Morden Application Development(II) Capstone Project Quiz-master-v2

Project Report

Demo Video Link: https://drive.google.com/file/d/1VbX-nayAJj0k_uTExWNC_HCdTfzYZ_c4/view?usp=sharing

Frontend Technologies used:

- VueJS
- Javascript
- CSS
- ChartJS
- HTML5
- Vuex
- Vue-Router
- Axios

Backend Technologies used:

- Python
- Flask
- Flask SQLAlchemy
- Flask Security
- SQLite
- Flask-Restful
- bcrypt

About the Project:

The project is based on a Quiz app with a single admin where the admin can do all the basic CRUD operations on subjects, quizzes, chapters and questions, On the other hand we have Users who can register/login and view Live, Upcoming and Past quizzes. They can attempt the quiz multiple times and can also view their scores and other details of the attempt.

Frontend Engineering:

The frontend was designed using VueJS as the frontend javascript framework, for efficient state management and user experience, Vuex was used as the state management tool. To modularize the application, the code was broken into different view components and routing was done for the pages whenever it was felt to be necessary.

LifeCycle methods in VueJS were used to make the essential fetch api calls in the vuex store which imporved the user experience and made the code efficient.

Backend Engineering:

Python programming language was used to implement the backend of the project, Flask-Restful was used to create powerful REST APIs for both the POST as well as GET methods. SQLite was used as the database technology and SQLAlchemy was used to create the ORM so that we can programmatically manipulate the database.

Flask-Security module was used to implement the RBAC and to differentiate user and admins, the user store was created for the same.

Bcrypt was used to hash the passwords of the users and the admin, an extra security salt was setup so as to increase the security.

We have the following models in the database:

chapter:

```
id INTEGER NOT NULL,
subject_id INTEGER NOT NULL,
name VARCHAR NOT NULL,
description VARCHAR NOT NULL,
PRIMARY KEY (id),
FOREIGN KEY(subject_id) REFERENCES subject (id)
```

question:

```
id INTEGER NOT NULL,
```

quiz_id INTEGER NOT NULL,

question_title VARCHAR(255) NOT NULL,

question_statement TEXT NOT NULL,

option_1 VARCHAR NOT NULL,

option_2 VARCHAR NOT NULL,

option_3 VARCHAR NOT NULL,

option_4 VARCHAR NOT NULL,

correct_option VARCHAR NOT NULL,

marks INTEGER NOT NULL,

created_at DATETIME,

PRIMARY KEY (id),

FOREIGN KEY(quiz_id) REFERENCES quiz (id)

quiz:

id INTEGER NOT NULL,

chapter_id INTEGER NOT NULL,

title VARCHAR(255) NOT NULL,

date_of_quiz DATE NOT NULL,

time_duration TIME NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY(chapter_id) REFERENCES chapter (id)

quiz_attempt:

id INTEGER NOT NULL,

user_id INTEGER NOT NULL,

quiz_id INTEGER NOT NULL,

subject_id INTEGER,

score INTEGER,

max_score INTEGER,

started_at DATETIME,

completed_at DATETIME,

total_time_taken INTEGER,

responses JSON,

PRIMARY KEY (id),

FOREIGN KEY(user_id) REFERENCES user (id),

FOREIGN KEY(quiz_id) REFERENCES quiz (id),

FOREIGN KEY(subject_id) REFERENCES subject (id)

role:

id INTEGER NOT NULL,

name VARCHAR NOT NULL,

description VARCHAR NOT NULL,

PRIMARY KEY (id),

UNIQUE (name)

subject:

id INTEGER NOT NULL,

name VARCHAR NOT NULL,

description VARCHAR NOT NULL,

PRIMARY KEY (id),

UNIQUE (name)

user:

id INTEGER NOT NULL,

email VARCHAR NOT NULL,

password VARCHAR NOT NULL,

fs_uniquifier VARCHAR NOT NULL,

active BOOLEAN,

PRIMARY KEY (id),

UNIQUE (email),

UNIQUE (fs_uniquifier)

user_answer:

id INTEGER NOT NULL,

attempt_id INTEGER NOT NULL,

question_id INTEGER NOT NULL,

selected_option VARCHAR NOT NULL,

is_correct BOOLEAN NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY(attempt_id) REFERENCES quiz_attempt (id),

FOREIGN KEY(question_id) REFERENCES question (id)

user_roles:

id INTEGER NOT NULL,

user_id INTEGER,

role_id INTEGER,

PRIMARY KEY (id),

FOREIGN KEY(user_id) REFERENCES user (id),

FOREIGN KEY(role_id) REFERENCES role (id)