Qwix.it

Household Services app

(Modern Application Development – I)

Student details

Name – Atharv Khare
Roll no. – 23f2004201
Email – 23f2004201@ds.study.iitm.ac.in
Mindful learner striving for personal and academic growth with integrity.

Project Overview

Qwix.it serves as a platform for connecting service professionals with customers seeking various household services. The application will enable customers to easily book services, track their requests, and provide feedback. Service professionals can accept or reject service requests, manage their requests and packags. The admin will oversee the platform, manage user accounts, and ensure smooth operations.

.

My Approach

1. Database Schema Design:

Designed the database schema, which serves as the foundation for storing and managing the application's data, and its relationships between tables for clarity.

2. User Flow and Wireframing

To visualize the user journey and interaction with the application, I created detailed wireframes, outlining the layout, navigation, and UI elements for each screen.

3. Template Creation:

I constructed the HTML templates using Jinja2, according to the wireframe designs. Each template was designed to be visually appealing, enhancing user engagement.

4. Flask Route Implementation:

I defined Flask routes to handle different user actions and requests.
These routes were organized based on user roles (admin, service professional, and customer). Then I implemented CRUD operations for various entities, such as users, services, and service requests. Finally additional features, such as search functionality, review systems, and notifications, were integrated.

Additional Features Implemented

Review Section: A section to view all reviews on professional dash, and on service packages for customers

Saved Packages: Option to save desired packages and view all saved packages on customer dash

Flag Packages: Ability for an admin to flag a suspicious package before deleting it

Special Services: Platinum (premium) and emergency service sections

Technologies used

Backend:

Flask: Python micro web framework
SQLAlchemy: Object-Relational Mapper (ORM)
SQLite: Database management system

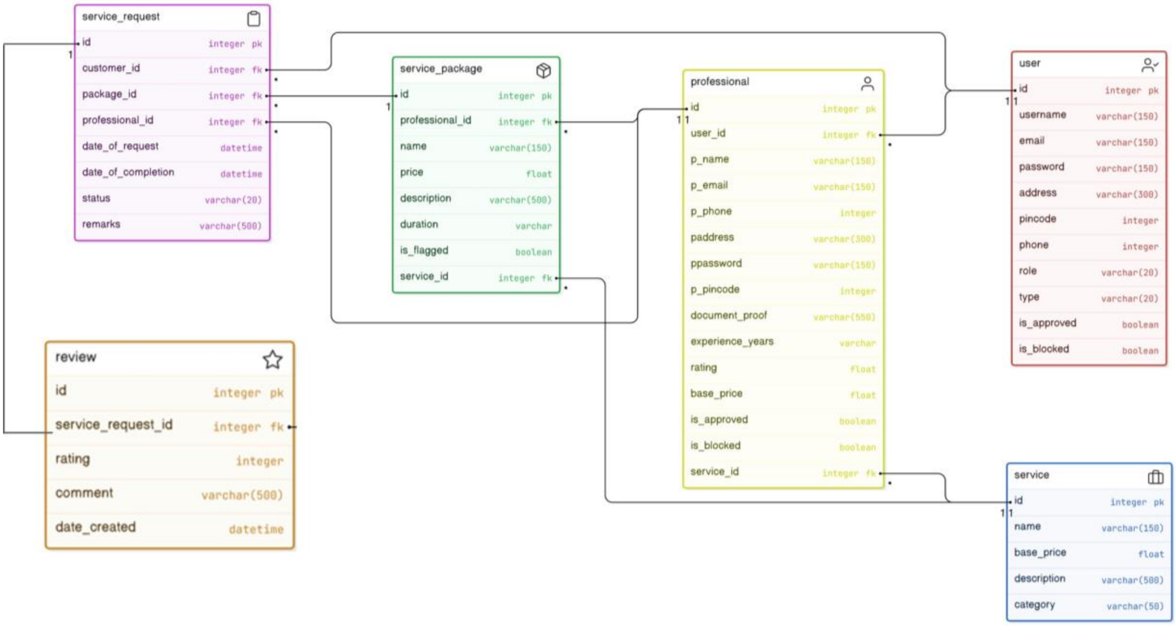
Frontend:

HTML: Structure of web pages
CSS: Styling of web pages
Bootstrap: CSS framework (used for accordion element)
Chart.js: JavaScript library for creating charts and graphs

Additional Libraries and Tools:

Flask-Session: Session management for storing user data across requests
Datetime: Date and time handling
Werkzeug: Secure password hashing and authentication
Pytz: Library for working with time zones (timezone)
Sqlalchemy.func: Provides functions for performing database operations like aggregation
Sqlalchemy.or_: Function for building OR clauses in SQL queries.

E-R Diagram



service_request	
id	integer pk
customer_id	integer fk
package_id	integer fk
professional_id	integer fk
date_of_request	datetime
date_of_completion	datetime
status	varchar(20)
remarks	varchar(500)

service_package	
id	integer pk
professional_id	integer fk
name	varchar(150)
price	float
description	varchar(500)
duration	varchar
is_flagged	boolean
service_id	integer fk

professional	
id	integer pk
user_id	integer fk
p_name	varchar(150)
p_email	varchar(150)
p_phone	integer
paddress	varchar(300)
ppassword	varchar(150)
p_pincode	integer
document_proof	varchar(550)
experience_years	varchar
rating	float
base_price	float
is_approved	boolean
is_blocked	boolean
service_id	integer fk

user	
id	integer pk
username	varchar(150)
email	varchar(150)
password	varchar(150)
address	varchar(300)
pincode	integer
phone	integer
role	varchar(20)
type	varchar(20)
is_approved	boolean
is_blocked	boolean

review	
id	integer pk
service_request_id	integer fk
rating	integer
comment	varchar(500)
date_created	datetime

service	
id	integer pk
name	varchar(150)
base_price	float
description	varchar(500)
category	varchar(50)

Wireframe↗

Project structure↗

Demonstration Video

https://drive.google.com/file/d/122T8s-6XwcYtmAFis1X-DSVTBAhcCSqq/view?usp=drive_link