**Nexturn CapStone Project Documentation**

**on**

**Parking Management System**

**1. Introduction**

1. Project Overview

2. Purpose

**2. System Architecture**

2.1. Backend Microservices

1. AuthService

2. CheckStatus

3. PaymentService

4. ReservedParking

5. UserProfile

2.2. Database Schema

1. aspnetuser

2. masterAvailability

3. masterSlot

4. masterReserved

5. payment

6. paymentReceipt

**3. Features and Functionalities**

1. User Registration and Authentication

2. Checking Parking Slot Availability

3. Handling Payments

**4.Components**

1. Admin Dashboard

2. Admin Login

3. Availability

4. Parking Available

5. Reserve Parking

6. Payment

7. User Profile

**5.Angular Concepts Used**

1. Bootstrap

2. TypeScript

3. Routing

4. Services

5. Form Validation

3.7. Local Storage (JWT Token)

**Introduction**

1. Project Overview

This project overview provides a description of Parking Management System (PMS). It includes a explanation of what the system does, its main objectives, and why it's important.

2. Purpose

This section explains why this documentation exists. It serves as a reference guide for various including developers, project managers, and end-users.

**System Architecture**

In this section, provides an overview of the system's architecture.

2.1. Backend Microservices

1. AuthService: Describe its role in detail, including user authentication and registration.
2. CheckStatus: Explains how this microservice manages parking slot availability.
3. PaymentService: Detail its function in handling payments and receipts.
4. ReservedParking: Describe its responsibilities regarding booking management.
5. UserProfile: Explains how user profiles are managed within this microservice.

2.2. Database Schema

1. Aspnetuser: Provide details on the user data stored here.
2. MasterAvailability: Explains the structure and use of this table.
3. MasterSlot: Detail information about parking slot records.
4. MasterReserved: Describe the booking status and related data.
5. Payments : Explain what payment information is stored here.
6. PaymentReceipt : Describe the structure and purpose of payment receipts.

**3. Features and Functionalities**

1. User Registration and Authentication

* Role Management: Allow administrators to assign and manage user roles.

2. Checking Parking Slot Availability

* Map Integration: Integrate maps to display parking facility layouts and available slots visually.
* Parking Alerts: Allow users to set parking availability alerts for their preferred locations.

3. Handling Payments

* Payment History: Provide users with a detailed history of their past payments.

**4. Components**

1. Admin Dashboard

Functionality:

* + Allows admin users to create parking spots for users by specifying location and date.
  + Utilizes Angular's ngOnInit lifecycle hook.
  + Implements form submission and shows success or failure notifications.
  + Uses Bootstrap for styling.
  + Developed using TypeScript.

2. Admin Login

Functionality:

* Provides admin user login functionality.
* Uses conditional logic (if-else) for authentication.
* Stores JWT tokens in local storage for user information.
* Shows login success or failure notifications.
* Uses Bootstrap for styling.

3. Availability

Functionality:

* Enables users to search for parking availability by location and date.
* Fetches data from the backend API.
* Utilizes conditional statements to determine if parking is available.
* Transfers data to the "Parking Available" component.

4. Parking Available

Functionality:

* Handles data transfer from the "Availability" component.
* Implements data transfer methods, including input-output, RxJS, and shared services.
* Utilizes DataTables for displaying data.
* Redirects to the "Reserve Parking" component when a user clicks on the book button.

5. Reserve Parking

Functionality:

* Manages user bookings through form submissions.
* Implements form validation using Angular's FormBuilder, FormGroup, and Validators.
* Updates parking slots in the backend via HTTP requests.
* Utilizes SweetAlert for notifications.
* Handles validation and payment functionality.

6. Payment

* Functionality:
* Manages payment processes.
* Saves payment amounts.
* Redirects users to the home page (Availability) after successful payments.

7. User Profile

Functionality:

* Displays user information, including booking details and basic information.
* Retrieves data from the backend API using stored procedures.
* Allows users to cancel bookings.
* Promotes code reusability through shared models.

**5. Angular Concepts Used**

1. Bootstrap

* Bootstrap is used for responsive and consistent styling of components.

2. TypeScript

* TypeScript is the primary language for Angular development, providing strong typing and object-oriented features.

3. Routing

* Angular routing is used to navigate between different components.

4. Services

* Services handle backend logic and API calls.

5. Form Validation

* Angular's form builder, form groups, and validators are employed for form validation.

6. Local Storage (JWT Token)

* Local storage is used to store JWT tokens for user authentication.