Vehicle Parking App

Author: Tiwari Kanak Suraj

Roll no – 23f2005464

23f2005464@ds.study.iitm.ac.in

I am currently enrolled in the BS program as a dual degree, and this marks my first official project in web development as a bs student. I have dedicated my full effort to understanding the end-to-end development process and aimed to implement everything I've learned through both academic concepts and practical experimentation.

Note:
To run this application, install the required dependencies using commands given below:

pip install -r "requirements.txt"

Then run the application using:

flask –-app app run –reload

or

flask run

Description

Project Title :Vehicle Parking App - V1
This is a multi-user web application designed to simplify the management of parking lots, individual parking spots, and user reservations.

Admins have an access to manage parking lot records, oversee user accounts, and view summaries. Regular users can browse available lots, and gain insights into their booking history and expenses using interactive charts.

AI/LLM –
I utilized **ChatGPT** to assist in roughly **15%** of the development process, primarily in:

Designing backend logic and routes

Testing and debugging

Generating and integrating visualizations using charts

This significantly improved the development speed and quality of the application.

Technologies used

Flask – Handles web requests, routes, and template rendering

Flask SQLAlchemy – ORM for defining and interacting with the database

SqlAlchemy core – for extending use of standard frame work

Jinja2 – For embedding dynamic content into HTML templates

SQLite – Lightweight local database

Chart.js – Used for plotting reservation cost charts

Database Schema Design

Each model includes a unique id as its primary key. The relationships are defined with proper foreign keys and cascading deletes to maintain integrity.

ER diagram image link : https://drive.google.com/file/d/1qh4zGrua8xOb80PJAgTMqcj19HCij1vk/view?usp=sharing

- **User**
  - Id, datatype: int, primary-key
  - Email, datatype: str, unique, not-null
  - Password, datatype: str, not-null
  - Full_Name, datatype: str, not-null
  - Address, datatype: str, not-null
  - Pincode, datatype: str, not-null
  - Relationship: One-to-Many with Reserve_parking_spot
- **Admin**
  - Id: admin-id, datatype: int, primary-key
  - email, datatype: str, unique, not-null
  - password, datatype: str, not-null
  - Full_Name, datatype: str, not-null
  - Relationship: One-to-Many with Parking_lot
- **Parking Lot**
  - Id: lot-id, datatype: int, primary-key
  - prime_location, datatype: str, not-null
  - Address, datatype: str, not-null
  - Pincode, datatype: str, not-null
  - Max_no_of_spots, datatype: int, not-null
  - price_per_hour_of_spot, datatype: int, not-null
  - admin_id, datatype: int, foreign-key → Admin.id, not-null
  - Relationship: One-to-Many with Parking_spot
- **Parking Spot**
  - Id: spot-id, datatype: int, primary-key
  - status, datatype: str, not-null (e.g., 'A' = available, 'R' = reserved)
  - lot_id, datatype: int, foreign-key → Parking_lot.id, not-null, cascade on delete
  - Relationship: One-to-Many with Reserve_parking_spot
- **Reserved_Parking_Spot**
  - Id: reservation-id, datatype: int, primary-key
  - user_id, datatype: int, foreign-key → User.id, not-null, cascade on delete
  - spot_id, datatype: int, foreign-key → Parking_spot.id, not-null, cascade on delete
  - parking_timestamp, datatype: datetime, not-null
  - end_parking_timestamp, datatype: datetime, nullable
  - duration, datatype: int, nullable (in hours)
  - Total_amount_user_paid, datatype: int, not-null, default: 0
  - vehicle_number, datatype: str, not-null

Architecture and Features :

- **app.py** – Entry point of the Flask application and also a main file for app context.
- **models/models.py** – Contains all database model definitions.
- **controllers/admin.py** – Routing logic and endpoint definitions of mostly related to admin.
- **controllers/user**.py - Routing logic and endpoint definitions of mostly related to user.
- **controllers/authentication**.py - Routing logic and endpoint definitions of mostly related to authentication.

- **templates/** – Contains all HTML templates for rendering views.
- **static/** – Holds CSS, images, and other assets.
- **instance/** – Stores the SQLite database file.

Features Summary:

1. Admin functionality:
    - Add, edit, delete parking lots
    - View user details and lot-specific summaries
    - Search users and lots
2. User functionality:
    - Search and book available spots
    - Track booking history
    - View interactive analysis charts
3. Additional functionality :
    - Admin can “Download reports” in pdf format

Video:

https://drive.google.com/file/d/1ZphjUguyMOLMXXhj7dkkCdQhyXqgwGpO/view?usp=sharing