

Quiz Master – V2

Project Report

Name: Manish Khatri

Roll No.: 21f3002258

Email: 21f3002258@ds.study.iitm.ac.in

I am a Diploma level student of the online B.S. Data Science and Applications course of IIT Madras.

Description

The "Quiz Master Application" is a comprehensive online quiz management system designed to facilitate interactive learning through subject-based quizzes. The platform enables administrators to create and manage educational content including subjects, chapters, and quizzes with multiple-choice questions. Students can register, browse available quizzes by subject and chapter, take timed assessments, and track their performance over time. The platform is built using SQLite for data persistence, Flask for the RESTful backend API, and Vue.js for the modern user interface.

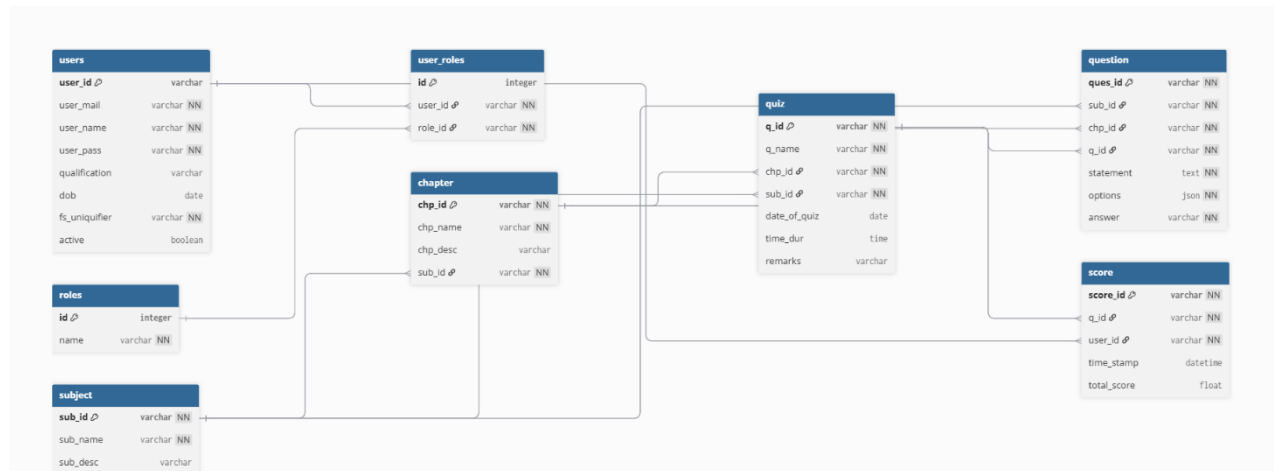
Redis provides intelligent caching to improve performance, while Celery handles background tasks including daily reminders, monthly reports, and CSV exports. The system implements role-based access control with separate interfaces for administrators and students.

Frameworks and Libraries used

- **Flask microframework** for backend API development
- **SQLite database** with Flask-SQLAlchemy as the ORM
- **Flask-Security** for authentication and role-based access control
- **Vue.js 3** with Composition API for reactive frontend
- **Bootstrap 5** for responsive UI design and styling
- **Chart.js** for interactive dashboard statistics and analytics
- **Werkzeug.security** for password hashing and verification
- **Redis** for caching and session management
- **Celery** with Redis broker for background task processing
- **SMTP** integration for email notifications and reports
- **CORS** support for cross-origin API communication

DB Schema Design

These are the tables in the database:-



Users Table (users):

- user_id (Primary Key): Unique identifier for each user
- user_mail: Email address for authentication and communication
- user_name: Full name of the user
- user_pass: Hashed password for security
- qualification: Educational background of the user
- dob: Date of birth
- fs_uniquifier: Flask-Security unique identifier
- active: Account status flag

Roles Table (role):

- id (Primary Key): Role identifier
- name: Role name (admin, customer)

UserRoles Table (user_roles):

- id (Primary Key): Relationship identifier
- user_id: Foreign key to users table
- role_id: Foreign key to role table

Subjects Table (subject):

- sub_id (Primary Key): Subject identifier
- sub_name: Name of the subject
- sub_desc: Description of the subject

Chapters Table (chapter):

- chp_id (Primary Key): Chapter identifier
- chp_name: Name of the chapter
- chp_desc: Description of the chapter
- sub_id: Foreign key to subject table

Quizzes Table (quiz):

- q_id (Primary Key): Quiz identifier
- q_name: Name of the quiz
- chp_id: Foreign key to chapter table
- sub_id: Foreign key to subject table
- date_of_quiz: Scheduled date for the quiz
- time_dur: Duration of the quiz
- remarks: Additional notes about the quiz

Questions Table (question):

- ques_id (Primary Key): Question identifier
- sub_id: Foreign key to subject table
- chp_id: Foreign key to chapter table
- q_id: Foreign key to quiz table
- statement: Question text
- options: JSON array of multiple choice options
- answer: Correct answer for the question

Scores Table (score):

- score_id (Primary Key): Score identifier
- q_id: Foreign key to quiz table
- user_id: Foreign key to users table
- time_stamp: When the quiz was taken
- total_score: Final score achieved

Architecture and Features

The application follows a client-server architecture with clear separation between frontend and backend. The project structure includes:

- **app.py:** Main Flask application entry point
- **backend/:** Contains models, routes, tasks, and configuration
- **frontend/:** Vue.js application with components and stores
- **instance/:** Database and configuration files

Features:

Authentication & Authorization:

- Common login form for Admin and Student users
- Token-based authentication with Flask-Security
- Role-based access control (RBAC)
- Secure password hashing and verification

Content Management:

- **Subject Management:** Create, read, update, delete subjects
- **Chapter Management:** Organize chapters within subjects
- **Quiz Management:** Create quizzes with date/time scheduling
- **Question Management:** Add MCQ questions with multiple options

Analytics & Reporting:

- **Real-time Charts:** Subject-wise performance, user attempts distribution
- **Performance Tracking:** Individual and aggregate statistics
- **Export Capabilities:** CSV downloads for data analysis
- **Dashboard Metrics:** Active users, quiz attempts, success rates

API Resource Endpoints

Flask RESTful has been used to create a Section resource with endpoints to get and post section details.

Presentation Video Link

<https://drive.google.com/drive/folders/1lgXR-DRtIL23lvjfDlv84oMbQdvH2chh?usp=sharing>