Author
Mantra Patel
24f2005922
24f2005922@ds.study.iitm.ac.in
I am a diploma-level student with a strong interest in full-stack web development and solving real-world problems using Python.

Project Details

This project is a web-based Parking Management System designed to streamline parking space allocation for users and provide admins with tools to monitor and manage parking lots effectively. The problem was approached by clearly separating user and admin functionalities, ensuring a smooth booking and release experience for users, and allowing admins to view summaries, search data, and manage slots efficiently.

AI/LLM usage: ~20%, mostly for debugging, UI refinement, error tracing, and getting styling suggestions. No code was directly copied without understanding.

Frameworks and Libraries Used

- Python
- Flask (Flask-SQLAlchemy, Blueprints)
- SQLite
- HTML, CSS
- Jinja2 Templating
- JavaScript (minor usage)
  Purpose: Flask was used for routing and backend logic; SQLAlchemy managed DB operations; CSS provided styling; Jinja2 for templating user/admin views.

ER Diagram and DB Schema

1. **User**: userid (PK), fullname, email, password, Address, pincode
2. **Admin**: admin (PK), password
3. **ParkingLot**: lot_id (PK), prime_location_name, price, address, pin_code, max_spots
4. **ParkingSpot**: spot_id (PK), lot_id (FK → ParkingLot), status
5. **ReserveParking**: res_id (PK, auto-increment), spot_id (FK → ParkingSpot), user_id (FK → User), vehicle_no, date, in_time, out_time, cost_unit_time
   All necessary foreign key constraints are applied to maintain data consistency.

API Resource Endpoints

No REST API endpoints were created.

Architecture

- Flask Blueprints are used to modularize code under controllers/ for user and admin
- Templates are stored in the templates/ folder, and static files like CSS and Images are in static/.
- Admin can manage parking lots and view user bookings, perform search and summary operations.
- Users can view available slots, make bookings, release vehicles, and view their parking history.
- Error handling and validation are included throughout the app.

Features Implemented

- Flask session security is implemented to track logged-in users securely
- Password verification with confirm-password validation during sign-up
- Animated confirmation screen after payment
- Auto-incremented reservation IDs and filtered booking history

Video

https://drive.google.com/file/d/10zPszBwyXoU2kspcngnodslCsWc9P8iz/view?usp=sharing