A-Z Household Service Application Report

1. Student Details

- Name: Nishant Kumar
- Roll No.: 22f3003042
- Course: Modern Application Development I

2. Project Overview

- **Title**: A-Z Household Service
- **Objective**: The Household Services Application is a multi-user application designed to connect customers seeking home services (such as cleaning service, gardening Service, etc.) with verified service professionals. The application also includes an admin role responsible for managing users, services, and monitoring interactions.

3. Problem Statement

The increasing demand for household services necessitates a platform where customers can easily book verified professionals and manage service requests. At the same time, service professionals need a streamlined way to manage their jobs, and admins need tools to oversee users and services.

The problem the Household Services Application solves:

- **Customers**: Easily book, search, and review service professionals.
- **Service Professionals:** Manage service requests, accept, or reject jobs, and build a trusted profile based on customer reviews.
- **Admin:** Oversee the system, approve professionals, manage services, and block users as needed.

4. Approach and Solution

To address the problem, the platform implements a role-based system:

- **Admin**: Manages users, approves professionals, creates, and updates services, and monitors the system.
- **Service Professionals**: Accept or reject customer requests, close completed requests, and receive ratings from customers.
- **Customers**: Search for services, book professionals, manage requests, and provide reviews for completed services.

The application is developed with the following structure:

- **User Authentication**: Separate login forms for admin, and same for both service professionals, and customers.
- **Service Request Management**: Customers can create, edit, and close service requests, while professionals can accept/reject them.
- **Admin Control**: Full management capabilities for services and users.

5. Framework and Libraries Used

The project is built using the following technologies:

- **Flask:** Used for the backend server and routing logic.
- **Jinja + Bootstrap:** Utilized for dynamic HTML generation and responsive frontend design.
- **Werkzeug:** Werkzeug is a collection of utilities for WSGI applications. Specifically, secure_filename from Werkzeug is used to securely manage file uploads by ensuring filenames are safe for the file system.
- **bcrypt:** Bcrypt is used for hashing and verifying passwords, ensuring user passwords are stored securely in the database. It provides a strong hashing algorithm resistant to brute-force attacks.
- **os:** The os module is used for handling operating system-level functions, such as creating file paths and interacting with the file system (e.g., saving uploaded files).
- **datetime:** The datetime module is used to manage dates and times, especially for recording timestamps like the date of service requests or their completion.

6. Core Functionalities

a. Admin Role

- **Admin Login**: Admins can log in through a dedicated interface and access the dashboard.
- **User Management**: The admin can view all users, approve service professionals, and block or unblock users based on reviews or activity.
- **Service Management**: Admins can create, update, and delete services.

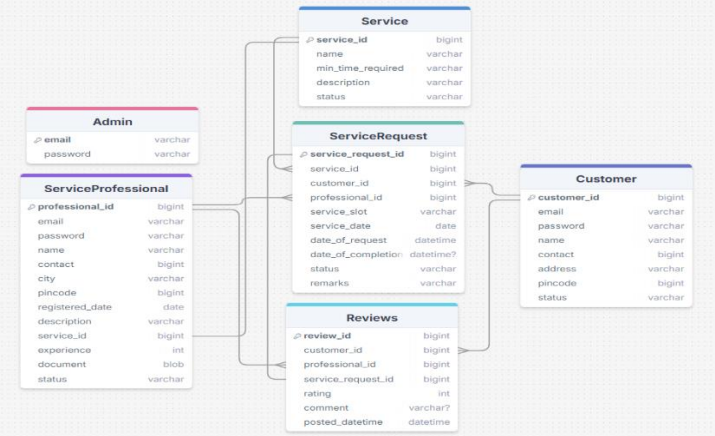
b. Service Professional Role

- **Profile:** Each professional has a profile with ID, name, service type, experience, etc.
- **Service Request Handling:** Professionals can view, accept, reject, and complete service requests. Once a service is complete, the request is marked as "closed."

c. Customer Role

- **Service Search:** Customers can search for services by name, location, or pin code.
- **Service Request Management:** Customers can create and close service requests and leave reviews for professionals after completion.

7. ER Diagram



8. Presentation Video

https://drive.google.com/file/d/1edbPYJHLc6afqLKwxQhjzdIYk747tsQ5/view?usp=sharing