PROJECT REPORT FOR MAD-I

Author

NEETESH KUMAR SRIVASTAVA

DS23F1001701

23f1001701@ds.study.iitm.ac.in

I am getting more excited while exploring new things, like the MAD-I project. It's new for me, but with the help of sessions, I'm fully confident in this area of web development. Even sometimes I got stuck while developing or creating things, Krishna Sir helped me a lot.

Description

The first thing we should focus on is the project statement, which replicates the whole project, that is, what should be done and how it should be done.

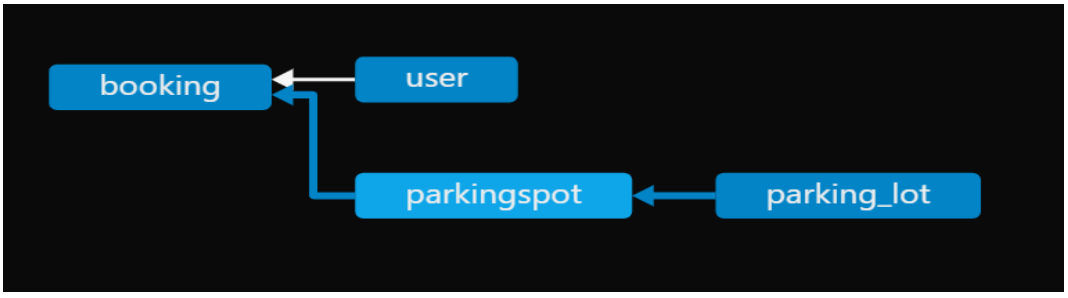
I used AI/LLM only for the CSS part to understand with the examples. For understanding purposes only.

TECHNOLOGIES USED:

I used Flask as a lightweight Python web framework, flask_sqlalchemy for database management, joinedload is a strategy to combine related data in the same query, werkzeug security for password hashing and verification, datetime for current time, base64 for encoding or decoding module(converting binary data to string), io python input or output module for converting image or chart into base64, matplotlib.pyplot for generating graph or charts, flash for temporary success or fail messages, os for file system operations like environment variables, file join, file delete and create directory.

DB Schema Design

Below is the structure of the DB, and it is a screenshot of the ER diagram from the Database.



User ⟶ Booking: One-to-many, Parkinglot ⟶ Parkingspot: One-to-many,

Parkingspot ⟶ Booking: One-to-many, User ⟶ can be an admin or a user.

PROJECT REPORT FOR MAD-I

COLUMNS:

1)user have id(primary key), username(unique),Fullname,address,pincode.

2)parkinglot have id(primary key), prime_locationname, address, pincode, price, number_of_spots

3)parkingspot have id(primary key), parking_id(foreign key), spot_number,status=A/O

4)booking have id(primary key), user_id(foreign key), spot_id(foreign key), vehicle_number, start_time, end_time, status, parking_cost

I chose this structure of the database because it is easy to understand, and basically, it best fits with the given wireframe. And it connects every table with ease.

Architecture and Features

The architecture of the project has one main file named app.py, and other folders like templates, static, which have HTML files and CSS files. In app.py, imported all requirements like modules and extensions. Models of the project mean the main database structure created in that same file, with relationships. Routes with their functions are also defined in the app.py file.

Features of the project are-> the first feature, like whenever you visit the index page, you should verify yourself whether you are an admin or a user.

If you are an admin, you get the admin home page, in which you can manage the parking done by users. Admin can view/add/delete parking lot areas and decrease the number of spots in that lot area. Admin can search for details of users and parking lot areas. Admin can view the summary of the parking details, like revenue by each parking lot area, and the number of spots available or occupied within a time frame.

If you are a user, you get a home page on which you can view or release the parking spot. The user can book an available parking spot by searching through the pincode or the address of the parking spot. The user can also view their historical parking details on their summary page.

Video link

https://drive.google.com/file/d/1oqrLYqz8GBabQzgpElAmYtncnAEHNm-D/view?usp=sharing