Project Report: Household Services Application

Student Details

- Name: Aditya Vaidhya
- Roll No.: 23f2000809
- Student Mail: 23f2000809@ds.study.iitm.ac.in

Project Details

Question Statement

Household Services Application: It is a multi-user app (requires one admin and other service professionals/customers) which acts as a platform for providing comprehensive home servicing and solutions.

Approach to the Problem Statement

The Household Services Application was designed to facilitate the interaction between service providers and customers. The application allows customers to book various household services, while service professionals can manage their service requests. The admin has the ability to manage users and services, ensuring smooth operation of the platform.The project was approached by first defining the core functionalities required for each type of user (admin, service professional, and customer). The database schema was then designed to support these functionalities, followed by the implementation of the backend using Flask. The frontend was created using HTML templates to provide a user-friendly interface.

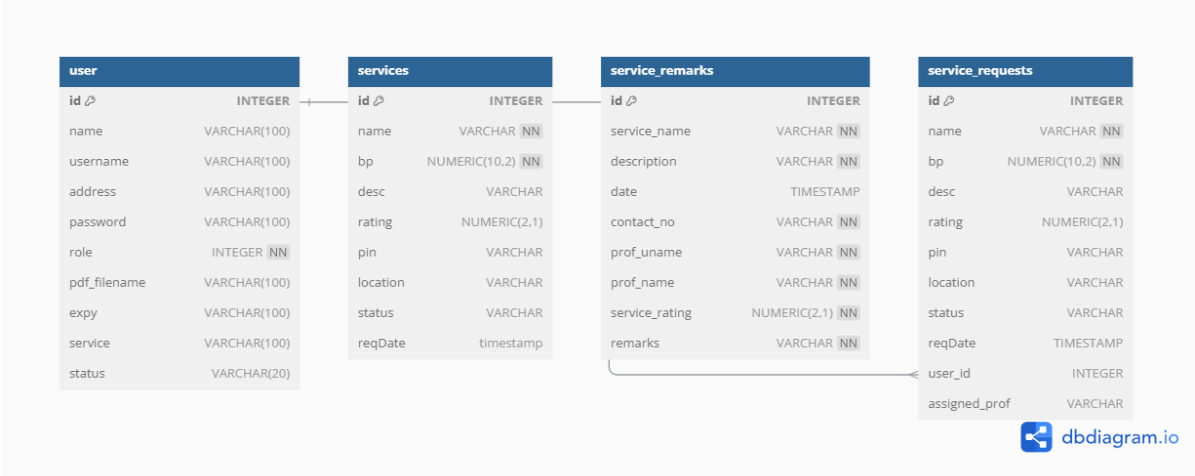
Frameworks and Libraries Used

- Flask: A micro web framework used for building the backend of the application.
- Flask-Login: Used for managing user sessions and authentication.
- Flask-SQLAlchemy: An ORM (Object Relational Mapper) used for database interactions.
- SQLite: A lightweight database used for storing application data.
- Jinja2: A templating engine used for rendering HTML templates.
- Werkzeug: A comprehensive WSGI web application library used for handling file uploads and security.

User Endpoints

- GET /: Renders the base homepage.
- GET /user_home: Renders the user home page with service requests and available services.
- POST /user_home: Handles actions on service requests (close or cancel).
- GET /find_service: Renders the service search page.
- POST /find_service: Handles service booking requests.

ER Diagram



Admin Endpoints

- GET /admin_home: Renders the admin home page with service requests and professionals.
- GET /admin/manage_cat: Renders the manage categories page.
- POST /admin/manage_cat: Handles creation, editing, and deletion of services.
- POST /admin_manage_profs: Handles approval or rejection of professional profiles.
- POST /admin/accept_prof: API endpoint to accept a professional.
- POST /admin/reject_prof: API endpoint to reject a professional.

Professional Endpoints

- GET /professional_home: Renders the professional home page with service requests.
- POST /professional_home: Handles actions on service requests (accept, reject, close).

Miscellaneous Endpoints

- GET /user_profile: Renders the user profile page.
- POST /user_profile: Handles updates to user profile information.
- GET /service_remarks: Renders the service remarks page.
- POST /service_remarks: Handles submission of service remarks.
- GET /user-chart: Generates and displays a chart of users by roles.

Video Presentation : MAD - 1 Project _ Video Presentation.mp4