

## Vehicle Parking App - V1

### Author

Name: Ayush Kumar Jha

Roll Number: 22f3000846

Student Email: 22f3000846@ds.study.iitm.ac.in

*I am a 3rd-year Computer Science student. I'm also pursuing a B.S. in Data Science from IIT Madras and currently at the Diploma level. I'm passionate about technology and aim to join a top tech company where I can apply my skills, gain hands-on experience, and keep growing. Grateful to all who supported me on this journey.*

### Description

This project implements a multi-user vehicle parking management system for 4-wheelers, supporting both admin and user roles. The app allows admins to manage parking lots and spots, while users can book, occupy, and release parking spots, with all actions tracked and summarized.

### Technologies Used

Flask (Web framework)

Flask-SQLAlchemy (ORM for SQLite)

Jinja2 (Templating engine)

Bootstrap (Frontend styling)

SQLite (Database)

Matplotlib (For summary charts)

### Purpose:

Flask provides a lightweight backend, SQLAlchemy simplifies database operations, Jinja2 enables dynamic HTML rendering, Bootstrap ensures responsive UI, and SQLite offers a simple, file-based database suitable for local demos. Matplotlib is used for generating visual summaries.

### DB Schema Design

#### User:

id (PK, int)

email (unique, string, not null)

password (string, not null)

name (string, not null)

address (string)

pincode (string)

role (string, default 'user')

***ParkingLot:***

id (PK, int)  
name (string)  
address (string)  
pincode (string)  
price\_per\_hour (float)  
max\_spots (int)

***ParkingSpot:***

id (PK, int)  
lot\_id (FK, int)  
status (string, default 'A')

***Reservation:***

id (PK, int)  
spot\_id (FK, int)  
user\_id (FK, int)  
vehicle\_number (string, not null)  
start\_time (datetime)  
end\_time (datetime, nullable)  
cost (float, nullable)  
status(string, default 'Booked')

**Architecture and Features**

The project follows MVC architecture

Controllers: All business logic and route handling are in the controller directory (routes.py, auth\_routes.py).

Models: Database models are defined in models.py.

Templates: All HTML templates are in the templates directory.

Static files: CSS and images are in the static directory.

**Features Implemented:**

Role-based authentication and dashboards  
Admin management of lots and spots  
User reservation, occupancy, and release of spots  
Parking history and summary charts  
Cost calculation based on parking duration  
Responsive UI with Bootstrap  
Search functionality for lots

**Video**

[Click here](#)