

HOUSEHOLD SERVICES WEB APP REPORT

Author:

- **Name:** Seetha Prajna DS
- **Roll No. :** 22f3003059
- **Email:** 22f3003059@ds.study.iitm.ac.in
- **About:** I am a masters graduate in Physics, currently pursuing the online BS Degree in Diploma Level. I love watching video essays and reading interesting articles.

About the App:

The household services app is web-based and can have multiple users. Customers can send requests to various approved professionals for different services managed by the admin. Customers and professionals can interact with each other through private or public requests. Customers can also rate and review the professional once the request is completed.

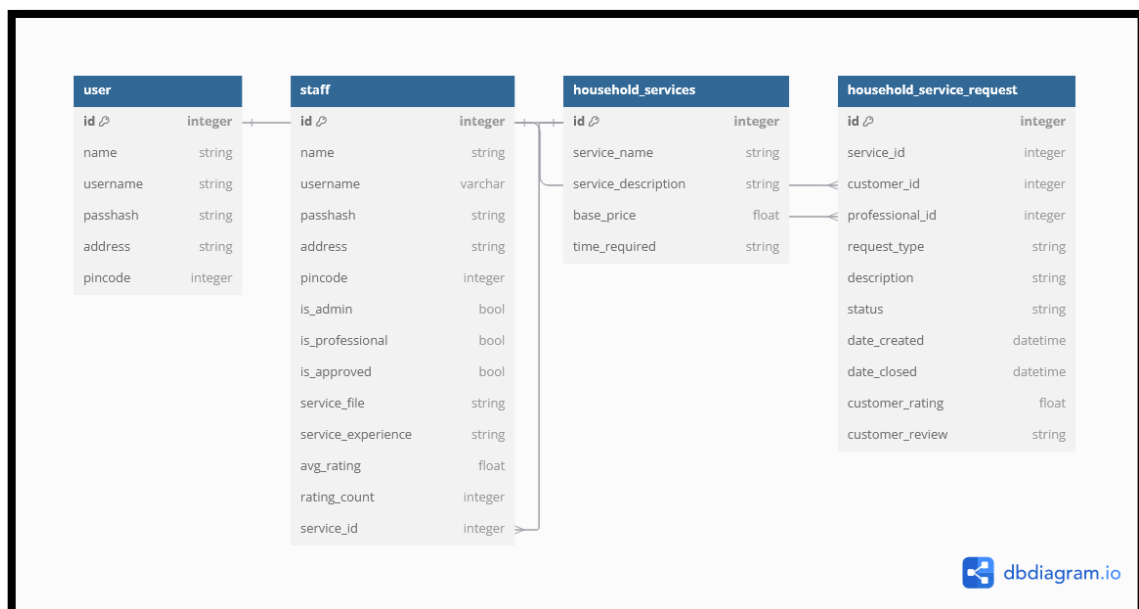
Frameworks used:

- **Flask** - for handling the backend of the app
- **Jinja2** - for templating and rendering
- **Bootstrap and MDB** - for layout and styling
- **Matplotlib and Seaborn** - for data visualization

Technologies used:

- **Flask (version 3.0.3)** - used as the core framework to build the app.
- **Flask-SQLAlchemy (version 3.1.1)** - for interface between the database and flask application.
- **Jinja2 (version 3.1.4)** - to create templates for dynamic web pages.
- **Matplotlib (version 3.9.2)** - for data visualization
- **SQLAlchemy (version 2.0.36)** - for interacting with the database through python.
- **SQLite** - as the database management system.
- **Werkzeug (version 3.1.3)** - to provide utility functions in terms of file handling and password management.

DB Schema and Design:



- **User**

- **id** (Primary Key)
- **Name** (Not Null)
- **Username** (Unique, Not Null)
- **Passhash** (Not Null)
- **Address** (Not Null)
- **Pincode** (Not Null)
- **requests_sent** (relationship with HouseholdServiceRequest) - to retrieve all the requests sent by a customer; a customer can send none or many requests.

- **Staff**

- **id** (Primary Key)
- **Name** (Not Null)
- **Username** (Unique, Not Null)
- **Passhash** (Not Null)
- **Address** (Nullable)
- **Pincode** (Nullable)
- **is_admin** (Boolean, Default: False)
- **is_professional** (Boolean, Default: False)
- **service_file** (Nullable)
- **service_experience** (Nullable)
- **avg_rating** (Float, Default = 0.0)
- **rating_count** (Integer, Default = 0)
- **service_id** (Foreign Key - HouseholdServices.id, Nullable)
- **service** (relationship with HouseholdServices) - to retrieve the service a professional provides; a professional can provide only one service
- **requests_received** (relationship with HouseholdServiceRequest) - to retrieve all the requests received by a professional; a professional can receive none or many requests.

- **HouseholdServices**

- **id** (Primary Key)
- **service_name** (Not Null)
- **service_description** (Not Null)
- **base_price** (Float, Default = 0.0, Not Null)
- **time_required** (String, Not Null)
- **professionals** (relationship with Staff) - to retrieve all the professionals enrolled for a particular service; one service can have none or many professionals enrolled to it.
- **requests** (relationship with HouseholdServiceRequest) - to retrieve all the requests for a particular service; one service can have none or many requests associated to it.

- **HouseholdServiceRequest**

- **id** (Primary Key)
- **service_id** (Foreign Key – HouseholdServices.id, Not Null)
- **customer_id** (Foreign Key – User.id, Not Null)
- **professional_id** (Foreign Key – Staff.id, Nullable)
- **request_type** (String (private/public), Not Null)
- **description** (Nullable)
- **status** (String (pending/accepted/closed/rejected), Not Null)
- **date_created** (Datetime, Not Null)
- **date_closed** (Datetime, Nullable)
- **customer_rating** (Float, Default = 0.0)

- **customer_review** (String, Nullable)
- **customer** (relationship with User) - to retrieve the customer associated with a request; a request can have only one customer associated with it.
- **req_service** (relationship with HouseholdServices) - to retrieve the service associated with a request; a request can be called for only one service at a time.
- **professional** (relationship with Staff) - to retrieve the professional associated with a request; a request can have only one professional associated with it.

Architecture

The Project folder consists of two folders: code and report. The code folder contains the instance folder, requirements.txt, app.py, config.py, models.py, and routes.py. It also contains two folders: static and templates. The static folder contains two folders to store images and pdfs uploaded. The templates folder contains the templates to be rendered. The report folder contains the report file: web_app_report.pdf in which the URL of the video demonstration of the app is attached.

Default Features

- **Admin**
 - The admin can create, edit, or delete services and set its details.
 - Admin can approve or reject (and delete) a service professional, regardless of their approval status.
 - Admin can delete customers.
 - Admin can search for customers and professionals based on address or pincode to view their request history or profile and reject/delete them.
- **Customer**
 - Customer can create, edit or delete private requests for a particular professional in any service.
 - Customer can search for professionals based on service name, address or pincode and view profiles, ratings and reviews.
 - Once the request is completed, customer can close the request upon which, rating and review can be given for the professional.
- **Professional**
 - Professional can view all the service requests sent.
 - Professional can accept or reject the requests sent.

Additional Features

- **Admin**
 - Admin can edit profile details.
 - Admin can view summarized details about customers and professionals.
- **Customer**
 - Customer can edit profile details.
 - Customer can send open requests to all the professionals in a particular service.
 - Customer can accept or reject the bid requests sent by the professionals.
- **Professional**
 - Professional can edit profile details and change the service if needed.
 - Professional can search for all the open requests based on address and pincode
 - Professional can send bid requests to the customer with an added service description.

Video Presentation

Video URL: https://drive.google.com/file/d/1QII3mGaHhFHiHvaHcqZeQREY_TnNzegJ/view?usp=sharing