

Project Report

Modern Application Development – I

Mayank Dhangar

23f1000599

23f1000599@ds.study.iitm.ac.in

FindMySpot Vehicle Parking App

FindMySpot is a smart vehicle parking management system built to simplify urban parking chaos. Whether you're a daily commuter or a casual driver, the app helps you locate nearby parking slots, pre-book them in real-time, and enjoy a seamless, contactless parking experience.

Key Features

- **User Roles:** Supports two distinct user roles:
 - ✓ **Admin:** Manages parking lots, spots, and monitors bookings.
 - ✓ **User or Customer:** Searches for parking, books spots, and views parking history.
- **Admin Capabilities:**
 - ✓ Add, update, and delete parking lots and spots.
 - ✓ View list of users and bookings.
 - ✓ Monitor usage statistics and parking history.
 - ✓ Manage spot availability in real-time.
- **User Capabilities:**
 - ✓ Register/login and manage their profile.
 - ✓ View available parking lots based on name, address, city or pin code.
 - ✓ Book available parking spots with vehicle details.
 - ✓ View and manage their booking history.
 - ✓ Release active bookings after the use.

Technologies and Frameworks

- **Flask:** Core web framework for backend routing and logic.
- **HTML + CSS:** For structuring and styling the frontend pages.
- **Jinja 2 Templates + Bootstrap:** Dynamic HTML rendering with responsive UI.
- **SQLite:** Lightweight relational database used for data storage.
- **Flask-SQLAlchemy:** ORM for interacting with the database.
- **Flask Login & Logout:** User session management for login/logout.
- **Flask RESTful:** For RESTful API creation and management.
- **Requests:** HTTP client used where needed for data communication.

Database Schema Summary

The database schema consists of the following core models:

1. Admin (**admin**):

- Stores admin details like **id**, **name**, **email**, **password**.
- Admin manages the system but doesn't directly interact with other entities.

2. User (**user**):

- Stores details of customers or users like **id**, **name**, **email**, **password**, **phone**, **city**, **pincode**.
- **Relationships:** Linked to multiple **Reserved Parking Spots**.

3. Parking Lot (**Parking_Lot**):

- Stores details of parking lots like **id**, **prime_location_name**, **price**, **address**, **city**, **pincode**, **maximum_number_of_spots**.
- **Relationships:** Linked to multiple **Parking Spot**, **Reserved Parking Spot**.

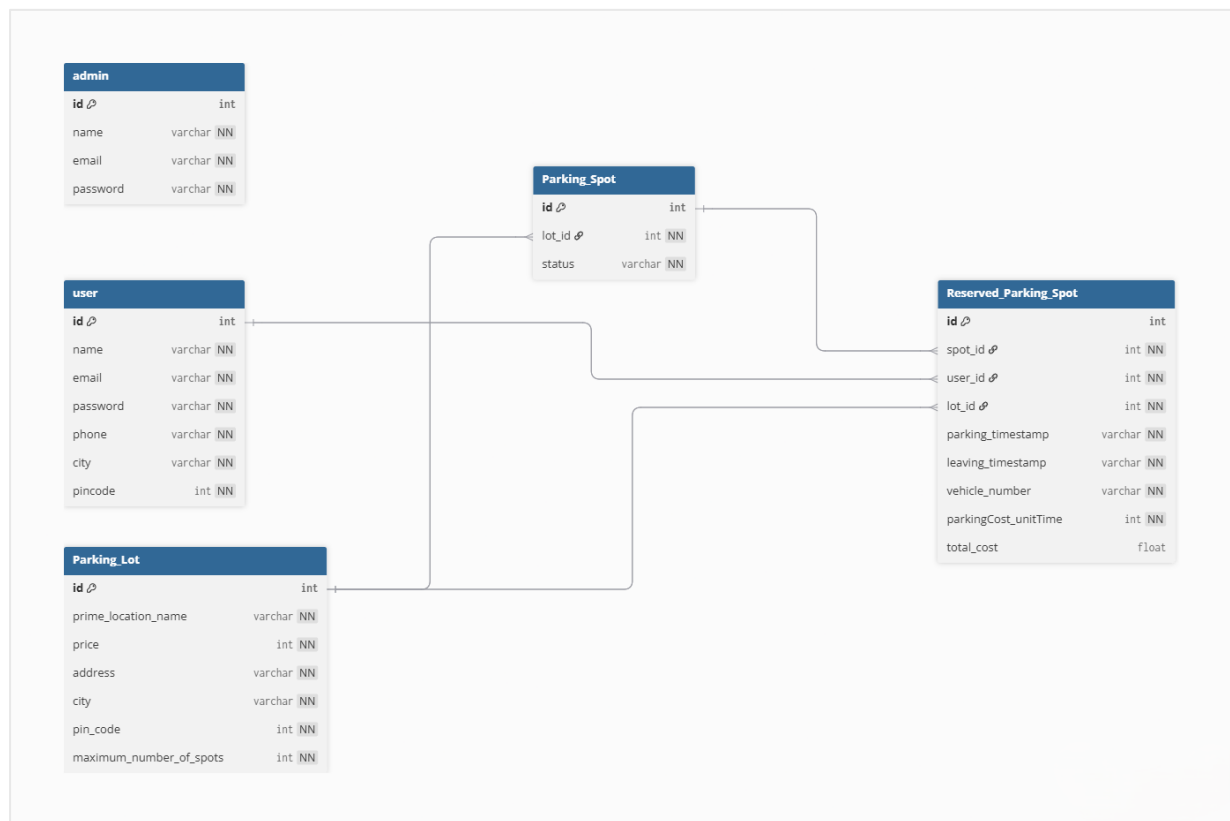
4. Parking Spot (**Parking_Spot**):

- Stores details of parking spot like **id**, **status**.
- **Relationship:** Linked to one **Reserved Parking Spot**.

5. Reserved Parking Spot (**Reserved_Parking_Spot**):

- Stores details of reserved parking spots or bookings like **id**, **parking_timestamp**, **leaving_timestamp**, **vehicle_number**, **parkingCost_unitTime**, **total_cost**.
- Linked to one **Parking Spot**, **Parking Lot** and **User**.

ER Diagram



File Structure Overview

1. Root Level

- **app.py:** Main entry point of the app.
- **.gitignore:** Contains all unnecessary files of project like cache.
- **README.md:** GitHub Project Description File.
- **Requirements.txt:** Contains all required python packages.

2. Backend (**backend/**)

- **api.py:** API endpoints of app functionality.
- **routes.py:** Managing routing of requests.
- **models.py:** Defines the database models.
- **create_data.py:** Contains initial data.

3. Instance (**instance/**)

- **vehicle_app_db.sqlite3:** SQLite database file.

4. Static (**static/**)

- Contains static assets (images, CSS, JS, etc.)

5. Templates (**templates/**)

- **Admin (**templates/admin/**)**: Admin HTML Templates.
- **User (**templates/user/**)**: User HTML Templates.
- **home.html, login.html, register.html**

Api Endpoints Used

```
api.add_resource(ParkingLotResource, "/api/parkingLot")
```

Controllers

```
@app.route("/")
@app.route("/logout")
@app.route("/register", methods = ["GET", "POST"])
@app.route("/login", methods =["GET", "POST"])
@app.route("/admin/dashboard")
@app.route("/admin/search", methods = ["GET", "POST"])
@app.route("/admin/summary")
@app.route("/user/dashboard")
@app.route("/user/search", methods = ["GET", "POST"])
@app.route('/user/summary')
@app.route("/parkingLots", methods=["POST"])
@app.route("/booking", methods=["POST"])
@app.route("/release/<int:booking_id>", methods=["POST"])
@app.route("/delete_parking/<int:lot_id>", methods=["POST"])
```

Video Link

[←-----Click Here to Access the Video----->](#)