

## Project Report

### Author

**Name:** Aditya Rajkumar Jha

**Roll Number:** 23f2003863

**Email:** [23f2003863@ds.study.iitm.ac.in](mailto:23f2003863@ds.study.iitm.ac.in)

I am currently at *Diploma level TERM 1 in BS Degree in Data Science and Applications* at IIT Madras.

### About the Project

The objective of this project was to develop a *Household Services Web Application* using Flask as the backend framework. The application is conceptually inspired by platforms like Urban Company but distinguishes itself in terms of functionality and business model.

### Technologies Used

#### 1. Flask

Used for request handling, rendering templates, and defining views/routes for the application.

#### 2. Flask-SQLAlchemy

Utilized for defining database models, executing query operations, and committing changes to the database.

#### 3. Jinja

Implemented for templating to provide flexibility in generating dynamic HTML documents.

#### 4. SQLite

Chosen for database management due to its lightweight and efficient nature.

#### 5. Frontend Technologies

**HTML:** For structuring the application.

**CSS (Vanilla and Bootstrap):** For aesthetic styling and responsive design.

**JavaScript:** Used minimally for client-side validation.

#### 6. Python Libraries

**NumPy, Pandas:** For data handling and manipulation.

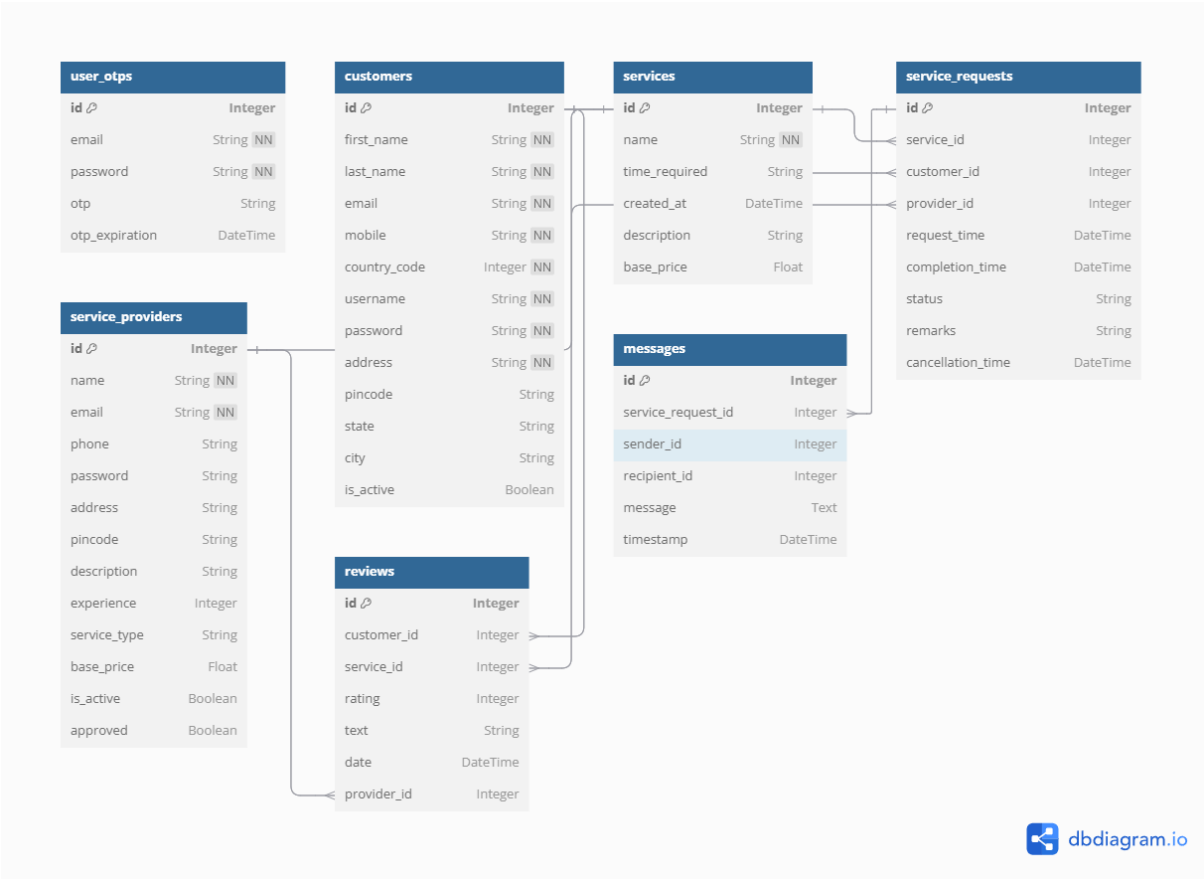
**Matplotlib, Seaborn:** For generating analytical reports and visualizations.

#### 7. SMTP Services

Integrated for enabling password recovery functionality for all Users

This project demonstrates my ability to apply core concepts of data science and web application development to solve practical problems. It also highlights my understanding of modern tools and technologies essential for creating efficient and user-friendly applications.

This is my ER diagram:



Architecture:

The architecture of this household services application is based on the **Model-View-Controller (MVC)** pattern

- Model:** Handles data representation and database interactions (e.g., Customer, Service, ServiceProviders, Service Request).
- View:** The user interface using **HTML, CSS, and Jinja2 templates**.
- Controller:** Manages the application logic, routes, and request handling in Flask

Components

Frontend:

HTML, CSS (Bootstrap for styling), JavaScript (for client-side interactivity).

Backend:

- Flask for routing and application logic.
- Flask-SQLAlchemy for ORM.
- SMTP for email handling

Database:

SQLite (lightweight for local deployment)

Features for the Household Services Application

Customer Features:

**Account Management:**

- Customer registration and login functionality.
- Password recovery via email.

**Browse Services:**

View all services and their details.

Filter and search for services by name or pincode.

**Service Requests:**

Book services from available service providers.

Cancel or close service requests.

View status of requests.

Chat with service provider

**Feedback and Reviews:**

Submit reviews and ratings for services received.

**Service Professional Features:**

**Account Management:**

Separate login page for service providers.

Profile setup, including details like expertise, services offered, and experience.

**Service Handling:**

View and manage assigned service requests.

Accept or Reject Service requests.

chat with customer for details.

**Profile Reviews:**

View customer feedback and ratings.

**Admin Features:**

**User Management:**

Approve or block users (customers/service providers).

View and verify service provider profiles.

**Service Management**

Create, update, or delete existing services.

Set base pricing and time requirements for services.

**Monitoring:**

Oversee all service requests and user activities.

Take action in case of disputes or fraudulent behavior.

Video Link:- [https://drive.google.com/file/d/13l1uEgZUcUemt-NxC4Z8KLOu5vVfohc\\_/view?usp=sharing](https://drive.google.com/file/d/13l1uEgZUcUemt-NxC4Z8KLOu5vVfohc_/view?usp=sharing)