Household Services Management System - Project Report

Student Details

- Name: Ayushman Singh
- Roll Number: 23f1001477
- Course: Mad1

Project Overview

Problem Statement

This project addresses the need for a web platform that connects customers seeking household services with qualified professionals. The platform supports three distinct user roles: customers, service professionals, and administrators.

Technical Stack

- **Backend:**
  - Flask 3.1.0 (Web framework)
  - SQLAlchemy 2.0.36 (Database ORM)
  - Python-dotenv 1.0.1 (Environment management)
- **Frontend:**
  - Bootstrap 5.3.3
  - Chart.js (for analytics)
  - Jinja2 templating

Database Design

The database schema comprises three primary models:

1. **Customer Model:**
   - Stores basic user information (email, name, contact)
   - Maintains a history of service requests
   - Handles authentication details
2. **Professional Model:**
   - Manages professional profiles including their service package, experience, and status.

```
class Professional(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(120), unique=True)
    full_name = db.Column(db.String(120))
    service_package_id = db.Column(db.Integer, db.ForeignKey('service_package.id'))
    experience = db.Column(db.Integer)
    status = db.Column(db.String(50), default="pending")
```

1. **Service Models:**
    - Defines service categories and packages
    - Implements a request tracking system

Key Features

1. **Authentication System:**
    - Provides role-based access control (customer, professional, admin)
    - Includes a workflow for professional verification
2. **Service Management:**
    - Enables customers to request services based on available packages.

```
@customer_services.route('/request_service/<int:package_id>', methods=['POST'])
def request_service(package_id):
    service_request = ServiceRequest(
        customer_id=session['user_id'],
        service_package_id=package_id,
        status="requested"
    )
    db.session.add(service_request)
    db.session.commit()
```

1. **Dashboard Systems:**
    - Customer Dashboard: View and track service requests.
    - Professional Dashboard: Manage assigned service requests.
    - Admin Dashboard: Monitor system activity and access analytics.

Project Structure

```
Project Root/
├── Code/
│   ├── auth.py          # Authentication
│   ├── services.py      # Service management
│   ├── models.py        # Database models
│   ├── app.py           # Main application
│   └── templates/       # HTML templates
└── static/              # Assets
```

Conclusion

This project successfully delivers a secure and scalable platform for managing household services. It effectively connects customers with service providers while providing administrative oversight. The modular architecture ensures maintainability and facilitates future enhancements.

[End of Report]