Project Report – ParkPal

Vehicle Parking V2 MAD2

Author

Ishita Tayal | 23f3001216@ds.study.iitm.ac.in

IITM BS in Data Science and Applications Diploma Level Student

Introduction & Description

ParkPal is a multi-user web application developed to streamline the management of 4-wheeler parking lots, individual parking spots, and vehicle reservations. It features admin-level control for lot management and user-level functionality for booking spots with real-time availability updates. Built with Flask, VueJS, SQLite, Redis, and Celery, ParkPal also incorporates authentication, reporting tools, and performance enhancements for a seamless experience.

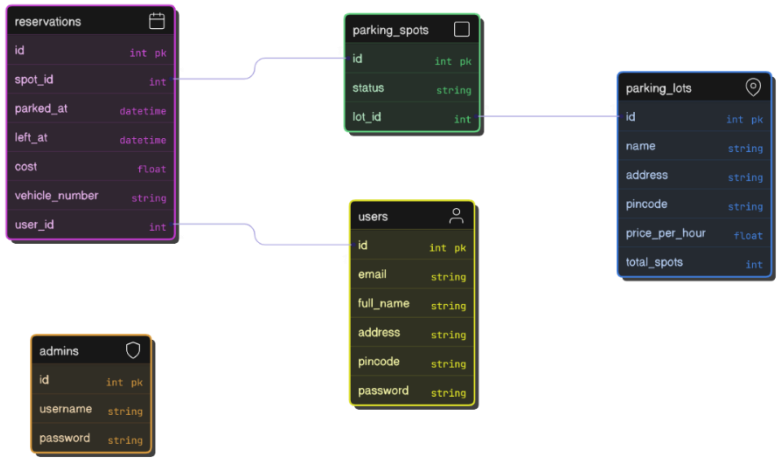
Technologies Used

- **Backend:** Flask, Flask-SQLAlchemy, SQLite, Redis, Celery
- **Frontend:** VueJS, Chart.js

Database Schema

The database efficiently manages all entities while ensuring data integrity through normalization and foreign keys.

- **User**: registered user details including email, password, full name, address, and pincode, and links to reservations.
- **Admin**: represents the system's superuser with privileged access to manage parking lots and users.
- **ParkingLot**: a parking lot with its name, address, pincode, hourly rate, total number of spots, and associated parking spots.
- **ParkingSpot**: individual parking spaces within a lot that track their occupancy status and reservation history.
- **Reservation**: user bookings with timestamps, cost, and vehicle number, linking users to their reserved spots.



API Design

Common routes
@app.route('/')
@app.route('/whoami', methods=['GET'])
@auth_bp.route('/logout', methods=['POST'])
Authentication routes
@auth_bp.route('/register', methods=['POST'])
@auth_bp.route('/user-login', methods=['POST'])
@auth_bp.route('/admin-login', methods=['POST'])
User routes
@auth_bp.route('/api/user/lots', methods=['GET'])
@auth_bp.route('/api/user/book', methods=['POST'])
@auth_bp.route('/api/user/release', methods=['POST'])
@auth_bp.route('/api/user/reservations', methods=['GET'])
@auth_bp.route('/api/user/summary', methods=['GET'])
@auth_bp.route('/api/user/profile', methods=['GET'])
@auth_bp.route('/api/user/profile', methods=['PUT'])
Admin routes
@auth_bp.route('/api/admin/profile', methods=['PUT'])
@auth_bp.route('/api/parking-lots', methods=['GET'])
@auth_bp.route('/api/parking-lots', methods=['POST'])
@auth_bp.route('/api/parking-lots/<int:lot_id>', methods=['PUT'])
@auth_bp.route('/api/parking-lots/<int:lot_id>', methods=['DELETE'])
@auth_bp.route('/api/users', methods=['GET'])
@auth_bp.route('/api/admin/summary', methods=['GET'])
@auth_bp.route('/api/parking-lots/<int:lot_id>/spots', methods=['POST'])
@auth_bp.route('/api/parking-spots/<int:spot_id>', methods=['DELETE'])
Search routes
@auth_bp.route('/api/search-parking', methods=['GET'])
Daily reminder route
@auth_bp.route('/api/run-daily-reminder', methods=['GET'])
Monthly report route
@auth_bp.route('/api/run-monthly-report', methods=['GET'])
CSV export routes
@auth_bp.route('/api/export-csv', methods=['POST'])

Architecture and Features

- Modular Architecture with clear separation of concerns across layers: backend API (Flask), frontend interface (VueJS), database management (SQLite), caching layer (Redis), and background task execution (Celery).
- Role-Based Access Control using session-based authentication to differentiate between admin-level operations and user-level interactions.
- Core Features include automatic parking spot allocation, real-time occupancy tracking, background task handling for notifications, and insightful usage analytics via scheduled reports.

Implementation Details

- Built a RESTful backend with Flask, leveraging SQLAlchemy ORM for efficient handling of users, parking lots, spots, and reservation data.
- Developed interactive admin and user dashboards using VueJS and Bootstrap, offering dynamic interfaces for bookings and status updates.
- Implemented Redis and Celery to handle background processes such as daily reminders, monthly usage summaries, and CSV data exports.

Conclusion

ParkPal is a well-structured, full-featured parking management system engineered for scalability, maintainability, and an intuitive user experience.

Demo Video

https://drive.google.com/file/d/13OdPRIDEwUOlM1yVkLMdoeHOkDv8rvwv/view?usp=sharing