

## Author

**Name:** Vanya Tentiwala

**Email:** 23f3002427@ds.study.iitm.ac.in

## Description

The Quiz Master application is a comprehensive educational platform designed to facilitate the creation and administration of quizzes across various subjects and chapters. The system features distinct user roles (admin and student), subject and chapter organization, quiz management, performance tracking, and analytics. The application includes backend jobs for monthly reports, daily reminders, and data export functionality in various formats.

## Technologies Used

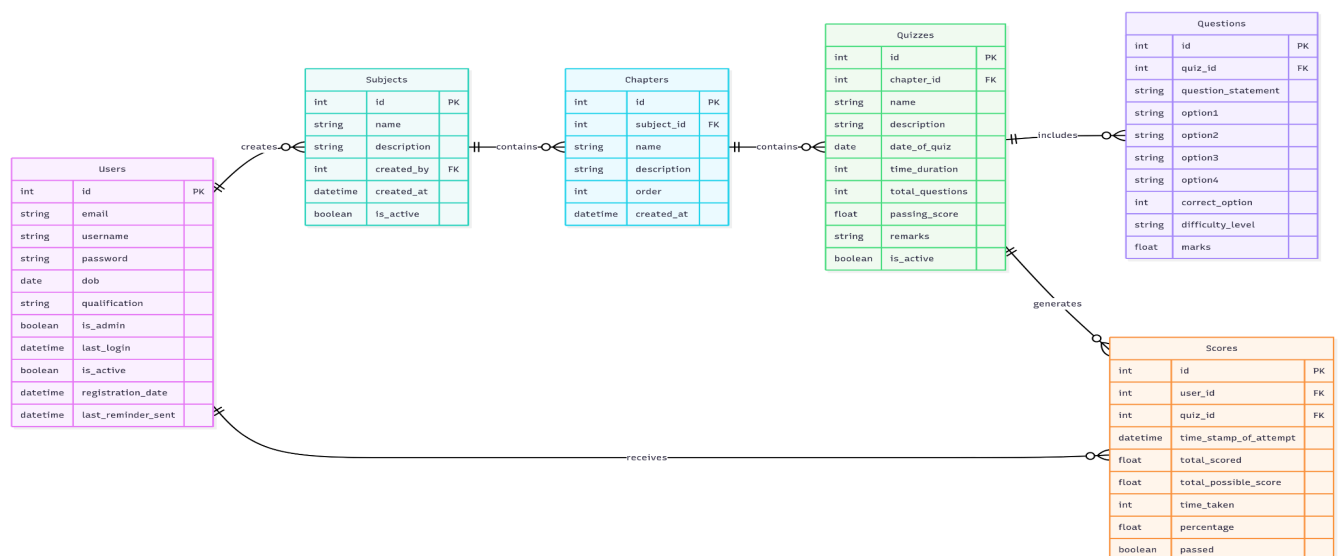
### Frontend:

- **Vue.js** - Progressive JavaScript framework for building user interfaces
- **Bootstrap** - CSS framework for responsive design
- **Axios** - Promise-based HTTP client for API requests
- **Chart.js** - JavaScript charting library for data visualization
- **CSV Export** - Client-side export capabilities

### Backend:

- **Flask** - Python web framework for server-side logic
- **Flask-SQLAlchemy** - ORM for database operations
- **SQLite** - Relational database
- **Redis** - Caching mechanism for improved performance
- **Flask-Bcrypt** - Password hashing for security
- **Flask-JWT-Extended** - Token-based authentication
- **Celery** - Task queue for background processing
- **Flask-Mail** - Email service integration

## DB Schema Design



## API Design: Quiz Master Application

### Implemented APIs:

- User Management: Registration, JWT authentication, RBAC for admins/users
- Subject/Quiz Management: CRUD for educational hierarchy (subjects → chapters → quizzes → questions)
- Quiz Taking: Attempt creation, submission, scoring, and history tracking
- Analytics: Dashboard statistics and performance metrics
- Export System: PDF/CSV generation for quiz results and reports
- Background Tasks: Scheduled reminders, reports, and data maintenance

### Architecture and Features

- MVC Structure: SQLAlchemy models, Vue.js views, Flask controllers
- Authentication: JWT tokens, RBAC, and bcrypt password hashing
- Admin Portal: Content creation and user management tools
- Student Interface: Quiz taking, performance review, and learning dashboards
- Notification System: Email alerts for reminders and new content
- Optimizations: Redis caching for frequently accessed data
- Task Scheduling: Celery-powered background processing

### Implemented Features

- User System: Secure authentication, role-based access, profiles
- Content Organization: Hierarchical subject/chapter/quiz structure
- Quiz Engine: Timed assessments with immediate feedback
- Analytics: Chart.js visualizations for performance tracking
- Communications: Email notifications and performance reports
- Data Export: Customizable exports in multiple formats

## Conclusion

Quiz Master delivers a comprehensive educational platform where administrators create structured content and students test knowledge through interactive quizzes. Its analytics system provides performance insights while notifications encourage consistent engagement. Built with Vue.js, Flask, and Redis, it combines an intuitive interface with robust backend capabilities.

## Video

 **Mad2 Project - Quiz Master**