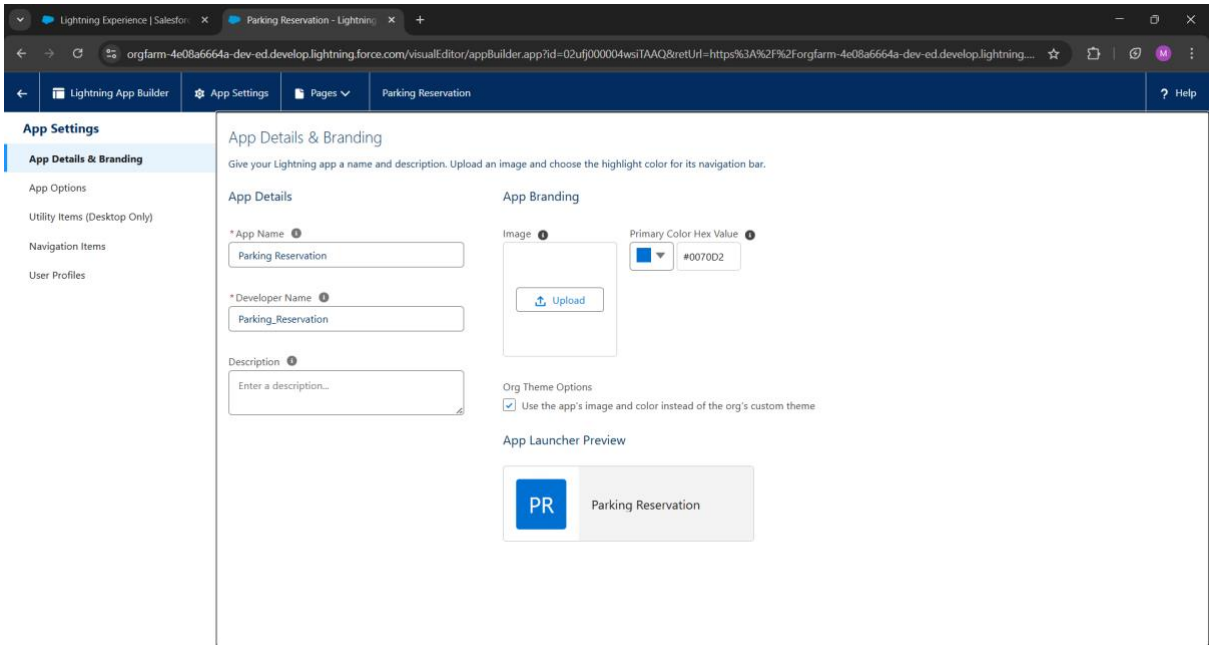
Phase 6: User Interface Development

Goal: To design and implement a user-friendly interface for the Parking Slot Reservation System in Salesforce, so employees and managers can easily interact with the system.

Step 1: Creating the Lightning App

- Open **App Manager** in Salesforce.
- Create a new Lightning App named **“Parking Reservation”**.
- Added the app branding (name, logo, navigation style).
- Assigned user profiles (Employee, Manager, Admin) to access this app.



Outcome: A dedicated workspace for Parking Slot Reservation activities.

Step 2: Designing Record Pages

- Customized **Parking Slot record page** to display:
    - Slot number, type (VIP/Guest/Employee), location, and availability status.
    - Related list of all **Reservations** linked to that slot.
- Customized **Reservation record page** to display:

- Reservation details (employee, slot, time, status).
- Approval status if required.

Outcome: Users can view and manage all slot and reservation details clearly.

Step 3: Adding Tabs for Navigation

- Created and added **two custom tabs**:
  1. **Parking Slots Tab** → List view of all available and reserved slots.
  2. **Reservations Tab** → Employees can see their reservations, managers can see all.

Outcome: Easy navigation between Parking Slots and Reservations.

Step 4: Home Page Dashboard

- Customized the **Home Page Layout** using Lightning App Builder.
- Added a **Dashboard component** that shows:
  - Total slots available vs reserved.
  - Slot utilization statistics.
  - Employee's upcoming reservations.

Outcome: At a glance view of parking space usage.

Step 5: Utility Bar for Quick Access

- Added a **Utility Bar component** with a **"New Reservation" button**.
- This opens a quick form for employees to directly reserve a slot without navigating.

Outcome: Simplifies and speeds up reservation process.

Step 6: Building Lightning Web Components (LWC)

- Developed a custom LWC named **"SlotSearchComponent"**:

    - Input fields for date & time.
    - Searches for available slots using Apex.
    - Displays results in a **Lightning Datatable**.
- Added a **"Reserve Now"** button in each row of datatable.

Outcome: Employees can search, view, and reserve slots interactively.

Step 7: Connecting Apex with LWC

- Wrote an **Apex Controller** with methods:
    - getAvailableSlots(dateTime) → Fetches available slots.
    - createReservation(slotId, employeeId, startTime, endTime) → Creates a reservation record.
- LWC makes **imperative calls** to Apex for real-time operations.

Outcome: Real-time slot booking connected to Salesforce database.

Step 8: Handling Events in LWC

- Implemented **event communication** between LWC components:
    - **Child Component** → Takes search input.
    - **Parent Component** → Receives search results and displays them.
- Used **Custom Events** for interaction.

Outcome: Smooth interaction between search and results components.

Step 9: Using Wire Adapters

- Used **@wire** to automatically fetch and refresh slot availability whenever data changes.
- Ensured that slot status updates in real-time when reservations are created or canceled.

Outcome: Live data sync without manual refresh.

Step 10: Navigation Service

- After a reservation is created, used **NavigationMixin** to redirect users to the reservation record page.
- Displays confirmation message and reservation details.

Outcome: User is immediately taken to the new reservation record for confirmation.

Final Output of Phase 6

✓ A complete **Parking Reservation Lightning App**.
✓ Employees can **search, view, and reserve slots** in real-time.
✓ Managers can **monitor utilization via dashboards**.
✓ System is **user-friendly, fast, and interactive** with LWC.