

Vehicle Parking App

Project Report

Author

Name: Bukya Jagadish Naik

Email: 23f3004085@ds.study.iitm.ac.in

College: IIT Madras

About Me: I am a Diploma student at IIT Madras with strong proficiency in Python, Flask, SQLAlchemy, and full-stack web development, with hands-on experience building real-world projects.

Description

This is a web-based Vehicle Parking System built using Flask. It enables users to register, book parking spots, manage reservations, and rate their parking experience.

Administrators can manage parking lots, monitor reservations, and review user feedback. The system ensures real-time accuracy of parking spot availability.

Technologies Used

- **Backend:** Python, Flask, Flask-Login, Flask-WTF, SQLAlchemy
- **Frontend:** HTML, CSS, Bootstrap
- **Database:** SQLite

Features & Architecture

Features Overview:

- Auth via Flask-Login and CSRF protection
- Role-based access (Admin/User)
- Reservation history and live status tracking
- Real-time Parking Spot Availability
- Rating System for Feedback
- Admin dashboard to manage users, lots, and spots
- Enable/disable lots and spots (is_active flag)
- Smooth user experience using Bootstrap modals

Structure:

```
parking_app_23f3004085/
├── app.py
├── models/
│   └── models.py
├── templates/
│   ├── base.html
│   ├── user_dashboard.html
│   └── admin_dashboard.html
├── static/
│   ├── css/
│   │   └── style.css
│   ├── images/
│   └── js/
│       └── script.js
├── controllers/
│   ├── user_dashboard_controller.py
│   ├── admin_dashboard_controller.py
│   └── auth_routes.py
├── forms.py
├── migrations/
├── instance/
│   └── parking_app.db
└── requirements.txt
```

Database Schema Design (Tables)

Users : id (PK), fullname, email, phone, pincode, address, role, password

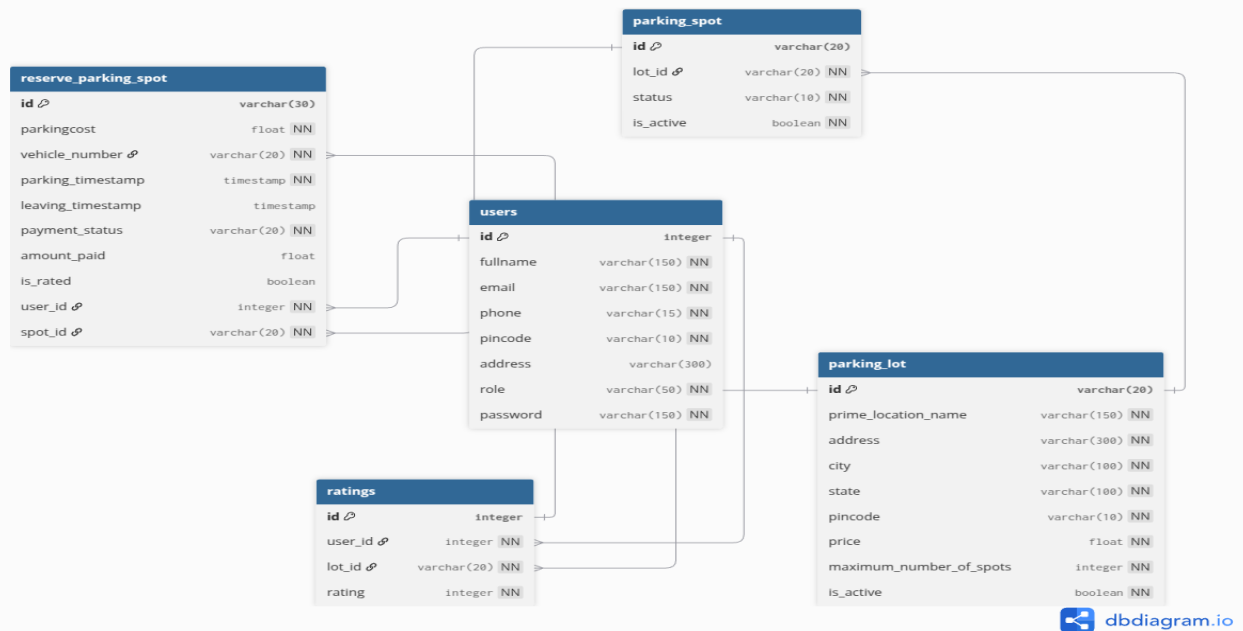
ParkingLot : id (PK), prime_location_name, address, city, state, pincode, price, maximum_number_of_spots, is_active

ParkingSpot : id (PK), lot_id (FK), status, is_active

ReserveParkingSpot : id (PK), user_id (FK), spot_id (FK), parkingcost, vehicle_number, parking_timestamp, leaving_timestamp, payment_status, amount_paid, is Rated

Rating : id (PK), user_id (FK), lot_id (FK), rating

All tables use foreign key relationships to maintain data integrity.



Design Rationale :

- Separate models for Users, Lots, Spots, Reservations, and Ratings
- is_active, status, and timestamps ensure real-time accuracy
- Ratings enable user feedback loop for service improvement

Video

Video Link:

<https://drive.google.com/file/d/1SdhqGgfWbpNBntQswftwZ3kzR836vz/view?usp=drivesdk>