PROJECT TITLE: VehicleBay-Vehicle Parking App-V1

COURSE: Modern Application
Development-I (MAD-I)

STUDENT NAME: Sneha Purakayastha

ROLL NUMBER: 24F3004572

EMAIL: 24f3004572@gmail.com

SUBMISSION DATE: 31/07/2025

PROJECT STATEMENT:

To design and develop a Vehicle Parking App-V1. It is a multi-user app (one requires an administrator and other users) that manages different parking lots, parking spots and parked vehicles using Flask for the backend, SQLite as the database, and Bootstrap, HTML, CSS and Jinja2 templating for the frontend. **Assuming that this parking app is for 4-wheeler parking.**

OBJECTIVE:

The application aims to simplify parking management by allowing users to reserve and vacate parking spots while providing administrators with full control over parking lots, spots, and monitoring occupancy and usage statistics.

APPROACH TO THE PROBLEM STATEMENT:

1. **Requirement Analysis:** The problem statement was studied to identify two key roles:
    - **Admin** (with full control over parking lots and spots)
    - **User** (responsible for booking and releasing spots)

2. **System And Database Design:** An **Entity-Relationship (ER) diagram** was designed to represent the relationships between users, parking lots, parking spots, and reservations. Based on this, the **SQLite database** was programmatically created.
3. **Technology Selection:** The application was developed using **Flask** for the backend, **Jinja2 templating with HTML, CSS, and Bootstrap** for the frontend, and **SQLite** for persistent data management, as specified in the project guidelines.
4. **Implementation Of Core Features:**
    - An **Admin Dashboard** was developed for creating, editing, and managing parking lots and monitoring spot occupancy, and view all the registered users.
    - A **User Dashboard** was created to allow account registration, login, automatic spot allocation when booking, and releasing of reserved spots.

- Booking logic was implemented to automatically assign the first available parking spot and releasing of reserved spots.

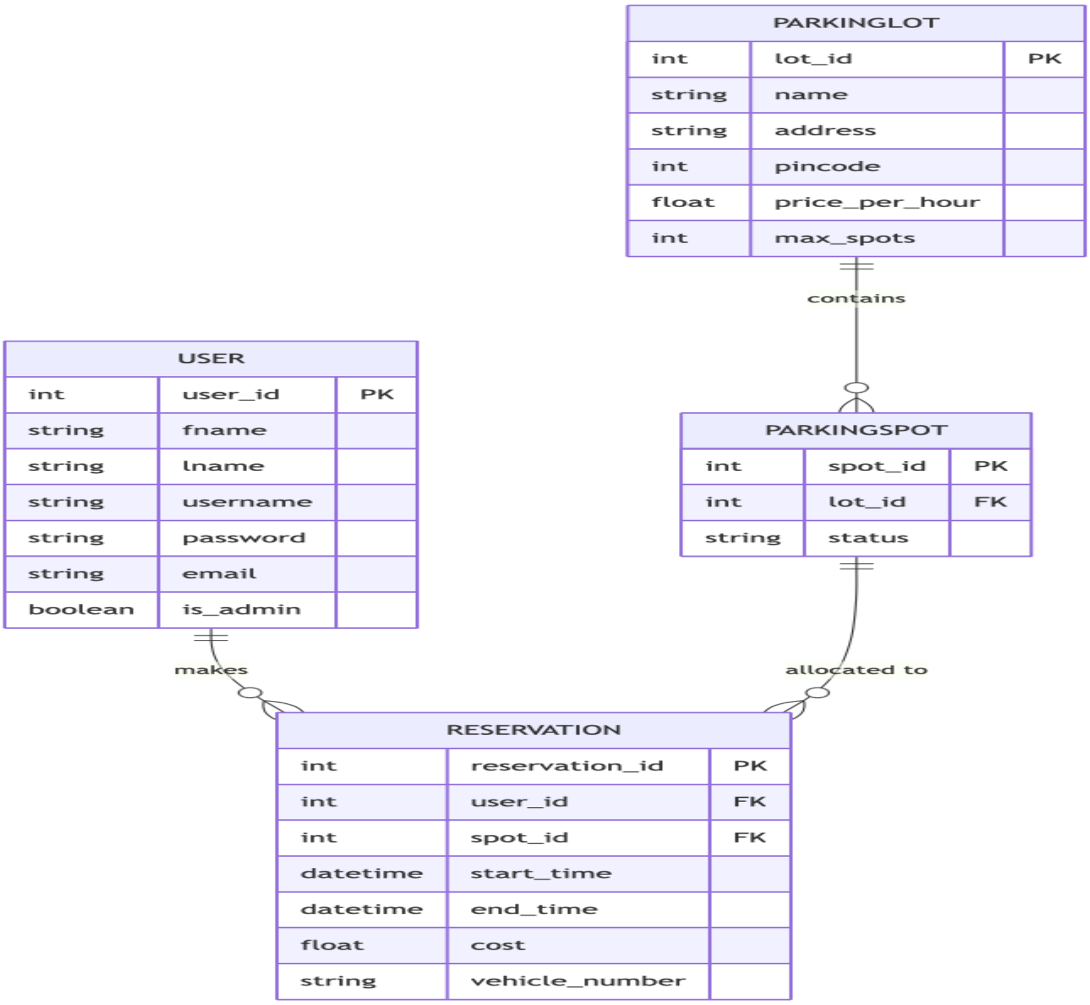
COMPONENT/TOOL	**PURPOSE/ROLE**	**AI USAGE %**	**DESCRIPTION**
Flask (App + API)	Backend logic, routing, request handling	15%	AI assisted in debugging route, while core API development was implemented manually
SQLite + SQLAlchemy	Data modelling, relationships	5%	The ER diagram was created using Mermaid Live Editor with AI; Implementation was done manually
HTML + Jinja2	Server-side template rendering	5%	AI assisted in identifying and correcting Jinja2 syntax errors; main UI design and logic were implemented manually.
Bootstrap / CSS	Styling & Layout	2%	AI was used occasionally to debug minor layout issues.
Chart.js	Displaying summary charts for admin and user dashboards	10%	AI assisted with chart configuration and debugging data binding issues; final chart design and integration were done manually.

FRAMEWORKS AND LIBRARIES USED:

The development of the **VehicleBay** application utilized the following frameworks and libraries:

- **Flask:** Used as the backend web framework to handle routing, server-side logic, and interaction with the database.
- **Jinja2 Templating:** Implemented for dynamic rendering of HTML pages with server-side data.
- **HTML and CSS:** Implemented for structuring and styling the web pages.
- **Bootstrap:** Integrated to design a responsive and user-friendly interface with prebuilt CSS components.
- **SQLite:** Implemented as the database to manage and store application data, with table creation ensuring compliance with the requirements.
- **Chart.js:** Utilized for generating visual representations of parking statistics and usage summaries.

ER DIAGRAM:



PARKINGLOT		
int	lot_id	PK
string	name	
string	address	
int	pincode	
float	price_per_hour	
int	max_spots	

USER		
int	user_id	PK
string	fname	
string	lname	
string	username	
string	password	
string	email	
boolean	is_admin	

PARKINGSPOT		
int	spot_id	PK
int	lot_id	FK
string	status	

RESERVATION		
int	reservation_id	PK
int	user_id	FK
int	spot_id	FK
datetime	start_time	
datetime	end_time	
float	cost	
string	vehicle_number	

API RESOURCE ENDPOINTS:

HTTP METHODS	ENDPOINT	DESCRIPTION
GET	/	Displays the home page
GET/POST	/login	Authenticates users via login form
GET/POST	/register	Registers a new user account
GET/POST	/admin/login	Authenticates the admin user
GET	/dashboard/<username>	Displays the user dashboard with reservations and history
GET	/view-lots/<username>	Shows all available parking lots and their availability

GET/POST	/book_lot/<lot_id>/<username>	Allows user to reserve a parking spot in the selected lot
POST	/release_spot/<reservation_id>/<username>	Releases a reserved spot and calculates the cost
GET/POST	/add_lot	Admin adds a new parking lot (with auto-created parking spots)
GET	/admin_dashboard/<username>	Displays the admin dashboard
GET/POST	/edit_lot/<lot_id>	Admins edits parking lot details (only if no occupied spots exist)
POST	/delete_lot/<lot_id>	Admin deletes a parking lot (only if no occupied spots exist)
GET	/view_users	Displays the list of registered non-admin users
GET	/parking_lot_details	Shows details of all parking lots
GET	/admin/occupied_spots	Displays currently occupied parking spots with user and time details

VIDEO LINK:

https://drive.google.com/file/d/1I2zv9dXa58obj3pjRc5Gx5wO0O07qkVy/view?usp=sharing