

Quiz Master - V2 Project Report

Chirantan Dey
Roll Number: 23f2001382
JANUARY 2025 TERM

1 Student Details

- Name: Chirantan Dey
- Roll Number: 23f2001382
- Term: JANUARY 2025 TERM
- Subject: Modern Application Development 2 - Project

2 Project Details

2.1 Question Statement

Quiz Master - V2 is a multi-user application that acts as an exam preparation site for multiple courses. The application requires an administrator role and supports multiple users.

2.2 Problem Statement Approach

The application was implemented as a comprehensive learning management system with the following key components and approaches:

2.2.1 Architecture

- **Backend Architecture**
 - Implemented a RESTful API using Flask-RESTful for modular and scalable endpoints
 - Used Flask-Security for robust user authentication and role-based access control
 - Incorporated SQLAlchemy ORM for efficient database operations and relationship management
 - Implemented Redis caching to improve performance of frequently accessed data
- **Frontend Design**
 - Utilized Vuex for centralized state management of user data and quiz states
 - Implemented Vue Router for client-side navigation with protected routes
 - Used Bootstrap for responsive design and consistent UI components
- **Database Design**
 - Designed a normalized database schema with proper relationships
 - Implemented subject-chapter-quiz hierarchy for organized content management
 - Created separate tables for questions and scores with appropriate foreign key constraints
 - Used SQLite for easy deployment and maintenance

2.2.2 Key Features Implementation

- **User Management**
 - Role-based access control (Admin/User roles)
 - Secure password hashing and token-based authentication
 - User profile management with qualifications and personal details
- **Quiz System**
 - Hierarchical organization (Subjects → Chapters → Quizzes)
 - Timed quiz attempts with automatic submission
 - Real-time score calculation and result storage
 - Multiple choice question format with immediate feedback
- **Administrative Features**
 - Complete CRUD operations for subjects, chapters, and quizzes
 - User management and approval system
 - Performance analytics and reporting dashboard
- **Background Processing**
 - Celery workers for handling asynchronous tasks
 - Automated daily email reminders for pending quizzes
 - Monthly performance report generation and distribution
 - Redis-based task queue management

2.2.3 Performance Optimization

- Implemented Redis caching for frequently accessed data
- Used database indexing for faster query performance
- Implemented lazy loading of relationships in SQLAlchemy
- Client-side caching of static assets

3 Frameworks and Libraries Used

- **Database Layer**
 - SQLite - Primary database
 - Flask-SQLAlchemy - ORM
 - Redis - Caching layer
- **Backend Framework**
 - Flask - Web framework
 - Flask-Security - Authentication
 - Flask-RESTful - API development
 - Celery - Background tasks
- **Frontend Framework**
 - VueJS - JavaScript framework
 - Vuex - State management
 - Bootstrap - UI components
 - Vue Router - Client routing

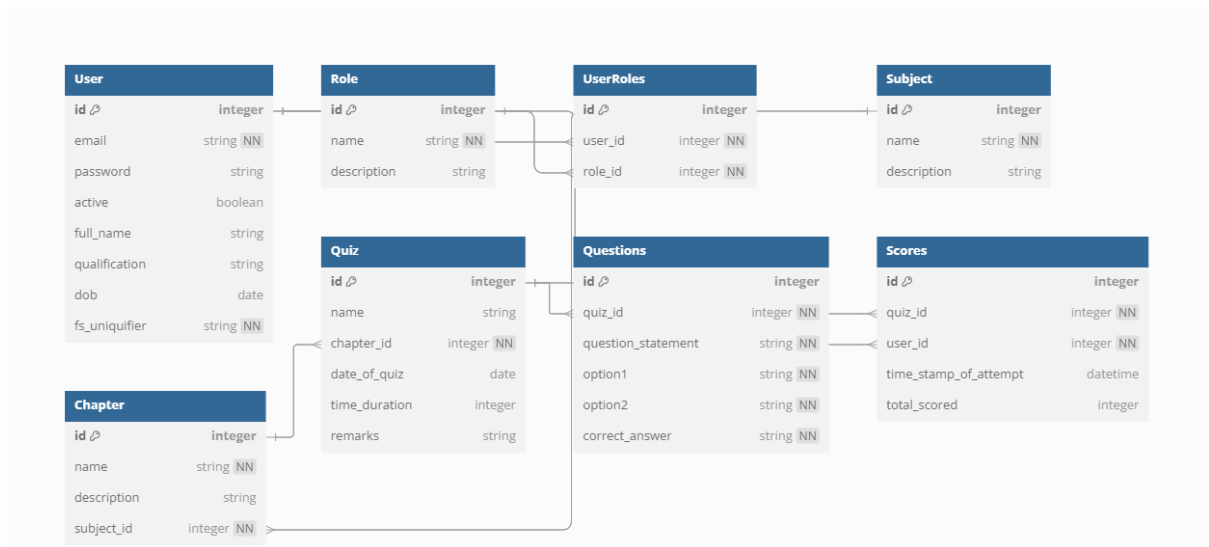


Figure 1: Entity Relationship Diagram

4 Database Design (ER Diagram)

5 API Documentation

Key endpoints include:

5.1 Authentication Endpoints

- POST /register - User registration
 - Accepts: email, password, full_name, qualification, dob
 - Returns: access_token, user details
- POST / - User login
 - Accepts: email, password
 - Returns: access_token, user details

5.2 Quiz Management Endpoints

- GET /api/quizzes - List all quizzes
- POST /api/quizzes - Create new quiz
- GET /api/questions - Get quiz questions
- POST /api/questions - Create quiz question

5.3 Score Management Endpoints

- GET /api/scores - Get user scores
- POST /api/scores - Submit quiz attempt

5.4 Content Management Endpoints

- GET /api/subjects - List all subjects
- POST /api/subjects - Create subject
- GET /api/chapters - List all chapters
- POST /api/chapters - Create chapter

5.5 Analytics Endpoints

- GET /api/charts/admin - Admin dashboard data
- GET /api/charts/user - User performance data

Video Link:-