Household Services Application - Project Report

AUTHOR Name : AKASH MAURYA
Email ID : 22f3003152@ds.study.iitm.ac.in

1. Introduction

- **Project Title:** Household Services Application - V2
- **Project Summary:** This project aims to create a multi-user household services platform where users can either request or provide household services. Three distinct user roles—Admin, Service Professional, and Customer—enable a streamlined process for service management.
- **Objectives:**
    - To facilitate an efficient booking and management system for household services.
    - To implement a role-based access system for secure and personalized experiences.

2. Technology Stack

- **Backend:** Flask, Flask-SQLAlchemy, Flask-Security for handling user authentication and role-based access control.
- **Frontend:** Vue.js for the user interface, Bootstrap for styling.
- **Database:** SQLite for storing user data, service requests, and roles.
- **Additional Tools:** Redis and Celery for handling asynchronous tasks.
- **Platform:** Deployed on a Linux virtual machine with PowerShell for Windows.

3. System Architecture

- **Database Models:**
    - **User Model:** Stores user information along with their role (Admin, Professional, Customer).
    - **Service Model:** Contains service details such as type, description, and available professionals.
    - **Role Model:** Defines the roles and access privileges.
- **Role-Based Access Control (RBAC):** Implements access restrictions based on roles:
    - **Admin:** Manages services and user accounts.
    - **Service Professional:** Can view and manage their accepted service requests.
    - **Customer:** Can request services, view order status, and manage their bookings.

4. Features and Functionalities

- **User Authentication and Role-Based Access:**
    - Built using Flask-Security and SQLAlchemy, users are directed to role-specific dashboards upon login.

    - Different dashboards for each role include functionalities tailored to their respective needs.
- **Service Request System:**
    - Customers can book services, and Service Professionals can accept or reject requests.
    - Admins can manage available services and user accounts.
- **Dynamic Signup Fields:**
    - The signup page includes role-specific fields, dynamically displayed based on the user's selected role.
- **Error Handling and Validation:**
    - Ensures users receive appropriate feedback on failed actions and error messages are logged for debugging.

5. Development Process

- **Initial Setup:** Created a virtual environment, installed dependencies, and set up Flask, Vue.js, celery, redis and database configurations.
- **Feature Implementation:**
    - Developed backend routes and integrated Vue.js components.
    - Implemented dynamic field addition and role-based access control.
- **Testing:** Ensured correct role-specific access and conducted functionality tests for all user roles.

6. Challenges and Solutions

- **Role-Based Dashboard Routing:** Faced initial issues with Vue.js routing after login. Solved by updating the `login.js` file to redirect based on roles.
- **Database Schema Errors:** Addressed by recreating the database after identifying schema conflicts.
- **Improving Security:** Integrated Flask-CORS for secure cross-origin requests and restricted access to role-specific pages.

7. Conclusion

- **Outcomes:** This project successfully creates an interactive platform for household services, providing specific functionality for admins, professionals, and customers.
- **Future Work:** Potential enhancements include integrating payment processing and adding a review system for professionals.

8. Presentation

- **Youtube:** https://youtu.be/w7FYLXgQptI
- **Drive:https://drive.google.com/file/d/1hPOiBQwko9aCblQB4fT62ht-ac07PON5/view?usp=sharing**