Project Report

Student Details

- Name: Jashan Tiwari
- Roll Number: 23f300922
- Course: Modern Application Development II
- Branch: BS in Data Science and Application
- College: IIT Madras

Project Title

Quiz Master V2: Modern Application Development II

Problem Statement

It is a multi-user app (one requires an administrator and other users) that acts as an exam preparation site for multiple courses.

Approach

The platform is built as a modern web-based application with these key functionalities:

- Role-based Access Control:
  Admin and User roles with permissions via JWT authentication.
- Quiz Management:
  Admins can create, update, and delete subjects, chapters, quizzes, and questions with hierarchical organization.
- User Dashboard:
  Users can browse quizzes, attempt them with real-time timers, and view their performance history.
- Advanced Analytics:
  Comprehensive scoring, performance tracking, and visual analytics (Chart.js).
- Background Jobs and Automation:
  Daily reminders for inactive users, monthly activity reports, and CSV exports via Celery & Redis.
- Performance Optimization:
  Redis caching to improve response times.

- Modern UI/UX:
  Responsive frontend using Vue.js 3 and Bootstrap.

Frameworks and Libraries Used

Database

- `SQLite`: Lightweight data storage
- `SQLAlchemy`: ORM (Object Relational Mapper)
- `Flask-Migrate`: For schema migrations

Backend

- `Flask`: REST API framework
- `Flask-RESTful`: Structured API
- `Flask-SQLAlchemy`: ORM integration
- `Flask-JWT-Extended`: Secure JWT authentication
- `Flask-CORS`: Cross-origin resource sharing

Frontend/UI

- `Vue.js 3`: Reactive frontend framework
- `Vue Router`: Routing
- `Vuex`: State management
- `Chart.js` & `Vue-ChartJS`: Analytics/Charts
- `Bootstrap 5`: Styling & responsiveness
- `FontAwesome`: Icons

Background Jobs & Caching

- `Redis`: Caching & message brokering
- `Celery`: Background/scheduled tasks
- `Flask-Caching`: App-level caching

Additional Libraries

- `Axios`: HTTP requests (Frontend)
- `JWT-Decode`: Token handling (Frontend)
- `Werkzeug`: Password hashing/security

Database Schema (ER Diagram Overview)

Entities:

- Users:
  Fields: id, username, email, password, full_name, qualification, dob, role
- Subjects:
  Fields: id, name, description
- Chapters:
  Fields: id, name, description, subject_id
- Quizzes:
  Fields: id, name, chapter_id, date_of_quiz, time_duration, remarks
- Questions:
  Fields: id, quiz_id, question_statement, option1-4, correct_option
- Scores:
  Fields: id, quiz_id, user_id, time_stamp_of_attempt, total_scored, reattempted

Relationships:

- `Subject → Chapters (One-to-Many)`
- `Chapter → Quizzes (One-to-Many)`
- `Quiz → Questions (One-to-Many)`
- `Quiz → Scores (One-to-Many)`
- `User → Scores (One-to-Many)`

API Resource Endpoints

1. Authentication APIs

- `POST /api/signup` – Register user
- `POST /api/login` – Login for admin/user

2. Admin Management APIs

- `GET/POST/PUT/DELETE /api/subject` – Manage subjects
- `GET/POST/PUT/DELETE /api/chapter` – Manage chapters
- `GET/POST/PUT/DELETE /api/quiz` – Manage quizzes
- `GET/POST/PUT/DELETE /api/question` – Manage questions
- `GET /api/users` – List users for admin

3. Quiz Management APIs

- `GET /api/quizzes` – List available quizzes
- `POST /api/score` – Submit quiz attempt
- `GET /api/user-scores` – User's quiz history

4. Analytics & Reporting APIs

- `GET /api/quiz-stats` – Quiz statistics
- `GET /api/user-summary` – User analytics
- `POST /api/export_users_csv` – Export user CSV (async)
- `GET /api/csv_result/<task_id>` – Download CSV

6. Background Jobs

- Daily reminders for inactive users
- Monthly activity reports
- Asynchronous CSV export

Contributors

Jashan Tiwari — Full Stack Developer
Contact — 23f30009222@ds.study.iitm.ac.in | +91 7814536156

The complete demonstration video for the project is available here:
link for video