

# Project Report: Quiz Master

---

## Student Details

**Name:** Mohd Afsar

**Roll Number:** 23f2002023

**Email:** 23f2002023@ds.study.iitm.ac.in

**Program:** B.S. Degree in Data Science and Applications, IIT Madras

**Current Term:** Diploma, Term 3

---

## Project Details

**Project Title:** Quiz Master

**Course:** Modern Application Development - 2 (MAD 2)

**Project Description:** Quiz Master is an interactive web-based platform designed to facilitate the creation, management, and evaluation of quizzes. The platform allows users to register, attempt quizzes, and track their performance. It supports subject- and chapter-based quizzes, ensuring a structured learning experience.

### Approach to Problem Statement:

- Understanding Requirements:** Initially, I analyzed the project requirements to define the core functionalities and user flows.
- Database Design:** Designed an efficient relational schema to manage users, quizzes, questions, and scores.
- Backend Development:** Implemented a RESTful API using Flask to handle authentication, quiz management, and result tracking.
- Frontend & UI Integration:** Created a user-friendly interface with dynamic quiz functionality.
- Testing & Debugging:** Performed rigorous testing to ensure smooth user experience and functionality.
- Documentation & Submission:** Referred to official Flask and SQLAlchemy documentation to validate the implementation.

---

## Technologies & Frameworks Used

- **Backend:** Flask, Flask-SQLAlchemy, Flask-Security
- **Database:** SQLite
- **Frontend:** Vue-CLI, CSS
- **Other Libraries:** datetime, bcrypt (for password hashing)

---

## ER Diagram

The database consists of multiple tables with the following relationships:

1. **User** (has a many-to-many relationship with **Role** and **Subject**)
2. **Qualification** (one-to-many with **User** and **Subject**)
3. **Subject** (one-to-many with **Chapter** and **Question**)
4. **Chapter** (one-to-many with **Quiz** and **Question**)
5. **Quiz** (one-to-many with **Score** and **Question**)
6. **UserQuizzes** (tracks user quiz attempts)
7. **Score** (stores user performance data)
8. **RecentActivity** (logs user interactions)

*(ER diagram not available)*

---

## API Resource Endpoints

*(API documentation YAML file attached separately)*

**YAML FILE NAME:** api-doc.yaml

---

## Architecture & Features

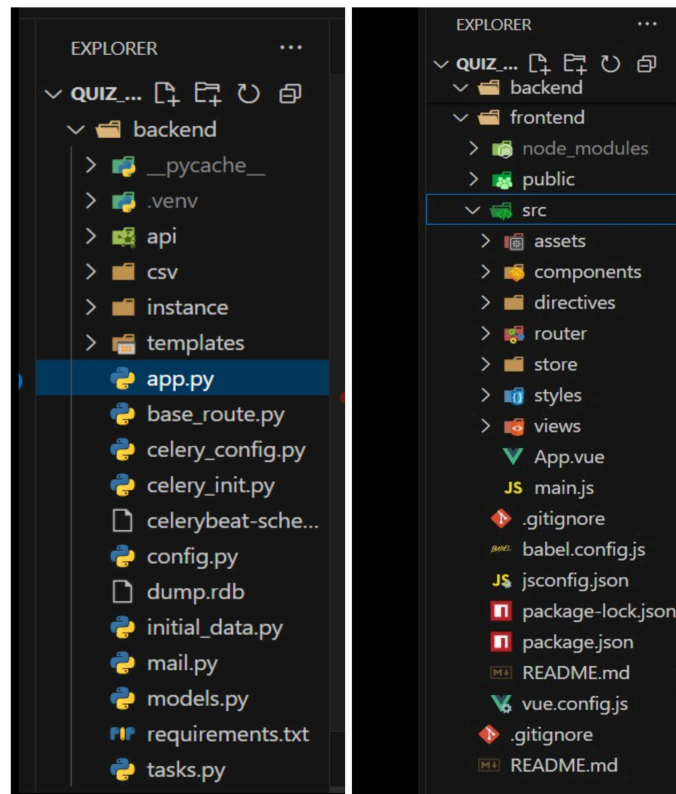
### System Architecture:

- **MVC Structure** (Model-View-Controller) followed for structured development.
- **Token-Based Authentication** implemented for secure access.
- **Role-Based Access Control (RBAC)** for admin and user functionalities.

### Key Features:

- **User Authentication** with role management.
- **Quiz Creation & Management** (by admin/authorized users).
- **Subject & Chapter-based quiz categorization.**
- **Quiz Attempt Tracking** and performance evaluation.
- **Interactive UI** for a seamless user experience.

### My Project directory:



---

## **Presentation Video Link**

**Video Link:** [Click Here](#)

---

**Submitted by:**

**Mohd Afsar**

**IIT Madras, B.S. in Data Science & Applications**