

# **PROJECT REPORT**

## **Household Services Website**

### **Student Details:**

**Name:** Kunal Sahu

**Roll No.:** 23dp2000037

**Email ID:** 23dp2000037@ds.study.iitm.ac.in

**Course:** Modern Application Development 1

### **Project Presentation Video-**

[https://drive.google.com/file/d/1yPZC2rq6d7LMNOSKhp6A1apOFRK6x/view?usp=drive\\_link](https://drive.google.com/file/d/1yPZC2rq6d7LMNOSKhp6A1apOFRK6x/view?usp=drive_link)

### **Project Details:**

#### **Overview of the Project:**

The Household Services Application is a versatile, multi-user platform that links customers with skilled service providers, offering a broad array of home services. It features three tailored user roles: Admin, Customer, and Service Professional. Customers can browse and book a range of household services, while service professionals handle and complete these bookings. The admin role oversees platform operations, ensuring efficiency and smooth management across all activities. This well-structured design facilitates seamless interactions and efficient service delivery, creating a streamlined experience for every user.

#### **How I approached the problem statement?**

Firstly, I revisited the foundational concepts from MAD-1 theory and refreshed my understanding of Flask and Jinja2. Adopting a step-by-step approach, I focused on simplicity, functionality, and scalability throughout the implementation. Key priorities included establishing role-based access control, integrating CRUD functionalities, and designing a structured workflow for booking and completing services. The solution is optimized to run seamlessly on a local machine, with APIs in place to support future integrations. While the initial stages were challenging, the project ultimately concluded successfully.

### **Technologies Used:**

1. **Flask:** A lightweight web framework for building web applications in Python.
2. **Jinja2(via render\_template):** Template engine for rendering dynamic HTML content.
3. **SQLAlchemy:** An Object Relational Mapper (ORM) for interacting with databases in Python.

4. **Flask-SQLAlchemy:** An extension that integrates SQLAlchemy with Flask for easier database management.
5. **DateTime:** A module to work with dates and times in Python.
6. **SQLite:** Database management system for storing application data.
7. **HTML/CSS/JavaScript:** Frontend technologies for user interface design and interactivity.
8. **ChartJS:** User for creating different types of charts on the admin dashboard
9. **BootStrap:** A CSS framework for building web pages with pre-styled components.

## Database Schema Design:

