Project Report for Final Project

Mordern Application Development - I

Author:

Harshal Kariya

21f3002536

21f3002536@ds.study.iitm.ac.in

Problem Statement

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

Description

In this project, we were supposed to use HTML, CSS, bootstrap, Dlask, SQLAlchemy, and other necessary modules to build an app for household services. We had to build a login/signup page for all the users including the admin and store the user details based on their roles inside the database. The app should work seamlessly and the CRUD operations should be working.

Admin functionalities-

There is only one admin and they can create new services, edit them, or even delete them. Admin can view the professionals and can flag/delete them too. Admin can also view all the service requests. Admin is accessible to a couple of services which shows the service requests based on their statuses (requested, accepted, completed) and various services opted by the customers based on the categories they belong to.

Professional functionalities-

A professional can view all the service requests that they have attended to and completed. They can also view the service requests accepted by them. They can see the pending requests that belong to the service they provide and have an option of accepting them. Once accepted the status of the service request changes from “requested” to “accepted” and subsequently this shows in the customer dashboard.

Customer functionalities-

Customers can view all the services available along with their details. They can book the service if the wish to. The service requests that belong to the customer are shown in the dashboard. Once a service is completed, the customer should close the service request, which will change the status of the request from “accepted” to “completed”. While closing, the customer should give a rating for the service and can also leave remarks.

Technologies used

1. Flask: used for building the web application

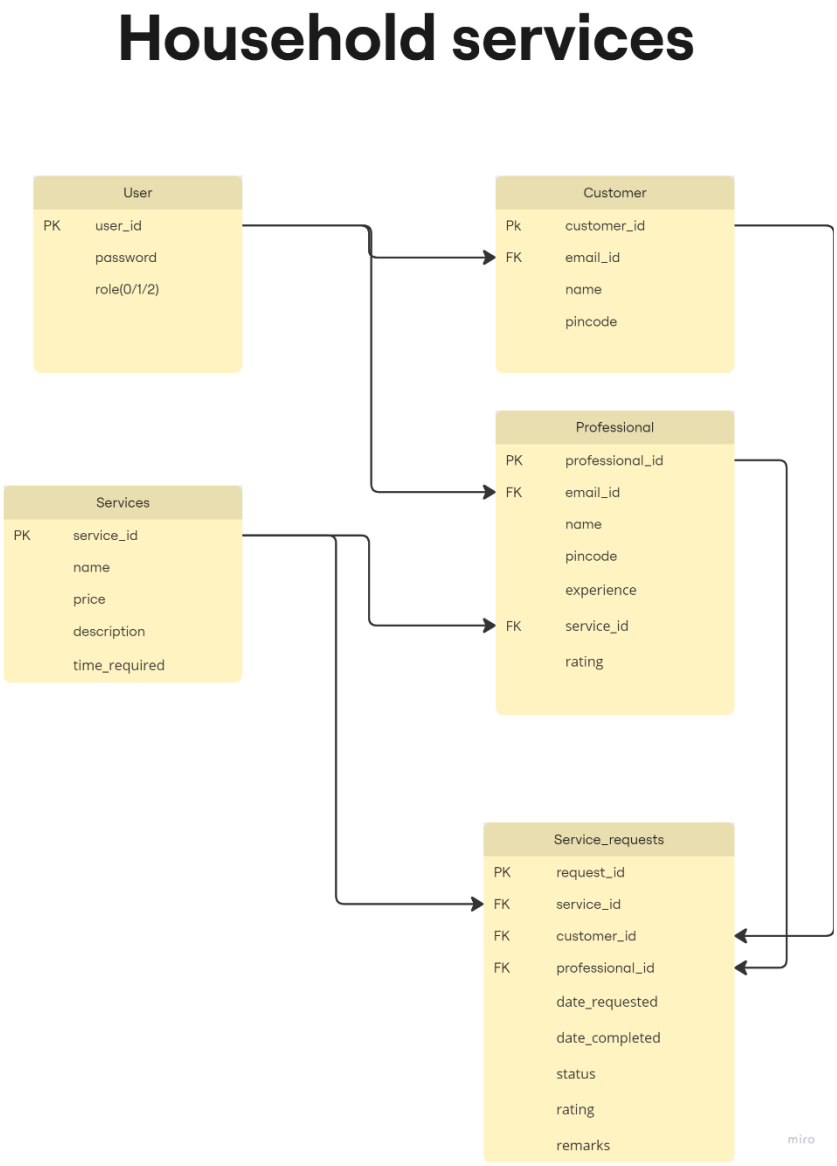
2. Flask-SQLAlchemy: extension of Flask, used to handle database connections across the app.

5. Datetime: This python module is used for time stamping the class operations.

6. Matplotlib: For data visualisation.

7. Requests: This python module is used to handle HTTP requests and responses.

DB Schema Design



The controllers that were used:

- @app.route("/login",methods=["GET","POST"])
- @app.route("/customer_register", methods=["GET","POST"])
- @app.route("/admin", methods=["GET","POST"])
- @app.route("/professional/<name>/<id>", methods=["GET", "POST"])
- @app.route("/customer/<name>/<id>", methods=["GET", "POST"])
- @app.route("/admin/add_service", methods=["GET", "POST"])
- @app.route("/admin/search", methods=["GET","POST"])
- @app.route("/customer/<name>/<id>/search", methods=["GET","POST"])
- @app.route("/admin/edit_service/<id>",methods=["GET","POST"])
- @app.route("/admin/delete_service/<id>",methods=["GET","POST"])
- @app.route("/admin/delete_prof/<id>",methods=["GET","POST"])
- @app.route("/customer/<name>/<id>/<sid>/book", methods=["GET", "POST"])
- @app.route("/professional/<name>/<id>/<rid>/accept",methods=["GET","POST"])
- @app.route("/customer/<name>/<id>/<rid>/close",methods=["GET","POST"])
- @app.route("/admin/summary")
- @app.route("/admin/view_prof/<pid>")
- @app.route("/admin/view_request/<rid>")
- @app.route("/professional/<name>/<pid>/view")

Video

Link:

https://drive.google.com/file/d/1_0eJmQOiQ74hYQ5imVBe9gTMrEPV7rOA/view?usp=sharing