MAD1 Project - Household Services Application

• Name: Nirmit Desai

• Roll No: 23F3001502

• Email: 23F3001502@ds.study.iitm.ac.in

• Description: Pursuing Btech. Computer Engineering(II year)

Problem Statement:

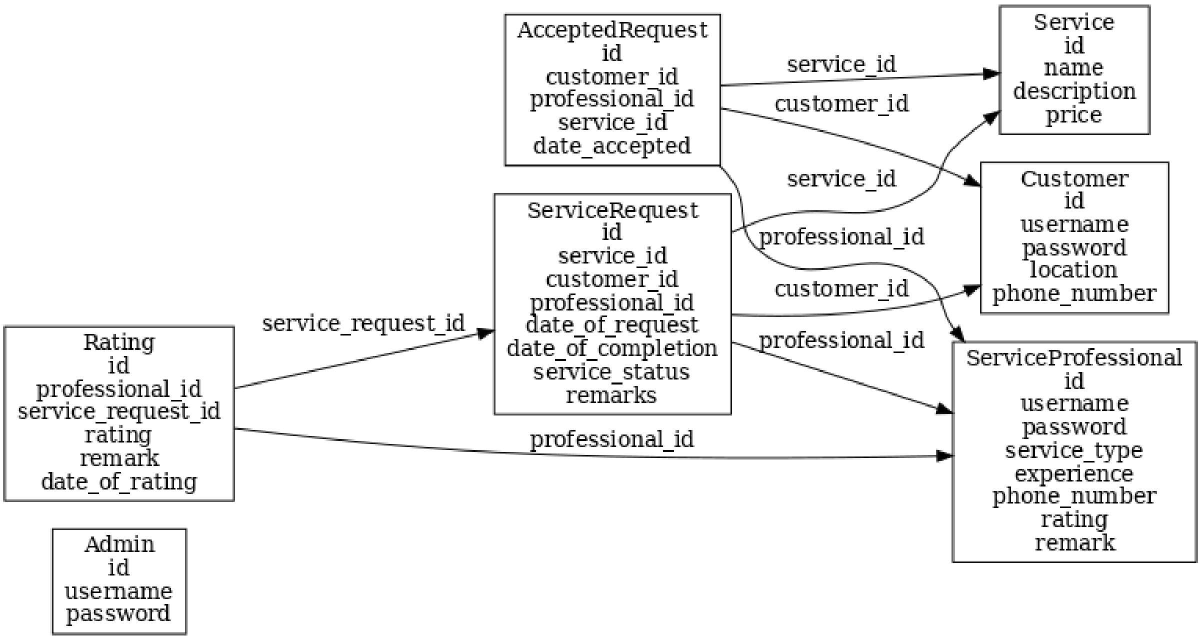
Design a platform that connects customers with service professionals, allowing customers to book services, service professionals to manage requests, and administrators to oversee the system.

Frameworks and Libraries Used

- **Backend Framework:** Flask
- **Database:** SQLAlchemy (SQLite backend)
- **Security:** werkzeug.security for password hashing
- **File Management:** werkzeug.utils for secure file handling
- **Authentication:** Flask-Login for session management
- **Others:** pytz (time zones), datetime (date/time operations), os (file handling)

Tables and Relationships:

- **Admin:** Manages customers, professionals, and services.
- **Customer:** Books services, rates professionals.
- **ServiceProfessional:** Provides services, interacts with customers.
- **Service:** Defines available services with descriptions and pricing.
- **ServiceRequest:** Links customers, professionals, and services with request details.
- **AccpetedRequest:** Displays accepted request on customer's homepage
- **Rating:** Tracks ratings and remarks for services.



ER DIAGRAM

API Resource Endpoints

/ (GET): Render the homepage.

/login/customer (POST): Handle customer login.

/register/customer (POST): Register a new customer.

/customer_homepage (GET): Show available services for the logged-in customer.

/services/<service_name> (GET/POST): View and book services by name.

/logout/customer (GET): Logout a customer and redirect to the homepage.

/edit_profile (GET/POST): Allow a customer to update their profile details.

/customer_summary/<int:customer_id> (GET): Display a customer's service requests and details.

/view_customers (GET): Display all customers in the database.

/view_services (GET): Display all services in the database.

/view_service_requests (GET): Display all service requests in the database.

/view_accepted_services (GET): Display all accepted services.

/login/admin (POST): Handle admin login.

/admin_homepage (GET): Render the admin dashboard with top customers, services, and professionals.

/delete_professional/<int:prof_id> (POST): Delete a professional's record from the database.

/add_service (POST): Add a new service to the database.

/logout/admin (GET): Logout the admin and redirect to the homepage.

/search/admin (GET/POST): Search for customers, professionals, or services.

/view_professional (GET): View all registered service professionals.

/professional/<int:professional_id>/ratings (GET): View ratings and remarks for a specific professional.

/login/professional (POST): Handle professional login.

/register/professional (POST): Register a new service professional.

/professional_homepage (GET): Display service requests for the logged-in professional.

/service_request/<int:request_id>/action (POST): Approve or reject a service request.

/edit_profile_professional (GET/POST): Allow a service professional to update their profile details.

/accepted_service/<int:service_id> (GET/POST): View details of a professional who accepted a service request.

/rate/<int:service_id> (GET/POST): Rate a professional after service completion.

/view_ratings (GET): View all ratings and remarks in the database.

/chat_with_customer (GET): Open chat functionality between a professional and a customer.

/chat_with_professional (GET): Open chat functionality between a customer and a professional.

Google drive link for video:

https://drive.google.com/file/d/14CQxchq0PwbkUSi0k-YJMnnHM3IwTUo8/view?usp=sharing