Problem Statement

In today's fast-paced environment, accessing reliable and efficient home servicing solutions remains a challenge for both service providers and customers. Existing platforms lack seamless coordination between multiple stakeholders - administrators, service professionals, and customers - leading to inefficiencies such as delayed service, lack of transparency, inconsistent quality, and poor user experience.

There is a need for a multi-user application that streamlines the interaction between service professionals, customers, and an administrator, providing a platform that offers comprehensive home servicing solutions. This platform should enable:

1. Service Professionals to offer and manage their services efficiently.
2. Customers to easily request, schedule, and track home services.
3. Administrators to oversee operations, ensure quality control, and manage user activities.

The solution must ensure real-time coordination, transparency, and reliability to improve the overall service experience and satisfaction for all users.

Frameworks & Libraries

- Flask
- Jinja2 Templates
- Bootstrap CSS & Icons
- SQLite for data storage
- SQLAlchemy
- Flask-WTF
- Flask-Bcrypt
- Flask-Login
- Email_Validator
- Chart JS

Roles

The platform will have three roles:-

1. Admin - root access - it is a superuser of the app and requires no registration.

- Admin login redirects to the admin dashboard
- Admin will monitor all the users (customers/service professionals)
- Admin will create a new service with a base price
- Admin will approve a service professional after verification of profile docs
- Admin will block customer/service professionals based on fraudulent activity/poor reviews

2. Service Professional - An individual that provides the service

- Login/Register
- Service professionals will accept/reject a request
- Each professional may have;
    - ID
    - Name
    - Date created
    - Description
    - service_type
    - Experience

- One professional is good at one of the services only
- He/she can accept/reject an assigned service request
- Professional profiles are visible based on customer reviews
- The professional will exit the location after the service is closed by the customer

3. Customer - an individual who has to book a service request
   - Login/Register
   - View/Search the service by the name/location pin code
   - Open/close a service request
   - He/she can post reviews/remarks on the closed service

Terminologies

Service - It refers to the type of service that the customer is looking for e.g. AC servicing, plumbing etc.
Each service may have:

- ID
- Name
- Price
- Time required
- Description

Service Request - A customer creates a service request providing the type of service the customers is looking for, when is it required etc.
A service request may contain the following attributes:

- id - primary key
- service_id(foreign key-services table)
- customer_id(foreign key-customer table)
- professional_id(foreign key-professional table)
- date_of_request
- date_of_completion
- service_status(requested/assigned/closed)
- remarks (if any)

Core Functionalities

1. Admin login and user login
   - A login/register form with fields like username, password etc. for customer, service professional and admin login
   - You can create separate forms for each type of user
   - You can either use a proper login framework, or just use a simple HTML form with username and password (we are not concerned with how secure the login or the app is)
   - The app must have a suitable model to store and differentiate all the types of user of the app.

2. Admin Dashboard - for the Admin
   - Admin login redirects to admin dashboard
   - Admin will manage all the users (customers/service professional)
   - Admin will approve a service professional after verification of profile docs
   - Admin will block customer/service professional based on fraudulent activity/poor reviews

3. Service Management - for the Admin
    - Create a new service with a base price.
    - Update an existing service - e.g. name, price, time_required and/or other fields
    - Delete an existing service

4. Service Request - for the customers
    - Create a new service request based on the services available
    - Edit an existing service request - e.g. date_of_request, completion status, remarks etc
    - Close an existing service request.

5. Search for available services
    - The customers should be able to search for available services based on their location, name, pin code etc.
    - The admin should be able to search for a professional to block/unblock/review them.

6. Take action on a particular service request - for the service professional
    - Ability to view all the service requests from all the customers
    - Ability to accept/reject a particular service request
    - Ability to close the service request once completed

Navigation

Users can navigate to different sections of the App using Navbar. Clicking on the App icon(top left corner of Navbar) shows the Home page. For Responsive screens, Navbar can be opened by clicking the Hamburger sign on the top right corner. Login page is common for all users. However, sign-up for Service Professional and Customer is separate. After sign-up, the user is redirected to login. On logging in, the user is presented with a Dashboard. On the navbar, towards the left side (top for responsive), 3 links - Dashboard, Search, Summary will be visible for navigating to respective sections. A dropdown with user logo is provided towards the right side (down for responsive) of the navbar to navigate to the profile page or logout. The profile page shows user details and has provision to edit user profile, and delete user account. The Dashboard, Search, Summary sections are different for Admin, Service Professional and Customer.