

Professional and Customer Interaction Coordination Platform

Households

Project report by **Keerthi Amudapu**
22f2000545

Modern Application Development – I
Sep, 2024 Term

Student Details

Name – Keerthi Amudapu

Roll no. – 22f2000545

Email – 22f2000545@ds.study.iitm.ac.in

About me

My academic journey began with a strong interest in technology, driven by my fascination with programming languages. I am passionate about contributing to and pushing the boundaries of technology. When I'm not focused on learning and developing my skills, I enjoy listening to music as a way to unwind and recharge.

Project Description

This application serves as a comprehensive platform for home servicing solutions, acting as an efficient connector between service professionals and customers. By streamlining the home servicing process, it makes it easier for customers to find reliable professionals while providing service providers with a platform to showcase their skills and manage their bookings effectively. The app features a robust admin interface that allows for seamless management of users, services, and bookings, ensuring a smooth experience for all parties involved.

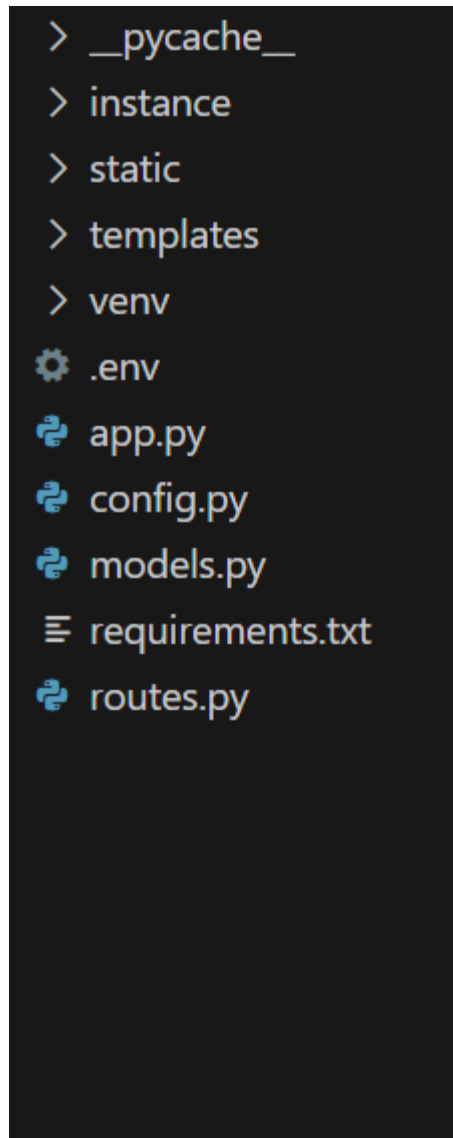
How i approached project statement

I utilized screencasts and live sessions to gain the knowledge and understanding necessary to tackle the project statement. By reviewing the wireframe and PDF provided by the course, along with the screencasts, I gained clarity on the project requirements.

To begin with, I created the necessary data models for the problem. The wireframe and PDF were particularly helpful in providing me with a foundational understanding of the project.

Technologies Used

1. **Flask:** Backend framework for building the web application.
2. **SQLAlchemy:** ORM (Object-Relational Mapping) tool for database interactions.
3. **SQLite:** Database management system for storing application data.
4. **HTML/CSS:** Frontend technologies for user interface design and interactivity.
5. **Flask-Login:** Extension for managing user sessions and authentication.
6. **DateTime:** Python library for handling date and time operations.
7. **Jinja2:** Template engine for rendering dynamic HTML content.
8. **Werkzeug:** Utility for securely managing passwords and authentication.
9. **Chart.js:** Library for creating different types of charts on the admin dashboard.
10. **Functools:** Python module that provides higher-order functions for functional programming tasks.



Architecture and Features:

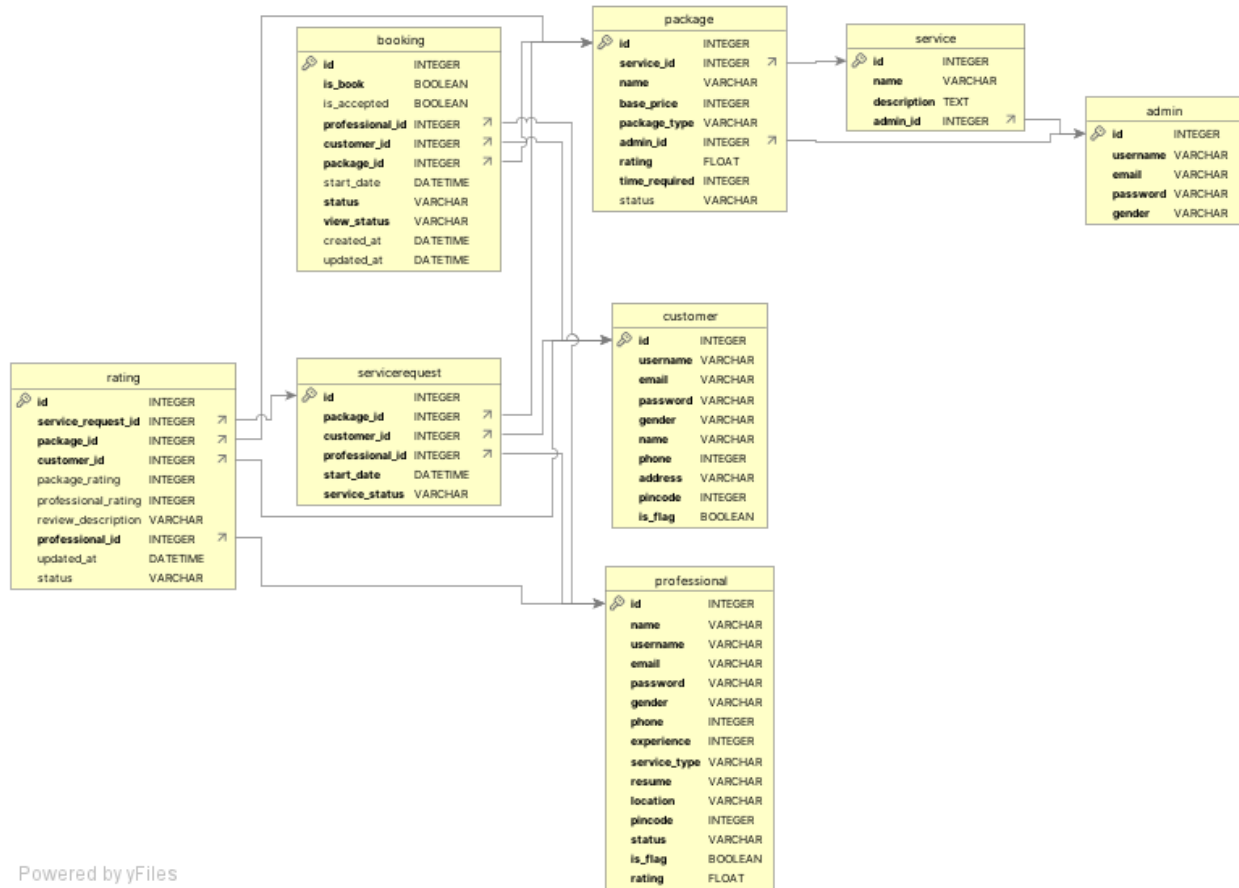
1. The project is organized using the Model-View-Controller (MVC) architecture, with the controllers handling logic and routing, templates for display, and models for interacting with a database.
2. Features implemented include:
 - New users can register and log in, while existing users (Admin, Professional, Customer) can log in using their username and password.

- Professional User: Professionals can accept or reject their assigned requests and search for requests on a particular date.
- Customer User: Customers can book service requests, select the corresponding professional, and search for available services to view the best packages.
- Admin User: The application has only one admin who creates all the services and their requirements.

3. Admin Login: The admin can manage influencers and sponsors (including blocking and unblocking them) The admin can also block professionals or customers and view all approved, flagged users, as well as all completed and pending service requests in the application.

DB Schema Design:

The database has several models/labels created. The database is designed to store the Professional and Customer information, Services, Packages, Service Requests and rating information.



Powered by yFiles

Challenges Faced:

The major challenge faced during the development of this project was implementing the user and admin login systems. It required careful planning and implementation to ensure that the login process was secure and robust. Another challenge was designing the database schema to support the various features of the platform while maintaining data integrity.

Future Improvements:

The project can be improved by adding more features such as better security, and enhanced search functionality. Additionally, the user interface can be improved to enhance the user experience.

Conclusions:

In conclusion, the project was successful in creating a platform for customers to book services and for professionals to manage their requests. The MVC architecture and use of Flask, Jinja2, and SQLAlchemy made the development process efficient and effective. The project can be improved in the future by adding more features and enhancing the user interface.

Project Video Demonstration Link:

Google Drive Link:

https://drive.google.com/file/d/1AShhwzmxN8Z6QCMfptzUB3Ik3Lp09Y_9/view?usp=sharing