

PROJECT NAME : Salesforce Health Cloud (Admin)

Team ID : LTVIP2024TMID13993

Team Size : 1

Team Leader : Nalla Bhumika

Create Patient Object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
- 2.

The screenshot shows the Salesforce Object Manager interface. At the top, there are tabs for Setup, Home, and Object Manager. The Object Manager tab is selected. In the top right corner, there is a 'Create' button with a dropdown arrow, which is highlighted with a red box. Below the tabs, there is a search bar labeled 'Search Setup'. The main area displays a table with columns: LABEL, API NAME, TYPE, DESCRIPTION, and LAST MODIFIED. A single row is visible for the 'Account' object. On the far right of the table, there is a link 'Custom Object from Spreadsheet'.

- Enter the label name >> Patient
- Plural label name>>Patients

The screenshot shows the 'New Custom Object' creation screen. At the top, it says 'SETUP New Custom Object'. Below that is a section titled 'Custom Object Information' with fields for 'Label' (Patient) and 'Plural Label' (Patients), both of which are highlighted with red boxes. There is also a note about starting with a vowel sound and a 'Content Name' dropdown set to 'None'. Further down, there is a section titled 'Enter Record Name Label and Format' with a note about record names. It shows 'Record Name' (Patient Name) and 'Data Type' (Text), both of which are highlighted with red boxes. There is also a note about context-sensitive help settings.

Enter Record Name Label and Format

- Record Name >>Patient name
- Data Type >> Text

This screenshot is identical to the previous one, showing the 'New Custom Object' creation screen. The 'Record Name' field now contains 'Patient Name' and the 'Data Type' field now contains 'Text', both of which are highlighted with red boxes. The rest of the interface remains the same, including the notes about context-sensitive help and content names.

3. Click on Allow reports.
4. Allow search >> Save.

Create Doctor Object :

The purpose of creating a Doctor object is to store and manage information about Patients.

To create an object: From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

- Enter the label name >>Doctor
- Plural label name >> Doctor
- Enter Record Name Label and Format
- Record Name >> Doctor name
- Data Type >> Text
- Click on Allow reports.
- Allow search >> Save

The screenshot shows the 'New Custom Object' page in the Salesforce Setup interface. The 'Label' field is set to 'Doctor' and the 'Plural Label' field is set to 'Doctors'. The 'Record Name' field is also set to 'Doctor'. The 'Data Type' is selected as 'Text'. Other settings like 'Description', 'Context-Sensitive Help Setting', and 'Content Name' are also visible.

Creating 2 more objects with label name as Appointment, Bed Reservation.

(Use “Auto Number” as a data type for Appointment, BedReservation where Display Format :A-{00} and B-{00} ,Starting Number :1).

The screenshot shows the 'New Custom Object' page for creating an 'Appointment' object. The 'Label' field is set to 'Appointment' and the 'Plural Label' field is set to 'Appointments'. The 'Record Name' field is set to 'Appointment'. The 'Data Type' is selected as 'Auto Number' with a display format of 'A-{00}' and a starting number of '1'. Other settings like 'Description', 'Context-Sensitive Help Setting', and 'Content Name' are also visible.

New Custom Object

Label: Bed Reservation Example: Account
 Plural Label: Bed Reservations Example: Accounts
 Starts with vowel sound:

The Object Name is used when referencing the object via the API.
 Object Name: Bed_Reservation Example: Account

Description:

Context-Sensitive Help Setting:
 Open the standard Salesforce.com Help & Training window
 Open a window using a Visualforce page

Content Name:

Enter Record Name Label and Format
 The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name: Bed Reservation Name Example: Account Name

Data Type: Auto Number
 Display Format: B-(00) Example: A-(0000) [What Is This?](#)
 Starting Number: 1

Fields & Relationships :

Creating Text Area Field In Patient Object

To create fields in an object:

1. Go to setup > click on Object Manager > type object name(Patient) in quick find bar>> click on the object.

Setup Home **Object Manager**

SETUP Object Manager

patient

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Patient	Patient__c	Custom Object		30/08/2023	<input type="checkbox"/>

2. Now click on “Fields & Relationships” >> New

SETUP > OBJECT MANAGER
Patient

Details **Fields & Relationships**

Page Layouts
 Lightning Record Pages
 Buttons, Links, and Actions

Fields & Relationships
 12 Items, Sorted by Field Label

Blood Group	Blood_Group__c	Picklist
Created By	CreatedBy	Lookup(User)

Quick Find Deleted Fields Field Dependencies Set History Tracking

3. Select Data type as “Text Area”.

The screenshot shows the Salesforce Object Manager interface for the 'Patient' object. On the left, there's a sidebar with various setup options like Details, Fields & Relationships, Page Layouts, etc. Under 'Fields & Relationships', the 'Text Area' option is highlighted with a red box and a red arrow pointing to it. A detailed description of the 'Text Area' field type is visible on the right side of the screen, explaining its functionality and character limits.

4. Click on Next

This screenshot shows the 'Edit Patient Custom Field' page for the 'Address' field. The 'Field Label' is set to 'Address'. The 'Data Type' is explicitly set to 'Text Area'. Other configuration options like 'Field Name', 'Description', 'Help Text', 'Data Owner', 'Field Usage', 'Data Sensitivity Level', and 'Compliance Categorization' are also visible.

5. Fill the above as following:

- Field Label: Address
- Field Name : gets auto generated
- Click on Next >> Next >> Save and new.

Creating The Date Field In Object Patient

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Patient) in quick find bar>> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as “Date” and click Next.
4. Given the Field Label as “DOB”.

SETUP > OBJECT MANAGER
Patient

Patient New Custom Field

Step 2. Enter the details Step 2 of 4

Field Label **DOB**

Field Name **DOB**

Description

Help Text

Required Always require a value in this field in order to save a record

Auto add to custom report type Add this field to existing custom report types that contain this entity

Default Value Show Formula Editor

Use formula syntax: Enclose text and persist value (AFL) names in double quotes: "Text__c". Include numbers without quotes (e.g. above percentages as decimals (0.10), and express date calculations in the standard format: (Today)+7). To reference a field from a Custom Metadata type record use: \$CustomMetadata_Type__mdtRecords\$!FieldName__c

5. Field Name will be auto populated, and click on Next >> Next >> Save.

Creating Formula Field In Patient Object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Patient) in quick find bar>> click on the object.

1. Now click on “Fields & Relationships” >> New.
2. Select Data type as “Formula” and click Next.
3. Give Field Label and Field Name as “Age” and select formula return type as “Number” and change the decimal values to zero and click next.

SETUP > OBJECT MANAGER
Patient

Patient New Custom Field

Step 2. Choose output type Step 2 of 5

Field Label **Age**

Field Name **Age**

Auto add to custom report type Add this field to existing custom report types that contain this entity

Formula Return Type

None Selected Select one of the data types below.

Number

Checkbox

Currency

Date

Date/Time

Percent

Calculate a boolean value
Example: [TODAY() > CloseDate]

Calculate a dollar or other currency amount and automatically format the field as a currency amount.
Example: [Gross Margin = Amount - Cost__c]

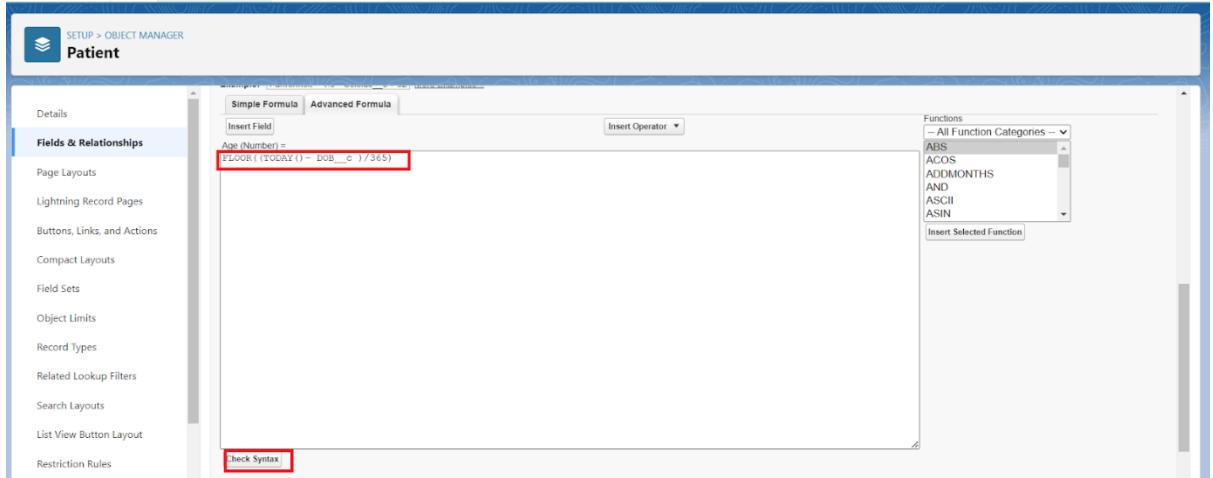
Calculate a date, for example, by adding or subtracting days to other dates.
Example: [Reminder Date = CloseDate + 7]

Calculate a datetime, for example, by adding a number of hours or days to another datetime.
Example: [Next = NOW() + 1]

Calculate a numeric value
Example: [Fahrenheit = 1.8 * Celsius__c + 32]

Calculate a percent and automatically add the percent sign to the number.
Example: [Discount = (Amount - Discounted_Amount__c) / Amount]

4. Under Advanced Formula write down the formula : $\text{FLOOR}((\text{TODAY}() - \text{DOB}_\text{c}) / 365)$.

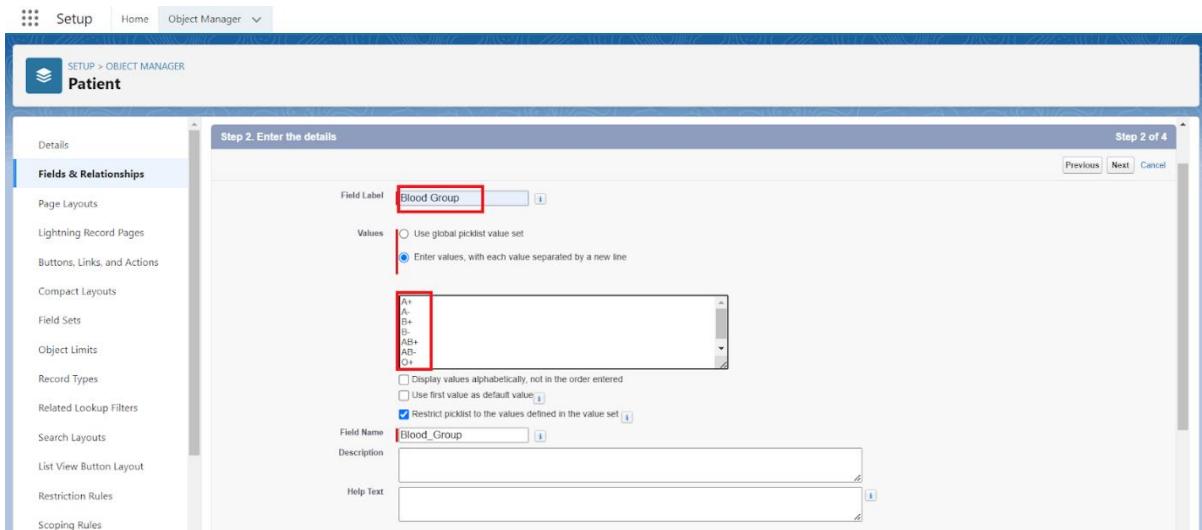


- click "Check Syntax" and Next >> Next>> Save & New.

Creating Picklist Field In Patient Object

To create fields in an object:

- Go to setup >> click on Object Manager >> type object name(patient) in quick find bar>> click on the object.
- Now click on “Fields & Relationships” >> New.
- Select Data type as “Picklist” and click Next.
- Enter Field Label as “Blood Group”.
- In values select “Enter values(A+,A-,B+,B-,AB+,AB-,O+,O-), with each value separated by a new line” and enter values as shown below.



- Click Next>> Next >> Next >> Save .

Creating The Email Field In Object Patient

To create fields in an object:

- Go to setup >> click on Object Manager >> type object name(Patient) in quick find bar >>click on the object.
- Now click on “Fields & Relationships” >> New

3. Select Data type as “Email” and click Next.
4. Given the Field Label as “Email”.

The screenshot shows the 'Patient' object in the Object Manager. On the left sidebar, 'Fields & Relationships' is selected. The main area is titled 'New Custom Field' and 'Step 2. Enter the details'. It shows two fields: 'Field Label' (Email) and 'Field Name' (Email), both of which are highlighted with red boxes. Below these are 'Description' and 'Help Text' fields, and checkboxes for 'Required' and 'Auto add to custom report type'. At the bottom right are 'Previous', 'Next', and 'Cancel' buttons, with 'Next' being highlighted.

5. Field Name will be auto populated, and click on Next? Next ? Save.

Creating The Phone Field In Object Patient

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Patient) in quick find bar>>click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as “Phone” and click Next.
4. Given the Field Label as “ Mobile”.

The screenshot shows the 'Edit Patient Custom Field' page for the 'Mobile' field. The 'Field Label' and 'Field Name' are both set to 'Mobile' and highlighted with red boxes. Other visible fields include 'Description', 'Help Text', 'Data Owner' (User), 'Field Usage' (None), 'Data Sensitivity Level' (None), and a 'Compliance Categorization' section with 'Available' (PII, HIPAA, GDPR) and 'Chosen' (None). At the top right are 'Change Field Type', 'Save', and 'Cancel' buttons, with 'Save' being highlighted.

Creating Lookup Relationship

To Create a relationship between Appointment & Patient Objects.

1. Go to the setup page >> click on object manager >> type object name(Appointment) in the quick find bar >>click on the object.
2. Click on fields & relationship >> click on New.
3. Select “Lookup relationship” as data type and click Next.
4. Select the related object “ Patient”.
5. Give Field Label as “Patient Name” and click Next.
6. Next >> Next >> Save.

Setup > Object Manager

Appointment

Field Information

Field Label	Patient Name	Data Type	Lookup
Field Name	Patient_Name		
Description			
Help Text			
Data Owner	User		
Field Usage	--None--		
Data Sensitivity Level	--None--		
Compliance Categorization	Available: PII, HIPAA, GDPR, PCI Chosen:		

Lookup Options

Related To	Patient	Child Relationship Name	Appointments
Related List Label	Appointments		
Required	<input type="checkbox"/> Always require a value in this field in order to save a record <input checked="" type="radio"/> Clear the value of this field. You can't choose this option if you make this field required.		

Creating Remaining Fields In Objects

Setup > Object Manager

Patient

Fields & Relationships

12 items. Sorted by Field Label		
Age	Age_c	Formula (Number)
Blood Group	Blood_Group_c	Picklist
Created By	CreatedBy	Lookup(User)
DOB	DOB_c	Date
Email	Email_c	Email
Gender	Gender_c	Picklist
Last Modified By	LastModifiedBy	Lookup(User)
Mobile	Mobile_c	Phone
Owner	OwnerId	Lookup(User,Group)
Patient ID	Patient_ID_c	Auto Number
Patient name	Name	Text(80)

Setup > Object Manager

Bed Reservation

Fields & Relationships

11 items. Sorted by Field Label		
Bed type	Bed_type_c	Picklist
BedReservation Name	Name	Auto Number
Booking Status	Booking_Status_c	Picklist
Created By	CreatedBy	Lookup(User)
End Time	End_Time_c	Time
Last Modified By	LastModifiedBy	Lookup(User)
Owner	OwnerId	Lookup(User,Group)
Patient	Patient_c	Lookup(Patient)
Reservation Date	Reservation_Date_c	Date
Start Time	Start_Time_c	Time

Fields & Relationships			
12 items, Sorted by Field Label			
Blood Group	Blood_Group__c	Picklist	▼
Created By	CreatedBy	Lookup(User)	▼
DOB	DOB__c	Date	▼
Doctor ID	Doctor_ID__c	Auto Number	▼
Doctor Name	Name	Text(60)	✓
Email	Email__c	Email	▼
Gender	Gender__c	Picklist	▼
Last Modified By	LastModifiedBy	Lookup(User)	▼
Mobile	Mobile__c	Phone	▼
Owner	OwnerId	Lookup(User,Group)	▼
Specialization	Specialization__c	Picklist (Multi-Select)	▼

Fields & Relationships				
9 items, Sorted by Field Label				
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment Date	Appointment_Date__c	Date	▼	▼
Appointment Name	Name	Auto Number	✓	▼
Created By	CreatedBy	Lookup(User)	▼	▼
Doctor Name	Doctor_Name__c	Lookup(Doctor)	✓	▼
End Time	End_Time__c	Time	▼	▼
Last Modified By	LastModifiedBy	Lookup(User)	▼	▼
Owner	OwnerId	Lookup(User,Group)	✓	▼
Patient Name	Patient_Name__c	Lookup(Patient)	✓	▼
Start Time	Start_Time__c	Time	▼	▼

Schema Builder

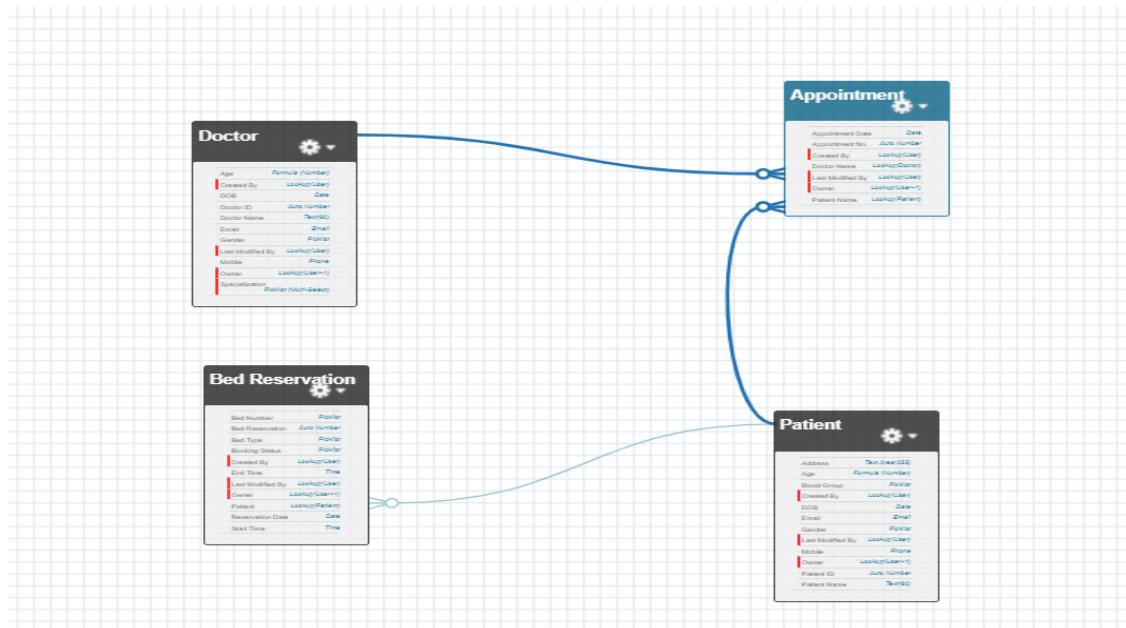
Schema Builder is a powerful tool within Salesforce that allows you to visualise, explore, and design the relationships between objects in your Salesforce organisation. It provides a graphical representation of the data model, making it easier to understand the structure and connections between different objects.

Creating Schema Builder

1. Go to setup >>click on Object Manager >> Schema Builder.

The screenshot shows the Salesforce Object Manager interface. At the top, there are navigation links for Setup, Home, and Object Manager. The Object Manager tab is selected. A red box highlights the 'Schema Builder' button in the top right corner of the page header. The main area displays a table of objects with columns for Label, API Name, Type, Description, Last Modified, and Deployed. The table includes standard objects like Account, Activity, and various appointment-related objects, along with custom objects like Asset and Asset Action.

2. Select objects >> Enter Objects as “Patient, Doctor, Appointment and Bed Reservation objects” in quick box and select them.



Creating The Field Dependencies

Use case:

Field Dependencies are used to create relationships between fields within an object. They allow you to control the visibility and availability of fields based on the values selected in other fields.

1. Go to setup > click on Object Manager > type object name(Bed Reservation) in quick find bar> click on the object.
2. Click on Fields & Relationships and click on the Bed Type field.
3. Search for Field Dependencies and click on New.

The screenshot shows the 'Custom Field Definition Detail' page for the 'Bed Reservation' object. The 'Fields & Relationships' tab is active. In the bottom right corner of the main content area, there is a red box around the 'Field Dependencies' tab.

4. Select Controlling Field as "Bed Type" and Depending field as "Bed Number">>> Continue.

The screenshot shows the 'New Field Dependency' wizard. Step 1: Select a controlling field and a dependent field. Click Continue when finished. Step 2: On the following page, edit the filter rules that control the values that appear in the dependent field for each value in the controlling field.

5. Select the "Bed Number" values of related "Bed Type" values like for regular select R1,R2 and Click on Include Values >> Save.

The screenshot shows the 'New Field Dependency' wizard. Step 2: On the following page, edit the filter rules that control the values that appear in the dependent field for each value in the controlling field. The grid shows the relationship between Bed Type and Bed Number. A legend indicates 'Included Value' (yellow) and 'Excluded Value' (green).

Creating The Validation Rule

Creating the validation rule for DOB field in Patient object

1. Go to setup >> click on Object Manager >> type object name(Patient) in quick find bar>> click on the object.
2. Click on the validation rule >> click New.

The screenshot shows the Salesforce Object Manager interface for the Patient object. The 'Validation Rules' section is displayed, showing a single entry named 'ValidationForDOB' which checks the DOB field. The 'Validation Rules' link in the sidebar is highlighted with a red box.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
ValidationForDOB	DOB	Check DOB	✓	Mishra Katoju, 25/08/2023, 11:14 am

3. Enter the Rule name as "ValidationForDOB".
4. Insert the Error Condition Formula as : DOB__C >= TODAY().

The screenshot shows the 'Validation Rule' creation screen for the Patient object. The 'Error Condition Formula' field contains the formula 'DOB__C >= TODAY()' and is highlighted with a red box. The 'Error Message' field contains 'Check DOB' and is also highlighted with a red box. The 'Error Location' dropdown is set to 'Field: DOB'.

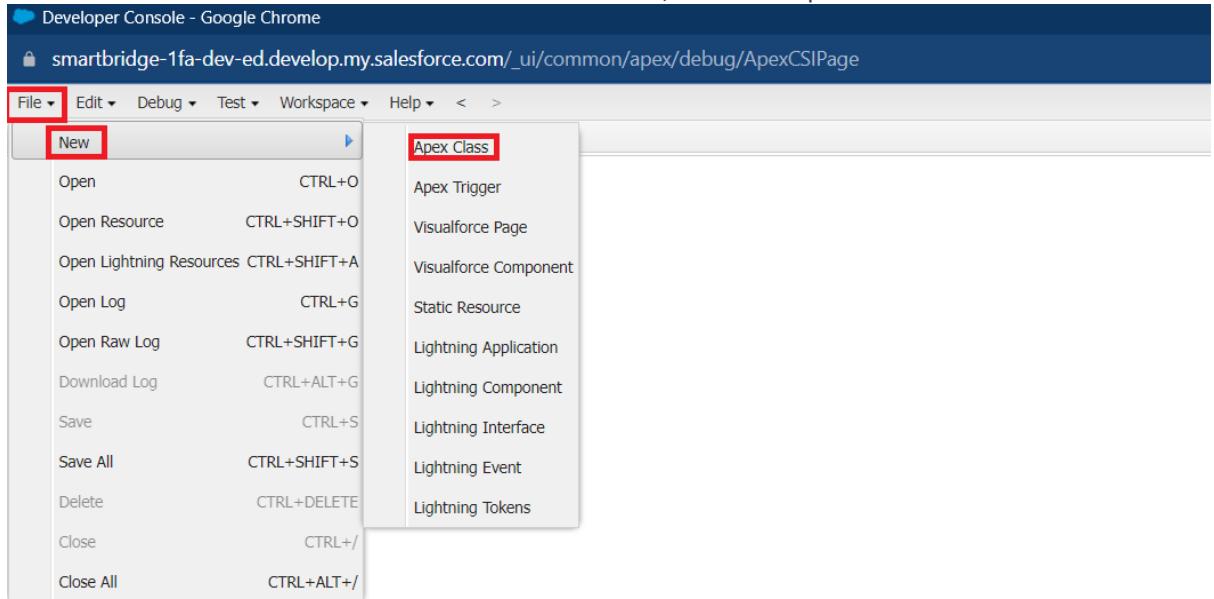
5. Enter the Error Message as "Check DOB", select the Error location as Field and select the field as "DOB", and click Save.

NOTE: Do same for DOB in Doctor Object.

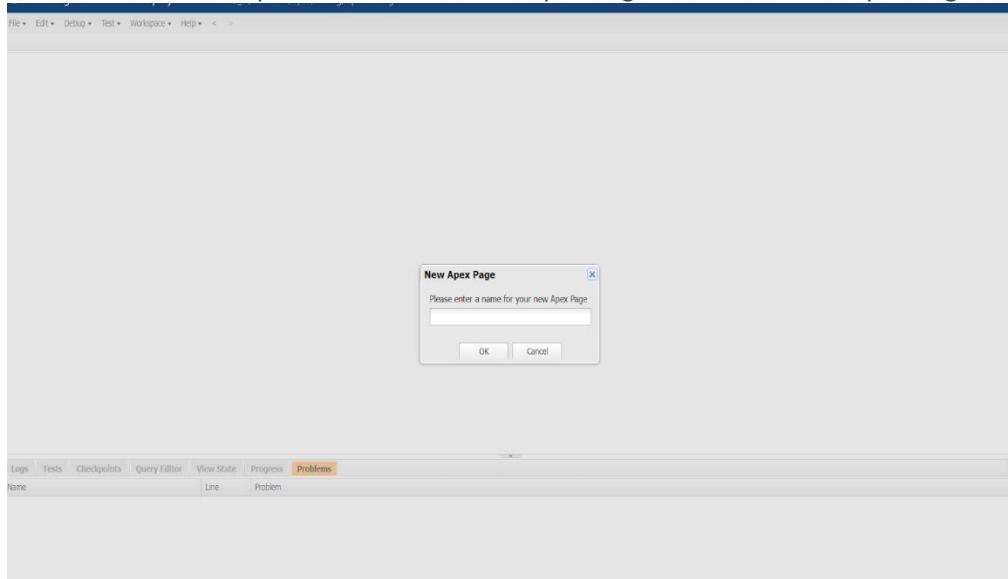
APEX & Visualforce Page:

Creating Apex Controller Class For Patient Object:

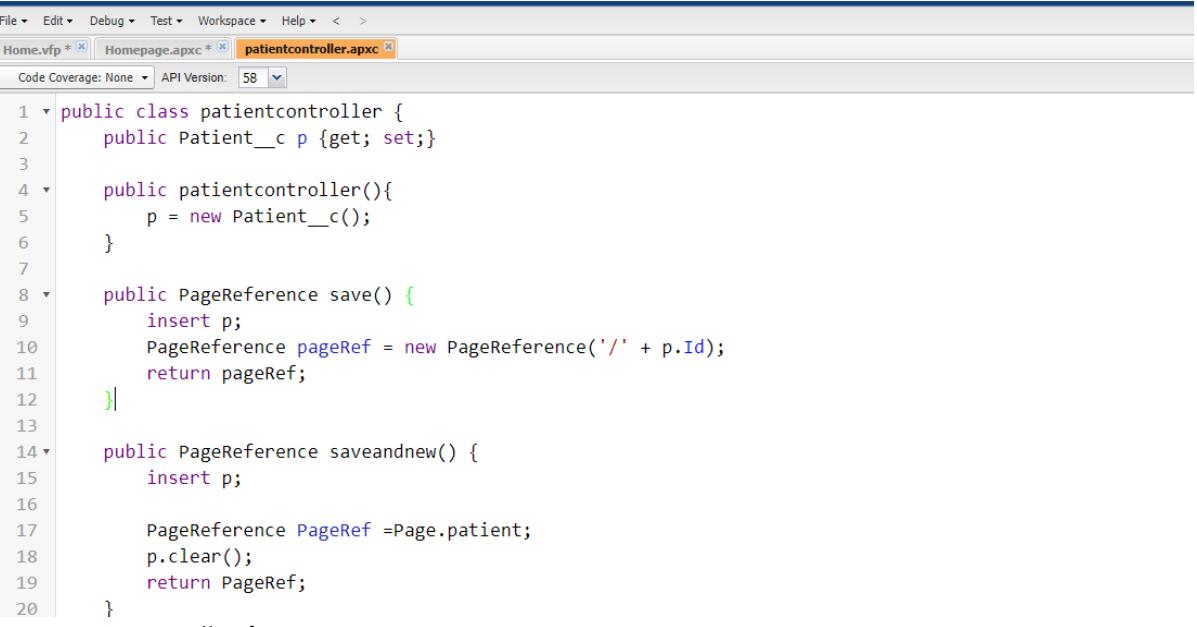
1. Go to setup >> click on gear symbol >> developer console >> there you can see many tools in the Toolbar of the new console window. Click on File, New and Apex Class.



2. Enter the name as " patientcontroller " for Apex Page to create a new Apex Page file.



3. Enter the code:



```
1 public class patientcontroller {
2     public Patient__c p {get; set;}
3
4     public patientcontroller(){
5         p = new Patient__c();
6     }
7
8     public PageReference save() {
9         insert p;
10        PageReference pageRef = new PageReference('/' + p.Id);
11        return pageRef;
12    }
13
14    public PageReference saveandnew() {
15        insert p;
16
17        PageReference PageRef =Page.patient;
18        p.clear();
19        return PageRef;
20    }
}
public class patientcontroller {
    public Patient__c p {get; set;}
}

public patientcontroller(){
    p = new Patient__c();
}

public PageReference save() {
    insert p;
    PageReference pageRef = new PageReference('/' + p.Id);
    return pageRef;
}

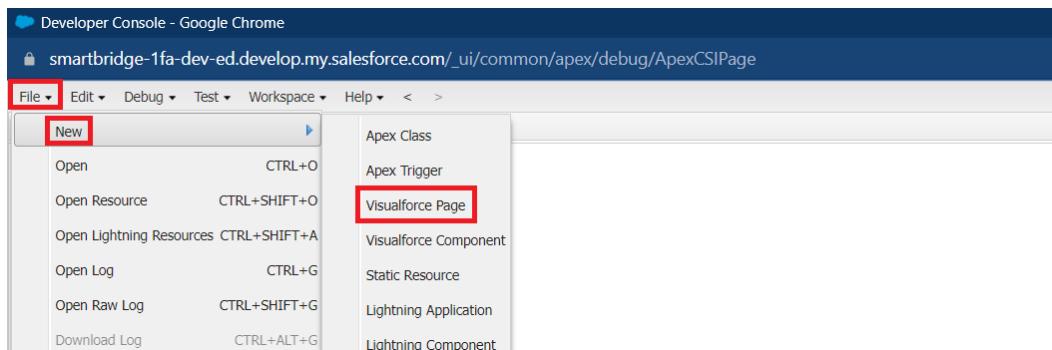
public PageReference saveandnew() {
    insert p;

    PageReference PageRef =Page.patient;
    p.clear();
    return PageRef;
}
}
```

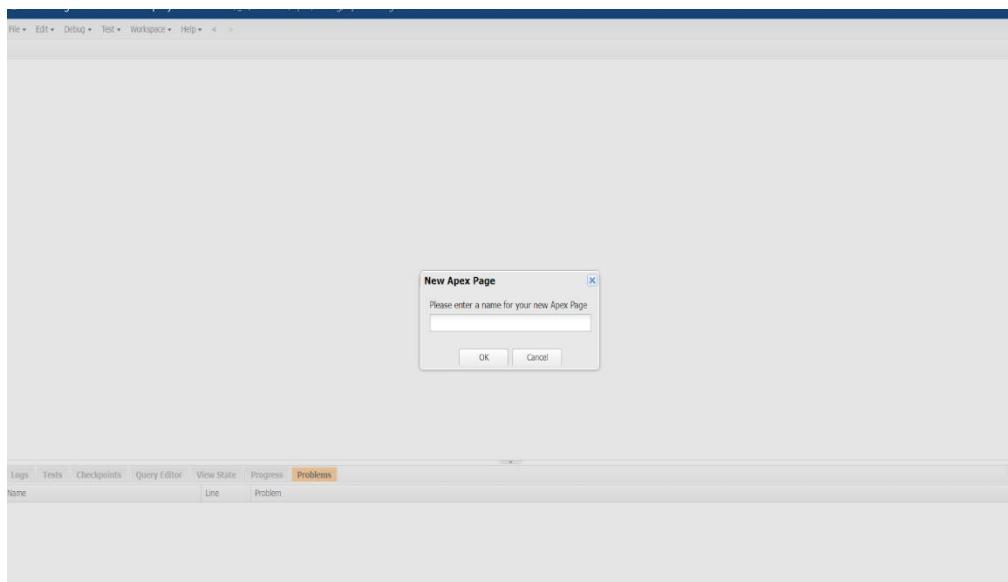
Creating Visualforce Page For Patient Object

It is as similar as java i.e, it also supports OOP(Object oriented programming) like Classes, objects, methods.

1. Go to setup >> click on gear symbol >> developer console >> there you can see many tools in the Toolbar of the new console window. Click on File, New and Visualforce page.



2. Enter the name as Patient for Apex Page to create a new Visualforce Page file.



3. Enter the code

Patient Visualforce Page:

```
<apex:page sidebar="false" showHeader="false" controller="patientcontroller">
<apex:form >
<apex:pageBlock title="NEW PATIENT DETAILS">

<apex:pageBlock >

<apex:pageBlockSection >
<apex:inputField value="{!!p.Patient_ID__c }"/>
<apex:inputField value="{!!p.Address__c }"/>
<apex:inputField value="{!!p.Name}" required="true"/>
<apex:inputField value="{!!p.DOB__c }"/>
<apex:inputField value="{!!p.Email__c }"/>
<apex:inputField value="{!!p.Age__c }"/>
<apex:inputField value="{!!p.Mobile__c }"/>
```

```

<apex:inputField value="{!!p.Gender__c}" />
<apex:inputField value="{!!p.Blood_Group__c}" />

</apex:pageBlockSection>

<apex:pageBlockButtons >
    <apex:commandButton value="Save" action="{!!save}" />
    <apex:commandButton value="Save & New" action="{!!saveandnew}" />
    <apex:commandButton value="Cancel" action="https://smartbridge-1fa-dev-ed.develop.lightning.force.com/lightning/n/Home_Page" immediate="true"/>
</apex:pageBlockButtons>
</apex:pageBlock>
</apex:form>

</apex:page>

```

Creating Apex Controller Class For Doctor Object:

Doctorcontroller Class Page:

```

public class Doctorcontroller {

    public Doctor__c d {get; set;}

    public Doctorcontroller(){
        d = new Doctor__c();
    }

    public PageReference save() {
        insert d;
        PageReference pageRef = new PageReference('/' + d.Id);
        return pageRef;
    }

    public PageReference saveandnew() {
        insert d;

        PageReference PageRef =Page.Doctor;
        d.clear();
        return PageRef;
    }
}

```

Creating Visualforce Page For Doctor Object

Doctor Visualforce Page:



The screenshot shows the Salesforce IDE interface with the 'Doctor.vfp' tab selected in the top navigation bar. The code editor displays the Visualforce page markup for creating a new doctor record.

```
1 <apex:page sidebar="false" showHeader="false" Controller="Doctorcontroller">
2 <apex:form>
3 <apex:pageBlock title="NEW DOCTOR DETAILS">
4
5     <apex:pageBlockSection>
6         <apex:inputField value="{!d.Doctor_ID_c }"/>
7         <apex:inputField value="{!d.Gender_c }"/>
8         <apex:inputField value="{!d.Name}" required="true"/>
9         <apex:inputField value="{!d.DOB_c }"/>
10        <apex:inputField value="{!d.Email_c }"/>
11        <apex:inputField value="{!d.Age_c }"/>
12        <apex:inputField value="{!d.Mobile_c }"/>
13        <apex:inputField value="{!d.Specialization_c }"/>
14    </apex:pageBlockSection>
15
16
17    <apex:pageBlockButtons>
18        <apex:commandButton value="Save" action=" {!save} " />
19        <apex:commandButton value="Save & New" action=" {!saveandnew} " />
20        <apex:commandButton value="Cancel" action="https://smartbridge-1fa-dev-ed.develop.lightning.force.com/lightning/n/Home_Page" immediate="true" />
21    </apex:pageBlockButtons>
22
23
24
25
26 </apex:pageBlock>
27 </apex:form>
28
29 </apex:page>
```

```
<apex:page sidebar="false" showHeader="false" Controller="Doctorcontroller">
<apex:form>
<apex:pageBlock title="NEW DOCTOR DETAILS">

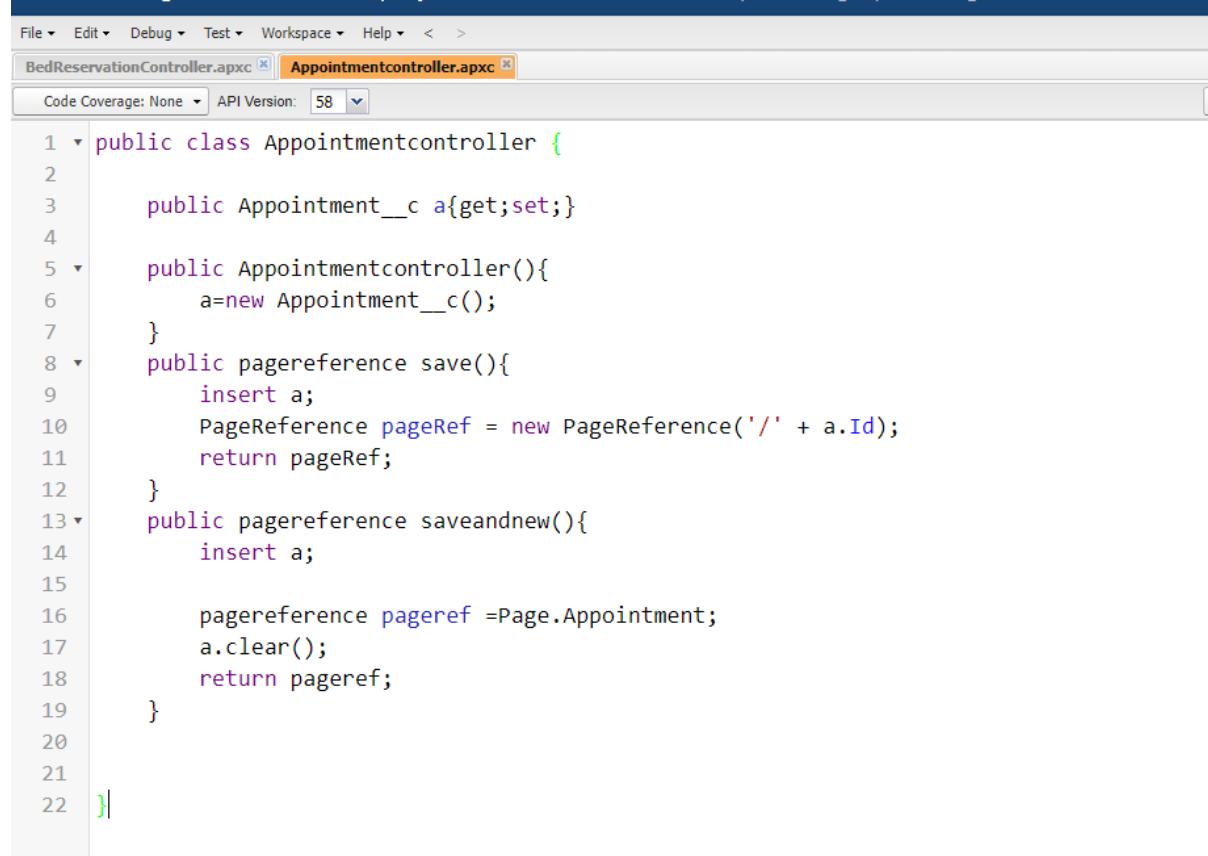
    <apex:pageBlockSection>
        <apex:inputField value="{!d.Doctor_ID_c }"/>
        <apex:inputField value="{!d.Gender_c }"/>
        <apex:inputField value="{!d.Name}" required="true"/>
        <apex:inputField value="{!d.DOB_c }"/>
        <apex:inputField value="{!d.Email_c }"/>
        <apex:inputField value="{!d.Age_c }"/>
        <apex:inputField value="{!d.Mobile_c }"/>
        <apex:inputField value="{!d.Specialization_c }"/>
    </apex:pageBlockSection>

    <apex:pageBlockButtons>
        <apex:commandButton value="Save" action=" {!save} " />
        <apex:commandButton value="Save & New" action=" {!saveandnew} " />
        <apex:commandButton value="Cancel" action="https://smartbridge-1fa-dev-ed.develop.lightning.force.com/lightning/n/Home_Page" immediate="true" />
    </apex:pageBlockButtons>

</apex:pageBlock>
</apex:form>
</apex:page>
```

Creating Apex Controller Class For Appointment Object:

Appointmentcontroller Class Page:



The screenshot shows the Salesforce IDE interface with the 'Appointmentcontroller.apxc' file open. The menu bar includes 'File', 'Edit', 'Debug', 'Test', 'Workspace', 'Help'. The top status bar shows 'BedReservationController.apxc' (closed), 'Appointmentcontroller.apxc' (active), 'Code Coverage: None', and 'API Version: 58'. The code editor displays the following Apex class:

```
1 public class Appointmentcontroller {
2
3     public Appointment__c a{get;set;}
4
5     public Appointmentcontroller(){
6         a=new Appointment__c();
7     }
8     public pagereference save(){
9         insert a;
10        PageReference pageRef = new PageReference('/' + a.Id);
11        return pageRef;
12    }
13    public pagereference saveandnew(){
14        insert a;
15
16        pagereference pageref =Page.Appointment;
17        a.clear();
18        return pageref;
19    }
20
21
22 }
```

```
public class Appointmentcontroller {

    public Appointment__c a{get;set;}

    public Appointmentcontroller(){
        a=new Appointment__c();
    }
    public pagereference save(){
        insert a;
        PageReference pageRef = new PageReference('/' + a.Id);
        return pageRef;
    }
    public pagereference saveandnew(){
        insert a;

        pagereference pageref =Page.Appointment;
        a.clear();
        return pageref;
    }

}
```

Creating Visualforce Page For Appointment Object :

Appointment Visualforce Page:



The screenshot shows the Visualforce page editor interface. At the top, there's a toolbar with 'Preview' and 'API Version: 15'. The main area contains the Visualforce page code.

```
1 <apex:page sidebar="false" showHeader="false" Controller="Appointmentcontroller">
2   <apex:form >
3
4     <apex:pageBlock title="Appointment DETAILS">
5
6       <apex:pageBlockSection >
7         <apex:inputField value="{!a.Name}" />
8         <apex:inputField value="{!a.Appointment_Date__c }"/>
9         <apex:inputField value="{!a.Doctor_Name__c }"/>
10        <apex:inputField value="{!a.Patient_Name__c }"/>
11        <apex:inputField value="{!a.Start_time__c }"/>
12        <apex:inputField value="{!a.End_Time__c }"/>
13      </apex:pageBlockSection>
14
15      <apex:pageBlockButtons >
16        <apex:commandButton value="Save" action=" {!save}" />
17        <apex:commandButton value="Save & New" action=" {!saveandnew}" />
18        <apex:commandButton value="Cancel" action="https://smartbridge-1fa-dev-ed.develop.lightning.force.com/lightning/n/Home_Page" />
19      </apex:pageBlockButtons>
20
21    </apex:pageBlock>
22  </apex:form>
23 </apex:page>
```

```
<apex:page sidebar="false" showHeader="false" Controller="Appointmentcontroller">
<apex:form >

  <apex:pageBlock title="Appointment DETAILS">

    <apex:pageBlockSection >
      <apex:inputField value="{!a.Name}" />
      <apex:inputField value="{!a.Appointment_Date__c }"/>
      <apex:inputField value="{!a.Doctor_Name__c }"/>
      <apex:inputField value="{!a.Patient_Name__c }"/>
      <apex:inputField value="{!a.Start_time__c }"/>
      <apex:inputField value="{!a.End_Time__c }"/>
    </apex:pageBlockSection>

    <apex:pageBlockButtons >
      <apex:commandButton value="Save" action=" {!save}" />
      <apex:commandButton value="Save & New" action=" {!saveandnew}" />
      <apex:commandButton value="Cancel" action="https://smartbridge-1fa-dev-
ed.develop.lightning.force.com/lightning/n/Home_Page" />
    </apex:pageBlockButtons>

  </apex:pageBlock>
</apex:form>
</apex:page>
```

Creating Apex Controller Class For Bed Reservation Object

BedReservationcontroller Class Page:

```
File Edit Insert Tools Window Help
BedReservationController.apx [ ] ApexControllerController.apex [ ]
Code-Change History API Version 19
1 public class BedReservationController {
2     public Bed_Reservation__c r { get; set; }
3
4     public BedReservationController() {
5         r = new Bed_Reservation__c();
6     }
7
8     public PageReference reservedBed(){
9         DateTime now = DateTime.now();
10
11        if (r.Reservation_Date__c < Date.today() || (r.Reservation_Date__c == Date.today() && r.Start_Time__c <= now.time())) {
12            ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'Booking time for the selected date has already passed.'));
13            return null;
14        }
15
16        List<Bed_Reservation__c> existingReservations = [
17            SELECT Id
18            FROM Bed_Reservation__c
19            WHERE Bed_Type__c = :r.Bed_Type__c
20            AND Reservation_Date__c >= :r.Reservation_Date__c
21            AND ((Start_Time__c >= :r.Start_Time__c AND Start_Time__c <= :r.End_Time__c)
22            OR (End_Time__c >= :r.Start_Time__c AND End_Time__c <= :r.End_Time__c))
23        ];
24
25        if (existingReservations.isEmpty()) {
26            // No overlapping reservations, proceed with booking
27            r.Booking_Status__c = 'Booked';
28            insert r;
29            PageReference pagerf = new PageReference('/' + r.Id);
30            return pagerf;
31        } else {
32            // Bed is already booked for the selected date and time range
33            ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'Bed is already booked for the selected date and time range.'));
34            return null;
35        }
36    }
37
38    public PageReference reserveAndBook(){
39        DateTime now = DateTime.now();
40
41        if (r.Reservation_Date__c < Date.today() || (r.Reservation_Date__c == Date.today() && r.Start_Time__c <= now.time())) {
42            ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'Booking time for the selected date has already passed.'));
43            return null;
44        }
45
46        List<Bed_Reservation__c> existingReservations = [
47            SELECT Id
48            FROM Bed_Reservation__c
49            WHERE Bed_Type__c = :r.Bed_Type__c
50            AND Reservation_Date__c >= :r.Reservation_Date__c
51            AND ((Start_Time__c >= :r.Start_Time__c AND Start_Time__c <= :r.End_Time__c)
52            OR (End_Time__c >= :r.Start_Time__c AND End_Time__c <= :r.End_Time__c))
53        ];
54
55        if (existingReservations.isEmpty()) {
56            // No overlapping reservations, proceed with booking
57            r.Booking_Status__c = 'Booked';
58            insert r;
59            PageReference pagerf = new PageReference('/' + r.Id);
60            r = new Bed_Reservation__c();
61            return pagerf;
62        } else {
63            // Bed is already booked for the selected date and time range
64            ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'Bed is already booked for the selected date and time range.'));
65            return null;
66        }
67    }
}
public class BedReservationController {
    public Bed_Reservation__c r { get; set; }

    public BedReservationController() {
        r = new Bed_Reservation__c();
    }

    public PageReference reserveBed(){
        DateTime now = DateTime.now();

        if (r.Reservation_Date__c < Date.today() || (r.Reservation_Date__c == Date.today() && r.Start_Time__c <= now.time())) {
            ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'Booking time for the selected date has already passed.'));
            return null;
        }

        List<Bed_Reservation__c> existingReservations = [
            SELECT Id
            FROM Bed_Reservation__c
            WHERE Bed_Type__c = :r.Bed_Type__c
            AND Reservation_Date__c = :r.Reservation_Date__c
            AND ((Start_Time__c >= :r.Start_Time__c AND Start_Time__c <= :r.End_Time__c)
            OR (End_Time__c >= :r.Start_Time__c AND End_Time__c <= :r.End_Time__c))
        ];
    }
}
```

```

if (existingReservations.isEmpty()) {
    // No overlapping reservations, proceed with booking
    r.Booking_Status__c = 'Booked';
    insert r;
    PageReference pageRef = new PageReference('/' + r.Id);
    return pageRef;
} else {
    // Bed is already booked for the selected date and time range
    ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'Bed is already
booked for the selected date and time range.'));
    return null;
}
}

public PageReference reserveandnew(){
    DateTime now = DateTime.now();

    if (r.Reservation_Date__c < Date.today() || (r.Reservation_Date__c == Date.today() &&
r.Start_Time__c <= now.time())) {
        ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'Booking time
for the selected date has already passed.'));
        return null;
    }

    List<Bed_Reservation__c> existingReservations = [
        SELECT Id
        FROM Bed_Reservation__c
        WHERE Bed_Type__c = :r.Bed_Type__c
        AND Reservation_Date__c = :r.Reservation_Date__c
        AND ((Start_Time__c >= :r.Start_Time__c AND Start_Time__c <= :r.End_Time__c)
        OR (End_Time__c >= :r.Start_Time__c AND End_Time__c <= :r.End_Time__c))
    ];

    if (existingReservations.isEmpty()) {
        // No overlapping reservations, proceed with booking
        r.Booking_Status__c = 'Booked';
        insert r;
        PageReference pageref = Page.BedReservation;
        r = new Bed_Reservation__c();
        return pageref;
    } else {
        // Bed is already booked for the selected date and time range
        ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'Bed is already
booked for the selected date and time range.'));
        return null;
    }
}
}

```

Creating Visualforce Page For Bed Reservation Object

BedReservation Visualforce Page:



The screenshot shows the Visualforce page editor interface. At the top, there are tabs for 'Preview' and 'API Version' set to '58'. Below the tabs, the page code is displayed in a monospaced font. The code defines a page with a controller of 'BedReservationController'. It contains a form with a page block section titled 'Bed Reservation' containing five input fields. It also includes a page block buttons section with three command buttons: 'Reserve Bed', 'reserveandnew', and 'Cancel'. The 'Cancel' button has an action pointing to a lightning page.

```
1 * <apex:page controller="BedReservationController">
2 *   <apex:form >
3 *     <apex:pageBlock >
4 *       <apex:pageBlockSection title="Bed Reservation">
5           <apex:inputField value="{!r.Bed_Type__c}" required="true"/>
6           <apex:inputField value="{!r.Bed_Number__c}" required="true"/>
7           <apex:inputField value="{!r.Reservation_Date__c}" required="true"/>
8           <apex:inputField value="{!r.Start_Time__c}" required="true"/>
9           <apex:inputField value="{!r.End_Time__c}" required="true"/>
10          <apex:inputField value="{!r.Patient__c}" required="true"/>
11      </apex:pageBlockSection>
12      <apex:pageBlockButtons >
13          <apex:commandButton action="{!reserveBed}" value="Reserve Bed"/>
14          <apex:commandButton value="reserveandnew" action="{!reserveandnew}" />
15          <apex:commandButton value="Cancel" action="https://smartbridge-1fa-dev-ed.develop.lightning.force.com/lightning/n/Home_Page" immediate="true" />
16      </apex:pageBlockButtons>
17  </apex:pageBlock>
18  <apex:pageMessages />
19 </apex:form>
20 </apex:page>
```

```
<apex:page controller="BedReservationController">
<apex:form >
<apex:pageBlock >
<apex:pageBlockSection title="Bed Reservation">
<apex:inputField value="{!r.Bed_Type__c}" required="true"/>
<apex:inputField value="{!r.Bed_Number__c}" required="true"/>
<apex:inputField value="{!r.Reservation_Date__c}" required="true"/>
<apex:inputField value="{!r.Start_Time__c}" required="true"/>
<apex:inputField value="{!r.End_Time__c}" required="true"/>
<apex:inputField value="{!r.Patient__c}" required="true"/>
</apex:pageBlockSection>
<apex:pageBlockButtons >
<apex:commandButton action="{!reserveBed}" value="Reserve Bed"/>
<apex:commandButton value="reserveandnew" action="{!reserveandnew}" />
<apex:commandButton value="Cancel" action="https://smartbridge-1fa-dev-
ed.develop.lightning.force.com/lightning/n/Home_Page" immediate="true" />
</apex:pageBlockButtons>
</apex:pageBlock>
<apex:pageMessages />
</apex:form>
</apex:page>
```

Creating Apex Controller Class For Home Page Tab

HomePage Class Page:

```
public class Homepage {  
  
    public List<Patient__c> patlists { get; set; }  
    public List<Doctor__c> doclists { get; set; }  
    public List<Appointment__c> applists { get; set; }  
    public List<Bed_Reservation__c> BedReserlists { get; set; }  
    public List<Bed_Reservation__c> matchingReservationslists { get; set; }  
    public String searchCriteria { get; set; }  
    public String selectedRecordId { get; set; }  
  
    public Homepage() {  
        patlists = [Select ID, Patient_ID__c, Name, Mobile__c, (Select ID, Name, Appointment_Date__c  
From Appointments__r) From Patient__c ORDER BY CreatedDate DESC LIMIT 10];  
        doclists = [select ID, Doctor_ID__c, Name, Mobile__c, (Select ID, Name, Appointment_Date__c  
From Appointments__r) from Doctor__c ORDER BY CreatedDate DESC LIMIT 10];  
        applists = [select ID, Name, Appointment_Date__c, Patient_Name__c, Doctor_Name__c,  
Doctor_Name__r.Name, Patient_Name__r.Name from Appointment__c ORDER BY CreatedDate  
DESC LIMIT 10];  
        BedReserlists = [select ID, Patient__c, Patient__r.Name, Bed_Number__c, Reservation_Date__c,  
Start_Time__c, End_Time__c from Bed_Reservation__c ORDER BY CreatedDate DESC LIMIT 10];  
    }  
  
    public PageReference performSearch() {  
        patlists = [SELECT ID, Patient_ID__c, Name, Mobile__c, (SELECT ID, Name,  
Appointment_Date__c FROM Appointments__r) FROM Patient__c WHERE Name LIKE :searchCriteria  
OR Mobile__c LIKE :searchCriteria OR Name LIKE :searchCriteria];  
        doclists = [SELECT ID, Doctor_ID__c, Name, Mobile__c, (SELECT ID, Name FROM  
Appointments__r) FROM Doctor__c WHERE Name LIKE :searchCriteria OR Mobile__c LIKE  
:searchCriteria];  
        Date searchDate = Date.valueOf(searchCriteria);  
        applists = [SELECT ID, Name, Appointment_Date__c, Patient_Name__c, Doctor_Name__c,  
Doctor_Name__r.Name, Patient_Name__r.Name FROM Appointment__c WHERE Name LIKE  
:searchCriteria OR Appointment_Date__c = :searchDate];  
  
        return null;  
    }  
  
    public PageReference deletePatient() {  
        try {  
            Patient__c patientToDelete = [SELECT Id FROM Patient__c WHERE Id = :selectedRecordId];  
            delete patientToDelete;  
            // Refresh the patient list after deletion  
            patlists = [Select ID, Patient_ID__c, Name, Mobile__c, (Select ID, Name,  
Appointment_Date__c From Appointments__r) From Patient__c];  
        } catch(Exception e) {  
        }  
    }  
}
```

```

        ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'An error
occurred while deleting the patient.'));
    }
    return null;
}

public PageReference deleteDoctor() {
try {
    Doctor__c doctorToDelete = [SELECT Id FROM Doctor__c WHERE Id = :selectedRecordId];
    delete doctorToDelete;
    // Refresh the doctor list after deletion
    doclists = [SELECT ID, Doctor_ID__c, Name, Mobile__c, (SELECT ID, Name,
Appointment_Date__c FROM Appointments__r) FROM Doctor__c];
} catch(Exception e) {
    ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'An error
occurred while deleting the doctor.'));
}
return null;
}
public PageReference deleteAppointment() {
try {
    Appointment__c appointmentToDelete = [SELECT Id FROM Appointment__c WHERE Id =
:selectedRecordId];
    delete appointmentToDelete;
    // Refresh the appointment list after deletion
    applists = [SELECT ID, Name, Appointment_Date__c, Patient_Name__c, Doctor_Name__c,
Doctor_Name__r.Name, Patient_Name__r.Name FROM Appointment__c];
} catch(Exception e) {
    ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'An error
occurred while deleting the appointment.'));
}
return null;
}
public PageReference deleteBedReservation() {
try {
    Bed_Reservation__c bedReservationToDelete = [SELECT Id FROM Bed_Reservation__c WHERE
Id = :selectedRecordId];
    delete bedReservationToDelete;
    // Refresh the bed reservation list after deletion
    BedReserlists = [SELECT ID, Patient__c, Patient__r.Name, Bed_Number__c,
Reservation_Date__c, Start_Time__c, End_Time__c FROM Bed_Reservation__c];
} catch(Exception e) {
    ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'An error
occurred while deleting the bed reservation.'));
}
return null;
}
}

```

Creating Visualforce Page For Home Tab

Home Visualforce Page:

- Visualforce name should be “Home”.

```
<apex:page showHeader="false" sidebar="true" Controller="Homepage" docType="html-5.0" >
<style type="text/css">
body {
background-image: url('https://www.prowesssoft.com/wp-content/uploads/2021/03/SalesforceHealthCareImage-B.jpg');
background-size: cover;
background-repeat: no-repeat;
background-attachment: fixed;
}
p {
font-weight: bold;
}
</style>

<script>
function clearPlaceholder() {
var inputField = document.getElementById('searchInput');
inputField.value = '';
inputField.style.color = '#000';
inputField.removeEventListener('click', clearPlaceholder);
}
</script>

<apex:form >
<apex:pageBlock title="NEW ENTRIES">
<center>

    <apex:inputText id="searchInput" style="color: #888;" onfocus="clearPlaceholder();"
value="{!searchCriteria}" />
        <apex:commandButton value="Search" action="{!performSearch}" rerender="resultsPanel"
/>
        <apex:commandButton value="New Patient" action="https://smartbridge-1fa-dev-
ed.develop.lightning.force.com/lightning/n/Patient"/>
        <apex:commandButton value="New Doctor" action="https://smartbridge-1fa-dev-
ed.develop.lightning.force.com/lightning/n/Doctor"/>
        <apex:commandButton value="New Appointment" action="https://smartbridge-1fa-dev-
ed.develop.lightning.force.com/lightning/n/Appointment"/>
        <apex:commandButton value="New BedReservation" action="https://smartbridge-1fa-dev-
ed.develop.lightning.force.com/lightning/n/BedReservation"/>
    </center>
</apex:pageBlock>
<apex:outputPanel id="resultsPanel">
<apex:pageBlock title="PATIENT DETAILS">
<apex:pageBlockTable value="{!patlists}" var="p">
```

```

<apex:column style="align:right" headerValue="Patient Names">
    <apex:outputLink value="/{!p.id}">{!p.Name}</apex:outputLink>
</apex:column>

<apex:column style="align:left" headerValue="mobile">
    <apex:outputLink value="/{!p.id}">{!p.Mobile__c}</apex:outputLink>
</apex:column>

<apex:column style="align:left" headerValue="Appointments">
    <apex:repeat value="{!p.Appointments__r}" var="b">
        <apex:outputLink value="/{!b.id}">{!b.Name}</apex:outputLink><br/>
    </apex:repeat>
</apex:column>

<apex:column style="align:left" headerValue="Appointment Date">
    <apex:repeat value="{!p.Appointments__r}" var="a"><br/>
        {!a.Appointment_Date__c}
    </apex:repeat>
</apex:column>
<apex:column style="align:center">
    <apex:commandButton value="Delete" action="{!!deletePatient}"
rerender="resultsPanel">
        <apex:param name="selectedRecordId" value="{!!p.Id}"
assignTo="{!!selectedRecordId}"/>
    </apex:commandButton>
</apex:column>
</apex:pageBlockTable>
</apex:pageBlock>

<apex:pageBlock title="DOCTOR DETAILS" >

<apex:pageBlockTable value="{!!doclists}" var="d">

    <apex:column style="align:right" headerValue="Doctor Names">
        <apex:outputLink value="/{!d.id}">{!d.Name}</apex:outputLink>
    </apex:column>

    <apex:column style="align:left" headerValue="mobile">
        <apex:outputLink value="/{!d.id}">{!d.Mobile__c}</apex:outputLink>
    </apex:column>

    <apex:column style="align:left" headerValue="Appointments">
        <apex:repeat value="{!!d.Appointments__r}" var="a">
            <apex:outputLink value="/{!a.id}">{!a.Name}</apex:outputLink>
        </apex:repeat>
    </apex:column>

    <apex:column style="align:left" headerValue="Appointment Date">

```

```

<apex:repeat value="{!d.Appointments__r}" var="b">
    {!b.Appointment_Date__c}
</apex:repeat>
</apex:column>
<apex:column style="align:center">
    <apex:commandButton value="Delete" action="{!!deleteDoctor}"
rerender="resultsPanel">
        <apex:param name="selectedRecordId" value="{!!d.Id}"
assignTo="{!!selectedRecordId}" />
    </apex:commandButton>
</apex:column>
</apex:pageBlockTable>
</apex:pageBlock>

<apex:pageBlock title="APPOINTMENT DETAILS" >

    <apex:pageBlockTable value="{!!applists}" var="a">

        <apex:column style="align:right" headerValue="Appointment No">
            <apex:outputLink value="/{!a.id}">{!a.Name}</apex:outputLink>
        </apex:column>

        <apex:column style="align:left" headerValue="Patient Name">
            <apex:outputLink value="/{!a.id}">{!a.Patient_Name__r.Name}</apex:outputLink>
        </apex:column>

        <apex:column style="align:left" headerValue="Appoinment Date">
            <apex:outputLink value="/{!a.id}">{!a.Appointment_Date__c}</apex:outputLink>
        </apex:column>

        <apex:column style="align:left" headerValue="Doctor Name">
            <apex:outputLink value="/{!a.id}">{!a.Doctor_Name__r.Name}</apex:outputLink>
        </apex:column>
        <apex:column style="align:center">
            <apex:commandButton value="Delete" action="{!!deleteAppointment}"
rerender="resultsPanel">
                <apex:param name="selectedRecordId" value="{!!a.Id}"
assignTo="{!!selectedRecordId}" />
            </apex:commandButton>
        </apex:column>
        </apex:pageBlockTable>
    </apex:pageBlock>
    <apex:pageBlock title="BedReservation DETAILS" >
        <apex:page showHeader="false" sidebar="true" Controller="Homepage">
            </apex:page>
        <apex:pageBlockTable value="{!!BedReserlists}" var="b">
            <apex:column style="align:right" headerValue="Patient Name">

```

```

        <apex:outputLink value="/{!b.id}">{!b.Patient__r.Name}</apex:outputLink>
    </apex:column>

    <apex:column style="align:right" headerValue="Bed Number">
        <apex:outputLink value="/{!b.id}">{!b.Bed_Number__c}</apex:outputLink>
    </apex:column>

    <apex:column style="align:left" headerValue="Reservation Date">
        <apex:outputLink value="/{!b.id}">{!b.Reservation_Date__c}</apex:outputLink>
    </apex:column>

    <apex:column style="align:left" headerValue="Start Time">
        <apex:outputLink value="/{!b.id}">{!b.Start_Time__c}</apex:outputLink>
    </apex:column>

    <apex:column style="align:left" headerValue="End Time">
        <apex:outputLink value="/{!b.id}">{!b.End_Time__c}</apex:outputLink>
    </apex:column>
    <apex:column style="align:center">
        <apex:commandButton value="Delete" action="{!!deleteBedReservation}">
            <apex:param name="selectedRecordId" value="{!!B.Id}">
            assignTo="{!!selectedRecordId}" />
        </apex:commandButton>
    </apex:column>
</apex:pageBlockTable>
</apex:pageBlock>

</apex:outputPanel>
</apex:form>

```

</apex:page>

Creating A Visualforce Tab

To create a Tab:(Patient)

1. Go to setup page >> type Tabs in Quick Find bar >> click on tabs >> click New (Visualforce tabs)

The screenshot shows the Salesforce Setup interface with the 'Tabs' tab selected. The 'Custom Tabs' section is displayed, which includes sections for 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. The 'Visualforce Tabs' section is highlighted with a red box around its title and the 'New' button.

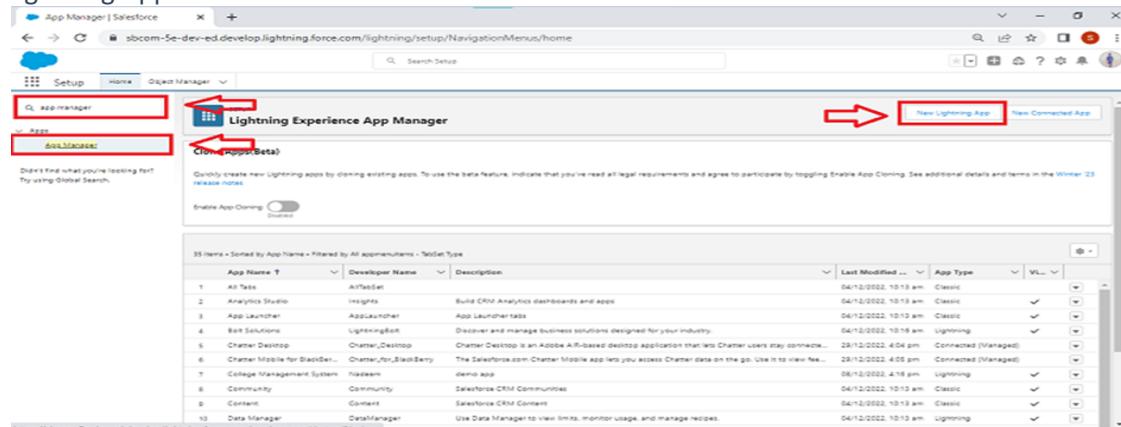
2. Select Visualforce Page as Patient>> Tab Label as Patient >>Tab Name (Auto Fill)>>Select any tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) keep it as default >> Save.

The screenshot shows the 'New Visualforce Tab' configuration page. It's Step 1 of 3. The 'Visualforce Page' dropdown is open, showing options like 'None', 'Appointment [Appointment]', 'BedReservation [BedReservation]', 'Doctor [Doctor]', 'Home [Home]', and 'Patient [Patient]'. The 'Patient [Patient]' option is highlighted with a red box. The 'Tab Label' dropdown is also open, showing 'None', 'Appointment [Appointment]', 'BedReservation [BedReservation]', 'Doctor [Doctor]', 'Home [Home]', and 'Patient [Patient]'. The 'Tab Name' field contains 'Patient [Patient]'. The 'Tab Style' dropdown is open, showing 'None', 'Standard', 'Lightning', and 'Custom'. The 'Splash Page Custom Link' dropdown is set to 'None'. The 'Description' field is empty. At the bottom right, there are 'Next' and 'Cancel' buttons, with the 'Next' button highlighted with a red box.

Create A Lightning App

To create a lightning app page:

Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.



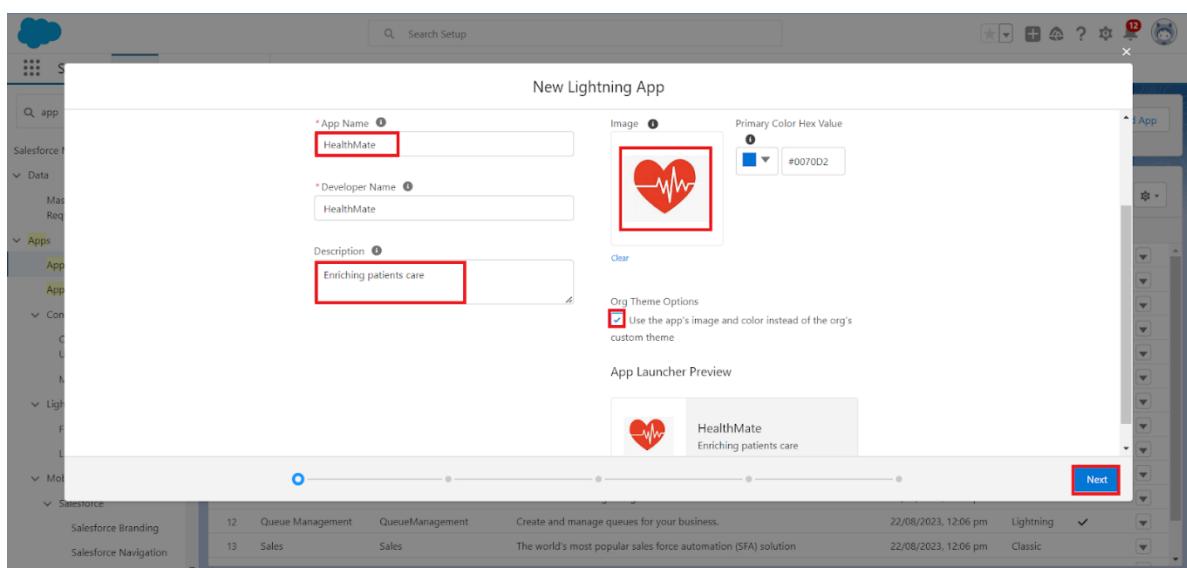
1. Fill the app name in app details and branding as follow

App Name : Healthmate

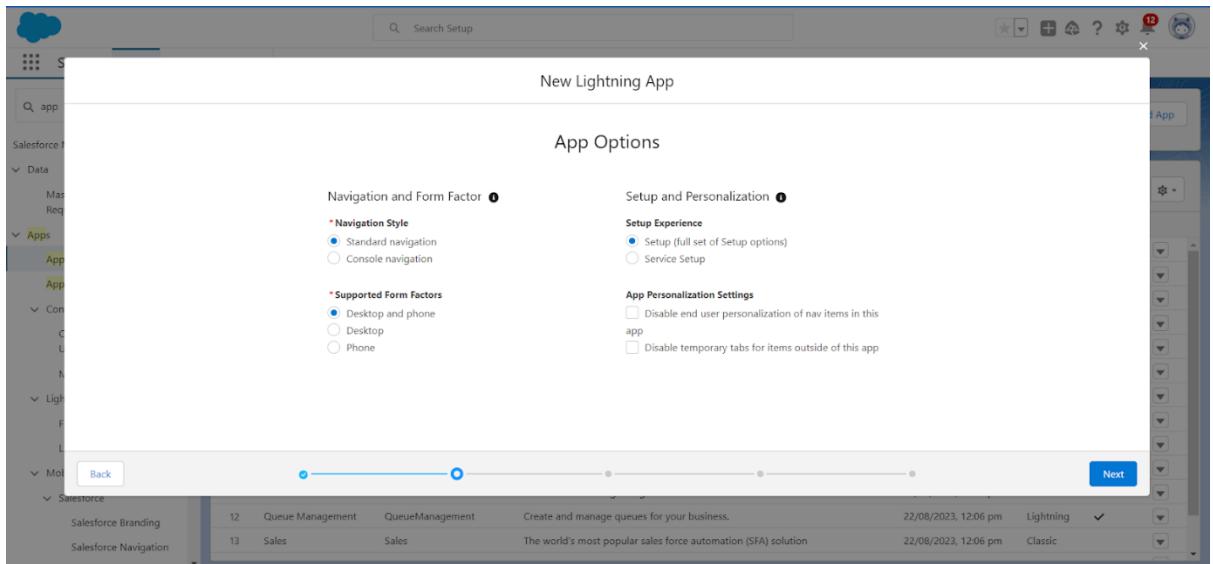
Developer Name : This will auto populated

Description :Enriching patients care

Image : optional (if you want to give any image you can otherwise not mandatory) Primary colour hex value : keep this default.



3. Then click Next >> (App option page)Set Navigation Style as Standard Navigation >> Next.



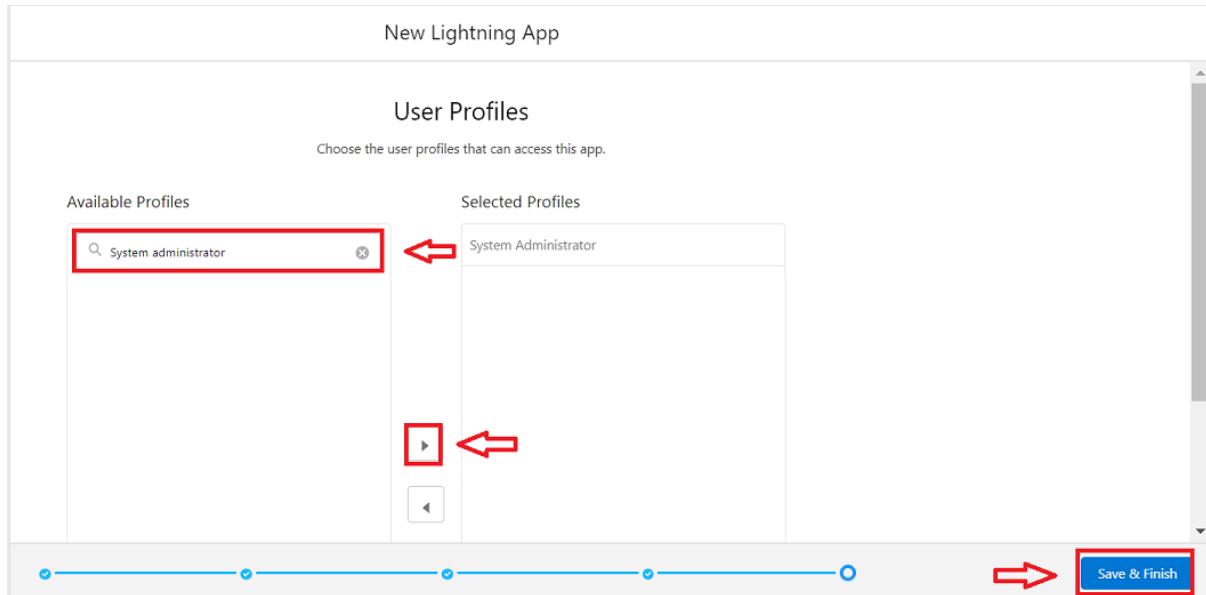
3. (Utility Items) keep it as default >> Next.

4. To Add Navigation Items:

Available Items	Selected Items
Accounts	Home Page
Alert Settings	Patient
All Sites	Doctor
Alternative Payment Methods	Appointment
Analytics	BedReservation
App Launcher	Reports
Appointment Invitations	Dashboards
Approval Requests	
Asset Action Sources	

Search for the item in the (Patient, Doctor, BedReservation, Appointment, Reports, Dashboard) from the search bar and move it using the arrow button >> Next >> Next.

5. To Add User Profiles:

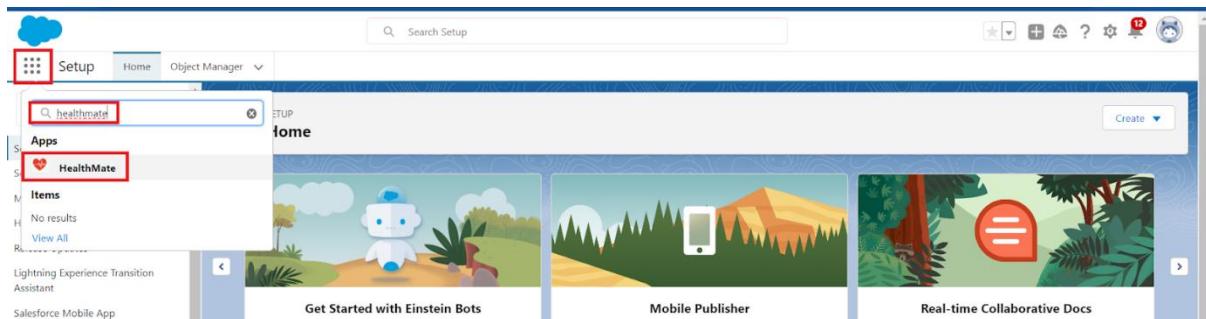


Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.

Change Actions

Cancel Actions:

1. Go to the setup page >> click “app Launcher” >>search for HealthMate.



2. This will be a user Interface which we get:

The screenshot shows the HealthMate application's homepage. At the top, there is a navigation bar with tabs: Home Page, Patient, Doctor, Appointment, BedReservation, Reports, and Dashboards. Below the navigation bar, there are several input forms:

- NEW ENTRIES:** A search bar with options for New Patient, New Doctor, New Appointment, and New BedReservation.
- PATIENT DETAILS:** Fields for Patient Names, mobile, Appointments, and Appointment Date.
- DOCTOR DETAILS:** Fields for Doctor Names, mobile, Appointments, and Appointment Date.
- APPOINTMENT DETAILS:** Fields for Appointment No, Patient Name, Appointment Date, and Doctor Name.
- BedReservation DETAILS:** Fields for Patient Name, Bed Number, Reservation Date, Start Time, and End Time.

- Click on the HomePage tab on Navigation Bar >> copy your browser's link specific to the HomePage.

The screenshot shows a browser window with the URL https://smartbridge-1fa-dev-ed.develop.lightning.force.com/lightning/n/Home_Page. The 'Home Page' tab in the navigation bar is highlighted with a red box. The page content is identical to the one shown in the first screenshot.

- Go to setup >> click on gear symbol?developer console >> there you can see many tools in the Toolbar of the new console window >> Click on File>> Open >> Select Entity Type as Pages >> double click on Patient in Entities .

The screenshot shows the 'Open' dialog in the Salesforce developer console. The left sidebar lists categories: Entity Type, Classes, Triggers, Pages (highlighted with a red box), Page Components, Objects, Static Resources, and Packages. The main area shows a table with columns: Entity Type, Entities, and Related. The 'Entities' column lists: Name, Namespace ▲, BedReservation, Doctor, Home, Appointment, and Patient (highlighted with a red box). The 'Related' column lists: Name, Extent, Direction, patientc..., ApexCla..., Referen..., Address, Custom..., Referen..., DOB, Custom..., Referen..., Age, Custom..., Referen..., Email, Custom..., Referen..., Gender, Custom..., Referen..., and Mobile, Custom..., Referen... .

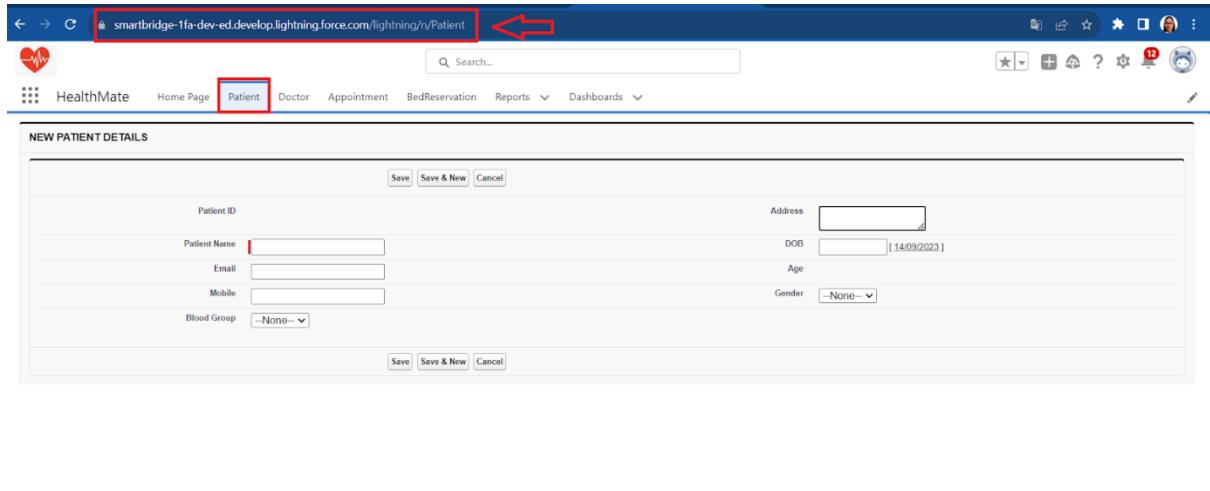
- In the code you find the below line for cancel button, you paste the copied link in action of cancel(link should be in apostrophe)l>>save.

- Replace the action url with your url from Home Page as shown in fig.

```
<apex:commandButton value="Cancel" action="https://smartbridge-1fa-dev-ed.develop.lightning.force.com/lightning/n/Home_Page" immediate="true"/>
```

HomePage Actions:

1. When you're in HealthMate App, Click on Patient tab on Navigation Bar >> copy your browser's url link specific to the Patient Page.



2. Go to setup >> click on gear symbol? developer console ,there you can see many tools in the Toolbar of the new console window>>Click on File>> Open>> Select Entity Type as Pages ?double click on Home Visualforce Page as Entities .

3. In the code you find the below lines



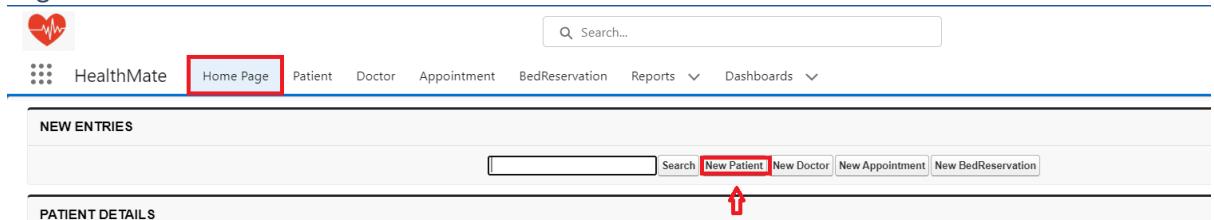
4. Do Paste the copied Browser's link(Patient Page) in action where value="New Patient" >> Save .

5. Follow 1-4 steps for copy and paste(Doctor,Appointment,BedReservation) browser's link where values are "New Doctor","New Appointment",

"New BedReservation"(Make sure you save code after changing actions).

Create A Record (Patient)

1. Click on App Launcher on the left side of the screen.
2. Search HealthMate & click on it.
3. When you're in the HomePage click on the New Patient button which will open the Patient Page .



4. Fill the Details and click on Save.

The screenshot shows the HealthMate application's patient management interface. At the top, there is a navigation bar with icons for search, refresh, and other system functions. Below the navigation bar, the 'Patient' tab is active. The main area is titled 'NEW PATIENT DETAILS'. It contains a form with the following fields:

- Patient ID (input field)
- Patient Name (input field)
- Email (input field)
- Mobile (input field)
- Blood Group (dropdown menu)
- Address (input field)
- DOB (input field with value '14/09/2023')
- Age (input field)
- Gender (dropdown menu with value 'None')

At the top and bottom of the form are buttons labeled 'Save', 'Save & New', and 'Cancel'.

View A Record(Patient)

1. Click on App Launcher on the left side of the screen.
2. Search HealthMate & click on it.
3. When you're in HomePage ,under the Patient title we can view the created record of Patient.
4. Click on any record name. you can see the details of the Patient Details.

Delete A Record(Patient)

1. Click on App Launcher on the left side of the screen.
2. Search HealthMate & click on it.
3. When you're in HomePage,under the Patient title >> we can view the created record of Patient >> Click the delete Button beside the created Record.

Note: Create at least 10 sample records for each of the objects: Patient, Doctor, Appointment and BedReservation.

Create Report :

- 1.Go to the app >> click on the reports tab.
- 2.Click New Report.
- 3.Select report type from category or from report type panel or from search panel >> click on start report.
- 4.Customise your report
- 5.From top on the left side click on report dropdown and select Joined Report and apply.
- 6.Save or run it and give the report name and save it.

Report

1. Go to the app >> click on the reports tab

2. Click New Report.
3. Select report type from category or from report type panel or from search panel (BedReservation with Patient) >> click on start report.
4. Customise your report
5. Save or run it and give the report name and save it.

Report: Bed Reservations with Patient
New Bed Reservations with Patient Report

Total Records: 6

Bed Type	Reservation Date	Start Time	6:00 am			8:00 am			Total
			9:00 am	12:00 pm	Subtotal	9:00 am	10:00 am	Subtotal	
Regular	14/04/2024	Record Count	1	0	1	0	0	0	1
	15/04/2024	Record Count	0	0	0	1	0	1	1
	16/04/2024	Record Count	0	0	0	1	0	1	1
	Subtotal	Record Count	1	0	1	2	0	2	3
Healing	14/04/2024	Record Count	1	0	1	0	0	0	1
	Subtotal	Record Count	1	0	1	0	0	0	1

Details (6 Rows) Click an intersection in the table above to filter details.

Patient: Patient Name	Bed Number	Bed Reservation: Created Date	Booking Status
Raj	R1	13/04/2024	Booked
Mukesh	R1	13/04/2024	Booked
Mukesh	R1	13/04/2024	Booked
Raj	H1	13/04/2024	Booked
Harshita	M2	13/04/2024	Booked
Harshita	M1	13/04/2024	Booked

Row Counts: Detail Rows: Subtotals: Grand Total: Stacked Summaries:

Joined Report
New Appointments with Doctor Name Report

Appointments with Doctor Name			
Appointments with Doctor Name block 1			
Patient Name	Appointment Date	Appointment: Appointment Name	Doctor Name: Doctor Name
Harshita	28/04/2024	A-02	Usha
	Subtotal	Count: 1	
Subtotal		Count: 1	
Raj	13/04/2024	A-01	Ranjit
	Subtotal	Count: 1	Dermatologist
Subtotal		Count: 1	
Total		Count: 2	

Create Dashboard :

1. Go to the app >> click on the Dashboards tabs.

Dashboards

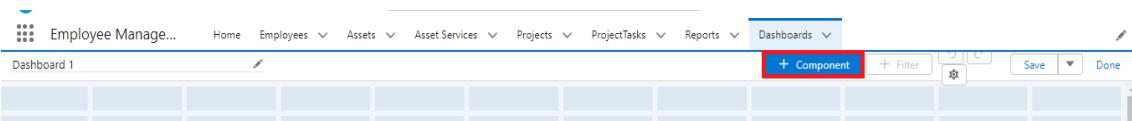
Recent
1 item

Dashboard Name	Description	Folder	Created By	Created On	Subscribed
----------------	-------------	--------	------------	------------	------------

New Dashboard **New Folder**

2. Give a Name and click on Create.

3. Select add component.



4. Select a Report and click on select.

Select Report

Reports

Recent

Created by Me

Private Reports

Public Reports

All Reports

Folders

Created by Me

Shared with Me

New Bed Reservations with Patient Report
Mishra Katoju · 20-Sept-2023, 10:53 am · Private Reports

New Appointments with Doctor Name Report
Mishra Katoju · 20-Sept-2023, 12:20 pm · Private Reports

Sample Flow Report: Screen Flows
Automated Process · 22-Aug-2023, 12:06 pm · Public Reports

5. Select Legend position as bottom and component theme as dark .

Report

New Bed Reservations with Patient | X

Use chart settings from report

Display As

Card

Grid

List

Gauge

Y-Axis

Bed Type

X-Axis

6. Click Add then click on Save and then click on Done.

