PROJECT REPORT – MAD 1 (SEP TERM)

BY ANISHA SETH (**23f1002056@ds.study.iitm.ac.in**)

Problem Statement: https://docs.google.com/document/d/1waf_CKBLk25fkwF-R4KS7wLq4KTIPhUcAtj6if5N-zo/pub

OVERVIEW:

I am a recent graduate who, since approximately two years, have been studying in this Data Science Program and have learnt a lot in every step of the way, whether it's the quizzes or lab assignments, end terms or programming tests and last but certainly not the least, this project.

Before starting the project, everything almost seemed impossible. And starting it felt like taking a plunge into the deep waters and though I got frustrated when some code didn't work my way or an extra single curly bracket was the reason that code didn't work, not only has this project taught me the wonders of what goes behind the screens(literally) but has also been a great learning curve in my journey to programming and web development.

DESCRIPTION:

The problem statement revolves around a multi-user app with admin(sole), customers and professionals taking on different responsibilities and functions in bringing this project together. The mandatory frameworks used were- Flask for application code, Jinja2 templates + Bootstrap for HTML generation and styling and SQLite for data storage. Herein, starting with a home page, we had make a login page for sign in of all users, two kinds of registration pages for customers and professionals respectively, three different kind of dashboards with home page, summary page, profile page and search functionality each based on the different users as well as implemented all the logic behind those functionalities. All in all, every user has a pivotal role bringing together a full circle to make this app work.

Admin: Creates, edits and deletes services, approves, rejects and deletes professionals and can also delete customers. Has access to all customers, professionals and service requests credentials. Is basically the soul of the app.

Customer: Sends booking requests, closes accepted requests and gives feedback. Has access to all the services. Eyes of the app.

Professional: The foundation of the app, accepts or rejects service requests and provides customer rating.

TECHNOLOGIES USED:

- **Flask(Flask, render_template, request, url_for, redirect)** for application code
- **Flask-SQLAlchemy:** ORM (Object-Relational Mapping) to handle database connections all around the app.
- **HTML/CSS/JINJA2 TEMPLATES+BOOTSRTAP 5** for frontend user interface design.
- **OS** to interact with Operating System.
- **Datetime** to time-stamping the python modules.
- **Werkzeug** to work with uploaded files and check if they are secure.
- **Matplotlib** to make interactive plots and for data visualization.
- **SQLite:** A DBMS used for storing data.

ARCHITECTURE AND FEATURES:

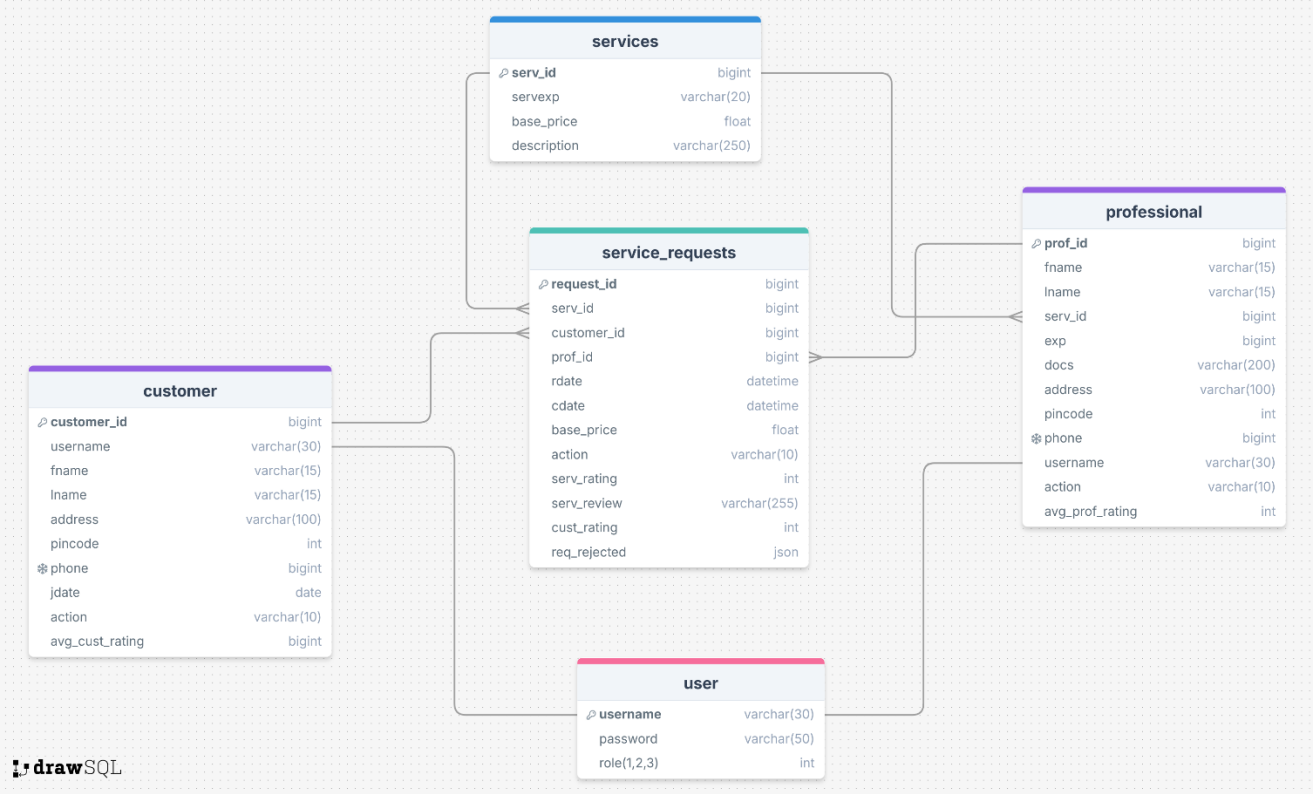
The Household application is named as **ServPro.**

- **Project Root Folder** contains a single project folder that in turn contains all codes, other folders and project report.
- **App.py**
- **Uploads** to save all resume uploads made by the professionals during registration.

- **Four Main Folders:**
    1. Backend- contains models.py and controllers.py
    2. Templates-All HTML templates
    3. Static- CSS, other images as well as matplotlib plot images.
    4. Instance Folder-file containing database

DB SCHEMA DESIGN

One username can have only one professional and customer and only one customer or professional can occupy a username (One-to-One). One customer can make many service requests and one service request can be made by only one customer. (Many-to-one-service requests to customer)Similar with professionals and services .One professional can expertise in only one service whereas one service can have many professionals.(Many-to-one for professionals to service.)



Presentation Link:

https://drive.google.com/file/d/1DF27zfMkN9EpNkd1Xo5f6DQjVHUTBYl-/view?usp=sharing