Vehicle Parking App - V1

Project Report

Author

Name: Alok Kumar Tripathi
Roll Number: 23F3003225
Email: 23f3003225@ds.study.iitm.ac.in

About Me:
I'm a second-year Data Science student at IIT Madras, with strong interests in product engineering, machine learning and AI. This project helped me apply my full-stack skills to solve a real-world urban problem.

Description

To build a **Flask-based Vehicle Parking Web Application** that enables users to seamlessly find, book, and manage parking spots in real-time, while allowing admins to oversee parking lots, monitor spot usage, manage user data, and analyze revenue and performance through an interactive dashboard.

Technologies Used

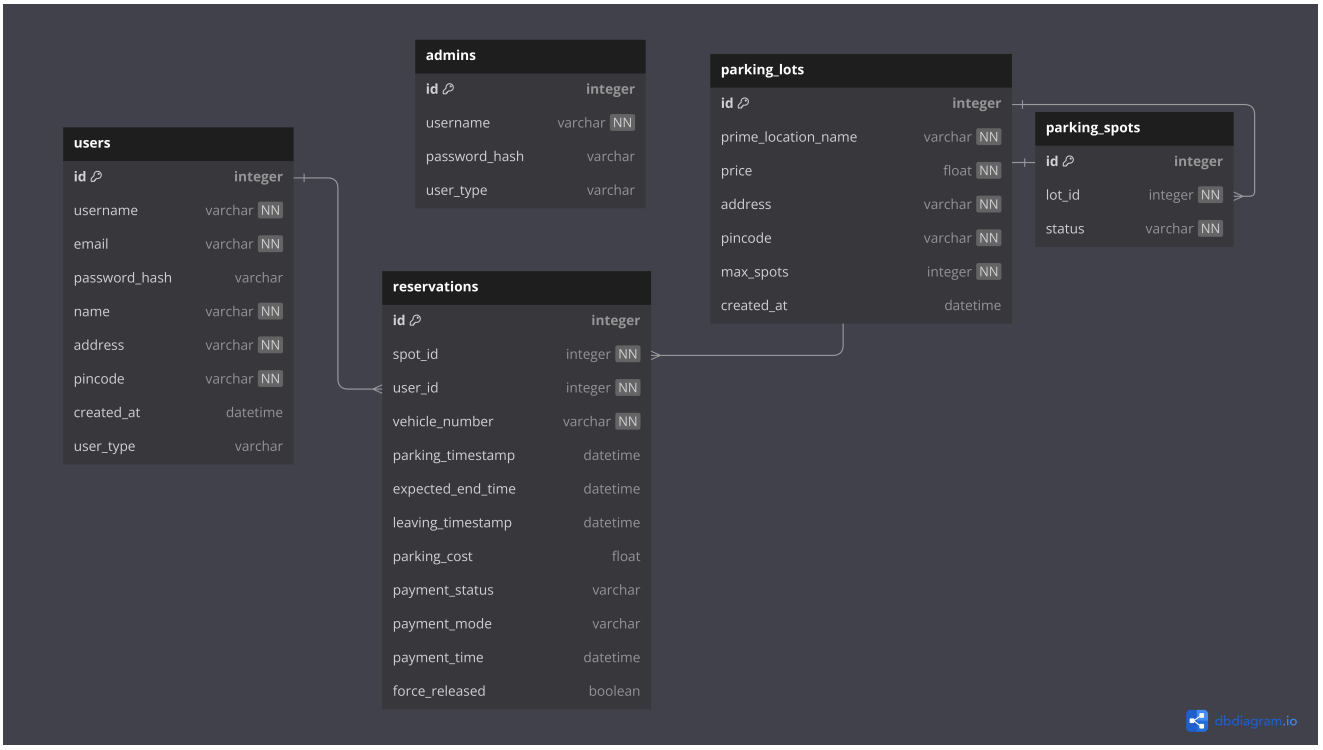
Backend: Flask, Flask-Login, Flask-WTF, SQLAlchemy, Werkzeug
Frontend: Bootstrap 5.3, JavaScript, jQuery, Chart.js, Font Awesome
Database: SQLite (lightweight, in-file DB for easy testing)

DB Schema Design

- **Users:**
  id, username, email, password_hash, name, address, pincode, created_at, user_type
- **Admins:**
  id, username, password_hash, user_type

- **ParkingLots:**
  id, prime_location_name, price, address, pincode, max_spots, created_at
- **ParkingSpots:**
  id, lot_id (FK), status
- **Reservations:**
  id, spot_id (FK), user_id (FK), vehicle_number, parking_timestamp, expected_end_time, leaving_timestamp, parking_cost, payment_status, payment_mode, payment_time, force_released

Design Rationale:
Clear role separation, referential integrity via foreign keys, and cascading deletes ensure smooth data flow and minimal redundancy.



API Design

Implemented RESTful APIs (`/api/*.py`) with JSON responses.

Endpoints:

- `POST /api/book` → Reserve spot
- `POST /api/release` → Free a reservation
- `GET /api/users` / `spots` / `availability`
- `POST /api/complete_profile`

Standard Response: `{ status, message, data }`
Docs: API YAML file submitted separately.

Note on Use of AI

AI tools (LLMs) were used during development for debugging, error resolution, and implementation brainstorming.

Architecture & Features

Structure:

```
parking_app_23f3003225/
├── app.py
├── models.py
├── forms.py
├── utils.py
├── requirements.txt
├── README.md
├── instance/
│   └── database.db
├── migrations/
├── routes/
│   ├── admin/
│   ├── api/
│   ├── main/
│   └── user/
├── static/
├── templates/
```

Features Overview:

- Auth via Flask-Login
- CSRF, secure sessions, role-based access
- Chart.js analytics for admins/users
- Light/dark UI, alerts, loading states
- All actions via modals (no reloads)

Key Functionality

User Dashboard:

- View & book slots with live timers
- Real-time availability status
- History, usage analytics, and profile prompts
- No double-booking allowed

Admin Panel:

- Add/edit/delete lots & spots
- View all users and reservation data
- Spot status: green (free), red (occupied)
- Analytics: revenue, usage, and performance graphs
- Filter users by activity/spend

Video

https://drive.google.com/file/d/189_sJC_mB-otASQqiNtc_jOrBovZwOka/view?usp=sharing