

PROJECT REPORT

Author

Name: Laksh Krishna Sharma

Roll Number: 23f3003262

Email: 23f3003262@ds.study.iitm.ac.in

I am a Computer Science enthusiast pursuing dual degrees—BS in Data Science from IIT Madras and B.Tech in Computer Science from Maharaja Agrasen Institute of Technology. My interests include building intelligent systems, machine learning, and quantitative finance. I actively contribute to open-source and work with technologies like MERN, Flask, FastAPI, ML, DL and Docker.

Description

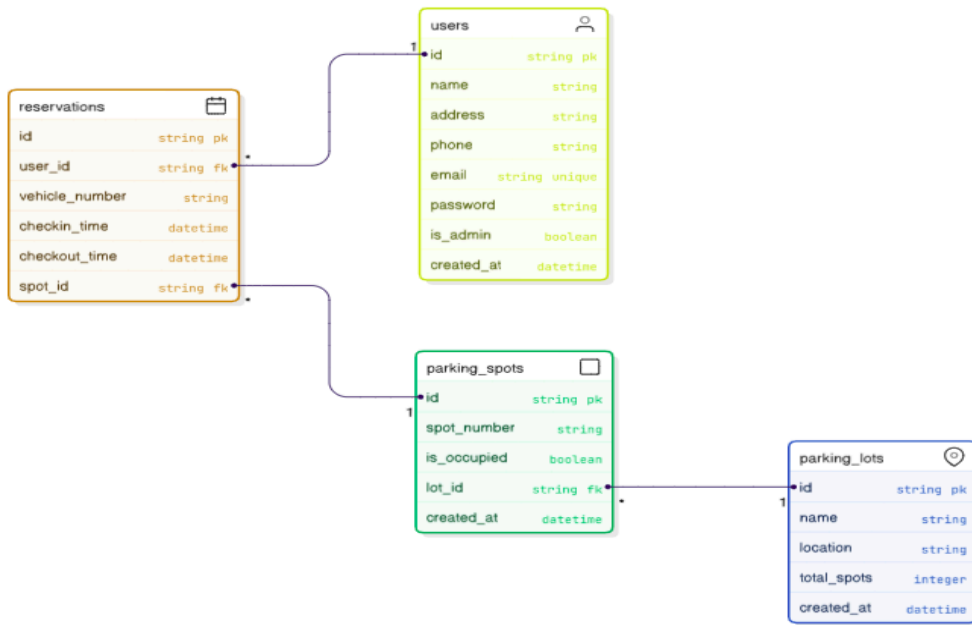
Parking Go is a web-based application developed using Flask, SQLite, and Bootstrap that enables users to book and release parking spots, while allowing admins to manage parking infrastructure efficiently.

Technologies Used

- **Flask:** Serves as the main web framework for rapid development.
- **Flask-WTF:** Provides form handling and validation.
- **SQLAlchemy:** Manages the database with an ORM layer for Python objects.
- **Flask-Login:** Manages user sessions and authentication.
- **WTForms:** Simplifies form creation with built-in validation rules.
- **Jinja2:** Enables clean and efficient templating for dynamic content.

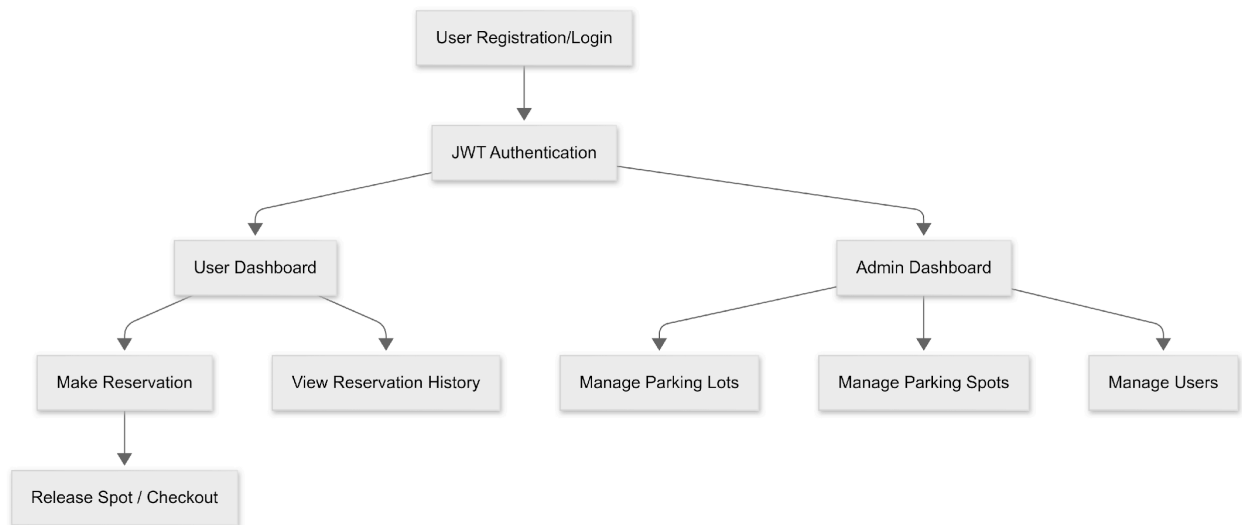
DB Schema Design

- **User:** id (Primary Key), name, address, phone, email (unique), password, is_admin, created_at
- **ParkingLot:** id (Primary Key), name, location, total_spots, created_at
- **ParkingSpot:** id (Primary Key), spot_number, is_occupied, lot_id (Foreign Key → ParkingLot), created_at
- **Reservation:** id (Primary Key), user_id (Foreign Key → User), spot_id (Foreign Key → ParkingSpot), vehicle_number, checkin_time, checkout_time



API Design

The Parking App API provides secure, role-based endpoints for managing users, parking lots, spots, and reservations. It enables admin control over parking inventory and allows users to reserve and release spots with authentication.



Architecture & Features

- **MVC Structure** – Clean separation using Flask with models, views, and controllers.
- **Role-Based Access** – Secure login system for users and admins.
- **Lot & Spot Management** – Admins manage lots/spots; dynamic availability tracking.
- **Real-Time Booking** – Live reservation, check-in/out, and billing based on duration.
- **Responsive UI** – Bootstrap + Jinja2 templates for mobile-friendly design.

Video

[Video Link](#)