Project Report

Author:
Your full name :THANISH P V
Roll number :24f2005577
Student email :24f2005577@ds.study.iitm.ac.in
A couple of lines about myself :HI i am Thanish P V , I am currently studying in UNIVERSITY COLLEGE MANGALORE

Description

This project implements an online parking management system for allowing users to register, book parking spots at different lots, and manage bookings through a secure web portal. Features include lot management, user authentication, booking history, and admin dashboards.

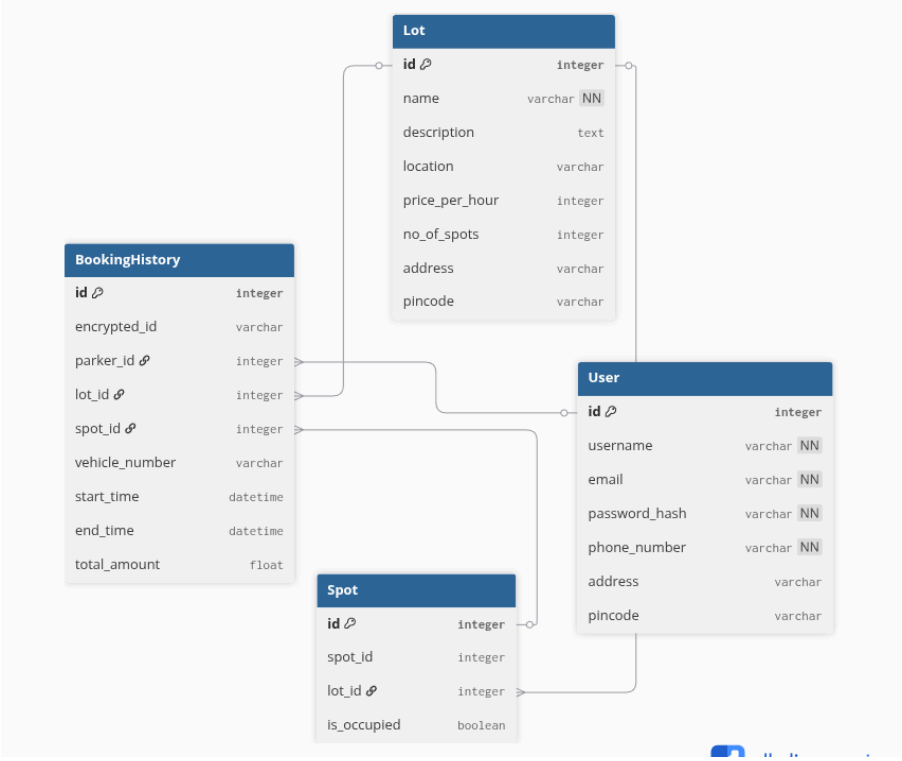
Technologies Used

- **Flask: Core backend web framework for routing and templates.**
- **Flask-SQLAlchemy: ORM for database interactions, managing tables and relationships.**
- **Flask-Bcrypt: Handles secure password hashing for user authentication.**
- **Flask-Login: Manages user sessions and authentication state.**
- **Flask-WTF: Simplifies form handling and validation.**
- **Flask-Simple-Crypt: Used for encrypting ID fields .**
- **Jinja2: Template rendering for all frontend pages.**
- **WTForms: Handles creation and validation of web forms.**
- **Bootstrap: Handles style elements of frontend pages.**
- **Chart js: Handles graph creation for summary.**

Use of AI / LLM

AI and Large Language Model (LLM) tools were used minimally throughout the project, accounting for less than 20% of total development efforts. The primary usage focused on assisting in understanding errors and suggesting solutions during the coding process. There was only limited application of AI for generating frontend code snippets and creating charts. Overall, AI contributed as a supportive tool, not as a core component, ensuring that most functionalities were designed and implemented manually.

Database Schema Design



Rationale

- The database design is normalized for efficient querying and clarity.
- Each table reflects a real-world entity or transaction.
- The encrypted_id field secures references.
- Foreign keys maintain referential integrity between related tables such as bookings, lots, users, and spots.

API Design

- User Registration and Login: Secure endpoints with hashed passwords, session management with Flask-Login.
- Lot Management: Admin endpoints for creating, updating, viewing, and deleting parking lots.
- Spot Management: Admin controls to add, remove parking spots per lot.
- Booking Operations:
  - Booking a spot
  - Ending a booking and calculating charges
  - Viewing booking history and summaries for both users and admins
- Encryption Utilities: All endpoint references use encrypted IDs for security in all frontend URLs and API transactions.
- Search APIs: Allow admin and users to search through lots, users, and bookings.

Architecture and Features

- Project Organization:
  - Routes and controllers are defined in routes.py.
  - Templates for different user interfaces are kept in the /templates directory
  - Static files for images are in the /static directory.

    - **Models and business logic are separated in /model and /function modules.**
- **Features Implemented:**
    - **User Authentication: Secure registration, login, editing profile, session management.**
    - **Admin Dashboard: Full CRUD (create, read, update, delete) operations for lots and spots, user and booking management.**
    - **Booking Workflow: Users can search, select, and book parking spots. Includes vehicle number registration and spot release with bill calculation.**
    - **Booking History: Keeps a full, searchable record of all bookings, with unique (encrypted) booking reference IDs.**
    - **Analytics: Admin and user dashboards display summarized statistics on bookings, revenue, and spot occupancy.**
    - **Form Validation: All forms have robust validation for required fields and input formats.**

GITHUB

https://github.com/24f2005577/PARKNG_ASSISTANT_V1.git