

## Project Report: Household Services Application

### Student Details

- **Name:** Jay Rajesh Tolia
- **Roll Number:** 24dp2000021
- **Email ID:** 24dp2000021@ds.study.iitm.ac.in

### Project Details

#### Project Title: Household Services Application

#### Project Statement:

The Household Services Application is a multi-user platform designed to provide comprehensive home servicing solutions. The application facilitates the interaction between customers, service professionals, and an admin. The application enables customers to request services, view service professionals, and post reviews after service completion. Service professionals can accept or reject service requests based on their availability. Admin users can manage service professionals and customer data, approve or block service professionals, and manage the services offered.

The application consists of three primary roles:

1. **Admin:** Manages users, services, and approval of service professionals.
2. **Service Professional:** Provides services to customers and manages service requests.
3. **Customer:** Requests services, views professionals, and posts reviews.

#### Approach:

To approach the project, I focused on dividing the development into the following phases:

1. **Setup:** The project was built using Flask, with SQLite as the database to store user and service data.
2. **User Authentication:** Implemented registration and login features for different roles (admin, service professional, and customer).
3. **Admin Features:** Admin is provided with a dashboard to manage users, services, and approve service professionals.
4. **Service Management:** The admin can create, update, or delete services, and customers can view and book them.
5. **Booking Process:** Customers can request services, and service professionals can accept or reject them.
6. **Review System:** After a service is completed, customers can review the service professional.
7. **Database Design:** The project required careful consideration of the relationships between the service, user, and booking data.

The front-end was built using Jinja2 templates with Bootstrap for styling, and SQLite was used as a database for data storage. All operations were implemented to run on the local machine.

### Frameworks and Libraries Used

1. **Flask:** The primary framework for the backend logic and routing.
2. **SQLite:** A lightweight database for storing user, service, and request data.
3. **Jinja2:** Templating engine for generating dynamic HTML.
4. **Bootstrap:** Front-end framework used for responsive design and UI components.
5. **Flask-Login:** For user authentication management (for session management).
6. **Flask-WTF:** For form handling and validation.
7. **Flask-SQLAlchemy:** ORM for database operations.

## ER Diagram

Below is the Entity-Relationship (ER) Diagram for the Household Services Application:



### Explanation:

- **Customer** has many **Service Requests** (1 to many).
- **Service** is linked to many **Service Requests** (1 to many).
- **Service Professional** is linked to many **Service Requests** (1 to many).

## API Resource Endpoints

The application doesn't expose REST APIs for full CRUD operations but does include the following basic API-like operations:

1. **POST** /register: For user registration (customer and service professional).
2. **POST** /login: For logging in users (admin, customer, service professional).
3. **POST** /book\_service: For creating a service request (customer).
4. **POST** /accept\_request: For service professionals to accept or reject service requests.
5. **GET** /view\_service\_requests: For service professionals to view and manage their requests.
6. **GET** /view\_professionals: For customers to view service professionals based on the service they need.

## Drive Link for Presentation Video

The project presentation video demonstrating the application's functionality is available at the following link:

<https://drive.google.com/file/d/1HrtPSgPqhF1mCmoH8DJzwmtCd1ZfdetR/view?usp=sharing>