

# Quiz Database Management System

PROJECT REPORT

By

**Ashmith S - RA2311026040005**

**Keshiv Raajh SK- RA2311026040042**

**Kenneth Roshan - RA2311026040068**

Under the guidance of

**Dr Dayana R**

*In partial fulfilment for the Course*

of

**21CSC205P – DATABASE MANAGEMENT SYSTEMS**

in the

Department of Computer Science & Engineering



**FACULTY OF ENGINEERING AND TECHNOLOGY**

**Department of Computer Science and Engineering**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**VADAPALANI CAMPUS**

**APRIL 2024**



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**Vadapalani Campus**

**Chennai**

**BONAFIDE CERTIFICATE**

Certified that this project report for the course  
**21CSC205P – DATABASE MANAGEMENT SYSTEMS** entitled in  
**“Quiz Database Management System”**  
is the Bonafide work\_ **Ashmith S - RA2311026040005** ,  
**KeshivRaajh SK- RA2311026040042** and **Kenneth Roshan -RA2311026040068**  
who carried out the work under my supervision.

**SIGNATURE**

**Dr Dayana R**

Assistant Professor

Department of CSE

SRM Institute of Science and Technology

Vadapalani campus

Chennai

**SIGNATURE**

**Dr P Chitra**

Professor & Head

Department of CSE(E.Tech)

SRM institute of Science and Technology

Vadapalani campus

Chennai

## ACKNOWLEDGEMENTS

We express our humble gratitude to our Honorable Chancellor **Dr. T. R. Paarivendhar**, Pro Chancellor (Administration), **Dr. Ravi Pachamoothu**, Pro Chancellor (Academics) **Dr. P. Sathyanarayanan**, Pro Chancellor (Admin) for the facilities extended for the completion of the project work.

We would record our sincere gratitude to our Vice Chancellor, **Dr. C. Muthamizhchelvan** and Registrar, **Dr. S. Ponnusamy**, for their support in completing our project work by giving us the best of academic excellence support system in place. We extend our sincere thanks to our Dean, **Dr. C V Jayakumar**, and Vice Principal – Academics, **Dr. C. Gomathy** and Vice Principal - Examination - **Dr. S. Karthikeyan** for their invaluable support.

We wish to thank **Dr P Chitra**, Professor & Head, Department of CSE E.Tech, SRM Institute of Science and Technology, Vadapalani Campus for her valuable suggestions and encouragement throughout the period of the project work and the course.

We extend our gratitude to our project co-ordinator and supervisor, **Dr Dayana R**, Assistant Professor, Department of CSE, SRM Institute of Science and Technology, Vadapalani Campus for providing us an opportunity to pursue our project under her mentorship. She provided us with the freedom and support to explore the research topics of our interest.

We sincerely thank all faculty of CSE E.Tech, staff and students of the department who have directly or indirectly helped our project.

# TABLE OF CONTENTS

ABSTRACT.....	1
1. INTRODUCTION.....	2
1.1 Project aim & Objectives	
1.1.1 Aim	
1.1.2 Objectives	
2. REQUIREMENTS.....	3
2.1 Software Requirements	
2.2 Hardware Requirements	
3. ER DIAGRAM.....	6
3.1 ER Diagram Explanation	
3.1.1 Entities	
3.1.2 Relationships	
4. TABLE STRUCTURE.....	9
4.1 Mcq's Table	
4.2 Student_Scores Table	
4.3 Subjects Table	
4.4 Users Table	
5. DML OPERATIONS.....	14
5.1 Insertion	
5.2 Deletion	
5.3 Updation	
5.4 Overall usage	

6. IDENTIFYING DEPENDENCIES.....	17
6.1 Entity-Relationship dependencies	
6.2 Attribute dependencies	
6.3 Functional dependencies	
6.4 Data Integrity Constraints	
7. NORMAIZATION OF DATABASE.....	19
7.1 Identifying Repetition of data	
7.2 Decomposition of Table	
7.3 Normalization Benefits	
8.FRONT END DEVELOPMENT.....	23
7.4 Introduction to Front End	
7.5 Designing the Front End	
9.RESULT.....	32
7.6 Execution of Application	
7.7 Project Screenshots	
10.CONCLUSION.....	34
11.FUTURE ENHANCEMENT.....	35
12. CERTIFICATE.....	34

## **ABSTRACT**

The **Quiz Database Management System** is a lightweight yet robust platform specifically engineered to simplify and enhance the process of creating, managing, and executing quizzes in a streamlined and efficient manner. Built using the Flask web framework for the backend and powered by SQLite3 for lightweight yet reliable database management, the system strikes a perfect balance between simplicity and functionality. It is designed to cater to a variety of use cases ranging from academic assessments and corporate training programs to personal self-evaluation tools.

At its core, the platform provides a structured interface for administrators to seamlessly manage quiz content, including adding, editing, categorizing, and organizing questions by topic, type (such as multiple choice, true/false, or short answer), and difficulty level. This categorization ensures that quizzes can be tailored to the specific needs of the target audience, whether it's for beginner learners or advanced users seeking more challenging material.

From a user's perspective, the system offers a clean, intuitive, and responsive interface that enhances the quiz-taking experience. Users can select quizzes based on their preferences or assigned modules, participate in real-time, and receive immediate feedback on their performance. Scoring is computed dynamically, with results displayed instantly upon quiz completion. This instant feedback mechanism helps users identify their strengths and areas needing improvement.

One of the standout features of the system is its **real-time leaderboard**, which promotes healthy competition and engagement, especially in classroom or group environments. It also supports user authentication and session tracking to ensure personalized experiences and secure access.

With its modular design and scalable structure, the Quiz Database Management System can easily be extended or integrated with larger educational platforms. Overall, it provides an accessible, flexible, and powerful solution for quiz administration, learning outcome tracking, and performance analytics in both educational and professional contexts.

# **1. INTRODUCTION**

## **1.1 Project Aim and Objectives**

### **1.1.1 Aim:**

The aim of this project is to develop a dynamic, web-based Quiz Database Management System using **Flask** as the web framework and **SQLite3** as the backend database. The system is intended to provide an interactive platform where users can take quizzes categorized by topic, question type, and difficulty level, while administrators can efficiently manage quiz content. The goal is to streamline the process of quiz creation, delivery, and performance evaluation in a secure, scalable, and user-friendly environment. This system is especially suited for educational institutions, training centers, and individuals looking to assess or enhance knowledge through self-paced quizzes. Additionally, it aims to provide real-time scoring, insightful feedback, and leaderboard generation to encourage engagement and track progress..

### **1.1.2 Objectives:**

#### **1. User-Friendly Interface**

To design and develop an intuitive and responsive web interface that allows users to easily navigate and interact with the quiz system.

#### **2. Efficient Database Design**

To implement a well-structured database using SQLite3 that effectively stores and manages quiz questions, user information, scores, and performance data.

#### **3. Categorization of Questions**

To organize quiz questions based on different types (e.g., multiple choice, true/false) to provide a varied and flexible quiz experience.

#### **4. Real-Time Scoring and Feedback**

To provide immediate scoring and feedback upon quiz completion, allowing users to instantly understand their performance.

#### **5. Administrative Functionality**

To enable administrators to add, edit, and delete questions, as well as manage users and view system analytics through a secure admin panel.

#### **6. Security and Authentication**

To ensure basic security features such as user authentication, input validation, and restricted admin access to protect data and system integrity.

## 2. REQUIREMENTS

This section outlines the essential software and hardware requirements necessary for the development, deployment, and execution of the Quiz Database Management System.

### 2.1 Software Requirements

To ensure smooth development and operation of the project, the following software components are required:

1. **Operating System:**  
Windows 11 or any compatible operating system (Windows 10/Linux/macOS).
2. **Backend Framework:**  
Python 3.11 with Flask Web Framework.
3. **Code Editor/IDE:**  
Visual Studio Code (VS Code)
4. **Database Management System (DBMS):**  
Sqlite3 for efficient data storage and management.
5. **Database Connector:**  
mysql-connector-python for establishing connectivity between Python (Flask) and MySQL.
6. **Frontend Technologies:**  
HTML5, CSS3, JavaScript with optional Bootstrap for responsive design and theme integration.
7. **Web Server:**  
Flask's built-in development server (for testing) or Apache/Gunicorn (for production deployment).
8. **Browser:**  
Google Chrome, Microsoft Edge, Mozilla Firefox (Latest versions preferred).



## **2.2 Hardware Requirements**

To develop and run the application effectively, the following minimum hardware specifications are recommended:

**1. Processor:**

Intel Core i3 or above (Intel Core i5 or higher recommended for optimal performance).

**2. RAM:**

Minimum 4 GB RAM (8 GB recommended for better multitasking and speed).

**3. Storage:**

At least 1 GB of free disk space for application files, dependencies, and local database storage.

**4. Display:**

Monitor with a minimum resolution of 1024 x 768 pixels.

**5. Input Devices:**

Standard keyboard and mouse or any compatible pointing device.

**6. Internet Connection:**

Required for package installation, updates, and external API integrations (optional).

### 3 .ER DIAGRAM

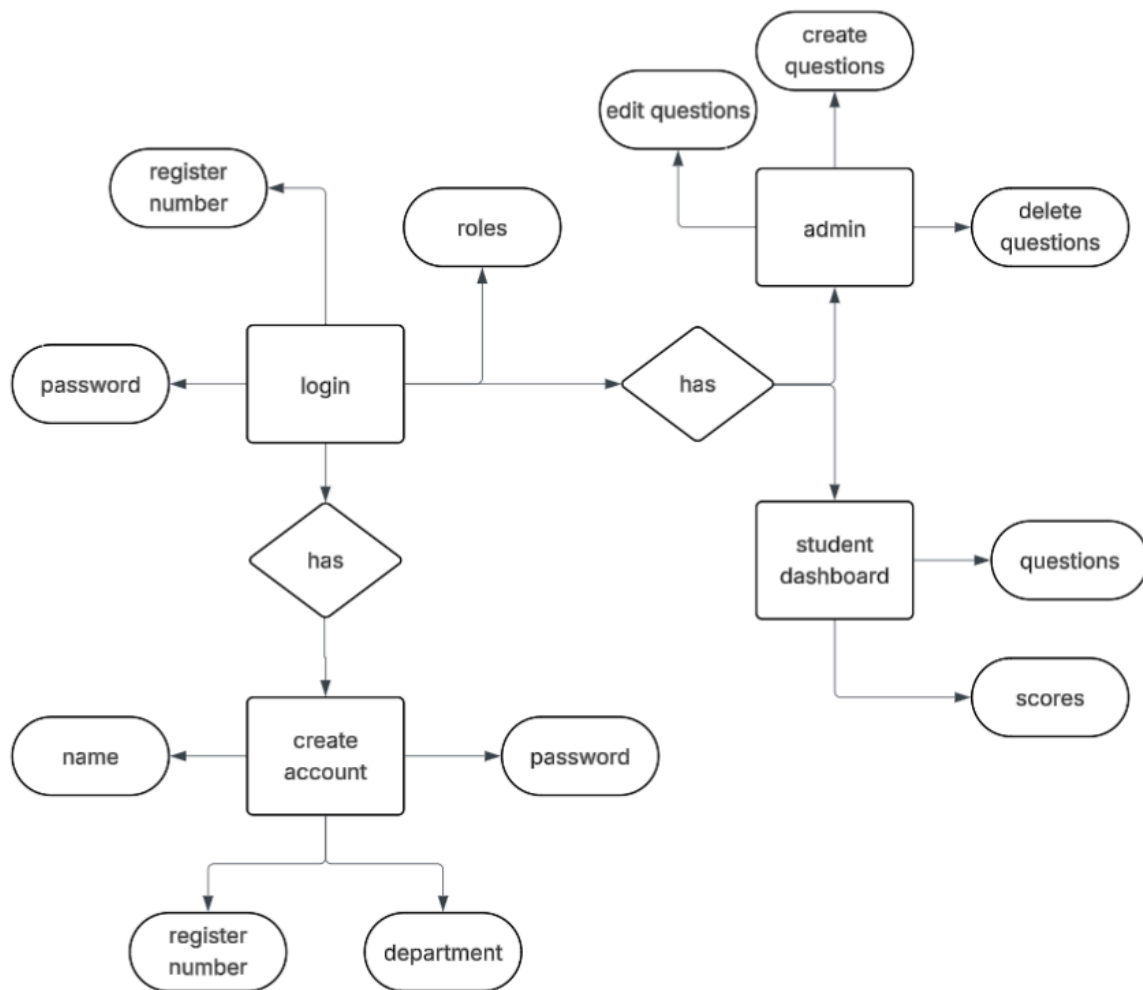


Fig. 3.1

#### 3.1 ER Diagram Explanation:

The Entity-Relationship (ER) diagram represents the core structure of an Online Quiz Management System. It outlines the entities involved, their attributes, and the relationships between them to ensure the system is both functional and logically organized.

### 3.1.1 Entities:

#### Create Account

This entity handles new user registrations for both students and admins.

- **Attributes:** Name, Register Number, Password, Department.
- **Register Number** acts as a unique identifier for users.

This ensures that only verified users are allowed access, maintaining a secure system.

#### Login

This entity authenticates users into the system based on their role.

- **Attributes:** Register Number, Password, Roles.  
Once logged in, the system checks the user's role (Admin or Student) to determine the interface and permissions.

#### Admin

This entity represents users with administrative privileges.

- **Functionalities:** Create Questions, Edit Questions, Delete Questions.  
Admins maintain the question bank by adding, modifying, or removing quiz questions.

#### Student Dashboard

This entity is accessible to users assigned the "student" role after login.

#### Functionalities:

- **Questions:** Access and attempt quizzes.
- **Scores:** View quiz results and track performance.

### 3.1.2 Relationships:

#### 1. **Login to Roles:**

A user logs in using their **Register Number** and **Password**. Once authenticated, the system identifies the user's **Role** to grant access to Admin or Student-specific dashboards.

**Relationship Type:** One-to-One.

#### 2. **Role to Admin/Student Dashboard:**

Depending on the assigned role (**Admin** or **Student**), the user is directed to the appropriate module.

- If **Admin**, the user can manage questions.
- If **Student**, the user can attempt quizzes and view their scores.

**Relationship Type:** One-to-One.

#### 3. **Admin to Questions Management:**

Admins can **Create**, **Edit**, and **Delete** questions, maintaining the question set for students.

**Relationship Type:** One-to-Many (since an admin can create, edit, or delete multiple questions).

#### 4. **Student Dashboard to Questions and Scores:**

Students access the quiz via the dashboard and submit answers. After attempting the quiz, their **Scores** are displayed for performance tracking.

**Relationship Type:** One-to-Many (a student can answer multiple questions and have multiple scores recorded).

## 4. TABLE STRUCTURE

The **Online Quiz Management System** includes several core tables designed to manage user accounts, quiz questions, subjects, and student scores. Each table has been structured to ensure data integrity, proper role management, and smooth handling of quiz-related operations.

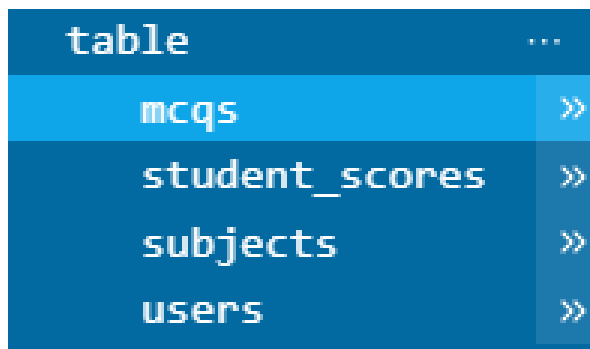


table	...
mcqs	>>
student_scores	>>
subjects	>>
users	>>

### 1. Users Table

The `users` table is responsible for storing the login credentials and profile details of all system users, which include both administrators and students.

Column Name	Data Type	Description
register_number	TEXT (Primary Key)	Unique identifier for each user (acts as a username).
role	TEXT (Not Null)	Describes the role of the user ( <code>admin</code> or <code>student</code> ).
name	TEXT (Not Null)	Stores the full name of the user.
class	TEXT	Represents the class or department for students; nullable for admins.
password	TEXT (Not Null)	Stores the encrypted password

The `users` table enables access control by differentiating between admin and student roles. Admins manage questions, while students take quizzes.

SELECT \* FROM users Schema Query Editor Auto Reload Find Other Tools...

register_number	TEXT NOT NULL PRIMARY KEY	role	TEXT NOT NULL PRIMARY KEY	name	TEXT NOT NULL	class	TEXT	password	TEXT NOT NULL
admin		admin		Administrator		NULL		\$2b\$12\$xcGwm3pIi1d...	
1		admin		Dayana		NULL		\$2b\$12\$FLuBsCV4s/Mg...	
1		student		John Pork		CSE		\$2b\$12\$Z3hcxj1WPKPV...	
5		student		Ashmith		CSE		\$2b\$12\$C.FbcSdjVe1X...	

## 2. Subjects Table

The `subjects` table contains the list of all available quiz subjects in the system.

Column Name	Data Type	Description
id	INTEGER (Primary Key, AutoIncrement)	Unique identifier for each subject.
name	TEXT (Not Null, Unique)	Name of the subject (e.g., Mathematics, Science).

This table ensures that all quiz questions are categorized under relevant academic subjects, allowing organized question management.

id	INTEGER PRIMARY KEY AUTOINCREMENT	name	TEXT NOT NULL UNIQUE	+
1		Mathematics		
2		Science		
3		English		
4		History		

### 3. MCQ Table

The MCQ table is designed to store quiz questions along with their respective answer choices and the correct answer.

Column Name	Data Type	Description
id	INTEGER (Primary Key, AutoIncrement)	Unique identifier for each question.
subject_id	INTEGER (Foreign Key)	References the <code>subjects</code> table to associate questions with subjects.
question	TEXT (Not Null)	The text of the quiz question.
option1-4	TEXT (Not Null)	Four answer choices provided for the question.
answer	TEXT (Not Null)	Specifies the correct answer (e.g., option1, option2).

This table enables admins to create, store, and maintain a bank of questions for each subject, ensuring the quiz system remains dynamic and scalable.

id INTEGER PRIMARY KEY AUTOINCREMENT	subject_id INTEGER NOT NULL REFERENCES subjects(id)	question TEXT NOT NULL	option1 TEXT NOT NULL	option2 TEXT NOT NULL	option3 TEXT NOT NULL	option4 TEXT NOT NULL	answer TEXT NOT NULL
4	1 name: 'Mathematics'	4+4	8	3	2	1	option1
5	1 name: 'Mathematics'	4+2	1	2	4	6	option4
6	2 name: 'Science'	a mammal lays eggs...	True	False	Partially true	no correct option	option2
8	3 name: 'English'	choose vowel	a	f	g	k	option1
9	3 name: 'English'	what is what	a	n	ds	d	option1

4. Scores Table

The `scores` table stores records of students' quiz attempts and their achieved scores.

Column Name	Data Type	Description
<code>id</code>	INTEGER (Primary Key, AutoIncrement)	Unique identifier for each score entry.
<code>register_number</code>	TEXT (Foreign Key)	References the student's <code>register_number</code> from the <code>users</code> table.
<code>subject_id</code>	INTEGER (Foreign Key)	Links to the <code>subjects</code> table to identify the quiz subject.
<code>score</code>	INTEGER (Not Null)	Stores the score obtained by the student.

This table plays a crucial role in monitoring student performance across subjects and allows both students and administrators to review quiz history and results.

<code>id</code> INTEGER PRIMARY KEY AUTOINCREMENT	<code>register_number</code> TEXT NOT NULL REFERENCES <code>users(register_number)</code>	<code>subject_id</code> INTEGER NOT NULL REFERENCES <code>subjects(id)</code>	<code>score</code> INTEGER NOT NULL	
1	the_student_registe...	1 name: 'Mathematic...	1	
2	the_student_registe...	1 name: 'Mathematic...	1	
3	the_student_registe...	1 name: 'Mathematic...	0	
4	the_student_registe...	1 name: 'Mathematic...	2	



## 5. Data Manipulation Language (DML) Operations

This section explores the Data Manipulation Language (DML) operations — including **Insertion**, **Deletion**, and **Updation** — and highlights their functionalities and application in the **Quiz Management System**.

### 5.1 Insertion

#### Functionality:

Insertion involves adding new records to the database tables.

#### Usage in Quiz Management System:

Insertion is a core part of multiple processes within the system:

- **(i) User Registration:**

When a student or admin creates a new account, their details — such as `register_number`, `role`, `name`, `class`, and `password` — are inserted into the `Users` table.

- **(ii) Question Creation:**

When administrators add new quiz questions, details like `subject_id`, `question`, four `options`, and the correct `answer` are inserted into the `Questions` table.

- **(iii) Subject Addition:**

When new academic subjects are introduced, their `name` is added to the `Subjects` table to categorize future questions.

- **(iv) Quiz Score Logging:**

After a student completes a quiz, their `register_number`, `subject_id`, and `score` are inserted into the `Scores` table to track performance.

## 5.2 Deletion

### Functionality:

Deletion involves the removal of existing records from the database.

### Usage in Quiz Management System:

- **(i) User Removal:**

If a student or admin account needs to be deactivated, the corresponding entry is deleted from the `Users` table.

- **(ii) Question Deletion:**

Administrators can remove outdated or incorrect questions from the `Questions` table to ensure the quality of quiz content.

- **(iii) Subject Deletion:**

Subjects that are no longer part of the curriculum can be deleted from the `Subjects` table, after ensuring that related questions are handled.

- **(iv) Score Record Deletion:**

Old or incorrect score records can be deleted from the `Scores` table for data cleanup or correction purposes.

## 5.3 Updation

### Functionality:

Updation modifies existing records in the database.

### Usage in Quiz Management System:

- **(i) User Account Update:**

Users can update their `password` or `class` information in the `Users` table using update operations.

- **(ii) Question Editing:**

If a question contains an error or needs improvement, the corresponding record in the `Questions` table can be updated.

- **(iii) Subject Name Update:**

If the name of a subject changes or needs correction, it can be updated in the `Subjects` table.

- **(iv) Score Modification:**

In case of errors in the evaluation or data entry, the `score` field in the `Scores` table can be updated to reflect the correct result.

## 5.4 Overall Usage in Quiz Management System

DML operations are fundamental to the smooth operation of the Quiz Management System. These operations allow users and administrators to add, remove, or modify records — ensuring that the system remains accurate, organized, and flexible for continuous use. Whether registering users, adding questions, recording scores, or managing subjects, DML operations enable the system to maintain dynamic and real-time data consistency.

## 6. Identifying Data Dependencies

In the Quiz Management System, data dependencies define the connections and logical relationships between various elements of the database. Recognizing these dependencies ensures a streamlined and optimized database structure that supports accurate data storage and retrieval.

### 6.1 Entity-Relationship Dependencies

- **Users and Scores:**

A one-to-many relationship exists between the `Users` table and the `Scores` table. Each user can have multiple entries in the `Scores` table, representing different quiz attempts across subjects.

- **Subjects and Questions:**

A one-to-many relationship is observed between the `Subjects` table and the `Questions` table. Each subject can have multiple questions associated with it, allowing subject-wise quiz generation.

- **Subjects and Scores:**

A one-to-many relationship exists between the `Subjects` table and the `Scores` table. Multiple score records may be linked to a single subject, representing the performances of different students.

## 6.2 Attribute Dependencies

- **User Register Number in Scores Table:**

The `register_number` in the `Scores` table is a foreign key that references the `register_number` in the `Users` table. This dependency ensures each score record is always associated with a valid registered user.

- **Subject ID in Questions and Scores Tables:**

The `subject_id` in both the `Questions` and `Scores` tables depends on entries in the `Subjects` table, ensuring that questions and scores are always linked to existing subjects.

## 6.3 Functional Dependencies

- **Score Calculation:**

The `score` field in the `Scores` table is functionally dependent on the number of correct answers submitted by the student during a quiz session.

- **Correct Answer Dependency:**

In the `Questions` table, the `answer` attribute is dependent on the set of multiple-choice options (`option1`, `option2`, `option3`, `option4`). Only one of these options can be marked as correct.

## 6.4 Data Integrity Constraints

- **Primary Keys:**

Every table in the Quiz Management System uses primary keys to ensure the uniqueness of records — for example, `register_number` in the `Users` table, `id` in the `Subjects` and `Questions` tables, and `id` in the `Scores` table.

- **Foreign Keys:**

Foreign key constraints maintain referential integrity:

- `register_number` in the `Scores` table references the `Users` table.
- `subject_id` in both `Questions` and `Scores` references the `Subjects` table.

These constraints ensure that all questions, scores, and users are valid and interconnected, reducing errors and maintaining data consistency.

## 7. Normalization of Database

Normalization is an essential database design technique aimed at reducing redundancy and enhancing data consistency by organizing the database into well-structured, logically related tables. It helps optimize storage, simplifies data maintenance, and ensures reliable data retrieval. In the context of the **Quiz Management System**, normalization has been applied to prevent data duplication and to promote a clean, scalable, and efficient database design.

### 7.1 Identifying Repetition of Data

In the initial design of the system, some data — particularly user and subject-related information — was found to be repeated across multiple records. For example:

- The same subject name appeared repeatedly in the **Questions** table for each associated question.
- User information such as **name**, **role**, and **class** could be repeated across quiz records in an unnormalized setup.
- Storing scores and questions directly in a single table would result in duplicated user and subject details, reducing efficiency.

### 7.2 Decomposition of Table

To address these issues, the database schema was decomposed into smaller, purpose-specific tables through normalization. The process was guided by the principles of First Normal Form (1NF), Second Normal Form (2NF), and Third Normal Form (3NF).

#### 7.2.1 Users Table

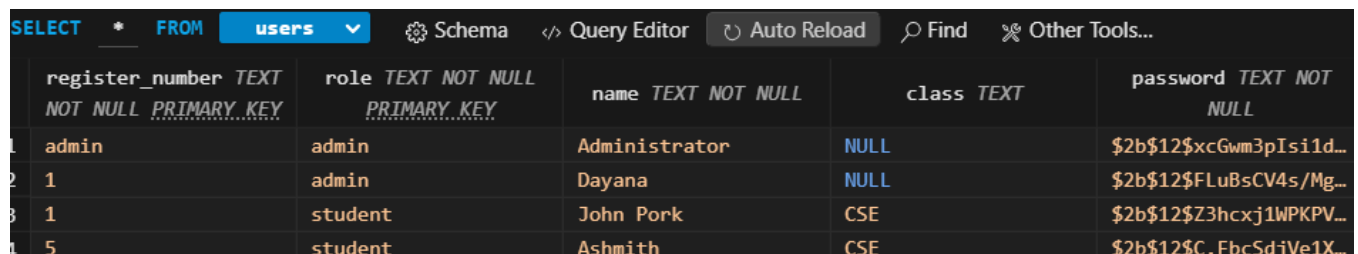
**Attributes:**

- **register\_number**: VARCHAR, NOT NULL (Primary Key)

- role: VARCHAR, NOT NULL
- name: VARCHAR, NOT NULL
- class: VARCHAR
- password: VARCHAR, NOT NULL

#### Purpose:

Stores unique user credentials and roles (admin or student), ensuring that personal details are maintained in one place and referenced wherever needed.



The screenshot shows a database query editor with the 'users' table selected. The table has five columns: register\_number, role, name, class, and password. The data is as follows:

register_number	role	name	class	password
admin	admin	Administrator	NULL	\$2b\$12\$xcGwm3pIsi1d...
1	admin	Dayana	NULL	\$2b\$12\$FLuB5CV4s/Mg...
1	student	John Pork	CSE	\$2b\$12\$Z3hcxj1WPKPV...
5	student	Ashmith	CSE	\$2b\$12\$C.FbcSdjVe1X...

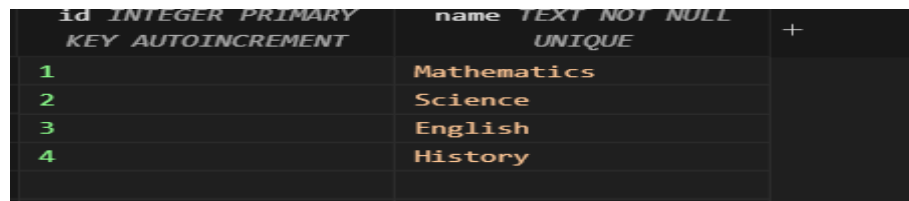
## 7.2.2 Subjects Table

#### Attributes:

- id: INT, AUTO\_INCREMENT (Primary Key)
- name: VARCHAR, NOT NULL

#### Purpose:

Defines the academic subjects for which quizzes are created, eliminating the need to repeatedly store subject names in the MCQ and Scores tables.



The screenshot shows a database query editor with the 'Subjects' table selected. The table has three columns: id, name, and an empty column. The data is as follows:

id	name	
1	Mathematics	
2	Science	
3	English	
4	History	

## 7.2.3 MCQ Table

### Attributes:

- `id`: INT, AUTO\_INCREMENT (Primary Key)
- `subject_id`: INT, NOT NULL (Foreign Key)
- `question`: TEXT, NOT NULL
- `option1`: VARCHAR, NOT NULL
- `option2`: VARCHAR, NOT NULL
- `option3`: VARCHAR, NOT NULL
- `option4`: VARCHAR, NOT NULL
- `answer`: VARCHAR, NOT NULL

### Purpose:

Stores quiz questions along with multiple-choice answers, linking each question to its subject via the `subject_id`. This avoids repeating subject names for each question.

<code>id</code> INTEGER PRIMARY KEY AUTOINCREMENT	<code>subject_id</code> INTEGER NOT NULL REFERENCES <code>subjects(id)</code>	<code>question</code> TEXT NOT NULL	<code>option1</code> TEXT NOT NULL	<code>option2</code> TEXT NOT NULL	<code>option3</code> TEXT NOT NULL	<code>option4</code> TEXT NOT NULL	<code>answer</code> TEXT NOT NULL
4	1 <code>name: 'Mathematic...</code>	4+4	8	3	2	1	option1
5	1 <code>name: 'Mathematic...</code>	4+2	1	2	4	6	option4
6	2 <code>name: 'Science'</code>	a mammal lays eggs...	True	False	Partially true	no correct option	option2
8	3 <code>name: 'English'</code>	choose vowel	a	f	g	k	option1
9	3 <code>name: 'English'</code>	what is what	a	n	ds	d	option1



## 7.2.4 Scores Table

### Attributes:

- `id`: INT, AUTO\_INCREMENT (Primary Key)
- `register_number`: VARCHAR, NOT NULL (Foreign Key)
- `subject_id`: INT, NOT NULL (Foreign Key)
- `score`: INT, NOT NULL

### Purpose:

Records the quiz scores for each user and subject combination without duplicating user or subject details. The use of foreign keys ensures that only valid users and subjects are referenced.

<code>id</code> INTEGER PRIMARY KEY AUTOINCREMENT	<code>register_number</code> TEXT NOT NULL REFERENCES <code>users(register_number )</code>	<code>subject_id</code> INTEGER NOT NULL REFERENCES <code>subjects(id)</code>	<code>score</code> INTEGER NOT NULL	
1	the_student_registe...	1 name: 'Mathematic...	1	
2	the_student_registe...	1 name: 'Mathematic...	1	
3	the_student_registe...	1 name: 'Mathematic...	0	
4	the_student_registe...	1 name: 'Mathematic...	2	

## 7.3 Normalization Benefits

- **Reduced Redundancy:**

By splitting the database into focused tables — `Users`, `Subjects`, `Questions`, and `Scores` — the same data is not unnecessarily repeated, which saves space and reduces errors.

- **Improved Data Integrity:**

If any piece of data, like a user's name or a subject title, needs updating, it is only changed in one place, avoiding inconsistency across the system.

- **Scalability:**

New users, subjects, questions, and quiz attempts can be added without modifying existing table structures, ensuring smooth expansion of the system.

- **Efficient Querying:**

Normalized tables improve query speed and accuracy, especially when performing joins to retrieve quiz reports, analyze performance, or generate question banks.

## 8.1 Introduction to Front-End Development

Front-end development refers to the creation of the visual and interactive components of a web application that users interact with directly. It includes the layout design, forms, tables, buttons, and user interface elements that make the system accessible and user-friendly. In the **Quiz Management System**, the front end is developed using **HTML, CSS, and Flask**. A small **JavaScript** module is used to support the application's visual behavior.

### 8.1.1 HTML & CSS (with Flask Templates)

- **HTML (HyperText Markup Language)** is used to structure the content and elements on each web page.
- **CSS (Cascading Style Sheets)** is used to control the visual presentation and layout, making the pages visually appealing.

### 8.1.2 Reasons for Choosing Flask + HTML/CSS

- **Lightweight and Simple:** Flask is a micro-framework that is easy to set up and ideal for small-to-medium-sized projects like this.
- **Tight Backend Integration:** Flask allows seamless communication between the front end and the backend database logic using Python.
- **Customizable Styling:** Using CSS provides full control over the look and feel of the application.
- **Template Reuse:** Using base templates and inheritance in Jinja2 avoids code duplication and simplifies layout consistency.
- **Responsive Layout:** CSS ensures that the application works well across desktops, tablets, mobile devices.

## 8.2 Designing the Front End

### Login Page

The **Login Page** serves as the primary access point for users of the Quiz Management System. It provides a secure and intuitive interface where users can select their role (such as Student or Administrator), enter their register number, and input their password to authenticate themselves. The login form is designed using HTML and styled with CSS to maintain a clean and responsive layout. Flask handles the backend logic, validating credentials against stored user data and managing user sessions. This ensures that only authorized users can access the system's functionalities based on their assigned roles, thereby maintaining data security and role-based access control.

The image displays two screenshots of the Quiz App front end. The top screenshot shows the Register page, which includes a header with the Quiz App logo and a Login link. The main content area features a Register form with fields for Username, Email address, Password, and Confirm Password. Below these fields is a dropdown menu for 'Register as:' with a 'Choose...' option, followed by a Register button and a link to 'Login here' for existing users. The bottom screenshot shows the Login page, which has a similar header with a Register link. The main content area features a Login form with fields for Username (pre-filled with 'JOHN PORK') and Password (masked with dots), followed by a Login button and a link to 'Register here' for new users. Both pages have a light blue background and a white form container. A copyright notice '© 2025 Quiz App. All rights reserved.' is visible at the bottom of the Login page.

Quiz App Login

Register

Username

Email address

Password

Confirm Password

Register as:

Choose...

Register

Already have an account? [Login here](#)

Quiz App Login Register

Login

Username

JOHN PORK

Password

.....

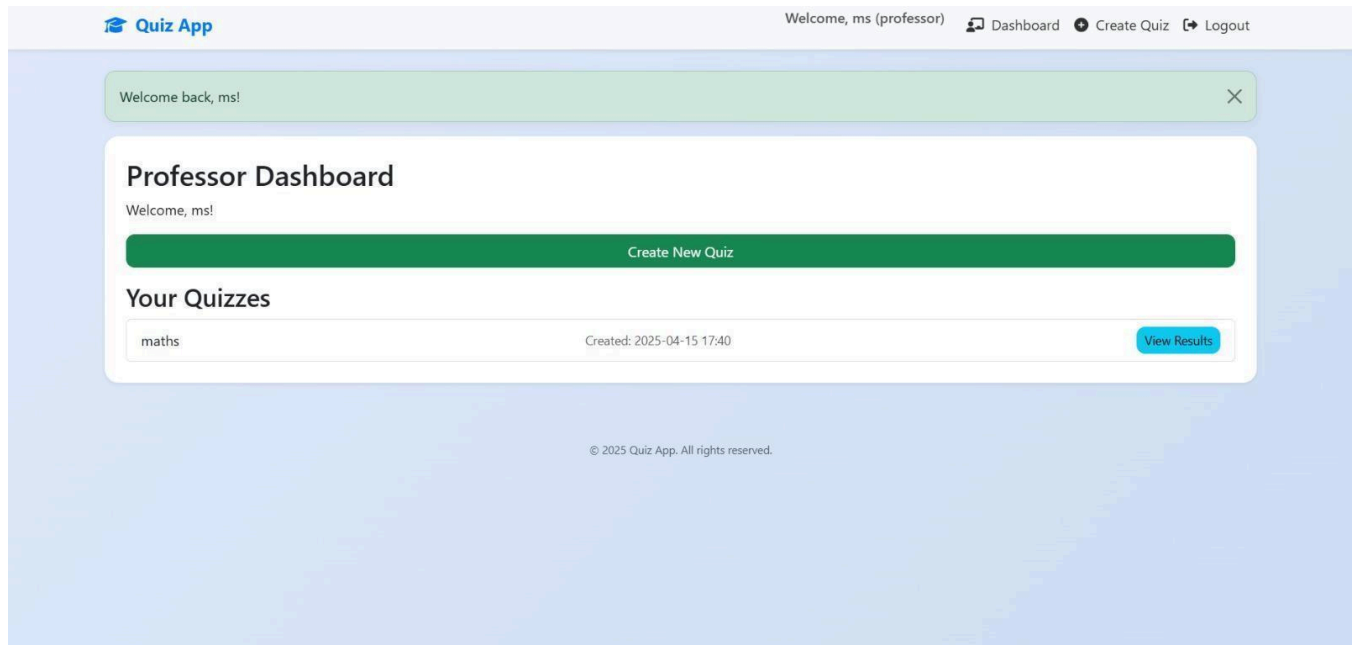
Login

Don't have an account? [Register here](#)

© 2025 Quiz App. All rights reserved.

## ADMIN PAGE:

The Admin Dashboard in the Quiz Management System allows administrators to manage subjects and multiple-choice questions (MCQs). Admins can add new subjects through a simple input field and button, and view or manage existing ones like Mathematics, Science, English, and History. This interface helps streamline quiz organization, ensures efficient content management, and gives full control over the academic content structure within the system.



## ADD, DELETE AND EDIT PAGE:

The Admin page provides essential tools for managing quiz subjects and their questions. Using the Add function, admins can easily create new subjects by entering a name and clicking the "Add Subject" button, instantly expanding the quiz categories. The Edit feature allows admins to update existing subject names or modify MCQs to keep content accurate and up to date. Meanwhile, the Delete option helps remove outdated or unnecessary subjects and questions, keeping the system organized and clutter-free. Together, these functions give admins full control over the quiz structure, ensuring the system remains efficient and relevant.

Quiz App

DashboardCreate QuizLogout

Create New Quiz

Quiz Title

Enter quiz title

Questions

Question 1

Question Text

Option A

Option B

Option C

Option D

Correct Option

Choose...

+ Add Another Question

✓ Create Quiz

## STUDENT DASHBOARD

The Student Dashboard allows users to participate in quizzes by selecting a subject and answering multiple-choice questions. After submitting their answers, students instantly receive their scores, helping them track their performance. The dashboard also displays a leaderboard that ranks students based on their quiz scores, adding a competitive and motivational element. This setup encourages learning, provides immediate feedback, and helps students monitor their progress in real-time.

Welcome back, JOHN PORK!



## Student Dashboard

Welcome, JOHN PORK!

### Available Quizzes

maths

By: ms.

Already Taken

English

By: 1

Already Taken

### Your Past Results

maths

Score: 0/1 (Submitted: 2025-04-15 17:41)

View Details

English

Score: 1/1 (Submitted: 2025-04-15 17:27)

View Details

© 2025 Quiz App. All rights reserved.

## Result for: English

**Your Score: 1 out of 1**

Submitted on: 2025-04-15 17:27:46 UTC

Excellent work!

Back to Dashboard

© 2025 Quiz App. All rights reserved.

## **9. RESULT**

The **Quiz Management System** was successfully developed using **Flask** as the backend framework and **MySQL** as the relational database system. The front end was built using **HTML**, **CSS**, and **Jinja2 templates**, offering a simple and responsive interface.

The application was tested across key functionalities such as:

- User registration and login
- Role-based access for Admins and Students
- Quiz creation (Admin)
- Quiz participation (Student)
- Automatic result calculation and viewing

All features worked smoothly under testing, providing a user-friendly and functional experience.

### **9.1 Execution of Application**

#### **Launching the Application:**

The Flask server is started using the command:

```
python app.py
```

Once started, the application can be accessed in a web browser at <http://127.0.0.1:5000/>.

#### **Registration and Login:**

Users can register by selecting a role (Admin or Student), then entering a name, register number, and password.

Once registered, users can log in to access their respective dashboards.



- **Admins** can create quizzes and view student results.
- **Students** can take quizzes and check their scores.

### **Quiz Creation (Admin):**

Admins can add quizzes by simply providing:

- A quiz title
- A set of multiple-choice questions
- Four options per question and the correct answer

No additional confirmation, description, or timer setup is required, keeping the process fast and straightforward.

### **Quiz Participation (Student):**

Students can:

- View available quizzes
- Answer each question by selecting one of the given options
- Submit the quiz directly when done

The system instantly evaluates the responses and stores the final score.

### **View Results:**

- **Students** see their scores immediately after submitting the quiz.
- **Admins** can access a list of all quiz submissions along with the student names and their

## **10.CONCLUSION**

The Quiz Management System project delivers an efficient, scalable, and user-friendly platform for managing online quizzes. From subject creation to question management and real-time quiz participation, the system simplifies complex processes through an intuitive web interface built with modern technologies such as Flask, SQLite 3, and responsive HTML/CSS design.

With features like secure authentication, role-based access control, and dynamic quiz rendering, the system ensures a smooth experience for both administrators and students. Real-time scoring, instant feedback, and leaderboard integration encourage active participation and foster healthy competition among learners. Administrators can seamlessly add, edit, or delete subjects and questions, maintaining an up-to-date and engaging quiz environment.

The use of SQLite 3 provides lightweight, reliable data storage and efficient retrieval, making the application easy to set up and deploy without complex server requirements. Flask's modular structure further enhances maintainability and scalability. Overall, this project not only demonstrates core principles of database management and full-stack development but also provides a practical and engaging solution for educational assessment. It establishes a solid foundation for future enhancements, such as advanced analytics, timed quizzes, and certification features

## 11. FUTURE ENHANCEMENTS

To improve the efficiency and user experience of the **Quiz Management System**, the following future enhancements are proposed:

- **Notification System:** Integrate email or in-app notifications to inform students about upcoming quizzes, results, or system updates.
- **Question Bank Management:** Develop a centralized question bank to allow Admins to reuse, categorize, and randomly generate quiz questions.
- **Timed Quizzes:** Add a timer feature to enforce time limits on quizzes for better assessment control.
- **Mobile App Integration:** Create a mobile application version of the system, enabling students to take quizzes and check scores on the go.
- **Result Analytics:** Provide detailed performance analytics including charts, topic-wise scores, and comparison with previous attempts.
- **Export Functionality:** Enable downloading quiz results in PDF or CSV format for easy reporting and academic tracking.

## 12.CERTIFICATE



# COURSE COMPLETION CERTIFICATE

The certificate is awarded to

**KESHIV RAAJH (RA2311026040042)**

for successfully completing the course

## Database and SQL

on April 12, 2025



*Congratulations! You make us proud!*

Issued on: Saturday, April 12, 2025  
To verify, scan the QR code at <https://verify.onwingspan.com>



Thirumala Arohi  
Executive Vice President and Global Head  
Education, Training & Assessment (ETA)  
Infosys Limited



## COURSE COMPLETION CERTIFICATE

The certificate is awarded to

**ASHMITH SADHEESH KUMAR (RA2311026040005)**

for successfully completing the course

**Database and SQL**

on April 13, 2025



*Congratulations! You make us proud!*



Issued on: Sunday, April 13, 2025  
To verify, scan the QR code at <https://verify.onwingspan.com>

Thirumala Arohi  
Executive Vice President and Global Head  
Education, Training & Assessment (ETA)  
Infosys Limited



## COURSE COMPLETION CERTIFICATE

The certificate is awarded to

**KENNETH ROSHAN A (RA2311026040068)**

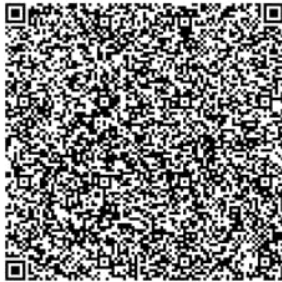
for successfully completing the course

**Database and SQL**

on April 20, 2025



*Congratulations! You make us proud!*



Issued on: Sunday, April 20, 2025  
To verify, scan the QR code at <https://verify.onwingspan.com>

Thirumala Arohi  
Executive Vice President and Global Head  
Education, Training & Assessment (ETA)  
Infosys Limited