

Project Report:

Vehicle Parking App

Author

Rathin Samanta

22f1000896

22f1000896@ds.study.iitm.ac.in

I am a passionate learner currently pursuing the Diploma in Programming & Data Science from IIT Madras.

Description

This project involves building a full-stack smart vehicle parking application where users can reserve, release and view parking history, while admins can manage parking lots, monitor data, and automate tasks like reminders and reports.

Technologies Used

Backend: Flask

Frontend: HTML, CSS, Bootstrap, Vue.js (CDN)

Database: SQLite

Authentication: Flask-JWT-Extended

Background Jobs: Celery + Redis

Email Service: Flask-Mail

Caching: Flask

DB Schema Design

1. User

- id (Integer, PK)
- username (String, unique, not null)
- password (String, hashed)
- role (String - 'admin' or 'user')

2. ParkingLot

- id (Integer, Primary Key)
- name (String)
- address (String)
- pin_code (String, 6 digits)
- price (Float, per hour)
- total_spots (Integer)
- spots: one parkinglot can have many parkingspots.

3. ParkingSpot

- id (Integer, Primary Key)
- lot_id (Foreign Key to ParkingLot)
- status (String - 'A' or 'O')
- reservations: one parkingspot can have many reservations

4. Reservation

- id (Integer, Primary Key)
- user_id (Foreign Key to User)

- spot_id (Foreign Key to ParkingSpot)
- parking_time(Datetime)
- leaving_time(Datetime)
- cost(float)
- user: one user can have many reservations

API Design

Authentication: /login, /register, /dashboard

Admin Operations: /admin/lots, /admin/dashboard, /admin/reservations

User Operations: /user/reserve/<lot_id>, /user/release, /user/dashboard, /user/reservations, /user/lots, /user/export

Background Triggers: /send-daily-reminder, /send-monthly-report, /send-test-email, /run-task Implemented using Flask decorators, JWT protection, and JSON responses.

Architecture and Features

app.py contains all API routes and application logic. **models.py** defines the database schema. **tasks.py** holds asynchronous and scheduled jobs (Celery) **config.py** contains environment configuration. **Frontend** file contains all the html and js files.

Features Implemented

JWT-based authentication with role-based access (admin/user). Parking spot reservation and release. Reservation history, charts. Admin dashboard with lot management

Caching of dashboard data using Redis. Celery jobs for daily reminder and monthly email reports. Export reservation data to CSV via async job. Role-based UI redirection with Vue.js Data visualization with Chart.js

AI(%) used in this Project

Category	Percentage
Backedend	8%
Frontend	6%
Styling/UI	2%
Auth+CRUD	5%
Charts/apis	4%

Video

<https://drive.google.com/file/d/1ShRLvIhuH4P6UxksxshsZeSUvYQn7dQs/view?usp=sharing>