

Project Report: Quiz Master

Author

Name: Chetankumar Tak

Roll Number: 23F3003467

Student Email 23f3003467@ds.study.iitm.ac.in

I'm a passionate learner currently pursuing the BS in Data Science from IIT Madras, with a keen interest in backend development and scalable applications.

Description

This project is a role-based quiz platform where admins can create quizzes and users can attempt them.

The system tracks scores, quiz performance, and supports CSV report downloads and monthly email summaries via Celery and Flask.

Technologies Used

- Flask: Main web framework
- Flask-SQLAlchemy: ORM for database modeling
- Flask-Security: Role-based authentication system
- Flask-Caching: For efficient caching (e.g., dashboard)
- Flask-Excel: For generating downloadable CSV reports
- Celery + Redis: For background tasks and scheduled reports
- Flask-RESTful: To expose REST APIs for frontend integration
- Bootstrap 5: For frontend styling (via Vue + HTML)
- Vue.js: For dynamic frontend components

Purpose: Each technology was chosen for simplicity, extensibility, and integration.

Celery enables background/scheduled tasks (like reports), Flask-Security simplifies auth, and Vue helps with dynamic dashboard behavior.

DB Schema Design

User Table: id, email, password, username, qualification, dob, created_at, active, fs_uniquifier

Relationships: scores, attempted questions, roles

Role-Based Access: roles_users, Role, User with user.roles

Quiz Structure: Subject Chapter Quiz Question

Constraints: cascade deletion, foreign key relationships

Score table links users to quizzes with performance data

Attempted table logs user responses

Design Rationale: This normalized structure ensures modularity and logical grouping (subject chapter quiz).

Score and Attempted tables allow tracking user performance at a granular level.

API Design

Implemented using Flask-RESTful and exposed under /api:

- GET /api/summary/admin Admin dashboard summary
- GET /api/summary/user Authenticated user quiz summary
- GET /create-csv Starts CSV generation (via Celery)
- GET /get-csv/<task_id> Checks and returns generated CSV
- POST /login, POST /register, GET /logout Auth & session management

Note: A separate YAML file documents all API endpoints and responses (as required). Architecture and Features

Organization:

- models.py: Database schema
- resources.py: API resources (Flask-RESTful style)
- routes.py: Custom Flask routes (CSV endpoints, etc.)
- app.py: App factory with init, role/user creation
- celery/tasks.py: All Celery-based background jobs
- templates/: HTML templates for email reports
- static/: Assets like images (e.g., company logo/stamp)

Features:

- Admin and User login via Flask-Security
- Role-based dashboard (admin sees all, user sees personal data)
- Real-time dashboard using Vue
- Export quiz data and performance as CSV
- Celery Beat scheduled task for monthly user score report
- Dynamic charts using Chart.js

Video

Link: https://drive.google.com/file/d/1xykpnlys23bJJdLUa_W9upXhvo4V_UzU/view?usp=sharing