

Student Details-

Name - Deepak Kumar Maurya

Roll number – 23f3004133

Email – 23f3004133@ds.study.iitm.ac.in

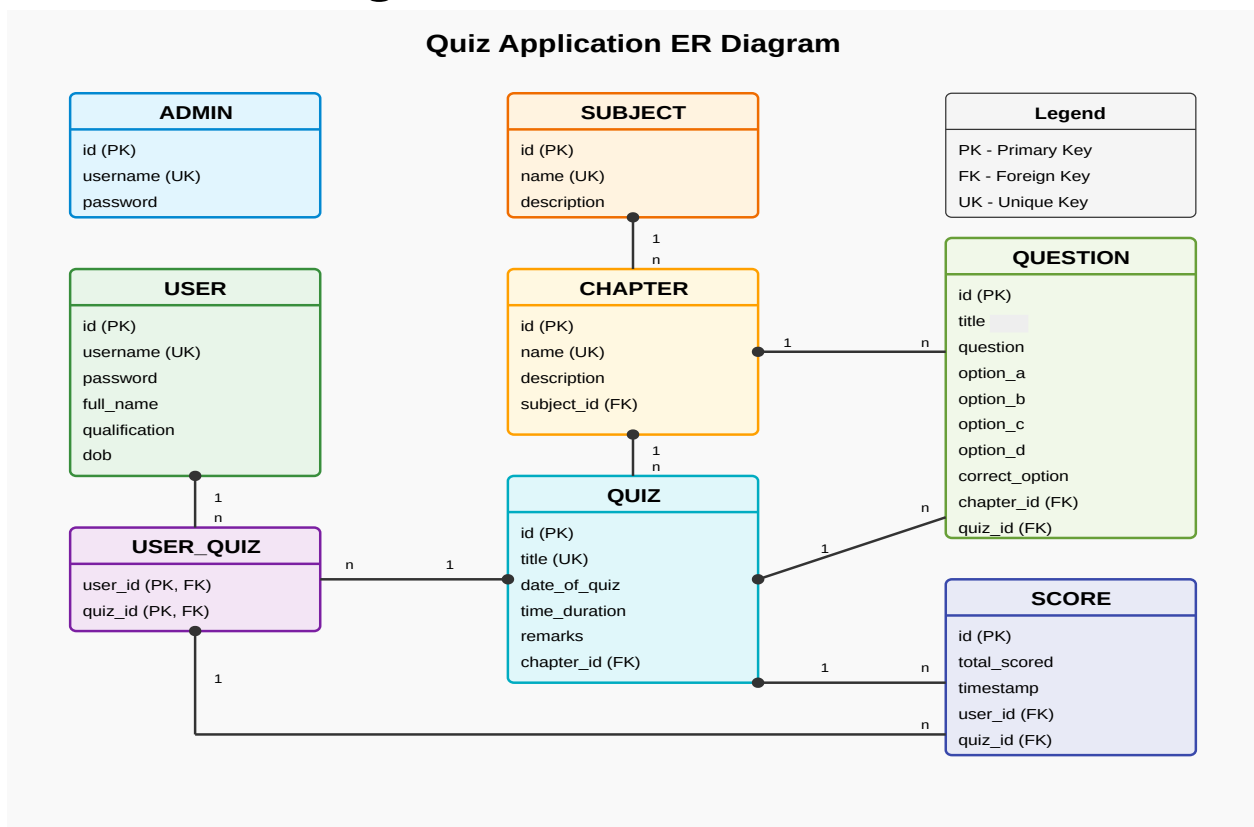
Description-

This web app is basically a single Administrator (Admin) and multi-user web application that will help for the exam preparation. Admin is allowed to all the CRUD operation for subject, chapter under subject, quiz under chapter and question under the quiz. The created quiz will then populated to the user dashboard the user can attempt the quiz within the specified timeline. Once quiz will be submitted by user or timeline end(quiz will submit automatically), user will redirected to the score board. Admin and user have search functionality and the summary part for the Analysis.

Frameworks and libraries used-

- **Flask framework-** for developing the application.
- **jinja2-** for templating and HTML generation.
- **Bootstrap5-** for styling and designing purpose.
- **Flask-SQLAlchemy-** for Sqlite3 database operations.
- **Flask-RESTful-** for implementing RESTful APIs.
- **Matplotlib-** for generation the chart based on the data.

DB Schema Design-



Tables-

- Admin – id is PRIMARY KEY and AUTO-INCREMENTED. All attribute are not null. Default Admin username will be “admin12345@gmail.com” and password will be “12345”.
- User – id is PRIMARY KEY and username is UNIQUE attribute. All attribute are not null except qualification and dob.
- Subject – id is PRIMARY KEY and name is UNIQUE and both are not null but description can be null.
- Chapter – id is PRIMARY KEY and name is UNIQUE and both are not null but description can be null.
- Quiz – id is PRIMARY KEY and title is UNIQUE
- Question – contains all question along with options and correct answer
- Score – will contains all the score
- UserQuiz – is a association table for User and Quiz table.

Relationships-

- Subject – Chapter (one to many relationship) – one chapter can have multiple subject but one subject can belong only one chapter.
- Chapter – Quiz (one to many relationship) – one chapter can have multiple quiz but one quiz can belong only one chapter.
- Quiz – Question (one to many relationship) – one quiz can have multiple question but one question can belong only one quiz.
- User – Quiz (many to many relationship) – one user can attempt multiple quiz and a quiz attempted by multiple users
- User – Score (one to many relationship) – one user can have many score but one score can belong to only one user.

API Design and Endpoints-

There is three 4 resources created for API using Resource class from Flask-RESTful. They can handle only for GET request in this application and output will be obtained in JSON formate.

- Subject API – endpoint is “/api/subjects”
- Chapter API– endpoint is “/api/chapters”
- Quiz API– endpoint is “/api/quizzes”
- Score API– endpoint is “/api/scores”

Other endpoints are:

- “/” or “/login” – login page
- “/register” – registration page
- “/admin_dash” – admin dashboard
- “/add_subject” – page for adding new subject
- “/<int:subject_id>/delete” – for delete a particular subject id
- “/<int:subject_id>/edit” – for update a particular subject id
- “add_chapter/<int:subject_id>” - page for adding new chapter under particular subject.

- “/admin_dash/<int:chapter_id>/delete”- for delete chapter from subject
- “/admin_dash/<int:chapter_id>/edit”- for update chapter in subjects
- “/add_quiz” – adding new quiz page
- “/quiz/<int:quiz_id>/delete”- delete the quiz
- “/quiz/<int:quiz_id>/edit”- update the quiz
- “/add_question/<int:quiz_id>” - add new question under quiz
- “/question/<int:question_id>delete” – to delete the question
- “/question/<int:question_id>edit” – to update the question
- “/user_read/<int:user_id>” – user details page that can be read by admin.
- “/subject_read/<int:subject_id>” - subject details page that can be read by admin
- “/chapter_read/<int:chapter_id>”- chapter details page that can be read by admin
- “/quiz_read/<int:quiz_id>”- quiz details page that can be read by admin
- “/question_read/<int:question_id>” - question details page that can be read by admin
- “/admin_summary”- for summary page of admin.
- “/user_dash/<int:user_id>” - user dashboard for a particular user
- “/user_quizview/<int:user_id>/<int:quiz_id>” - to see the quiz details
- “/user_score/<int:user_id>” - the scoreboard of user
- “/user_summary/<int:user_id>” - for summary page of user
- “/search” - to search for admin (user/subject/chapter/quiz/question)
- “/user_search/<int:user_id>” - to search for user (subject/chapter/quiz)
- “/quiz_handle/<int:user_id>/<int:quiz_id>” - to handle the quiz attempt and record the score obtained by user within given timeline.
- “/user_subject_read/<int:user_id>/<int:subject_id>” - to read the details of subject on user dashboard
- “/user_chapter_read/<int:user_id>/<int:chapter_id>” - to read the details of chapter on user dashboard
- “/user_quiz_read/<int:user_id>/<int:quiz_id>” - to read the details of quiz on user dashboard

Architecture of files-

application → (api.py, controllers.py, database.py, models.py)

static → (style.css)

templates → (all html templates)

Quiz Master (root folder) → (application, static, templates)

Video-

<https://drive.google.com/file/d/1iq6WjeUlv8Py6vKlkeCIFiJIoqNEZIF5/view?usp=sharing>