

PROJECT REPORT

A Z HOUSEHOLD SERVICES

QUESTION STATEMENT:

In the project, we were required to make Household Services Application using HTML, CSS, Bootstrap, flask, SQLAlchemy and other necessary modules to build it.

DESCRIPTION:

When the development server link is clicked, the main page of the app is displayed. From there, users are directed to the login page, which includes options such as "Register as Professional" for service professionals and "Create Account" for customers to register. The page also features fields for registered e-mail and password, allowing all three user types to log in to their respective dashboards.

ADMIN DASHBOARD:

- add, delete, edit and read all the services details (name, price, time required, description, etc)
- edit, delete, approve, reject based on the resume attached professionals registered for the service
- keep track of the service request generated
- edit your profile, summary (bar chart), search and logout

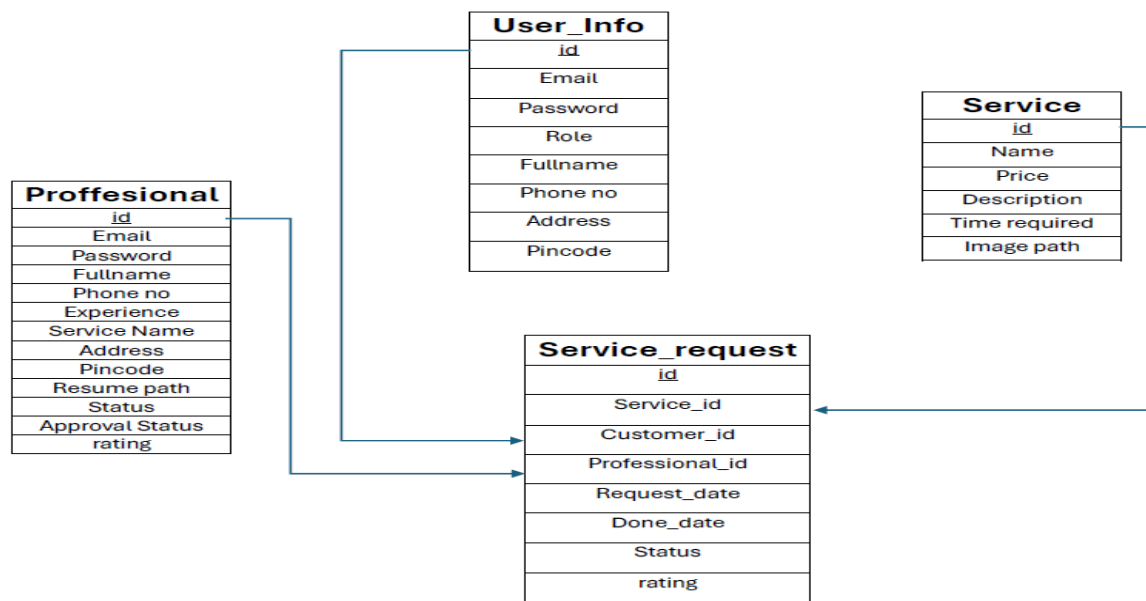
CUSTOMER DASHBOARD:

- View the available service details, and the respective professionals details, ability to book the professional.
- Search, Summary, Edit profile functionality.
- Close the request upon completion by giving the feedback.

PROFESSIONAL DASHBOARD:

- View the service generated by customer and can either accept/reject it
- Search, Summary ,Edit profile functionality

E R DIAGRAM:



FRAMEWORK AND LIBRARIES USED:

- | | | |
|-------------|--------------|--------------|
| 1. HTML/CSS | 4.SQLite | 7.Matplotlib |
| 2.Bootstrap | 5.Jinja2 | 8.Werkzeug |
| 3.Flask | 6.SQLAlchemy | 9.datetime |

API DESIGN:

User_Api: Implemented using get, post, put and delete methods for the User_Info table

"/api/get_user";"/api/add_user";"/api/edit_user/<id>";"/api/delete_user/<id>"

ARCHITECTURE :

- App.py: includes the python code.
- /static/service_images: includes the service images uploaded
- /uploads/resumes: includes the resumes uploaded by the professionals
- /instance: includes the sqlite database
- /static/images: includes the summary folders and logo of the app
- /static/styles: main.css for style
- /backend: includes models.py (to create database tables),controllers.py(includes all @app.routes and api_controllers.py(API resource)

PROJECT DEMO DRIVE LINK:

<https://drive.google.com/file/d/1b2wSvOlVTqdns7Yy1ZpaTK4xbGUBeDny/view?usp=sharing>