

Author

Name: M.H.Ananta sri

Roll number: 23f2000427

Student email: 23f2000427@ds.study.iitm.ac.in

About Author: A curious student with a passion for learning and experimenting. Believe in learning from mistakes and finding better ways to improve.

Description

The Quiz Master App is an exam preparation site for multiple subjects.

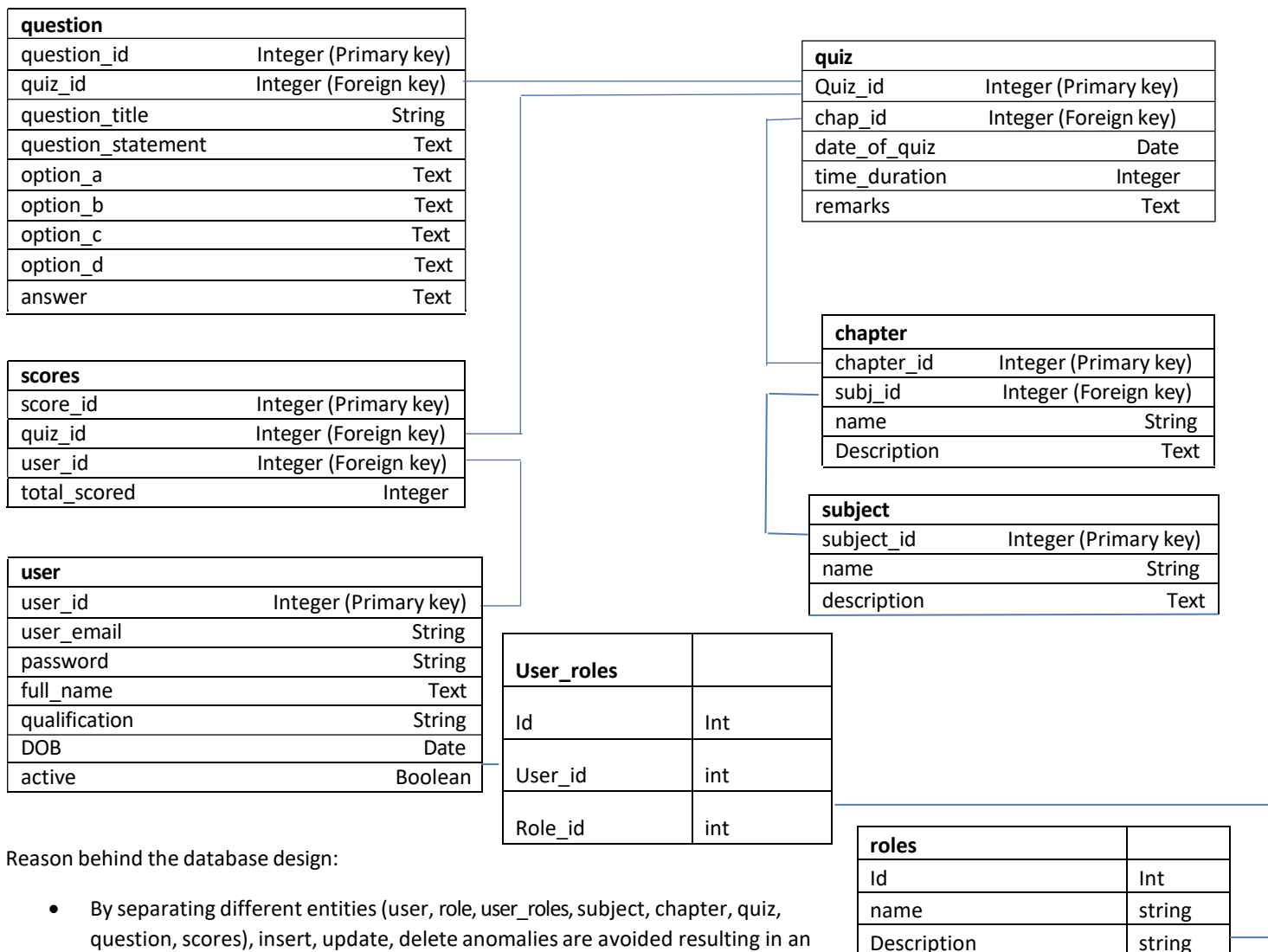
The “Admin” (Quiz Master) can create, edit, and delete quizzes on various chapters.

The “Users” can attempt any of these quizzes and view their scores.

Technologies used

1. Backend – Flask (REST API)
2. SQLite – for storing and managing application database
3. Flask-SQLAlchemy – helps in interacting with the SQLite Database
4. For Frontend – VueJS + Bootstrap + JS
5. Chart.js – used for visualization
6. datetime – used for handling date and time operations
7. Background jobs -Redis-celery

DB Schema Design



Reason behind the database design:

- By separating different entities (user, role, user_roles, subject, chapter, quiz, question, scores), insert, update, delete anomalies are avoided resulting in an organised database management.

- In the user table, the “role” attribute helps to distinguish between admin and users. The default value for role is user. This attribute helps ensure that only the admin can create, modify, or delete the quizzes and other details while the users are only allowed to attempt these quizzes.
- In the subject table, the subject_id attribute helps to identify each subject.
- Relationship between tables is easy to establish.
- In the chapter table, chapter_id identifies the chapter and subj_id is the foreign key that identifies the subject to which the chapter belongs.
- In the quiz table, Quiz_id identifies the quiz and chap_id is the foreign key that identifies the chapter to which the quiz belongs.
- In the question table, question_id identifies the question and quiz_id is the foreign that identifies the quiz to which the question belongs.
- In the scores table, score_id identifies the unique score, user_id and quiz_id are the foreign keys that link the score to the respective user and quiz.

Architecture

Follows M-V-C architecture as:

- **Models:** SQLAlchemy models define all entities and relationships.
- **Views (Frontend):** VueJS components communicate via REST API.
- **Controllers (API):** Flask routes handle login, quizzes, submissions, admin actions, exports, and reports.

App Features

- Authentication – secured login, register, and logout.
- Role-based access:

Admin – Only the admin can create, edit, and delete subjects, chapters, quizzes, and questions.

Users – Can only view or attempt the quizzes on the scheduled dates within the given time duration. They can view their score. Users can only view and attempt the quizzes available on or after the current date

- Search:
Admin can search for subjects, quizzes, and users.
- Summary:
Admin – can view charts on “subject-wise user attempts” and “subject-wise top scores”.
Users – can view charts on “month-wise number of quizzes attempted” and “subject-wise number of quizzes”.
- Triggers download of quiz activity report for users
- User receives email/chat notifications for: New quizzes, Progress reports
- Exports all quiz/user performance reports (CSV)

Video

https://drive.google.com/file/d/1GoiEHERzJlilmqHbzuSxts9gq1oSLNkS/view?usp=drive_link