VEHICLE PARKING APP PROJECT REPORT

**Author**

Name: Anish Abhyankar

Roll Number: 24F1001752

Student Email: 24f1001752@ds.study.iitm.ac.in

Greetings everyone , my name is Anish Abhyankar and I am pursuing B.Tech degree in electrical engineering from Delhi Technological University and BS degree in Data Science and Programming from IIT-Madras. I have always been fond of the tech world and able to gain deep knowledge about it from the Data Science Program

**Description**

The Vehicle Parking App Project :It is a system for booking parking spots for vehicles.It lets the user to register and login and then allows user to reserve parking spot from a number of different lots .It also gives details about the summary ,profile and parking history of the user .On the admin side it allows to add,delete and edit lots and spots and also gives all the functions of the user and the registered users also.

Technologies Used:

1. Flask-It is a lightweight web framework that makes it simple to build applications.It has minimal external dependencies ,making it easy to work with and maintain.It is used to handle routing requests and responses
2. Flask-SQLAlchemy- It is a wrapper around SQLAlchemy specifically designed to integrate with Flask applications . Object-Relational Mapper to interact with SQLite database
3. SQLite- It is a lightweight ,serverless relational database used for storing users,parking lots,parking spaces and reservations
4. Werkzeug Security- It is used for hashing and verifying passwords
5. Jinja2- It is an extensible templating enginre which have a similar coding syntax like python and hence easy to work with.It is used to render dynamic HTML pages with python variables.
6. Bootstrap- It provides inbuilt JavaScript plugins and default CSS for building responsive layouts,styling ,buttons and other UI components.
7. Matplotlib- Python plotting library used to generate bar charts showing occupancy graph and reservation summary.
8. Flask-Dotenv-It securely loads configuration variables,such as database paths and API keys from a .env file.This ensures sensitive config values like SECRET\_KEY load securely into flask.

**Database Schema and Table Descriptions**

1. User Table

* **Id** : Primary Key,unique identifier for each user
* **Username**:Unique Login credential not nullable
* **Passhash**:User’s secure hashed password ,not nullable
* **Name**:User.s full name,nullable
* **Email**:Users’s email id,unique ,nullable
* **Address**:User’s address ,nullable
* **Pincode**:User’s address pincode,nullable
* **Isadmin**: Boolean,flags if the user is admin
* Relationship : One to Many with ReserveParkingspot

**Purpose**: Stores personal and account information of all the users in the

system

1. Parkinglot Table

* **Id**: Primary Key, unique identifier for each parking lot
* **Primary\_location\_name**: Display name of the location, unique, not nullable
* **Price**: Price per hour for parking, not nullable
* **Address**: Address of the parking lot, not nullable
* **Pin\_code**: Pincode of the lot's location, not nullable
* **Maximum\_number\_of\_spots**: Total number of spaces allowed in the lot
* **Relationship**: One-to-many with Parking Space , and one-to-many with ReserveParkingspot

**Purpose**: Holds data for different parking lot locations, pricing, and their total capacity.

**3)Parkingspace Table**

* **Id**: Primary Key, unique identifier for each parking space
* **Lot\_id**: Foreign Key referencing parkinglot, not nullable
* **Status**: Indicates if the space is available or occupied, not nullable
* **Relationship**: One-to-many with Reserveparkingspot

**Purpose**:  
 Tracks each individual parking space within a lot, along with its current status (occupied/available).

1. ReserveParkingspot Table

* **Id**: Primary Key, unique identifier for each reservation
* **Lot\_id**: Foreign Key referencing, Parkinglot not nullable
* **Spot\_id**: Foreign Key referencing Parkingspace, not nullable
* **User\_id**: Foreign Key referencing user, not nullable
* **Vehicle\_number**: User's vehicle registration number, not nullable
* **Parking\_timestamp**: Time when the user parked, not nullable
* **Leaving\_timestamp**: Time when the user left the spot, nullable
* **Parking\_cost**: Calculated based on time and lot price, nullable
* **Relationship**: Many-to-one with, User,Parkingspace and Parkinglot

**Purpose**:  
Records all parking reservations, linking users to their reserved spot and lot, along with timing and cost data.

**Architecture**

**The project is organized using a MVC -Model,View,Controller architechture.The main app is initialized by app.py with configuration settings loaded from config.py**

* **Models** are defined in models.py using Flask-SQLAlchemy
* **Routes and Controllers** are defined in routes.py, handling all user and admin logic such as login, booking, and dashboard views.
* **Jinja2** was used to built the HTML templates
* **Graphs** are kept in the static folder

**Core Features**

* User authentication with secure password hashing and session management.
* Admin panel to add, edit, or delete parking lots and spaces.
* Spot booking system where users can reserve and release parking spots.
* Real-time spot status updates (available / occupied).
* Search functionality by location name or pincode.
* **Summary dashboards** for both admin (lot-wise chart) and user (booking history chart) using Matplotlib.
* **Responsive UI** using Bootstrap

AI Usage: AI was strictly used only for identifying and resolving programming errors in Python.Flask and SQLAlchemy.All code was written and reviewed manually.The usage of AI would be approximately only around 10-20% overall only for errors.

Project Video: [Click here](https://drive.google.com/file/d/10YTXOfABktBRcACMN7LRICZb51VZL7fW/view?usp=sharing)