Quiz Master V2 – MAD II Project Report

Name: Prakhar Mishra
Roll Number: 23F1001380
Course: Modern Application Development II (Jan 2025)
Project Title: Quiz Master – V2

1. Problem Statement

The objective was to build a multi-user quiz platform with distinct roles for **admin** and **user**. Admins manage subjects, chapters, quizzes, and questions. Users register, log in, attempt quizzes, and track their progress. The platform had to be built strictly using **Flask (API)**, **SQLite**, **VueJS (UI)**, **Bootstrap**, **Redis**, and **Celery**.

2. My Approach

I began by planning the database schema, then moved to route structure, UI flow, and background jobs.
Key modules were:

- **Admin (Quiz Master):** Create and manage academic content
- **User:** Attempt quizzes and view scores
- **Backend Jobs:** Daily reminders, monthly reports, CSV exports

I followed the milestone tracker carefully, testing functionality after each step. Attending the **Offline Boot Camp at IIT Madras** significantly enhanced my understanding of full-stack development. The mentors helped me learn Flask architecture, Redis, and Celery scheduling in-depth.

3. Technologies and Tools Used

- **Flask:** RESTful APIs, auth, and backend logic
- **SQLite:** Relational database using SQLAlchemy
- **VueJS:** Reactive UI with components
- **Bootstrap:** Responsive layout and design
- **Redis:** Caching layer + message broker
- **Celery:** Background job processing and scheduling
- **Others:** Flask-Security, Flask-JWT, HTML5 form validation

4. Key Features Implemented

Admin (Quiz Master):

- CRUD operations on subjects, chapters, quizzes, questions
- Set quiz timings
- Search users and manage data
- Trigger CSV report exports
- View summary dashboard

User:

- Register and login securely
- Attempt quizzes with timer
- View past attempts and scores
- Trigger CSV export of personal quiz history

Backend Jobs (Celery + Redis):

- Daily email reminders to inactive users

- Monthly quiz performance reports
- Asynchronous CSV exports

5. API Endpoints

Auth:

- `POST /api/auth/register` – Register a new user
- `POST /api/auth/login` – Login and receive token

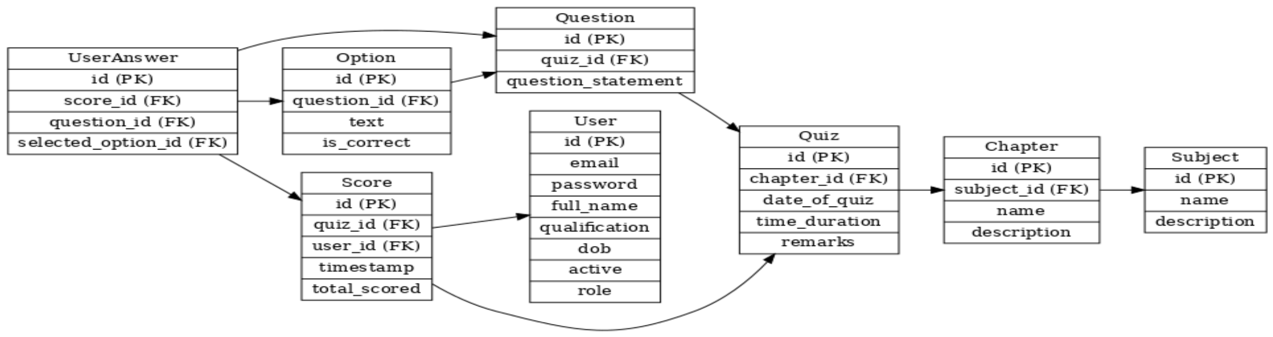
Admin:

- `POST /api/admin/subject` – Create subject
- `PUT /api/admin/quiz/:id` – Edit quiz
- `DELETE /api/admin/question/:id` – Delete question
- `GET /api/admin/users` – View user list

User:

- `GET /api/user/quizzes` – List available quizzes
- `POST /api/user/attempt/:quiz_id` – Submit quiz answers
- `GET /api/user/scores` – View past quiz attempts
- `POST /api/user/export-csv` – Trigger user CSV report

ER Diagram



My Learning Experience

This project gave me real-world experience building a full-stack web app. I did not use AI to generate code — only for project idea brainstorming and logic refinement. I also referred to a few senior projects only for structural ideas.
The offline boot camp and structured milestones pushed me to explore beyond course content. I now understand user auth, API caching, background workers, and how real systems handle scale and modularity.

Presentation Video:
https://drive.google.com/drive/folders/1RmW5um_n3-APm8atzJRe4YlwZtdMb8fp?usp=drive_link

AI used all over project is 3-4%