Modern Application Development 2 - Project (Quiz Master - V2)

Name: Sayan Bhowmick

Roll No: 22f3001657

Email: 22f3001657@ds.study.iitm.ac.in

I am Sayan Bhowmick, a B-Tech Engineer in the field of Electronics and Communication. As a staunch believer in continuous learning, I find it prudent to enroll myself in this Bachelor's degree program.

Description:

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

In this app, the admin creates quizzes for corresponding chapters which belong to their corresponding subjects, wherein the admin can manage the subjects, its corresponding chapters, quizzes & all the questions by performing CRUD operations. Also, the admin will be able to search for any user from the admin dashboard to find any user.

The new user can register in the app and when signed in can view the various quizzes available for the user to attempt. Also, there is a search functionality using which the user can search for all the subjects on which all the quizzes are based on.

Overall, this quizmaster app aims to streamline operations, enhance user experience, and promote efficient quiz access to the users to build & grow a learners' community.

Technologies Used:

flask - Flask for creating the Backend of the application - **Controller**

Flask-sqlalchemy - SQLAlchemy for Creating database - **Model**

VueJS - for creating web pages using JavaScript - **View**

CSS - to add styling effects

flask_security(token based) - Security

Celery – Schedules

and

Redis – Backend database and Cache

Gmail SMTP App – SMTP server

Architecture and Features:

The code for the app is organised based on the user type and the tasks to be performed by them. The code is placed inside the project folder as an **app.py** file, but all the routes are placed inside **routes.py** file. The app starts with the landing page where one can see the home page welcome note and total 3 buttons on the navigation bar - Home,

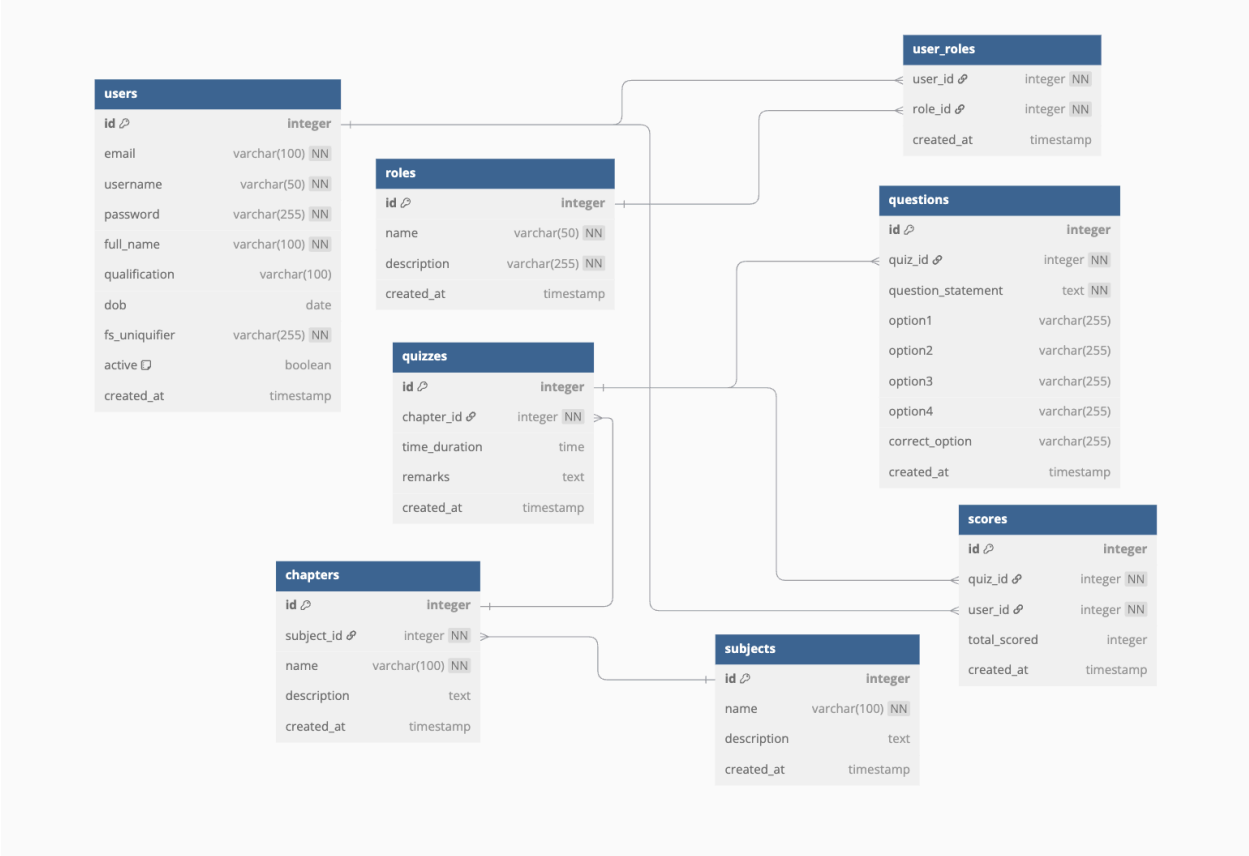
Login and Register, the login page is rendered when a user clicks on the **Login** button and similarly the registration page is rendered when a user clicks on the **Register** button. Once a new user is registered, the user can log in. In this app, there are two types of users, namely **admin** and **general user**.

Initially the admin logs in which opens up the **admin dashboard**, which has a button to add new subjects. The admin has relevant links to modify or delete any subject. The other links provided in the admin dashboard enables the admin to create, update & delete chapters for corresponding subjects, quizzes for corresponding chapters and questions for corresponding quizzes for the users to attempt.

The general user, when logged in, will be directed to the **user dashboard**, with various links for the user to see the various quizzes to attempt. Also, the user dashboard has a search box to perform search the subject names for which the user can expect quizzes on its dashboard. The other links allow the user to check its previous attempted quizzes , get its overall performance report to be downloaded in a CSV file and finally there is a link for the user to log out.

All the web pages are saved in the **pages** folder in the frontend folder and using the respective API is connected to the backend flask server. The file **models.py** which is responsible for creating and maintaining the database has **eight** tables namely **User**, **Role**, **UserRoles**, **Subject**, **Chapter**, **Quizzes**, **Questions**, and **Score**. The controllers used to access the model and modify the view are placed in the file **routes.py**.

DB Schema Design:



Flowchart of the App:

@app.route("/login", methods=["POST"])

@app.route("/logout", methods=["POST"])

@app.route("/register", methods=["POST"])

@app.route("/admin_dashboard", methods=["GET"])

@app.route("/user_dashboard", methods=["GET"])

@app.route("/create_subject", methods=["POST"])

@app.route("/delete_subject/<int:subject_id>", methods=["DELETE"])

@app.route("/update_subject/<int:subject_id>", methods=["PUT"])

@app.route("/get_subject/<int:subject_id>", methods=["GET"])

@app.route("/create_chapter/<int:subject_id>", methods=["POST"])

@app.route("/get_chapter/<int:chapter_id>", methods=["GET"])

@app.route("/update_chapter/<int:chapter_id>", methods=["PUT"])

@app.route("/delete_chapter/<int:chapter_id>", methods=["DELETE"])

@app.route("/get_chapters/<int:subject_id>", methods=["GET"])

@app.route("/get_quizzes/<int:chapter_id>", methods=["GET"])

@app.route("/create_quiz/<int:chapter_id>", methods=["POST"])

@app.route("/delete_quiz/<int:quiz_id>", methods=["DELETE"])

@app.route("/get_quiz/<int:quiz_id>", methods=["GET"])

@app.route("/update_quiz/<int:quiz_id>", methods=["PUT"])

@app.route("/create_question/<int:quiz_id>", methods=["POST"])

@app.route("/get_question/<int:question_id>", methods=["GET"])

@app.route("/update_question/<int:question_id>", methods=["PUT"])

@app.route("/delete_question/<int:question_id>", methods=["DELETE"])

@app.get("/get_all_quizzes")

@app.get("/get_questions/<int:quiz_id>")

@app.route("/get_time_duration/<int:quiz_id>", methods=["GET"])

@app.route('/submit_quiz', methods=['POST'])

@app.route("/get_scores/<int:quiz_id>", methods=["GET"])

@app.route("/search", methods=["POST"])

@app.route("/search_subject", methods=["POST"])

@app.route("/previousAttempts", methods=["GET"])

@app.route('/user_performance', methods=['GET'])

@app.get('/admin_dashboard_data')

@cache.cached(timeout = 60)

Video Link: Project Demo