

Project Report On Salesforce Health Cloud (DEVELOPER) - (Long - Term)

Milestone – 01:

Create Salesforce Org

Go to developers.salesforce.com/Signup.

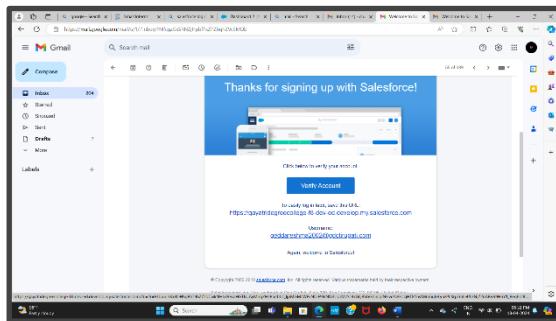
Click on sign up.

On the sign-up form, enter the following details:

1. First name & Last name: GEDDA RESHMA
2. Email: geddareshma2002@gmail.com
3. Role: Developer
4. Company: GAYATRI DEGREE COLLEGE - TIRUPATI
5. Country: India
6. Postal Code: 517501
7. Username:geddareshma2002@gdctirupati.com

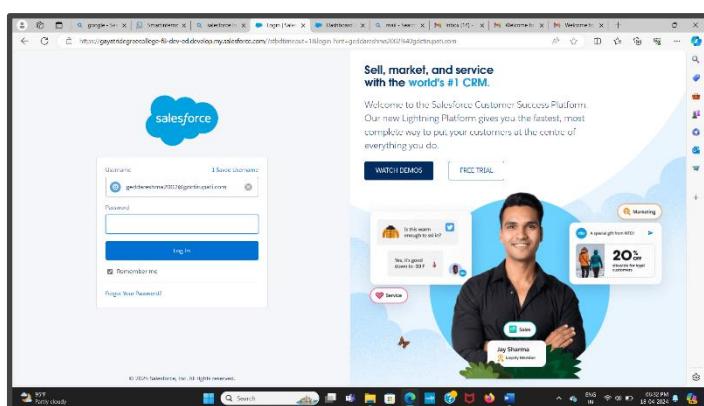
Account Activation

Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email takes 5-10min,



Login to your Salesforce Account

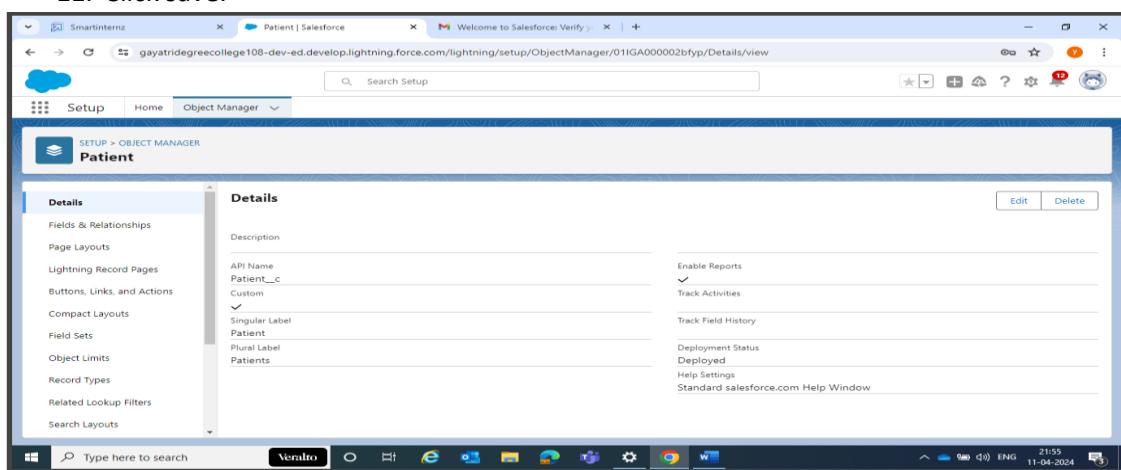
1. Go to [Salesforce.com](https://www.salesforce.com) and Click on login.
2. Enter the username and password that you just created.
3. After login this is the home page you will see.



Milestone – 02: Creation of Objects

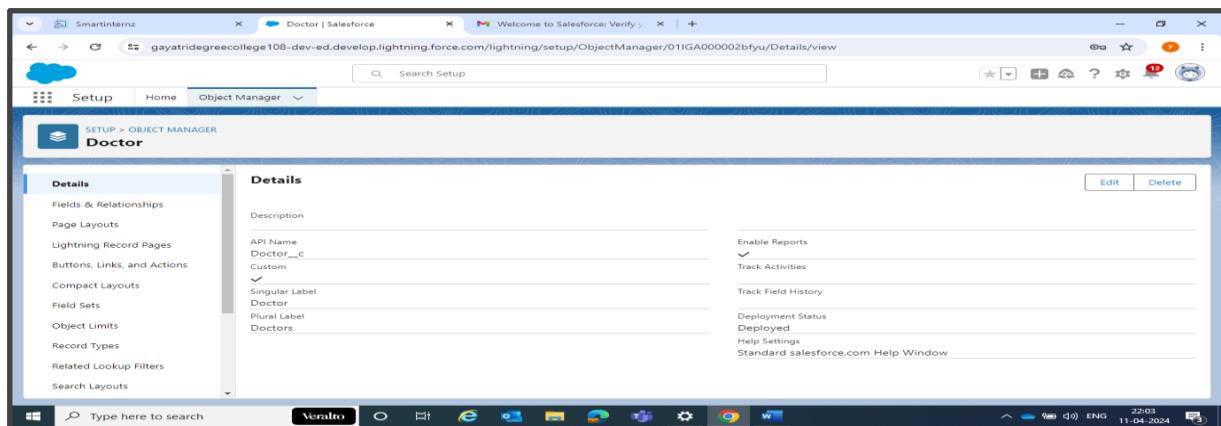
Object – Patient:

1. Click on the gear icon and then select Setup.
2. Click on the object manager tab just beside the home tab.
3. After the above steps, have a look on the extreme right you will find a create drop down click on that and select custom object.
4. On the custom object definition page create the object as follows:
 5. Label: Patient
 6. Plural label: Patients
 7. Record name: Patient Name
 8. Data type: Text
 9. Check the Allow Reports
 10. Check the Allow the Search
 11. Click save.



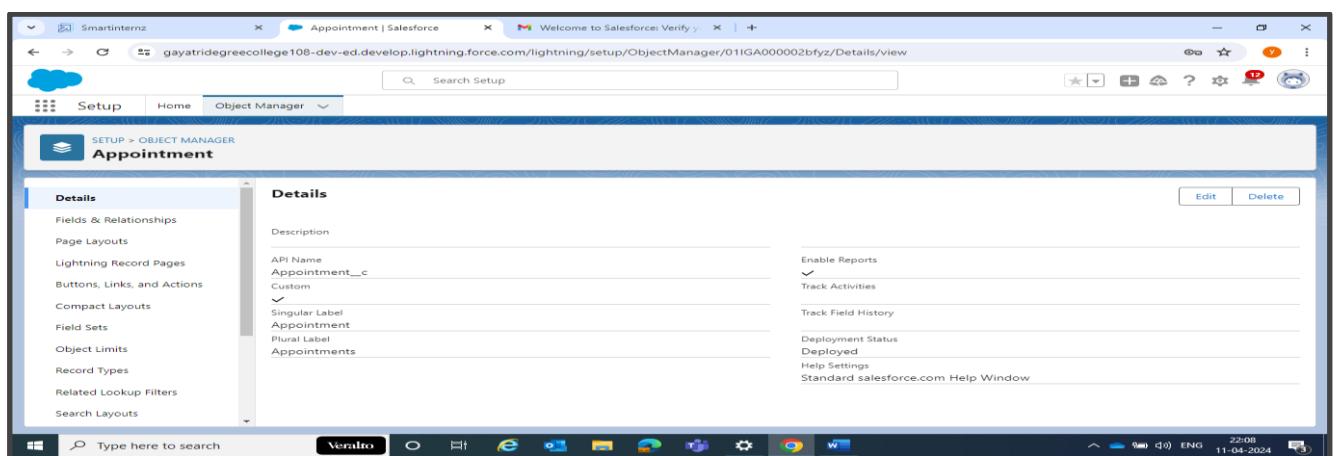
Object – Doctor

1. Click on the gear icon and then select Setup.
2. Click on the object manager tab just beside the home tab.
3. After the above steps, have a look on the extreme right you will find a create drop down click on that and select custom object.
4. On the custom object definition page create the object as follows:
 5. Label: Doctor
 6. Plural label: Doctors
 7. Record name: Doctor Name
 8. Data type: Text
 9. Check the Allow Reports
 10. Check the Allow the Search
 11. Click save.



Object: Appointment

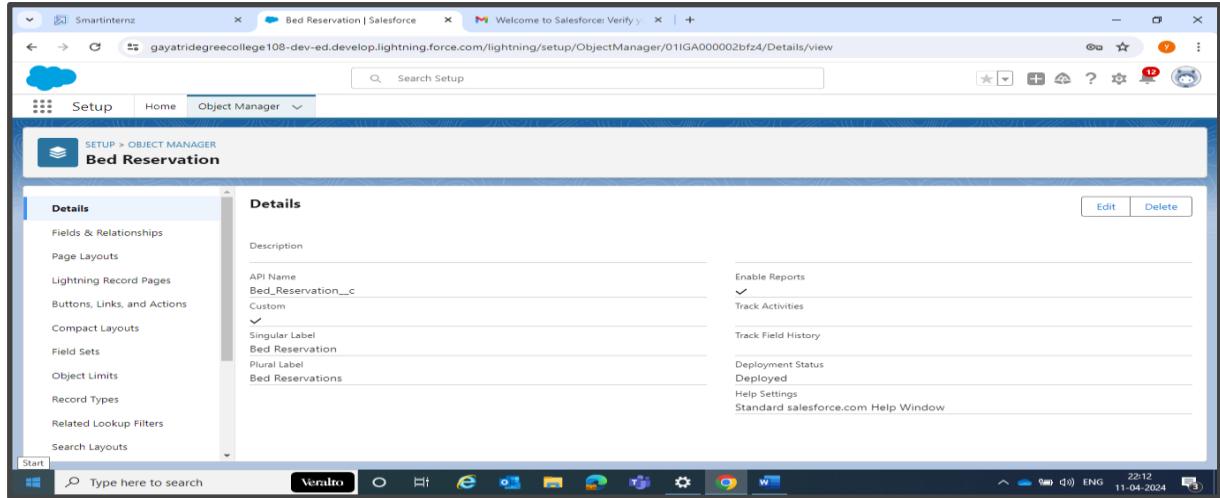
1. Click on the gear icon and then select Setup.
2. Click on the object manager tab just beside the home tab.
3. After the above steps, have a look on the extreme right you will find a create drop down click on that and select custom object.
4. On the custom object definition page create the object as follows:
5. Label: Appointment
6. Plural label: Appointments
7. Record name: Appointment No
8. Data type: Auto number (A- {00})
9. Check the Allow Reports
10. Check the Allow the Search
11. Click save.



Object: Bed Reservation

1. Click on the gear icon and then select Setup.
2. Click on the object manager tab just beside the home tab.
3. After the above steps, have a look on the extreme right you will find a create drop down click on that and select custom object.
4. On the custom object definition page create the object as follows:
5. Label: Bed Reservation
6. Plural label: Bed Reservations
7. Record name: Bed Reservation No

8. Data type: Auto number (B- {00})
9. Check the Allow Reports
10. Check the Allow the Search
11. Click save.

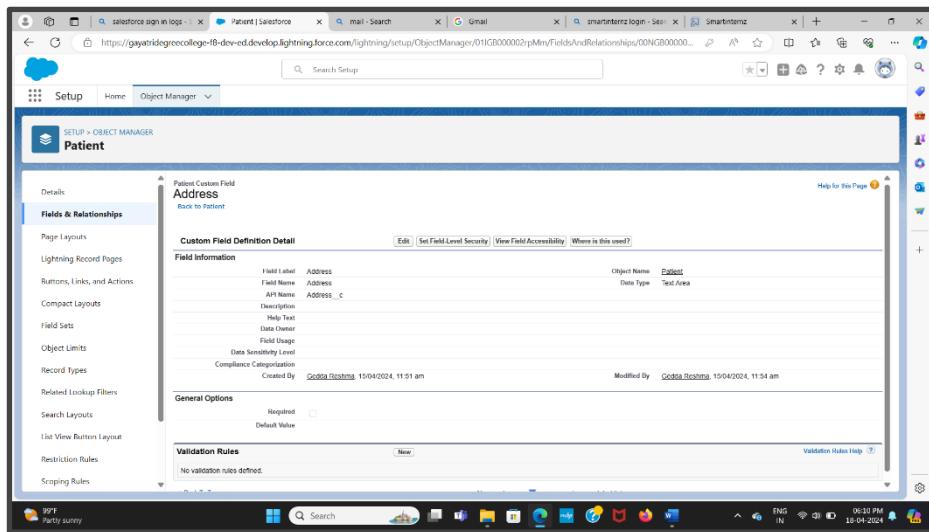


Milestone 03: fields and relationships:

Creating Text Area field on Parent Object:

To create fields in an object:

1. Data type as “Text Area”.
2. Click on Next
3. Fill the above as following:
4. Click the gear icon and select setup .
5. click on Object Manager and type object name - Patient in quick find bar.
6. click on the patient object.
7. Now click on “Fields & Relationships” and click New
 - Select Field Label: Address
 - Field Name: gets auto generated.
 - Click on Next >> Next >> Save and new.



Creating the Date Field on object – patient

To create fields in an object:

1. Click the gear icon and select setup and click on Object Manager and type object name - Patient in quick find bar.
2. Click on the Patient object.
3. Now click on “Fields & Relationships” and select New
4. Select Data type as “Date” and click Next.
5. Given the Field Label as “DOB”.
6. Field Name will be auto populated and click on Next >> Next >> Save.

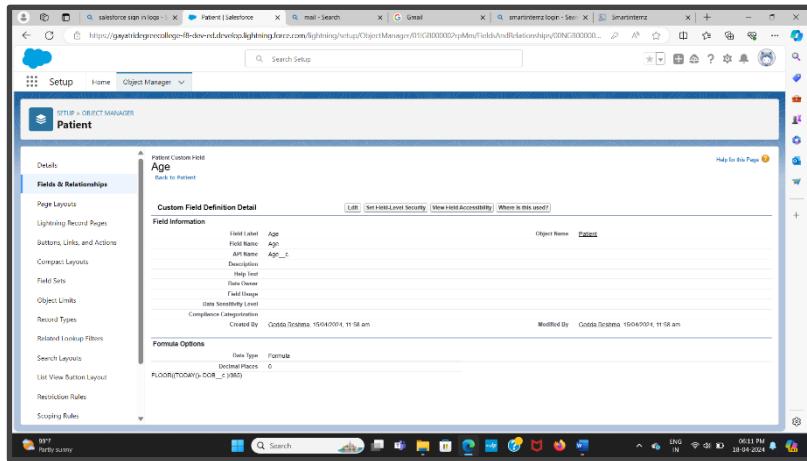
Fields & Relationships				
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text Area(255)		
Age	Age__c	Formula (Number)		
Blood Group	Blood_Group__c	Picklist		
Created By	CreatedById	Lookup(User)		
DOB	DOB__c	Date		
Email	Email__c	Email		
Gender	Gender__c	Picklist		

Creating Formula Field on Patient Object

To create fields in an object:

1. Click the gear icon and select setup and click on Object Manager and type object name - Patient in quick find bar.
2. Click on the Patient object.
3. Now click on “Fields & Relationships” and select New.
4. Select Data type as “Formula” and click Next.
5. 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.

6. Under Advanced Formula write down the formula: FLOOR ((TODAY ()- DOB__c)/365).
7. click “Check Syntax” and Next >> Next>> Save & New.



Creating Picklist Field on Patient Object :

To create fields in an object:

1. Click the gear icon and select setup and click on Object Manager and type object name - Patient in quick find bar.
2. Click on the Patient object.
3. Now click on “Fields & Relationships” and select New.
4. Select Data type as “Picklist” and click Next.
5. Enter Field Label as “Blood Group”.
6. 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.
7. Click Next>> Next >> Next >> Save .

Fields & Relationships					
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED	
Address	Address__c	Text Area(255)			
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			

Creating the Email Field on Patient Object

To create fields in an object:

1. Click the gear icon and select setup and click on Object Manager and type object name - Patient in quick find bar.

2. Click on the Patient object.
3. Now click on “Fields & Relationships” and select New.
4. Select Data type as “Email” and click Next.
5. Given the Field Label as “Email”.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address_c	Text Area(255)		
Age	Age_c	Formula (Number)		
Blood Group	Blood_Group_c	Picklist		
Created By	CreatedById	Lookup(User)		
DOB	DOB_c	Date		
Email	Email_c	Email		
Gender	Gender_c	Picklist		

Creating the Phone Field on patient Object:

1. Click the gear icon and select setup and click on Object Manager and type object name - Patient in quick find bar.
2. Click on the Patient object.
3. Now click on “Fields & Relationships” and select New.
4. Select Data type as “Phone” and click Next.
5. Given the Field Label as “ Mobile”.

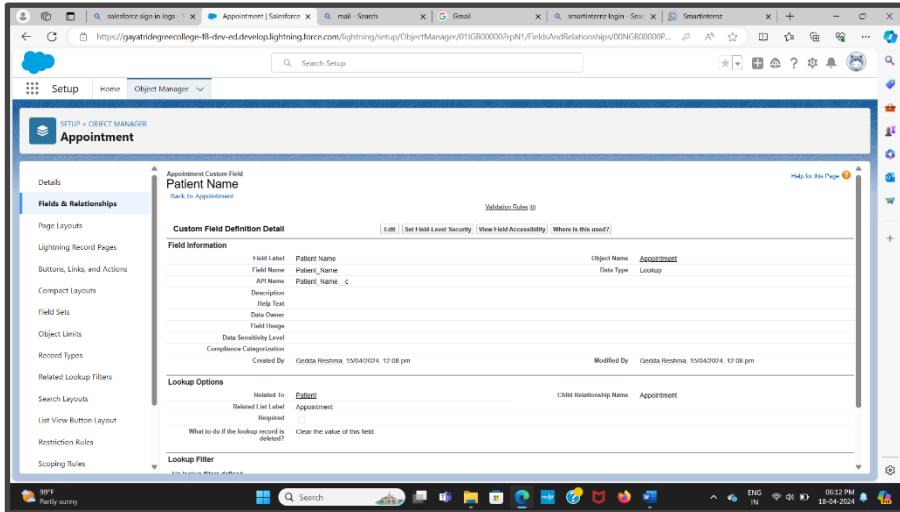
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
DOB	DOB_c	Date		
Email	Email_c	Email		
Gender	Gender_c	Picklist		
Last Modified By	LastModifiedById	Lookup(User)		
Mobile	Mobile_c	Phone		
Owner	OwnerId	Lookup(User,Group)		
Patient ID	Patient_ID_c	Auto Number		
Patient Name	Name	Text(80)		

Creating Lookup Relationship

To Create a relationship between Appointment & Patient Objects.

1. Click the gear icon and select setup and click on Object Manager and type object name - Patient in quick find bar.
2. Click on the Patient object.
3. Now click on “Fields & Relationships” and select New.

4. Select “Lookup relationship” as data type and click Next.
5. Select the related object “Patient”.
6. Give Field Label as “Patient Name” and click Next.
7. Next >> Next >> Save.



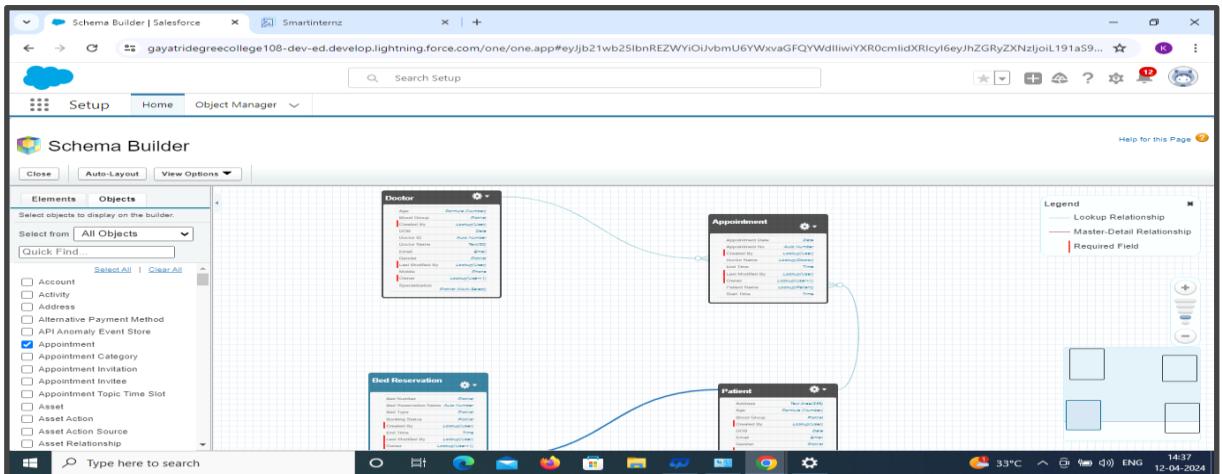
Creating Remaining Fields on object:

Now create the remaining fields using the data types mentioned.

Schema Builder:

Creating Schema Builder

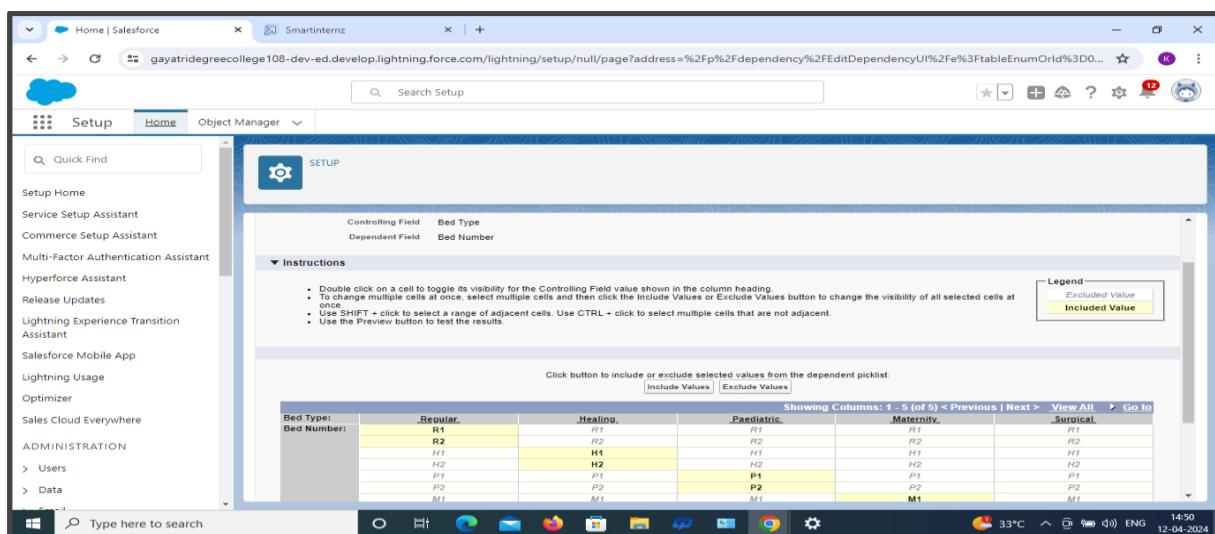
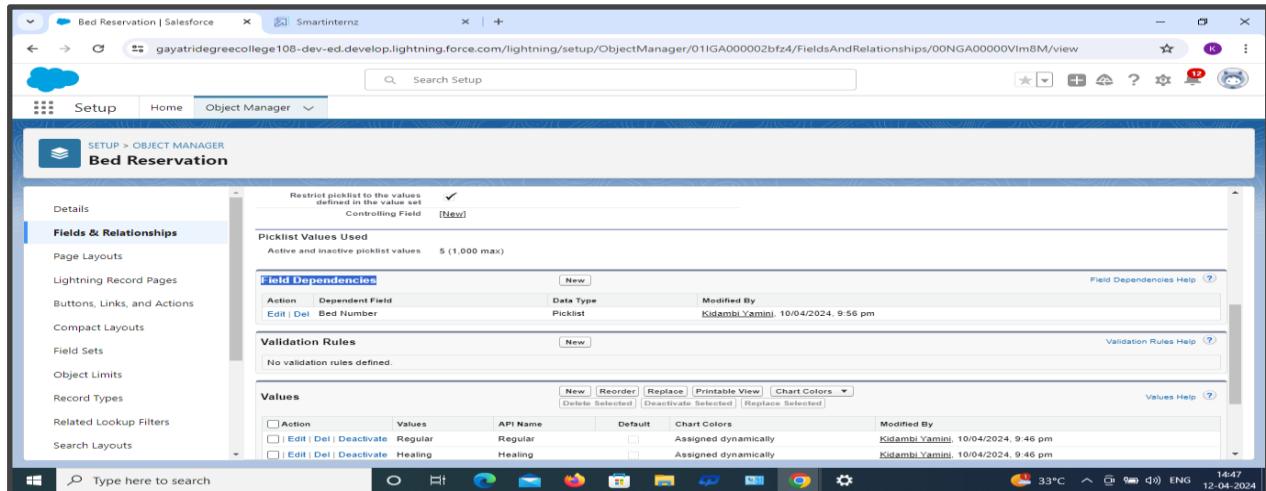
1. Go to setup >>click on Object Manager >> Schema Builder.
2. Select objects >> Enter Objects as “Patient, Doctor, Appointment and Bed Reservation objects” in a quick box and select them.



Creating the Field Dependencies :

1. Click the gear icon and Go to setup.
2. click on Object Manager and type object name- Bed Reservation in the quick find bar and click on the selected object.
3. Fields & Relationships and click on the Bed Type field.

4. Search for Field Dependencies and click on New.
5. Select Controlling Field as “Bed Type” and Depending field as “Bed Number” and Continue.
6. Select the “Bed Number” Click on values of related “Bed Type” values like for regular select R1,R2 and Click on Include Values >> Save.



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

The top screenshot shows the 'Validation Rule Edit' screen for the 'Patient' object. A validation rule named 'ValidationForDOB' is being created with the formula `DOB__c >= TODAY()`. The bottom screenshot shows the 'Validation Rules' list screen, which displays one item: 'ValidationForDOB'.

Rule Name	Error Location	Error Message	Active	Modified By
ValidationForDOB	DOB	Check DOB	✓	Gedis Rehma, 15/04/2024, 3:34 pm

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 and Visualforce Page :

Creating Apex Controller Class For Patient Object:

1. Go to setup , click on gear symbol and select developer console.
2. There you can see many tools in the Toolbar of the new console window. Click on File, New and Apex Class.
3. Enter the name as “ patient controller ” for Apex Page to create a new Apex Page file.
4. Enter the code.
5.

```
public class patient_controller {
```
6.

```
    public Patient__c p {get; set;}
```
7.

```
    public patient_controller(){}
```
8.

```
        p = new Patient__c();
```
9.

```
}
```
10.

```
    public PageReference save() {
```
11.

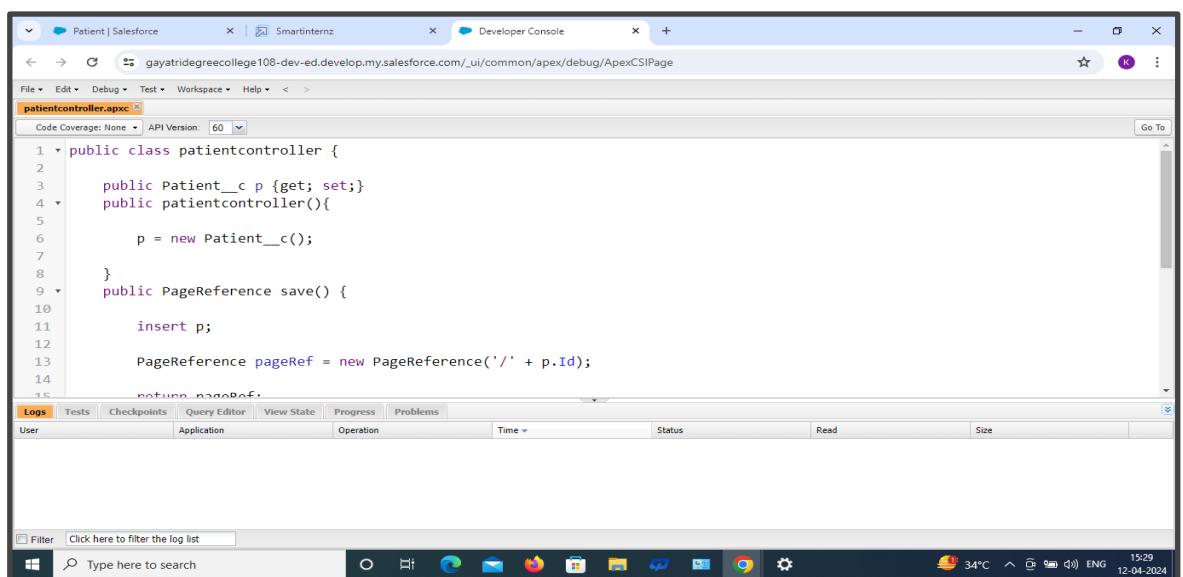
```
        insert p;
```
12.

```
        PageReference pageRef = new PageReference('/'+ p.Id);
```

```

13.     return pageRef;
14. }
15. public PageReference save new() {
16.     insert p;
17.     PageReference PageRef =Page.patient;
18.     p.clear();
19.     return PageRef;
20. }
21. }

```



Creating Visualforce Page For Patient Object :

1. Go to setup, click on gear symbol and select developer console.
2. There you can see many tools in the Toolbar of the new console window. Click on File, New and Visualforce page.
3. Enter the name as Patient for Apex Page to create a new Visualforce Page file.
4. Enter the code.

Patient Visualforce Page:

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for 'Patient | Salesforce', 'Smartinternz', 'Developer Console', and several other tabs related to the project. Below the tabs, the main area displays the Apex code for a page named 'patientcontroller.apex'. The code defines a page with a sidebar and header disabled, using a 'patient controller' as the controller. It contains a form with a page block titled 'NEW PATIENT DETAILS'. This block includes a section with input fields for Patient ID, Address, Name (which is required), DOB, Email, Age, Mobile, Gender, and Blood Group. At the bottom of the page block, there are buttons for Save, Save & New, and Cancel. The code concludes with closing tags for the page and form.

```

<apex:page sidebar="false" showHeader="false" controller="patient controller">
<apex:form>
    <apex:pageBlock title="NEW PATIENT DETAILS">
        <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="{!save new}"/>
            <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>

```

Code :

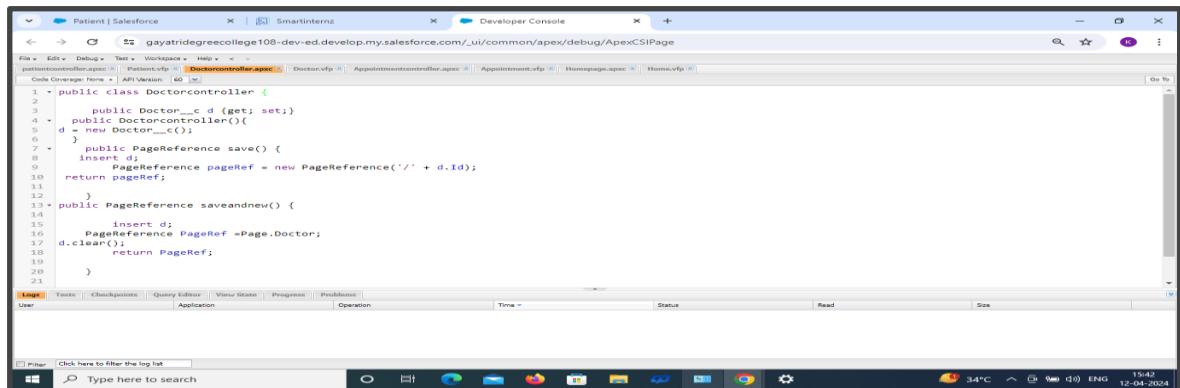
```

<apex:page sidebar="false" showHeader="false" controller="patient controller">
<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="{!save new}"/>
                <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:pageBlock>
</apex:form>
</apex:page>

```

Creating Apex Controller Class For Doctor Object:

Doctorcontroller Class Page:



The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for Patient | Salesforce, Smartinternz, and Developer Console. Below the tabs, there are tabs for Doctor.vfp, Appointmentcontroller.apex, Appointment.vfp, Homepage.apex, and Home.vfp. The main area displays the code for the Doctorcontroller class:

```
1 * public class Doctorcontroller {
2     public Doctor__c d {get; set;}
3     public Doctorcontroller(){
4         d = new Doctor__c();
5     }
6     public PageReference save() {
7         insert d;
8         PageReference pageRef = new PageReference('/' + d.Id);
9         return pageRef;
10    }
11 }
12 * public PageReference saveandnew() {
13     insert d;
14     PageReference PageRef =Page.Doctor;
15     d.clear();
16     return PageRef;
17 }
18 */
19 */
20 */
21 */

Log Test Checkpoint Query Editor View State Progress Problems
```

Code :

```
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 save new() {  
        insert d;  
        PageReference PageRef =Page.Doctor;  
        d.clear();  
        return PageRef;  
    }  
}
```

Creating Visualforce Page For Doctor Object:

Doctor Visualforce Page:

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for Patient | Salesforce, Smartinternz, and Developer Console. Below the tabs, there's a file list with patientcontroller.apex, Patient.vfp, Doctorcontroller.apex, Doctor.vfp (highlighted in orange), Appointmentcontroller.apex, Homepage.apex, Home.vfp, and Appointment.vfp. The main area displays the Apex code for Doctor.vfp:

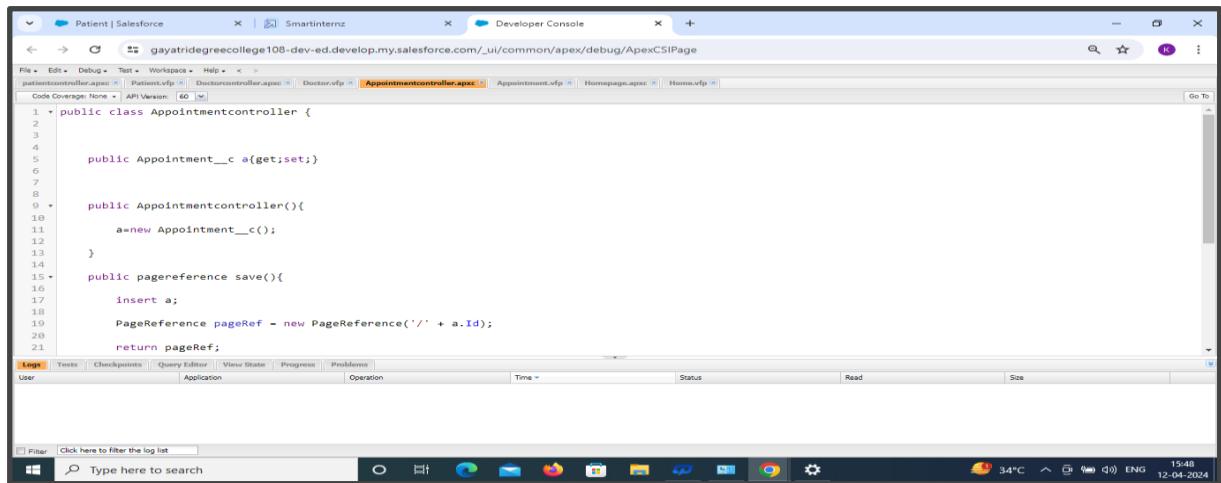
```
<apex:page sidebar="false" showHeader="false" Controller="Doctorcontroller">
<apex:form>
<apex:pageBlock title="NEW DOCTOR DETAILS">
<apex:pageBlockSection>
<apex:inputField value="{!d.Doctor_ID__c }"/>
<apex:inputField value="{!d.Gender__c }"/>
<apex:inputField value="{!d.Name}" required="true"/>
<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:pageBlockSection>
<apex:pageBlockButtons>
<apex:commandButton value="Save" action=" {!save} " />
<apex:commandButton value="Save & New" action=" {!save new} " />
<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>
```

Code :

```
<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=" {!save new} " />
<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:

Appointment Controller Class Page:



The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for Patient | Salesforce, Smartinternz, and Developer Console. Below the tabs, the code editor displays the Appointmentcontroller.apex file. The code defines a class named Appointmentcontroller with methods for saving a new appointment record. The developer console also shows a log tab at the bottom.

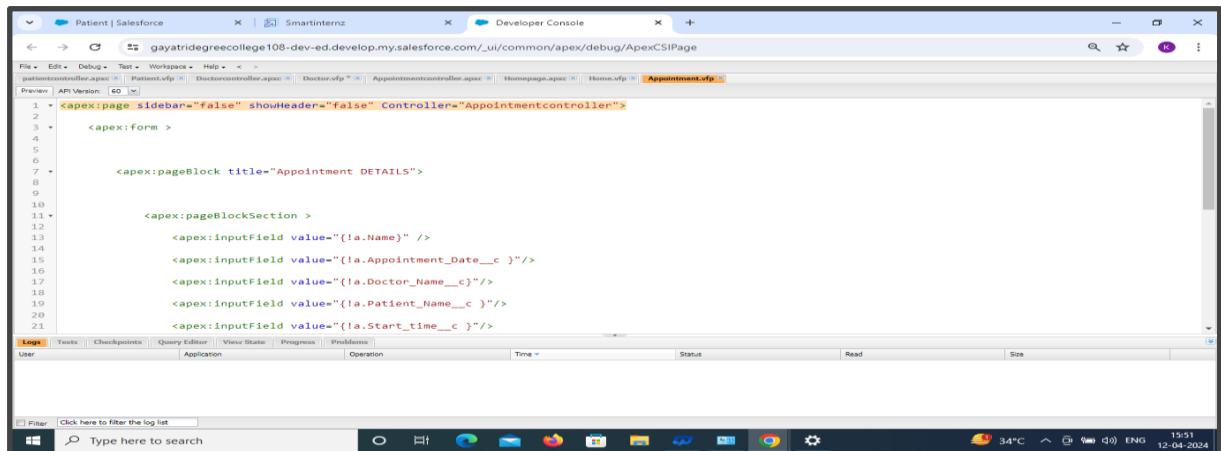
```
1 public class Appointmentcontroller {
2
3     public Appointment__c a{get;set;}
4
5     public Appointmentcontroller(){
6         a=new Appointment__c();
7     }
8
9     public pagereference save(){
10        insert a;
11        PageReference pageRef = new PageReference('/' + a.Id);
12        return pageRef;
13    }
14
15 }
```

Code :

```
public class Appointment Controller {  
    public Appointment__c a{get;set;}  
    public Appointments Controller(){  
        a=new Appointment__c();  
    }  
    public pagereference save(){  
        insert a;  
        PageReference pageRef = new PageReference('/' + a.Id);  
        return pageRef;  
    }  
    public pagereference save new(){  
        insert a;  
        pagereference pageref =Page.Appointment;  
        a.clear();  
        return pageref;  
    }  
}
```

Creating visual page for appointment for object :

Appointment VisualForce Page:



Code:

```

<apex:page sidebar="false" showHeader="false" Controller="Appointment
Controller">
<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="{!save new}" />
            <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

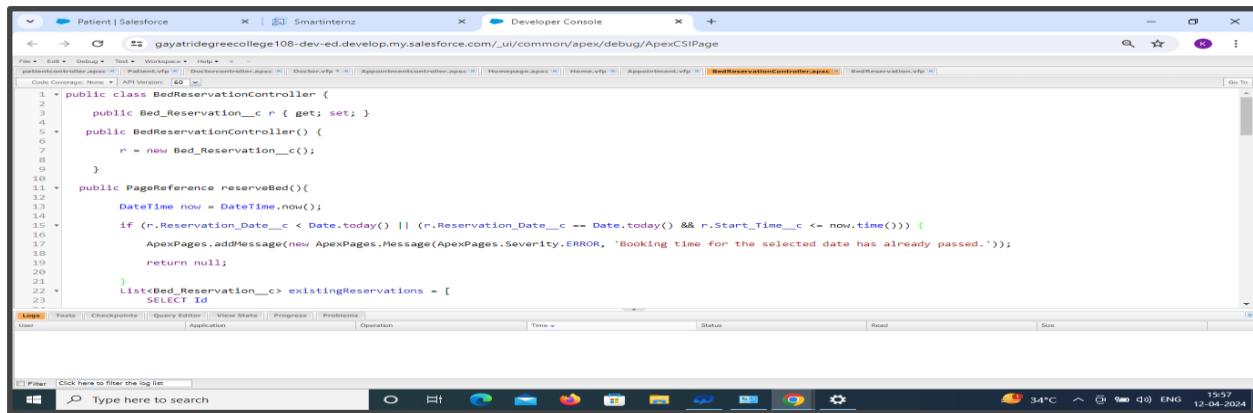
BedReservationcontroller Class Page:

Code :

```

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;
}

```

```

}

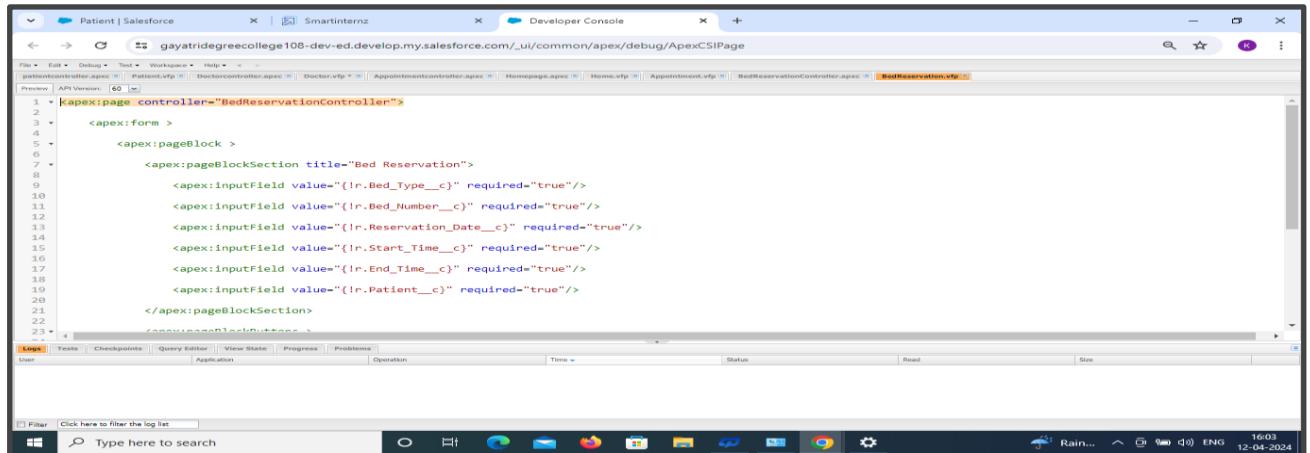
public PageReference reserve new(){
    DateTime now = DateTime.now();
    if (r.Reservation_Date__c < Date.today() || (r.Reservation_Date__c == Date.today() &&
r.Start_Time__c <= now.time())) {
        ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'Booking time
for the selected 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 :

NOTE: Follow the same steps as mentioned in activity 2.

BedReservation Visualforce Page



The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes 'Patient | Salesforce', 'Smartinternz', and 'Developer Console'. Below the navigation is a tabs section with 'patientcontroller.apex', 'patientcontroller.vfp', 'DoctorController.apex', 'AppointmentController.apex', 'Homepage.apex', 'Home.vfp', 'Appointment.vfp', 'BedReservationController.apex', and 'BedReservation.vfp' (which is currently selected). The main area displays the Apex code for the BedReservation.vfp page. The code defines a form with a page block containing five input fields (Bed Type, Bed Number, Reservation Date, Start Time, End Time) and a page block section with a command button to reserve a bed. The code uses the BedReservationController as the page controller. The bottom of the screen shows the Windows taskbar with various application icons.

```
<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="reserve new" action="!reserve new" />
            <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>
```

Code :

```
<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="reserve new" action="!reserve new" />
            <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:

NOTE: Follow the same steps as mentioned in Activity 1.

HomePage Class Page:

```

1 + public class Homepage {
2
3     public List<Patient__c> patlists { get; set; }
4
5     public List<Doctor__c> doclists { get; set; }
6
7     public List<Appointment__c> apolists { get; set; }
8
9     public List<Bed_Reservation__c> BedReserlists { get; set; }
10
11    public List<Bed_Reservation__c> matchingReservationslists { get; set; }
12
13    public String searchCriteria { get; set; }
14
15    public String selectedRecordId { get; set; }
16
17    public Homepage() {
18
19        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];
20
21        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];
22
23        apolists = [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];
24
25        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];
26
27    }
28

```

Creating Visualforce Page For Home Tab

Home Visualforce Page:

- Visualforce name should be “Home”.

```

1 <apex:page showHeader="false" sidebar="true" Controller="Homepage"
2
3     docType="html-5.0" >
4
5         <style type="text/css">
6             body {
7                 background-image:
8                     url('https://www.prowesssoft.com/wp-
9                     content/uploads/2021/03/SalesforceHealthCareImage-B.jpg');
10
11                background-size: cover;
12                background-repeat: no-repeat;
13                background-attachment: fixed;
14            }
15
16            p {
17                font-weight: bold;
18            }
19
20
21             <apex:inputText id="searchInput" style="color: #0000ff; outline: none;" placeholder="Search Criteria" value="searchCriteria" />
22
23             <apex:commandButton value="Search" action="{!performSearch}" rerender="resultPanel" />
24
25             <apex:commandButton value="New Patient" action="https://gatrdegrecollge108-dev-ed.lightning.force.com/lightning/n/Patient"/>
26
27             <apex:commandButton value="New Doctor" action="https://gatrdegrecollge108-dev-ed.lightning.force.com/lightning/n/Doctor"/>
28
29             <apex:commandButton value="New Appointment" action="https://gatrdegrecollge108-dev-ed.lightning.force.com/lightning/n/Appointment"/>
30
31             <apex:commandButton value="New BedReservation" action="https://gatrdegrecollge108-dev-ed.lightning.force.com/lightning/n/BedReservation"/>
32
33         </apex:pageBlock>
34
35         <apex:outputPanel id="resultPanel">
36             <apex:pageBlock title="PATIENT DETAILS">
37
38                 <apex:pageBlockTable value="{!patlists}" var="p">
39                     <apex:column style="align:left" headerValue="Patient Name">
40                         <apex:outputText value="!{p.Name}"/>
41                     </apex:column>
42                     <apex:column style="align:left" headerValue="Mobile">
43                         <apex:outputText value="!{p.Mobile__c}"/>
44                     </apex:column>
45
46                     <apex:column style="align:left" headerValue="Appointments">
47                         <apex:repeat value="!{p.Appointments__r}" var="b">
48                             <apex:outputLink value="!{b.Id}!{b.Name}"/>
49                         </apex:repeat>
50                     </apex:column>
51
52                 </apex:pageBlockTable>
53             </apex:pageBlock>
54         </apex:outputPanel>
55
56     </apex:page>
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Code :

```

<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:pageBlockTable>
        </apex:pageBlock>
    </apex:outputPanel>

```

```

<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}"
rerender="resultsPanel">
<apex:param name="selectedRecordId" value="{!!B.Id}"
assignTo="{!!selectedRecordId}" />
</apex:commandButton>
</apex:column>
</apex:pageBlockTable>
</apex:pageBlock>
</apex:outputPanel>
</apex:form>
</apex:page>

```

Tabs :

What is Tab:

A tab is like a user interface that is used to build records for objects and to view the records in the objects.

To create a Tab:

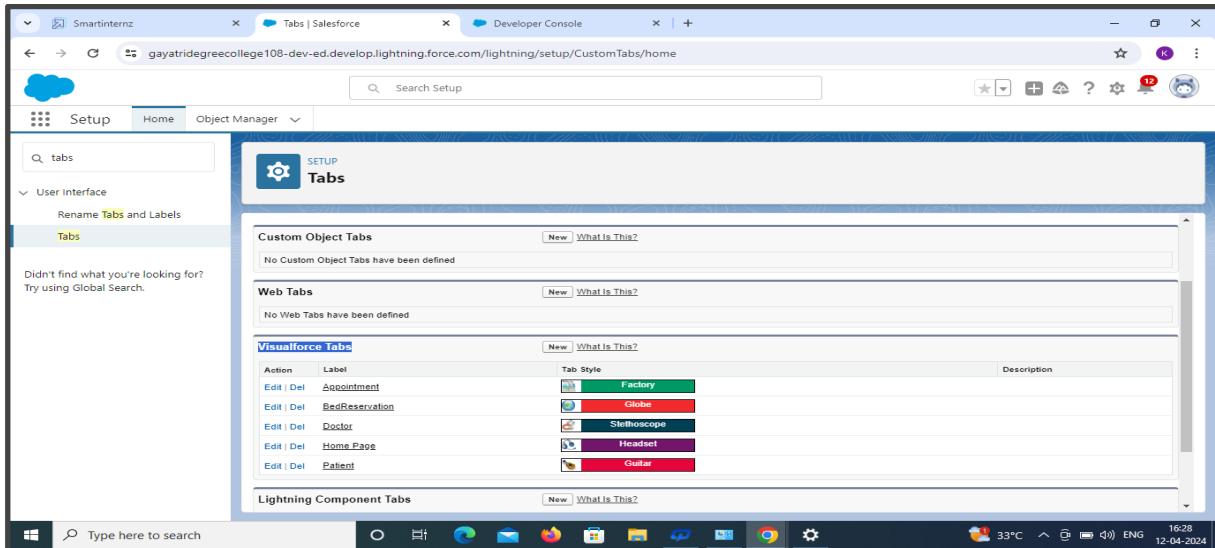
Patient :

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

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.

Note :

Create 4 more tabs with Doctor, Appointment, Home page and BedReservation.



The Lightning App :

Create A Lightning App :

1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App. Fill the app name in app details and branding as follow

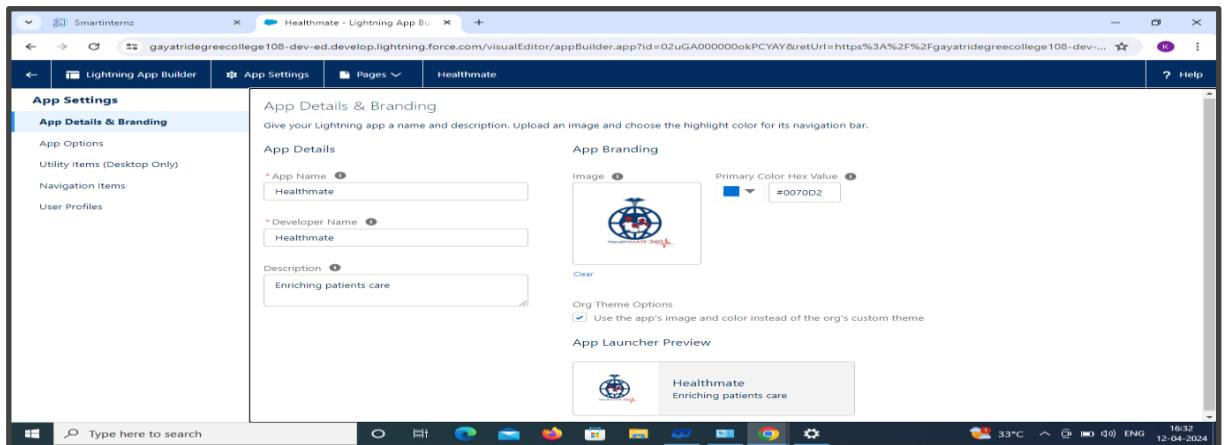
App Name : Healthmate

Developer Name : This will auto populated

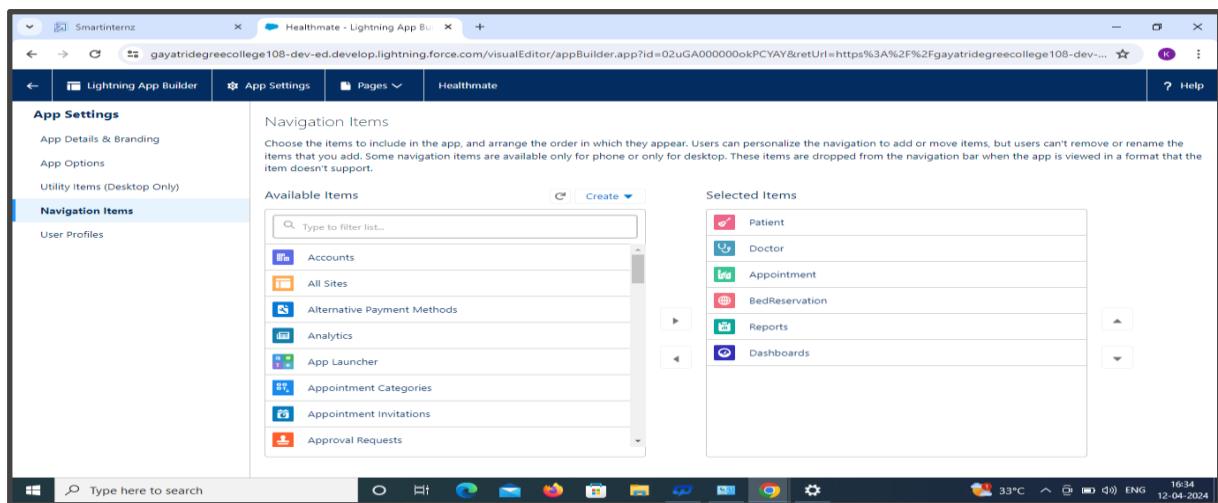
Description : Enriching patients care

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

1. Then click Next >> (App option page) Set Navigation Style as Standard Navigation >> Next.
2. (Utility Items) keep it as default >> Next.



3. To Add Navigation Items:



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.

4. To Add User Profiles:

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

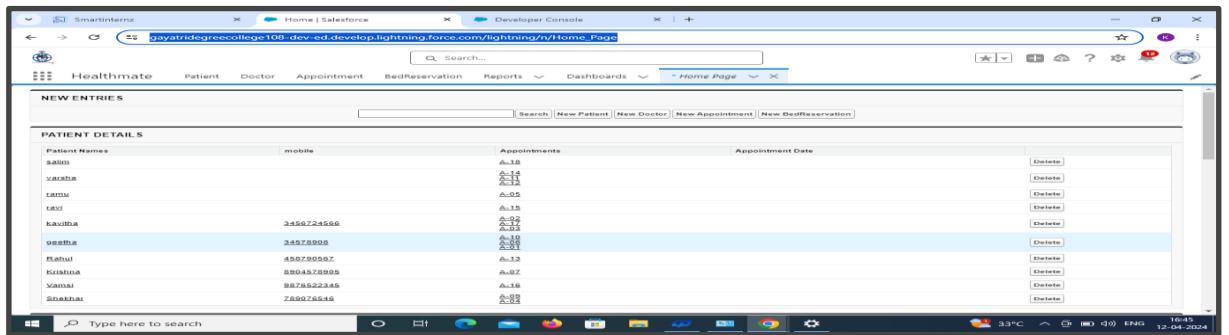
Change Actions:

Cancel Actions:

1. Go to the setup page >> click “app Launcher” >>search for HealthMate.
2. This will be a user Interface which we get:
3. Click on the HomePage tab on Navigation Bar >> copy your browser's link specific to the HomePage.
4. 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
5. Entity Type as Pages >> double click on Patient in Entities .

Go to setup >> click on gear symbol?developer console >> there you can see many tools in the

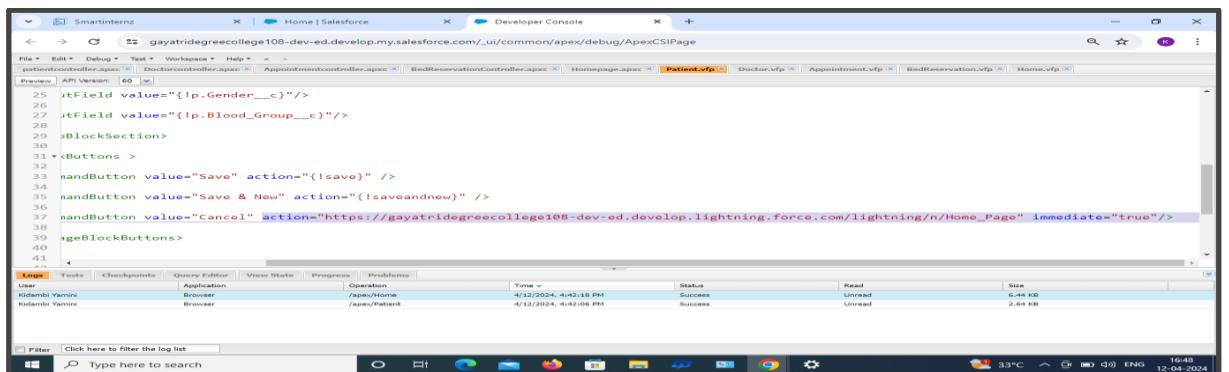
Toolbar of the new console window >> Click on File>> Open >> Select Entity Type as Pages >> double click on Patient in Entities .



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 form Home Page as shown in fig.

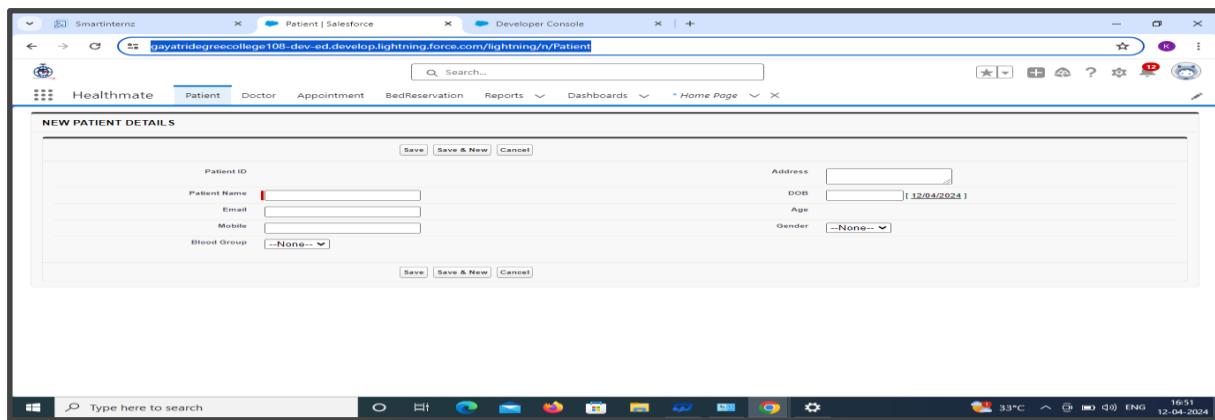
```
<apex:commandButton value="Cancel"
action="https://smartbridge-1fa-dev-
ed.develop.lightning.force.com/lightning/n/Home_Page" immediate="true"/>
```



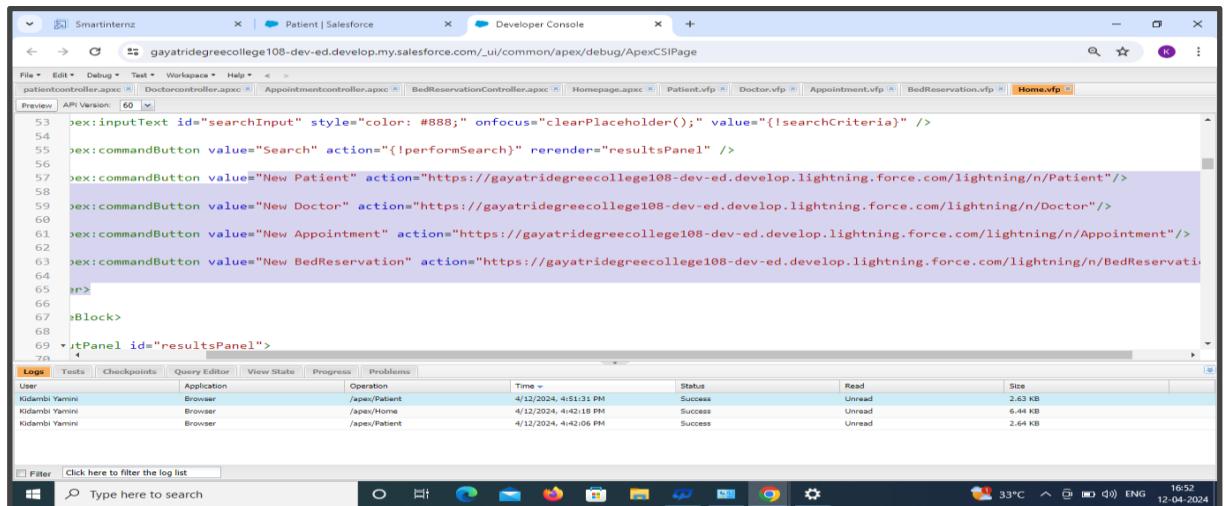
Note :Do Paste the same Browser's link(HomePage) of Action in Doctor,Appointment and BedReservation Visualforce Pages(Make sure you save code after changing actions)

HomePage Actions:

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



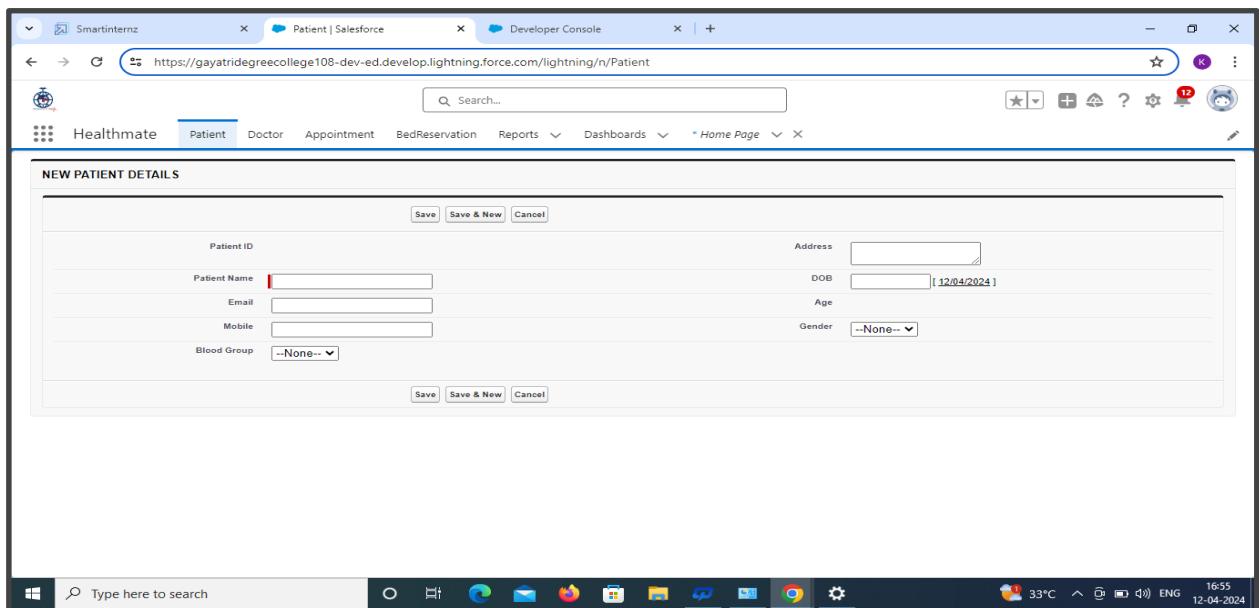
2. Go to setup >> click on gear symbol?developer console ,there you can see many tools in the Toolbar of the new console window>>Click on File>> Open>> Select Entity Type as Pages ?double click on the Home Visualforce Page as Entities .
3. In the code you find the below lines.
4. Do Paste the copied 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).



User Adoption:

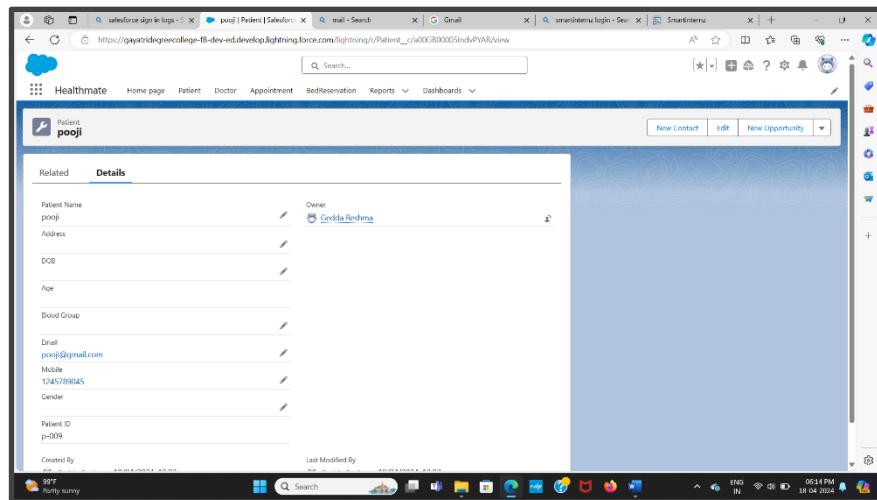
Create A Record (Patient)

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.



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.

Report :

Create Report:

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

1. 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
 - From top on the left side click on report dropdown and select Joined Report and apply.
 - Add fields from the left pane as shown below.
5. Save or run it and give the report name and save it.

The screenshot shows a Joined Report titled "New Appointments with Doctor Name Report". The report structure includes a header section for "Appointments with Doctor Name" and "Appointments with Doctor Name block 1". The main body contains a grid of data with columns for "Patient Name", "Appointment Date", "Doctor Name", and "Specialization". The data is organized by patient (Subtotal for kavitha, Krishna, and Subtotal) and then by doctor (Subtotal for A-01, A-10, A-17, A-02, A-03, A-07, and Varsha). Each row shows the doctor's name and specialization. The report also includes summary rows for each patient group and a grand total at the bottom.

Report:

Go to the app >> click on the reports tab:

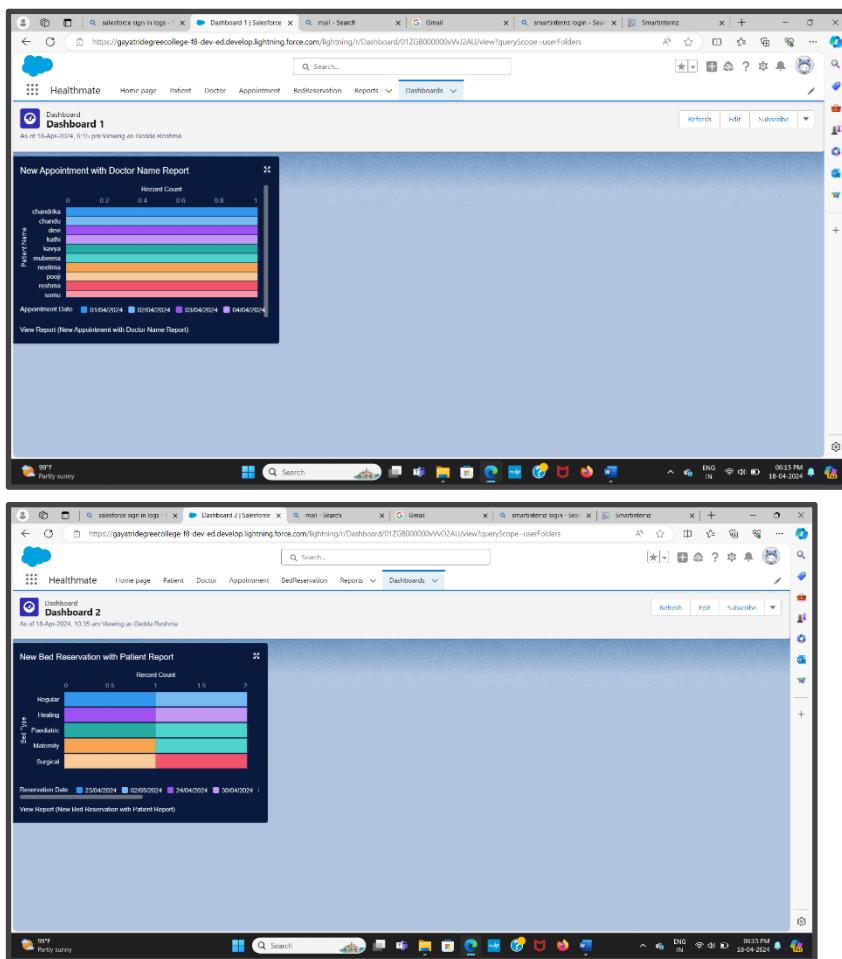
1. Click New Report.
2. Select report type from category or from report type panel or from search panel (BedReservation with Patient) >> click on start report.
3. Customise your report
 - Add fields from the left pane as shown below.
 - Save or run it and give the report name and save it.

The screenshot shows a Joined Report titled "New Bed Reservations with Patient Report". The report structure includes a header section for "Total Records" and "11". The main body contains a grid of data with columns for "Bed Type", "Reservation Date", "Start Time", "End Time", and "Patient: Patient Name". The data is organized by bed type (Regular, Healing), reservation date (20/04/2024, 26/04/2024, 19/04/2024), and patient name (Krishna, Ramu, geetha). The report also includes summary rows for each bed type and date group, and a detailed table at the bottom showing individual bed reservations with columns for Patient Name, Bed Number, Booking Status, and Bed Reservation: Created Date.

Dashboards :

Create Dashboard:

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



---THE END---

