# Phase 6: User Interface Development (Migrant Health CRM)

## Objective

The goal of Phase 6 is to design and implement a user-friendly Lightning interface for the Migrant Health CRM system. This interface allows healthcare field workers, case managers, and administrators to efficiently manage patients, appointments, and migrant records through Salesforce.

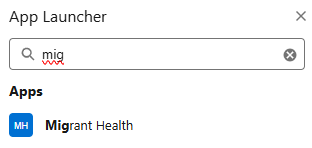
## Deliverables

* Lightning App: Migrant Health CRM with navigation items for Patients, Appointments, Migrant Records, and Reports.
* Custom Tabs: Tabs for each custom object (Patient\_\_c, Appointment\_\_c, Migrant\_Record\_Number\_\_c).
* Lightning Record Page: Redesigned Patient record page with related lists and custom component integration.
* Quick Action: Launch the Patient Intake Flow from the Patient record.
* Lightning Web Component (LWC): patientSearch component to search patients and create appointments.
* Apex Controller: PatientController class to support search and appointment creation.

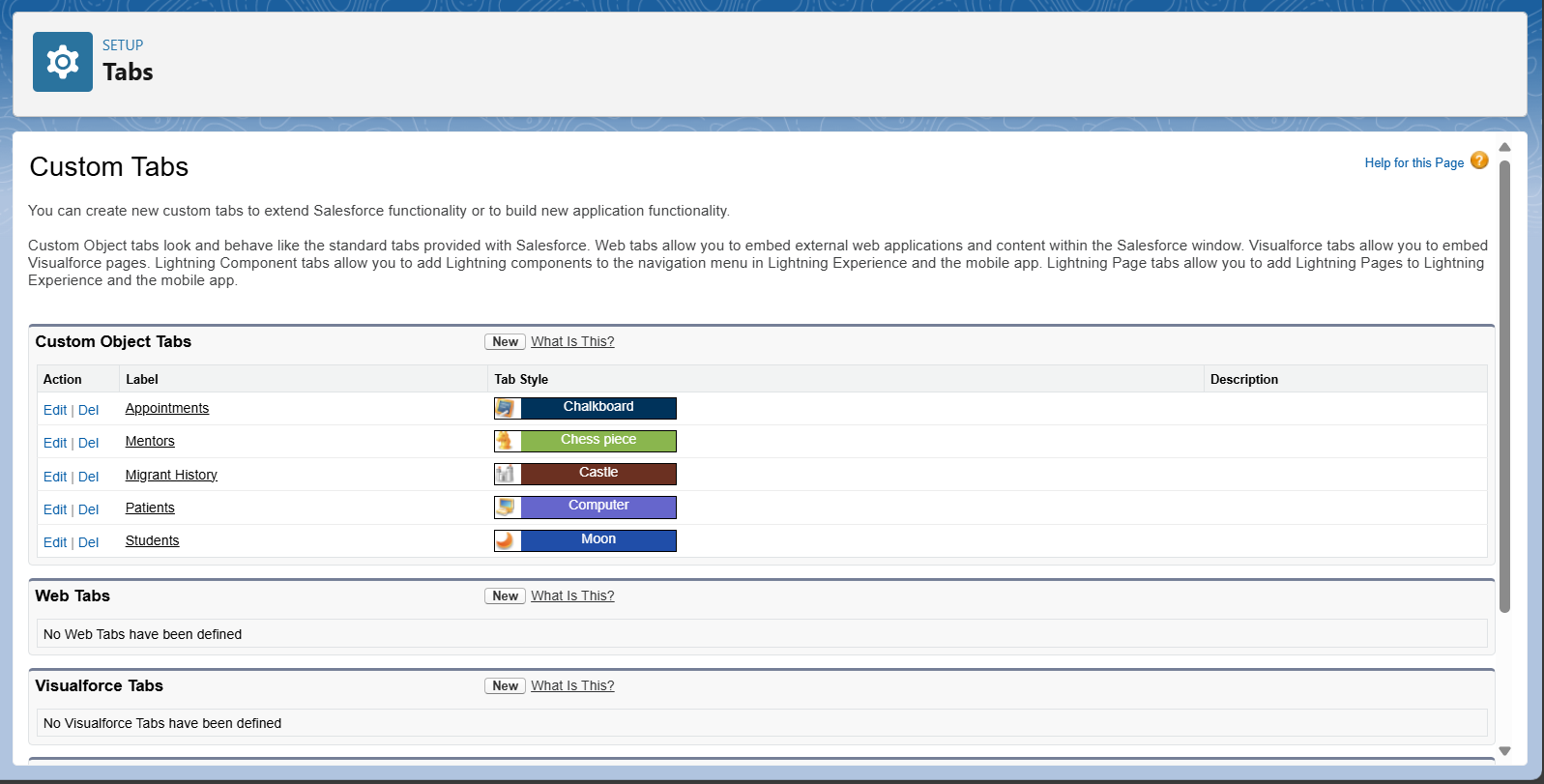
## Step-by-Step Implementation

### A. Create the Migrant Health CRM Lightning App

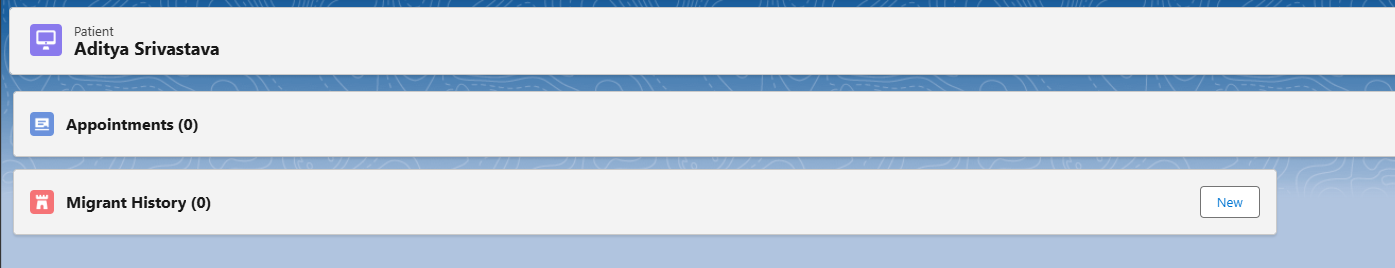
1. Navigate to Setup → App Manager → New Lightning App.
2. Provide details: Name: Migrant Health CRM, Description: Migrant Health CRM — migrant health checkup
3. Add navigation items: Patients, Appointments, Migrant Records, Reports.
4. Optional: Add a Utility Bar item for New Patient Intake Flow.
5. Assign the app to the System Administrator profile.



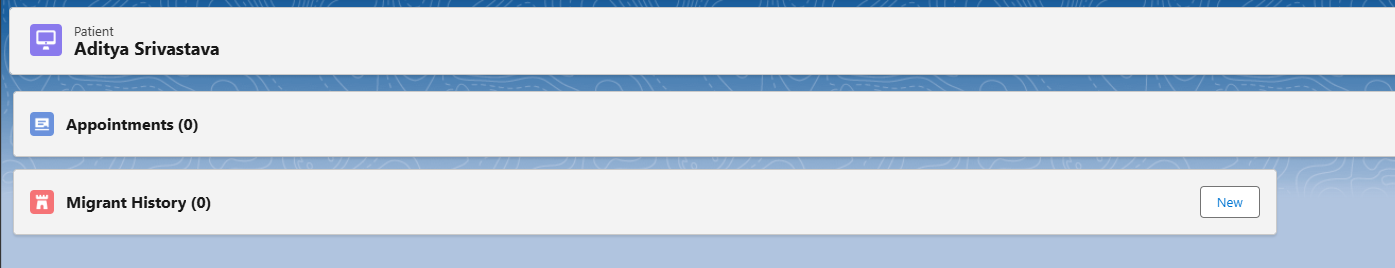
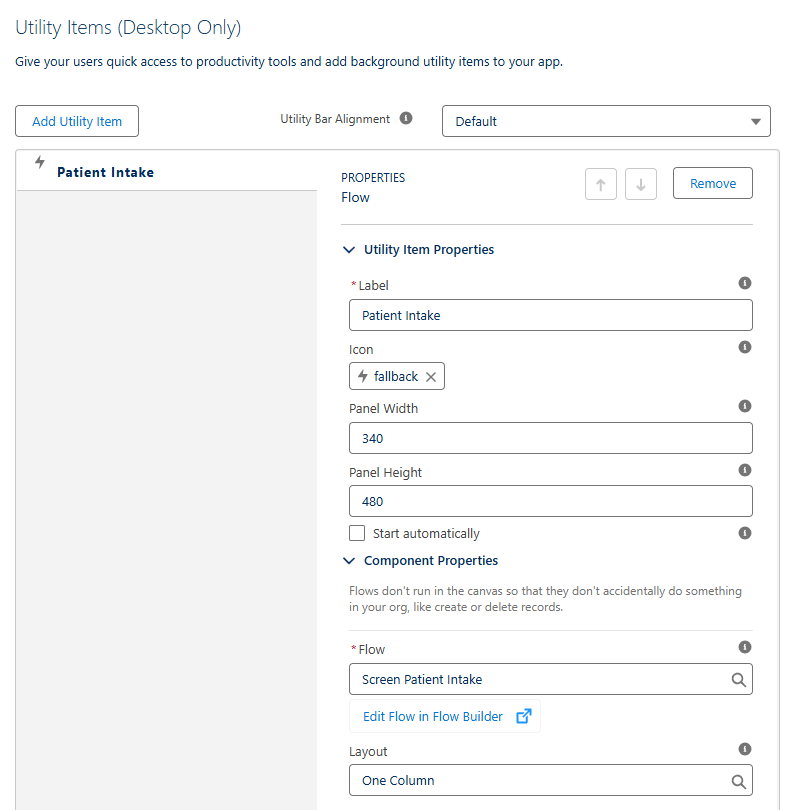
### B. Create Tabs for Custom Objects

1. Setup → Tabs → New.
2. Create tabs for: Patient\_\_c, Appointment\_\_c, Migrant\_Record\_Number\_\_c.
3. 

### C. Build the Patient Lightning Record Page

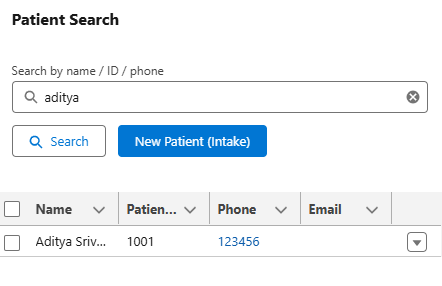
1. Setup → Object Manager → Patient\_\_c → Lightning Record Pages → New.
2. Select Record Page → choose a Header and Right Sidebar template.
3. Add components: Highlights Panel, Related Lists (Appointments, Migrant Records), Custom Component Region (patientSearch LWC).
4. Save and activate for desktop and mobile.
5. 

### D. Add Quick Action for Patient Intake Flow

1. Object Manager → Patient\_\_c → Buttons, Links, and Actions → New Action.
2. Action Type: Flow → select Patient\_Intake\_Flow.
3. Label: New Patient Intake.
4. Add to the Patient page layout under Salesforce Mobile and Lightning Actions.
5. 
6. 

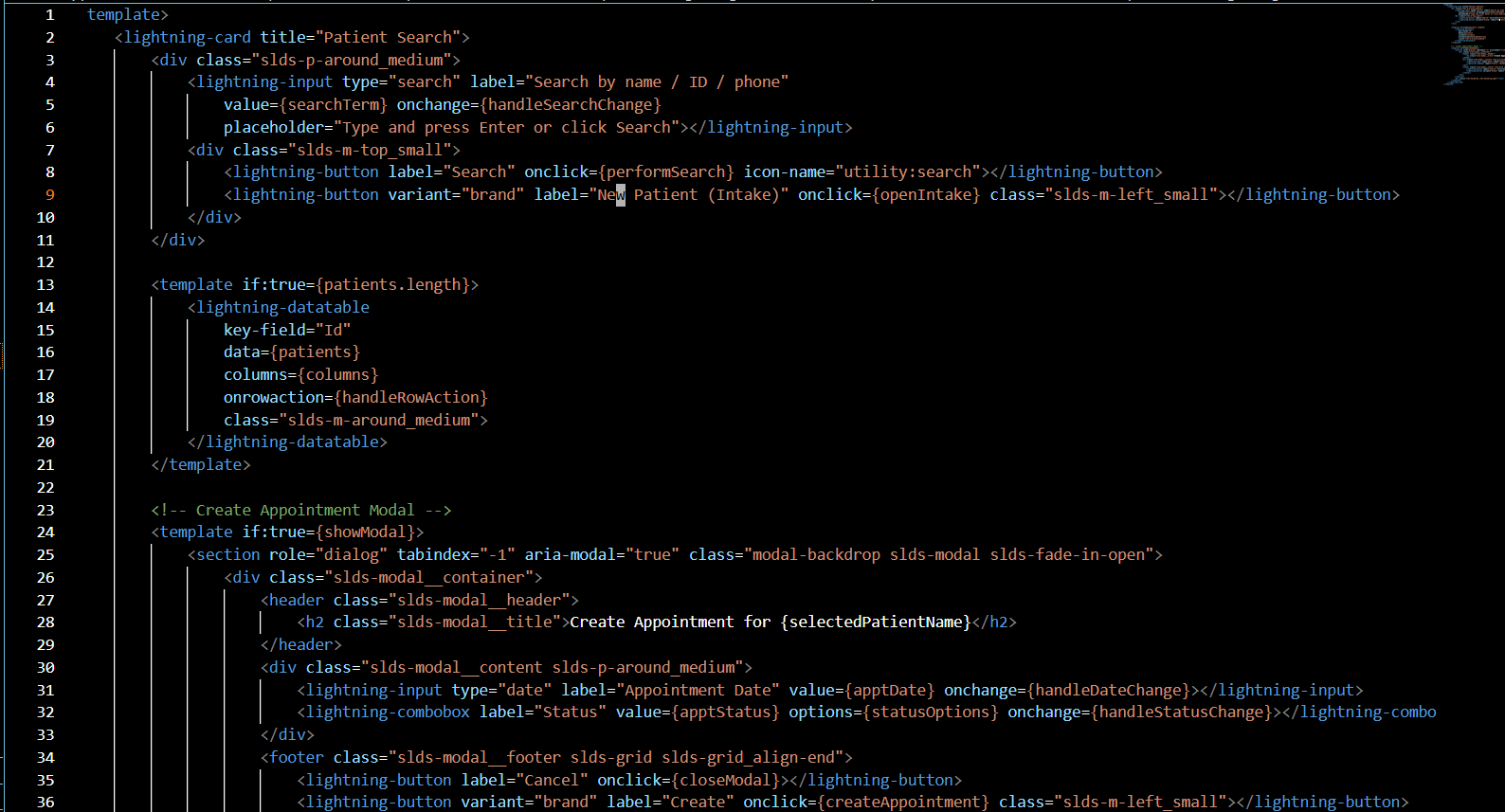
### E. Lightning Web Component: patientSearch

Below is the Apex Controller and sample LWC structure used to implement the patient search and appointment creation functionality.





public with sharing class PatientController {  
 @AuraEnabled(cacheable=true)  
 public static List<Patient\_\_c> searchPatients(String q) {  
 if (String.isBlank(q)) return new List<Patient\_\_c>();  
 String term = '%' + String.escapeSingleQuotes(q.trim()) + '%';  
 return [  
 SELECT Id, Name, Patient\_ID\_\_c, Phone\_\_c, Email\_\_c, DOB\_\_c  
 FROM Patient\_\_c  
 WHERE Name LIKE :term OR Patient\_ID\_\_c LIKE :term OR Phone\_\_c LIKE :term  
 LIMIT 50  
 ];  
 }  
  
 @AuraEnabled  
 public static Id createAppointment(Id patientId, String appointmentDateStr, String status) {  
 Date aptDate;  
 if (String.isBlank(appointmentDateStr)) {  
 aptDate = Date.today();  
 } else {  
 try {  
 aptDate = Date.valueOf(appointmentDateStr);  
 } catch (Exception ex) {  
 aptDate = Date.today();  
 }  
 }  
 Appointment\_\_c a = new Appointment\_\_c(  
 Patient\_\_c = patientId,  
 Appointment\_Date\_\_c = aptDate,  
 Status\_\_c = status == null ? 'Scheduled' : status  
 );  
 insert a;  
 return a.Id;  
 }  
}



<template>  
 <lightning-card title="Patient Search">  
 <div class="slds-p-around\_medium">  
 <lightning-input type="search" label="Search by name / ID / phone"  
 value={searchTerm} onchange={handleSearchChange}  
 placeholder="Type and press Enter or click Search"></lightning-input>  
 <div class="slds-m-top\_small">  
 <lightning-button label="Search" onclick={performSearch} icon-name="utility:search"></lightning-button>  
 <lightning-button variant="brand" label="New Patient (Intake)" onclick={openIntake} class="slds-m-left\_small"></lightning-button>  
 </div>  
 </div>  
  
 <template if:true={patients.length}>  
 <lightning-datatable  
 key-field="Id"  
 data={patients}  
 columns={columns}  
 onrowaction={handleRowAction}  
 class="slds-m-around\_medium">  
 </lightning-datatable>  
 </template>  
  
 <!-- Create Appointment Modal -->  
 <template if:true={showModal}>  
 <section role="dialog" tabindex="-1" aria-modal="true" class="modal-backdrop slds-modal slds-fade-in-open">  
 <div class="slds-modal\_\_container">  
 <header class="slds-modal\_\_header">  
 <h2 class="slds-modal\_\_title">Create Appointment for {selectedPatientName}</h2>  
 </header>  
 <div class="slds-modal\_\_content slds-p-around\_medium">  
 <!-- Appointment Form Elements Here -->  
 </div>  
 </div>  
 </section>  
 </template>  
 </lightning-card>  
</template>

