Author

Abhijith Sriram V
24F2004242
24f2004242@ds.study.iitm.ac.in
A Diploma student at IIT Madras pursuing a BS Degree, and currently in my second year of B.Tech Information Technology at SSN College of Engineering, Chennai. I am a musician, I play the keyboard and guitar. I like exploring new technologies and ideas.

Description

To build a vehicle parking application with 2 roles, one as a user and the other as admin(predefined). The users can register, login, view, choose, parking spots of parking lots. The admin can login, view, add, update, delete parking spots of parking lots(if unoccupied) at any point of time.

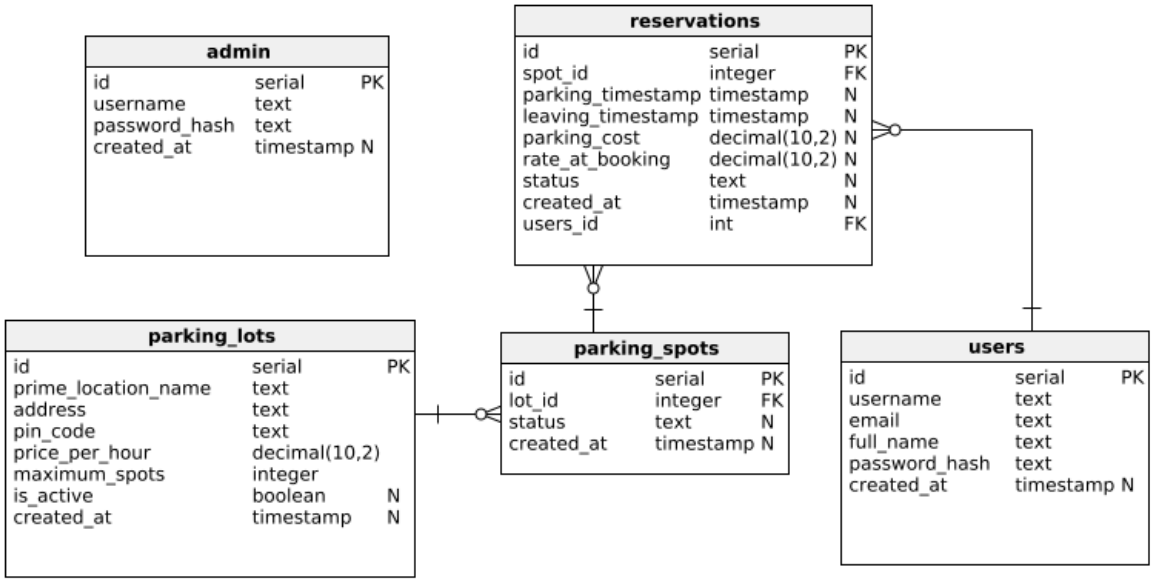
AI/LLMs used: Claude.ai. Primarily used for idea refinement, code critic, and debugging purposes in completion of various milestones. About 10 percent in each milestone.

Technologies used

- Flask: Web framework for application backend and routing
- Jinja2: Template engine for dynamic HTML rendering
- SQLite: Lightweight database for data persistence
- HTML/CSS: Frontend presentation with custom responsive design
- JavaScript: Real-time cost updates and interactive features
- Python: Core programming language for business logic

Purpose: These technologies provide a complete web application stack that's easy to deploy, maintain, and doesn't require external dependencies, making it ideal for a parking management system

DB Schema Design



Design Rationale: Normalized design ensures data integrity with proper foreign key relationships. Status fields enable real-time tracking, and separate admin table maintains role separation. The schema supports scalability and efficient querying for parking operations

API Design

Created REST API endpoint for real-time cost calculation:

GET /api/current_cost/<reservation_id>: Returns current parking cost, duration, and billing details for active sessions

Implementation: Uses Flask route with JSON response, calculates cost based on parking duration and hourly rates. Enables real-time cost updates in the frontend without page refresh

Architecture and Features

Project Organization: MVC architecture with Flask routes as controllers in app.py, Jinja2 templates in /templates directory, and database functions in database_setup.py. Static CSS embedded in base template for simplified deployment.

Core Features Implemented: User registration/authentication, parking lot management, real-time spot reservation, automatic cost calculation with hourly billing, comprehensive analytics dashboards, and responsive UI. Additional features include real-time cost tracking, detailed cost breakdown analysis, and admin system monitoring with performance metrics.

Video

https://drive.google.com/file/d/18575VGTwe_xaxg_KwHz4TNKLYtIjDKt7/view?usp=sharing