Household Management MAD II – project report

Author:

Name: Aman Yadav

Roll no: 21f3003230

Email: 21f3003230@ds.study.iitm.ac.in

Description:

Household manage app is a multi-user web application, where customer can book the services and professional can provide the services. Customer can also close the service and give the rating and review to the professional for that service one completed. It also contains admin functionality, who can perform CRUD operation on the services and manage users.

Technologies Used:

- Flask: Backend framework for routing and API management.
- Flask-SQLAlchemy: ORM for handling database interactions.
- SQLite3: Lightweight relational database.
- Redis: Caching and task queue backend.
- Celery: For asynchronous tasks like reminders and reports.
- JWT: Token-based user authentication.
- Werkzeug: Secure password hashing and validation.
- Vue.js: Frontend framework for building responsive interfaces.
- Chart.js: Visualizes data in interactive charts.
- Bootstrap: Frontend templates and styles.
- pytz: Timezone handling (IST).

Database Schema:

Users: Stores user details such as name, role, email, and hashed password.

ServiceTypes: Defines categories of services (e.g., plumbing, cleaning).

Services: Tracks specific services with descriptions and pricing.

ServiceProfessionals: Details professionals, their expertise, and associated services.

Customers: Captures customer contact and address details.

ServiceRequests: Records service requests, their status, ratings, and reviews.

API Design:

Here are some of the important APIs for your project:

1. POST /login: Authenticates user and returns a token.
2. GET /services_: Retrieves all services, optionally filtered by service_type_id.
3. GET /service/int:service_id/professionals: Lists professionals for a specific service.
4. POST /service_request: Books a service.
5. GET /customer/service_requests: Lists all service requests for the customer.
6. PUT /customer/service_requests/int:request_id: Updates service request status.
7. GET /professional/dashboard: Retrieves professional dashboard.
8. GET /service_requests_for_professionals: Lists service requests for professionals.
9. POST /admin/create_new_service: Creates a new service.
10. GET /admin_all_services_requests: Lists all service requests.

Features Implemented:

1. User Authentication: Users can log in and log out, receiving a token upon successful authentication.
2. Service Retrieval: Customers can retrieve all available services, optionally filtered by service type.
3. Professional Listing: Customers can list all available professionals for a specific service.
4. Service Booking: Customers can book a service with a professional.
5. Service Request Management: Customers can view, update, and delete their service requests.
6. Professional Dashboard: Professionals can view their dashboard and manage service requests.
7. Admin Management: Admins can create, edit, and delete services and service types, and view all service requests.

Presentation Video :Click here to view