HOME GENIE

HOUSEHOLD SERVICES APPLICATION - V2

Final Project Report

BY PUNIT KUMAR

23F1002051 | Modern Application Development-2 | Sept, 2024 Term

Author

Name – Punit Kumar

Roll No. – 23f1002051

Email – 23f1002051@ds.study.iitm.ac.in

About me - I'm currently pursuing BS in Data Science in IIT Madras. Combining a background in Mathematics & Computer Applications with a passion for coding, I'm adept at Python, web development frameworks, and machine learning.

Description of Project

The *Household Services Application - V2 named* 'HomeGenie' is a comprehensive, multi-user platform designed to bridge the gap between service professionals and customers, offering a seamless solution for managing household services. This application enables users to book, track, and manage various home services such as plumbing, AC repair, salon services and so on, all through a user-friendly interface.

Technologies Used

Backend

- **Flask** – Backend Framework for building the web application.
- **SQLite** – Database Management System for storing application data.
- **Redis** – A fast, in-memory key-value store used for caching and improving API performance.
- **Celery**: A distributed task queue for managing background jobs like scheduled reminders and CSV exports.

Frontend

- **VueJS** – A progressive JavaScript framework for building responsive, interactive, and dynamic user interfaces.
- **HTML/CSS** – Frontend Technologies for user interface design and interactivity.

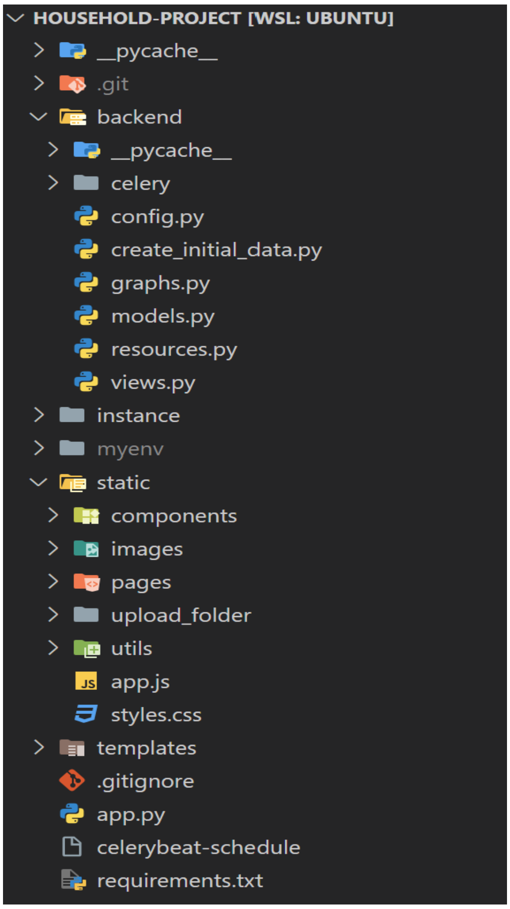
Other Tools and Libraries

- **Flask-Excel** – For creating CSV files
- **Datetime** – Python library for handling date and time operations.

- **MailHog -** An email testing tool used to capture and inspect outgoing emails during development.
- **Matplotlib** – Python library for creating different types of chart.

Architecture

Here, the main.py file contains the main code to run the webapp (HomeGenie – *Household Services Application*) is a module which contains all the files for app. The description of files inside Household Project are as follows:



```
HOUSEHOLD-PROJECT [WSL: UBUNTU]
  > __pycache__
  > .git
  v backend
    > __pycache__
    > celery
      config.py
      create_initial_data.py
      graphs.py
      models.py
      resources.py
      views.py
  > instance
  > myenv
  v static
    > components
    > images
    > pages
    > upload_folder
    > utils
      app.js
      styles.css
  > templates
    .gitignore
    app.py
    celerybeat-schedule
    requirements.txt
```

Features

1. Admin Login and Professional and User Login (RBAC)
    - A login/register form with fields like username, password etc. for professional, user and admin login.
    - The app have a suitable model to store and differentiate all the types of users of the app..

2. Admin Dashboard
    - Admin login redirects to Admin Dashboard.
    - Admin can manage all the users (customers/service professional).
    - Admin can approve a service professional after verification of profile docs
    - Admin has capability of flag/unflag the unappropriated customers and service professionals.

3. Service Management – for the Admin
    - Create new Service with base price
    - Update an existing service - e.g. name, price, time_required and description
    - Delete an existing service

4. Service Request – For the Customers
    - Create a new service request based on the services available.
    - Edit an existing service request - e.g. message etc.
    - Close an existing service request.

5. Search for available services
    - The customers should be able to search for available services based on their location, name, pin code etc.
    - The admin should be able to search for a professional to block/unblock/review them.

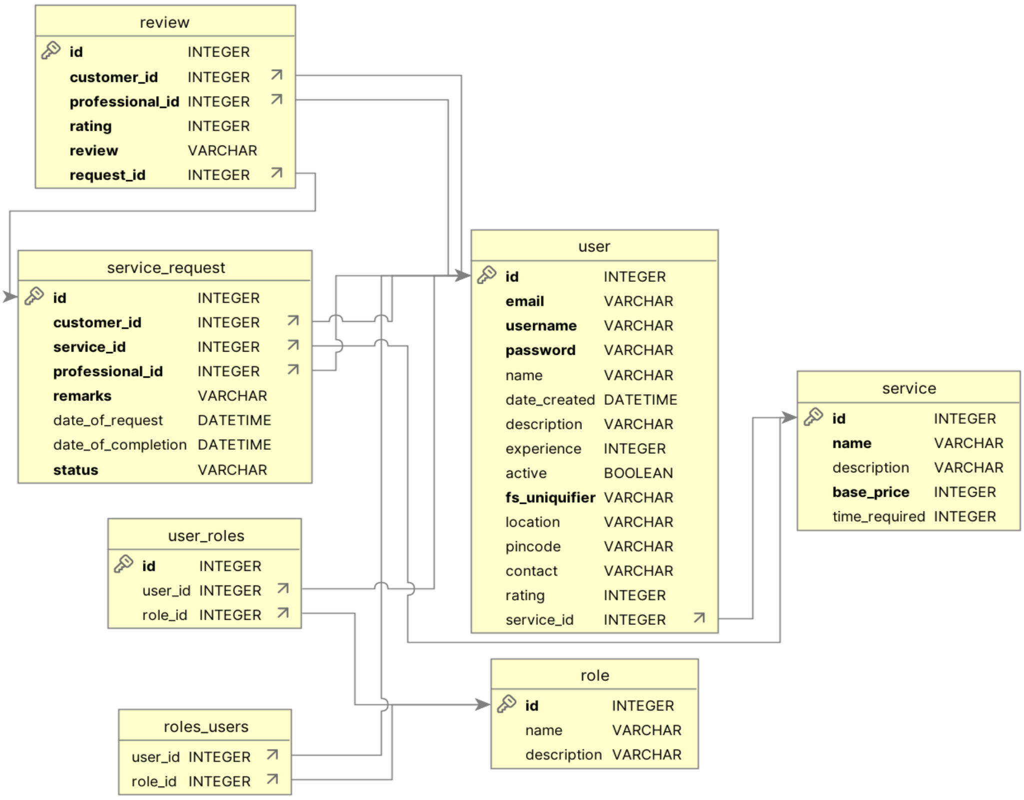
6. Take action on a particular service request - for the service professional
    - Ability to view all the service requests from all the customers
    - Ability to accept/reject a particular service request

7. **Backend Jobs**
    - Daily Reminders - The application can send daily reminders to service professionals on mail
    - Monthly Activity Report – Monthly activity report is send to all customers on first day of every month.
    - Export CSV – Admin can download all the service request in csv format from the dashboard.

DB Schema Design

The database schema for *HomeGenie* is structured to effectively support the multi-user application’s functionalities, including user management, service requests, and role-specific operations. It is designed using **SQLite** as the database backend, ensuring simplicity and reliability for the platform.

The schema consists of interconnected tables that define the relationships between **Admin**, **Service Professionals**, **Customers**, **Services** and **ServiceRequests**. It also accounts for the dynamic nature of service requests and supports caching and background tasks.



review		
id	INTEGER	
customer_id	INTEGER	↗
professional_id	INTEGER	↗
rating	INTEGER	
review	VARCHAR	
request_id	INTEGER	↗

service_request		
id	INTEGER	
customer_id	INTEGER	↗
service_id	INTEGER	↗
professional_id	INTEGER	↗
remarks	VARCHAR	
date_of_request	DATETIME	
date_of_completion	DATETIME	
status	VARCHAR	

user		
id	INTEGER	
email	VARCHAR	
username	VARCHAR	
password	VARCHAR	
name	VARCHAR	
date_created	DATETIME	
description	VARCHAR	
experience	INTEGER	
active	BOOLEAN	
fs_uniquifier	VARCHAR	
location	VARCHAR	
pincode	VARCHAR	
contact	VARCHAR	
rating	INTEGER	
service_id	INTEGER	↗

service	
id	INTEGER
name	VARCHAR
description	VARCHAR
base_price	INTEGER
time_required	INTEGER

user_roles		
id	INTEGER	
user_id	INTEGER	↗
role_id	INTEGER	↗

role	
id	INTEGER
name	VARCHAR
description	VARCHAR

roles_users		
user_id	INTEGER	↗
role_id	INTEGER	↗

Video Link:

https://drive.google.com/file/d/14AYN4wehn1_LG1wL_uY_uDLu9q7aIUoq/view?usp=drive_link

Thank You