UrbanClap: HouseHold Services

**Project Title-UrbanClap**

**Project Report by-**

**23f2003200 Ish Sachdeva**

**Modern Application Development – II**

**January, 2024 Term**

# Student details

***Name****-* Ish Sachdeva

***Roll Number****- 23f2003200*

***Email****- 23f2003200@ds.study.iitm.ac.in*

*About Me -* I am Ish Sachdeva in Diploma Level and this is my

Modern Application two Development Project

# Project Video → [Project](https://drive.google.com/file/d/1KhZMCwuxkFXPS4DulUTSAni3RjQu_UvN/view?usp=sharing) [Presentation](https://drive.google.com/file/d/1d-s0HHHujZYtXy9-D3TeUleUoAoN0nqu/view?usp=sharing) [Link](https://drive.google.com/file/d/1d-s0HHHujZYtXy9-D3TeUleUoAoN0nqu/view?usp=sharing)

# Project Description

UrbanClap is a full-stack web application that connects service professionals with customers, providing a platform for booking and managing various home services.

# Core Features

1. User Management

* **Multi-role System**:
* Customers: Can book services and manage their requests
* Service Professionals: Can offer services and manage their work
* Administrators: Can manage the entire platform

2. Service Management

* Service listing with details (name, description, base price, time required)
* Service categorization
* Dynamic pricing
* Service availability tracking

3. Booking System

* Service request creation
* Real-time status tracking
* Payment integration
* Review and rating system

4. Location Management

* City-based service availability
* Pincode verification
* Service area management

5. Admin Dashboard

* User management (block/unblock users)
* Professional verification
* Service management
* City management
* Analytics and reporting
* Export functionality (CSV)

6. Automated Notifications

* Daily reminders for pending services
* Monthly reports
* Email notifications using Flask-Mail
* Celery for background task processing

# Technical Stack

Backend

* **Framework**: Flask (Python)
* **Database**: SQLite with SQLAlchemy ORM
* **Authentication**: JWT (JSON Web Tokens)
* **API**: RESTful architecture
* **Caching**: Redis
* **Task Queue**: Celery
* **Email**: Flask-Mail

Frontend

* **Framework**: Vue.js
* **Routing**: Vue Router
* **UI**: Bootstrap
* **State Management**: Local Storage
* **API Integration**: Axios

Security Features

* JWT-based authentication
* Password hashing
* Role-based access control
* CORS protection
* Input validation

Additional Features

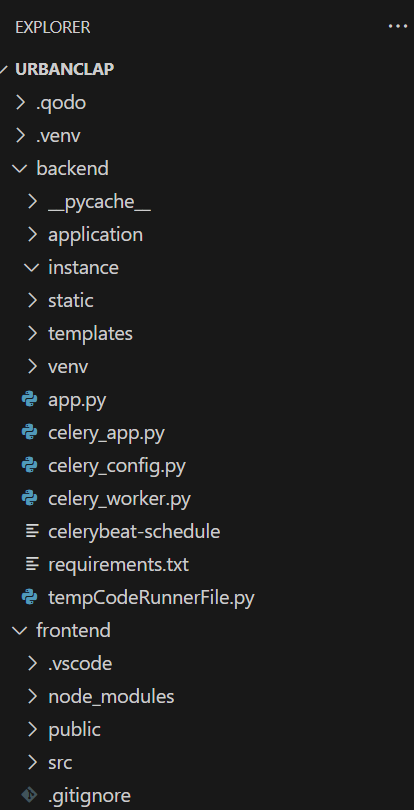
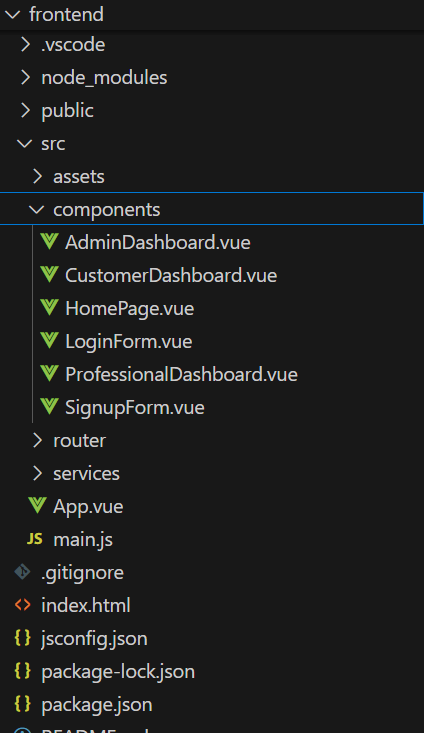
* Caching for improved performance
* Background task processing
* CSV export functionality
* Responsive design
* Real-time status updates

Architecture

* **MVC Pattern**: Model-View-Controller architecture
* **API-First Design**: RESTful API endpoints
* **Modular Structure**: Separate components for different functionalities
* **Scalable Design**: Easy to add new features and services

This project serves as a comprehensive service marketplace platform, allowing users to book various services while providing administrators with tools to manage the platform effectively.

***Folder Structure For the Project***



ER Diagram

