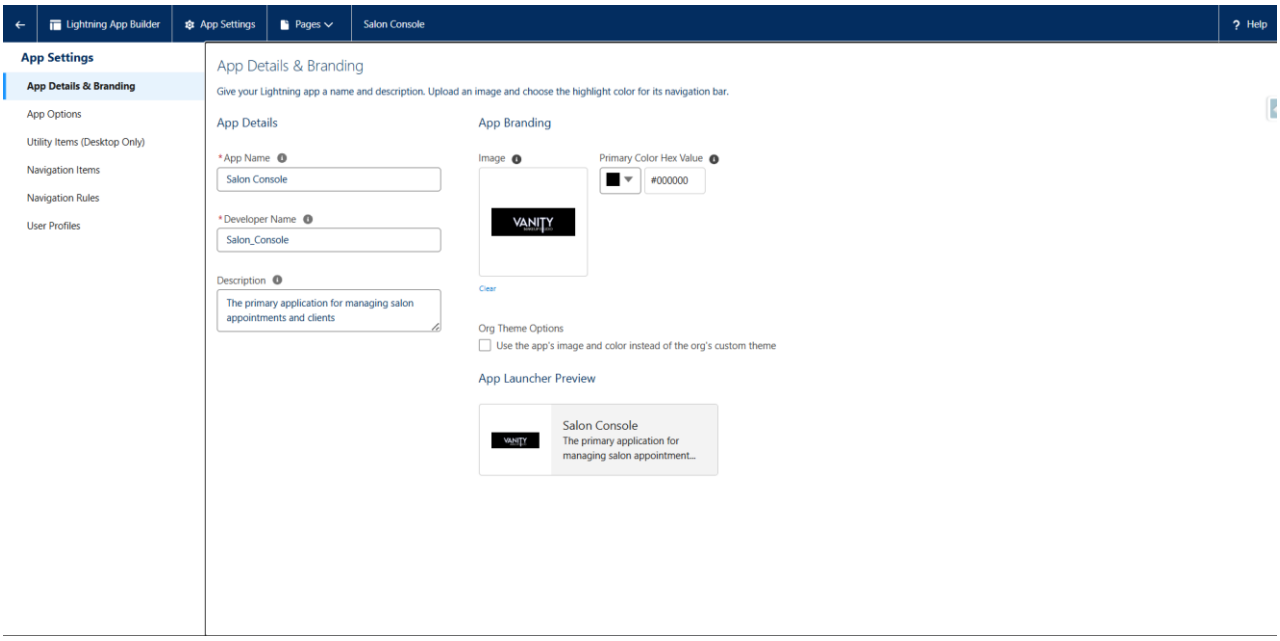
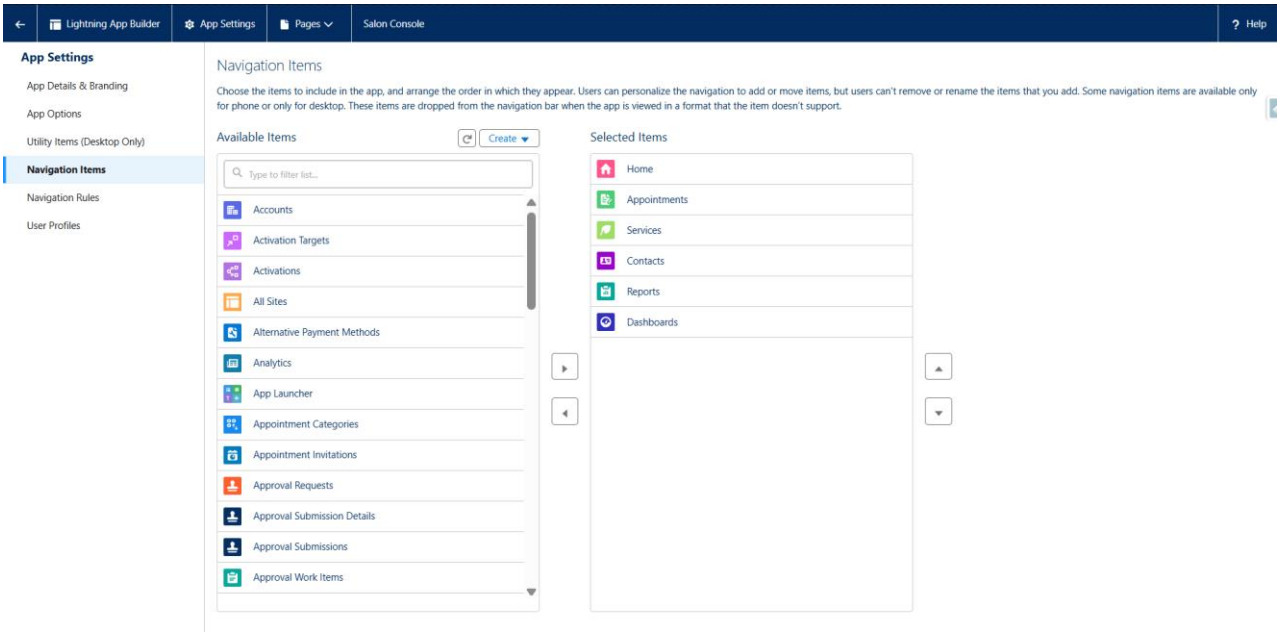
Phase 6 - User Interface Development

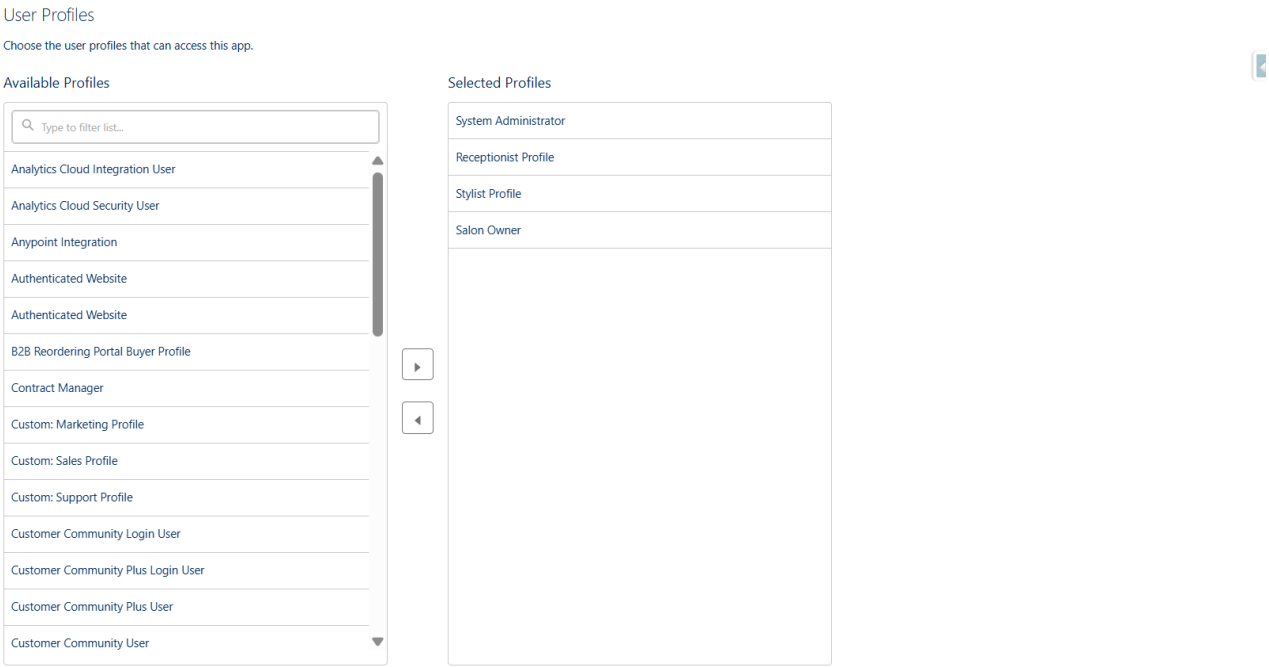
This document provides a detailed, step-by-step summary of all user interface (UI) tasks completed for Phase 6 of the Salon Management System project. The objective of this phase was to build a clean, branded, and efficient application experience for all salon users.

1. Lightning App Builder, Tabs, & Utility Bar

- **Purpose:** To create a dedicated, branded "home" for the salon's operations, moving users out of the standard "Sales" app and into a custom-built environment. This provides a focused workspace with only the necessary tools and information.
- **Steps Followed (Creating the "Salon Console" App):**
  - Navigated to **Setup** > **App Manager** and launched the **New Lightning App** wizard.
  - **App Details:** Named the application `Salon Console` and provided a description.
  - **Navigation Style:** Selected **Console** navigation to allow users (especially the busy receptionist) to open multiple records in separate workspace tabs.
  - **Utility Bar:** Used the **Add Utility Item** feature to add the **Recent Items** component to the app's footer, giving all users a quick way to access their record history.
  - **Tabs:** From the "Navigation Items" screen, the **Appointments**, **Services**, and **Contacts** tabs were selected and ordered to be the primary navigation for the application.
  - **Profile Assignment:** The new `Salon Console` app was assigned to the **Receptionist Profile**, **Stylist Profile**, and **System Administrator** profiles, making it visible and accessible to all project users in the App Launcher.







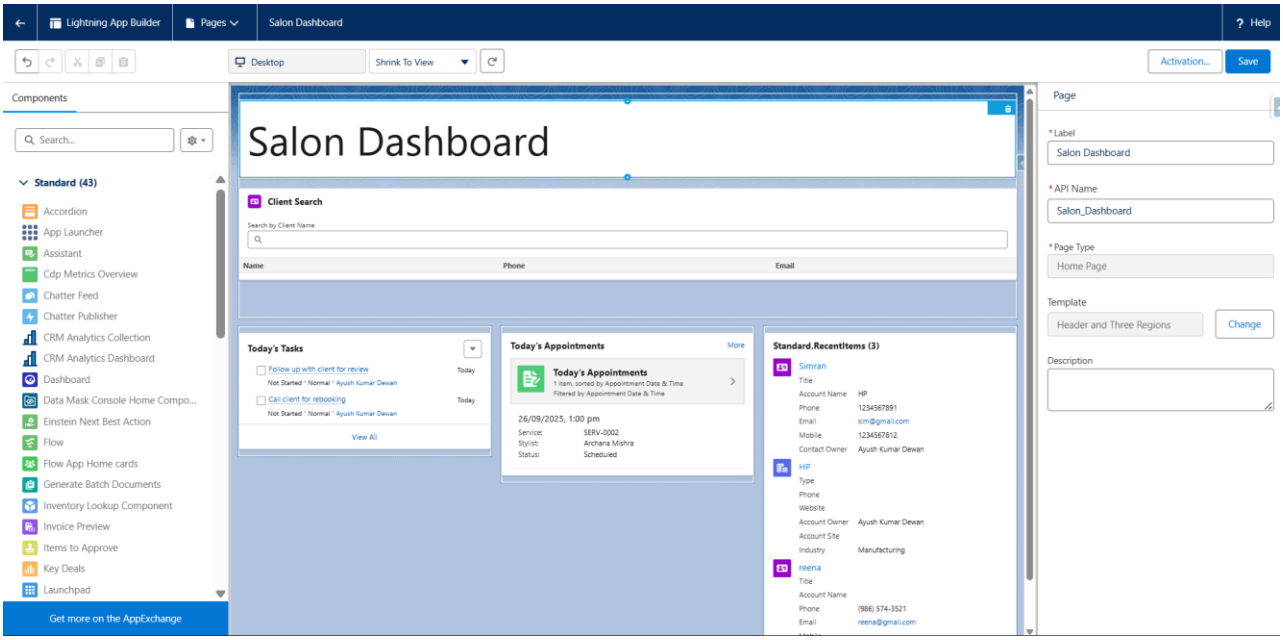
---

2. Record Pages

- **Purpose:** To customize the layout of a record's detail page to be more logical and efficient. The goal was to place the most important information in high-visibility areas for the user.
- **Steps** Followed (Customizing the Appointment **Page):**
    1. Navigated to an existing `Appointment` record within the new `Salon Console` app.
    2. Launched the **Lightning App Builder** by clicking the **gear icon ⚙ > Edit Page**.
    3. **Template Changed:** The page's template was changed from the default to **"Header and Right Sidebar"** to create a more organized two-column layout.
    4. **Components Organized:**

- The standard **Record Detail** component was placed in the main (left) column to display all the fields from the page layout.
- A **Related** List - **Single** component was added to the sidebar on the right. This component was configured to specifically display the **Approval History** related list, making it easy for users to see the status of discount requests.

5. **Activation:** The new, customized page was saved and **Activated**, then set as the **Org Default** for all users.



3. Home Page Layouts

- **Purpose:** To create a tailored "dashboard" experience for the primary user, the receptionist. Instead of a generic home page, this custom layout provides immediate access to the most relevant information for their daily tasks.
- **Steps Followed (Creating the Receptionist Home Page):**
  1. **Prerequisite Created:** A new `Appointment` **List View** named `Today's Bookings` was created. It was filtered to show only records where the `Status` is "Scheduled" and the `Appointment` Date & `Time` is for `TODAY`.
  2. Navigated to the **Home** tab and launched the **Lightning App Builder** (`Edit Page`).
  3. The existing Home Page was **cloned** to create a new, editable version named `Receptionist Home Page`.
  4. **Components Added:** The new page was designed by dragging and dropping key components onto the canvas:
     - A **List View** component was configured to show the `Today's Bookings` list.
     - The standard **Tasks** component was added so the receptionist can see their follow-up and rebooking tasks.
     - The **Recent Items** component was added for quick navigation.
  5. **Activation & Assignment:** The new page was saved and **Activated**. Using the "App and Profile" assignment option, it was specifically assigned to the **`Salon`**

`Console` app for users with the `Receptionist Profile`. This ensures other users (like the Stylist) will still see the standard home page.

4. LWC (Lightning Web Components), Apex with LWC, Events, etc.

- **Purpose:** To understand the role of Lightning Web Components (LWC) as the professional, code-based framework for building custom user interface components that are not available with the standard drag-and-drop tools.
- **Steps Followed:** For this project, mastering the declarative UI tools (Lightning App Builder, Page Layouts, Home Pages) was the primary goal. **No LWC code was written.** The key learning was identifying the use case where LWC *would* be the correct tool—for example, building