PROJECT REPORT

TASKMASTER HOUSEHOLD SERVICES

AUTHOR

Name: Aqdas Khan

Roll No: 23f2001156

Email: 23f2001156@ds.study.iitm.ac.in

Hello I'm a Diploma Level Student, I am passionate about coding and enjoy building creative and efficient solutions to real-world problems. Programming excites me because it allows me to turn ideas into reality, and I constantly strive to learn new technologies and improve my skills. I thrive on challenges and take pride in writing clean, functional code that makes a difference

PROJECT DETAILS

The TaskMaster **Household Services App** is a web application designed to connect customers with professionals for various household tasks. Built using **Flask**, **HTML**, **CSS**, and **SQLAlchemy**, the app offers a seamless platform for browsing services, booking appointments, and managing schedules. It features user authentication, professional profiles, and service categorization for easy navigation. The backend ensures efficient data management with robust database relationships. This app simplifies the process of finding reliable household services, making life more convenient for users.

TECHNOLOGIES USED

1. **Flask**: A lightweight Python web framework used for building the app's backend and routing.
2. **HTML**: Used for structuring the web pages and creating the app's user interface.
3. **CSS**: Styled the application, ensuring a responsive and visually appealing design.
4. **Bootstrap**: A front-end framework used to create responsive and mobile-friendly web pages quickly.
5. **SQLAlchemy**: An ORM tool for managing the database and establishing relationships between models.
6. **SQLite**: Used for storing and managing application data efficiently.
7. Jinja2: Templating engine for rendering dynamic content.

DATABASE ARCHITECTURE

1. **Costumer**:
    - Attributes: id(primary key), email, full_name, password, pincode, Address, role, is_blocked.
    - Relationships: One-to-Many with Service_request.
2. **Professional**:
    - Attributes: id(primary key), email, full_name, password, service, experience, pincode, Description, role, is_approved, is_rejected, is_active.
    - Relationships: One-to-Many with Service_request.
3. **Service**:
    - Attributes: id(primary key), name, price, time_required, Description.
    - Relationships: One-to-Many with Service_request.
4. **Service_request**:
    - Attributes: id(primary key), service_id, costumer_id, professional_id, date_of_request, preferred_date, date_of_completion, status, remarks, rating, review.
    - Relationships: Many-to-One with Customer, Professional, and Service.

EXPLANATORY VIDEO: VIDEO LINK