

# Project Report: IITM Online Degree – MAD 2 PARKWISE (A VEHICLE PARKING APP)

## Author

Mokana Priyaa S

23f3003011

[23f3003011@ds.study.iitm.ac.in](mailto:23f3003011@ds.study.iitm.ac.in)

I am an undergraduate student pursuing BTech Artificial Intelligence & Data Science in K.L.N. College of Engineering who will be graduating in the year 2026.

## Description

ParkWise is a Vehicle Parking Management System built using Flask (Python) for the backend and Vue2JS for the frontend. It supports user and admin roles, real-time parking spot reservation, autoallocation, occupancy management, and email notifications on reservation and release. The system includes a clean, role-based dashboard, scheduled job support, and parking history tracking.

## Technologies used

### Backend (Python + Flask)

- **Flask** – Web application framework
- **Flask-Login** – User session and authentication
- **SQLAlchemy** – ORM for database models • **SQLite** – Lightweight relational database
- **Flask-Mail** – Sending confirmation emails
- **Celery + Redis** – Asynchronous job handling
- **Python-dotenv** – Manage environment variables

### Frontend (Vue.js)

- **Vue.js 2** – Reactive JavaScript framework
- **Vue Router** – Frontend routing
- **Bootstrap** – UI styling and responsive layout
- **Fetch API** - Used for making API calls from frontend to Flask backend

## Key Functionalities of Technologies

- **Flask:** API routing and server-side logic
- **Flask-Login:** Secure login and role-based dashboard access
- **SQLAlchemy:** Defines and manages DB models (User, Reservation, Parking Lot, Parking Spot)
- **Celery + Redis:** Scheduled jobs (e.g., sending daily email reminders)
- **Flask-Mail:** Sends email notifications to users after reservation and releasing spots
- **Vue.js:** Renders dynamic frontend views with real-time data updates • **Bootstrap:** Provides a responsive and aesthetic UI
- **Fetch API** - Used for making API calls from frontend to Flask backend • **Python-dotenv:** Loads sensitive credentials securely from .env file

## DB Schema Design



## Architecture and Features

The project follows an **MVC (Model-View-Controller)** structure:

- **Model (Database table):** SQLAlchemy Models for Users, Admins, Lots, Spots, Reservations
- **View (VueJS Templates):** VueJS components (e.g., Dashboard.vue, ReserveSpot.vue)
- **Controller (Routes in Flask app):** Manage logic for user authentication, booking operations, and dashboard access.

Video link: [https://drive.google.com/file/d/1Rz5dMWQ8RGtpe\\_G191LsChyEjpAZyED4/view?usp=sharing](https://drive.google.com/file/d/1Rz5dMWQ8RGtpe_G191LsChyEjpAZyED4/view?usp=sharing)