**Quiz Master V2 – Final Project Report**

# 👤 Author

Name: Vishal Bhandari

Roll Number: 22F1001645

Email: vishalbhandari@study.iitm.ac.in

I'm a passionate learner currently pursuing my B.S. in Data Science from IIT Madras. I'm particularly interested in full-stack development, scalable systems, and building AI-driven applications.

# 📌 Description

This project involves building a full-stack web application for multi-user quiz interaction. It supports quiz management by admins and quiz participation and performance tracking for users.

# 🧰 Technologies Used

• Flask – Web Framework  
• Flask-SQLAlchemy – ORM for SQLite database  
• Flask-Login – Authentication  
• Celery – For background tasks  
• Redis – Asynchronous task queue broker  
• Flask-Mail – Email reporting  
• Bootstrap 5 – Frontend styling  
• Vue 3 (CLI) – Frontend framework  
• Chart.js – Data visualization  
These were chosen for their performance, simplicity, and excellent community support in academic projects.

# 🗃️ DB Schema Design

Main Tables: User, Subject, Chapter, Quiz, Question, Score.  
  
• User: user\_id (PK), username (unique), password\_hash, role, created\_at  
• Subject: subject\_id (PK), name  
• Chapter: chapter\_id (PK), title, subject\_id (FK)  
• Quiz: quiz\_id (PK), title, chapter\_id (FK)  
• Question: question\_id (PK), quiz\_id (FK), question\_text, correct\_answer  
• Score: score\_id (PK), quiz\_id (FK), user\_id (FK), total\_score, timestamp  
  
Constraints: Foreign keys for referential integrity, cascade on delete.  
Design was normalized for modularity, efficiency, and scalability.

# 🔌 API Design

APIs were created for user registration, login, quiz creation, quiz attempts, and analytics. Endpoints are RESTful and include authentication guards. The detailed YAML for Swagger/OpenAPI documentation is submitted separately.

# 🏗️ Architecture & Features

The backend is organized into `models`, `controllers`, and `routes`. Templates are segregated by user roles under `templates/admin/` and `templates/user/`. Vue handles the dynamic frontend components.  
  
Features implemented:  
• Role-based login (admin/user)  
• CRUD for Subject, Chapter, Quiz, Question  
• Quiz attempt interface with timer  
• Score calculation and history dashboard  
• CSV export (user & admin)  
• Monthly email summary (Celery task)  
• Google Chat reminders via webhook

# 🎬 Video Demo

https://drive.google.com/file/d/1B\_2H2BmWDGezvLKohi3xJhkJmol3ZACt