

# Quiz Master

## Introduction

Quiz Master is a robust, single-page application (SPA) designed for exam preparation, supporting an Admin (superuser) and multiple Users. It enables course management, quiz creation, proctored assessments, and performance tracking. Built with a modern technology stack, it ensures scalability, security, and seamless local demo functionality.

## Technology Stack

- **Backend:** Flask (powers API endpoints), SQLite (data storage), Redis (caching for speed), Celery (schedules background tasks).
- **Frontend:** VueJS (dynamic SPA interface), VueJS Advanced with CLI (build tools), Bootstrap (responsive styling).
- **Templating:** Jinja2 (initial page load only).
- **Libraries:** ChartJS (visual charts), Flask-Mail (email notifications), Google Chat Webhooks (chat alerts), Stripe (payment processing), and Axios (API communication).

## System Architecture

- **Backend:** Flask handles request-response cycles, serving RESTful APIs. SQLite stores data with SQLAlchemy, Redis caches responses, and Celery schedules tasks like email reminders.
- **Frontend:** VueJS operates as a Single-Page Application (SPA), loading once and updating dynamically via Axios HTTP requests to the backend. Bootstrap ensures a responsive layout across devices.
- **Authentication:** JWT (JSON Web Tokens) secures endpoints, assigning tokens to distinguish Admin and User roles.
- **Request-Response:** Axios sends requests (e.g., GET for quiz data, POST for submissions) and processes JSON responses from Flask APIs.

## Roles and Functionalities

### Admin (Quiz Master)

- **Access:** Logs in with pre-existing superuser credentials; redirects to `admin_dashboard.vue`.
- **Functionalities:**
  - **Content Management:** Creates, updates, and deletes subjects, chapters, quizzes, and questions using CRUD operations. Attributes include name, description, time\_duration, and pay\_required.
  - **Dashboard:** `AdminSummary.vue` displays real-time stats (user count, quiz totals, revenue) and ChartJS visuals (e.g., performance trends, engagement metrics).
  - **Search and Pagination:** Filters content (e.g., by subject name) and paginates lists for easy navigation.
  - **Reports:** Generates and downloads CSV reports, including system summaries, subject analyses, revenue breakdowns, quiz popularity, and all quiz data.

### User

- **Access:** Registers with username (email), password, full\_name, etc.; logs in to access features.
- **Functionalities:**
  - **Quiz Participation:** Browses subjects/chapters, selects quizzes (proctored or regular), and starts attempts via Axios calls.
  - **Proctored Exams:** Includes timers, tab-switching detection (3-strike limit), and event logging in `QuizEventLog` to ensure fairness.
  - **Performance Tracking:** Views scores in `userscore.vue`, with options to sort (e.g., by date), search (e.g., by subject), paginate results, and download attempts or insights as CSV files.
  - **Dashboard:** Shows performance, mastery, and engagement through interactive charts.

## Core Features

- **SPA Design:** VueJS delivers a fast, single-page experience with no full page reloads.
- **Secure Login:** JWT tokens authenticate requests, ensuring role-based access.
- **Proctored Quizzes:** Monitors tab switches and logs actions for integrity.
- **Search and Pagination:** Simplifies finding and browsing content or results.
- **Responsive UI:** Bootstrap adapts the interface for phones and desktops.
- **Payments via Stripe:** Enables secure purchases of premium quizzes.
- **Performance Boost:** Redis caches API responses; Celery runs tasks like notifications asynchronously.

## Scheduled Tasks (Celery with Crontab)

- **Daily Emails:** Sent at 9 AM IST to remind Users of new quizzes (send\_daily\_reminder).
- **Monthly Reports:** Emails detailed quiz stats on the 1st of each month (send\_monthly\_report).
- **Google Chat Alerts:** Notifies about new or featured quizzes at 9 AM, 1 PM, and 5 PM IST via webhooks.
- **Reports (CSV) Downloads:** Admin exports all quiz data; Users export personal attempts.

## Database Schema (SQLite)

- **Users:** Stores user profile (e.g., email, full\_name) and links to user quiz attempts and payments. *Why?* Keeps User account and history intact.
- **Admins:** Holds superuser details (e.g., username, password). *Why?* Let the Admin manage the system.
- **Subjects:** Lists topics like “Physics.” *Why?* Organizes study areas.
- **Chapters:** Breaks subjects into sections like “Mechanics.” *Why?* Groups related quizzes.
- **Quizzes:** Saves quiz details (e.g., date\_of\_quiz, time\_duration). *Why?* The core of user practice.
- **Questions:** Stores questions and answers (e.g., question\_statement, correct\_option). *Why?* What user answer in quizzes.
- **QuizAttempts:** Tracks user quiz tries (e.g., total\_score, total\_time\_taken). *Why?* Shows user progress.
- **QuestionAttempts:** Records user answer choices (e.g., selected\_option). *Why?* Details user performance.
- **UserActivity:** Logs actions like logins (e.g., activity\_type). *Why?* Monitors usage.
- **QuizEventLogs:** Watches proctored quiz events (e.g., event\_type). *Why?* Ensures fair play.
- **QuizPayments:** Tracks payments (e.g., amount\_paid, payment\_status). *Why?* Manages quiz purchases.
- **QuizCarts:** Lists quizzes users’ve selected to buy (e.g., quiz\_id). *Why?* Acts as a shopping cart.

## Extra Features

- **Charts:** Visualizes progress with ChartJS.
- **Notifications:** Emails (payments, results) via Flask-Mail; Google Chat for quiz updates.
- **Input Validation:** Checks data on both frontend and backend.

## Conclusion

Quiz Master is a professional SPA with secure authentication, proctored quizzes, and optimized performance. It’s a complete solution for exam prep and administration.

**Demo Video Link :** [Google drive Link](#)