
Vehicle Parking System - Project Documentation

Author

- **Full Name:** Rajveer Kharade
 - **Roll Number:** 23f3000717
 - **Student Email:** 23f3000717@ds.study.iitm.ac.in
 - **About Me:**
I am a software engineering student passionate about developing web applications using modern technologies. I have experience in full-stack development with Python, JavaScript, and database design. I enjoy creating user-friendly interfaces and building scalable backend systems.
-

Project Description

This project is a **Vehicle Parking System** designed to manage parking lot operations for both administrators and users. Key functionalities include:

- Admin-controlled parking lot creation, updates, and real-time monitoring
 - User-friendly interface for booking, releasing, and tracking parking history
 - Automated email reminders and monthly reports
 - Real-time updates with performance-focused caching
 - Exportable reports for insights and analysis
-

Technologies Used

Backend Technologies

Technology	Purpose
Flask	Core framework to handle HTTP and RESTful APIs
Flask-SQLAlchemy	ORM for seamless database interactions
Flask-CORS	Manage cross-origin requests between frontend and backend
Flask-Mail	Handle automated email communication
Flask-APScheduler	Background tasks (e.g., reminders, reports)
Flask-Caching	Improve performance with data caching
Werkzeug	Secure password hashing and utility functions
PyJWT	Token-based user authentication
SQLite	Lightweight database for development/testing

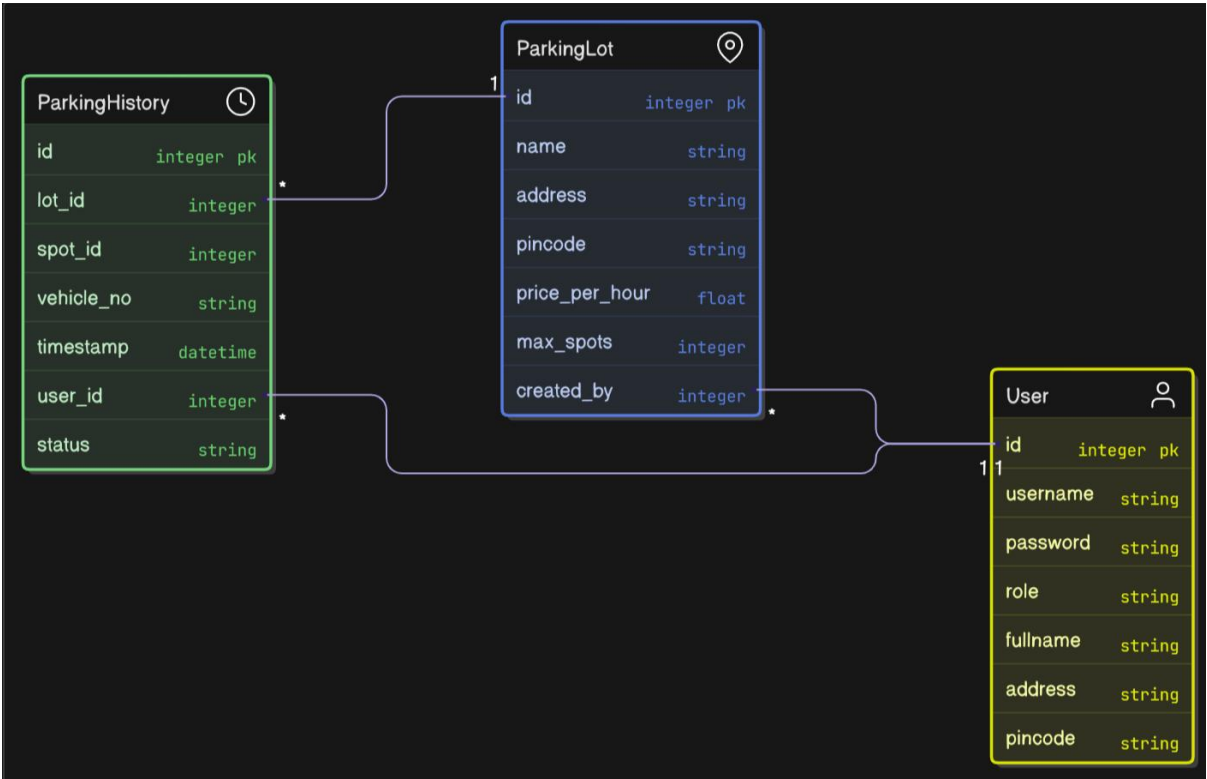
Frontend Technologies

Technology	Purpose
Vue.js 3	Modern reactive frontend framework
Vue Router	SPA routing and page navigation
Axios	API communication with backend
Bootstrap 5	Responsive design and UI styling
Bootstrap Icons	Consistent icon styling throughout UI
Vite	Fast build tool and dev server
Chart.js	Interactive visualizations for statistics

Development Tools

- **ESLint** – Code quality enforcement
- **Vitest** – Unit testing for Vue components
- **JSDOM** – Simulated DOM environment for frontend testing

Database Schema Design



Design Rationale

- **Normalization:** Ensures minimal redundancy
 - **Foreign Keys:** Preserve relationships and enable JOIN operations
 - **Scalable Structure:** Supports multi-user and multi-lot environments
 - **Secure & Role-based:** Admin and user operations are logically separated
-

API Design Overview

Authentication APIs

Method	Endpoint	Description
POST	/auth/login	Login & JWT token issue
POST	/auth/register	New user registration
POST	/auth/logout	Logout and token invalidation

Admin APIs

Method	Endpoint	Description
GET	/admin/parking-lots	List all lots (paginated)
POST	/admin/add-parking-lot	Create a new lot
PUT	/admin/edit-parking-lot/<id>	Edit lot info
DELETE	/admin/delete-parking-lot/<id>	Delete lot if no active bookings
GET	/admin/parking-lots/<id>/spots	View real-time spot status
GET	/admin/summary	Dashboard with analytics
GET	/admin/registered-users	View user base
POST	/admin/trigger-reminders	Manual email trigger
POST	/admin/trigger-monthly-reports	Send reports manually

User APIs

Method	Endpoint	Description
GET	/user/profile	View profile (cached)
PUT	/user/profile	Update user details
POST	/user/export-csv	Download parking history
GET	/user/dashboard	User summary and activity
GET	/user/my-parking-summary	Statistics overview
POST	/user/book-parking	Reserve a spot

Method	Endpoint	Description
POST	/user/release-parking	Free the parking spot

Architecture & Project Structure

Backend Structure

- **app.py** – Main Flask entry & configuration
- **models.py** – DB models and relationships
- **auth_routes.py** – Authentication routes
- **admin_routes.py** – Admin-specific functionality
- **user_routes.py** – User endpoints
- **reminder_job.py** – APScheduler email/report logic
- **extensions.py** – Mail, cache, JWT configuration

Frontend Structure

- **components/**
 - AdminDashboard.vue – Admin view
 - UserDashboard.vue – User homepage
 - UserProfile.vue – Profile editing
 - Login.vue / Register.vue – Auth interfaces
 - **router.js** – Routes with guards
 - **main.js** – Vue app setup
-

Presentation Video link

https://drive.google.com/file/d/1Sn1usZSqX8OOWv1I7LWcoPIgCDS28geJ/view?usp=drive_link