PROJECT REPORT

Modern Application Development -1

SWIFT PARK

VEHICLE PARKING MANAGEMENT SYSTEM
BY
JEBA KUMAR P
23f2004163

Author :

I am Jebakumar , currently in the diploma level of this program . I'm really into data science , neuroscience and theology . I love exploring how the brain works , how faith fits into all and how tech can make a difference . Right now , I'm also into development and love backend work and always looking to grow wherever I can.

Application Description :

This is an web-based parking management system and also an multi user application with two interfaces (user and admin) . This application allows admin to perform CRUD operations and oversee users while User can view parking lots and filter them based on their preferences and reserve parking lots. This app minimizes confusion and maximizes parking storage.
I have used 30% of AI/LLM for frontend logic and while testing and debugging , migrations and routing

Tech-Stack and Library used   :

Below are the basic tech stack and libraries that i have used in this application ,
- Flask : lightweight backend framework to handle routing and server side logic
- HTML & CSS : frontend framework for structuring and styling
- Bootstrap 5 : CSS framework used to create responsive UI
- Charts.js : to visualize parking status dynamically through charts
- SQLite : Lightweight relational database
- SQLAlchemy , werkzeug.security , Flask Login are the additional libraries used for ORM functions , handle user authentication and hashing password
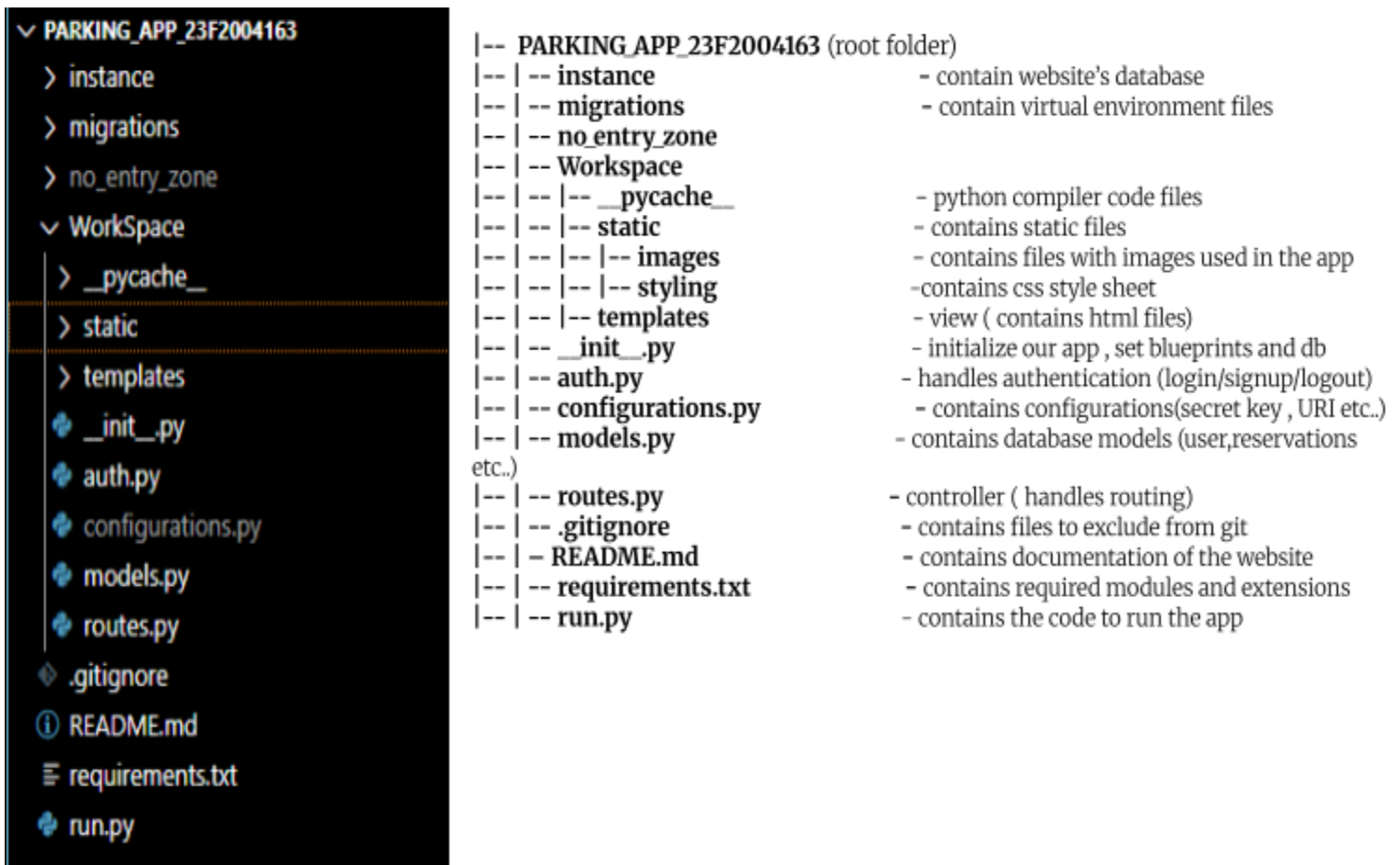
DB schema design :

There are a total of 5 database tables , it has features like foreign keys , is_deleted for soft deleting and to ensure data integrity and prevents loss of data . Overall , it facilitates parking management and reservations . Below are the schema names that i have used in my application ,
- Users   : contains user information
- ParkingLot : contains parking lot information
- ParkingSpot : contains parking spot information with lot id as foreign key
- Vehicles : contains vehicle information of the user with user id as foreign key

- Reservations : contains reservation information with user id , lot id and spot id as foreign keys
- ER Diagram link :

Architecture and Features :

MVC (Model - View - Controller) is a software architecture used to

`|-- PARKING_APP_23F2004163` (root folder)	
`|--	-- instance`	- contain website's database
`|--	-- migrations`	- contain virtual environment files
`|--	-- no_entry_zone`	
`|--	-- Workspace`	
`|--	-- |-- __pycache__`	- python compiler code files
`|--	-- |-- static`	- contains static files
`|--	-- |-- |-- images`	- contains files with images used in the app
`|--	-- |-- |-- styling`	-contains css style sheet
`|--	-- |-- templates`	- view (contains html files)
`|--	-- __init__.py`	- initialize our app , set blueprints and db
`|--	-- auth.py`	- handles authentication (login/signup/logout)
`|--	-- configurations.py`	- contains configurations(secret key , URI etc..)
`|--	-- models.py`	- contains database models (user,reservations etc..)
`|--	-- routes.py`	- controller (handles routing)
`|--	-- .gitignore`	- contains files to exclude from git
`|--	– README.md`	- contains documentation of the website
`|--	-- requirements.txt`	- contains required modules and extensions
`|--	-- run.py`	- contains the code to run the app

separate concerns in web applications. It divides an application into three interconnected components

Video presentation :