Project Report : HomeVibes

Name: Abhay Sharma Roll No: 23F3001400

Course: MAD I

Institute: IITM (BS in Data Science and Applications)

Presentation Video Link:
https://drive.google.com/file/d/1ZwdWUbruLCzyO69xwEX9eahBE9CuBgYA/view?usp=sharing

Project Title : Household Services Application (HomeVibes)

Project Statement:

It is a multi-user app (requires one admin and other service professionals/ customers) which acts as platform for providing comprehensive home servicing and solutions.

Approach and Execution:

Step 1: Listing Functionalities and Requirements

I meticulously outlined the key features and requirements:

- **Service Management:** Create, update, and manage service details.
- **User Management:** Verify professionals, block users on basis of poor reviews or fraudlent activity. Track user data (personal info, ratings, reviews) etc.
- **Service Requests:** Allow customers to request and track services while ensuring professionals can manage assigned requests.
- **Rating System:** Enable both customers and professionals to rate each other for transparency and accountability.
- **Search Functionality:** Simple, user-friendly search for services and professionals.
- **CRUD Operations:** Efficiently implemented to maintain database integrity.

Step 2: Step-by-Step Development Process

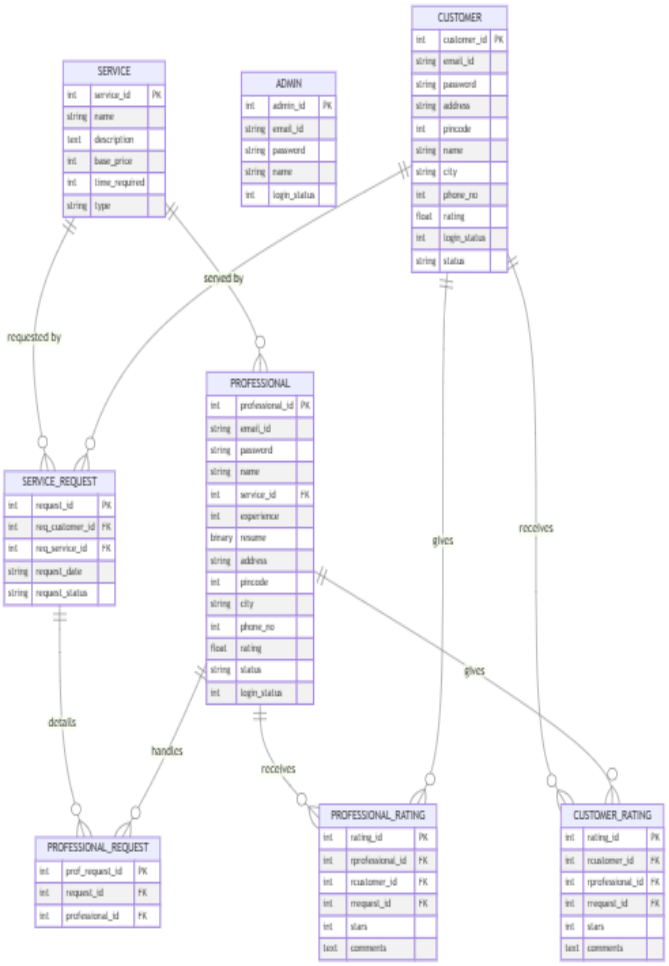
I approached the project methodically, focusing on one functionality at a time:

1. **Sign-up Pages**: Designed and implemented both frontend and backend validations for seamless user registration.
2. **Dashboards**: Built dedicated dashboards for admin, customers, and professionals to organize their respective activities.
3. **Core Functionalities**: Implemented features for service requests, rating systems, and search options.
4. **Profile and Summary Pages**: Added user profiles and summary pages to provide a holistic view of activity for all users.
5. **Error Handling and Enhancements**: Ensured smooth navigation by fixing errors and enhancing the overall system.
6. **CRUD APIs**: Developed RESTful APIs for customer and professional operations to enable efficient data interaction and integration.

Frameworks Used:

HTML5, CSS, JavaScript, SQLite3, DB Browser, Flask, Flask-SQLAlchemy, Flask-RESTful, Jinja2, Matplotlib (for Pie Charts and Bar Graphs), Mermaid Editor (for ER Diagram), Swagger.io (for API testing).

ER Diagram:



API Resource Endpoints:

Customer APIs:

- POST /api/customer: Registers a new customer.
- GET /api/customers/<customer_id>: Retrieves customer details.
- PUT /api/customers/<customer_id>: Updates customer details.
- DELETE /api/customers/<customer_id>: Deletes a customer profile.

Professional APIs:

- POST /api/professional: Registers a professional.
- GET /api/professional/<professional_id>: Retrieves professional details.
- PUT /api/professional/<professional_id>: Updates professional details.
- DELETE /api/professional/<professional_id>: Deletes a professional profile.