

Project Report

Course: Modern Application Development II

Term: September 2024

Student Details

Name: Gurdeep

Roll Number: 22F3001660

Email: 22f3001660@ds.study.iitm.ac.in

Project Title: HomeHub

Objective:

The project aims to develop a household services platform that seamlessly connects customers with skilled professionals for daily tasks such as cleaning, plumbing, pest control, and more. The platform resolves challenges like finding reliable help, ensuring trustworthiness, and negotiating terms by providing a one-stop solution. Key highlights include verified professionals, secure transactions, and streamlined service management.

Technology Stack Backend:

- **Flask:** Framework for API development, role-based access control (RBAC), and application logic.
- **SQLite:** Database management for storing structured data.
- **Redis:** For caching frequently used data and managing background jobs.
- **Celery:** For task scheduling, reminders, and batch job management.

Frontend:

- **Vue.js:** Frontend framework for dynamic user interfaces.
- **Bootstrap:** For styling and responsive design.
- **animate.css:** For smooth UI animations.

Other Tools & Techniques:

- Password hashing for security.
- JWT-based RBAC for secure endpoint access.
- SQLAlchemy ORM for database operations.

Key Functionalities

1. Role-Based Features:

- **Admin:** Manages users, services, and professionals; authorizes service providers.
- **Customer:** Requests and tracks services, leaves ratings and feedback.
- **Professional:** Accepts or rejects service requests, provides updates on completion.

2. Service Management:

- Create, retrieve, update, and delete services.
- Ensures real-time data consistency through caching and background jobs.

3. Service Request Management:

- Customers can create requests for services.
- Professionals can accept or reject requests based on availability. ○ Requests can be tracked until completion, with ratings and remarks added at the end.

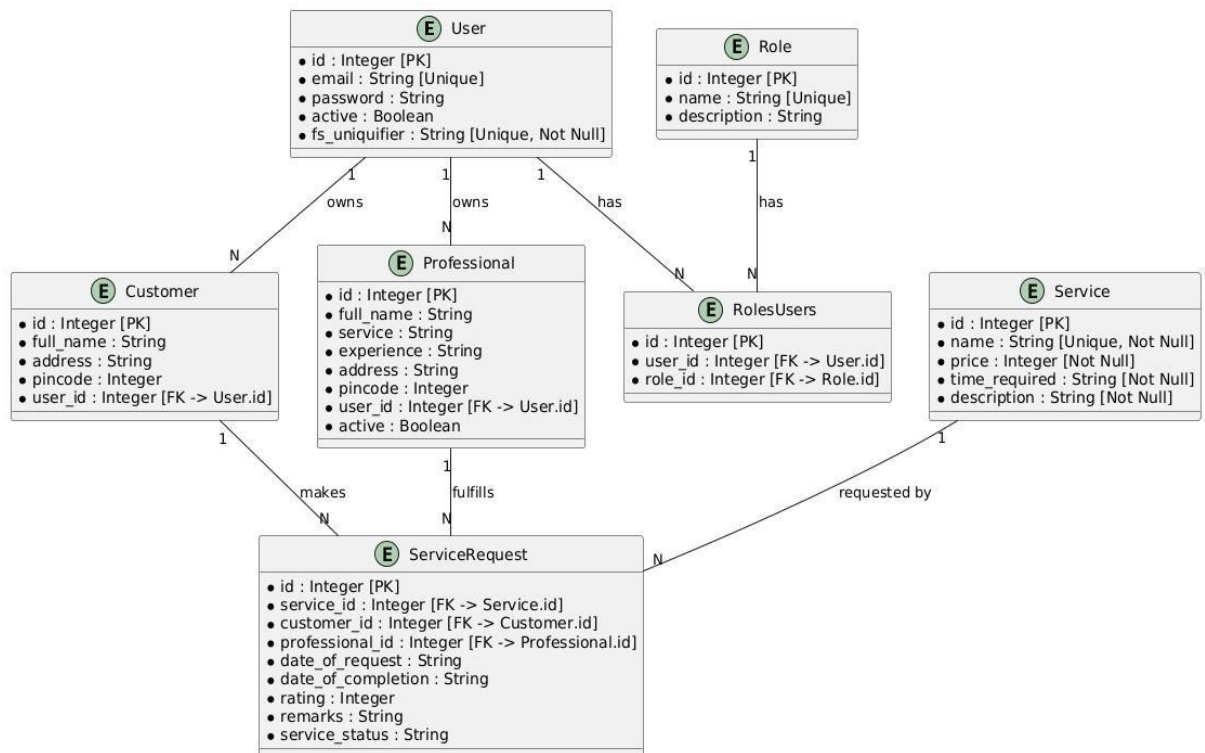
4. Enhanced User Experience:

- Dynamic navigation with Vue.js.
- Frequent caching of data for faster response times.

Database Design (ER Diagram) Key

Tables:

- **User:** Stores user details and credentials.
- **Role:** Defines roles such as Admin, Customer, and Professional.
- **Roles_Users:** Links users to their respective roles.
- **Service:** Lists available services (e.g., plumbing, pest control).
- **Customer:** Stores customer profiles.
- **Professional:** Maintains professional profiles.
- **Service_Request:** Tracks customer requests and professional assignments.



API Resource Endpoints

All endpoints are built using the Flask-RESTful framework and secured with JWT-based authentication.

1. Service Management

- **GET /api/services:** Retrieve all services (open to all authenticated users).
- **POST /api/services:** Add a new service (Admin only).
- **GET /api/update/service/<id>:** Retrieve specific service details for editing (Admin only).
- **POST /api/update/service/<id>:** Update specific service details (Admin only).

2. User Management

- **GET /api/customers:** Retrieve all customers (Admin only).
- **POST /api/customers:** Register a new customer.
- **GET /api/professionals:** Retrieve all professionals (Admin only).
- **POST /api/professionals:** Register a new professional.

3. Service Request Management

- **GET /api/request/service:** Retrieve all service requests (authenticated users).
- **POST /api/request/service:** Create a new service request (Customers).
- **GET /api/accept/service-request/<id>:** Reject a service request (Professionals).

- **POST /api/accept/service-request/<id>:** Accept a service request (Professionals).
 - **POST /api/service-request/customer:** Retrieve all requests made by a specific customer.
 - **POST /api/close/service-request/<id>:** Close a service request with a rating and remarks.
-

Technical Features

- **Role-Based Access Control (RBAC):** Secure routing and endpoint control using JWT.
 - **Caching:** Frequently accessed data cached via Redis for faster retrieval.
 - **Password Security:** Passwords are hashed for data privacy.
 - **Task Scheduling:** Batch jobs and reminders implemented with Celery.
 - **Optimized Frontend:** Dynamic and responsive UI powered by Vue.js.
-

Presentation Video

Access the detailed project demonstration video here: [**HomeHub Presentation Video**](#)

Or use the following link: -

https://drive.google.com/file/d/1UusZvtOt2CR9n9d4_yzDEeOaL6jLgrOS/view?usp=sharing