PROJECT REPORT - VEHICLE PARKING MANAGEMENT SYSTEM

STUDENT DETAILS:

Name: N Tejasvini

Email: 22f002775@ds.study.iitm.ac.in

PROJECT OVERVIEW:

A vehicle parking management system was developed using Flask and Jinja2 templates. The project handles parking lot management and reservations through a template-driven interface.

APPROACH:

The project was developed using Flask for the backend with Jinja2 templates for views. SQLAlchemy was used for database management, and Flask-Login handles user authentication. Bootstrap was integrated for responsive design.

TECH STACK USED:

- Backend: Flask (Python) with SQLAlchemy

- Frontend: Jinja2 templates with Bootstrap 5

- Database: SQLite

- Authentication: Flask-Login

- Forms: Flask-WTF

DATABASE DESIGN:

The system uses 4 main tables:

1. Users
   - Stores user info and authentication details
   - Has role-based access (admin/user)
2. ParkingLots
   - Contains lot details like name, location, pricing
   - Tracks total and available spots
3. ParkingSpots
   - Individual spots in each lot
   - Maintains status (available/occupied)
4. Reservations
   - Links users, spots, and payment info
   - Tracks parking duration and costs

CORE FEATURES:

Admin Side:

- Parking lot management (add/edit/delete)
- Real-time spot monitoring
- User management
- View booking records
- Manage parking spots

User Side:

- Book parking spots
- View available spots
- Track current bookings
- View booking history
- Profile management

Template Structure:

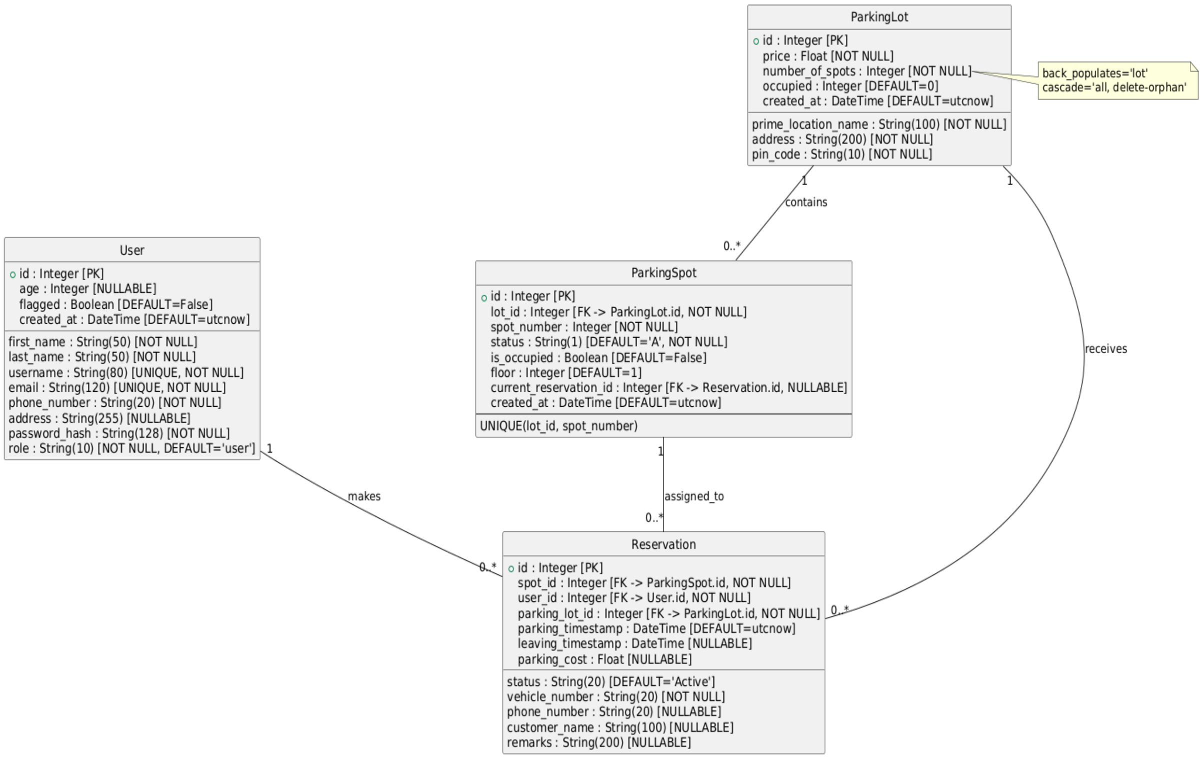
Admin Templates:

- dashboard.html
- manage_lots.html
- manage_users.html
- view_bookings.html

User Templates:

- dashboard.html
- book_spot.html
- my_bookings.html
- profile.html

ER DIAGRAM


ParkingLot
id : Integer [PK]
price : Float [NOT NULL]
number_of_spots : Integer [NOT NULL]
occupied : Integer [DEFAULT=0]
created_at : DateTime [DEFAULT=utcnow]
prime_location_name : String(100) [NOT NULL]
address : String(200) [NOT NULL]
pin_code : String(10) [NOT NULL]
back_populates='lot'
cascade='all, delete-orphan'
1
contains
0..*
1
receives
User
id : Integer [PK]
age : Integer [NULLABLE]
flagged : Boolean [DEFAULT=False]
created_at : DateTime [DEFAULT=utcnow]
first_name : String(50) [NOT NULL]
last_name : String(50) [NOT NULL]
username : String(80) [UNIQUE, NOT NULL]
email : String(120) [UNIQUE, NOT NULL]
phone_number : String(20) [NOT NULL]
address : String(255) [NULLABLE]
password_hash : String(128) [NOT NULL]
role : String(10) [NOT NULL, DEFAULT='user']
1
makes
0..*
ParkingSpot
id : Integer [PK]
lot_id : Integer [FK -> ParkingLot.id, NOT NULL]
spot_number : Integer [NOT NULL]
status : String(1) [DEFAULT='A', NOT NULL]
is_occupied : Boolean [DEFAULT=False]
floor : Integer [DEFAULT=1]
current_reservation_id : Integer [FK -> Reservation.id, NULLABLE]
created_at : DateTime [DEFAULT=utcnow]
UNIQUE(lot_id, spot_number)
1
assigned_to
0..*
Reservation
id : Integer [PK]
spot_id : Integer [FK -> ParkingSpot.id, NOT NULL]
user_id : Integer [FK -> User.id, NOT NULL]
parking_lot_id : Integer [FK -> ParkingLot.id, NOT NULL]
parking_timestamp : DateTime [DEFAULT=utcnow]
leaving_timestamp : DateTime [NULLABLE]
parking_cost : Float [NULLABLE]
status : String(20) [DEFAULT='Active']
vehicle_number : String(20) [NOT NULL]
phone_number : String(20) [NULLABLE]
customer_name : String(100) [NULLABLE]
remarks : String(200) [NULLABLE]
0..*

This ER diagram illustrates the relationships among four entities: User, ParkingLot, ParkingSpot, and Reservation. Each entity includes key attributes, with primary keys (PK) and foreign keys (FK) marked. The diagram shows how users make reservations, parking lots consist of multiple spots, and spots are linked to reservations.

Development Notes:

The project implementation applied Flask and Jinja2 concepts from the course material AI tools were used in approximately 18–20% of the project, mainly for enhancing UI styling, improving template layout, and learning advanced frontend effects (such as GASP and SVG animations). All core business logic was done independently.

VIDEO LINK : https://drive.google.com/file/d/1BAmjJ6vB3i4GD17p2X5Ed-0NsFmcn6a3/view?usp=sharing