# Project Report

**Name**: Abhijeet kumar

**Roll Number**: 24f2007359

**Email**: 24f2007359@ds.study.iitm.ac.in

**Brief Description**: I am a Standalone Degree Student pursuing BS in Data Science.

## Parking lot App

**Project Description**:

The Parking Lot Management System is a comprehensive web application designed to streamline parking operations and provide an efficient solution for both users and administrators. The system enables users to reserve parking spots in real-time, track their parking history, and manage their reservations through an intuitive web interface. Administrators can monitor parking lot occupancy, manage multiple parking locations, view detailed analytics, and generate revenue reports.

**Key Features**:

- Real-time parking spot reservation and management
- Multi-location parking lot support with different pricing
- User authentication and role-based access control
- Comprehensive admin dashboard with analytics and reporting
- Vehicle information tracking and management
- Automated cost calculation based on parking duration
- Reservation history and user statistics
- RESTful API endpoints for data access
- Responsive web interface with modern UI/UX

**Technologies Used**

• Flask 3.0.2 - Web framework for handling requests and routing

• Flask-SQLAlchemy 3.1.1 - Database ORM and management

• Flask-Login 0.6.3 - Authentication and user session management

• Flask-WTF - Form handling and CSRF protection

• Jinja2 - Templating engine for dynamic HTML generation

• SQLAlchemy 2.0.27 - Database abstraction layer

• Werkzeug 3.0.1 - WSGI utilities and security features

• SQLite - Database storage (configurable for PostgreSQL)

• HTML5/CSS3/JavaScript - Frontend technologies

• Bootstrap - Responsive UI framework

• Chart.js - Data visualization for analytics

## DB Schema Design:

**User Table**:

- id (Integer, Primary Key) - Unique user identifier

- username (String(80), Unique, Not Null) - User login name

- password_hash (String(128)) - Hashed password for security

- role (String(20), Not Null, Default='user') - User role (user/admin)

- reservations (Relationship) - One-to-many relationship with Reservation table

**Admin Table**:

The system uses the same User table with role-based access control. Admins are distinguished by having role='admin' in the User table. This design allows for:

- Unified authentication system

- Flexible role management

- Simplified user administration

- Consistent session handling

**Additional Database Tables**:

- ParkingLot - Stores parking lot information (name, address, price, capacity)
- ParkingSpot - Individual parking spots within lots (status, lot association)
- Reservation - Parking reservations with user, vehicle, and timing details

The database design follows normalized principles with proper foreign key relationships and supports complex queries for analytics and reporting.

Project MVP : https://drive.google.com/file/d/1z-DXerncEW7jYytgSGXvnve4TZpKIS64/view?usp=sharing