

Author

Name – Shreyas Pandey

Roll No – 23f3001837

Student Email – 23f3001837@ds.study.iitm.ac.in

I am a dual degree student currently at diploma level, having interest in tech and entrepreneurship, trying to build some tech business related to AI.

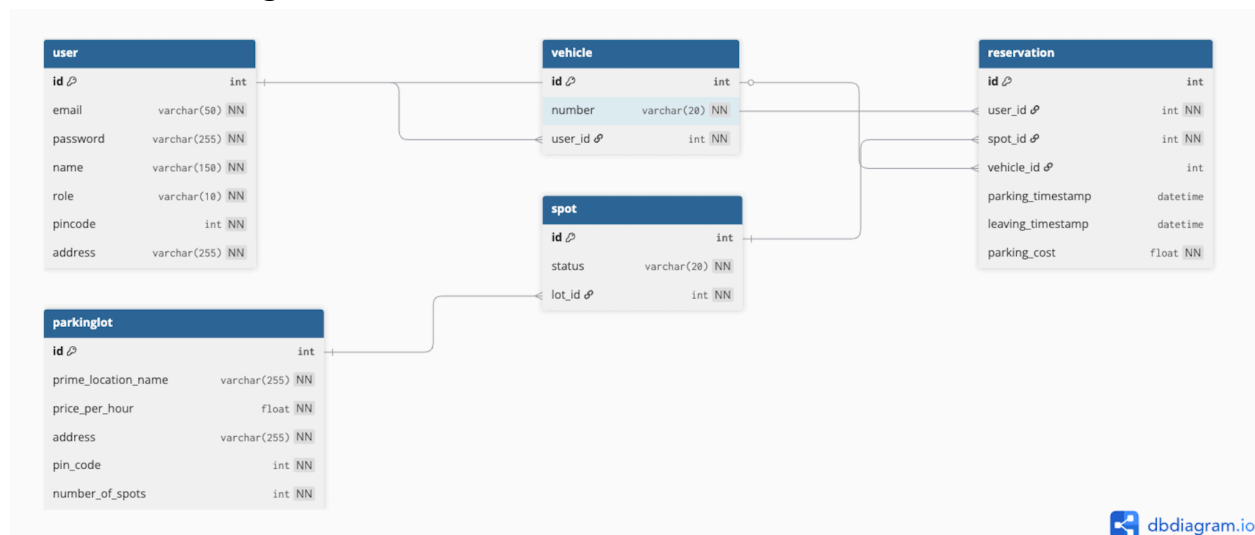
Description

Vehicle Parking App is a multi-user web app designed to manage vehicle parking in a smart and efficient way. It is built using Flask for the backend, Jinja2 for templating, HTML, CSS, and Bootstrap for frontend, and SQLite as the database. The app provides features like vehicle entry and exit logging, slot assignment, admin dashboard for monitoring, and user panel for viewing parking history. Admins can manage parking slots, view usage statistics, and generate reports.

Technologies Used

1. **Python** – Primary programming language for building the app.
2. **Flask** – Used to develop backend logic and handle HTTP requests.
3. **Flask-SQLAlchemy** – Used to define models and interact with the SQLite database, simplifying CRUD operations.
4. **Jinja2** – Used for templating by rendering dynamic HTML.
5. **Matplotlib** – Used to generate usage charts for admin insights.
6. **SQLite** – Lightweight database for storing vehicle and parking slot data.

DB Schema Design



Architecture and Features

Architecture

1. `app.py` starts the Flask app and initializes the database and admin details.
2. `controllers/` folder contains route logic for handling requests.
3. `instance/` folder stores the SQLite database.
4. `models/` folder contains `models.py` with the database schema.
5. `static/` folder includes CSS and images (e.g., for charts).
6. `templates/` folder contains all the HTML templates.

Features

1. An admin account with predefined credentials is created at the start of the app (cannot register new admin through UI).
2. Users can register with name, vehicle details, and login credentials.
3. Admin can add/edit/delete parking lots and spots and monitor slot availability.
4. Users can check parking availability and log their vehicle entry/exit.
5. Summary section for users to view their parking history.
6. Admin panel includes analytics and search functionalities to filter vehicle logs by user, date, lot, spot etc.
7. Automatic timestamp logging for vehicle entry and exit, with duration-based charge calculations (if applicable).

Video

 mad1 video.mp4

AI USAGE

Component / Tool	MAD-1 Usage %
Flask (App + API)	2%
SQLite + SQLAlchemy	1%
HTML + Jinja2	2%
Bootstrap / CSS	10%
JavaScript / AJAX (vanilla)	5%
Authentication (Login/Auth)	1%

Admin CRUD Logic	2%
User Logic (Quiz/Booking)	2%
Validation (Client + Server)	2%
Charts / Data Visualization	3%
Testing / Debugging	<1%
APIs / External Integration	2%