HOUSEHOLD SERVICE

Student name: Shouvik Roy
Student email id : 23f1001348@ds.study.iitm.ac.in

SUMMARY: 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.

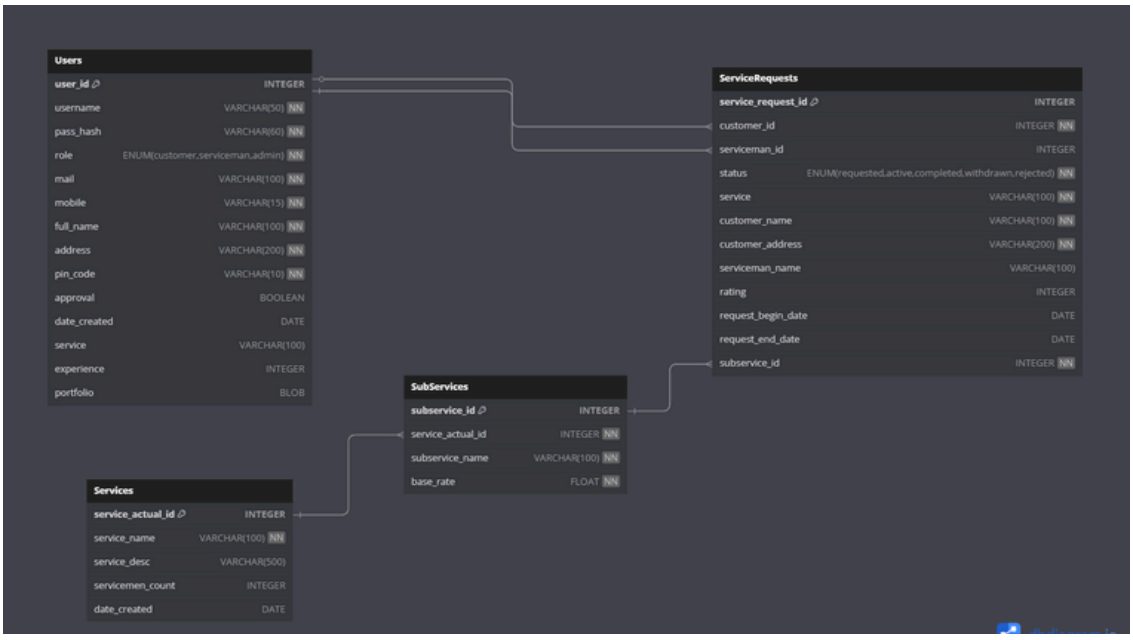
MATERIALS I HAVE USED:

- **Frontend:** CSS for styling , VUE and HTML for structure
- **Backend:** flask,flask_sqlalchemy, redis, celery
- **Database:** Sqlite3 for storing the data of the data models

DATASTRUCTURE:

Data Models:

- **Usermodel:**Stores all the data of users with specificroles: Admin, Customers, Service professional
- **Servicemodel:**Stores all the data related to service such as price, service, description, assigned professioanal
- **ServiceRequestmodel:**Stores all the data of service requests made by customers on which date, to which professional, for which specific service
- **SubServiceRequestmodel**: Stores all the sub services under the main service. name and price of sub services



CORE FUNCTIONALITIES:

- **Login System**:
  - Separatelogin/registerformsforadmin,customer,andservice professional.
  - Storesanddifferentiatesusertypesintheapp'sdatabase.
- **Admin Dashboard**:
  - Managesallusers(approveorblockcustomersandprofessionals).
  - Approvesprofessionalsafterprofileverification.
  - Blocksusersbasedonfraudulentactivityorpoorreviews.
- **Service Management (Admin)**:
  - Add,update,ordeleteservices(includesbaseprice,timerequired,etc.).
- **Service Requests (Customer)**:
  - Create,edit,andcloseservicerequestsbasedonavailableservices.
- **Service Search**:
  - Customerscansearchforservicesbylocation,name,orpincode.
  - Admincansearchforprofessionalstoreview,block,orunblock.
- **Service Request Actions (Professional)**:
  - Viewallservicerequests.
  - Accept/rejectandcloseservicerequests.

- **Scheduled Jobs:**
  1. Daily reminders for professionals (Google Chat Webhooks/SMS/Email)
  2. Check pending service requests & send alerts
  3. Reminders sent every evening

- **Monthly Report:**
  4. HTML-based activity report for customers
  5. Service details, requests, closures
  6. Sent via email on the 1st of every month

- **User-Triggered Job:**
  7. Export closed service requests as CSV
  8. Includes service_id, customer_id, professional_id, date, remarks
  9. Admin dashboard trigger
  10. Batch job execution & completion alert

- **Performance & Caching:**
  11. Implement caching for faster API responses
  12. Set cache expiry for data freshness
  13. Optimize API performance

DEVELOPMENT APPROACH:

Initial Startup:

- Created the requirements.txt file, framed the directory according to the given structure, finally implemented all the codes JS, HTML, PYTHON and HTML

Feature Development:

- Created the separate js file for routes.
- Created role based entery in backend using python.

Code testing:

Tested all the routes, buttons, and backend functionalities.

LEARNINGS:

Understanding Multi-User Application Structure:

- Gained knowledge of organizing and managing a multi-user app with distinct roles and access levels.

Exploring Backend and Frontend Functionalities:

- Learned about various backend and frontend features, enhancing the app's functionality.

Overcoming Challenges:

- Encountered and resolved numerous errors, deepening understanding of code syntax and practical application.

DRIVE LINK:

[CLICK HERE]