‘HOUSEHOLD SERVICES APPLICATION’

PROJECT REPORT

BY

SATYAM SAURABH
Roll: 22f3001024

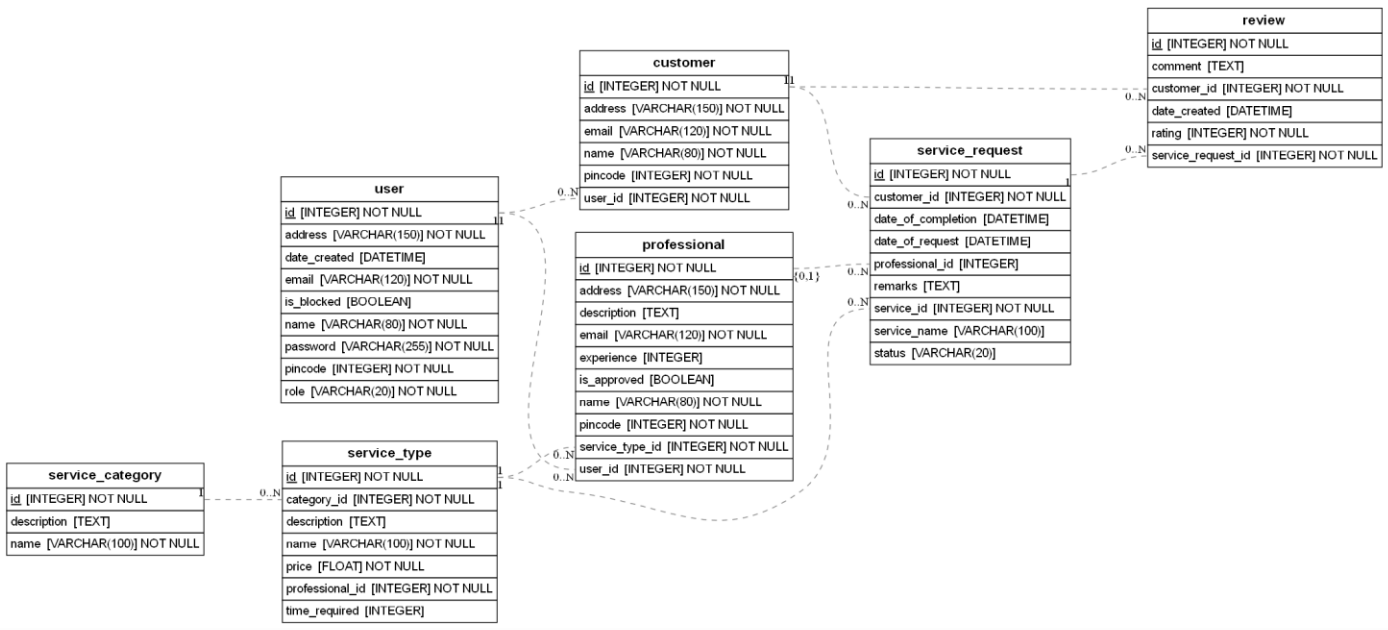
Introduction

The "Household Services Application" is a multi-user platform designed to connect customers with service professionals for comprehensive home servicing and solutions. The application supports three user roles: Admin, Service Professional, and Customer. Admins have oversight over users, services, and service requests. Service Professionals can accept or reject service requests, while Customers can create service requests and review completed services.

Frameworks and Libraries Used

- Flask: Used for request handling, rendering templates, and defining views/routes in the application.
- Flask-SQLAlchemy: Utilized for defining models, performing database query operations, and committing changes to the SQLite database.
- Jinja2: Employed for templating, providing greater flexibility to the HTML documents used in the platform.
- SQLite: Chosen as the database for storing user information, services, service requests, and other related data.
- Bootstrap: Used for HTML generation and styling to create a responsive and visually appealing user interface.

DB Schema Design


review
id [INTEGER] NOT NULL
comment [TEXT]
customer_id [INTEGER] NOT NULL
date_created [DATETIME]
rating [INTEGER] NOT NULL
service_request_id [INTEGER] NOT NULL
customer
id [INTEGER] NOT NULL
address [VARCHAR(150)] NOT NULL
email [VARCHAR(120)] NOT NULL
name [VARCHAR(80)] NOT NULL
pincode [INTEGER] NOT NULL
user_id [INTEGER] NOT NULL
service_request
id [INTEGER] NOT NULL
customer_id [INTEGER] NOT NULL
date_of_completion [DATETIME]
date_of_request [DATETIME]
professional_id [INTEGER]
remarks [TEXT]
service_id [INTEGER] NOT NULL
service_name [VARCHAR(100)]
status [VARCHAR(20)]
user
id [INTEGER] NOT NULL
address [VARCHAR(150)] NOT NULL
date_created [DATETIME]
email [VARCHAR(120)] NOT NULL
is_blocked [BOOLEAN]
name [VARCHAR(80)] NOT NULL
password [VARCHAR(255)] NOT NULL
pincode [INTEGER] NOT NULL
role [VARCHAR(20)] NOT NULL
professional
id [INTEGER] NOT NULL
address [VARCHAR(150)] NOT NULL
description [TEXT]
email [VARCHAR(120)] NOT NULL
experience [INTEGER]
is_approved [BOOLEAN]
name [VARCHAR(80)] NOT NULL
pincode [INTEGER] NOT NULL
service_type_id [INTEGER] NOT NULL
user_id [INTEGER] NOT NULL
service_category
id [INTEGER] NOT NULL
description [TEXT]
name [VARCHAR(100)] NOT NULL
service_type
id [INTEGER] NOT NULL
category_id [INTEGER] NOT NULL
description [TEXT]
name [VARCHAR(100)] NOT NULL
price [FLOAT] NOT NULL
professional_id [INTEGER] NOT NULL
time_required [INTEGER]
0..N
1
{0,1}

The database schema for the Household Services Application includes several key tables:

1. **Users**: Stores general user information such as user_id, username, password, and role (admin, service professional, customer).
2. **Professionals**: Linked to the Users table via user_id, contains fields like professional_id, name, description, service_type, experience, and date_created.
3. **Customers:** Also linked to the Users table, includes customer_id, name, and other relevant information.
4. **Service_type:** Stores service-related information, including service_id, name, price, time_required, and description.
5. **Service_category:** Stores service type category, and description about the related service type.
6. **Service_requests**: Linked to services, customers, and service professionals, includes fields like request_id, service_id, customer_id, professional_id, date_of_request, date_of_completion, service_status, and remarks.
7. **Review**: Linked to service_request which stores reviews and remarks about the service_request along with the customer_id and date_posted.

These tables are interconnected to support functionalities like service management, service request handling, and role-based access.

Architecture and Features

The project structure includes:

- app.py: Contains the main application code, controllers, and database configurations.
- templates/: Folder containing HTML templates served by the application.
- static/: Folder containing CSS stylesheets and other static assets.

Features:

1. Login and Registration
    - Separate forms for admin, service professional, and customer login.
    - User registration based on roles.
2. Role-Based Access Control
    - Admins, service professionals, and customers have different access levels and permissions.
3. Admin Features
    - User and Service Monitoring: View all users, services, and service requests.
    - Service Management: Create, update, and delete services.
    - User Management: Approve service professionals, block/unblock users.
    - Statistics and Reporting: Display statistics on active users, services, and service requests.
4. Service Professional Features
    - View Service Requests: Access all service requests assigned to them.
    - Accept/Reject Requests: Ability to accept or reject service requests.
    - Profile Management: Update professional profile information.

5. Customer Features
    - Create Service Requests: Customers can create new service requests.
    - Search Services: Search for available services by name, location, or pin code.
    - Review Services: Post reviews and remarks on completed services.
6. Service Request Management
    - Create, edit, and close service requests.
    - Track service request status (requested/assigned/closed).
7. Search Functionality
    - Customers can search for services.
    - Admins can search for services and booking in platform.

Implementation Details

1. User Authentication: Implemented using Flask's session management and custom login forms for each user type.
2. Database Operations: Utilized Flask-SQLAlchemy for CRUD operations on services, users, and service requests.
3. Template Rendering: Jinja2 templates are used in combination with Bootstrap for creating responsive and dynamic web pages.
4. Form Validation: Implemented both frontend (HTML5 and JavaScript) and backend (Flask-WTF) form validation for data integrity.

Conclusion

The Household Services Application provides a comprehensive platform for connecting customers with service professionals, streamlining the process of home service management. Through its role-based access system and intuitive interface, it offers a user-friendly experience for all stakeholders involved in the home service ecosystem.

PROJECT DEMO VIDEO : https://youtu.be/oM2jEDn0Gjg?si=E3gx2QgTnoC17RCc