

Author

Anubhav Gupta

24f1000225

24f1000225@ds.study.iitm.ac.in

I'm a student of IIT Madras BS in Data Science. I'm interested in full-stack development.

Description

This project is about building a web-based vehicle parking management system. It includes user and admin roles, booking/reservation system, and cost calculation based on duration.

AI/LLM Used: Approximately **30% overall**

This includes:

Backend (Flask + SQLAlchemy):

Used for routing, models, database integration, and logic.

Frontend (HTML + Jinja2 + Bootstrap): Handled manually.

All the code was written and tested by me. LLM assistance was used only for debugging and clarifying errors, not for complete code generation.

Technologies Used

- Flask (for backend logic and routing)
 - SQLite (for database management)
 - SQLAlchemy (for ORM)
 - HTML/CSS, Bootstrap (for frontend)
 - Flask extensions: flask_sqlalchemy, flask_bcrypt, flask_session
- Flask was chosen for simplicity and rapid prototyping. Bootstrap was used for responsive design.

DB Schema Design

User:

id, name, email, password, role, is_admin

ParkingLot:

id, location, address, pincode, price, max_spots, spots

ParkingSpot:

Id, spot_number, is_occupied, lot_id (FK), status

Booking:

id, user_id (FK), lot_id (FK), spot_id (FK), start_time, end_time, cost, is_active

Relationships:

User ↔ Booking: One-to-Many

ParkingLot ↔ ParkingSpot: One-to-Many

ParkingSpot ↔ Booking: One-to-Many (each spot can be reused after release)

API Design

Basic RESTful APIs were created for fetching lot and reservation data in JSON. Future scope includes adding full API endpoints with authentication using Flask-Restful.

Architecture and Features

The app follows the MVC pattern: - Views (routes) in views.py and auth.py - Templates: Stored in /templates for frontend rendering (login, dashboard, etc.) - Models: Defined in (models.py)

Implemented Features:-

Role-based login and redirection (admin/user) - Admin dashboard to manage lots, view users, and monitor spots - User dashboard to book, release and view booking history - Cost calculation based on parking duration - Flash messages and form validations

Video :

[video link](#)