## Author

JAYAPRAKASH REDDY PERAM
23f1001675
23f1001675@ds.study.iitm.ac.in
I am passionate about software engineering and full-stack development. This project gave me hands-on experience in building scalable backend systems using Flask, Celery, and Vue.

## Description

This project implements a Vehicle Parking Management System where users can book, reserve, and manage parking spots.

The backend supports scheduled tasks (daily reminders, monthly reports) and async jobs (CSV export).

AI/LLM Usage: Around 20–30% of the implementation involved assisted coding. I also used it as a helper for learning.

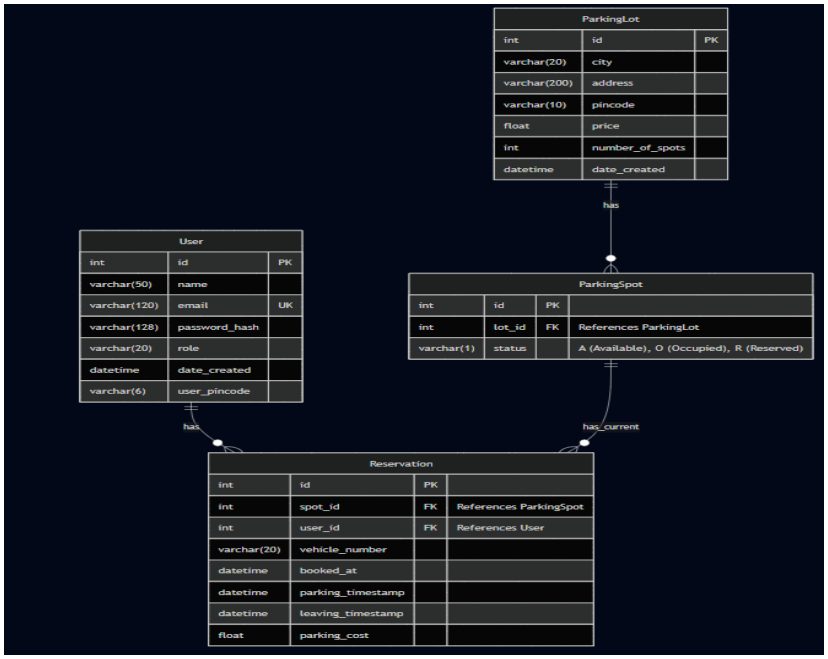
## Technologies used

**Backend**: Flask, Flask-JWT-Extended (authentication), Flask-CORS, Flask-Caching, SQLAlchemy (ORM)
**Async Jobs**: Celery + Redis (background tasks and scheduling)
**Frontend**: Vue.js (Vite) with Axios for API calls
**Database**: SQLite (simple, file-based DB for project)
**Mailing**: SMTP + MailHog (development mail server for reminders and reports)
These are the frameworks and libraries used

## DB Schema Design

**Users**: id (PK), name, email (unique), password_hash, role (admin/user), user_pincode, date_created
**ParkingLots**: id (PK), lot_name, city, date_created
**ParkingSpots**: id (PK), lot_id (FK → ParkingLots.id), status (available / occupied / reserved), date_created
**Reservations**:
id (PK), user_id (FK → Users.id), spot_id (FK → ParkingSpots.id), booked_at, leaving_timestamp, parking_cost, remarks

Following is the ER diagram:




## API Design

Implemented REST APIs for:

- **Auth** → Register, Login, JWT-based auth
- **Admin** → Manage lots/spots, view statistics
- **User** → Bookings, reservations, export CSV, receive reminders
- **Async jobs** → CSV export endpoint, monthly report, daily reminders

## Architecture and Features

**Architecture**:

- Backend (Flask) → routes/ folder for controllers, jobs/ folder for Celery tasks, models.py for DB schema, app.py initializes extensions.
- Frontend (Vue) → components/ folder with all .vue components, Axios connects to Flask APIs.
- Background Jobs → Celery workers handle scheduled tasks (beat + worker).
- Mailer → mailer.py handles SMTP with MailHog.

**Features Implemented**:

- JWT Authentication for users/admins
- Parking spot booking and reservation management
- Async Job → User-triggered CSV export of reservation history
- Scheduled Jobs → Daily reminders (if new lot) and Monthly report (user's activity)
- Performance → Flask-Caching for key APIs
- Responsive Vue.js frontend with statistics dashboard

## Video

https://drive.google.com/file/d/1tF4k-KowG6SZBlcDpvvJ0eL9QWIuiPBA/view?usp=sharing