Project Report
Modern Application Development I
Household Services Application

Name - Pratik Dey
Roll no. - 21f2000710

Description:

The Household Service App is a platform that connects customers, service professionals, and administrators to streamline household services. Customers can browse and request services like AC repair or plumbing, specifying schedules and preferences. Services are defined by attributes such as ID, name, price, and duration.

The app offers role-specific features:

- **Admins** manage users, verify professionals, update services, and address fraud or complaints.

- **Customers** search for services, manage requests, and track status.

- **Professionals** view and respond to service requests, managing their workflow efficiently.

This user-friendly solution ensures seamless service delivery and management.

Approach:

The application was developed using Flask as the web framework to manage routing, process requests, and facilitate interactions with the database. For database operations, SQLAlchemy ORM was employed to define models for key entities such as Customer, Professional, Service, and ServiceRequest, ensuring efficient communication with the database.

Dynamic content was rendered using Jinja2 templates, allowing data—such as customer details and service requests—to be integrated seamlessly into the HTML templates from Flask routes. The user interface was styled using CSS, with static files being linked and served by Flask to create an appealing design. HTML forms were implemented to collect user inputs, such as customer information or service requests, which were processed by Flask routes and stored in the database.

In summary, the backend was handled by Flask, database operations were managed through SQLAlchemy, dynamic content was rendered with Jinja2, and the visual design was achieved using CSS, resulting in a cohesive full-stack solution.

Frameworks and Libraries used:

The following technologies have been used:

Flask: A lightweight Python web framework for building web applications. It manages the application setup, configurations, routing, and rendering of HTML templates.

SQLAlchemy: A Python Object Relational Mapper (ORM) that simplifies interactions with relational databases (such as SQLite) by using Python objects instead of raw SQL queries.

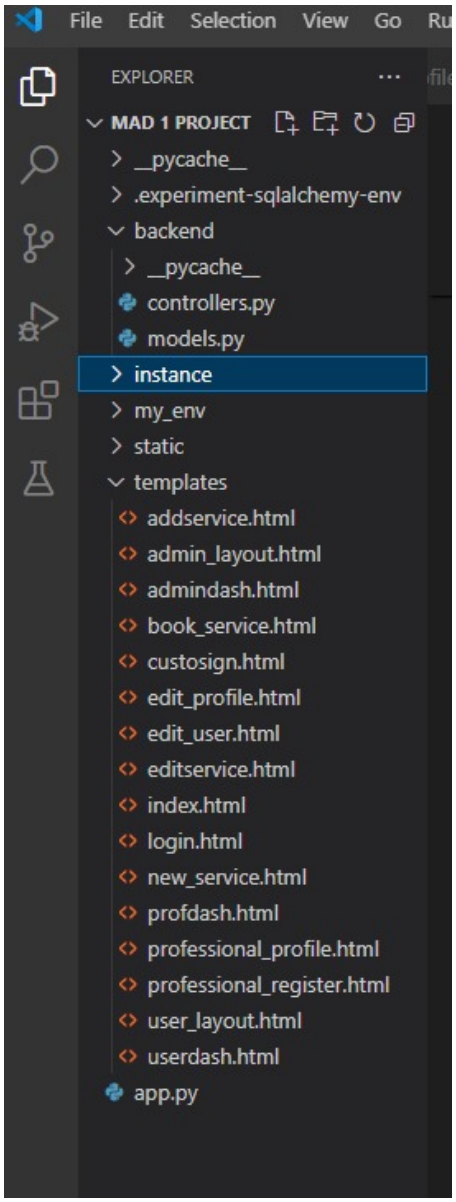
SQLite: A lightweight, file-based relational database management system, used to store data locally in a .sqlite3 file.

HTML Templates (Jinja2): The Jinja2 templating engine is integrated with Flask to render dynamic HTML content using the render_template() function.

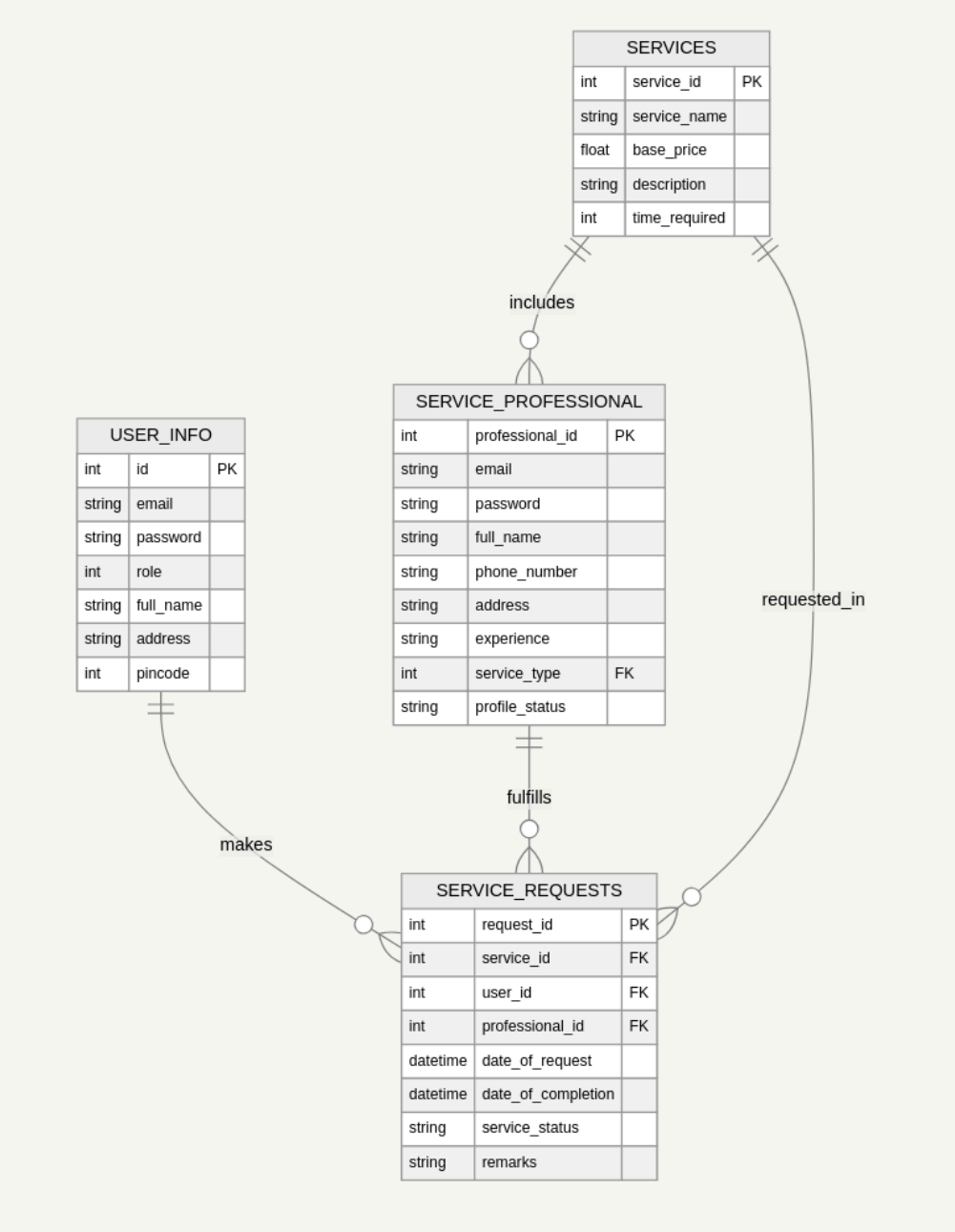
CSS: Used to style the web application by defining layouts, colors, fonts, and other design elements, with static files served through Flask.

The application is a Flask-based web solution for a "Household Service" system, incorporating database integration, user sessions, and dynamic content rendering.

File Structure:



Database Schema:



Presentation Video Link:

21f2000710.mkv
https://drive.google.com/file/d/1qIx1mintyUWDeRZ9nSbp99qo2FfxSxM3/view?usp=sharing