Project Report

Project Title

Quiz Management System

Student's details :

Name: Hafiza Noor

Roll Number: 23f2004875

Email: 23f2004875@ds.study.iitm.ac.in

> I am a passionate backend developer who enjoys building scalable Flask applications and exploring full-stack development. Beyond coding, I find joy in reading, walking, and embracing nature. I strive to balance technical growth with mindfulness, and I'm driven by curiosity—both in solving real-world problems through code and appreciating the world offline.

Problem Statement

The goal of this project is to develop a scalable and modular web-based Quiz Management System that allows:

- **Admins** to manage subjects, chapters, quizzes, and questions
- **Users** to register, attempt quizzes, and view scores
- Efficient architecture that supports evaluation and performance analytics

---

Project Description

This system enables users to register and take quizzes based on categorized subjects and chapters. Users can track their scores and review their quiz performance. Admins can create and manage subjects, chapters, questions, and quiz configurations.

Technologies and Frameworks Used

Technology	Purpose
Python, Flask	Backend web framework
SQLAlchemy	ORM for database schema and queries
Flask-JWT-Extended	Secure token-based authentication
Redis + Flask-Caching	Cache admin dashboard data
PostgreSQL/MySQL	Relational database
Vue.js	Frontend UI framework
Werkzeug	Image upload and security

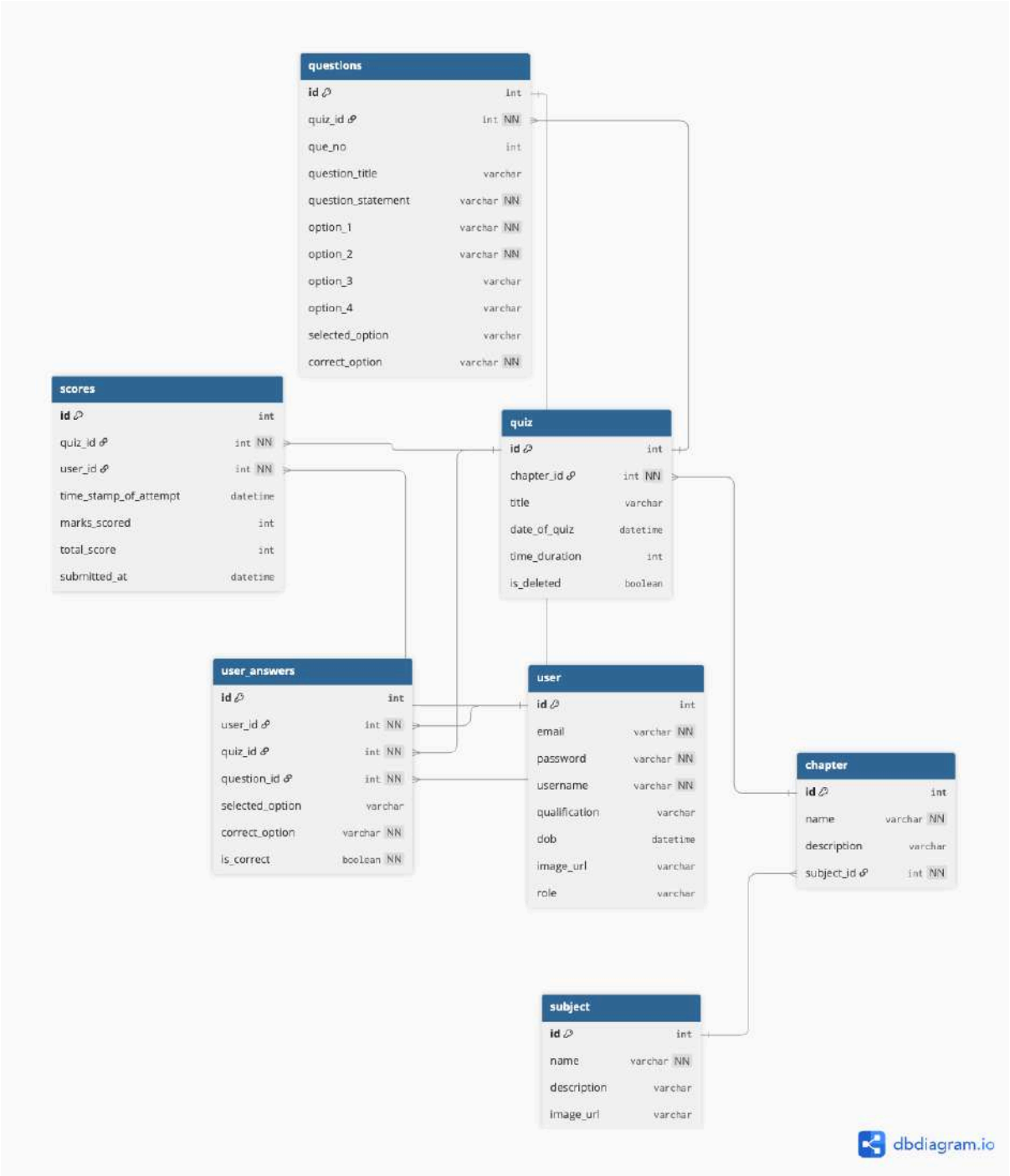
Approach

- Built with a **Flask RESTful API** backend and a **Vue.js** frontend
- Data schema designed using SQLAlchemy with normalized relationships
- JWT handles authentication securely
- Redis provides caching for faster admin dashboard access
- Asynchronous jobs for user engagement (reminders, activity reports)
- Modular folder structure separates user/admin routes

Database Schema (ER Diagram)

Tables:

- `User`: email, password, username, dob, qualification, image_url, role
- `Subject`: name, description, image_url
- `Chapter`: name, description, subject_id (FK)
- `Quiz`: title, date_of_quiz, time_duration, is_deleted, chapter_id (FK)
- `Questions`: quiz_id (FK), options, correct/selected answer
- `Scores`: quiz_id (FK), user_id (FK), submitted_at, marks
- `UserAnswers`: user_id, quiz_id, question_id, selected_option, correct_option, is_correct



Architecture Overview

```
project/
│
├── app.py                          # App entry point
├── backend/
│   ├── models.py                   # SQLAlchemy DB models
│   ├── routes/
│   │   ├── admin/                  # Admin-specific APIs
│   │   └── user/                   # User-specific APIs
│   └── static/
│       ├── subjects/               # Subject images
│       └── profiles/               # Profile images
├── frontend/
│   ├── src/
│   │   ├── components/
│   │   │   ├── admin_components/   # Vue files for admin dashboard
│   │   │   └── user_components/    # Vue files for user dashboard
```

Features Implemented

- User registration and login with JWT
- Role-based access control (admin/user)
- Admin panel for managing subjects, chapters, quizzes, and questions
- Quiz participation with auto-evaluation
- Scoreboard and analytics dashboard
- Caching for high-read routes
- Default profile and subject images
- Async jobs: monthly reports and daily quiz reminders

Presentation Video

🎥 Watch Project Demo