Scarlett - Household Services Web Application

Project Report

Student Details

- Name: Dabgar Milav Jayeshkumar
- Roll Number: 21f1005510

Project Overview

Scarlett is a comprehensive web application designed to streamline the process of booking and managing household services. The platform connects customers with verified service professionals while providing administrative oversight for quality control and service management.

Problem Statement and Approach

Problem Statement

The project aims to create a secure and efficient platform for household services that addresses:

- The need for verified and reliable service professionals
- Streamlined service booking and management
- Quality control through reviews and ratings
- Administrative oversight of service delivery

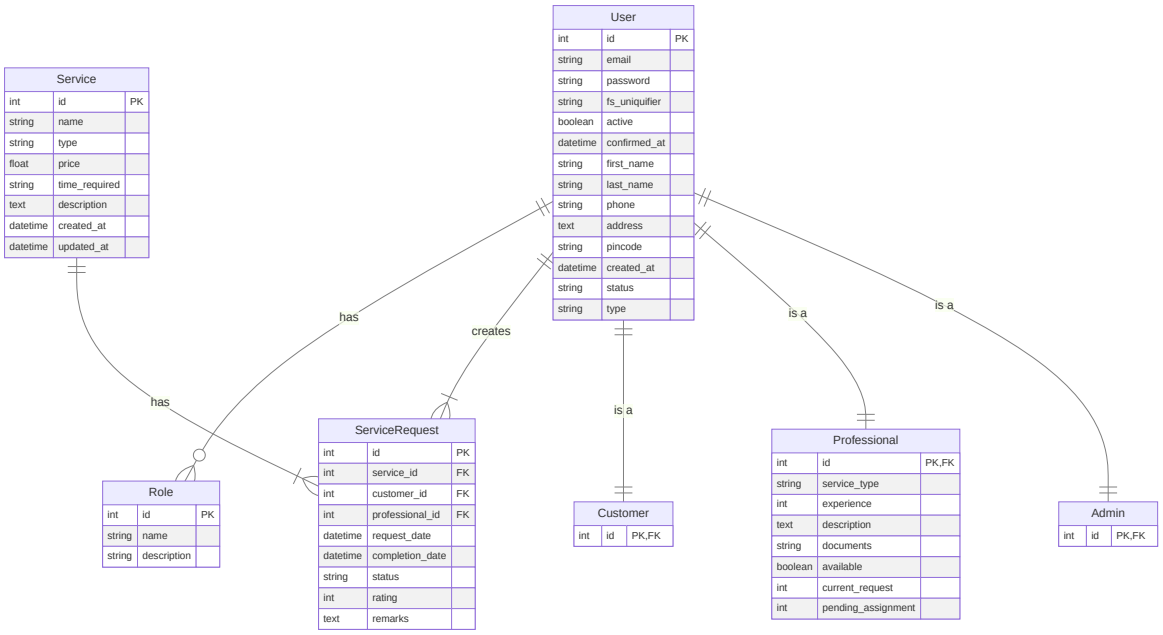
Implementation Approach

1. **User Role System**:
    - Implemented three distinct user roles (Customer, Professional, Admin)
    - Each role has specialized interfaces and permissions
    - Flask-Security integration for robust authentication and authorization
2. **Service Management**:
    - Categorized service listings
    - Dynamic pricing system
    - Professional assignment algorithm
    - Service request lifecycle management
3. **Quality Control**:
    - Professional verification system
    - Customer review and rating system
    - Administrative monitoring tools

Frameworks and Libraries Used

1. **Backend Framework**:
   - Flask - Web application framework
   - SQLAlchemy - Database ORM
   - Flask-Security - Authentication and authorization
   - Flask-WTF - Form handling and validation
2. **Database**:
   - SQLite (Development)
   - PostgreSQL (Production)
3. **Additional Libraries**:
   - UUID - Unique identifier generation
   - Werkzeug - File handling and security
   - ITSDangerous - Token generation for security features

Database ER Diagram



API Resource Endpoints

1. **Main System**:
   - `/` - Landing page
   - `/dashboard` - Role-based dashboard routing
2. **Authentication & Profile**:
   - `/register/customer` [GET, POST] - Customer registration
   - `/register/professional` [GET, POST] - Professional registration with document upload
   - `/profile` [GET, POST] - Profile management
   - `/profile/document` - Professional document access

3. **Customer System**:
    - `/customer/dashboard` - Service overview
    - `/customer/summary` - Activity summary
    - `/customer/service/<type>` - Category-based service browsing
    - `/customer/request/<request_id>/review` [GET, POST] - Service reviews
    - `/customer/request/<request_id>/update` [GET, POST] - Request management
4. **Professional System**:
    - `/professional/dashboard` - Work management interface
    - `/professional/summary` - Performance overview
    - `/professional/toggle-availability` [GET, POST] - Availability control
    - `/professional/request/<request_id>/reject` - Assignment management
    - `/professional/request/<request_id>/complete` - Service completion
5. **Administrative System**:
   Service Management:
    - `/admin/service/create` [GET, POST] - Service creation
    - `/admin/service/<id>/view` - Service monitoring
    - `/admin/service/<id>/edit` [GET, POST] - Service modification

   User Management:
    - `/admin/professional/<id>/document` - Verification system
    - `/admin/professional/<id>/approve` [POST] - Professional approval
    - `/admin/request/<request_id>/assign` [POST] - Request assignment
6. **Statistics API**:
    - `/api/admin/stats` - System statistics
    - `/api/professional/stats` - Performance metrics
    - `/api/customer/stats` - Usage analytics

Key Features

1. **User Management**:
    - Role-based access control
    - Professional verification system
    - Profile management
    - Document verification
2. **Service System**:
    - Categorized service catalog
    - Dynamic pricing
    - Service request lifecycle
    - Rating and review system
3. **Professional Features**:

- Availability management
- Request assignment
- Service completion tracking
- Performance metrics

4. **Administrative Tools**:
    - User verification
    - Service management
    - Request monitoring
    - System statistics
5. **Security Features**:
    - Flask-Security integration
    - Document verification
    - Token-based authentication
    - Role-based permissions

Project Presentation Link - https://drive.google.com/drive/folders/1_xaHllyFH6EAmz8A_HOrMXK4Mafv-aap