Quiz Master Project Report

# Author

Name: Sidhaarth Shree  
Roll Number: 22f3001480  
Email: [22f3001480@ds.study.iitm.ac.in](mailto:22f3001480@ds.study.iitm.ac.in)

# Description

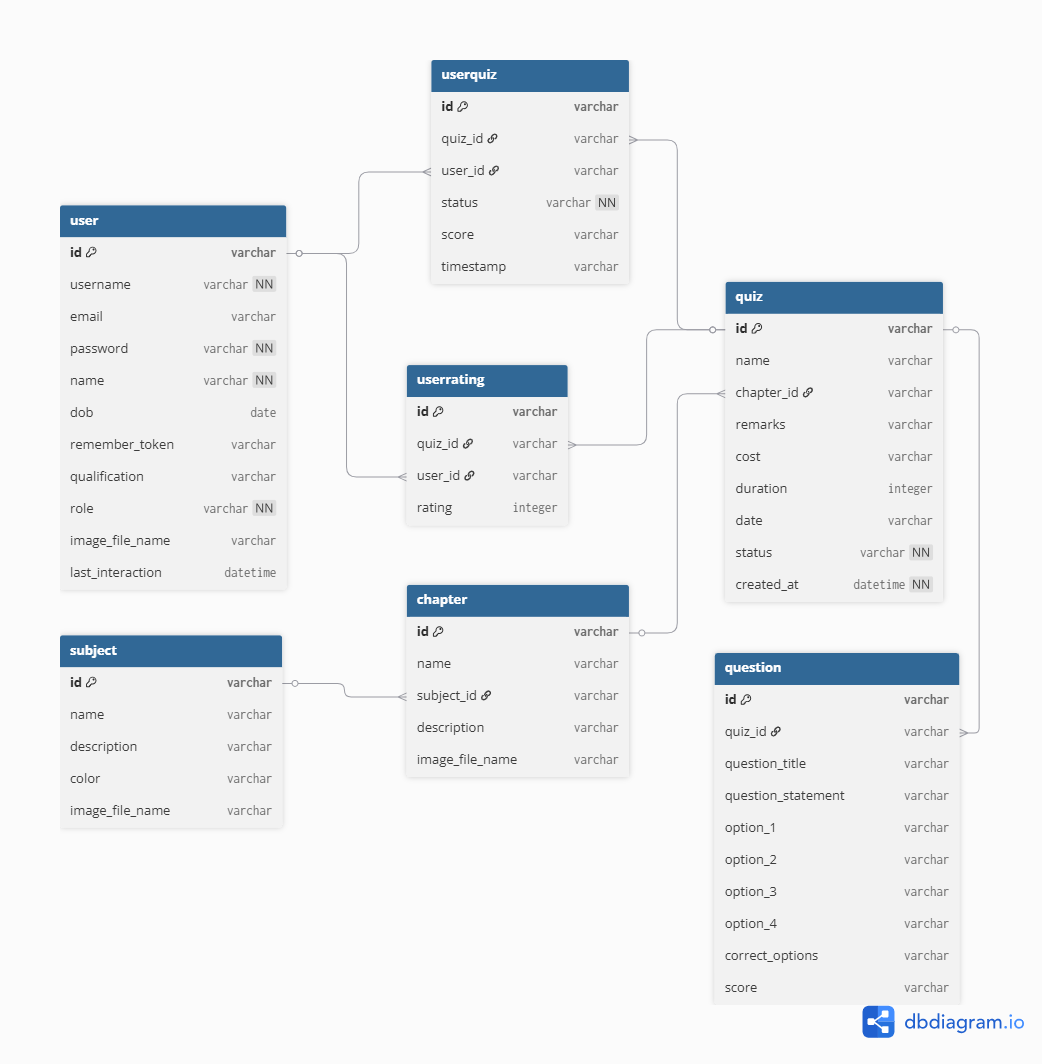
A full-stack quiz management system enabling user authentication, quiz participation, scoring, reporting, and persistent user state. Backend is powered by Flask and Celery (with Redis), and the frontend is built with Vue.js.

# Technologies Used

Backend:  
 - Python 3  
 - Flask (REST API framework)  
 - Flask-Login (user authentication)  
 - Flask-CORS (CORS handling)  
 - SQLAlchemy (ORM for SQLite DB)  
 - Celery (background tasks)  
 - Redis (cache and Celery broker)  
  
Frontend:  
 - Vue.js (SPA framework)  
 - Axios (HTTP client)  
 - HTML/CSS/JS  
  
Purpose: Flask provides a lightweight API backend, SQLAlchemy manages DB, Celery/Redis handle async tasks and caching, Vue.js delivers a reactive UI, and Flask extensions add authentication and CORS support.

# DB Schema Design (Summary)

- User: id (PK), username, password, name, role, email, dob, qualification, image\_file\_name, last\_interaction  
- Quiz: id (PK), name, date, duration, cost, status, chapter\_id (FK)  
- Question: id (PK), quiz\_id (FK), question\_statement, options, correct\_options, score  
- UserQuiz: id (PK), user\_id (FK), quiz\_id (FK), score, timestamp, status  
- UserRating: id (PK), user\_id (FK), quiz\_id (FK), rating  
- Subject, Chapter: id (PK), name, image\_file\_name, color, subject\_id (FK)



# API Design

RESTful APIs for:  
 - User registration & login  
 - Quiz listing & participation  
 - Answer submission & scoring  
 - User rating & profile management  
 - CSV export of quiz data  
 - Persistent user tab state (Redis/Celery)  
  
All endpoints are implemented in Flask blueprints. Async endpoints (e.g., CSV export, tab state caching) use Celery tasks. API details are defined in a separate YAML file.

# Architecture and Features

- Backend: Organized into blueprints (routes/), models (models.py), Celery tasks (tasks.py), and configuration files. HTML templates for reports.  
- Frontend: Vue.js SPA in frontend/ with modular components for each view.  
- Features: User authentication, quiz registration, answer submission, scoring, statistics, CSV export, persistent user tab state (Redis/Celery), admin dashboard, user profile with image upload, and quiz statistics.  
- CORS: Enabled for seamless frontend-backend communication.