Author

Nadendla Venkata Sanjana - 23f1001240
23f1001240@ds.study.iitm.ac.in

3rd Year B.Tech Student of VNR Vignana Jyothi Institute of Engineering and Technology pursuing Computer Science in Artificial Intelligence and Machine Learning

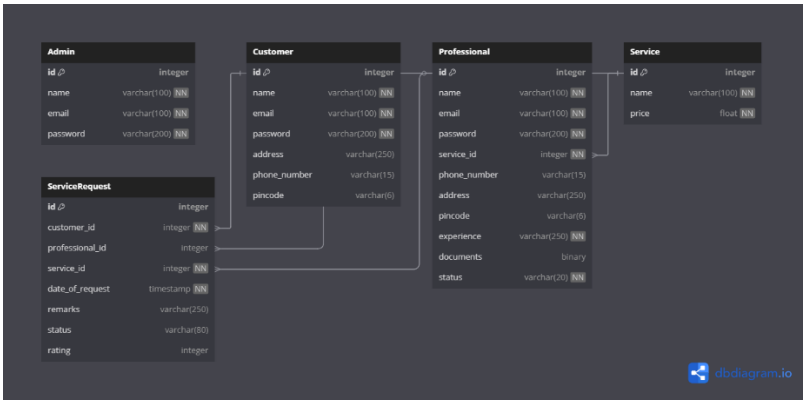
Description

Homify is a multi-user platform that connects customers with service professionals for home maintenance tasks such as Plumbing and AC servicing. Built with Flask, SQLite, Vue.js, Redis and Celery, Homify features role-based access control (admin, service professionals, customers), seamless service request management with automated backend jobs like reminders and reports. Homify ensures efficient service handling, secure authentication and a responsive UI.

Technologies used

Flask Framework: Provides a lightweight backend framework to handle routing and APIs.
SQLite3: Manages the database for CRUD operations on customers, providers, services, and service requests.
JWT: Manages secure token-based authentication for admins, customers, and providers.
Vue-JS & Chart JS: VueJS for the frontend & Chart JS for visualisation
Datetime Module: Manages and manipulates date and time values in service requests.

DB Schema Design



The Homify database is designed to manage a multi-user platform connecting customers with service professionals. It includes tables for **Admin**, **Customer**, **Professional**, **Service**, and **ServiceRequest** ensuring structured data storage. Admins oversee users and services, Customers create service requests and Professionals accept assignments based on their expertise.

API Design

The API follows REST principles, providing structured endpoints for user authentication, service management and service requests. It uses JWT authentication, with responses formatted as structured JSON along with appropriate HTTP status codes for seamless integration. The system supports role-based access control (RBAC), ensuring that admins, customers and service professionals can only access authorized functionalities. Additionally, background jobs are handled using Celery and Redis, enabling automated reminders, monthly reports and service request exports for improved efficiency.

Architecture and Features

Core Functionality

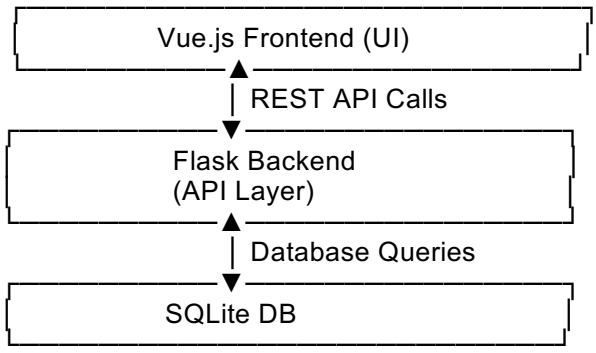
- JWT-based authentication ensures secure login/logout and route protection.
- Service search filters by name only (case-insensitive) for simplicity.
- Mock suggested services display popular options with ratings (no backend dependency).
- Booking system includes date selection and special instructions.

User Experience

- Responsive design works on all devices (Bootstrap's grid system).
- Modal-based booking avoids page reloads for smoother interactions.
- Clear error messages guide users during failed API calls (e.g., expired tokens).

Performance & Security

- Token refresh automatically handles session expiry without manual login.
- Minimal API payloads reduce bandwidth usage (e.g., no redundant price filters).



Drive link of Video presentation:

https://drive.google.com/file/d/17BWxgFR8nP1ypUbMltNyg3CHIr5t_n00/view?usp=sharing