

PROJECT REPORT

Welcome to QuizGen

Track your progress, challenge your limits, and grow smarter every day.

PROJECT REPORT BY VAIBHAV SONI

23F2004675

MODERN APPLICATION DEVELOPMENT – II

JANUARY – 2025 TERM

IIT MADRAS

Student Details

Name	Vaibhav Soni
Roll Number	23f2004675
Email	23f2004675@ds.study.iitm.ac.in

ABOUT ME

**I AM A FULL STACK DEVELOPER.
I'M A KNOWLEDGE SEEKER WHO IS DRIVEN TO STUDY,
DEVELOP, AND SUCCEED IN JAVA,
PYTHON, DBMS AND FLASK.
I'M ALSO EXCITED TO WORK HARD AND CONTRIBUTE
TO THE SUCCESS OF MY TEAM.**

PROJECT DESCRIPTION

Quiz Master is a multi-role exam preparation platform designed for students and educators, It allows admins (Quiz Masters) to create structured quizzes across different subjects and chapters, while users (students) can attempt quizzes, track their scores, and receive performance reports.



QuizGen- Tech Stack

Backend & API

- **Flask** – Lightweight Python web framework for backend logic.
- **Flask-SQLAlchemy** – ORM for seamless SQLite database interactions.
- **Flask-RESTful** – Simplifies RESTful API endpoint creation.
- **Flask-Security** – Robust authentication/authorization (replaces JWT for role-based access).
- **Flask-Caching** – Implements Redis-based caching for performance.
- **Flask-Excel** – Generates CSV/Excel exports for quiz data (user/admin dashboards).

Frontend

- **Vue.js** – Reactive UI with Vue CLI for modular development.
- **Bootstrap** – Responsive styling and pre-built components.

Database & Storage

- **SQLite** – Lightweight, file-based database for user/quizzes/scores.

Performance & Background Jobs

- **Redis** – Caching layer + message broker for Celery.
- **Celery** – Handles async tasks (daily reminders, monthly reports, CSV exports).

Testing & Utilities

- **MailHog** – Local email testing for reminders and reports.

Database Schema



The database schema is designed to support multi-role authentication (Admin/User), structured quiz management (Subjects → Chapters → Quizzes → Questions), and performance tracking.

Key tables include:

User: Stores email, password (hashed), and profile data (full name, qualification, DOB), linked to roles via `user_roles`.

Role: Defines permissions (e.g., "admin", "user") with Flask-Security integration.

Subject/Chapter: Hierarchical organization of study topics.

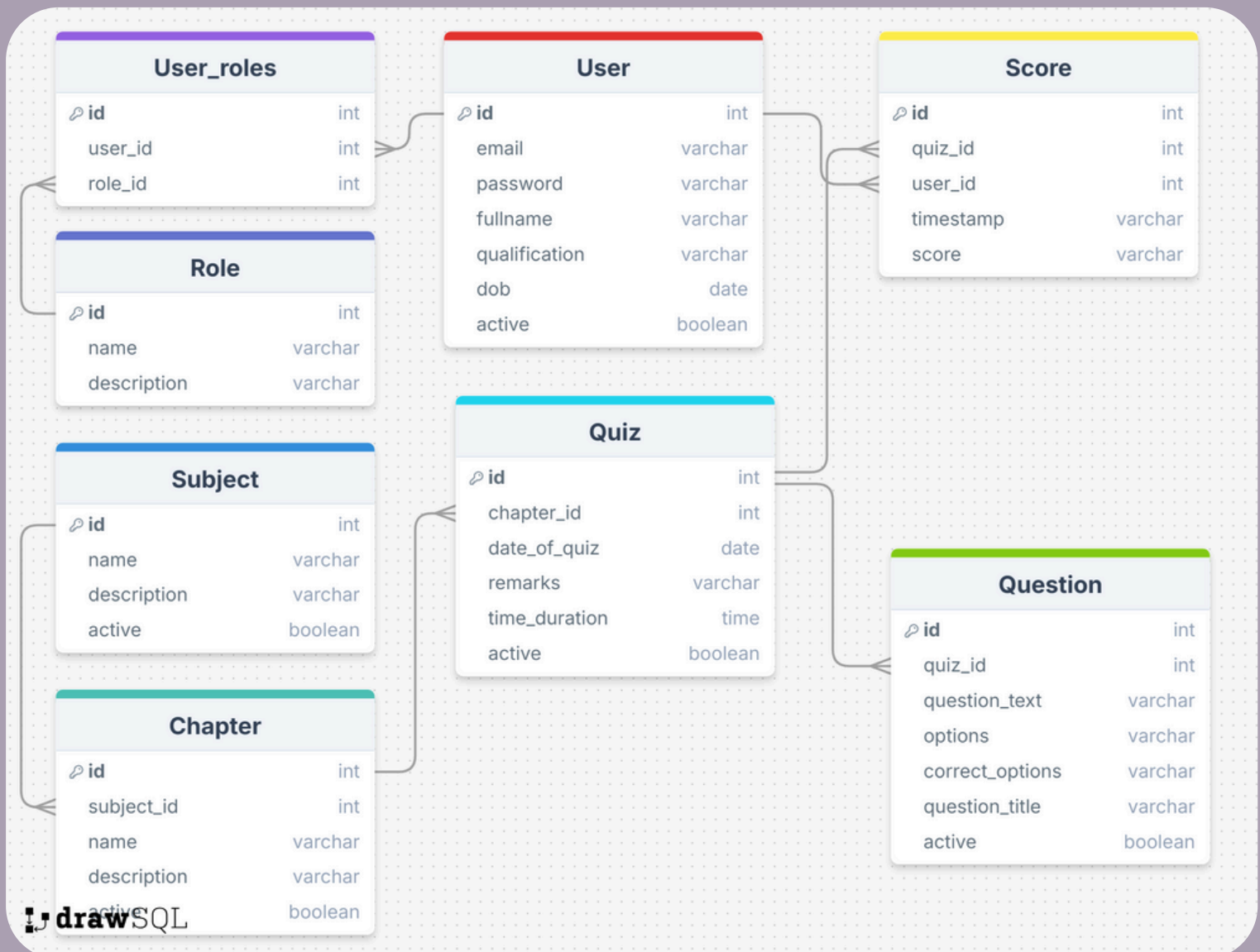
Quiz: Timed assessments with metadata (duration, remarks), tied to chapters.

Question: MCQs (single correct answer) stored with options as comma-separated strings.

Score: Tracks user attempts, timestamps, and scores for analytics.

Relationships ensure efficient querying (e.g., `user.scores`, `quiz.questions`), while active flags enable soft deletion.

Database ER Diagram



Directory Structure

```
QuizGen/
├── Quiz-generator/
│   ├── backend/
│   ├── api/
│   ├── app/
│   ├── tasks/
│   ├── CreateData.py
│   ├── frontend/
│   │   ├── assets/
│   │   ├── components/
│   │   ├── pages/
│   │   ├── utils/
│   │   ├── app.js
│   │   ├── daily.html
│   │   ├── index.html
│   │   └── monthly_report.html
│   ├── instance/
│   │   └── database.sqlite3
│   ├── ER_diagram.png
│   ├── local_setup.sh
│   ├── main.py           //EntryPoint
│   ├── README.md
│   ├── requirements.txt
│   ├── start_application.sh
│   ├── start_beat.sh
│   ├── start_mailing.sh
│   ├── start_redis.sh
│   └── start_worker.sh
└── ProjectReport.pdf
```

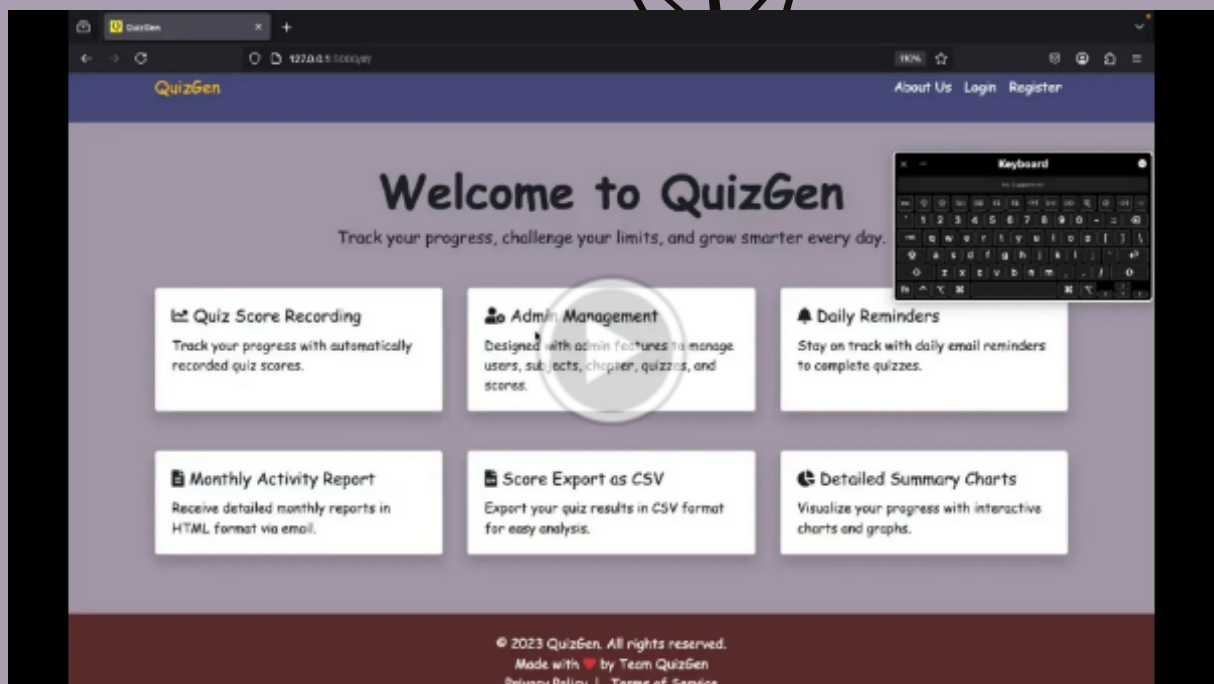


Demonstration Video



Link: [click me](https://drive.google.com/file/d/16PTD8y75gmB-HCmemrKdMZSep47nk59_/view?usp=sharing)

https://drive.google.com/file/d/16PTD8y75gmB-HCmemrKdMZSep47nk59_/view?usp=sharing



**Thank you
for your attention**