Vehicle Parking App

Final report for the MAD I capstone project

Submitted by

Name: VINAY S R

Roll number: 21f1005235

IITM Online BS Degree Program,

Indian Institute of Technology, Madras, Chennai

Tamil Nadu, India, 600036

Contents

1 Project Title , problem statement and approach

Title: Vehicle Parking App V1

Problem Statement:

It is a multi-user app (one requires an administrator and other users) that manages different parking lots, parking spots and parked vehicles. Assume that this parking app is for 4-wheeler parking.

The goal was to design and develop a full-stack web application that allows users to book parking spots, and enables admins to manage spots, view summaries, and generate reports. The system should support different user roles and provide analytics to monitor parking lot usage and revenue trends.

Approach:

- The application was built using the Flask framework for the backend.
- The database schema was designed to support entities like Users, Bookings, Parking Lots, Parking Spots, and Release History.
- Templates were rendered using Jinja2, and data visualizations were implemented using Chart.js.
- A role-based login system was integrated using Flask-Login and decorators for access control.
- REST-style API endpoint was defined for user registration.

During the course of the project, I conducted extensive research to identify similar applications in the market, among which Park+ served as a primary reference. Although my project was not fully developed, it encompassed several complex aspects that required careful attention.

A key focus was placed on delivering a visually appealing user interface. After considerable exploration, I finalized a consistent color scheme and styling guidelines to be applied throughout the application. This foundational design work informed the creation of HTML templates and CSS files. To enhance the visual presentation, I leveraged AI tools such as Perplexity and ChatGPT, which assisted in refining the styling and improving user experience.

At the outset, the project plan was flexible and evolved over time, with an agile

development methodology guiding the iterative process. The wireframe provided a central point of reference that ensured alignment across design and development activities.

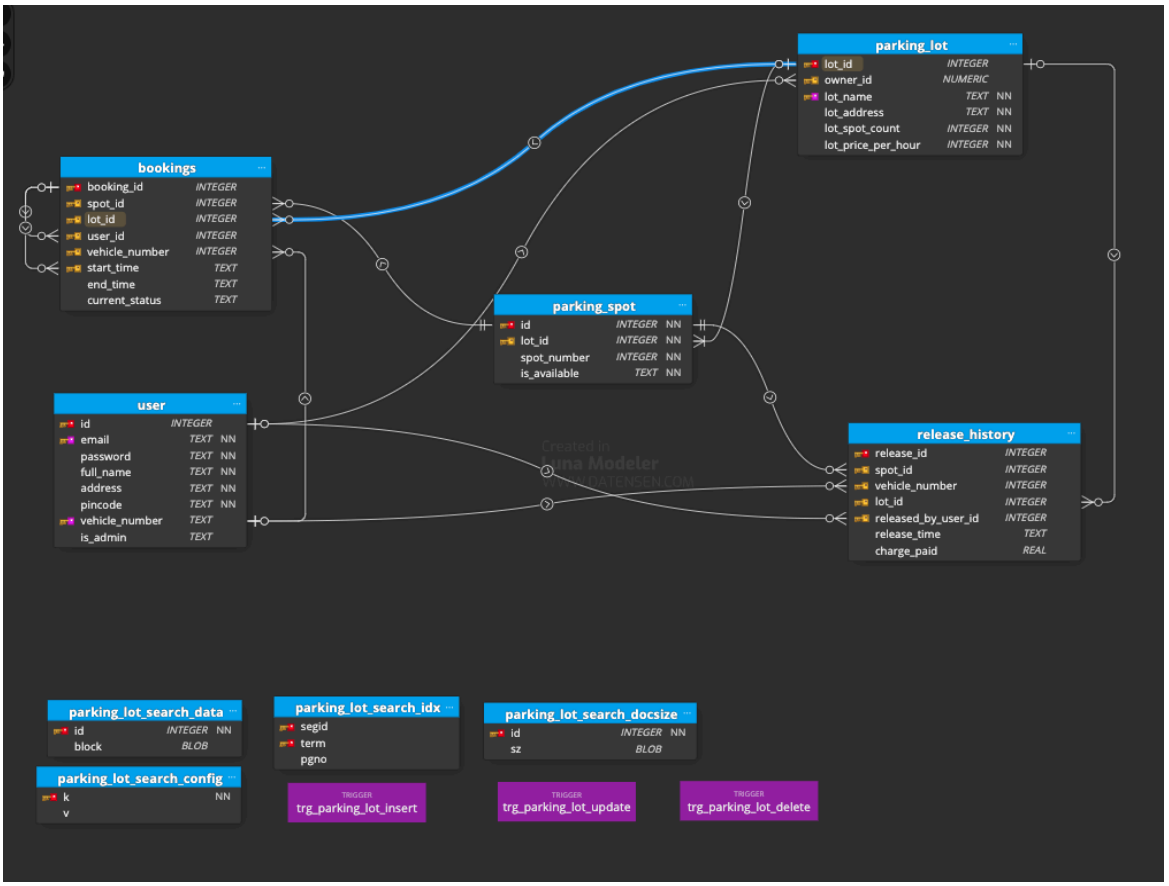
One of the major challenges encountered was in designing the database schema. It often became necessary to revisit and redesign components of the schema that had already been implemented, which contributed to significant development time overhead. Moreover, implementing the business logic, such as dynamically reducing the count of available parking spots upon deletion and preventing the deletion of parking spots when occupied, presented complex logical hurdles. In these cases, taking a step back and approaching the problem anew the following day proved effective in overcoming the obstacles.

Overall, I have endeavored to ensure the application functions logically and robustly. While there were instances of difficulty and delay, the iterative and reflective development approach facilitated continuous progress and problem resolution.

2 Framework and Libraries Used

Framework/Library	**Purpose**
Flask	Backend web framework
SQLAlchemy	ORM for database interaction
Flask-Login	User Authentication and Session Management
Chart.js	Charts
Jinja2	Template Rendering

3 Entity RelationShip Diagram



Credit- image generated from lunamodeler software

4 Relationship

User ↔ Bookings: One-to-Many;

User ↔ Parking_lot: One-to-Many

Parking_lot ↔ Parking_spot: One-to-Many

User ↔ Bookingst: One-to-Many

Parking_spot ↔ Bookings: One-to-Many

Parking_lot ↔ Bookings: One-to-Many

5 API End PoInts

API End Point	Method	Function
/api/users	POST	To register new users

6. AI / LLM Usage Declaration

Throughout the development process, a variety of established resources were utilized, including official documentation, Stack Overflow, and authoritative framework references such as Flask and SQLAlchemy. While Large Language Models (LLMs) like Perplexity and ChatGPT were consulted, their use was limited to high-level clarifications, bug resolution support, and syntax verification—functioning similarly to standard programming resources to explore potential options. Specifically, these AI tools were leveraged primarily to assist with cascading style sheet (CSS) styling enhancements rather than to generate core project logic.(I'd estimate around 10% of the entire project)

7. Video Link

Video Link