

Author

Varun Sharma

23f2005028

23f2005028@ds.study.iitm.ac.in

Description

The Quiz Master Project is an application where users can attempt multiple-choice questions. It also includes features like quiz search, subject and quiz statistics via charts, and integration with a Flask framework for backend.

Technologies used

- Frontend: Vue.js (via CDN), Bootstrap.
- Backend: Flask with Flask-security and authentication.
- Task Queue: celery with Redis for background task handling.
- SQLite: Database for data persistence.
- Email & other tools: Mailhog for testing email functionalities, chart.js for visualizations.

DB Schema Design

Stores user-related details:

Column Name	Data Type	Constraints	Description
id	Integer	Primary Key, Auto-increment	Unique identifier for each user
username	String(150)	Unique, Not Null	User's login username
password	String(150)	Not Null	Hashed password for authentication
fullname	String(150)	Not Null	Full name of the user
qualification	String(150)	Not Null	User's academic qualification
dob	Integer	Not Null	Date of birth (should ideally be a <code>Date</code> , but stored as <code>Integer</code>)

Reasoning:

- Username is unique to prevent duplicate accounts.
- Password should be stored securely using hashing techniques.

Stores different subjects available for quizzes:

Column Name	Data Type	Constraints	Description
id	Integer	Primary Key, Auto-increment	Unique identifier for each subject
subjectname	String(150)	Unique, Not Null	Name of the subject
subjectdescription	String(150)	Unique, Not Null	Description of the subject

Reasoning:

- Subject name and subject description are unique to avoid duplicate subjects.

- Each subject can have multiple chapters, form a one-to-many relationship with chapters.

Stores chapter-wise breakdown of each subject:

Column Name	Data Type	Constraints	Description
id	Integer	Primary Key, Auto-increment	Unique identifier for each chapter
chaptername	String(150)	Unique, Not Null	Name of the chapter
chapterdescription	String(150)	Unique, Not Null	Description of the chapter
subject_id	Integer	Foreign Key (subjects.id), Not Null	References <code>subjects</code> table

Reasoning:

- A chapter belongs to a subject (many-to-one relationship).
- A subject can have multiple chapters (one-to-many relationship).

Stores information about quizzes related to different chapters:

Column Name	Data Type	Constraints	Description
id	Integer	Primary Key, Auto-increment	Unique identifier for each quiz
chapter_id	Integer	Foreign Key (chapters.id), Not Null	References <code>chapters</code> table
date_of_quiz	Date	Not Null	Date when the quiz is scheduled
time_duration	Integer	Not Null	Time limit for the quiz in minutes

Reasoning:

- A quiz is linked to a chapter (many-to-one relationship).
- Multiple quizzes can belong to the same chapter (one-to-many relationship).
- Date of Quiz ensures quizzes are scheduled at a specific date.

Stores questions related to quizzes:

Column Name	Data Type	Constraints	Description
id	Integer	Primary Key, Auto-increment	Unique identifier for each question
quiz_id	Integer	Foreign Key (quizzes.id), Not Null	References <code>quizzes</code> table
title	String	Not Null	Title or category of the question
question_statement	String	Not Null	Full text of the question
option1	String	Not Null	First answer option
option2	String	Not Null	Second answer option
option3	String		Third answer option (optional)
option4	String		Fourth answer option (optional)
correct_option	String(10)	Not Null	Stores the correct answer option (e.g., "option1")

Reasoning:

- A quiz consists of multiple questions (one-to-many relationship).
- The correct answer is stored as a reference.
- Some questions may have fewer than four options, so option3 and option4 are optional.

Stores the scores of users after attempting quizzes:

Column Name	Data Type	Constraints	Description
id	Integer	Primary Key, Auto-increment	Unique identifier for each score entry
quiz_id	Integer	Foreign Key (quizzes.id), Not Null	References <code>quizzes</code> table
user_id	Integer	Foreign Key (users.id), Not Null	References <code>users</code> table
time_stamp_of_attempt	DateTime	Default: <code>datetime.utcnow</code>	Records the time of quiz attempt
total_scored	Integer	Not Null	Score obtained by the user

Reasoning:

- A user can attempt multiple quizzes, forming a many-to-one relationship.
- Time stamp of attempt helps track when the quiz was taken.

Relationships Between Tables

1. One-to-Many Relationships:

- **Subjects** → **Chapters** (subjects.id → chapters.subject_id)
- **Chapters** → **Quizzes** (chapters.id → quizzes.chapter_id)
- **Quizzes** → **Questions** (quizzes.id → questions.quiz_id)
- **Users** → **Scores** (users.id → scores.user_id)
- **Quizzes** → **Scores** (quizzes.id → scores.quiz_id)

2. Many-to-One Relationships:

- **Scores** → **Users** (scores.user_id → users.id)
- **Scores** → **Quizzes** (scores.quiz_id → quizzes.id)

Architecture and Features

User Authentication and Profile Management:

- Users have **unique usernames and hashed passwords** for authentication.
- The system stores **user profiles**, including full name, qualification, and date of birth.

Subject and Chapter Management:

- Subjects are categorized with a name and description.
- Each subject contains multiple chapters to organize content effectively.

Quiz Management:

- Each quiz is associated with a chapter.
- Quizzes have a date and a time limit for completion.

Question Management:

- Each quiz consists of multiple-choice questions with up to four options.
- One of the options is marked as the correct answer.

Score Tracking:

- Each user's quiz attempts are recorded with scores and timestamps.
- The system tracks when a quiz was taken and how much the user scored.

Quiz Summary and Analytics:

- Users can view a summary of their attempted quizzes.
- Admins can see quiz participation trends.

Video

<https://drive.google.com/file/d/1VdAWYTrkE0d6D6QuIQDL1RUKF2ky5E03/view?usp=sharing>