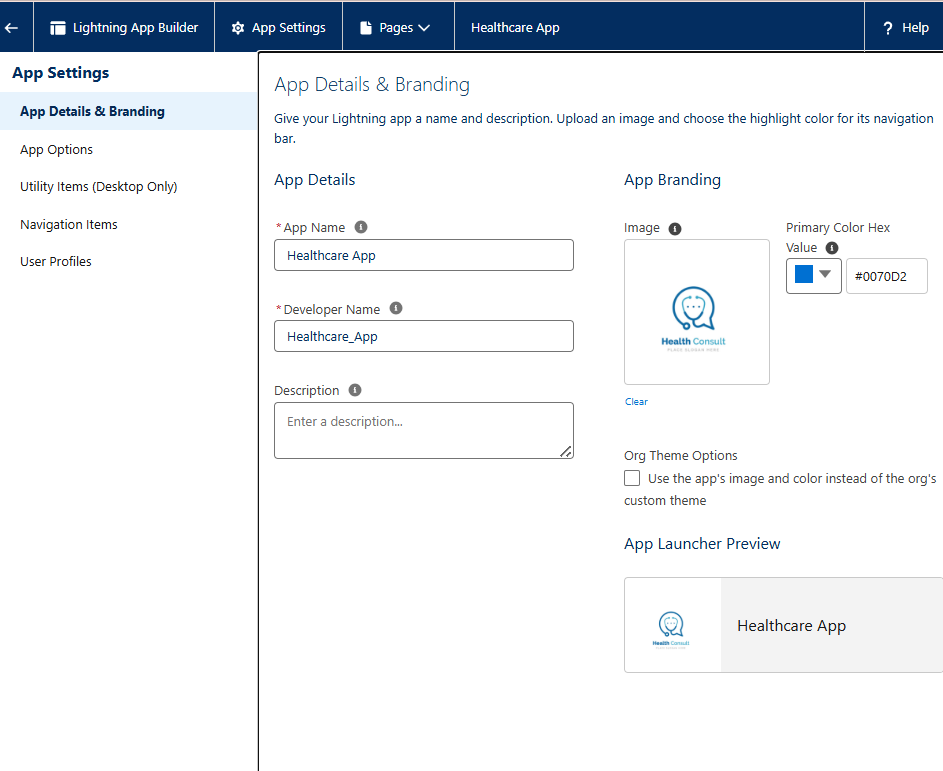
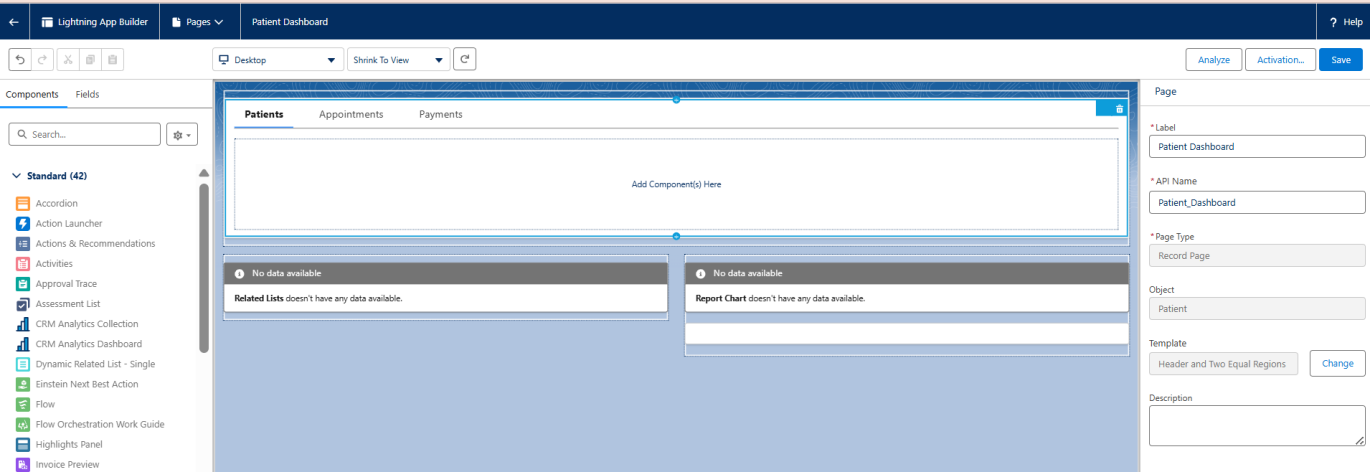
**Project : Healthcare Telecommunication Systems**

***Phase 6: User Interface Development***

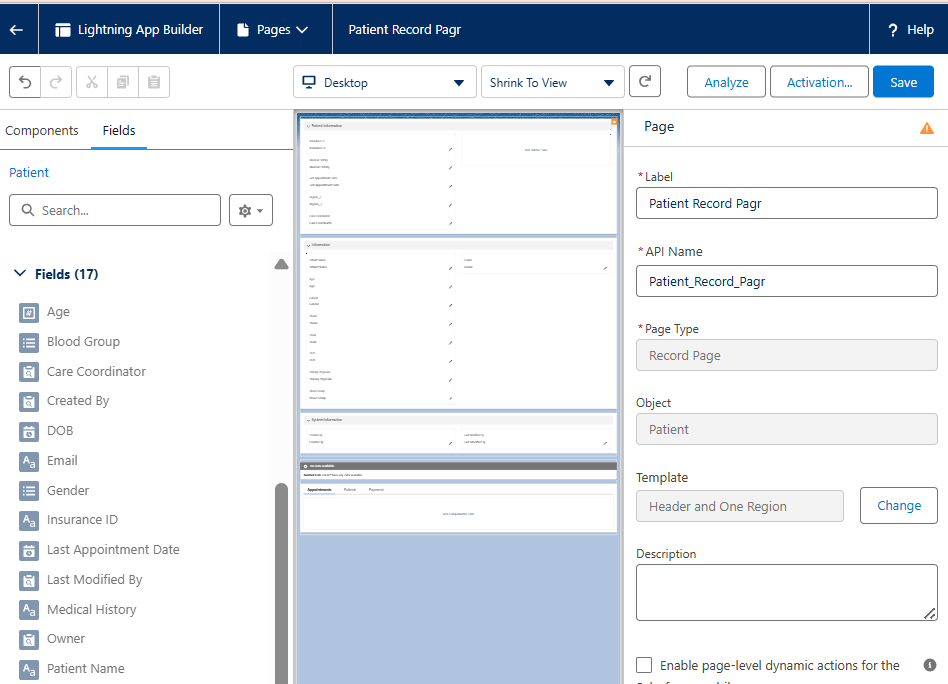
1. ***Lightning App Builder***
2. Lightning App Builder — Create the App Open App Manager.
3. Go to Setup in Salesforce ---> In the Quick Find box, type App Manager ---> Click App Manager.
4. Click New Lightning App ---> Name the App as HealthCare App



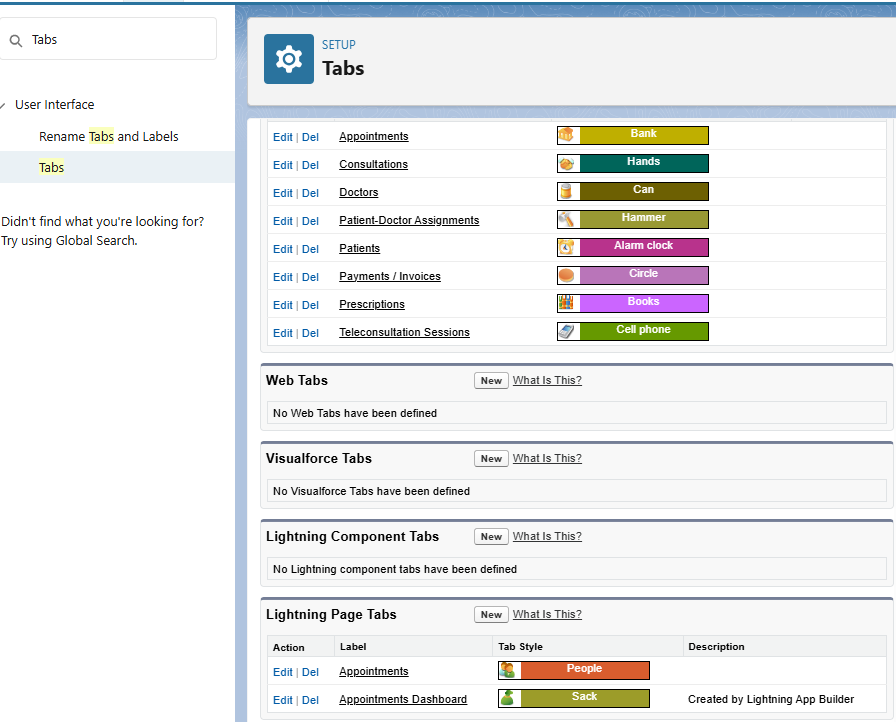
1. Open Lightning App Builder.
2. Go to Setup → Lightning App Builder --> Click New.



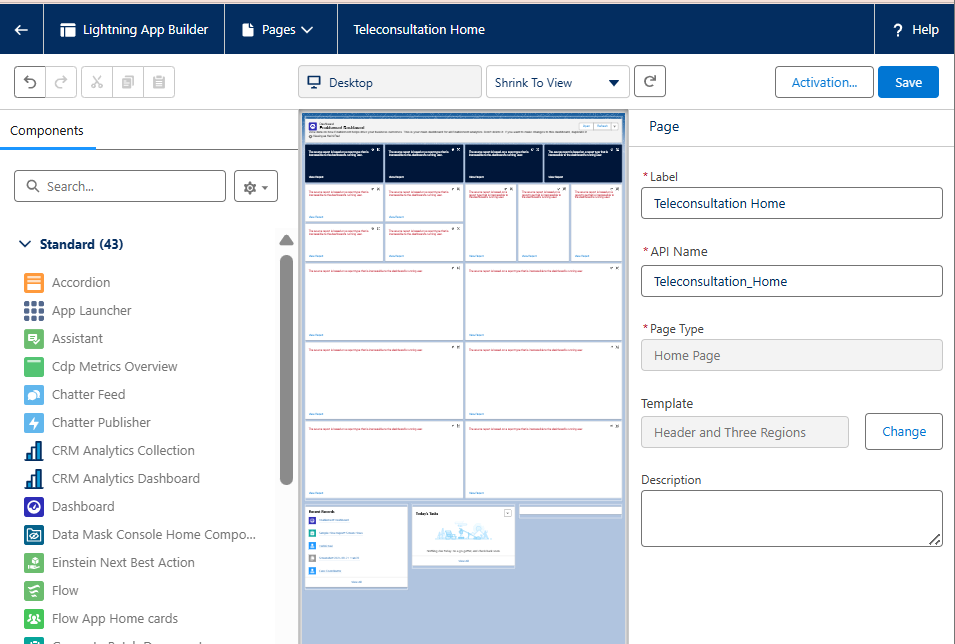
1. ***Home Page Layouts***
2. Go to **Setup → Object Manager → [Object] → Lightning Record Pages → New.**
3. Choose a **Template** (Header + Tabs or One Column).
4. Drag **Standard and Custom Components** onto the page.
5. Add **Related Lists, Tabs, Quick Actions**.
6. Save & activate for **Org Default / App / Profile**.



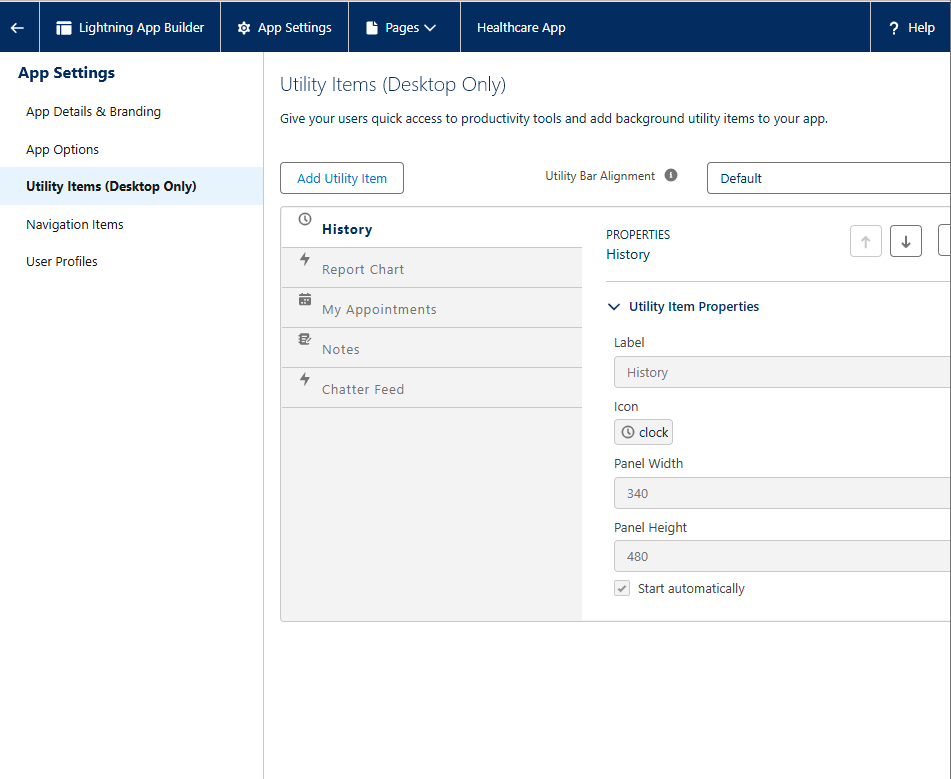
1. ***Tabs***
2. Go to **Setup → Tabs → Lightning Page Tabs**.
3. Click **New** and choose the **Object or Lightning Page**.
4. Name the tab (e.g., Appointments) and assign an **icon/color**.
5. Add tabs to **Navigation Menu** in Lightning App Builder.



1. ***Home Page Layouts***
2. Go to **Setup → Lightning App Builder → Home Page → New**.
3. Choose a **Template** (Standard Home Page or Console Home).
4. Add **Components**: Dashboard charts, Recent Records, Tasks, News.
5. Assign **Activation**: Org Default, App Default, Profile Default.



1. ***Utility Bar***
2. Go to **Setup**
3. In **Quick Find**, type **App Manager** → Click **App Manager**.
4. Find your **Healthcare Teleconsultation Lightning App**.
5. Click the dropdown → **Edit**.
6. Choose Utility Item Type ---> Report chart, My Appointments, Notes, Chatter Feed.



1. **Custom LWC**
2. Create Apex Controller for creating LWC
3. **Code :**

public with sharing class AppointmentController {

@AuraEnabled(cacheable=true)

public static List<Appointment\_\_c> getTodaysAppointments() {

Date today = Date.today();

Id currentUserId = UserInfo.getUserId();

return [SELECT Id, Name, Appointment\_Date\_\_c, Patient\_\_r.Name

FROM Appointment\_\_c

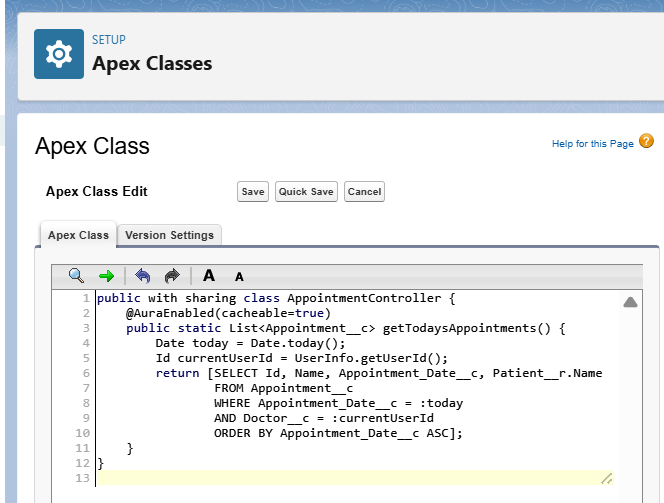
WHERE Appointment\_Date\_\_c = :today

AND Doctor\_\_c = :currentUserId

ORDER BY Appointment\_Date\_\_c ASC];

}

}



1. **Create LWC**
2. Open a terminal inside your Salesforce DX project folder and run:

**Code :**

sfdx force:lightning:component:create \

--type lwc \

--componentname patientCard \

--outputdir force-app/main/default/lwc

1. **This generates a folder:**

force-app/main/default/lwc/patientCard/

├── patientCard.html

├── patientCard.js

├── patientCard.js-meta.xml

└── patientCard.css (optional, create if needed)

**todayAppointments.html**

<template>

<lightning-card title="Today’s Appointments">

<template if:true={appointments.data}>

<ul>

<template for:each={appointments.data} for:item="appt">

<li key={appt.Id}>

{appt.Appointment\_Date\_\_c} - {appt.Patient\_\_r.Name}

</li>

</template>

</ul>

</template>

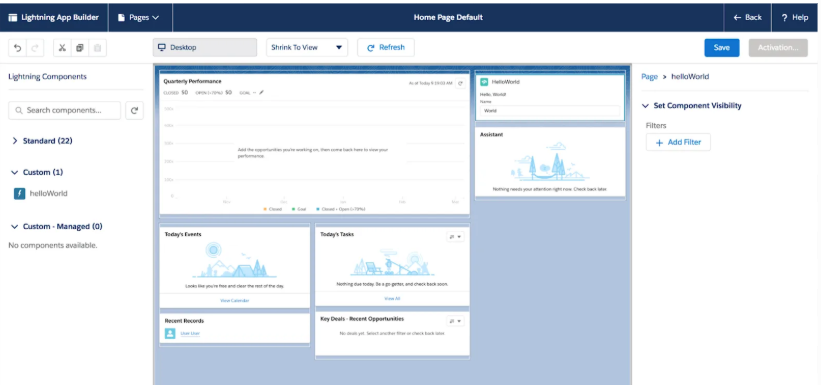
<template if:true={appointments.error}>

<p>Error loading appointments.</p>

</template>

</lightning-card>

</template>



1. ***Apex with LWC***

Create an **Apex class** annotated with @AuraEnabled(cacheable=true) if it returns data.

**Name :** PatientController

**Code :**

public with sharing class PatientController {

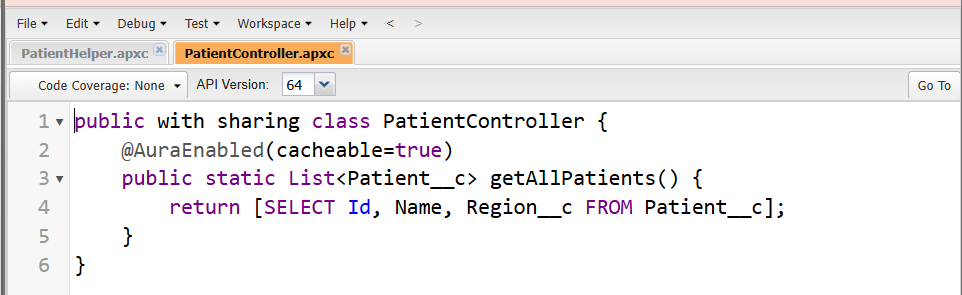
@AuraEnabled(cacheable=true)

public static List<Patient\_\_c> getAllPatients() {

return [SELECT Id, Name, Region\_\_c FROM Patient\_\_c];

}

}



1. **Events in LWC**

Sibling or Cross-Component Communication (Lightning Message Service – LMS)

1. Create a Message Channel:
2. Setup → Lightning Message Channels → New → PatientSelectionChannel.
3. Add a single field: e.g., selectedPatientId

**Publish Message from Component A:**

import { LightningElement } from 'lwc';

import { publish, MessageContext } from 'lightning/messageService';

import PATIENT\_CHANNEL from '@salesforce/messageChannel/PatientSelectionChannel\_\_c';

export default class ComponentA extends LightningElement {

@wire(MessageContext)

messageContext;

handleSelect(patientId) {

const message = { selectedPatientId: patientId };

publish(this.messageContext, PATIENT\_CHANNEL, message);

}

}

**Subscribe in Component B:**

import { LightningElement, wire } from 'lwc';

import { subscribe, MessageContext } from 'lightning/messageService';

import PATIENT\_CHANNEL from '@salesforce/messageChannel/PatientSelectionChannel\_\_c';

export default class ComponentB extends LightningElement {

@wire(MessageContext)

messageContext;

connectedCallback() {

this.subscription = subscribe(this.messageContext, PATIENT\_CHANNEL, (message) => {

this.selectedPatientId = message.selectedPatientId;

// Handle selected patient data

});

}

}