“House to Home” Household Services Web Application

A project report for the MAD-II

Submitted by

Name: SUJAY D

Roll number: 22F2001590

IITM Online BS Degree Program,

Indian Institute of Technology, Madras, Chennai

Tamil Nadu, India, 600036

Contents

I recalled a poem which I leant in class 6th in my childhood and realized that it suits the Web App's purpose in making a House to Home.

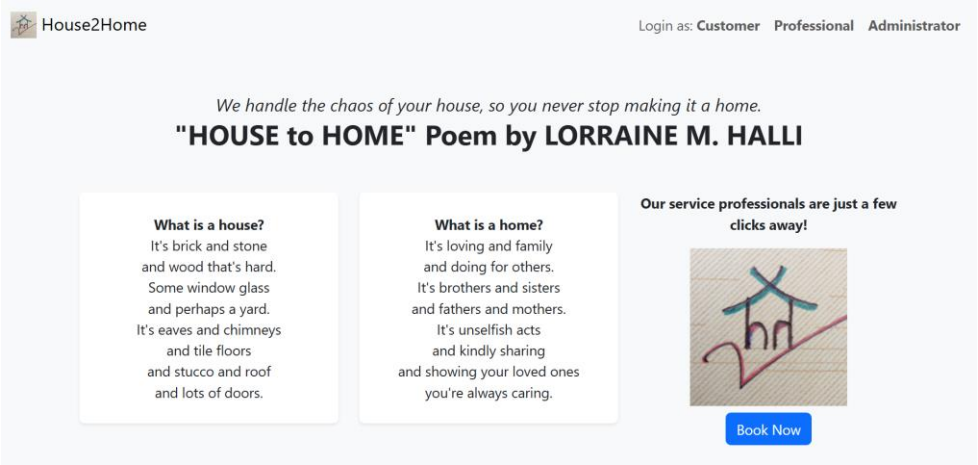

Figure 1: Screenshot of Index Page of the Household Service Web App.

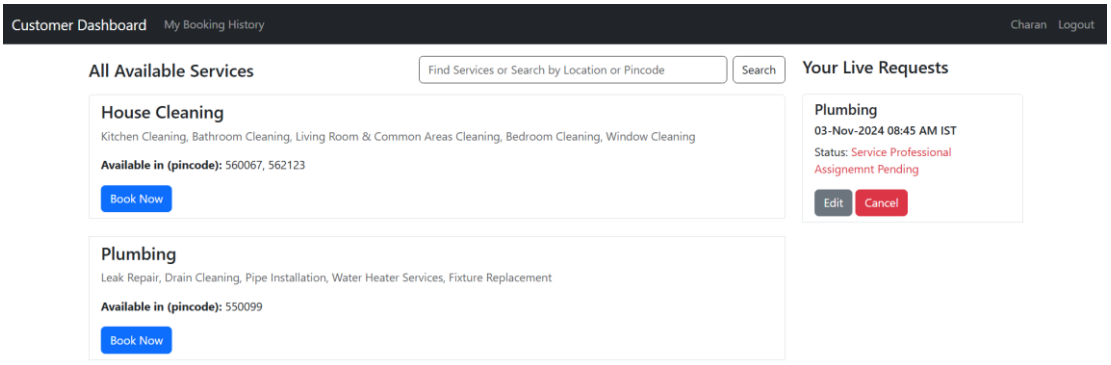

Figure 2: Screenshot of Customer Dashboard where a service can be booked.

1 Student Details

My name is Sujay D. I am a student of Data Science and Application program of IIT Madras. My roll no is 22f2001590. My student mail id is 22f2001590@ds.study.iitm.ac.in

I am passionate about Artificial Intelligence, Machine Learning and Coding in general. Currently I am learning HTML, CSS and JS for making Web Apps and for backend Flask.

2 Project Description

The app features a multi-tier login/register system with dedicated forms for customers, service professionals, and admins. Each user logs into their respective dashboard based on their role.

Admins manage customers and service professionals, approving profiles after document verification and blocking users for fraudulent activity or poor reviews. They also oversee service management, handling service creation, updates, and deletions.

Customers can request services, modify existing requests, and partially close them upon completion by paying the service fee. A search function allows them to filter services by location, name, or pin code.

Service professionals can view customer requests, accept or reject them, and mark them as completed. Admins can also search for professionals to manage their status.

The backend automates daily reminders, sends monthly activity reports via email, and allows users to export service history as CSV files asynchronously

3 Technologies Used

Flask for application code, Vue3 + Bootstrap for UI and styling, SQLite for data storage, Matplotlib for graph creation

4 Architecture and Features

The project follows a structured organization with two main directories: backend and frontend. The backend contains celery for asynchronous task handling, controllers for application logic, and models for database interactions. The frontend manages application pages in src/views, while routing is handled in src/router.

At the root level, essential files include a PDF document, app.py (the backend entry point), an SQLite database, and requirements.txt for dependencies. Additionally, caching is implemented for efficient user management, improving performance and responsiveness.

5 Database Schema Design and Terminologies

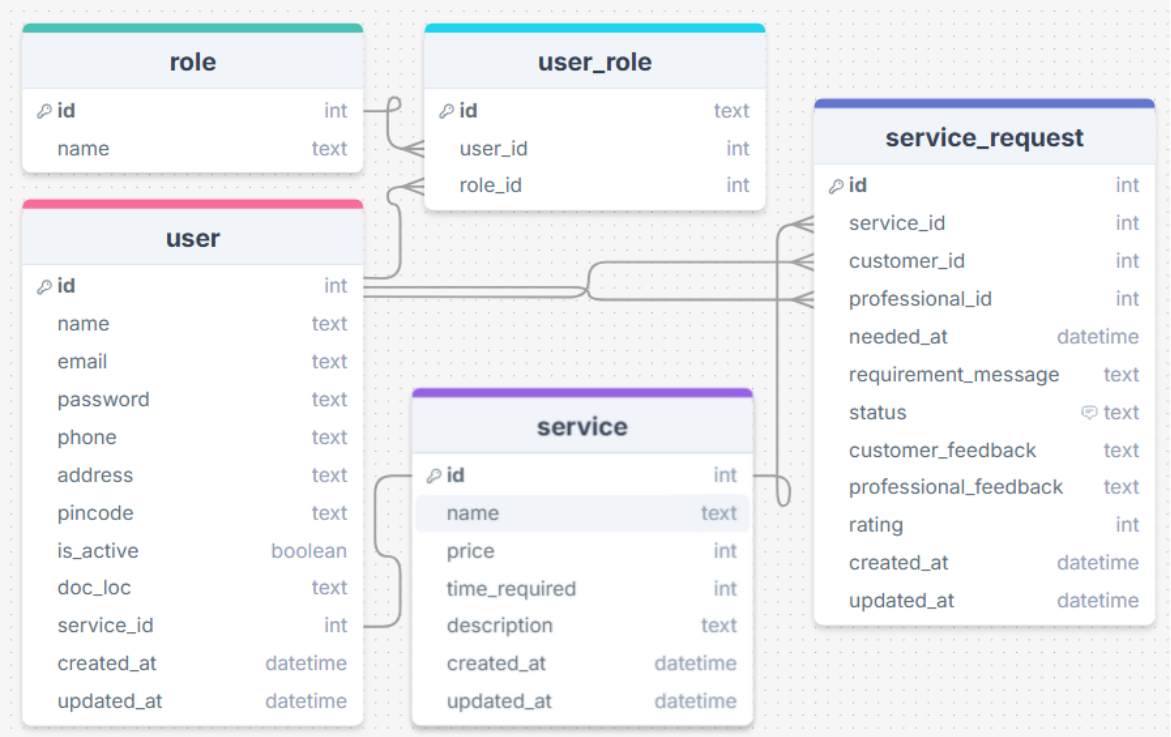

Figure 3: ER Diagram of Database Schema designed in https://drawsql.app/

6 API Design

1. **Login and Logout Controllers**
   /customer_signup, /professional_signup, /customer_login, /professional_login, /admin_login, /logout

2. **Admin Controllers**
   /admin_dashboard, /users, /services with CRUD endpoints, /app_summary

3. **Customer Controllers**
   /customer_dashboard, /book_now, /edit_request, /cancel_request, /pay_request,

4. **Service Professional Controllers**
   /professional_dashboard, /accept_request, /reject_request, /close_request

7 Web App Walkthrough Demo Video

https://drive.google.com/file/d/15OxHin87Hy0pbrIqAz9XtFQCSdSOID44/view