Author

Name:-Mahaprasad Nanda
ID:-23f3001830
Email:-23f3001830@ds.study.iitm.ac.in
I am a diploma level student at IIT Madras BS degree program along with that I am Pursuing BTech in Information Technology from IIIT Bhubaneswar.

Description

The Vehicle Parking App manages parking lots, spots, and reservations, allowing users to book and admins to manage parking spaces through a scalable web application created for this project.AI/LLM assistance was used to the extent given by IIT Madras for UI enhancement and to make the summary charts and route logics.

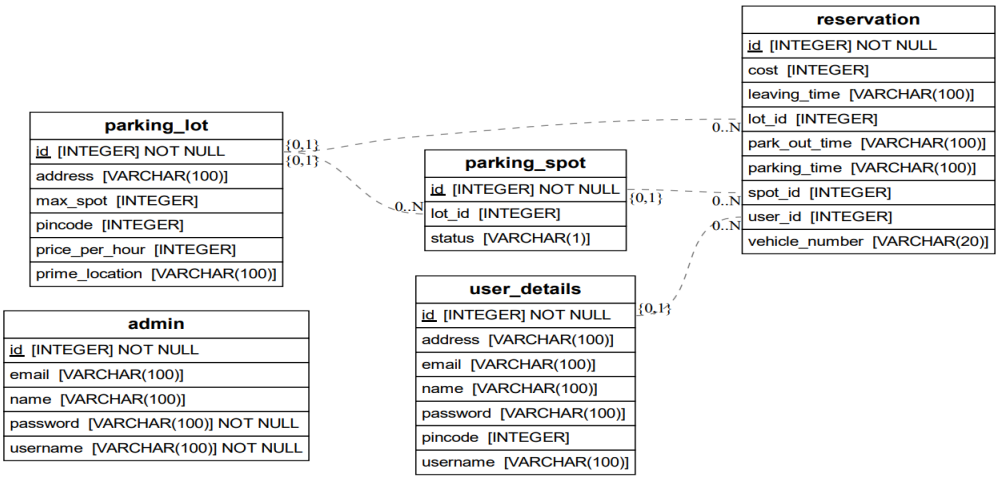
Technologies Used:

1. **Flask** – Core web framework to build the application.
2. **Flask-SQLAlchemy** – ORM used for database modeling and interaction.
3. **SQLite** – Lightweight relational database for local development.
4. **Flask-Login** – Manages user sessions and authentication securely.
5. **Jinja2** – Templating engine for rendering HTML dynamically.
6. **Bootstrap** – CSS framework for responsive and clean UI.

Purpose Behind Using These:

- **Flask & Extensions:** Enables modular and scalable development of web apps with built-in support for routing, sessions, and database integration.
- **SQLite:** Lightweight and easy to use, perfect for small-to-medium projects without complex database needs.
- **Jinja2 & Bootstrap:** Provides dynamic content rendering with a modern, mobile-friendly interface for improved user experience.

DB Schema Design



Entity Separation:

- parking_lot and parking_spot allow scalable spot management per lot.

- user_details and admin tables ensure security isolation.

Reservation-Centric:

The central reservation table maintains a complete audit trail with multiple timestamps for accurate billing and vehicle tracking.

Foreign Keys:

- Ensure data integrity while enabling efficient queries.
- Redundant lot_id in reservations optimizes lot-wise reporting.

Data Types:

- VARCHAR timestamps provide flexibility.
- INTEGER costs avoid precision issues.

Business Logic:

- Status field enables real-time availability checks.
- Vehicle numbers link physical cars to digital records.

Scalability:

Design supports future enhancements while maintaining performance through strategic indexing and denormalization.

API DESIGN

This is a traditional web application built with Flask, not an API-based application as using API was optional according to the project document .

Architecture and Features

The parking management system follows a structured Flask architecture with clear separation of concerns. The main application entry point is app.py, which initializes the Flask app, database, and creates default admin credentials. Controllers are centralized in controllers/controllers.py, containing all route handlers for admin and user functionalities. Templates are organized in the templates/ directory with separate files for admin pages (admin_dash.html, admin_add.html, etc.) and user pages (user_dash.html, user_book.html, etc.). Database models are defined in models/db.py using SQLAlchemy ORM, establishing relationships between parking lots, spots, users, reservations, and admin entities. Static assets (CSS, JS) are stored in dedicated directories for styling and client-side functionality.

Features Implementation

Default Features: User registration/login with session management, parking spot booking and release with automatic cost calculation, real-time spot status updates, and comprehensive parking history tracking. Admin capabilities include parking lot management (add/edit/delete), spot monitoring, and user oversight.Enhanced navigation bars with modal-based summaries using Chart.js for data visualization. Admin dashboard displays revenue pie charts and occupancy bar charts, while user dashboard shows personalized parking statistics.

Implementation Details: Features are implemented through Flask routes with proper session validation, SQLAlchemy queries for data persistence, Jinja2 templating for dynamic content rendering, and JavaScript for interactive UI components. Chart.js integration provides real-time data visualization, while Bootstrap ensures responsive design across all pages.

Video

https://drive.google.com/file/d/16L3HIojDpoXtzGPC-TAh_W1O_L2VUp-2/view?usp=sharing