**Quiz Master - V1**

**A project report for the MAD-I**

Submitted by

Name: Sachin

Roll number: 21f2000817



IITM Online BS Degree Program,

Indian Institute of Technology, Madras, Chennai

Tamil Nadu, India, 600036

Contents

[1 Student Details 1](#_Toc3127)

[2 Project Description 1](#_Toc3128)

[3 Problem Solving Approach 2](#_Toc3129)

[4 Framework And Technologies Used 2](#_Toc3130)

[6 Demo video 3](#_Toc3131)

# 1 Student Details

My name is Sachin, and I am currently pursuing the Data Science and Applications program at IIT Madras with roll number 21f2000817. I am also a first year Master of Computer Application(MCA) student at Kurukshetra university. My academic interests lie in data science, and I have moderate proficiency in Python, javascript, c++.

# 2 Project Description

**Quiz Master - V1** is a comprehensive multi-user platform designed to facilitate exam preparation through interactive quizzes. The application supports distinct user roles: **Admin** and **Users (Students)**, each with specific functionalities to ensure a seamless learning experience. Built with a robust backend framework and an intuitive frontend, the platform provides essential features such as user authentication, quiz management, score tracking, and search functionality.

Key functionalities include an **Admin Dashboard**, allowing administrators to oversee all activities, including managing quizzes, users, and reviewing performance analytics. Admins can create, update, and delete quizzes, assign them to relevant subjects or chapters, and monitor user engagement. **Users** can browse available quizzes, attempt them to assess their knowledge, and track their performance through a detailed summary, including average scores, highest scores, and past attempts.

Additionally, the platform features an advanced **Search System**, enabling users to find quizzes based on subject or chapter keywords. A structured **Score Tracking System** ensures users can view their progress over time, helping them focus on areas needing improvement. The application serves as a **one-stop solution** for students looking to enhance their learning experience, providing an engaging, efficient, and user-friendly environment for quiz-based assessments.

# 3 Problem Solving Approach

To develop the **Quiz Master - V1** application, the project was divided into a series of structured steps. Initially, the core application logic was set up in **app.py**, including necessary imports and configurations. The database schema was implemented in **models.py** using Object-Relational Mapping (ORM) to define tables such as **Users, Quizzes, Scores, Subjects, and Chapters**, ensuring seamless interaction with **SQLite**.

Next, **user authentication** was developed, including login and registration functionality, with routes defined in **routes/user.py** and templates styled using **Bootstrap**. The **dashboard for users** was implemented to display available quizzes and track scores. The **Admin Dashboard** was developed to manage users, quizzes, and performance analytics efficiently.

A **search functionality** was introduced, allowing users to find quizzes based on subjects or chapters, enhancing accessibility. The **score tracking system** was refined to display key insights such as **total quizzes taken, average scores, and highest scores**. Comprehensive **error handling and debugging** were performed to ensure smooth functionality. Finally, the development process concluded with **rigorous testing** and a demonstration showcasing the platform’s features, ensuring a user-friendly and efficient exam preparation experience.

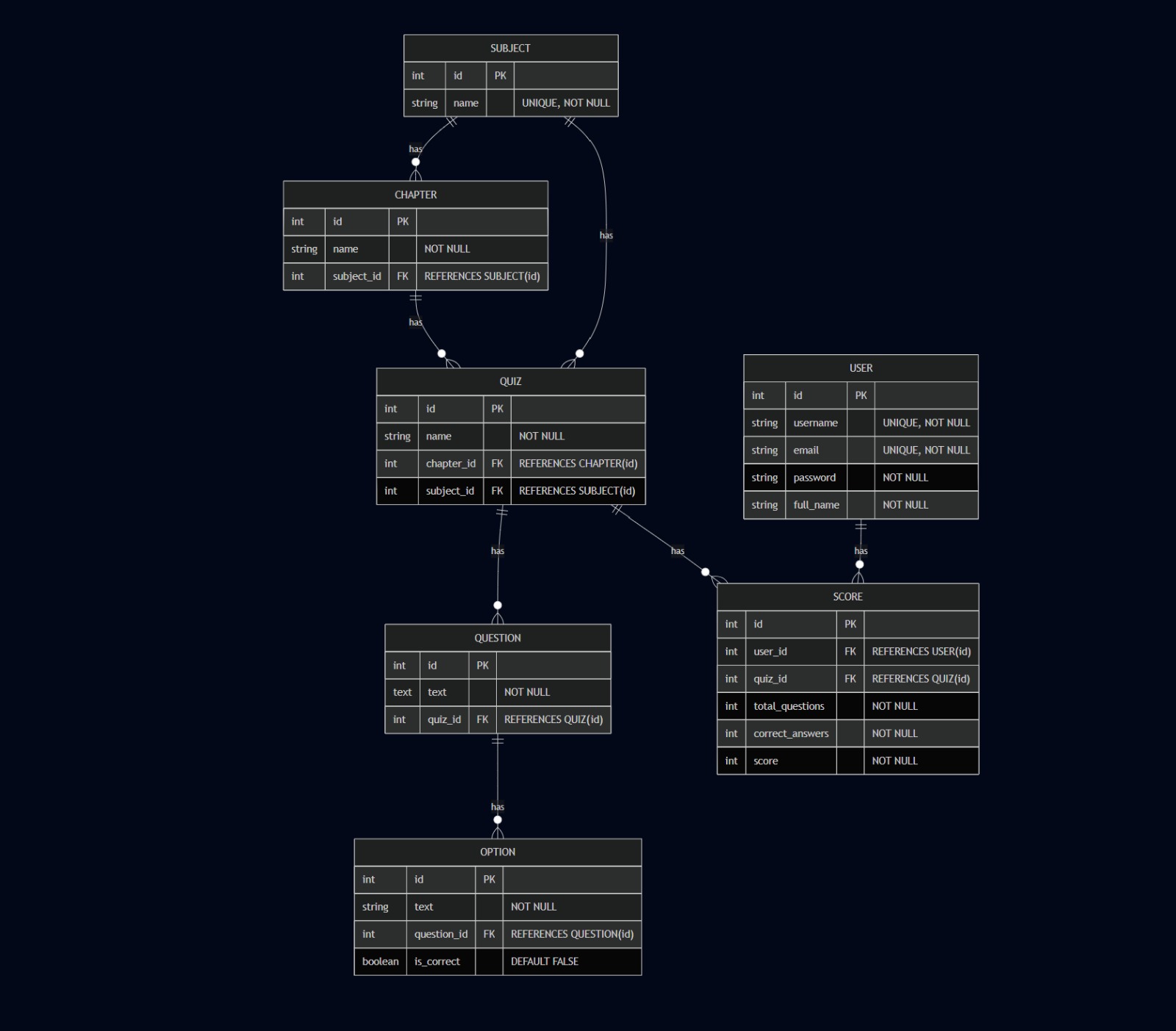
# 4 Framework And Technologies Used

1. **Flask**: Backend framework for routing and business logic.
2. **Jinja2 Templates**: Dynamic HTML rendering.
3. **Bootstrap**: Responsive UI design.
4. **SQLite**: Lightweight database for data storage.

## 5 Database Schema Design and Terminologies

This ER diagram represents the database structure for a household services application, showcasing the relationships between four main entities: **SERVICE\_PROFESSIONAL**, **CUSTOMER**, **SERVICE**, and **SERVICE\_REQUEST**. The **SERVICE\_PROFESSIONAL** table contains details about service providers, including their unique id, contact information, expertise, experience, and associated service type through the service\_id foreign key. The **CUSTOMER** table holds information about users requesting services, including their unique id, contact details, and account status. Both entities contribute to handling and fulfilling service requests.

The **SERVICE** table lists the types of services available, with attributes such as name, description, and base\_price. The **SERVICE\_REQUEST** table acts as a bridge between **CUSTOMER**, **SERVICE**, and **SERVICE\_PROFESSIONAL**, linking them through foreign keys (customer\_id, service\_id, and professional\_id) to record detailed transactions. It tracks information such as the request date, status, customer feedback, and ratings. Relationships are clearly defined: service professionals *provide* services, customers *make* service requests, and professionals *handle* these requests. This structure supports efficient data management for the application, ensuring smooth operations between users and service providers.



# 6 Demo video

Here is my google drive [link](https://drive.google.com/file/d/1KbERy26LajfZQejRUzbDGw_ItivX7dcg/view) for my video presentation for mad1 project