW CHART FOR QUIZ MANAGEMENT SYSTEM

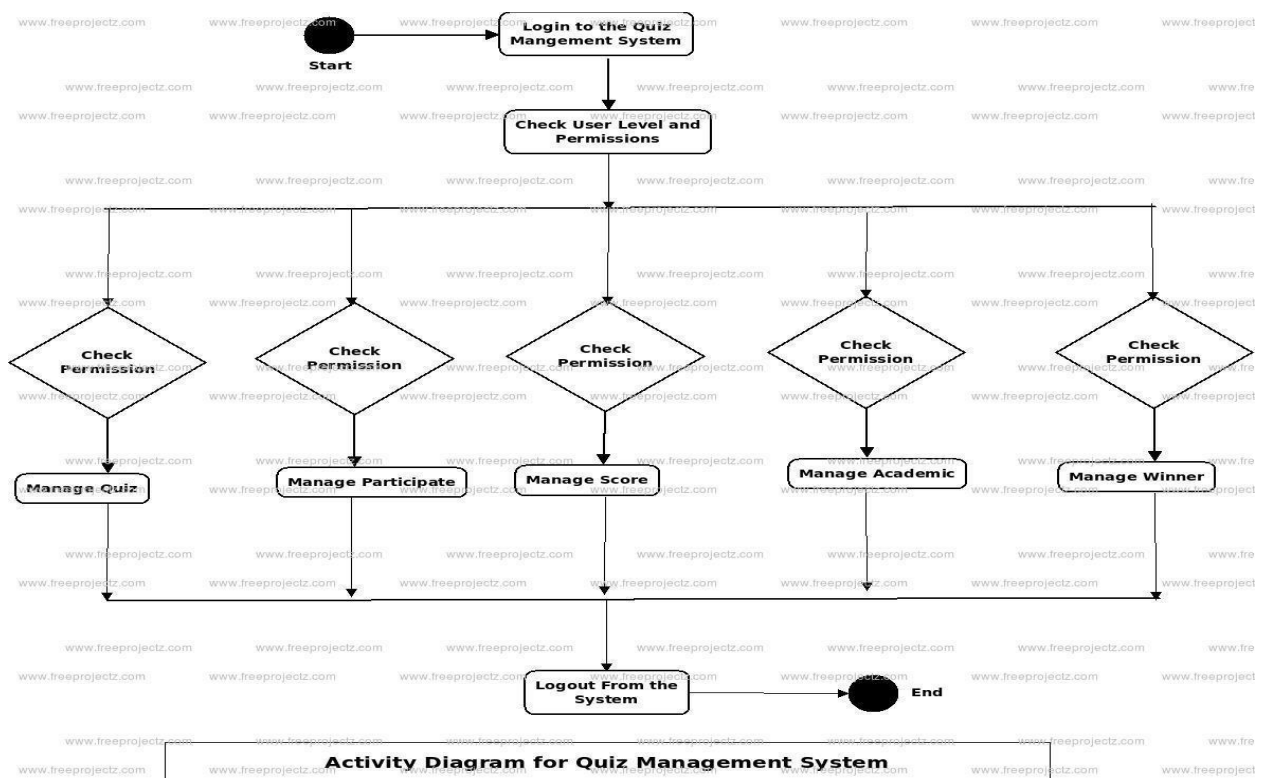


Figure 2.2 Flow Chart

## 2.4 Normalisation

Database Normalization is a technique of organizing the data in the data base. Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update, Deletion Anomalies. It is a multistep process that puts data into tabular form by removing duplicated data from the relational tables.

Normalization is used for mainly two purposes:

- Eliminating redundant(useless) data.
- Ensuring data dependencies make sense i.e., data id logically stored.

**2.4.1 First Normal Form (1NF):**

As per First Normal Form

- a) There are no duplicated rows, as shown in the Table 2.1
- b) Each cell is single valued or atomic.

**2.4.2 Second Normal Form (2NF):**

As per Second Normal Form, the Tables 2.2 , 2.3 and 2.4 are in 2NF if every non-prime attribute is not partially dependent on any key of the table.

**2.4.3 Third Normal Form (3NF):**

Third Normal Form applies that every non-prime attribute of table must be dependent on primary key, or we can say that, there should not be the case that a non-prime attribute is determined by another non-prime attribute . So, this transitive functional dependency should be removed from the table and also the table must be in the Second Normal Form.

The normalization process involves dividing large tables into smaller, related tables and defining relationships between them. Let's examine the normalization of the given database, focusing on the tables: admin, user information table ,club and event management table, relationship and membership table, communication table and notification table.

1. QuizAttempts:

This table appears to be already in a normalized form. The primary key (id) uniquely identifies each userID, and there are no apparent issues with redundancy.

**Table 2.1 :QuizAttempts**

AttempID	UserID	QuestionText	SelectedOption
1	101	What is 2+2?	B

1 NF: The Table2.1 satisfies 1NF as all attribute have atomic value

## 2. User Information Table:

The user information table seems to be well-structured with a primary key (id).

**Table 2.2 : User Information**

Id	username	name	gender	password	avatar
1	Abc123	KumKum	Female	123456	A
2	Def234	Megha	Female	123456	A

**Table 2.3 : User Information**

GenderID	gender
1	Abc123
2	Def234

:

**Table 2.4 User Information**

AvatarID	Avatar
1	A
2	A

2 NF: The table satisfies 2NF as there are no partial dependencies.

### 3. Quiz Management Table:

The Quiz management table should have a composite primary key consisting of id, StudentId, and SubjectId. This ensures uniqueness for each result entry.

**Table 2.5: Quiz management**

Clubid(PK)	Title	Subject	Description	Created by	Date created	Time limit (minutes)	Total Questions	Difficulty
1	Math	Math	Basic math concepts	1	2025-05-01	30	10	easy
...	...	...	...	...	...	...	...	...

1 NF: Table 2.5 satisfies 1NF as all attributes have atomic values.

2 NF: Table 2.5 satisfies 2NF as there are no partial dependencies.

3 NF: Table 2.5 satisfies 3NF as there are no transitive dependencies.

### 4. Responses

Table:

**Table 2.6 : Responses**

id (PK)	Attempt ID	Question ID	SelectedOption	AnswerTime(seconds)
1	201	1	C	45
...	...	...	...	...

1 NF: Table 2.6 satisfies 1NF as all attributes have atomic values.

2 NF: Table 2.6 satisfies 2NF as there are no partial dependencies.

### 5. Attempts Table:

The Attempts table tracks user participation in quizzes. It records each attempt made by a user, including the quiz they attempted, the user who took it, the date of the attempt, their total score, and the status of the attempt (e.g., completed or in-progress). This helps monitor user progress and quiz performance over time.

**Table 2.7 : Attempts Table**

Id(PK)	UserID	Quiz-Id	AttemptDate	TotalScore
1	1	101	2025-05-03	90
...		...		...

1 NF: Table 2.7 satisfies 1NF as all attributes have atomic values.

2 NF: Table 2.7 satisfies 2NF as there are no partial dependencies.

3 NF: Table 2.7 satisfies 3 NF as there are no transitive dependencies.



## CHAPTER 3

# FRONT-END

### 3.1 HTML

Hypertext Markup Language (HTML) is the main markup language for creating web pages and other information that can be displayed in a web browser.

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like `<html>`), within the web page content. HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some tags represent empty elements and so are unpaired, for example `<img>`.

### 3.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document.

### 3.3 JavaScript

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multiparadigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. It is a programming language that is primarily used to create interactive and dynamic web page content. It allows developers to create things such as form validation, image sliders, and other interactive elements on a website. Alongside HTML and CSS, JavaScript is one of the core technologies of the Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behavior and all major web browsers have a dedicated JavaScript engine to execute it.

### **3.4 System Requirements**

#### **Software Requirements:**

- Browser
- VS Code
- MySQL Version 5.5 or above
- Latest browser : Chrome, Firefox, Safari etc
- Operating System : Any (Linux, Windows, Mac etc)

### **3.5 Connectivity**

Node server.js is used to connect frontend and backend.



## CHAPTER 4

### IMPLEMENTATION

#### 4.1 MySQL

##### -- Create database

```
CREATE DATABASE IF NOT EXISTS quiz_management;
```

##### USE quiz\_management;

##### -- Users Table

```
CREATE TABLE IF NOT EXISTS users ( id INT  
AUTO_INCREMENT PRIMARY KEY, username  
VARCHAR(255) NOT NULL UNIQUE, password  
VARCHAR(255) NOT NULL,  
role ENUM('admin', 'user') DEFAULT 'user', -- Defines roles for users (Admin or regular  
user)  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

##### -- Quizzes Table

```
CREATE TABLE IF NOT EXISTS quizzes ( id  
INT AUTO_INCREMENT PRIMARY KEY,  
title VARCHAR(255) NOT NULL, subject  
VARCHAR(255) NOT NULL, created_by INT  
NOT NULL,  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
FOREIGN KEY (created_by) REFERENCES users(id) ON DELETE CASCADE --  
Admin or creator of the quiz  
);
```

**-- Questions Table**

CREATE TABLE IF NOT EXISTS questions ( id  
INT AUTO\_INCREMENT PRIMARY KEY,

---

quiz\_id INT, question\_text TEXT  
NOT NULL, option\_1 VARCHAR(255)  
NOT NULL, option\_2 VARCHAR(255)  
NOT NULL, option\_3 VARCHAR(255)  
NOT NULL, option\_4 VARCHAR(255)  
NOT NULL,  
correct\_option INT NOT NULL, -- 1, 2, 3, or 4 based on the correct answer  
FOREIGN KEY (quiz\_id) REFERENCES quizzes(id) ON DELETE CASCADE  
);

**-- Answers Table (optional, for recording individual answers if needed)**

CREATE TABLE IF NOT EXISTS answers ( id INT  
AUTO\_INCREMENT PRIMARY KEY, user\_id INT, question\_id  
INT, selected\_option INT,  
FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE,  
FOREIGN KEY (question\_id) REFERENCES questions(id) ON DELETE CASCADE  
);

**-- Results Table**

CREATE TABLE IF NOT EXISTS results ( id  
INT AUTO\_INCREMENT PRIMARY KEY,  
user\_id INT, quiz\_id INT, score INT,  
completed\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
FOREIGN KEY (user\_id) REFERENCES users(id),  
FOREIGN KEY (quiz\_id) REFERENCES quizzes(id)  
);

-- **Sample data for users (Admin and regular users)**

INSERT INTO users (username, password, role) VALUES

```
('admin', 'admin_password', 'admin'),  
('user1', 'password123', 'user'),  
('user2', 'password123', 'user');
```

**-- Sample data for quizzes**

```
INSERT INTO quizzes (title, subject, created_by) VALUES  
('JavaScript Basics Quiz', 'Programming', 1),  
('Math Quiz', 'Mathematics', 1);
```

**-- Sample data for questions (based on the quizzes)**

```
INSERT INTO questions (quiz_id, question_text, option_1, option_2, option_3, option_4,  
correct_option) VALUES  
(1, 'What is JavaScript?', 'A programming language', 'A movie', 'A framework', 'A music genre',  
1),  
(1, 'What does DOM stand for?', 'Document Object Model', 'Domain Object Model',  
'Document Operation Model', 'Direct Object Model', 1),  
(2, 'What is 2 + 2?', '3', '4', '5', '6', 2),  
(2, 'What is the square root of 16?', '3', '4', '5', '6', 2);
```

**-- Sample results (user quiz results)**

```
INSERT INTO results (user_id, quiz_id, score) VALUES  
(2, 1, 2), -- User1 completed the JavaScript Basics Quiz with 2/2 correct answers (3,  
2, 1); -- User2 completed the Math Quiz with 1/2 correct answers
```

**TESTING**

Here's how Testing Validation could look for the Quiz Management System, as shown in the Table 5.1:

**Table 5.1 : Testing Validations**

<b>Sl. no</b>	<b>INPUT</b>	<b>OUTPUT/ BEHAVIOR</b>	<b>REMARKS</b>
1	Open the application (URL)	Displays the homepage with options for login, signup, and viewing quizzes	Quiz management system is launched
2	Admin Login: Without credentials	Invalid credentials message	Login denied for missing credentials
3	User Login with wrong credentials	Invalid Credentials	Login is denied with incorrect credentials
4	User Login with correct credentials	User Dashboard is displayed, showing quiz options	User login is successful
5	Admin creates a new quiz with valid title and subject	Quiz is added to the database and is visible in the Admin dashboard	Quiz creation is successful
6	Admin creates a new quiz with missing fields (e.g., no title)	Error message prompting to fill all fields	Quiz creation failed due to missing files.

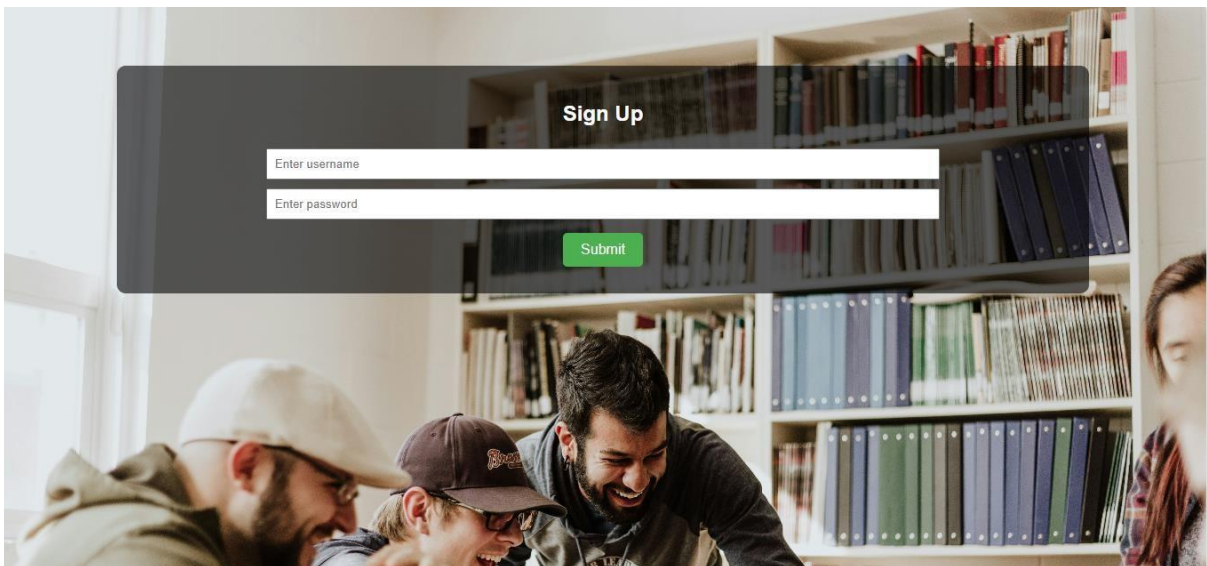
In the first set of test cases, the focus is on retrieving, inserting, updating, and searching operations within the database. In the retrieval scenario, a query is executed to gather comprehensive information about classes, emphasizing the need for accurate display of class names, numeric identifiers, and sections. This test ensures that the database successfully stores and retrieves class-related data. Moving to the insertion test case, a new student is added to the `user-information` table, and the expected result involves confirming the accurate incorporation of student details, including their name, roll ID, email, gender, date of birth, and the associated class. This validates the database's capability to handle new data entries seamlessly.

In the second set of test cases, the emphasis is on deletion, data integrity, and triggers. The deletion test involves removing a class from the `user-information` table, triggering the automatic removal of corresponding students from the `user-information` table and their associated results from the `club and event management` table. This validates the functionality of the implemented triggers, ensuring data consistency. The update test case involves modifying the marks of a student in a specific subject within the `communication table` table, confirming the successful reflection of changes. The search operation centers on finding a student using their email address, ensuring that the retrieval of student information based on this criterion is accurate. The data integrity test assesses the prevention of inserting records into `user information` with non-existent student or class IDs, upholding the integrity of foreign key constraints. Finally, the triggers test introduces a new subject into `relationships and membership table` and verifies the trigger's functionality in automatically adding the subject to all existing classes in `notification and notice table`. This ensures that triggers respond appropriately to maintain data coherence across related tables.

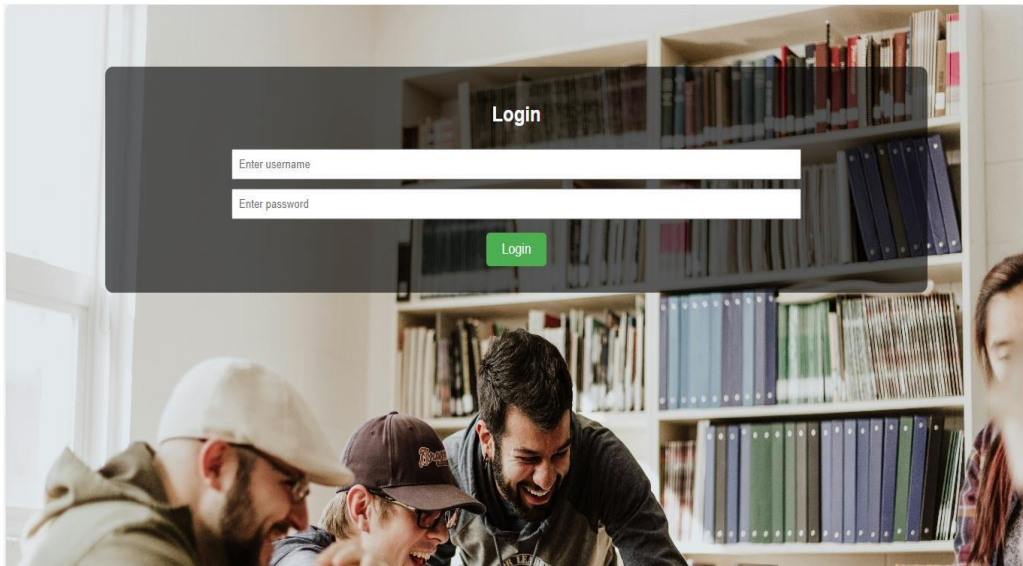
## SNAPSHOTS



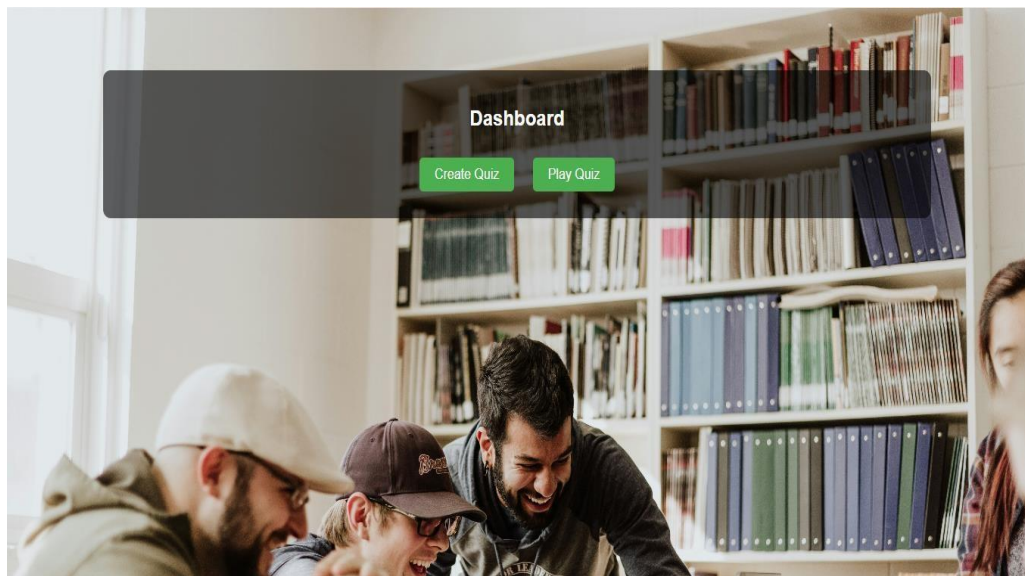
**Figure 6.1 Home Page**



**Figure 6.2 Sign up page**

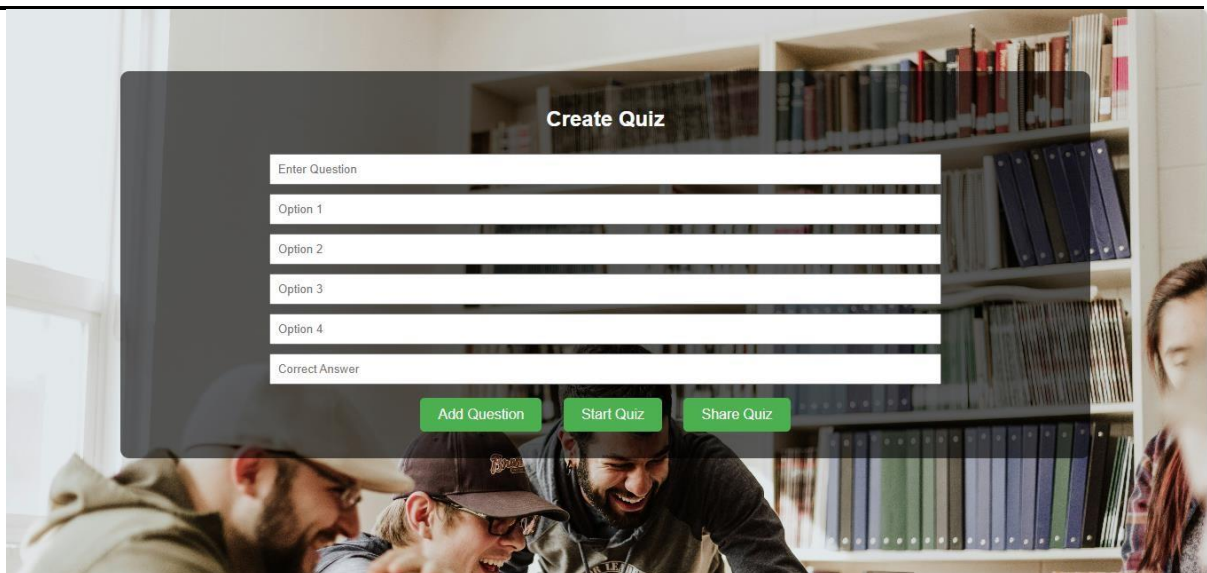


**Figure 6.3 Log in Page**

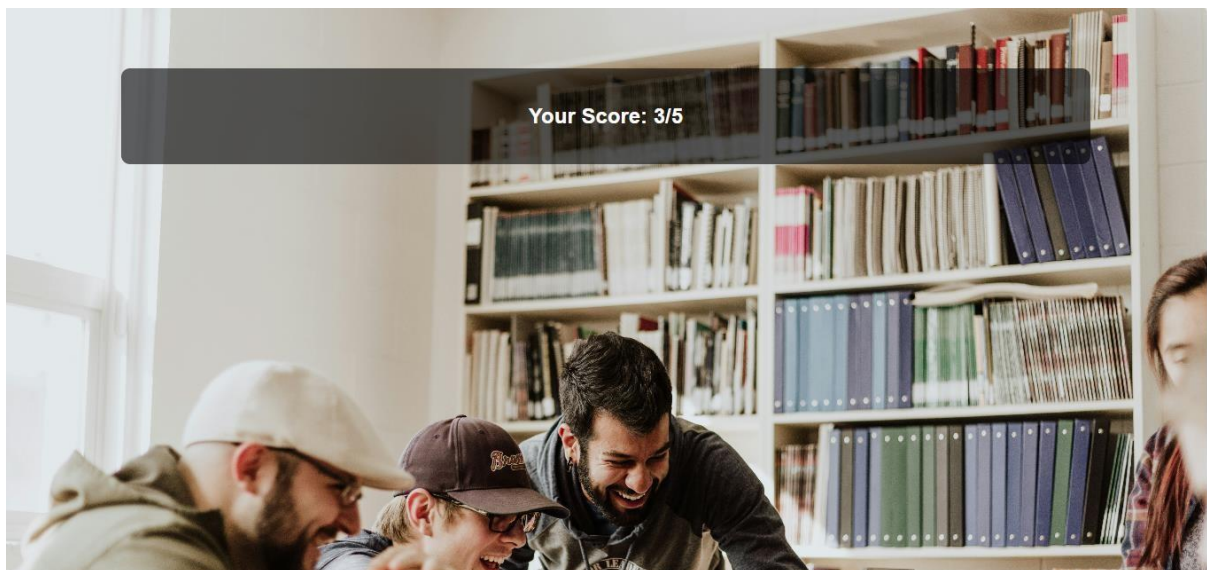


**Figure 6.4 DashBoard**





**Figure 6.5 Create Quiz**



**Figure 6.6 Result**

## CHAPTER 7 APPLICATIONS

### 1. Educational Institutions

Schools, colleges, and universities use quiz systems to conduct online assessments, practice tests, and mock exams. It helps teachers track student performance, reduce manual grading, and provide instant feedback.

---

### 2. Corporate Training

Organizations use quiz platforms for employee training and certification. Quizzes can test knowledge after training sessions, ensure regulatory compliance, and identify skill gaps among employees.

---

### 3. Recruitment and Skill Evaluation

Companies integrate quizzes into their hiring process to screen candidates. Timed aptitude, technical, and reasoning tests help shortlist qualified applicants efficiently and objectively.

---

### 4. E-learning Platforms

Online education platforms like Coursera, Udemy, and Khan Academy use quiz systems to evaluate learners, enhance engagement, and offer progress tracking through frequent assessments.

---

### 5. Competitive Exam Preparation

Quiz systems are essential tools for students preparing for competitive exams like UPSC, GRE, GATE, or SAT. They provide topic-wise questions, timed practice, and performance analytics.

## CHAPTER 8 CONCLUSION

The **Quiz Management System** is a powerful and efficient platform designed to streamline the process of conducting quizzes, assessments, and evaluations in both educational and professional environments. By automating the creation, administration, and evaluation of quizzes, the system significantly reduces the manual workload for educators and administrators, while also enhancing the user experience for students and participants. Its ability to handle multiple subjects, manage user data, and deliver instant feedback makes it a valuable tool in modern learning and testing environments.

Moreover, the system promotes **transparency, scalability, and accessibility**. Users can participate in quizzes from any location, at any time, which is especially beneficial in distance learning and remote work scenarios. The centralized database ensures that all records of quizzes, user results, and activity logs are maintained accurately and can be retrieved for future reference or analysis. This enhances not only performance tracking but also decision-making processes for educators and trainers.

In conclusion, the Quiz Management System is not just a testing tool but a comprehensive solution for knowledge evaluation and skill development. It bridges the gap between traditional examination systems and modern digital needs by offering a secure, user-friendly, and versatile platform. As technology continues to influence how we learn and grow, systems like this play a crucial role in making education and training more effective, inclusive, and data-driven.

## REFERENCES

- [MDN Web Docs](#)
- [W3Schools](#) –
- Node.js Official Documentation
- [MySQL Official Documentation](#)
- [Express.js Documentation](#)