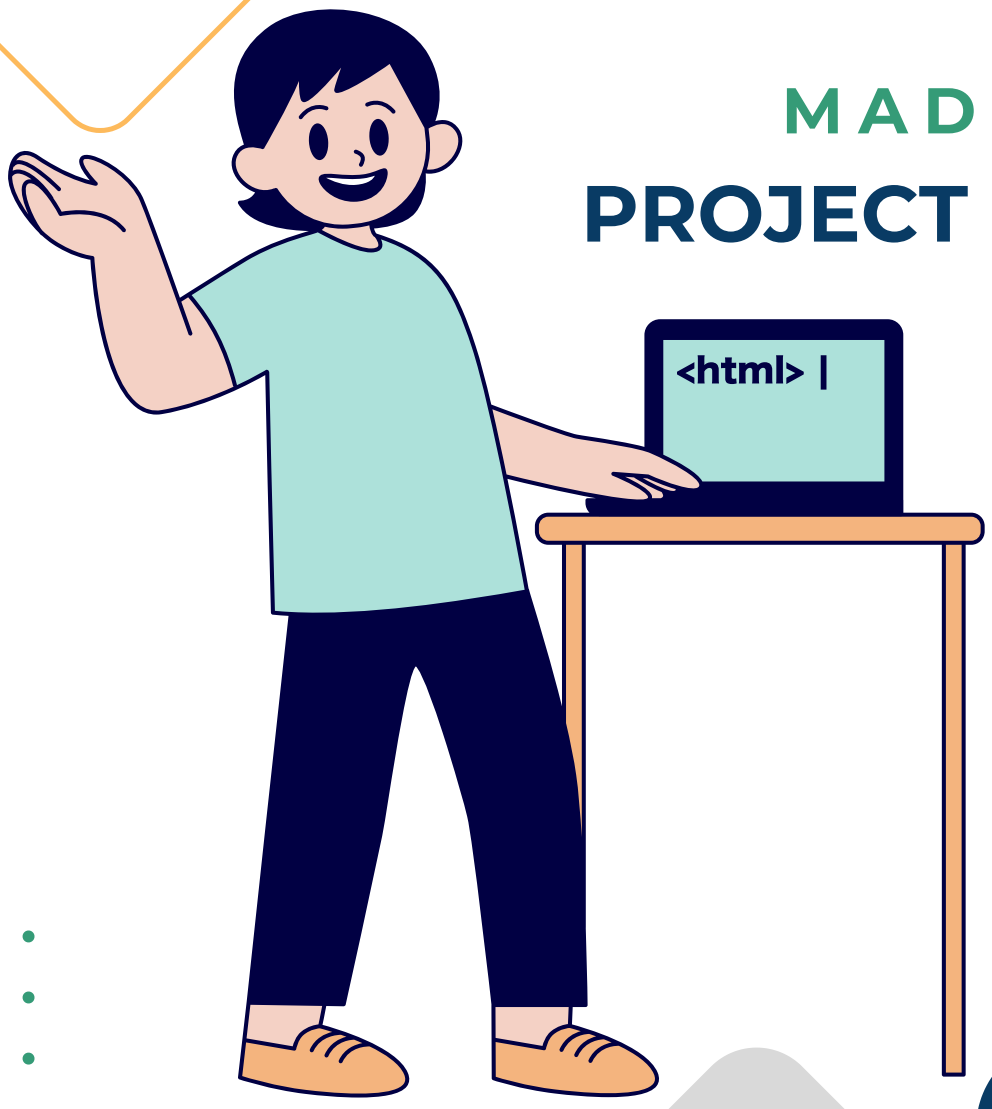
ParkZone

MAD-II

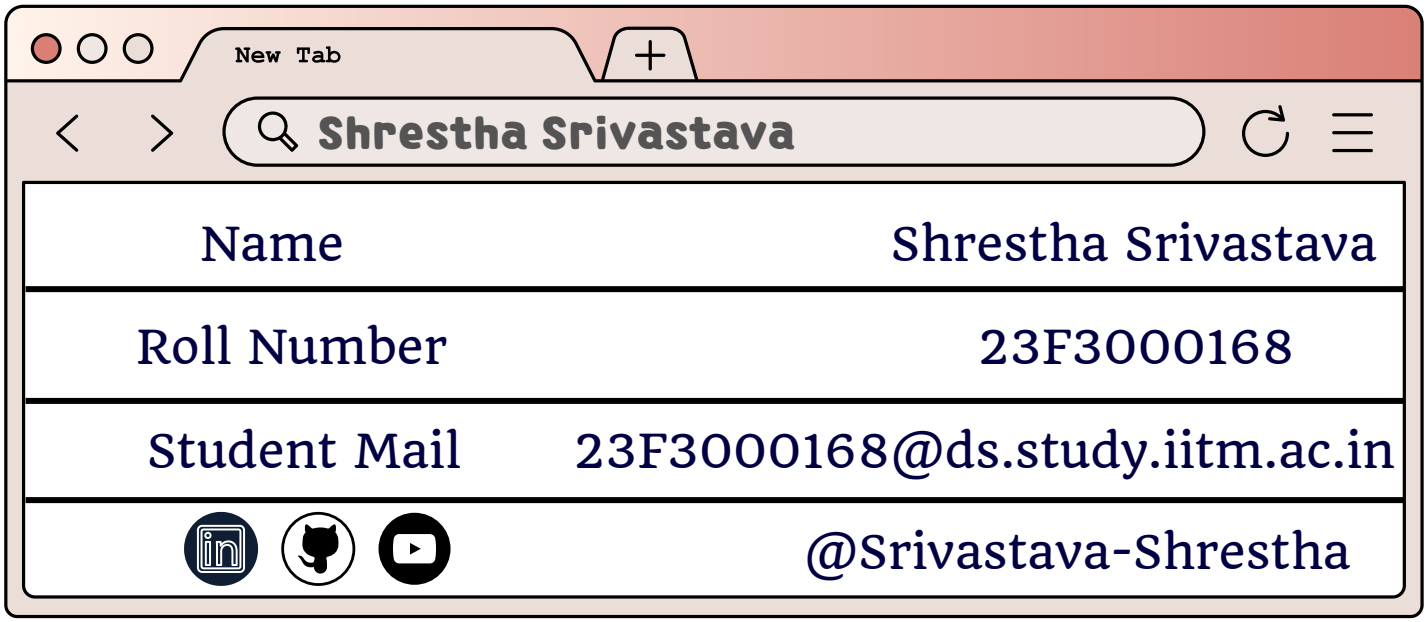
PROJECT REPORT



Prepared by :

Shrestha Srivastava

About Me



Project Details

ParkZone : Vehicle Parking App

ParkZone- is a smart, multi-user platform designed to manage 4-wheeler parking efficiently. Admins can create and update parking lots, define spot availability, set prices, and monitor all user activity through a central dashboard. Users can register, view available lots, book a spot, park their vehicle, release it, and view their parking history. The app sends daily reminders to encourage timely bookings, generates monthly usage reports with parking summaries, and allows users to export their parking data for records. It ensures seamless coordination between parking managers and users, making parking access organized, reliable, and hassle-free.

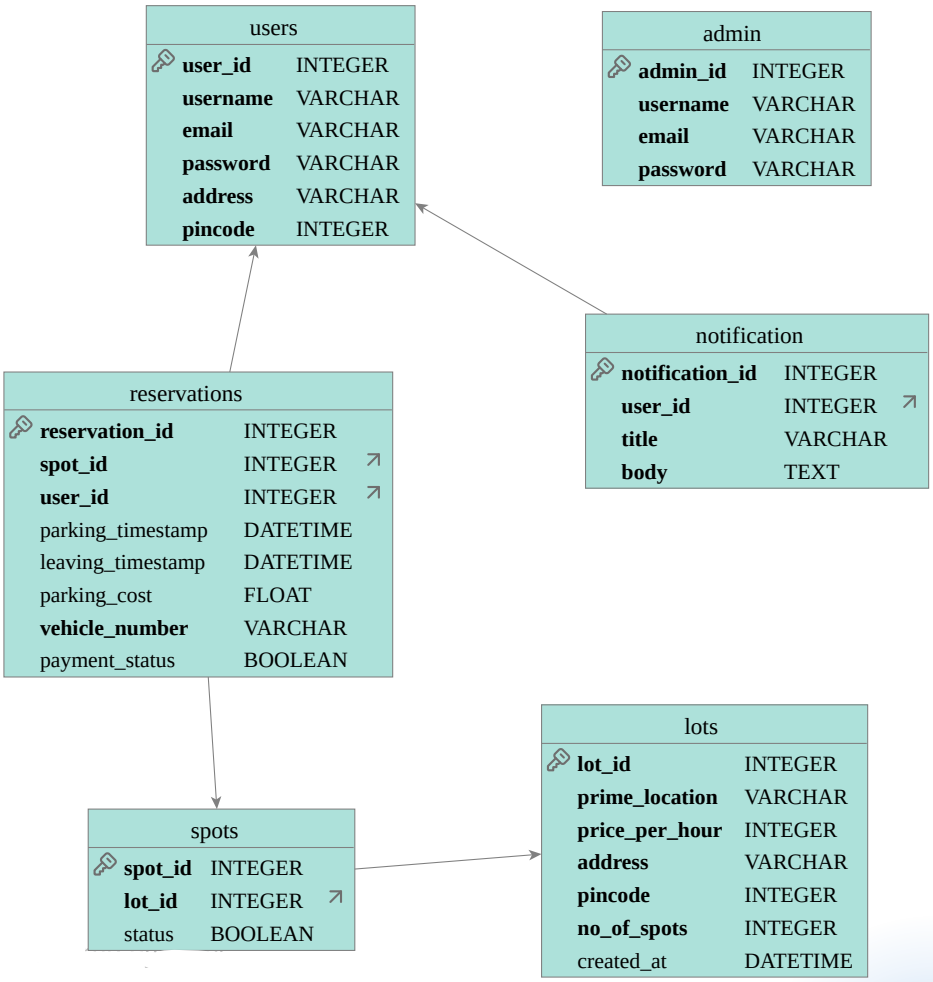
TECHNOLOGIES USED

- **Flask**: Lightweight Python web framework to handle the backend API.
- **VueJS:** JavaScript framework for building the user interface.
- **Bootstrap:** CSS framework for responsive design and styling.
- **Flask-SQLAlchemy:** ORM for managing database interactions.
- **Flask-RESTful**: For creating RESTful APIs.
- **Flask-JWT-Extended:** Handles authentication with JWT tokens.
- **Flask-Caching**: Adds caching for improved performance.
- **Celery:** Asynchronous task queue for background jobs like reminders and monthly reports.
- **Redis**: Caching layer and supports background job management.
- **SQLite**: Lightweight database for storing user data and service requests.
- **Razorpay**: Payment gateway for handling transactions.
- **Chart.js**: Library for creating interactive charts.

Briefly elaborate on what you want to discuss.

DB Schema

The ParkZone database schema is built to manage a multi-user vehicle parking system with clear role-based access and real-time tracking. The Users table stores registered user details, enabling them to book and manage parking spots, while the Admin is a predefined superuser with full control over lots, spots, and users. The Lot table holds details like location, price, address, and number of spots, each linked to multiple entries in the Spot table that track individual spot availability and status. The Reservations table logs all user bookings, capturing timestamps, cost, and status transitions from reserved to occupied to released. The schema supports efficient data management, enabling the admin to monitor lot usage and users to view their booking history. With clear relational links and timestamp tracking, ParkZone ensures a smooth, transparent, and scalable parking experience.



Directory Structure

```
└── ParkZone
├── client
│   ├── index.html
│   ├── package.json
│   ├── package-lock.json
│   ├── README.md
│   ├── src
│   │   ├── App.vue
│   │   ├── assets
│   │   ├── components
│   │   ├── composable
│   │   ├── main.js
│   │   ├── router
│   │   ├── store
│   │   ├── style.css
│   │   ├── utils.js
│   │   └── views
│   └── vite.config.js
├── README.md
└── server
    ├── app
    │   ├── api
    │   ├── config.py
    │   ├── __init__.py
    │   ├── models
    │   ├── templates
    │   └── utils
    ├── instance
    │   └── ParkZone.db
    ├── requirements.txt
    └── run.py
└── Project Report.pdf
```

API Resource

Authentication & Authorization
- Signup: /api/signup
- Login: /api/login

User Profile & Management
- User Info: /api/user
- User Detail: /api/user/<int:user_id>

Parking Lot Management
- Create/View/Update/Delete Lot: /api/lot
- Single Lot Operations: /api/lot/<int:lot_id>

Parking Spot Management
- Spot Detail: /api/spot/<int:spot_id>

Reservation Management
- Create Reservation: /api/reservation
- Reserve Spot by ID: /api/reservation/spot/<int:spot_id>
- View/Update/Delete Reservation: /api/reservation/<int:reservation_id>

Payments & Transactions
- Payment: /api/payment

Analytics & Statistics
- Dashboard Stats: /api/stats

Data Export
- Export Parking History: /api/export

AI Percentage

Category	Usage %
Frontend (HTML/JS)	20%

Presenatation Video

Google Drive :
https://drive.google.com/file/d/1wWr6dtRciizkVFLZuuna4XKoM1kQCZqZ/view?usp=sharing

Thank You
For Your Attention!

Shrestha Srivastava

30 July 2025