Author

Name : Gurudu Rohith Reddy
Roll No : 22f1000824
Email : 22f1000824@ds.study.iitm.ac.in

About Me

Currently I am pursuing diploma in programming and data science from IITM . I look forward to getting the Bsc degree from IITM . I give my regards to IITM for making me an IITian .

Description

The **Household Services Application - V2** is a multi-user platform designed to streamline home service requests by connecting customers with verified service professionals while providing administrators with full control over operations. Built with Flask, SQLite, Redis, and Celery for efficient backend processing, and Vue.js with Bootstrap for a responsive frontend, the platform ensures seamless service management. Key features include role-based authentication, service request handling, real-time notifications, and automated reporting. Admins oversee users and services, professionals manage job requests, and customers can book and review services, ensuring a smooth and trustworthy service experience.

Technologies Used

The **Household Services Application - V2** is built using a combination of frontend and backend technologies to ensure efficient performance and scalability.

- **Frontend:** Vue.js (HTML, CSS, JavaScript) – Used for building interactive and responsive web pages.

- **Backend:** Flask – A lightweight Python framework for handling API requests and backend logic.

- **Database:** Flask-SQLAlchemy – Facilitates communication between Python and the SQLite database. DB Browser for SQLite is used for creating and managing the database.

- **Caching & Task Queue:** Redis – Used for API caching and as a message queue for handling asynchronous jobs through Celery.

- **Email Service:** Flask-Mail – Handles email notifications and service-related communication.

- **Development Tools:** Visual Studio Code – The primary development environment for writing and testing code.

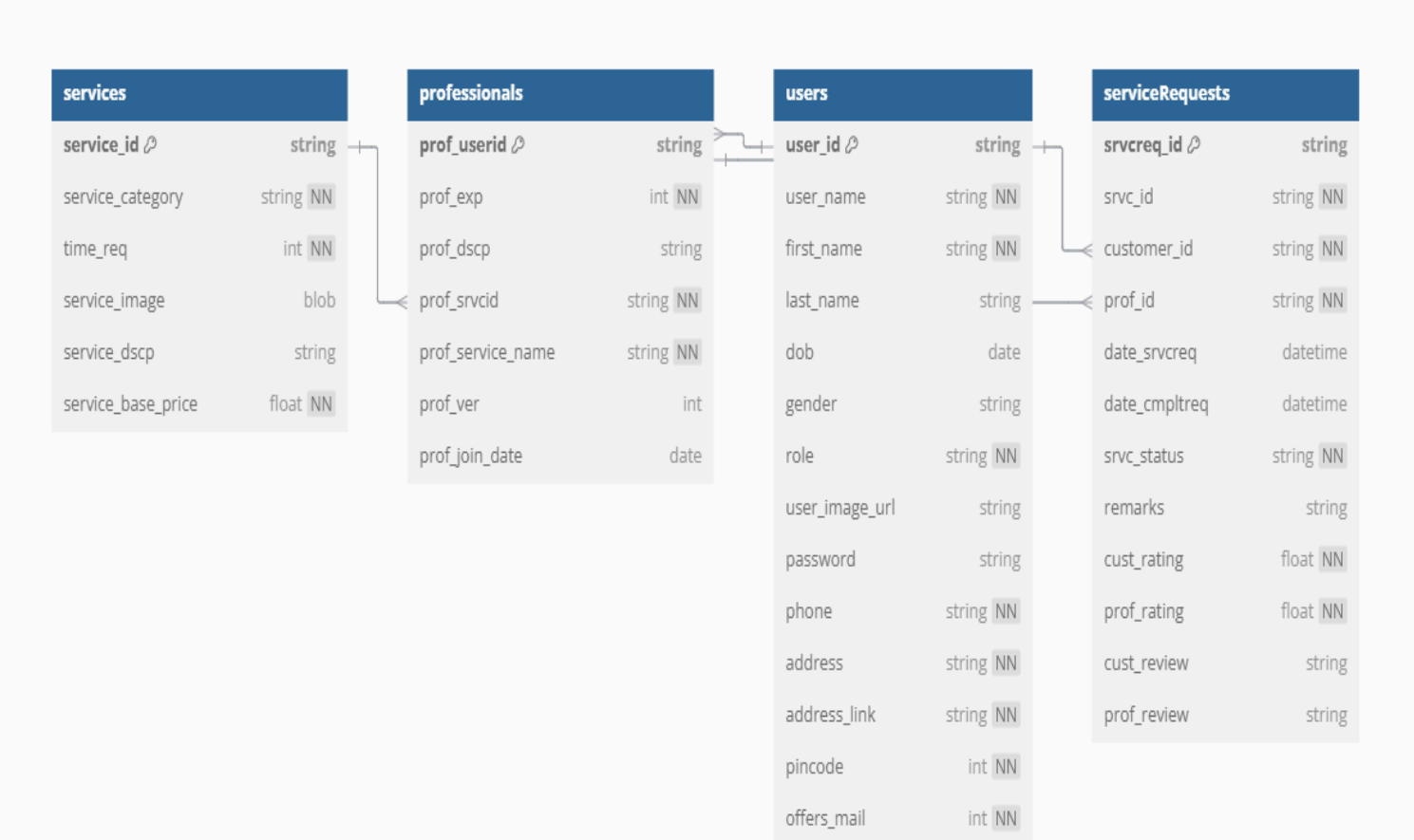
These technologies were chosen to create a robust and scalable web application that efficiently manages service requests, user interactions, and background tasks.

DB Schema Design :

Database contains 4 tables :

1) users
2) professionals
3) services
4) serviceRequests

Entity Relationship Diagram



Features :

1) **Role-Based Access Control (RBAC) & Secure Authentication**

- The platform ensures secure login and registration using JWT-based authentication.
- Different user roles (Admin, Service Professional, Customer) have specific access permissions, ensuring data security and controlled operations.

2) **Service Management & Requests Handling**

- Admins can create, update, and manage services (e.g., plumbing, electrical, cleaning).
- Customers can browse services, place requests.

- Service professionals can accept/reject service requests and complete assigned jobs.

3) **User & Service Professional Management**

- Admins oversee all users and approve or block service professionals based on performance or fraud detection.
- Service professionals receive monthly reports on completed services and can rate customers based on experience.
- Customers can review and rate service professionals, ensuring quality control.

4) **Automated Reports & Notifications** 📧

- The system sends automated reports (via email) to service professionals about their completed jobs.
- Admins can track pending, completed, and rejected service requests for better platform management.

5) **Performance Optimization with Caching & Asynchronous Tasks**

- Redis caching improves response times by storing frequently accessed data.
- Celery with Redis handles background tasks efficiently, like sending reports & notifications without slowing down the system

Video Link

Click on the drive link below :

MAD2-project_demo_video…