IITM Online BS Degree Program,

Indian Institute of Technology, Madras, Chennai

Tamil Nadu, India, 600036

A Final report for the MAD2 capstone Project

Submitted by

Name: Pallavi Sinha

Roll number: 21F2000980

I have made Quiz-Master application, which is a multi-user application that acts as an exam preparation site for multiple courses. The platform will support two roles:

- **Admin (Quiz Master)** – Responsible for managing users, subjects, chapters, quizzes, and questions.
- **User** – Can register, log in, attempt quizzes, and view scores.
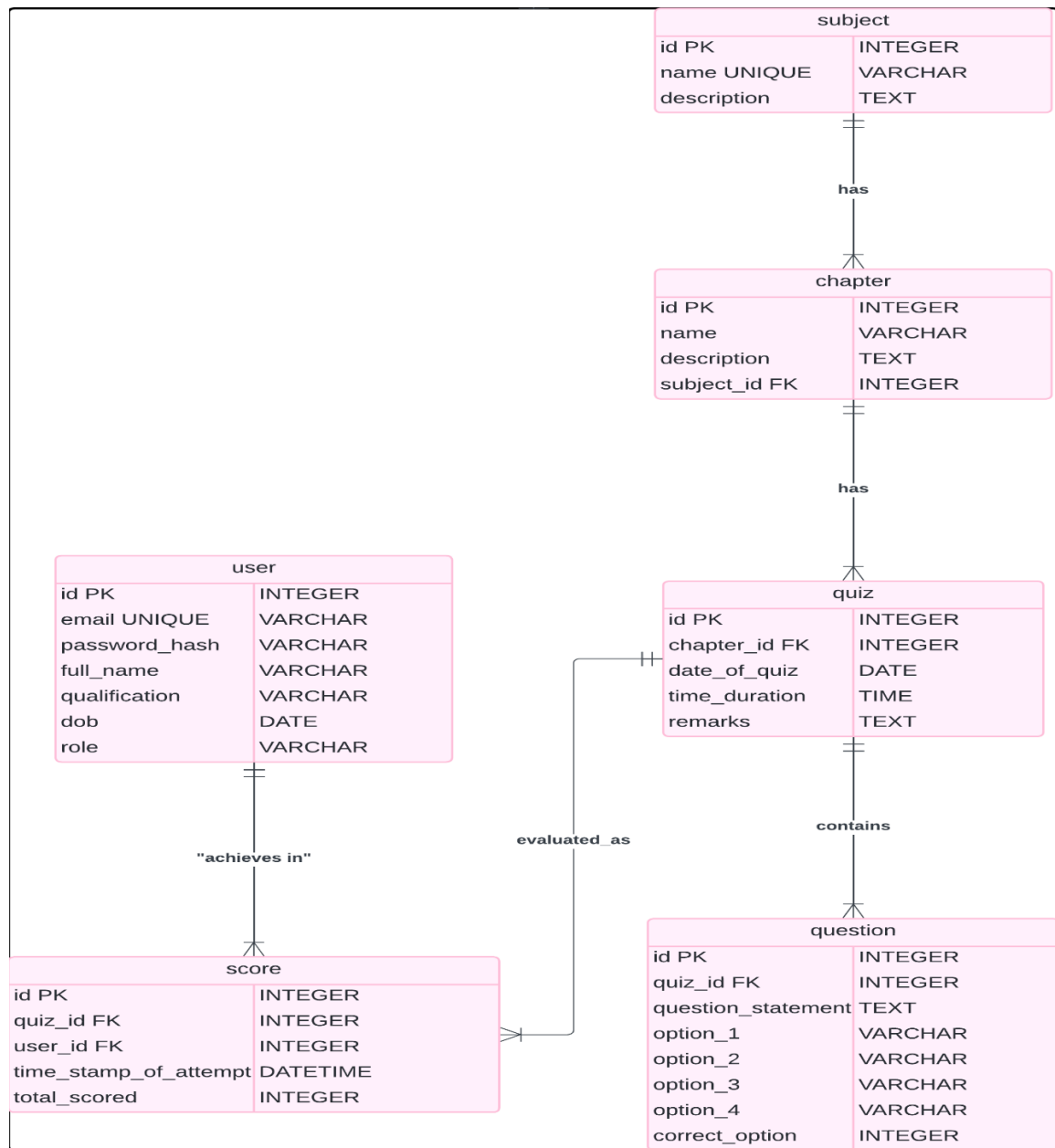
Frame Work Used –

- Flask for the API

- Vue-JS with CLI for UI

- SQLite for database

- Redis for caching

- Redis and Celery for batch jobs

Model description –

The ER diagram includes the following tables:

1. **User Table:** Stores user details such as Id, email, password, full name, qualification, Dob and scores
2. **Subject Table:** Contains subject details created by the admin.
3. **Chapter Table:** Subdivides subjects into chapters.

4. **Quiz Table:** Stores quiz details, including chapter ID, date of quiz, duration, and remarks.
5. **Question Table:** Stores quiz questions along with multiple-choice options.
6. **Scores Table:** Stores quiz scores along with details of user attempts.

subject	
id PK	INTEGER
name UNIQUE	VARCHAR
description	TEXT

has

chapter	
id PK	INTEGER
name	VARCHAR
description	TEXT
subject_id FK	INTEGER

has

user	
id PK	INTEGER
email UNIQUE	VARCHAR
password_hash	VARCHAR
full_name	VARCHAR
qualification	VARCHAR
dob	DATE
role	VARCHAR

quiz	
id PK	INTEGER
chapter_id FK	INTEGER
date_of_quiz	DATE
time_duration	TIME
remarks	TEXT

evaluated_as

"achieves in"

contains

score	
id PK	INTEGER
quiz_id FK	INTEGER
user_id FK	INTEGER
time_stamp_of_attempt	DATETIME
total_scored	INTEGER

question	
id PK	INTEGER
quiz_id FK	INTEGER
question_statement	TEXT
option_1	VARCHAR
option_2	VARCHAR
option_3	VARCHAR
option_4	VARCHAR
correct_option	INTEGER

System Design –

The Quiz Master Application is built using Vue.js for the frontend and Flask for the backend, with SQLite as the database. Admins have a dedicated dashboard where they can manage quizzes, view user performance, and export quiz data as a CSV. The application provides secure authentication, allowing users to register, log in, and take quizzes while tracking their progress. The admin panel enables quiz management, including creating,

editing, and deleting quizzes. The system is designed for efficiency and scalability, ensuring a smooth user experience with optimized API calls

API Endpoints -:

Authentication:

- POST /login – User login
- POST /refresh – Refresh JWT token
- POST /register – User registration
- User & Admin:
- GET/POST /users – Manage users
- GET/PUT/DELETE /users/<user_id> – User actions
- GET /adminscores – Admin view of scores
- GET /adminscores/<user_id> – User-specific scores
- GET /summary – Admin summary report
- GET /avgmarks – View average marks
- Subjects & Chapters:
- GET/POST /subjects – Manage subjects
- GET/PUT/DELETE /subjects/<subject_id> – Subject actions
- GET/POST /chapters – Manage chapters
- GET/PUT/DELETE /chapters/<chapter_id> – Chapter actions
- Quizzes & Questions:
- GET/POST /quizzes – Manage quizzes
- GET/PUT/DELETE /quizzes/<quiz_id> – Quiz actions
- GET /quiz/<chapter_id> – View quizzes by chapter
- GET /QuizQues/<quiz_id> – View quiz questions
- POST /scores – Submit quiz score
- GET /scores/<user_id> – View user scores

The Quiz Master application was successfully developed with all core functionalities implemented. It enables seamless quiz management, user participation, and result tracking, making it an effective learning tool.

Video Presentation link –

https://drive.google.com/drive/folders/1t63Q35al7yla1hHK1mhe7h8izWhTZe73?usp=sharing

In this link I have shown the demo of how my application is working. With all of its functionalities.