Project Report

Modern Application Development I Project

Author:

Name: **Ayush Kumar**, Roll no: **24f1002092**, Email: **24f1002092@ds.study.iitm.ac.in**, I am currently a student at the **Diploma Level**.

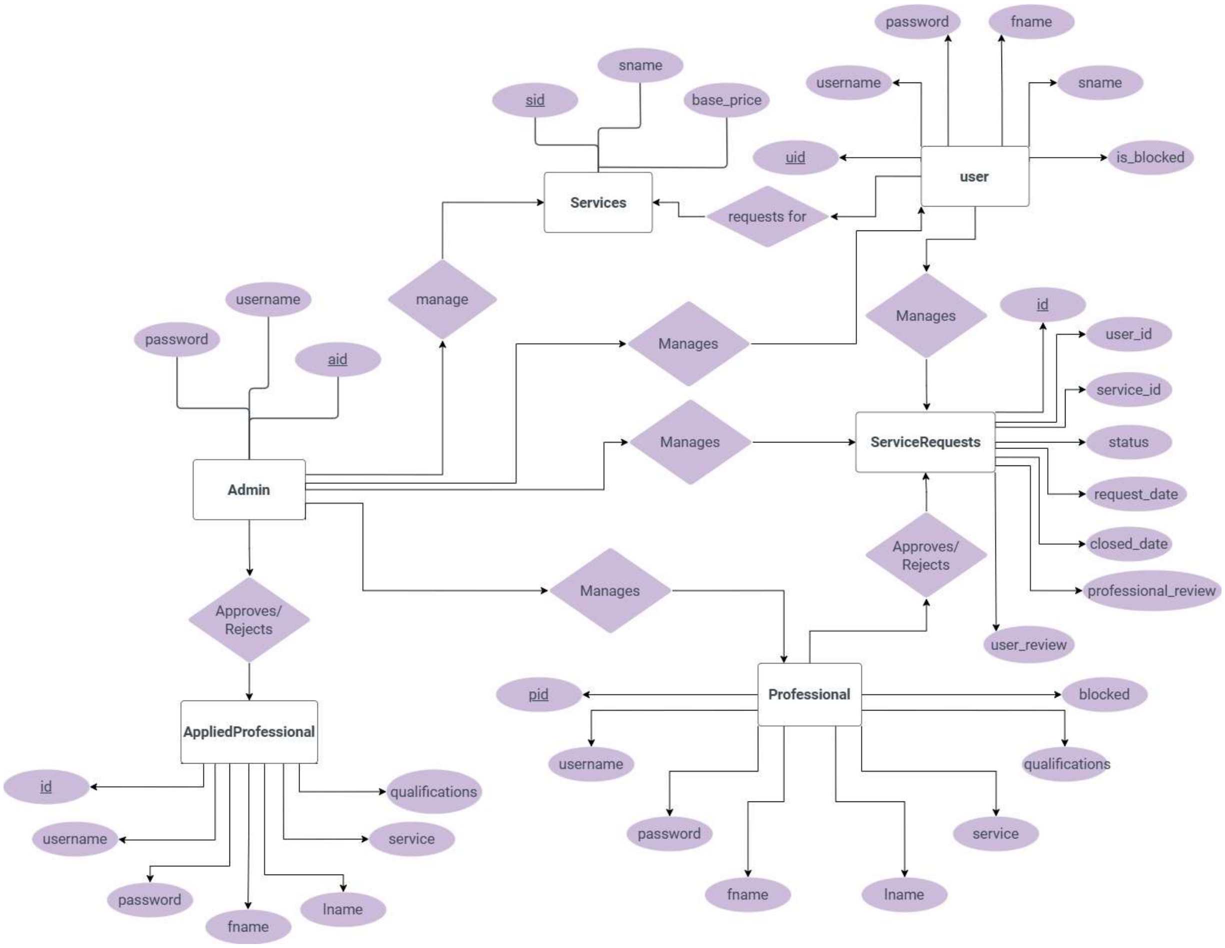
Description:

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. I approached the problem statement by designing a structured Flask application with clearly defined roles, database models, and dynamic routes to handle user, professional, and admin functionalities seamlessly.

Frameworks & Libraries Used:

Backend:	**Frontend:**	**Database:**
Flask: For routing and handling HTTP requests. **Flask-SQLAlchemy**: ORM for database models and queries. **Werkzeug**: For secure password hashing (generate_password_hash). **Flask-Session**: Manages user sessions (login state).	**Jinja2**: Template engine for rendering dynamic HTML. **Bootstrap**: CSS framework for responsive and styled UI.	**SQLite**: Lightweight relational database.

ER Diagram:



Db Schema Design:

User:

- uid: Unique identifier for the user (Primary Key).
- username: User's unique username.
- password: User's password.
- fname: User's first name
- lname: User's last name (optional).
- is_blocked: Boolean indicating if the user is blocked.

Services:

- sid: Unique identifier for the service (Primary Key).
- sname: Name of the service (unique).
- base_price: Base price of the service.

Professional:

- pid: Unique identifier for the professional (Primary Key).
- username: Professional's unique username.
- password: Professional's password.
- fname: Professional's first name.
- lname: Professional's last name (optional).
- service: Name of the service the professional provides.
- qualifications: Professional's qualifications (optional).
- blocked: Boolean indicating if the professional is blocked.

Admin:

- aid: Unique identifier for the admin (Primary Key).
- username: Admin's unique username.
- password: Admin's password.

AppliedProfessional:

- id: Auto-generated identifier for the professional (Primary Key).
- username: Professional's unique username.
- password: Professional's password.
- fname: Professional's first name.
- lname: Professional's last name.
- service: Name of the service the professional provides.
- qualifications: Professional's qualifications.

ServiceRequest:

- id: Unique identifier for the request (Primary Key).
- user_id: Foreign Key linking to the User model (uid).
- service_id: Foreign Key linking to the Services model (sid).
- status: Status of the request (default is 'Pending').
- request_date: Date the request was made (defaults to current time).
- closed_date: Date the request was closed (optional).
- user_review: Review given by the user (optional).
- professional_review: Review given by the professional (optional).
- user: Relationship linking to the User model.
- service: Relationship linking to the Services model.

API Resource End Points: In this project, API resource endpoints are primarily defined using Flask routes. While the project is focused on serving HTML pages (a server-rendered web app), some endpoints act as resources for specific tasks like accepting or rejecting requests, managing services, etc.

1. **User Endpoints:**
   - POST /user/request_service/<service_id>: Create a service request.
   - POST /user/delete_request/<request_id>: Delete a pending request.
   - POST /user/close_request/<request_id>: Mark a request as closed with a review.
2. **Professional Endpoints:**
   - POST /professional/accept_request/<request_id>: Accept a request.
   - POST /professional/reject_request/<request_id>: Reject a request.
   - POST /professional/complete_request/<request_id>: Complete a request with a review.
3. **Admin Endpoints:**
   - POST /admin/services/delete/<service_id>: Delete a service and its requests.
   - POST /admin/professionals/block/<professional_id>: Block a professional.
   - POST /admin/professionals/unblock/<professional_id>: Unblock a professional.
   - POST /admin/professionals/accept/<application_id>: Approve a professional.
   - POST /admin/professionals/reject/<application_id>: Reject a professional

Video Link: https://drive.google.com/file/d/1HwImKdP7_6zNapfdrssFai3raAnW5xy6/view?usp=sharing