MODERN APPLICATION DEVELOPMENT - I

Student Details:

- **Name**: Parth Kacha
- **Roll No.:** 23f1002650
- **Email**: 23f1002650@ds.study.iitm.ac.in
- **Submission Term**: September Term 2024
- **App. Name**: **A-Z Household Services**

Project Details:

Project Title: Household Services Application

Problem Statement:

The project aims to create a multi-user platform that connects service professionals with customers in need of household services. The application should allow customers to request services, view service professionals, review services after completion, and allow professionals to accept or decline service requests. Additionally, the admin should be able to manage the entire system, including services, professionals, and customer requests.

Approach:

Approached the problem by dividing the functionality into different sections for customers, service professionals, and admins:

- **Customer Features:**
    - Customers can sign up, view available services, search by service name/address/price, and request service bookings.
    - They can track the status of their requests and provide reviews after the service is completed.
    - If multiple choice of professionals is there for single service, then customer can book service by seeing average rating of that professional.

- **Professional Features**:
    - Service professionals can sign up, view service requests based on their expertise, search by address/service status, and accept or reject these requests.
    - They can update the status of their service requests.

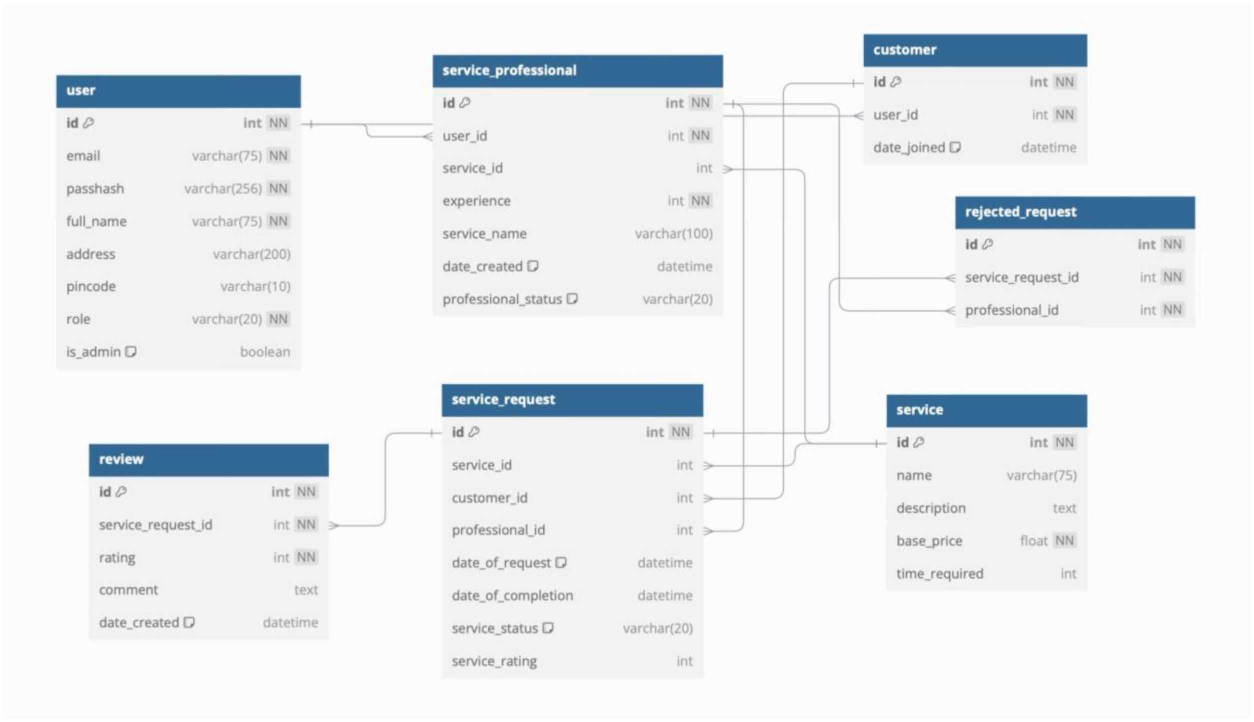
- **Admin Features**:
    - Admin can manage services, view requests, assign professionals, search service information, customer details, professional details, block customer or professional if any fraudulent is seen and monitor the entire system.

Using Flask as the backend framework to implement various routes to handle the different functionalities required for the platform, such as service request management, creating services, and customer feedback, etc.

Frameworks and Libraries Used:

1. **Flask**: A lightweight web framework to build the application backend and manage routing.
2. **Jinja2**: A templating engine to dynamically render HTML templates.
3. **SQLite**: A Database Management System to store application data.
4. **Flask-SQLAlchemy**: To interact with the database using an ORM.
5. **Werkzeug**: Utility for securely managing passwords and authentication.
6. **Session Management**: Used for maintaining user sessions and authentication.
7. **Datetime**: Python library for handling date and time operations.
8. **Html/Css/Javascript**: To create the structure and content of the web pages, to style the web pages, to enhance the dynamic behavior of the application.
9. **Bootstrap**: To design a responsive and visually appealing frontend.
10. **ChartJS:** For creating different types of charts on the admin/customer/professional dashboards.

ER Diagram:



The database for this project is designed with the following tables and relationships:

User Table:

- Stores user credentials and personal details.

Service Table:

- Stores the different services that customers can request.

ServiceRequest Table:

- Stores information about customer requests for services, including the live status of the request.

Review Table:

- Stores the feedback given by the customers after the service is completed.

ServiceProfessional Table:

- Stores information about the service professionals, including their expertise and personal details.

Customer Table:

- Separates the customers from user by role for further usage.

Relationships:

- A **user** can either be a **customer** or a **professional** (One-to-Many relationship between User and ServiceProfessional).
- A **service request** is linked to a **customer** and a **service** (Many-to-One relationship between ServiceRequest and User, and ServiceRequest and Service).
- **Reviews** are linked to **service requests** (One-to-One relationship between Review and ServiceRequest).
- A **service** can have multiple **service professionals** (Many-to-Many relationship between Service and ServiceProfessional).

Project Video:

https://youtu.be/zsettysl-Nk