PROJECT REPORT

Modern Application Development II

Quiz Master

Author: MANAS RASTOGI

Email: 22f3001477@ds.study.iitm.ac.in

Project Statement:

It is a multi-user app (one requires an administrator and other users) that acts as an exam preparation site for multiple courses.

Technologies Used:

- Vue.js (CLI)
- Vue Router
- Axios
- Bootstrap
- CSS
- JavaScript
- LocalStorage
- Redis for caching
- Redis and Celery for batch jobs
- Database: SQLite
- Flask
- Python

Database Schema:

The Database schema comprises following Classes

- users: Stores user information, including id, email, password_hash, full_name, qualification, date_of_birth, and is_admin status. The id is the primary key, and email is a unique identifier.

- subjects: Holds data about different subjects, with columns for id (primary key), name, and description. This table organizes the broad topics covered.

- chapters: Contains details for each chapter within a subject, including id (primary key), name, description, and subject_id (foreign key referencing the subjects table).

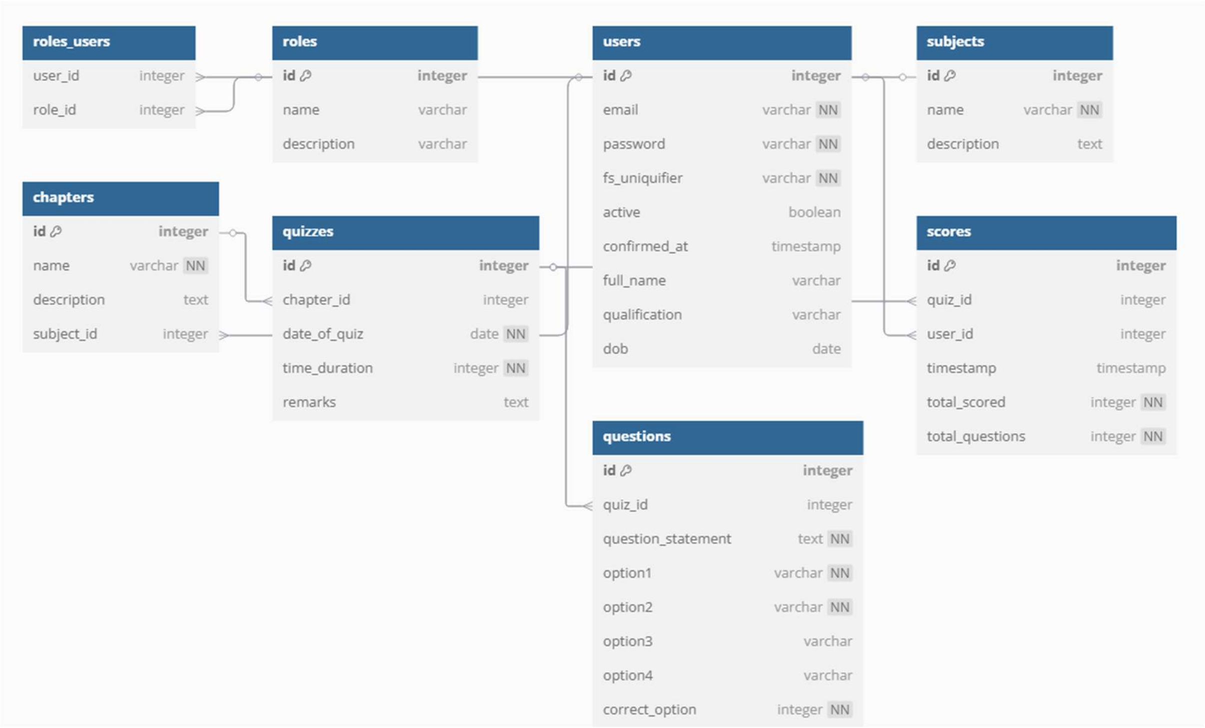
- quizzes: : Stores information on about quizzes, featuring id (primary key), chapter_id (foreign key referencing the chapters table), date_of_quiz, time_duration, and remarks. This table defines the properties of each quiz.

- questions: Holds the questions for each quiz, including id (primary key), quiz_id (foreign key referencing the quizzes table), question_statement, option1-option4, and correct_option. This table contains the individual questions that make up the quizzes

.• scores: Stores scores for each user's quiz attempts, with columns for id (primary key), quiz_id (foreign key referencing quizzes), user_id (foreign key referencing users), score, time_stamp_of_attempt, and time_taken. This table tracks user performance on quizzes..

DB Schema Design:

The Entity Relationship Diagram below shows the Database Schema and relationship:



roles_users	
user_id	integer
role_id	integer

roles	
id	integer
name	varchar
description	varchar

users		
id	integer	
email	varchar	NN
password	varchar	NN
fs_uniquifier	varchar	NN
active	boolean	
confirmed_at	timestamp	
full_name	varchar	
qualification	varchar	
dob	date	

subjects		
id	integer	
name	varchar	NN
description	text	

chapters		
id	integer	
name	varchar	NN
description	text	
subject_id	integer	

quizzes		
id	integer	
chapter_id	integer	
date_of_quiz	date	NN
time_duration	integer	NN
remarks	text	

scores		
id	integer	
quiz_id	integer	
user_id	integer	
timestamp	timestamp	
total_scored	integer	NN
total_questions	integer	NN

questions		
id	integer	
quiz_id	integer	
question_statement	text	NN
option1	varchar	NN
option2	varchar	NN
option3	varchar	
option4	varchar	
correct_option	integer	NN

<u>Architecture Design :</u>

The Model View Controller-MVC architecture has been implemented in this project. The project has following structure:

1. Templates - Contain all views divided into further sub-folders for clarity.
2. Models - Contain model.py defining classes for model as per schema.
3. Instance - Contains sqlite database file.
4. Static - Contains .css files
5. Application python file – contains routes, controllers.

Architecture and Features:

The main application logic and routes are within app.py, while models.py defines the database models.

The HTML templates for rendering different pages like the admin dashboard, quizzes, user scores, and

login/registration forms are located in the templates folder.

Static files such as CSS stylesheets (admin.css, style.css) are stored in the static directory.

Admin Features:

1. User Management: Can search, add (implicitly through registration), and potentially manage user

accounts.

2. Subject Management: Create, edit, and delete subjects.

3. Chapter Management: Create, edit, and delete chapters within subjects.

4. Quiz Management: Create, edit, and delete quizzes under specific chapters, including specifying

the date and dura on.

5. Ques on Management: Create, edit, and delete MCQ questions inside a specific quiz.

6. Data Overview: View summary charts related to users, subjects, and quizzes.

User Features:

1. Registration/Login: Can register and log in to the system with appropriate credentials.

2. Quiz Attempt: Can attempt any quiz of interest.

3. Score Recording: Every quiz score is recorded and stored.

4. Attempt History: Can view a history of previous quiz attempts.

5. Performance Tracking: Able to view summary charts of their own performance.

6. Timer: May encounter quizzes with a merge to manage me during the a empty.

Video link:

https://drive.google.com/drive/folders/1mnh-QmOLlHOEb3A-d-o6I86v80ZgW7ZS?usp=sharing