Salesforce Project Phase-1

Healthcare Patient Management and Telemedicine CRM

Phase 1: Problem Understanding & Industry Analysis

- **Requirement Gathering:**
  - **Patients:**
    1. Easy registration & onboarding
    2. Online/offline appointment booking
    3. Telemedicine consultation
    4. Access to prescriptions & medical history
  - **Doctors:**
    1. Patient record management
    2. Appointment scheduling
    3. Add or update prescriptions
    4. View consultation history
  - **Hospital Admins:**
    1. Manage doctors, patients, and staff
    2. Billing and reports
    3. Security and compliance with medical standards
- **Stakeholder Analysis:**
  - **Primary Stakeholders:** Patients, Doctors, Hospital Admin
  - **Secondary Stakeholders:** Receptionists or office staff, Pharmacy, Insurance providers
- **Business Process Mapping:**

  Flow chart: Patient Registration -> Appointment Booking ->Doctor Consultation ->Prescription Generation -> Follow-up and Billing -> Reports and Analytics
- **Industry-Specific Use Case Analysis:**

  Problem in Healthcare:
  - Hospitals struggle with fragmented patient records, making it hard for doctors to see a complete medical history.
  - Long waiting times for appointments due to manual scheduling.
  - Lack of centralized telemedicine support, forcing patients to use external apps.
  - No real-time hospital bed availability tracking, especially during emergencies

In Salesforce how can we solve it:

- Patient 360 View → A unified record for each patient with history, prescriptions, and lab reports.
- Automated Appointment Scheduling → Flows & Process Builder to reduce waiting time.
- Telemedicine Integration → Embed Zoom or Teams calls within Salesforce for online consultations.
- Bed Availability Management → Real-time updates via dashboards so doctors/admins see availability instantly.

❖ **AppExchange Exploration:**

- **Health Cloud:** inspiration for patient management
- **Zoom or Teams Integration Apps:** for telemedicine video calls
- **DocuSign for Salesforce:** for digital prescriptions & approvals
- **Payment Gateway Apps:** for billing and insurance payments

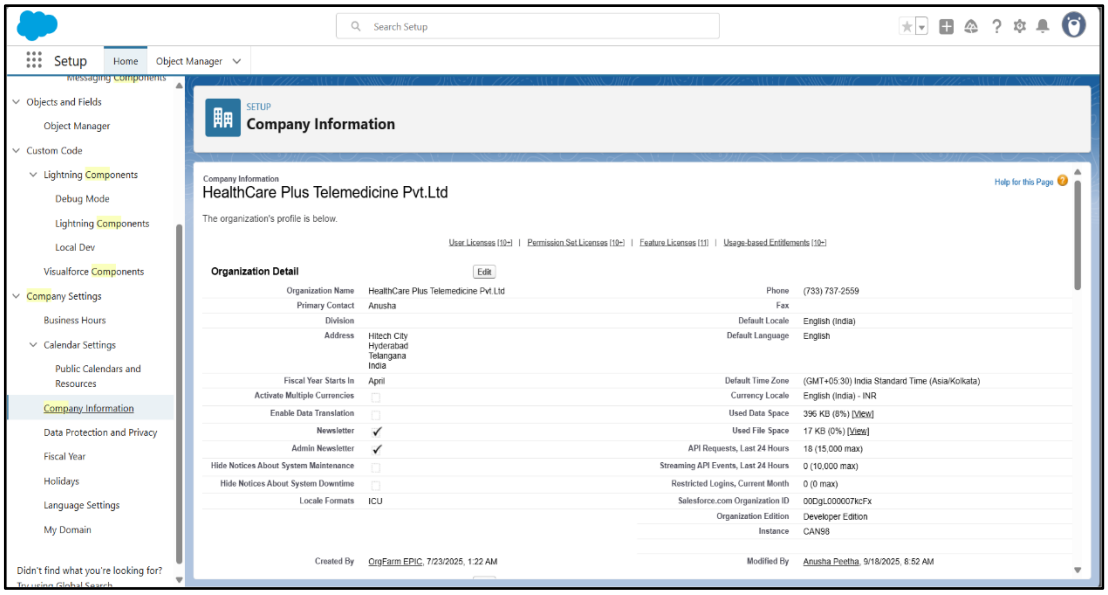
Salesforce Project Phase-2

Health Patient Management and Telemedicine CRM

Phase 2: Org setup & Configuration

- **Salesforce Editions:**
  - Login into the existing salesforce org.
  - Click on the Setup.
  - In the Quick Find box, type Company Information and click on it.
  - Look for the field Organization Edition like Developer Edition.

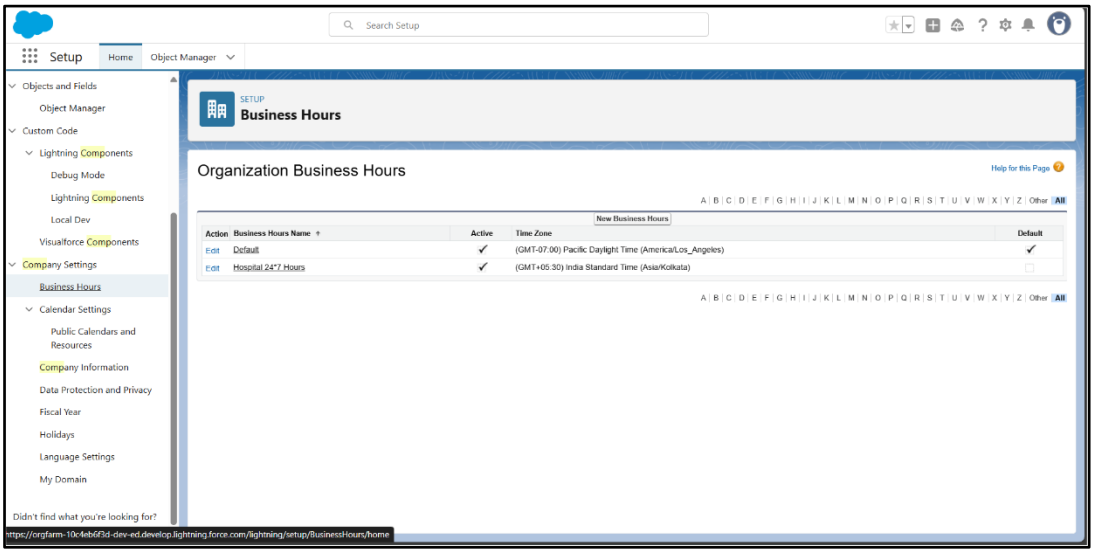
- **Company Profile Setup:**
  - Login to existing Salesforce org.
  - Click on the Setup.
  - In the Quick Find box, type Company Information and click on it.
  - On the Company Information page, click on edit.
  - Fill the details as below:
    1. Organization Name: HealthCare Plus Telemedicine Pvt. Ltd
    2. Primary Contact: Anusha
    3. Street: Hitech City
    4. City: Hyderabad
    5. State: Telangana
    6. Country: India
    7. Default Locale: English (India)
    8. Default Currency: INR
    9. Default Time Zone: (GMT+05:30) India Standard Time (Asia/Kolkata)
    10. Click Save.



❖ Business Hours & Holidays Setup:

1. Business Hours:

- Login to existing Salesforce org.
- Click on Setup.
- In the Quick Find box, type Business Hours and click on the Business Hours.
- Click New.
- Fill the details:
  1. Business Hours Name: Hospital 24*7 Hours
  2. Time Zone: (GMT+05:30) India Standard Time (Asia/Kolkata)
  3. Enable the Active.
  4. Set to all days that is Monday to Sunday.
  5. Click Save.



2. Holidays:

- Login to existing Salesforce org.
- Click on Setup.
- In the Quick Find box, type Holidays and click on the Holidays.
- Click New.
- Fill the details for 5 holidays:

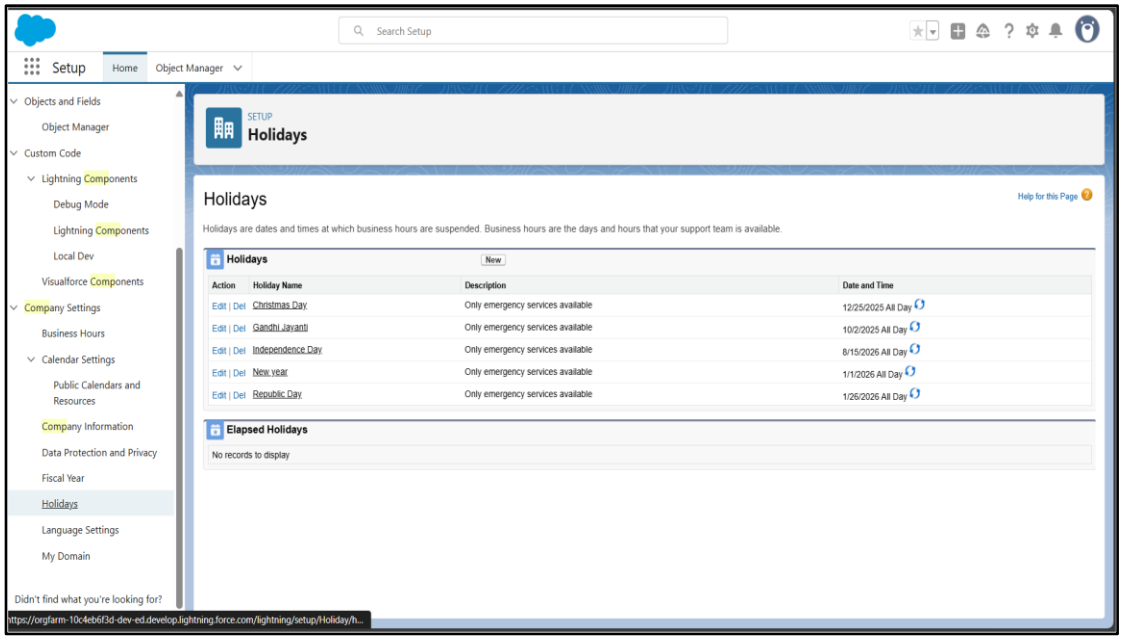
1. Holiday Name: New Year
   Description: only for emergency cases
   Date: 1/1/2025
   Enable the Recurrence
   Business Hours: All Days
   Click Save

2. Holiday Name: Republic Day
   Description: only for emergency cases
   Date: 1/26/2025
   Enable the Recurrence
   Business Hours: All Days
   Click Save

3. Holiday Name: Independence Day
   Description: only for emergency cases
   Date: 8/15/2025
   Enable the Recurrence
   Business Hours: All Days
   Click Save

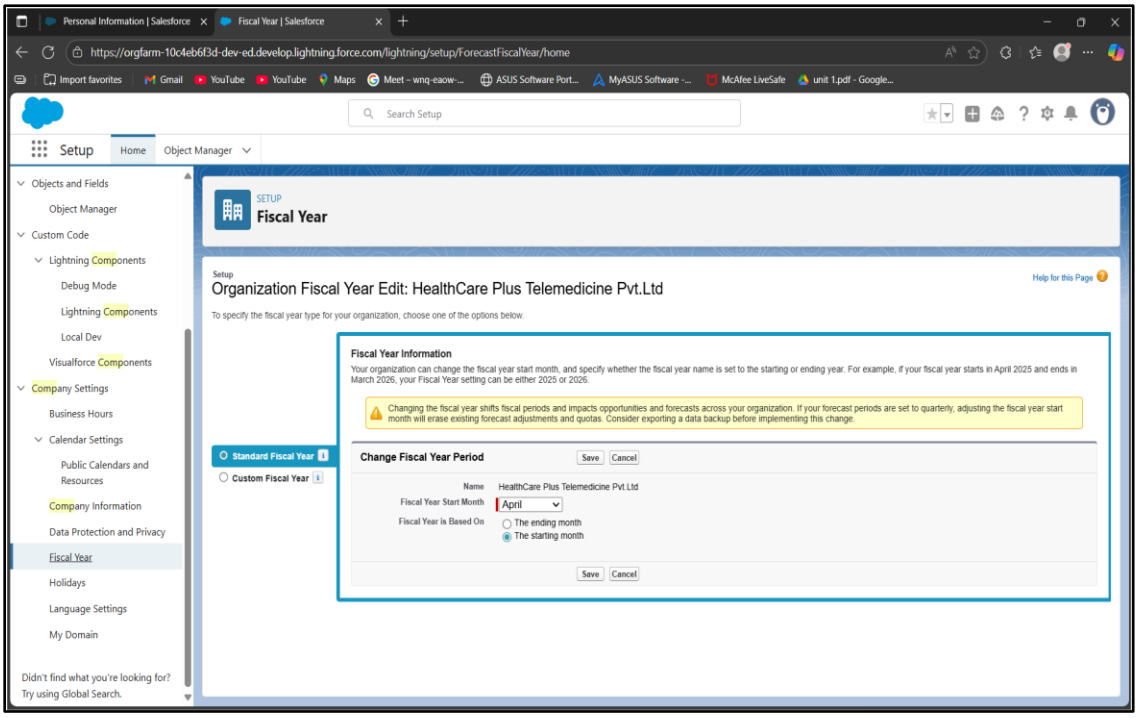
4. Holiday Name: Gandhi Jayanti
   Description: only for emergency cases
   Date: 10/2/2025
   Enable the Recurrence
   Business Hours: All Days
   Click Save

5. Holiday Name: Christmas Day
   Description: only for emergency cases
   Date: 12/25/2025
   Enable the Recurrence
   Business Hours: All Days
   Click Save



❖ Fiscal Year Settings:

- ➢ Login to existing salesforce org.
- ➢ Click on the Setup.
- ➢ In the Quick Find box, type Fiscal Year and click on it.
- ➢ Fiscal Year Start Month: April
- ➢ Select the starting month.
- ➢ Click Save.



❖ **User Setup & Licenses:**

1. **Check how many Licenses:**
   - ➢ In the Quick Find box, type Company Information and click it.
   - ➢ Look under the User Licenses.
2. **User Setup:**

Create Roles:

1. In the Quick Find box, type Roles and click it.
2. Click on Setup Roles.
3. Click on Add Role to create a new Role.
4. Fill the details:
   - Label: HealthCare Admin
     Role Name: HealthCare_ Admin
     Click on save
   - Label: Doctor/Specialist
     Role Name: Doctor_ Specialist
     Click on save
   - Label: Patient Coordinator
     Role Name: Patient_ Coordinator
     Click on Save

Create Users:

1. In the Quick Find box, type Users and click it.
2. Click on New user.
3. Fill the details:
   - First Name: Anusha
     Last Name: Peetha
     Alias: anu
     Email: anusha1901@gmail.com
     License: Salesforce
     Profile: System Admin
     Click on save and new
   - First Name: Dr. Ramesh
     Last Name: Shaik
     Alias: Ramesh
     Email: ramesh1902@gmail.com

License: Salesforce Integration
Profile: Standard User
Click on save and new

- First Name: Priya
  Last Name: Kamala
  Alias: fiya
  Email: priya1903@gmail.com
  License: Salesforce Integration
  Profile: Standard User
  Click on save

❖ **Profiles:**

- Go to Setup.
- In the Quick Find box, type the Profiles and then click on it.
- HealthCare Admin Profile:
  - Find the System Administrator and click on clone.
  - Profile Name: HealthCare Admin Profile
  - Click save.
- Doctor Profile:
  - Find the Standard User and click on clone.
  - Profile Name: Doctor Profile
  - Click save.
  - Click on edit.
  - In the Object Permissions give access to Patients, Appointments, Telemedicine (Read, Create, Edit).
  - Click save.
- Patient Coordinator Profile:
  - Find the Standard User and click on clone.
  - Profile Name: Patient Coordinator Profile
  - Click save.
  - Click on edit.
  - In the Object Permissions give access to Patients, Appointments, Bed Availability (Read, Create, Edit), Telemedicine Records (Read).
  - Click on save.

❖ **Roles:**

- In the Quick Find box, type Roles and click it.

- Click on Setup Roles.
- Click on Add Role to create a new Role.
- Fill the details:
    - Label: HealthCare Admin
      Role Name: HealthCare_ Admin
      Click on save

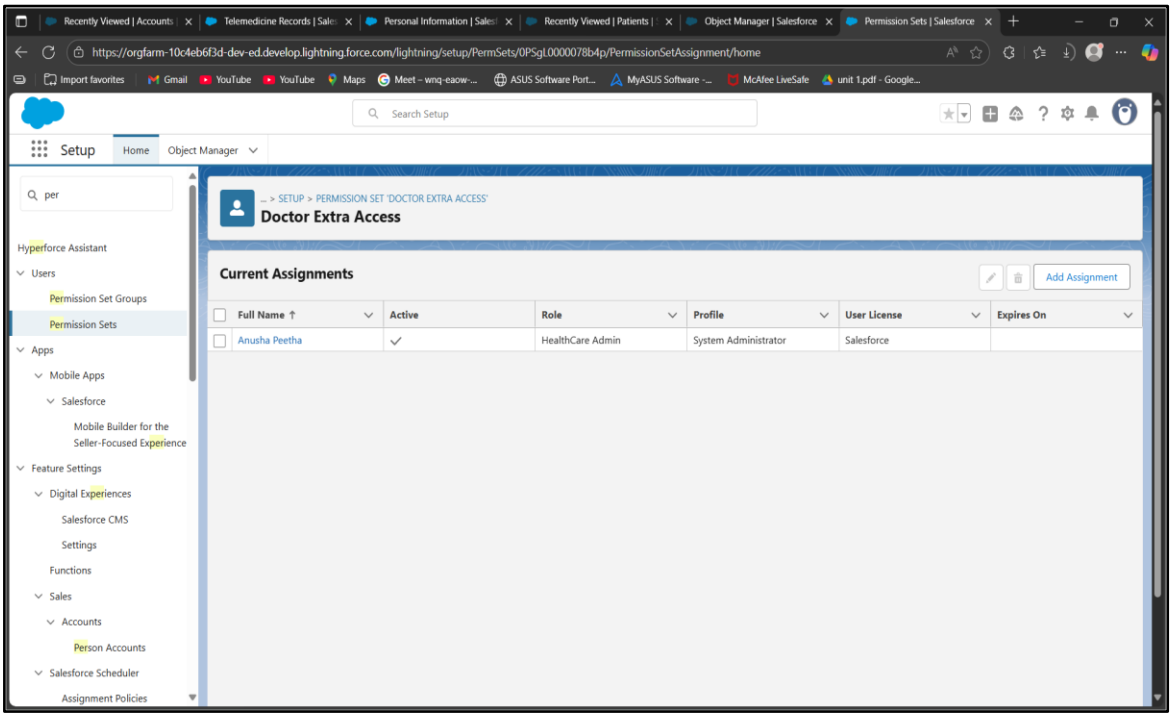
    - Label: Doctor/Specialist
      Role Name: Doctor_ Specialist
      Click on save

    - Label: Patient Coordinator
      Role Name: Patient_ Coordinator
- Click on Save

❖ **Permission Sets:**

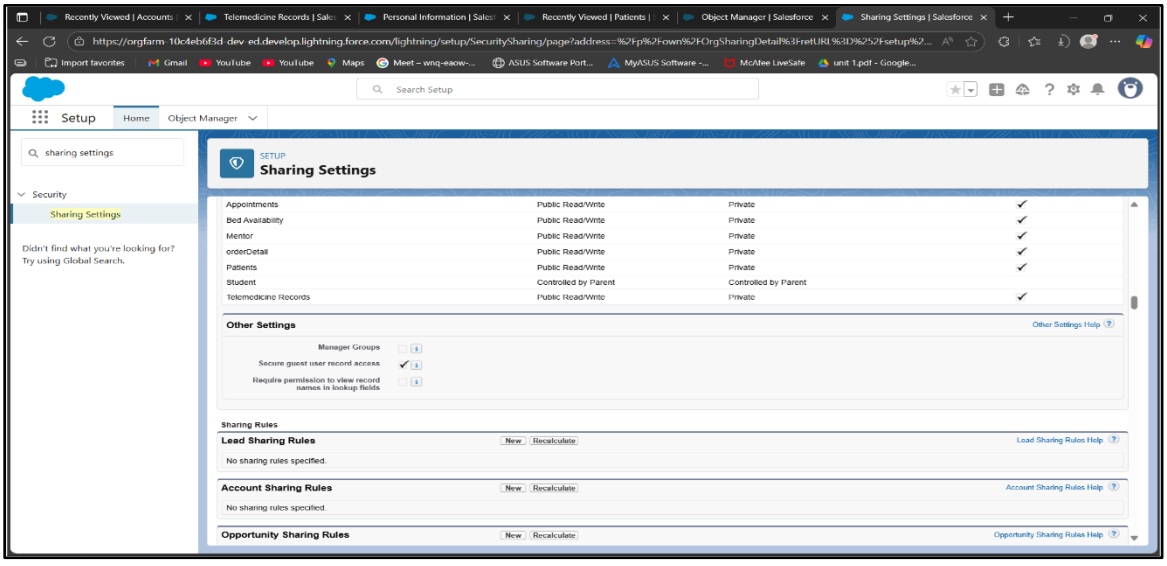
- Click on the Setup.
- In the Quick Find box, type the Permission sets and click on it.
- Click on New.
- Fill the details:
    - Label: Doctor Extra Access
    - API Name: Doctor_ Extra_ Access
    - User License: Salesforce
- Click on save.
- Now, open the Permission Set just created.
- Go to the Object Setting and select the object in the drop box Patients, Appointments, Telemedicine Records, Bed Availability.
- Click on edit.
- In object settings, enable the
    - Patients: Read, Create, Edit
    - Appointments: Read, Create, Edit
    - Telemedicine Records: Read, Create, Edit
    - Bed Availability: Read
- Click save.
- Now, click on the Manage Assignments and Add Assignments.

- Select Anusha Peetha and click Assign and then Done.



❖ **OWD:**

- In the Quick Find box, type sharing settings and click on it.
- Scroll down to the Organization Wide Section and click on edit.
- Change the default internal access for the following objects:
  Patients - Private, Appointments – Private, Bed Availability – Public Read only, Telemedicine Records – Private, Payments - Private
- Click save.



❖ **Sharing Rules:**

- Click on the Setup.
- In the Quick Find box, type the Sharing settings and click on it.
- Scroll down to the Sharing Rules.
- Patients:
  - Label: Doctors Access
  - Rule Name: Doctor_ Access
  - Select your rule Type: Based on Role
  - Select which records to be shared: Roles, Doctor
  - Select the users to share with: Roles, Doctor
  - Select the level of access for the users: Read/write
  - Click save.
- Appointments:
  - First you need create Appointment_ status_ c for that you need to go to object manager and then click on the appointments.
  - Now, then click on the Fields & Relationships.
  - Click on New and select Text and then next.
  - Field Label: Appointment_ status_ c, Length: 10, Field Name: Appointment_ status_ c, click next and again next and then Save.
  - Label: Coordinator Access
  - Rule Name: Coordinator_ Access
  - Select your rule Type: Based on criteria
  - Select which records to be shared: Appointment_ status_ c, equals, Scheduled
  - Select the users to share with: Roles, Patient Coordinator
  - Select the level of access for the users: Read only
  - Click Save.
- Bed Availability:
  - First you need create bed_ status_ c for that you need to go to object manager and then click on the Bed Availability.
  - Now, then click on the Fields & Relationships.
  - Click on New and select Text and then next.
  - Field Label: bed_ status_ c, Length: 10, Field Name: bed_ status_ c, click next and again next and then Save.
  - Label: Beds_ share_ staff
  - Rule Name: Beds_ share_ staff
  - Select your rule Type: Based on criteria
  - Select which records to be shared: Appointment_ status_ c, equals, Available

- Select the users to share with: Roles, Patient Coordinator
- Select the level of access for the users: Read only
- Click save.

➢ Telemedicine Records:

- First, you need to create tele_ status_ c for that you need to go to object manager and then click on the appointments.
- Now, then click on the Fields & Relationships.
- Click on New and select Text and then next.
- Field Label: tele_ status_ c, Length: 10, Field Name: tele_ status_ c, click next and again next and then Save.
- Second, you need to create public groups for that you need to go to public groups by typing in Quick Find box.
- Now enter the details: Label: Doctors& Coordinators, Group Name: Doctors_ Coordinator, Search: Roles, Available Members: Doctors and Patient Coordinator and then save.
- Label: tele_ coordinator
- Rule Name: tele_ coordinator
- Select your rule Type: Based on criteria
- Select which records to be shared: tele_ status_ c, equals, Scheduled
- Select the users to share with: public groups, Doctors& Coordinators
- Select the level of access for the users: Read/write
- Click save.

❖ **Login Access Policies:**

➢ Click on the Setup.
➢ In the Quick Find box, type Login Access Policies and click on it.
➢ Check the box “Administrations can log in as any user”.
➢ Click save.
➢ Test as HealthCare Project users:

- Go to Setup.
- In Quick Find box, type the Users and click it.
- Find the user you want to test: Doctor, Patient Coordinator, Patient, Bed availability

➢ Click login next to that user.
➢ You are now logged in as that user.
➢ Test their tabs, records, and permission.

❖ **Dev Org Setup:**

We created a Developer Org and set it up for our Healthcare Management project. We configured custom objects like Patients, Doctors, Appointments, Telemedicine Records,

and Bed Availability. Then we created a Healthcare Lightning App, assigned profiles and users, set Org-Wide Defaults, created sharing rules, and tested login access policies to ensure the right visibility and permissions.

- **Sandbox Usage:**
  - HealthCare Patient Management and Telemedicine CRM project, since we are working with a Salesforce Developer Org Sandboxes are not available.
  - Instead, the Developer Org itself acts as both Development and Testing environment.

- **Deployment Basics:**
  - Sandbox is not available, so all development and testing are done directly in the main org.
  - Custom objects like Patients, Doctors, Appointments, Telemedicine Records, Bed Availability are created and configured here.
  - Profiles, Roles, OWD, Sharing Rules, and Login Access Policies are applied to simulate real-world hospital access control.
  - Test Users are created to verify permissions, visibility, and workflow automation.

Salesforce Project Phase-3

Health Patient Management and Telemedicine CRM

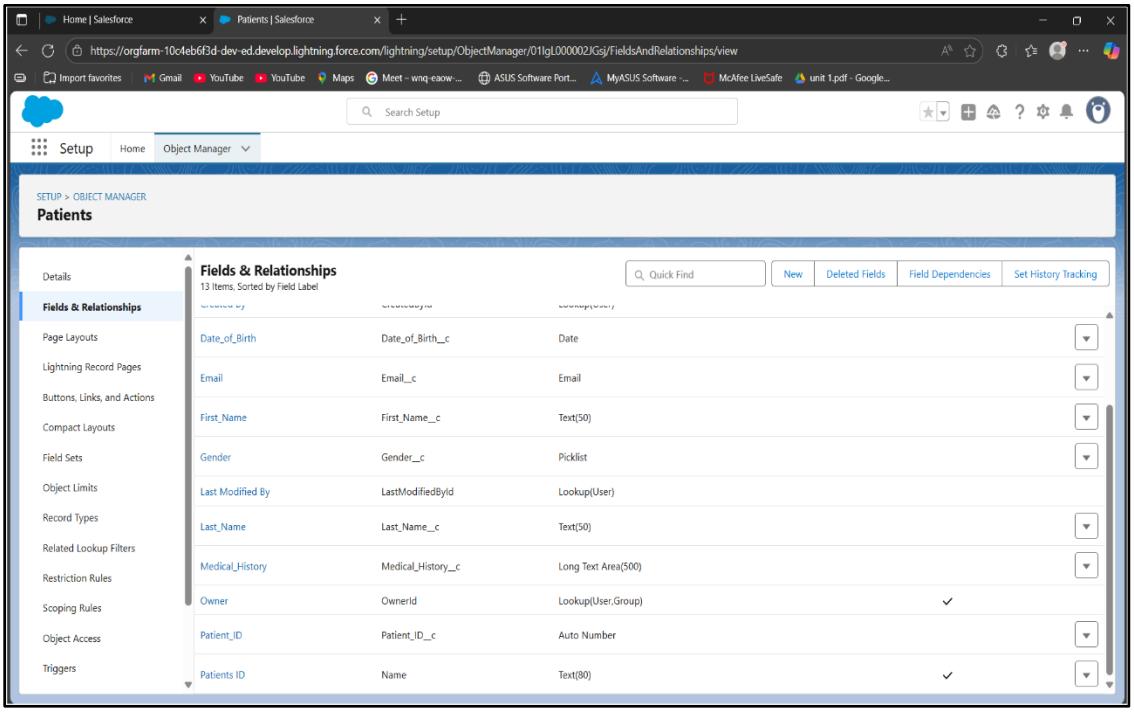
Phase 3: Data Modeling & Relationships

- **Standard and Custom Objects:**
  All the required standard and custom objects for the project have already been created in the previous phase.
  Standard Objects: User
  Custom Objects: Patients, Appointments, Telemedicine Records, Bed Availability, Payments

- **Fields:**
  - Go to Setup.
  - Click on the Object Manager, select the Patients.
  - Now click on the Fields & Relationships and click on New.
  - Patients:

Data Type	Field Name	Steps
Auto Number	Patient_ ID	Auto Number -> Next -> Starts with :1 -> Next-> Next-> Next -> Next -> Save
Text	First_ Name	Text -> Next -> Length: 50 -> Next-> Next -> Next -> Next -> Next -> Save
Text	Last_ Name	Text -> Next -> Length: 50 -> Next-> Next -> Next -> Next -> Next -> Save
Date	Date_ of_ Birth	Date -> Next -> Next-> Next -> Next -> Next -> Next -> Save
Picklist	Gender	Picklist -> Next ->Enter values (Male, Female,

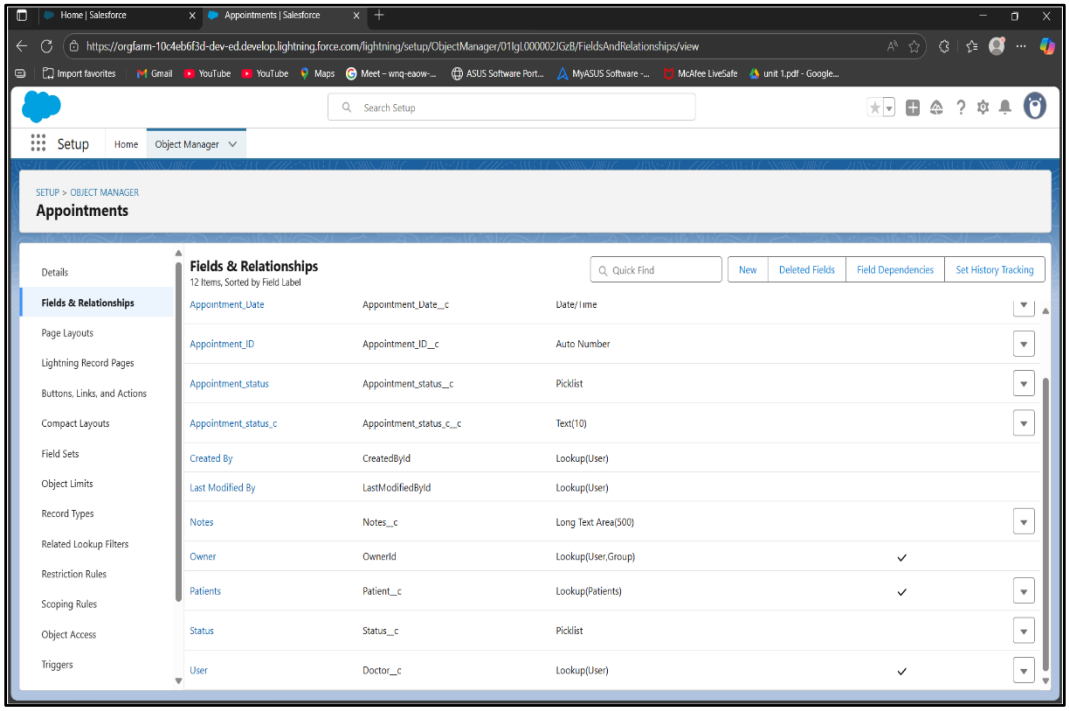
		Other)-> Next-> Next -> Next -> Next -> Next -> Save
Phone	Contact_ Number	Phone -> Next -> Next-> Next -> Next -> Next-> Save
Email	Email	Email -> Next -> Length: 255 -> Next-> Next -> Next -> Next -> Save
Text Area	Address	Text Area -> Next -> Length: 255 -> Next-> Next -> Next -> Next -> Save
Long Text Area	Medical_ History	Text -> Next -> Length: 500 -> Next-> Next -> Next -> Next -> Save



- Appointments Object:

Data Type	**Field Name**	**Steps**
Auto Number	Appointment_ ID	Auto Number -> Next -> Starts with :1 -> Next-> Next-> Next -> Next -> Save
Lookup	Patient	Lookup-> Next -> Related to Patients ->

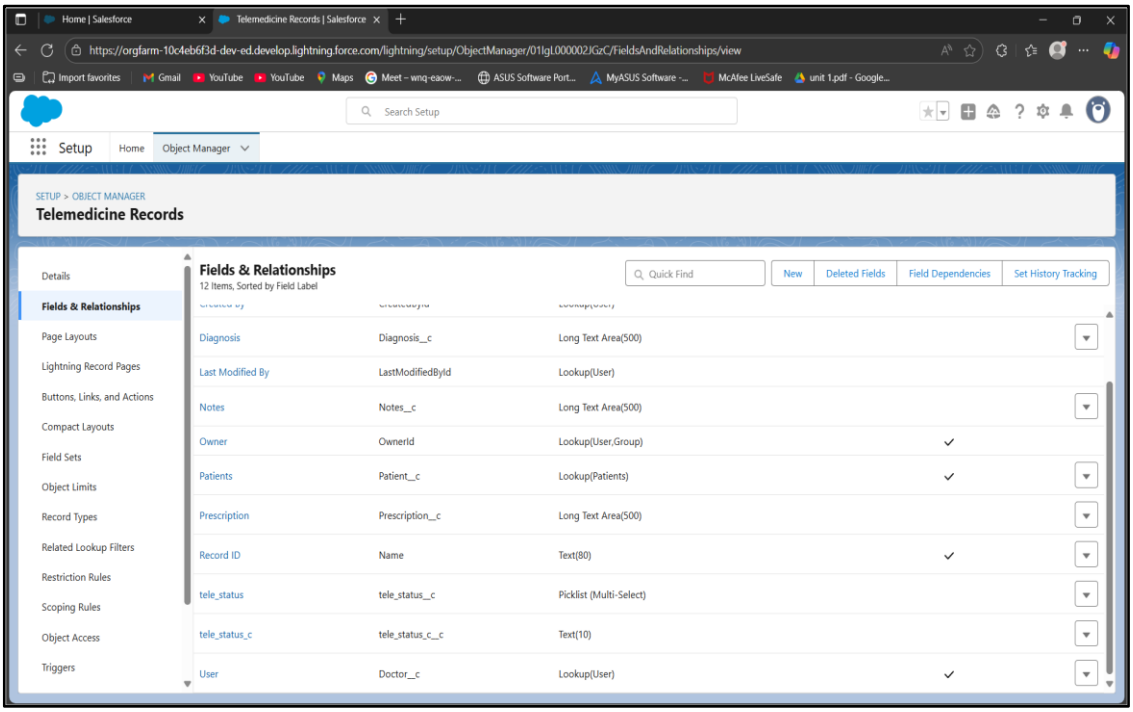
		Next-> Next -> Next -> Next -> Save
Lookup	Doctor	Lookup-> Next -> Related to User -> Next-> Next -> Next -> Next -> Save
Date/Time	Appointment_ Date	Date/Time -> Next -> Next-> Next -> Next -> Next -> Save
Picklist	Status	Picklist -> Next ->Enter values (Scheduled, Completed, Cancelled)-> Next-> Next -> Next -> Next -> Save
Long Text Area	Notes	Text Area (Long) -> Next -> Length: 500-> Next-> Next -> Next -> Next -> Save



- Telemedicine Records object:

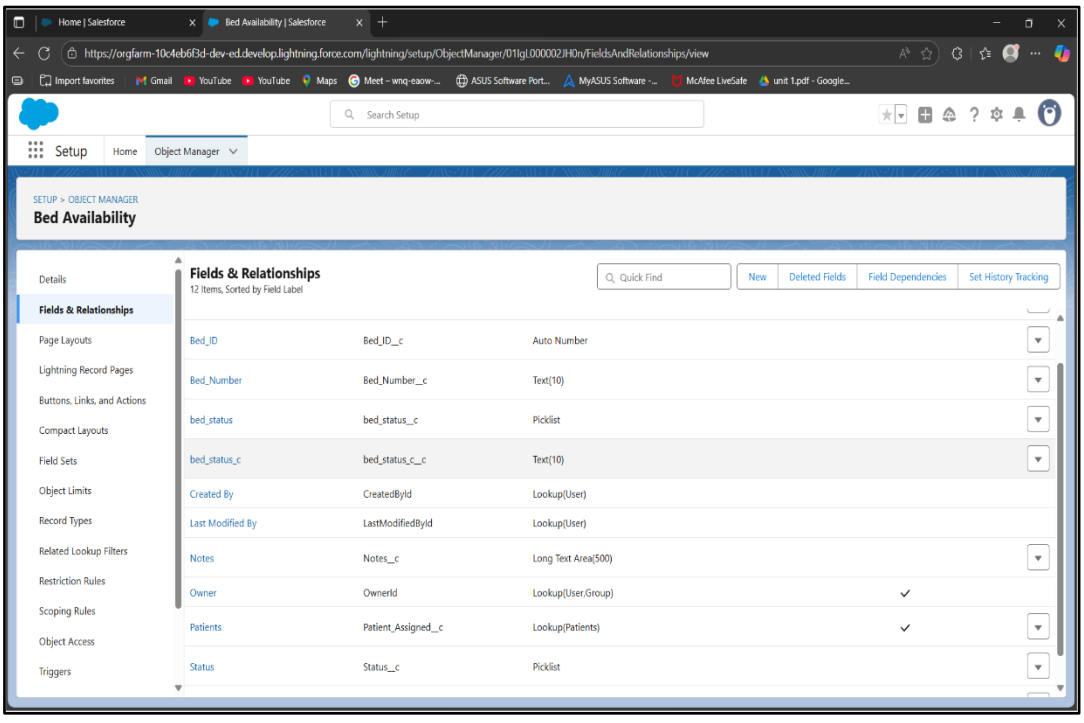
Data Type	Field Name	Steps
Auto Number	Record_ ID	Auto Number -> Next -> Starts with :1 -> Next->Next-> Next -> Next -> Save

Lookup	Patient	Lookup-> Next -> Related to Patients -> Next-> Next -> Next -> Next -> Save
Lookup	Doctor	Lookup-> Next -> Related to User -> Next-> Next -> Next -> Next -> Save
Date/Time	Consultation _Date	Date/Time -> Next -> Next-> Next -> Next -> Next -> Save
Long Text Area	Diagnosis	Text Area (Long) -> Next -> Length: 500 ->Next-> Next -> Next -> Next-> Save
Long Text Area	Prescription	Text Area (Long) -> Next -> Length: 500 ->Next-> Next -> Next -> Next-> Save
Long Text Area	Notes	Text Area (Long) -> Next -> Length: 500 ->Next-> Next -> Next -> Next-> Save



- Bed Availability object:

Data Type	Field Name	Steps
Auto Number	Bed_ ID	Auto Number -> Next -> Starts with :1 -> Next->Next-> Next -> Next -> Save
Text	Ward	Text -> Next -> Length: 50 ->Next-> Next -> Next -> Next-> Save
Text	Bed _Number	Text -> Next -> Length: 10 ->Next-> Next -> Next -> Next-> Save
Picklist	Status	Picklist -> Next ->Enter values (Available, occupied, Maintenance)-> Next-> Next -> Next -> Next -> Save
Lookup	Patient _Assigned	Lookup-> Next -> Related to Patients -> Next-> Next -> Next -> Next -> Save
Long Text Area	Notes	Text Area (Long) -> Next -> Length: 500 ->Next-> Next -> Next -> Next-> Save



- Payments:

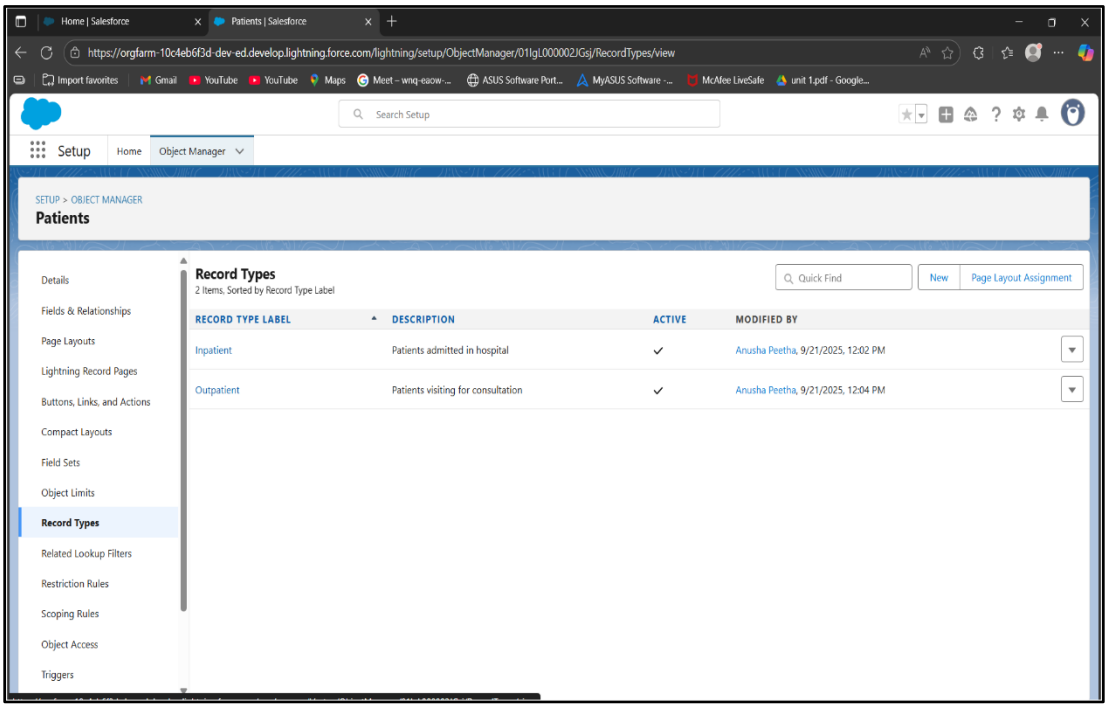
Data Type	Field Name	Steps
Lookup	Patient	Lookup -> Next -> Related to Patients -> Next->Next-> Next -> Next -> Save
Lookup	Appointment	Lookup -> Next -> Related to Appointments -> Next->Next-> Next -> Next -> Save
Currency	Amount	Text -> Next -> Length: 16, Decimal: 2->Next-> Next -> Next -> Next-> Save
Date	Payment Date	Next-> Next -> Next -> Save
Picklist	Payment Method	Picklist -> Next ->Enter values (Cash, Card, UPI, Insurance)-> Next-> Next -> Next -> Next -> Save
Text Area	Notes	Text Area -> Next ->Next-> Next -> Next -> Next-> Save
Picklist	Status	Picklist -> Next ->Enter values (Pending, Completed, Failed)-> Next-> Next -> Next -> Next -> Save

❖ **Record Types:**

- Go to Setup.
- Click On Object Manager and click on required object.
- Click on Record Type and then click New.
- Patients:
  1. Inpatient:
     - Record Type: Master
     - Record Type Label: Inpatient
     - Record Type Name: Inpatient
     - Description: Patients admitted in hospital

- Active: enable
- Select, Apply a different layout for each profile
- Click on save.

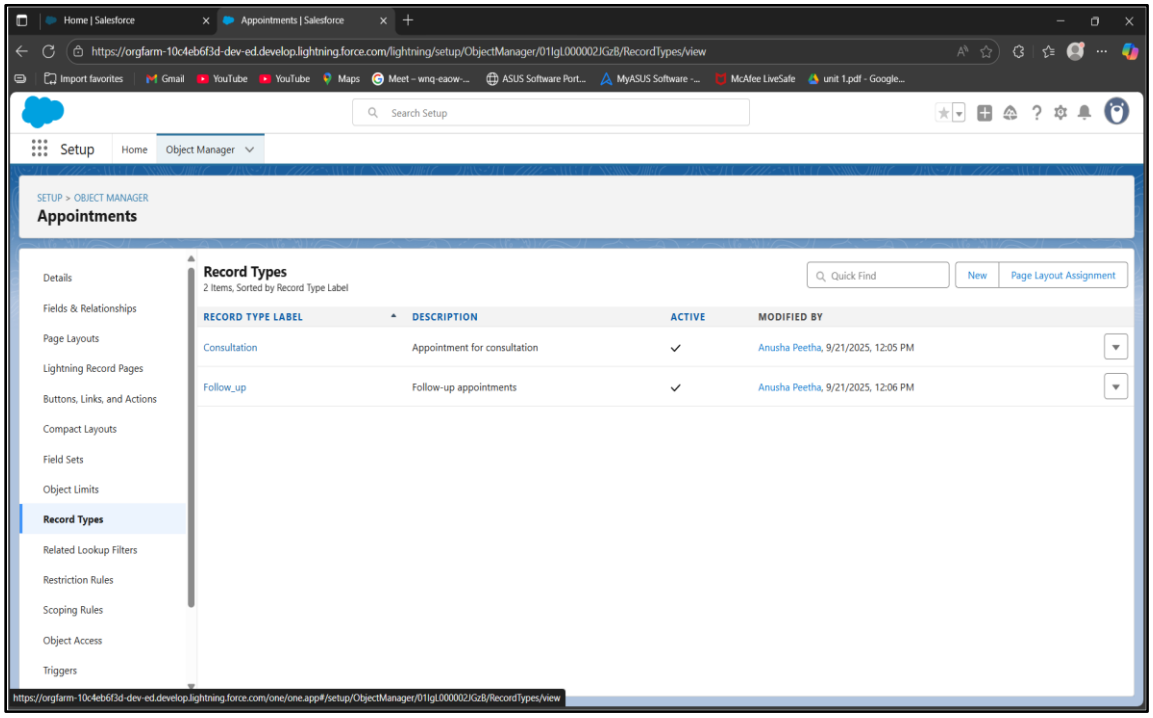
2. Outpatient:
    - Record Type: Master
    - Record Type Label: outpatient
    - Record Type Name: outpatient
    - Description: Patients visiting for consultation
    - Active: enable
    - Select, Apply a different layout for each profile
    - Click on save.



➢ Appointments:

1. Consultation:
    - Record Type: Master
    - Record Type Label: Consultation
    - Record Type Name: Consultation
    - Description: Appointment for consultation
    - Active: enable
    - Select, Apply a different layout for each profile
    - Click on save.
2. Follow-up:

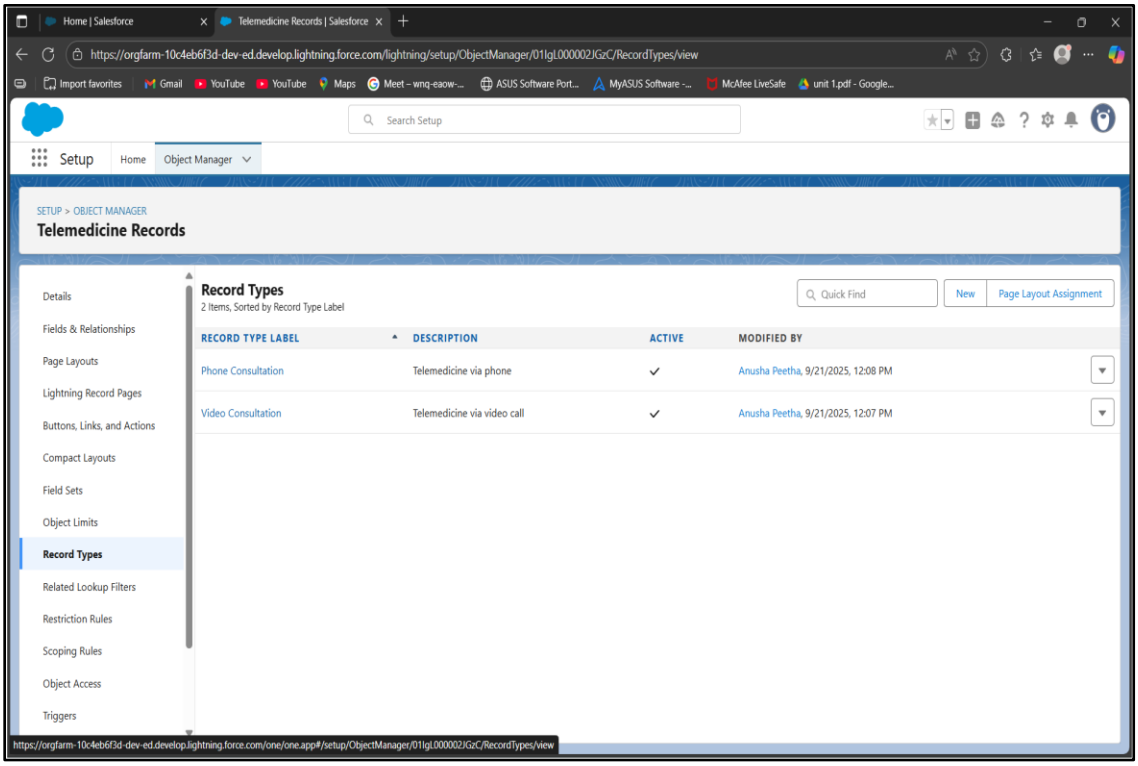
- Record Type: Master
- Record Type Label: Follow-up
- Record Type Name: Follow-up
- Description: Follow-up appointments
- Active: enable
- Select, Apply a different layout for each profile
- Click on save.



➢ Telemedicine Records:

1. Video consultation:
    - Record Type: Master
    - Record Type Label: Video Consultation
    - Record Type Name: Video _Consultation
    - Description: Telemedicine via video call
    - Active: enable
    - Select, Apply a different layout for each profile
    - Click on save.
2. Phone Consultation:
    - Record Type: Master
    - Record Type Label: Phone Consultation
    - Record Type Name: Phone_ Consultation

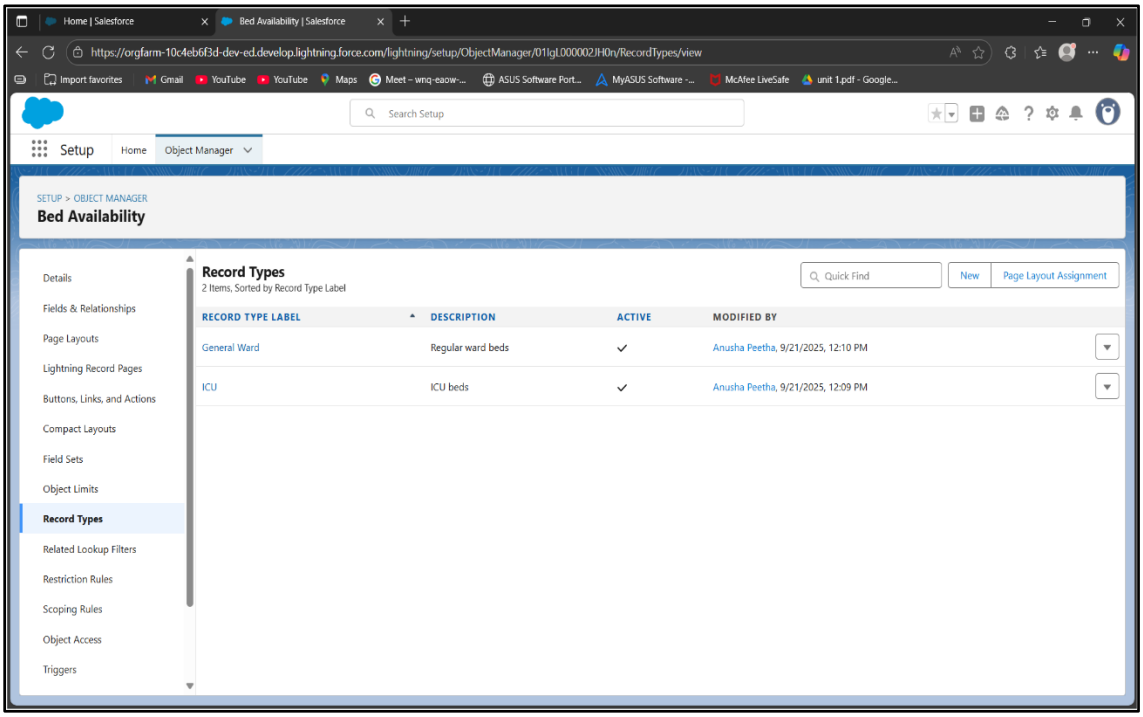
- Description: Telemedicine via phone call
- Active: enable
- Select, Apply a different layout for each profile
- Click on save.



➢ Bed Availability object:

1. ICU:
    - Record Type: Master
    - Record Type Label: ICU
    - Record Type Name: ICU
    - Description: ICU beds
    - Description: Telemedicine via video call
    - Active: enable
    - Select, Apply a different layout for each profile
    - Click on save.
2. General Ward:
    - Record Type: Master
    - Record Type Label: General Ward

- Record Type Name: General _Ward
- Description: Regular ward beds
- Active: enable
- Select, Apply a different layout for each profile
- Click on save



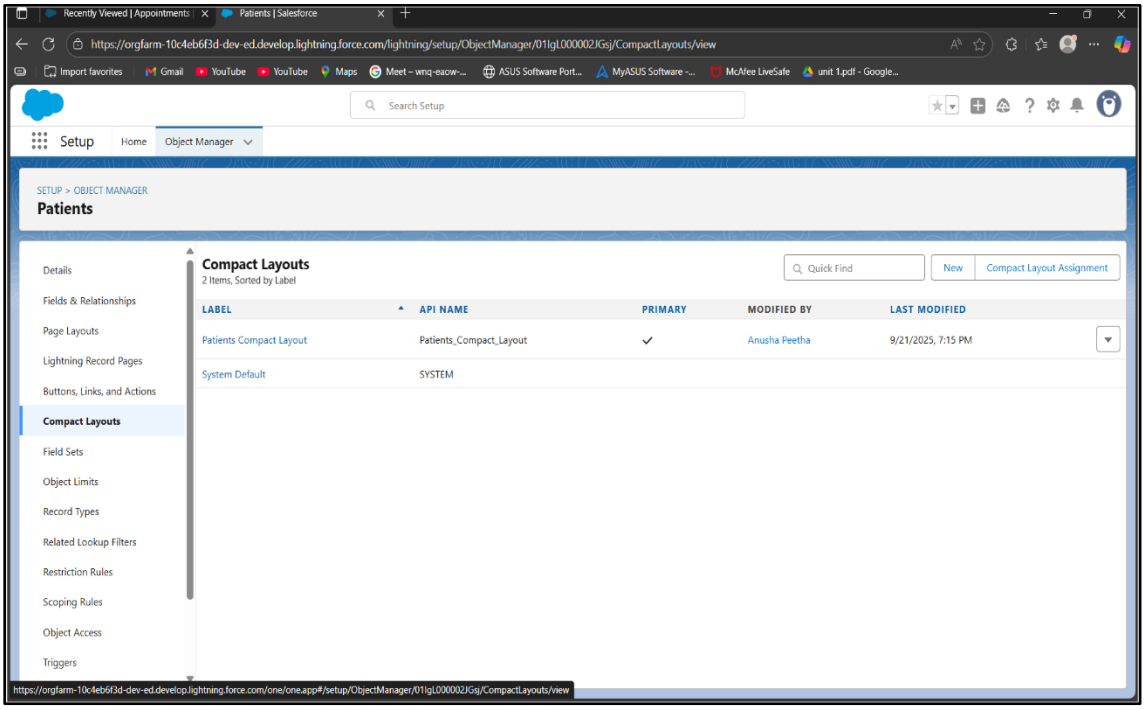
❖ **Page Layouts:**

- ➢ Go to Setup.
- ➢ Click on the Object Manager.
- ➢ Find the required object Patients, Appointments, Bed availability, Telemedicine Records, Payments and click on it.
- ➢ Find the Page Layouts in the left panel and click on it.
- ➢ Now click on edit on the default Page Layout.
- ➢ See that all the Fields are in the Information Section.
- ➢ If not in the Information Section drag the fields into the Information Section.

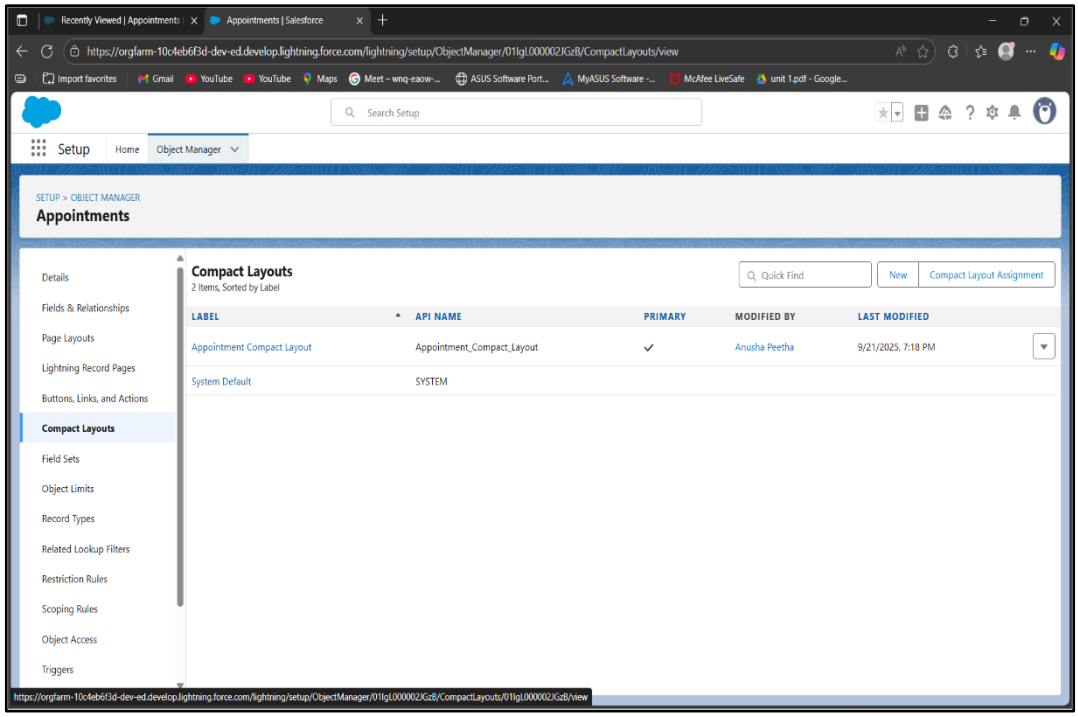
❖ **Compact Layouts:**

- ➢ Go to Setup.
- ➢ Click on the Object Manager.

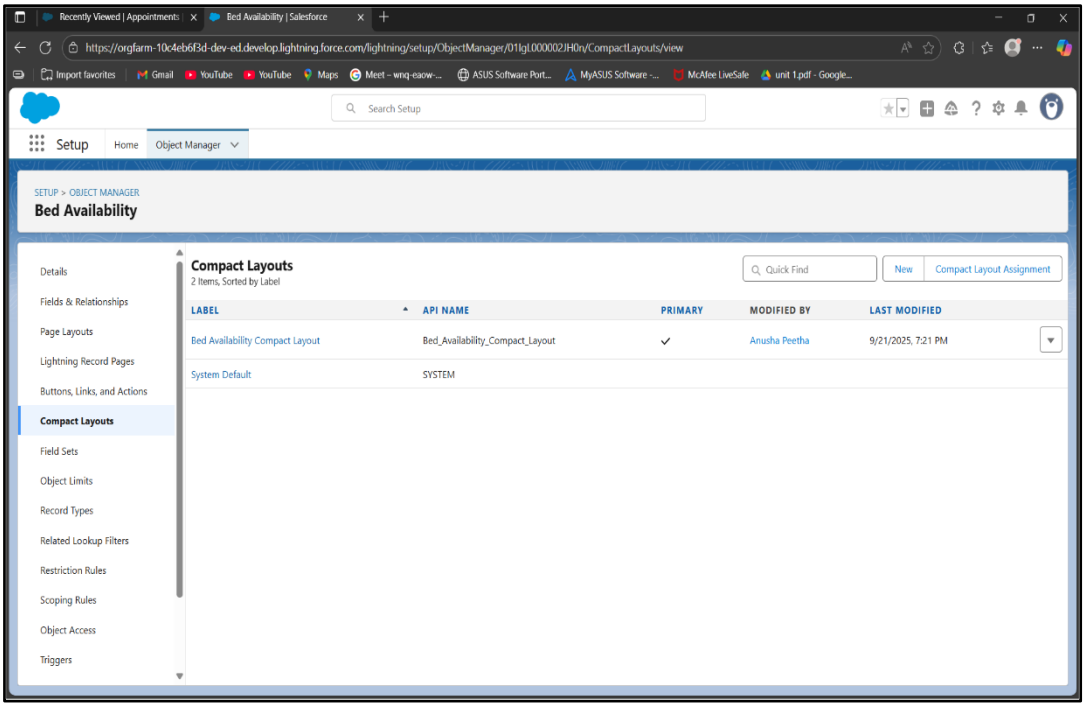
- Find the required object Patients, Appointments, Bed availability, Telemedicine Records and click on it.
- Find the Compact Layouts in the left panel and click on it.
- Now click on New.
- Patients:
    - Label: Patients Compact Layout
    - Name: Patients _Compact _Layout
    - Fields to display: Patient _ID, Record Type, first _Name, Last _Name, Gender, Date _ of_ Birth, Contact _Number, Email
    - Click on Save.
    - Now click on Compact Layout Assignment and click on edit.
    - In the two drop down, select the Patient Compact Layout.
    - Click on save.



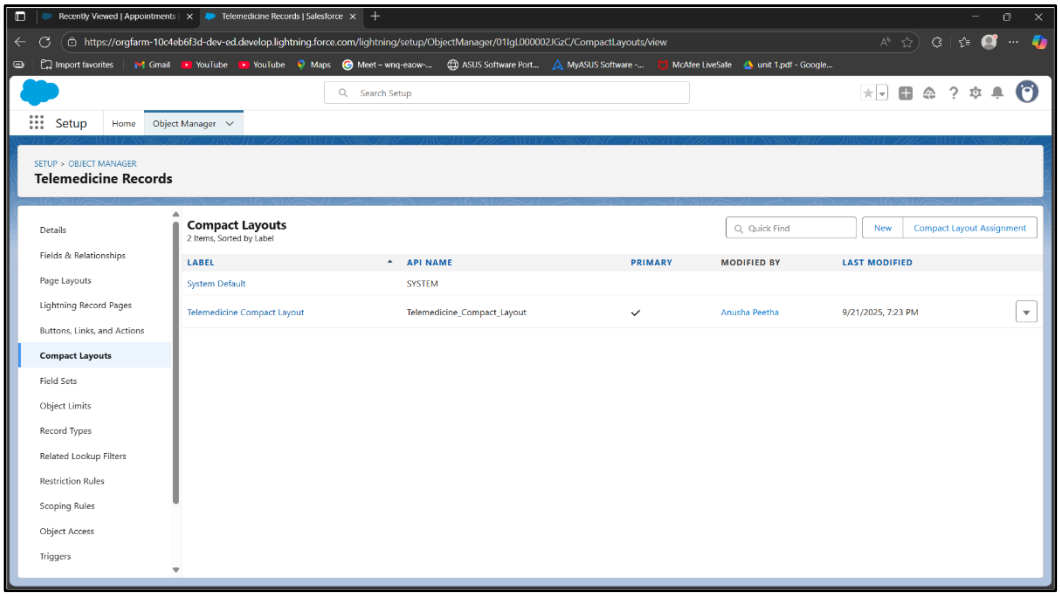
- Appointments:
    - Label: Appointment Compact Layout
    - Name: Appointment _Compact _Layout
    - Fields to display: Appointment _ID, Record Type, Appointment _Date, Appointment _status _c, User, status
    - Click on Save.
    - Now click on Compact Layout Assignment and click on edit.
    - In the two drop down, select the Appointment Compact Layout.
    - Click on save.



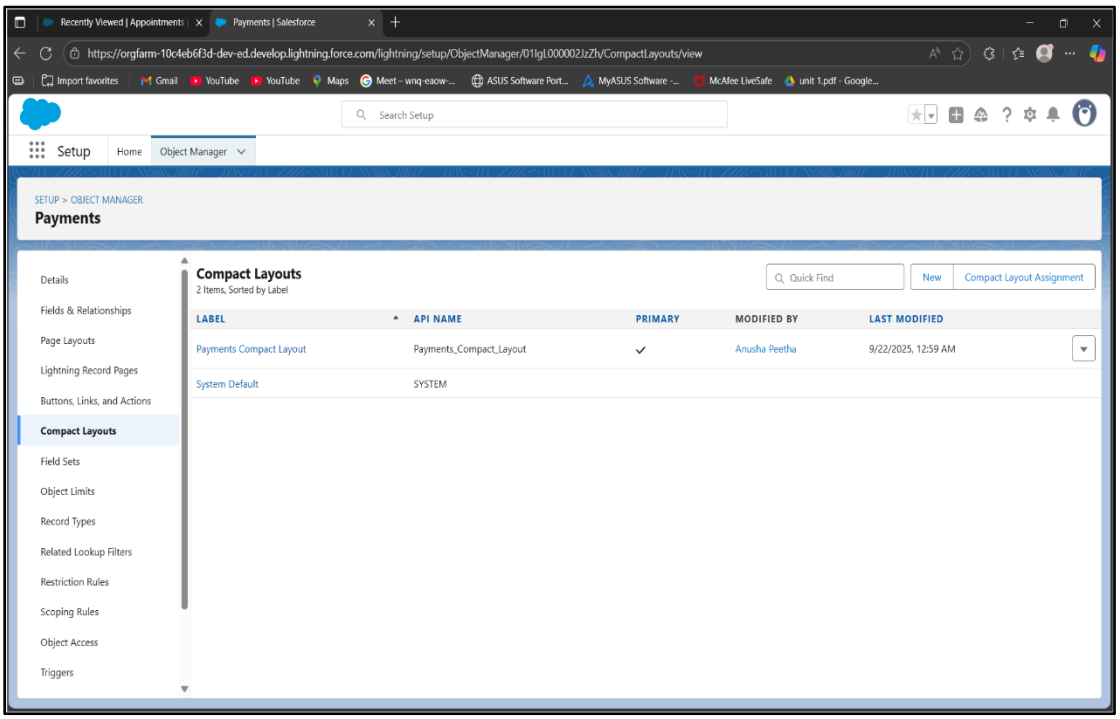
- Bed Availability:
  - Label: Bed Availability Compact Layout
  - Name: Bed _Availability _Compact _Layout
  - Fields to display: Bed _ID, Record Type, Bed _Number, bed _status _c, Patients, status, ward
  - Click on Save.
  - Now click on Compact Layout Assignment and click on edit.
  - In the two drop down, select the Bed Availability Compact Layout.
  - Click on save.



- Telemedicine Records:
  - Label: Telemedicine Compact Layout
  - Name: Telemedicine _Compact _Layout
  - Fields to display: Record ID, Record Type, Consultation _Date, tele _status _c, User
  - Click on Save.
  - Now click on Compact Layout Assignment and click on edit.
  - In the two drop down, select the Telemedicine Compact Layout.
  - Click on save.

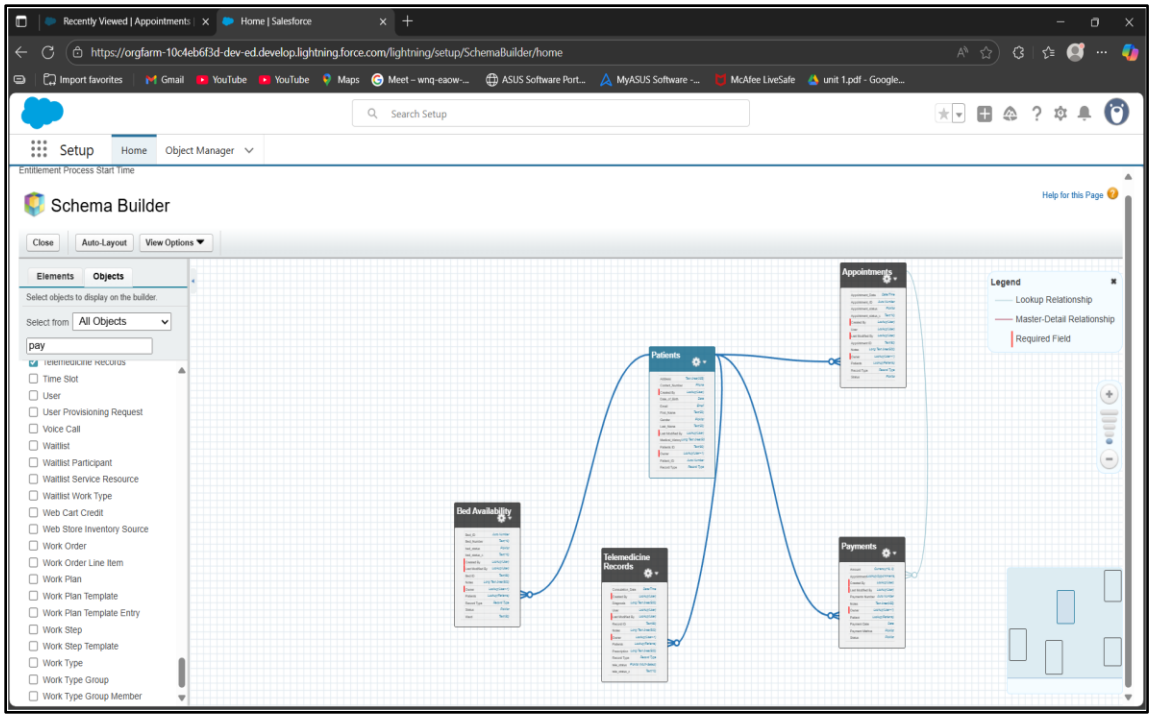


- Payments:
    - Label: Payment Compact Layout
    - Name: Payment _Compact _Layout
    - Fields to display: Payment Number, Patient, Amount, Status
    - Click on Save.
    - Now click on Compact Layout Assignment and click on edit.
    - In the two drop down, select the Payment Compact Layout.
    - Click on save.



❖ **Schema Builder:**

- Go to Setup.
- In the Quick Find box, type the Schema Builder and click on it.
- In the left panel, enable the Objects:
  Patients, Doctors, Appointments, Telemedicine Records, Bed Availability and Payments.
- Click on the Auto – layout.



❖ **Lookup vs Master-Detail vs Hierarchical Relationships:**

➢ Lookup Relationships:

- Patients → Appointments
  Each patient can have multiple appointments. An appointment record looks up to the patient.

- Doctors → Appointments
  Each appointment looks up to a doctor, but if the doctor record is deleted, the appointment can still exist.

- Patients → Telemedicine Reports
  Telemedicine reports are linked to patients using a lookup relationship.

➢ Master – Detail Relationships:

- Appointments → Payments
  A payment record depends on the appointment. If the appointment is deleted, the related payment record is also deleted.

➢ Hierarchical Relationship:

- User Object
  A senior doctor can supervise junior doctors or staff members. This relationship type is only available on the User object and helps in approvals and hospital reporting structures.

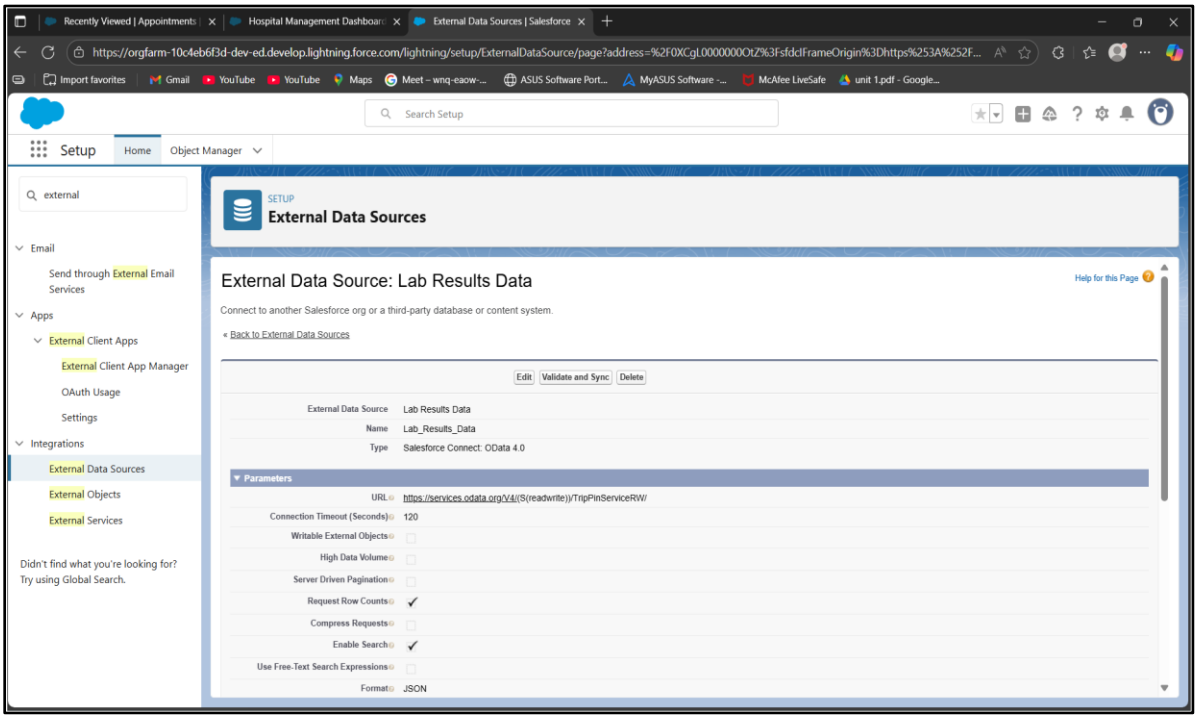
❖ **Junction Objects:**

- Go to Setup.
- Click on the Object Manager and click on create.
- Select the Custom Object.
- Fill the details:
  - Label: Patient Doctor Assignment
  - Object Name: Patient _Doctor _Assignment _c
  - Enable Allow Reports, Allow Activities and track reports
- Click on Save.
- Now in the Patient Doctor Assignment Object go to Fields & Relationships and click on New.
- Select the master – detail relationship and click Next
- Fill the details:
  - Field Label: Patient
  - Field Name: Patient _c
- Click on save.
- Again, go to Fields & Relationships and click on New
- Select the master – detail relationship and click Next
- Fill the details:
  - Field Label: Doctor
  - Field Name: Doctor _c
- Click on Save
- Click on the Assign Page Layout
- In the drop box, select the Patient Doctor Assignment Layout.
- Click save.
- Click on the App Launcher, select Reports
- Click on New Report
- Select the Patient Doctor Assignment
- Click start Report
- Drag the Patient and Doctor to the report columns
- Click on Save & Run
- Name the report: Patient-Doctor Assignments
- Create and choose the HealthCare report and then save
- Click on the App Launcher, select the Dashboards
- Click on New Dashboard
- Enter Name: Hospital Management Dashboard and then create
- Click on +widget and choose table
- Select the Patient-Doctor Assignments report
- Click on save and then Done.

❖ **External Objects:**

- Go to Setup.
- In the Quick Find box, type the External Data Sources click on it.
- Click on the New External Data Sources.
- Fill the details:
  - Label: Lab Results Data

- Name: Lab _Results _Data
- Type: OData 4.0
- URL: https://services.odata.org/V4/(S(readwrite))/TripPinServiceRW/
- Identity Type: Named Principal
- Authentication Protocol: No Authentication
- Click on Save.
- Click on Validate and Sync
- Select people and click on sync.

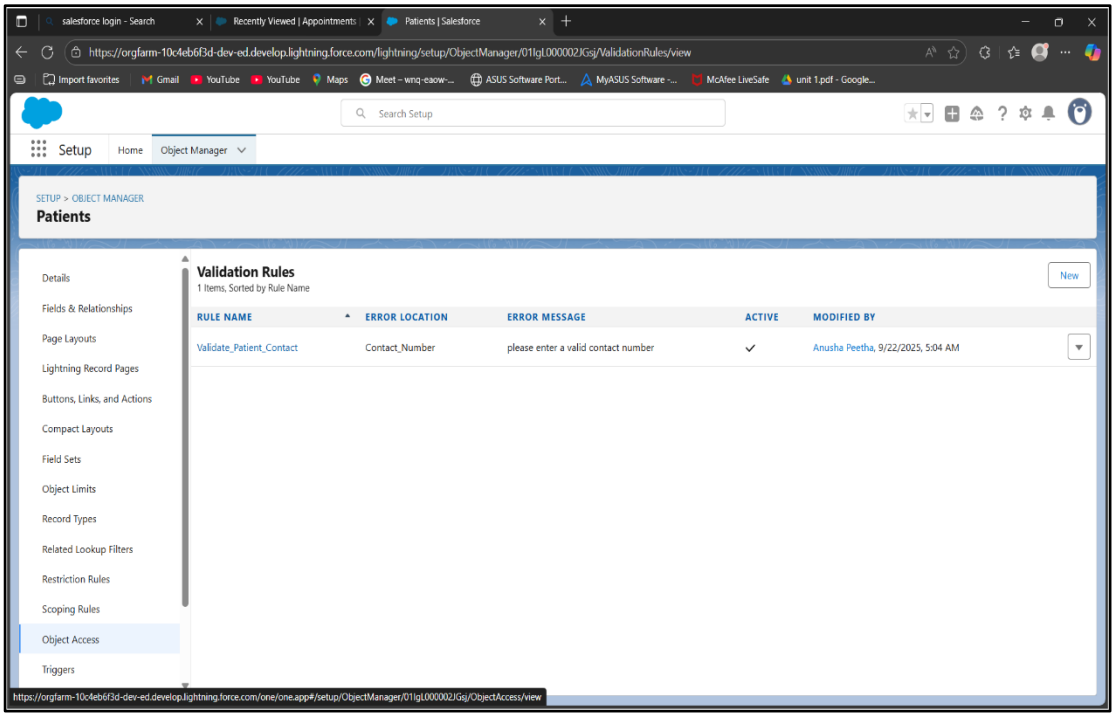


Salesforce Project Phase-4

Health Patient Management and Telemedicine CRM

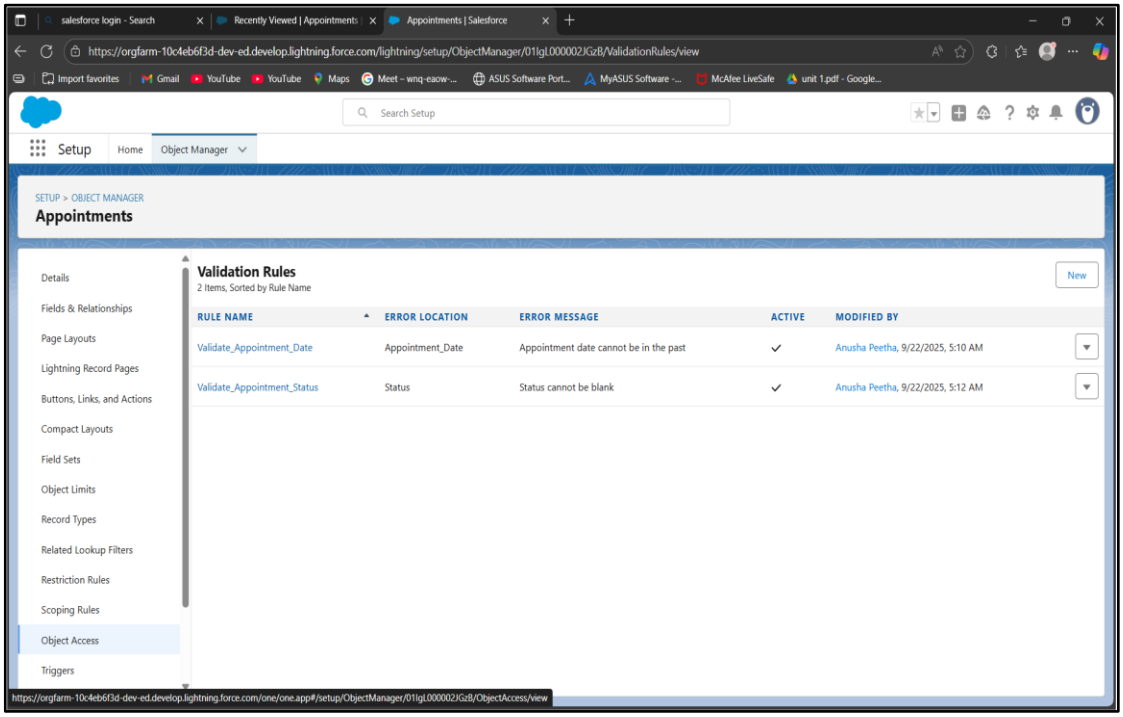
Phase 4: Process Automation (Admin)

- **Validation Sets:**
  - Go to Setup.
  - Click on the Object Manager.
  - Find the required objects: Patients, Appointments, Telemedicine Reports, Bed Availability
  - Click on Validation Sets and click on New.
  - Patients:
    - Rule Name: Validate _Patient _Contact
    - Error Condition Formula: contact _ c = ‘10’ and click check syntax
    - Error Message: Please enter a valid contact number
    - Error Location: Field and then Contact _Number

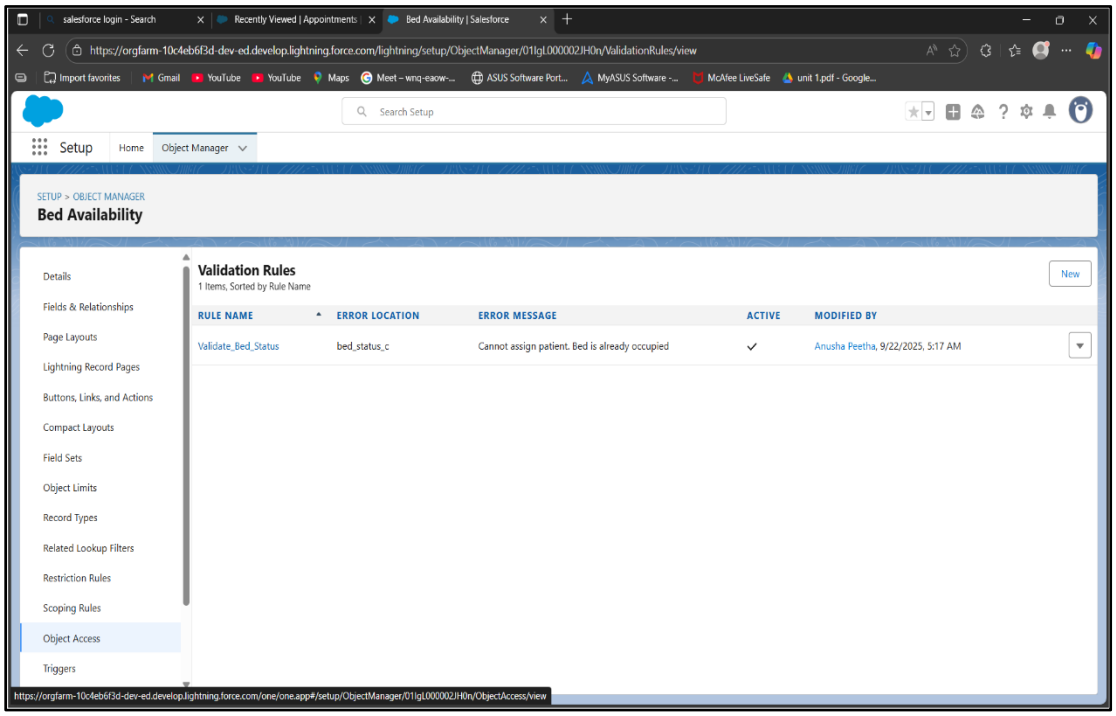


- Appointments:
  1. Appointment _Date:
     - Rule Name: Validate _Appointment _Date
     - Error Condition Formula: Appointment _Date __c < NOW () and click check syntax
     - Error Message: Appointment date cannot be in the past
     - Error Location: Field and then Appointment Date

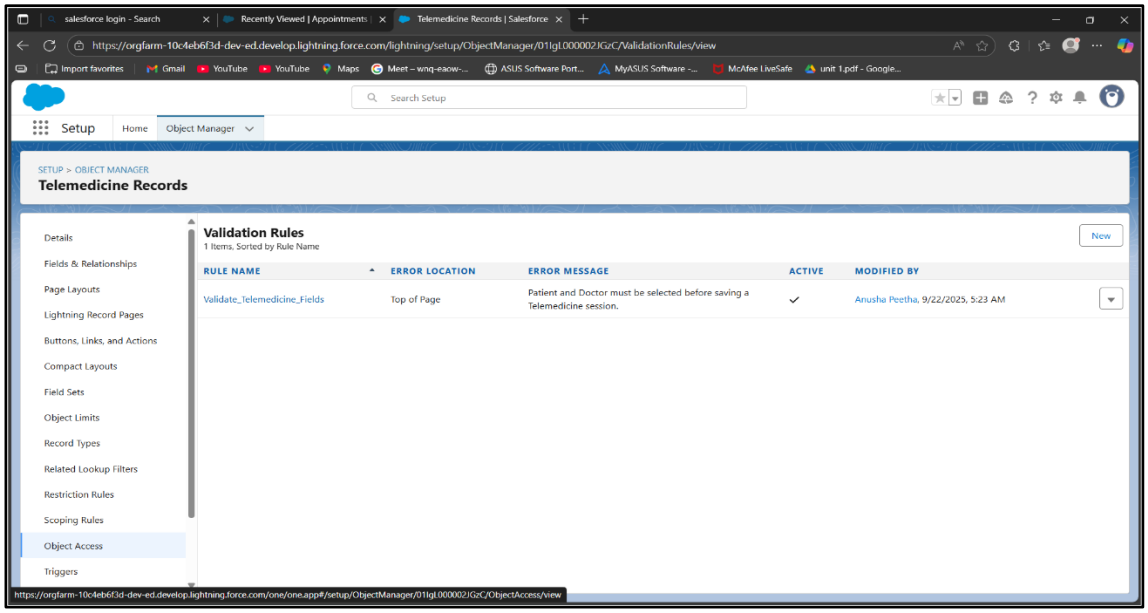
  2. Status:
     - Rule Name: Validate _Appointment _Status
     - Error Condition Formula: ISPICKVAL (Status __c, "Cancelled") and click check syntax
     - Error Message: Status cannot be blank
     - Error Location: Field and then Status



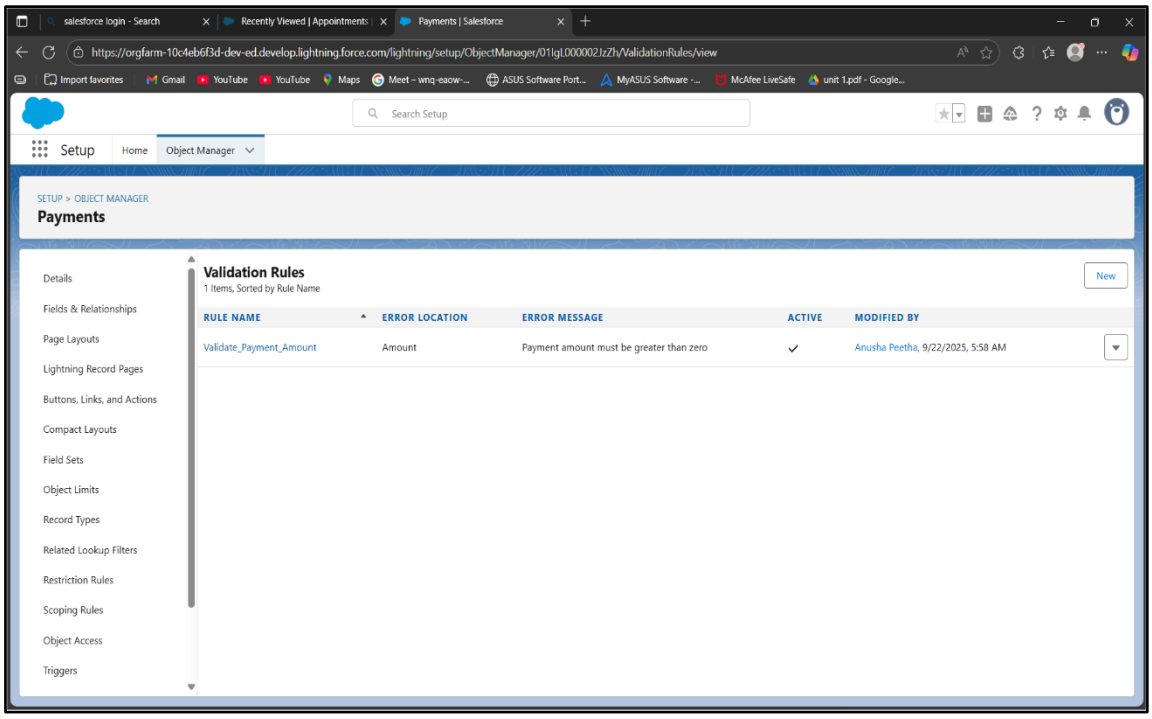
- Bed Availability:
  - Rule Name: Validate _Bed _Status
  - Error Condition Formula: ISPICKVAL (Bed _Status __c, "Occupied") and click check syntax
  - Error Message: Cannot assign patient. Bed is already occupied.
  - Error Location: Field and then bed _status



- Telemedicine Records:
  - Rule Name: Validate _Telemedicine _Fields
  - Error Condition Formula: ISBLANK (Patient __c) || ISBLANK (Doctor _c) and click check syntax
  - Error Message: Patient and Doctor must be selected before saving a Telemedicine session.
  - Error Location: Top of page

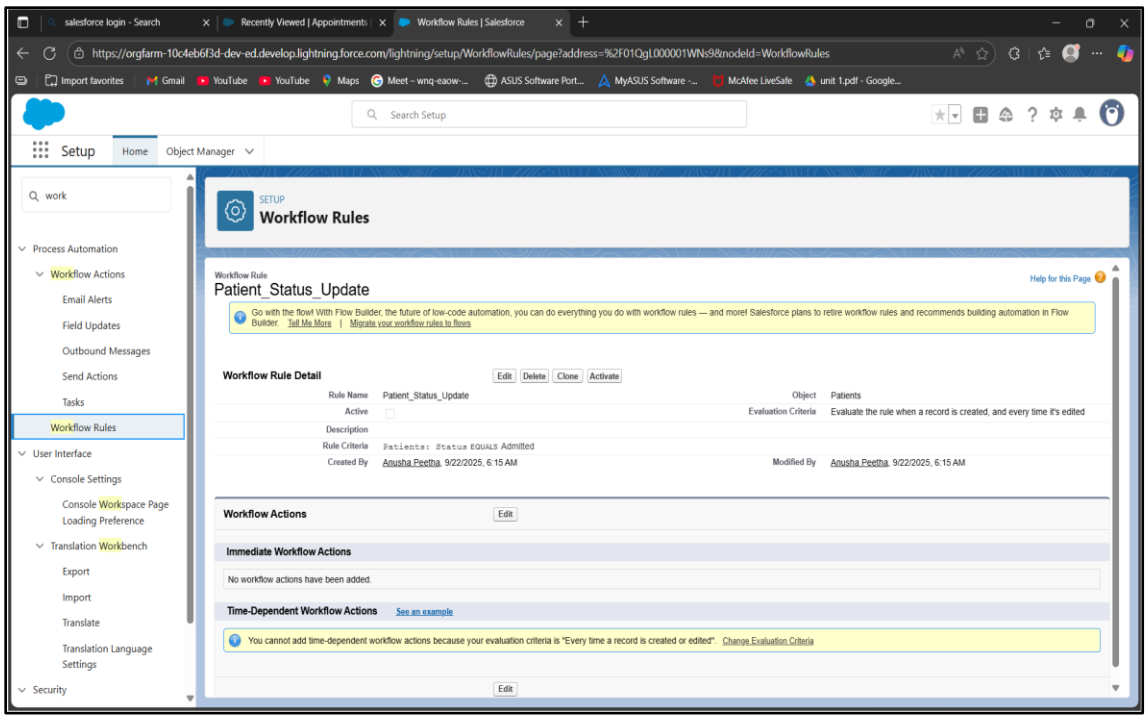


- Payments:
    - Rule Name: Validate _Payment _Amount
    - Error Condition Formula: Amount _c <= 0 and click check syntax
    - Error Message: Payment amount must be greater than zero
    - Error Location: Top of page

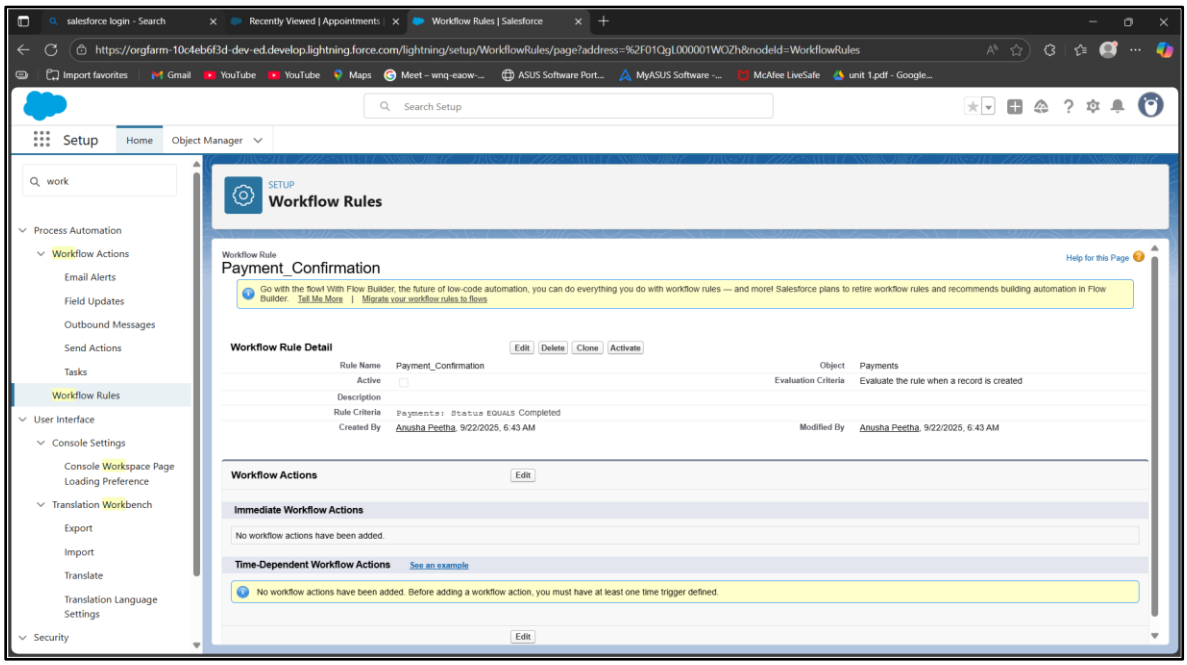


❖ **Workflow Rules:**

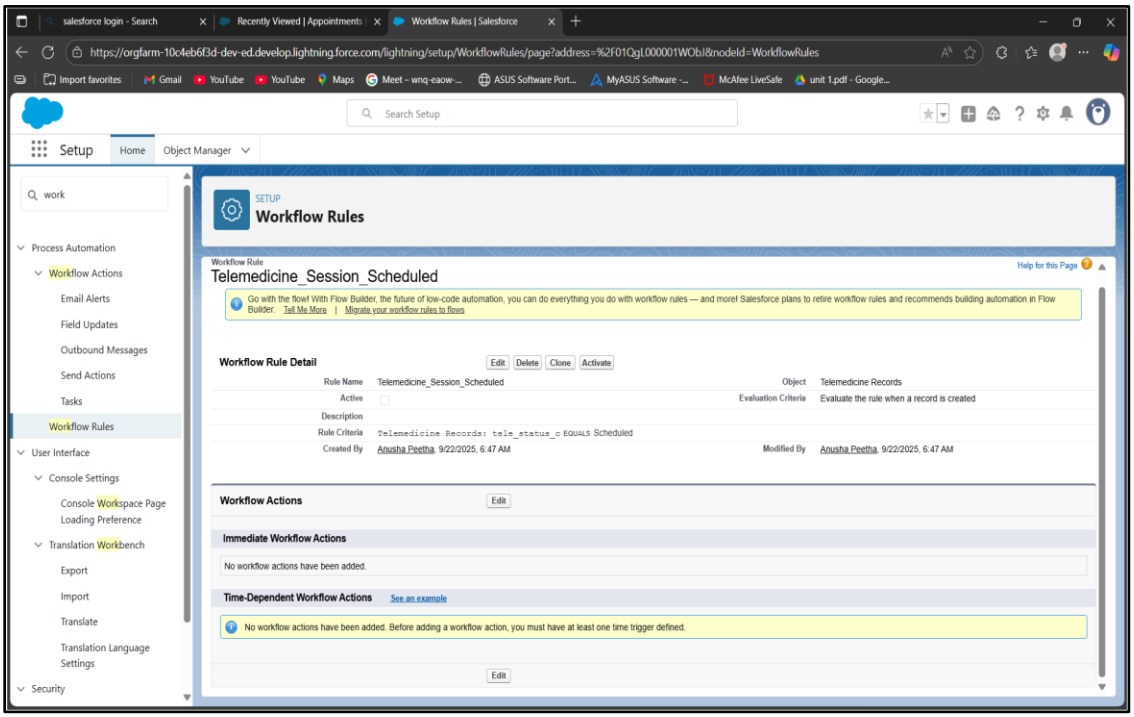
- Go to Setup.
- In the Quick Find box, type the Workflow Rules and click on it.
- Click continue.
- Now click on the New Rule and click on Continue on the Workflow Rule.
- Select the objects: Patients, Payments, Telemedicine Records,
- Click Next.
- Patients:
    - Rule Name: Patient _Status _Update
    - Evaluate Rule: created, and every time it's edited
    - Rule Criteria:
      Field: Status _c, Operator: =, Value: 'Admitted'
    - Click Save & Next and Done



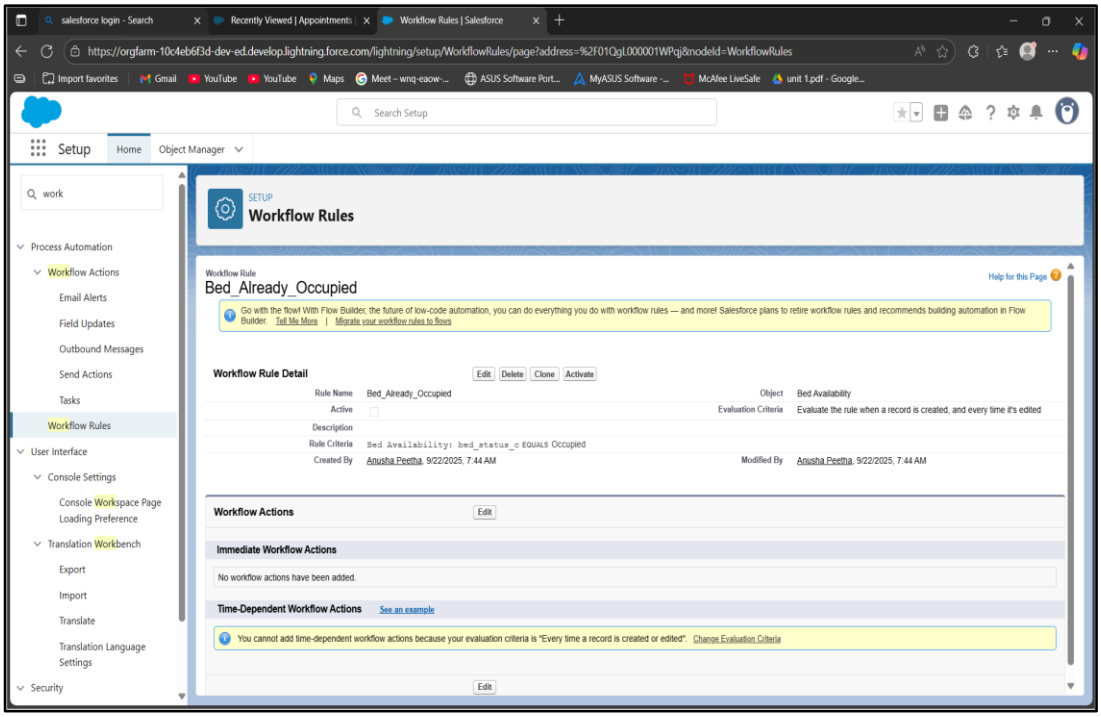
- Payments:
  - Rule Name: Payment _Confirmation
  - Evaluate Rule: Created
  - Rule Criteria:
    Fields: status _c, Operator: =, Value: Completed
  - Click Save & Next and then Done



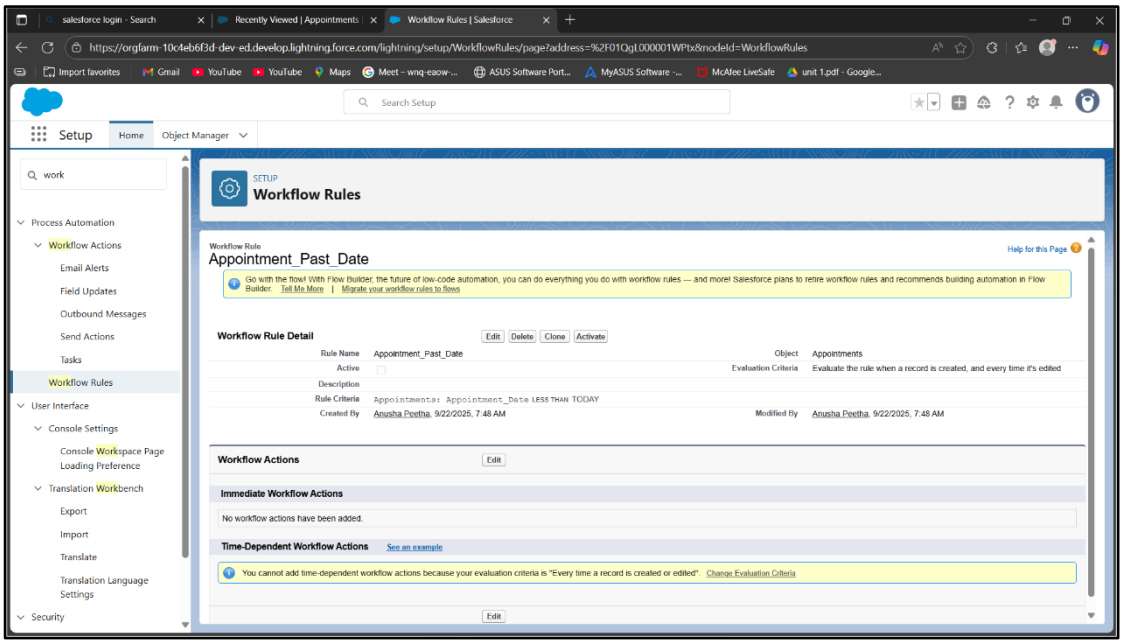
- Telemedicine Records:
  - Rule Name: Telemedicine _session _scheduled
  - Evaluate Rule: created
  - Rule Criteria:
    Field: status _c, Operator: =, Value: Scheduled
  - Click on Save & Next and then click Done



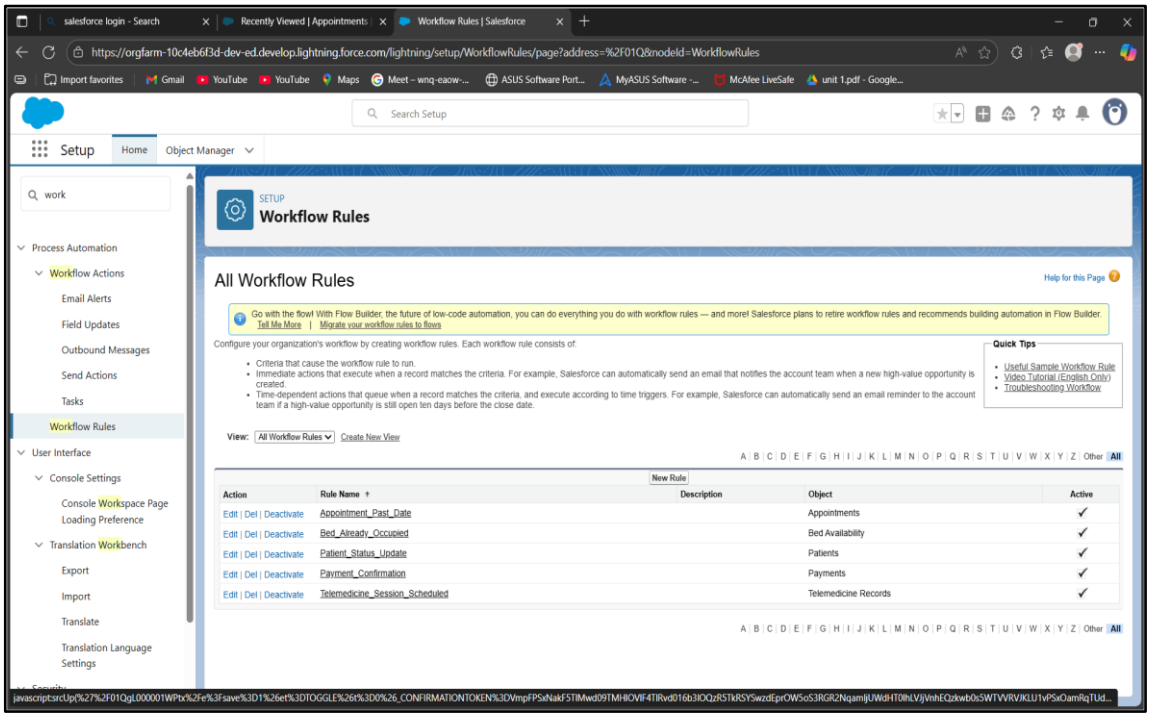
- Bed Availability:
  - Rule Name: Bed _Already _Occupied
  - Evaluation Criteria: created, and every time it's edited
  - Rule Criteria:
    Field: bed _status _c, Operator: =, Value: Occupied
  - Click on Save & Next and then click Done



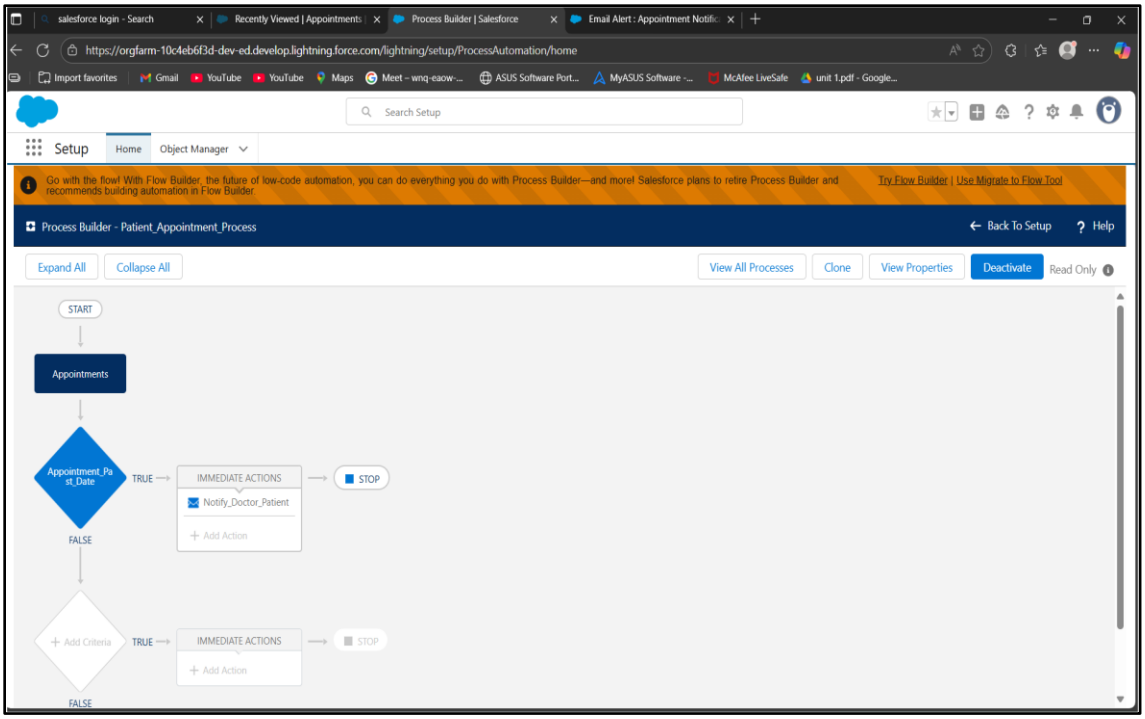
- Appointments:
  - Rule Name: Is _Past _Appointment
  - Evaluation Criteria: created, and every time it's edited
  - Rule Criteria:
    Field: Appointment _Date, Operator: <, Value: TODAY
  - Click on Save & Next and then click Done



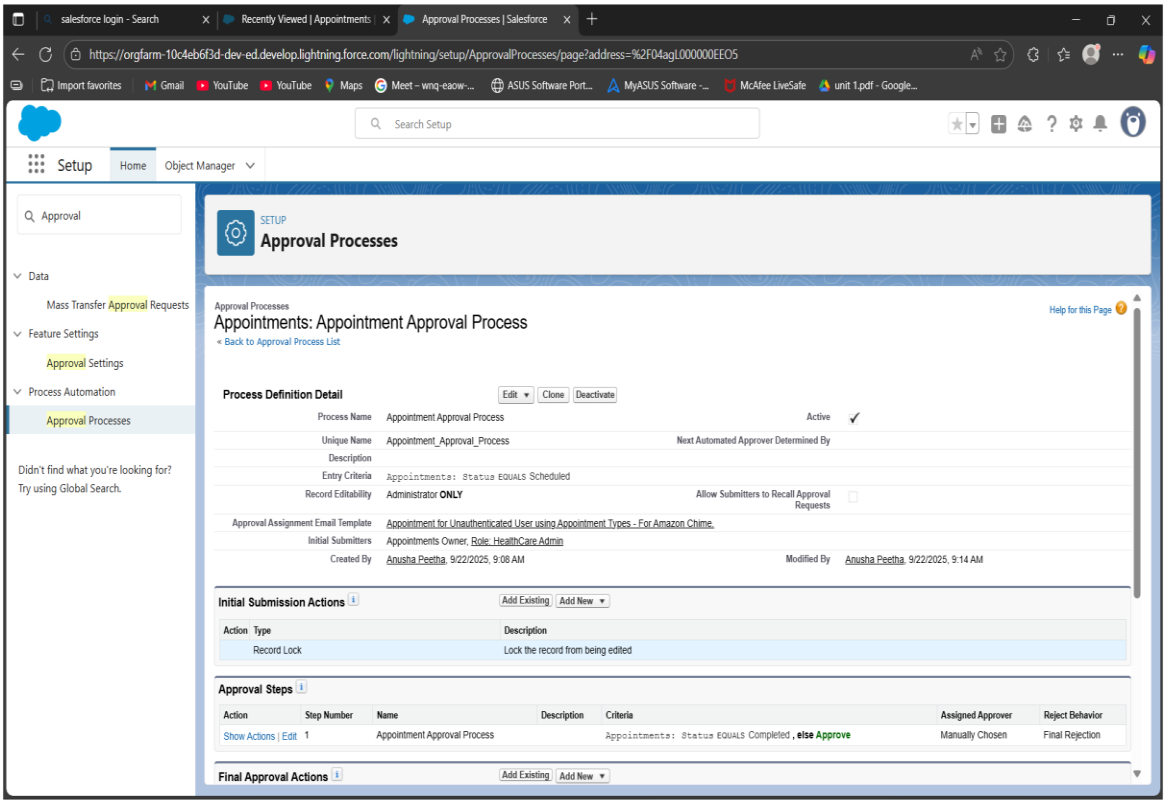
- After creating Workflow Rules for Objects click on the Activate.



- Process Builders:
  - Go to Setup.
  - In the Quick Find box, type the Process Builder and click on it.
  - Click on New.
  - Fill the details:
    - Process Name: Patient _Appointment _Process
    - API Name: Patient _Appointment _Process
    - The Process starts when: A record changes and then click save.
    - Click on +Add Object
    - Object: Appointments
    - Start the Process: when a record is created or edited and then click save.
    - Click +Add Criteria
    - Name: Appointment _Past _Date
    - Criteria: the field reference
    - Click +Add Action and Action Type: Email Alert
    - Action Name: Notify _Doctor _Patient
    - Appointment Notification and click on save
    - Click Activate and then Confirm



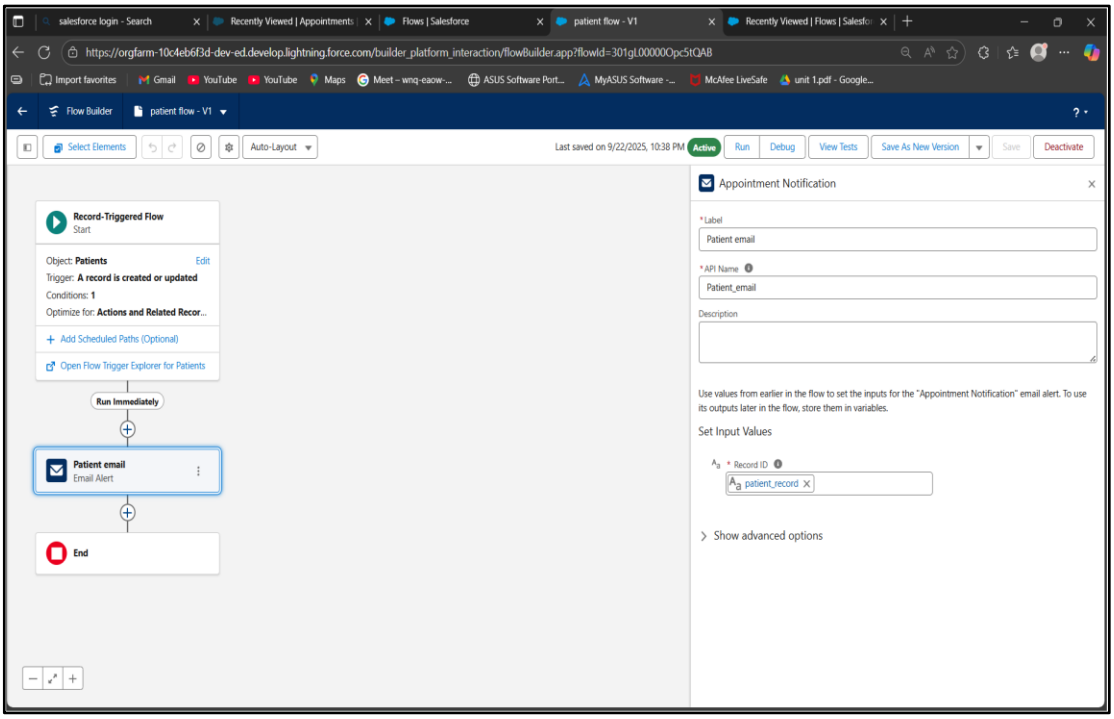
- **Approval Process:**
  - Go to Setup.
  - In the Quick Find box, type the Approval Settings and click on it.
  - Select Appointments.
  - Select the Standard Wizard.



- Process Name: Appointment Approval Process
- Field: Appointment status, Operator: =, Value: Scheduled
- Next and then Appointment Notification
- Select Appointment ID, Appointment Status, Patients, Record Type, Appoint Date, User
- Next, Save and then Click on Activate.

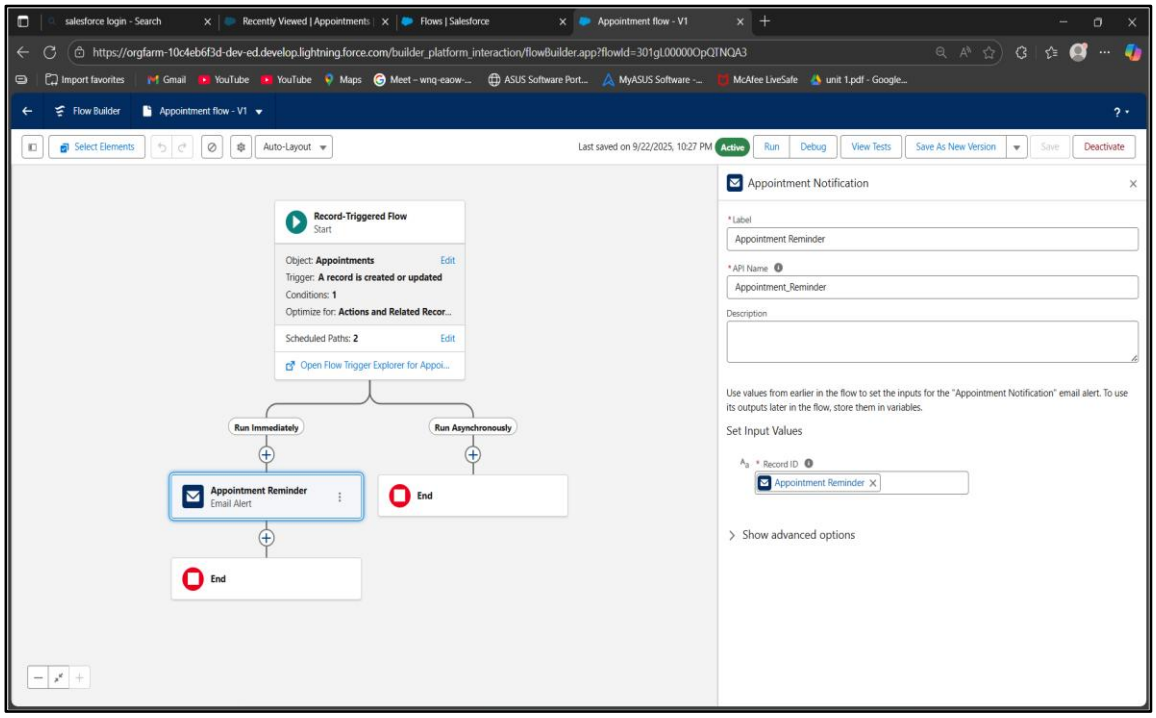
❖ **Flow Builder (Screen, Record-Triggered, Scheduled, Auto-launched):**

- Go to the Setup.
- In the Quick Find box, type the Flow Builder then click on it.
- Click on New Flow.
- Patients:
  - Click on the Record – Triggered Flow
  - Object: Patient _c
  - Trigger: A record is created or updated
  - Select the Condition is met
    Field: status, Operator: =, Value: Confirmed
  - Add element and click on the send email
  - Select the existing one.
  - Select the record on existing one.
  - Name: Patient flow and then save.
  - Click save and then activate.



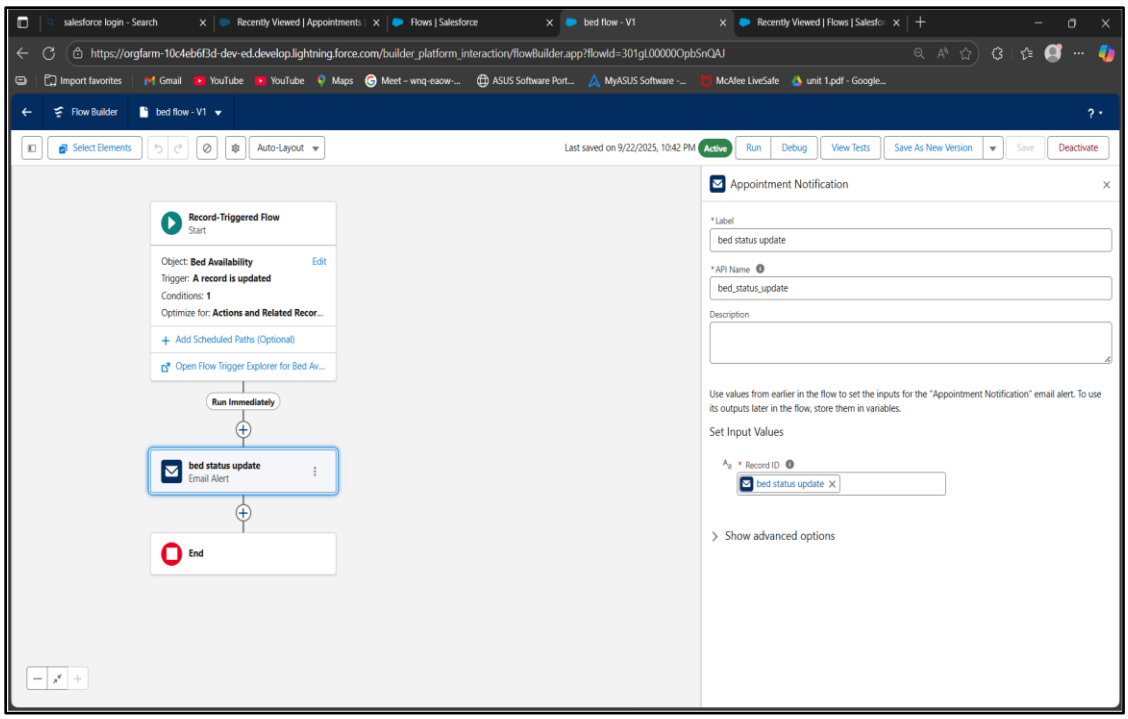
- Appointments:
    - Click on the Record- Triggered Flow
    - Object: Appointment _c
    - Trigger: A record is created or updated
    - Select the Condition is met

      Field: Appointment _status, Operator: =, Value: Scheduled
    - Add element and click on the send email
    - Select the existing one.
    - Select the record on existing one.
    - Name: Appointment flow and then save.
    - Click save and then activate



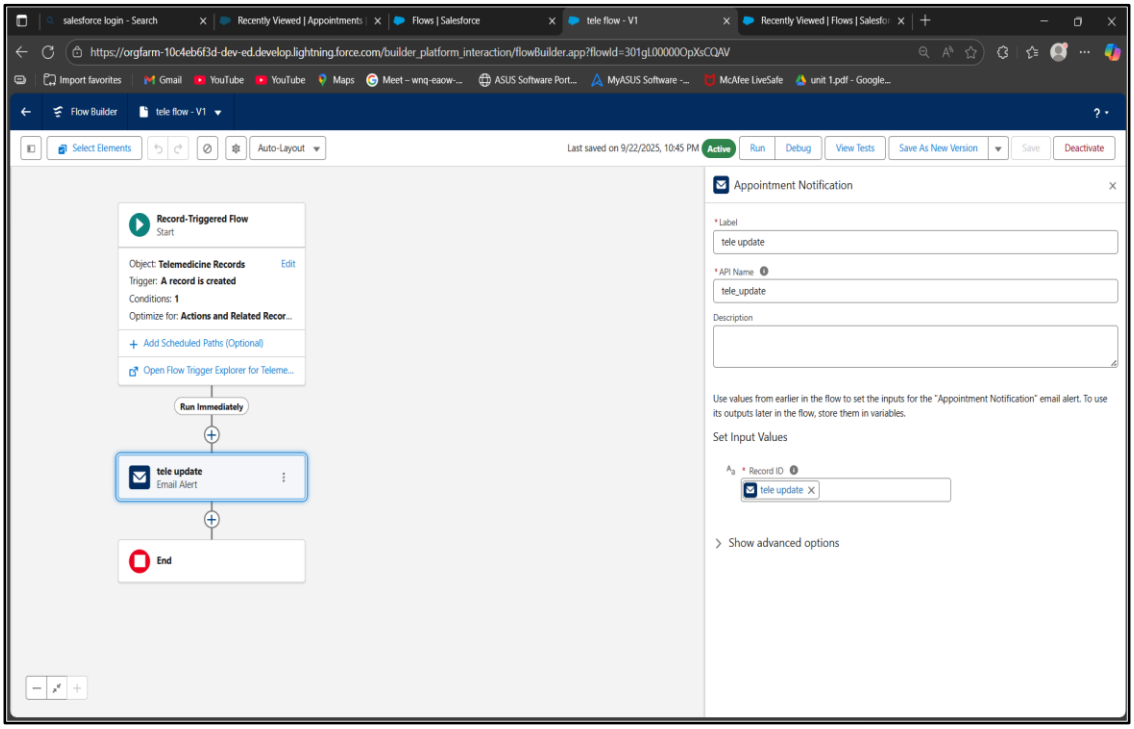
- Bed Availability:
    - Click on the Record – Triggered Flow
    - Object: Bed _Availability _c
    - Trigger: A record is updated
    - Select the Condition is met

      Field: status, Operator: =, Value: Available
    - Add element and click on the send email
    - Select the existing one.
    - Select the record on existing one.
    - Name: Bed flow and then save.
    - Click save and then activate



- Telemedicine Records:
  - Click on the Record – Triggered Flow
  - Object: Telemedicine _c
  - Trigger: A record is created
  - Select the Condition is met

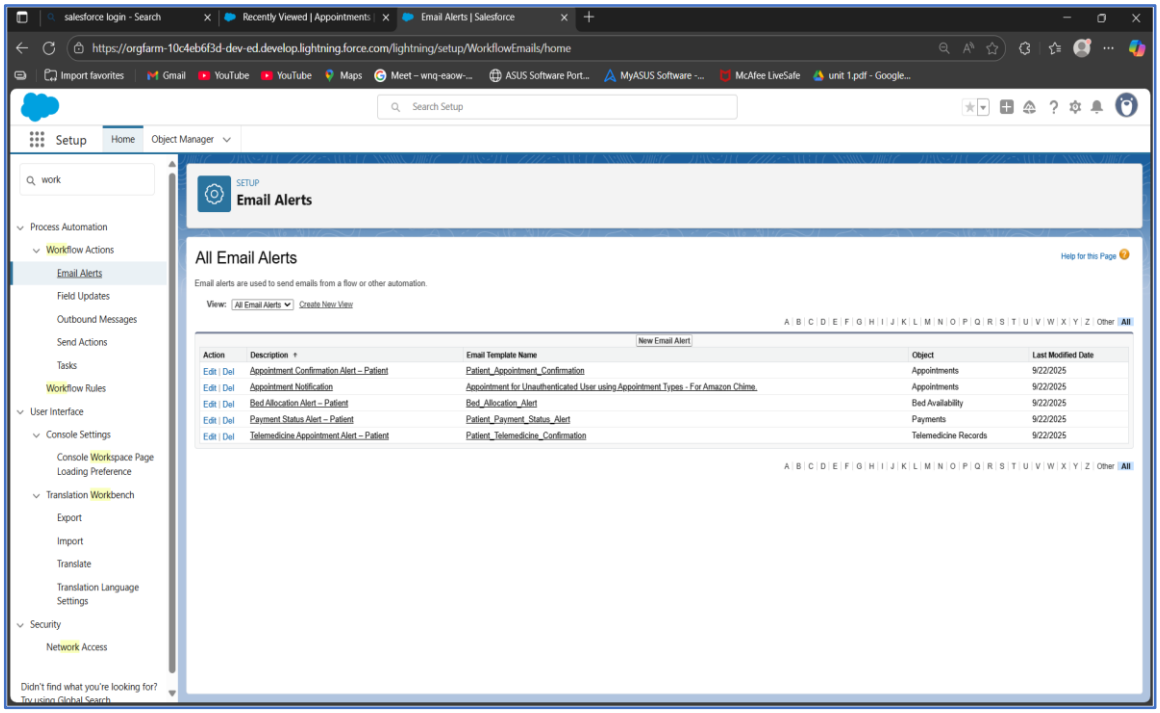
    Field: tele _status, Operator: =, Value: Scheduled
  - Add element and click on the send email
  - Select the existing one.
  - Select the record on existing one.
  - Name: tele flow and then save.
  - Click save and then activate.



❖ **Email Alerts:**

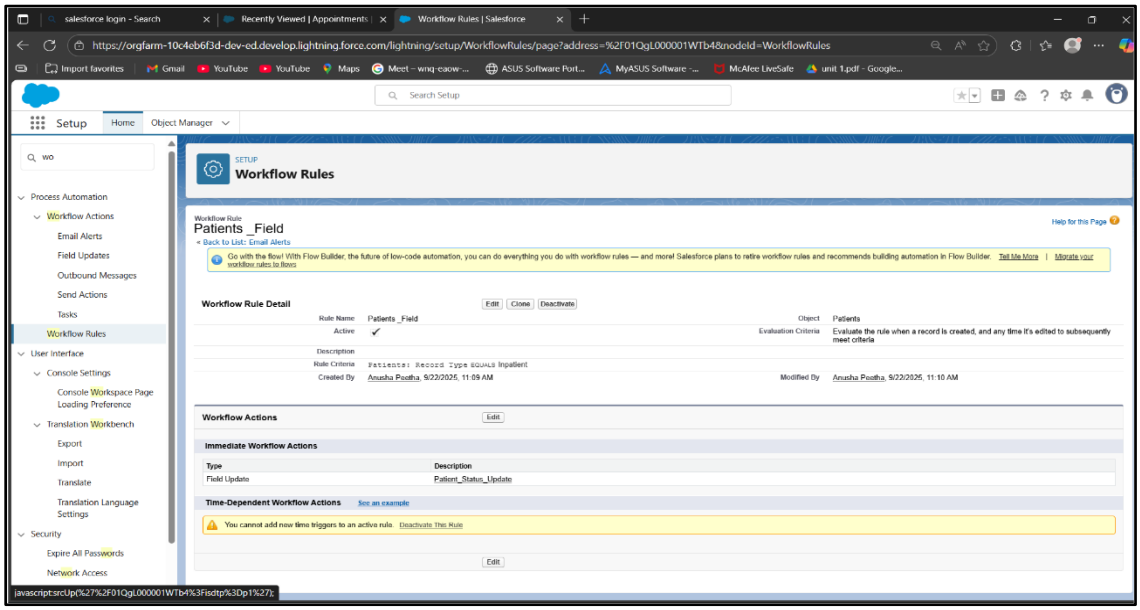
- Go to Setup.
- In the Quick Find box, type the Email Alerts and click on it.
- If you don't have the Email template first you need to create the Email template by typing the Classic email template in Quick Find box.
- Create New Template.
- Appointments:
  - Description: Appointment Confirmation Alert – Patient
  - Object: Appointment _c
  - Email Template: Patient _Appointment _Confirmation
  - Recipients: HealthCare Admin, Patient Coordinator, Doctor
- Bed Allocation:
  - Description: Bed Allocation Alert – Patient
  - Object: Bed Availability _c
  - Email Template: Patient _Bed _Allocation _Alert
  - Recipients: HealthCare Admin, Patient Coordinator, Doctor
- Telemedicine Reports:
  - Description: Telemedicine Appointment Alert – Patient
  - Object: Telemedicine _c
  - Email Template: Patient _Telemedicine _Confirmation
  - Recipients: HealthCare Admin, Patient Coordinator, Doctor
- Payments:
  - Description: Payment Status Alert -Patient
  - Object: Payment _c

- Email Template: Patient _Payment _Status _Alert
- Recipients: HealthCare Admin, Patient Coordinator, Doctor



❖ **Fields Updates:**

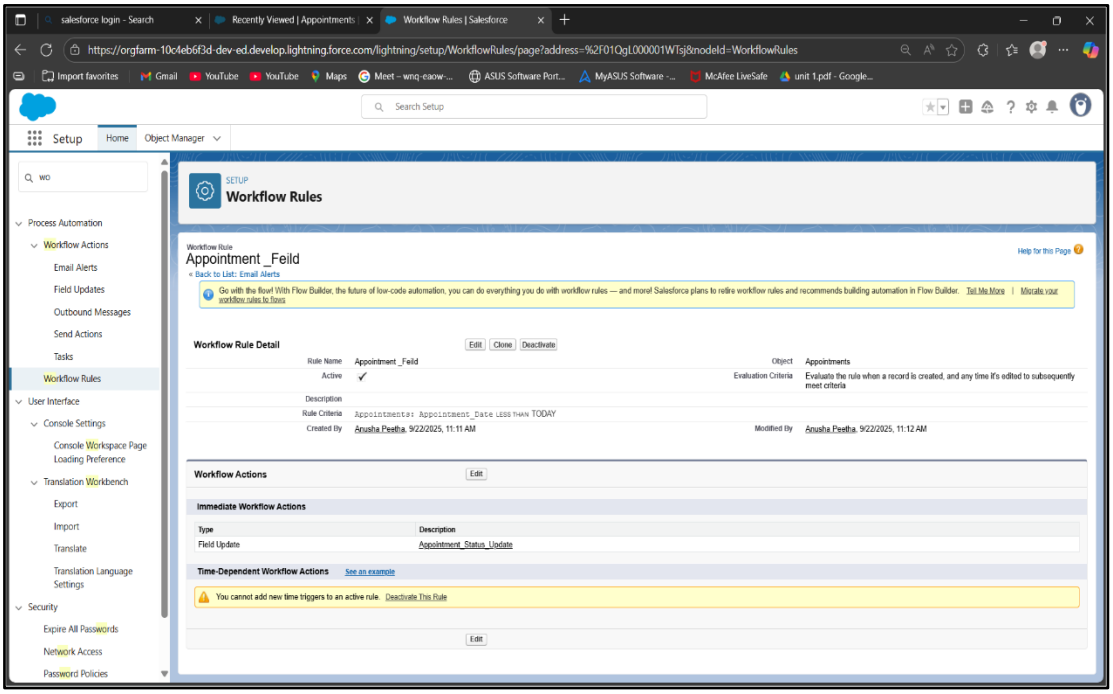
- ➢ Go to Setup.
- ➢ In the Quick Find box, type the Workflow Layout and click it.
- ➢ Click on New Rule.
- ➢ Select Object: Patient _c
  - Rule Criteria:
    Fields: Patient, Operator: =, Value: Inpatient



- Click on Save
- Click on Add immediate Action and then select New Field Update
- Name: Patient _Status _Update
- Field to update: Status _c
- New value: Admitted
- Click on Save and then Activate

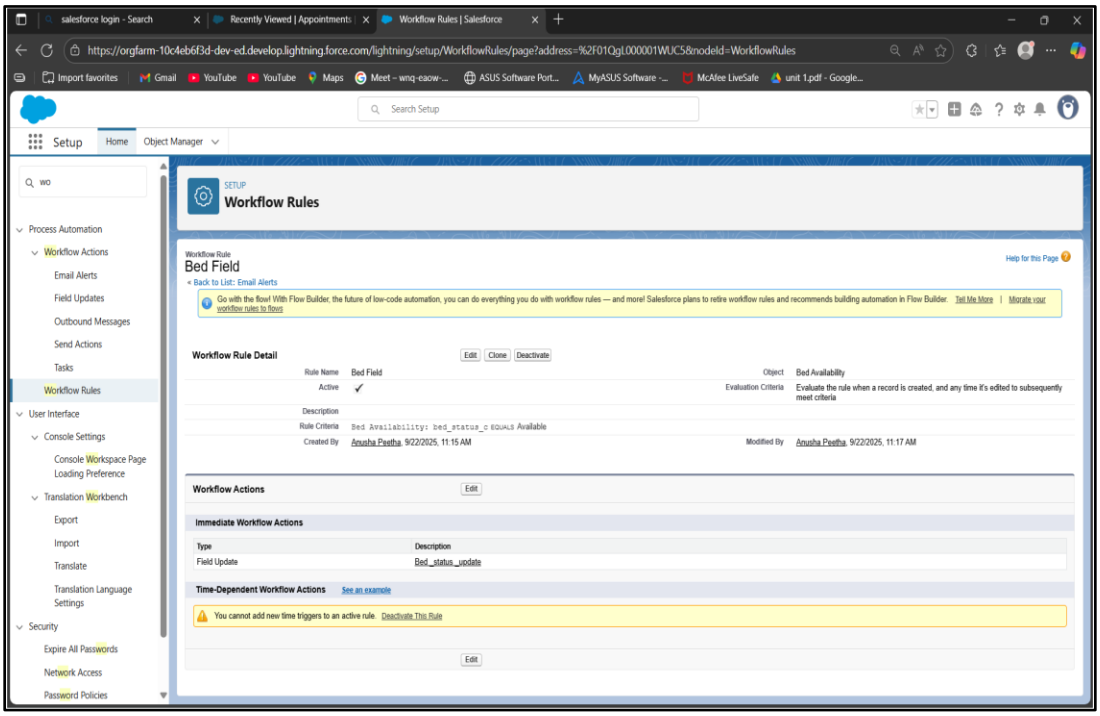
➢ Select Object: Appointment _c

- Rule Criteria:
  Fields: Appointment _Date, Operator: less than, Value: TODAY
- Click on Save
- Click on Add immediate Action and then select New Field Update
- Name: Appointment _Status _Update
- Field to update: Status _c
- New value: Completed
- Click on Save and then Activate

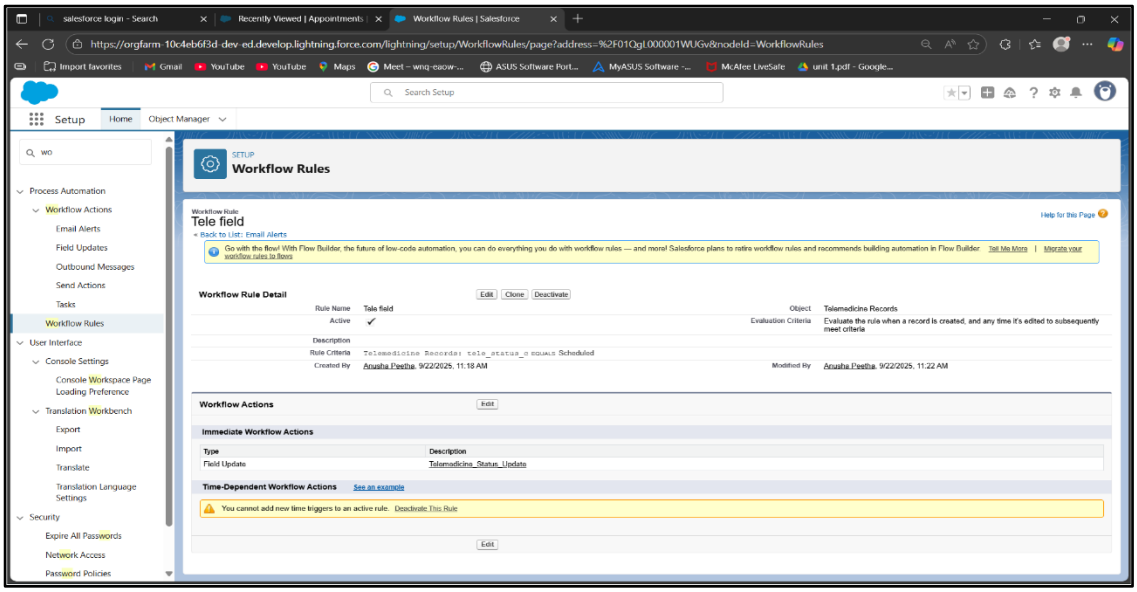


➢ Select Object: Bed _Availability _c

- Rule Criteria:
  Fields: Bed _status, Operator: =, Value: Available
- Click on Save
- Click on Add immediate Action and then select New Field Update
- Name: Bed _Status _Update
- Field to update: Status _c
- New value: Available
- Click on Save and then Activate



- Select Object: Telemedicine Reports
  - Rule Criteria:
    Fields: Telemedicine _status, Operator: =, Value: Completed
  - Click on Save
  - Click on Add immediate Action and then select New Field Update
  - Name: Tele _Status _Update
  - Field to update: Status _c
  - New value: Completed



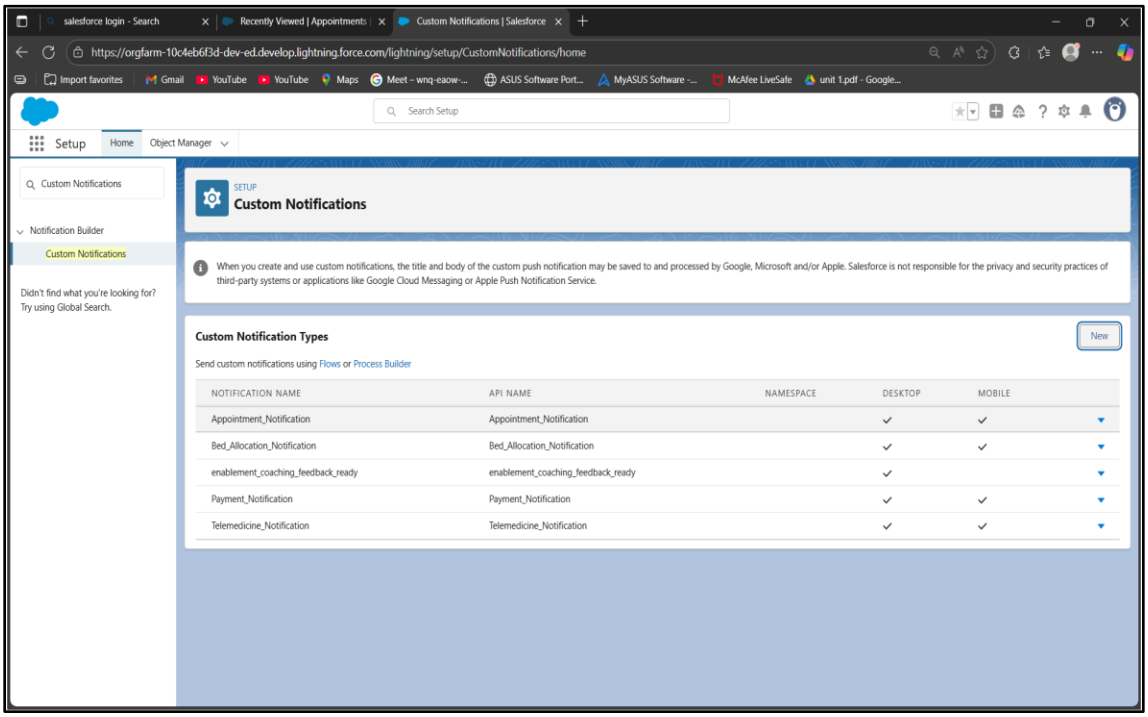
- Click on Save and then Activate

❖ **Tasks:**

Task setup was finished in the previous step. No additional changes are needed for tasks.

❖ **Custom Notifications:**

- ➢ Go to Setup.
- ➢ In the Quick Find box, type the Custom Notifications and click on it.
- ➢ Click on New.
- ➢ Appointment:
  - Name: Appointment _Notification
  - Supported Channels: Desktop, Mobile
- ➢ Bed Availability:
  - Name: Bed _Allocation _Notification
  - Supported Channels: Desktop, Mobile
- ➢ Telemedicine Reports:
  - Name: Telemedicine _Notification
  - Supported Channels: Desktop, Mobile
- ➢ Payments:
  - Name: Payment _Notification
  - Supported Channels: Desktop, Mobile
- ➢ Click on Save.

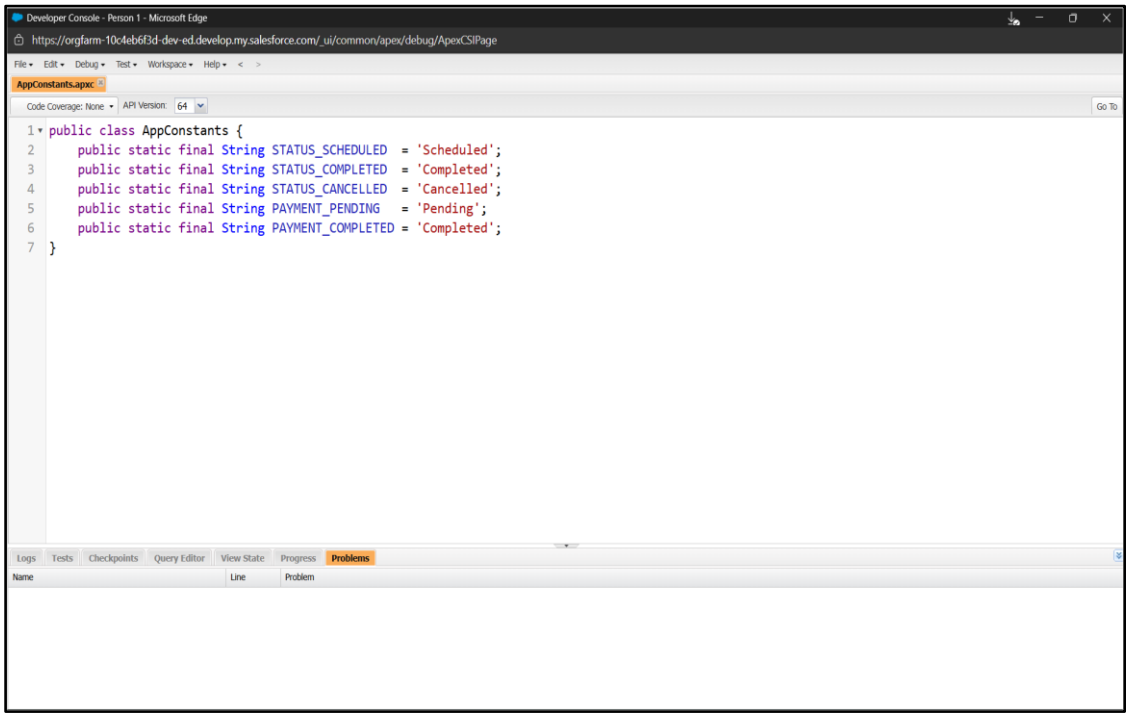


Salesforce Project Phase-5

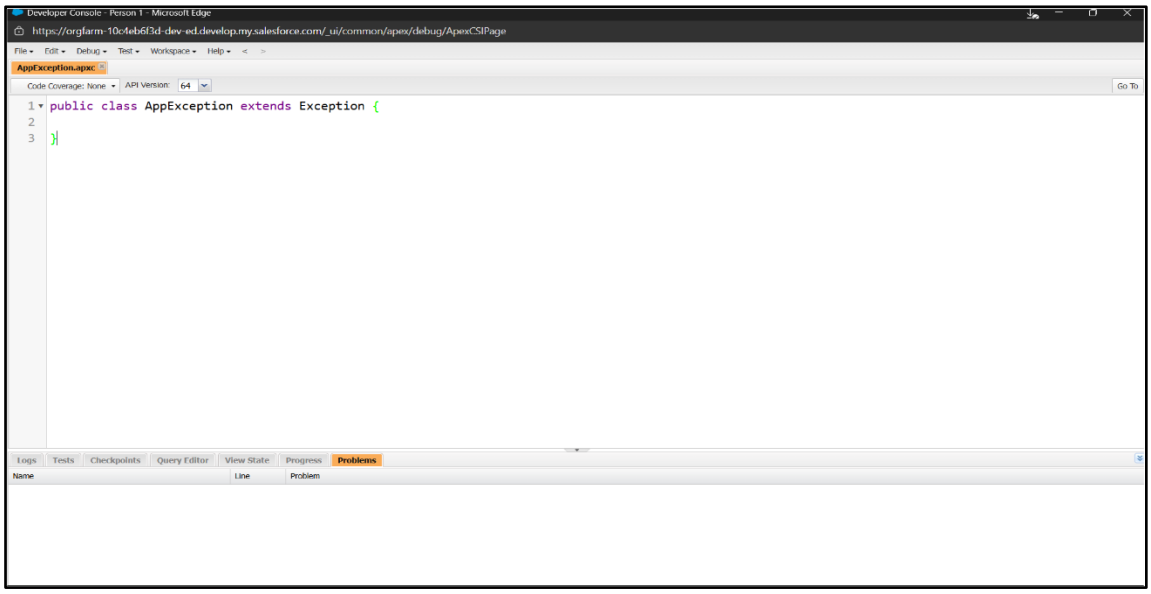
Health Patient Management and Telemedicine CRM

Phase 5: Apex Programming (Developer)

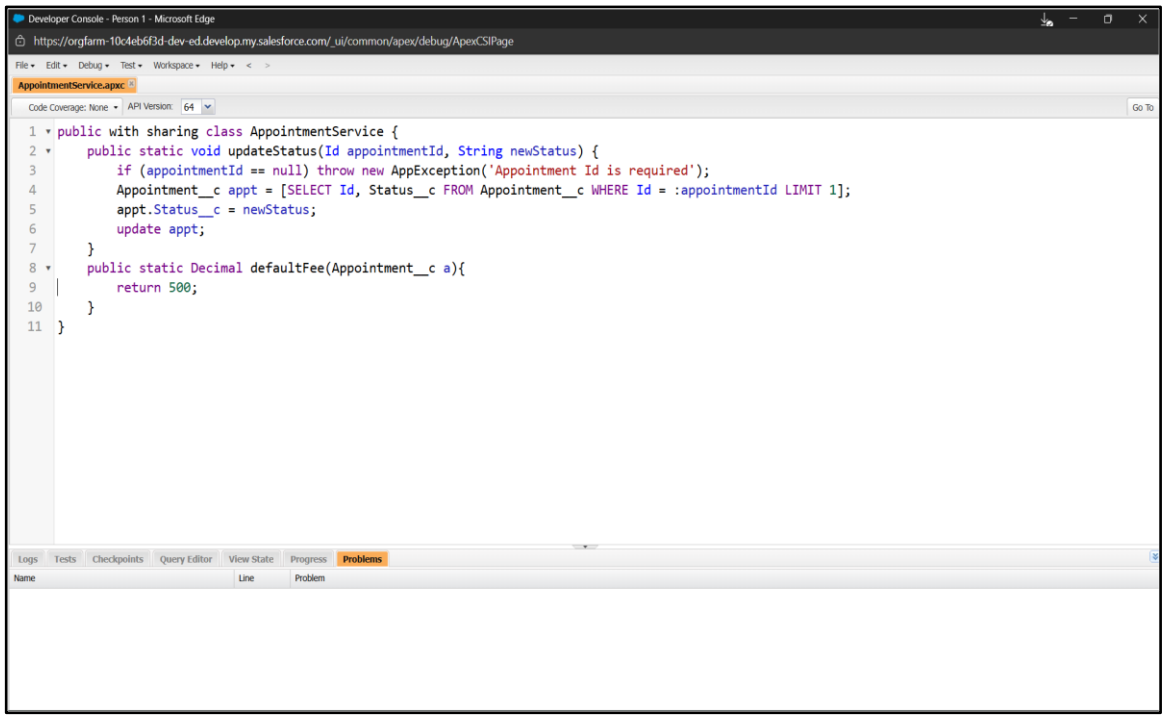
- **Classes & Objects:**
  - Go to the Developer Console.
  - Click on File and select the New and then click on Apex class
  - Click ok.
  - **AppConsonstants.apxc**



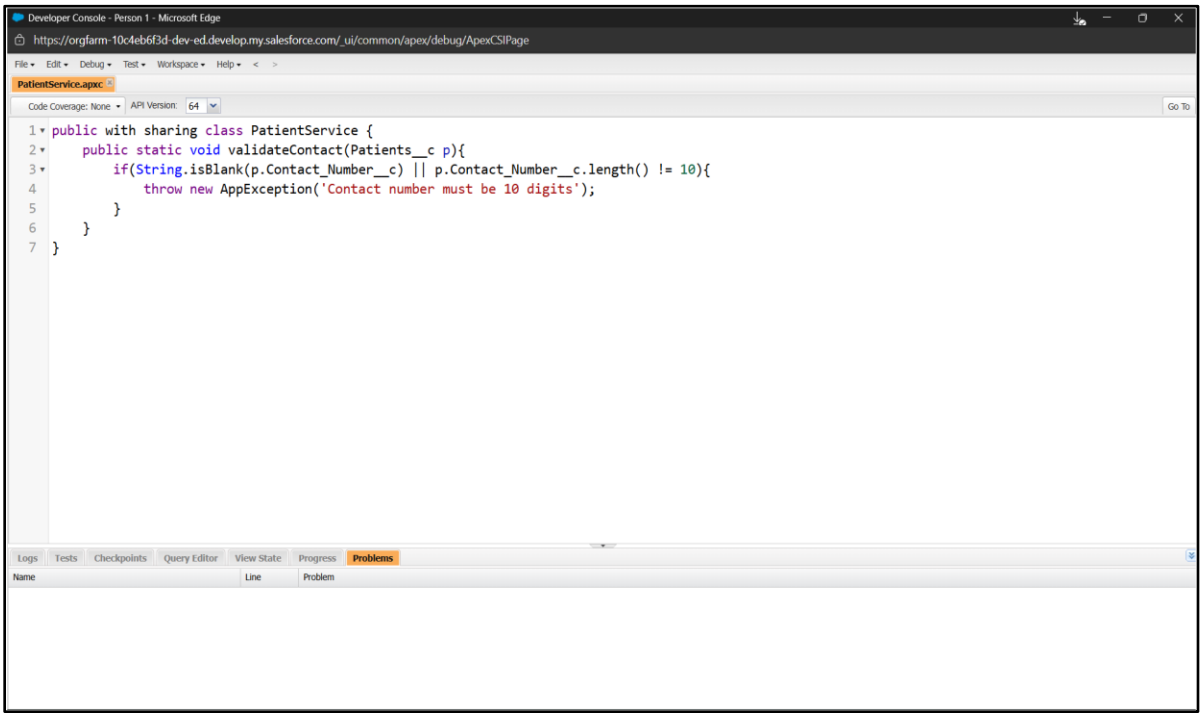
- **AppException.apxc**



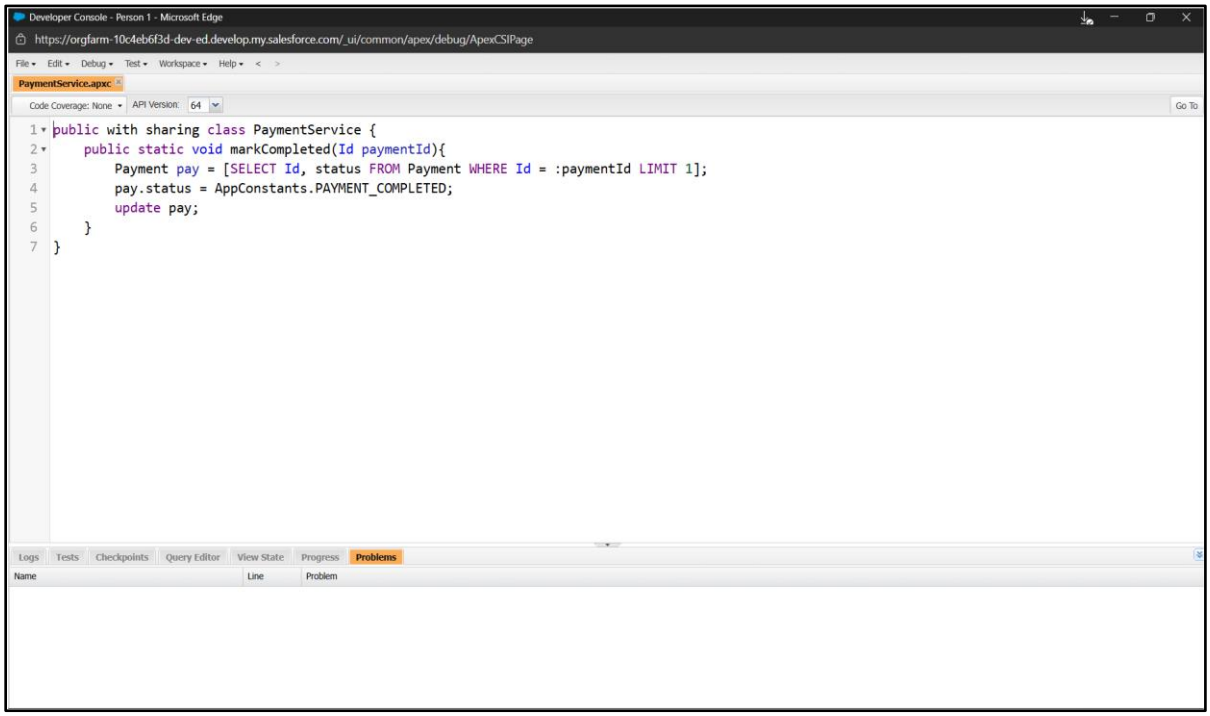
- **AppointmentService.apxc**



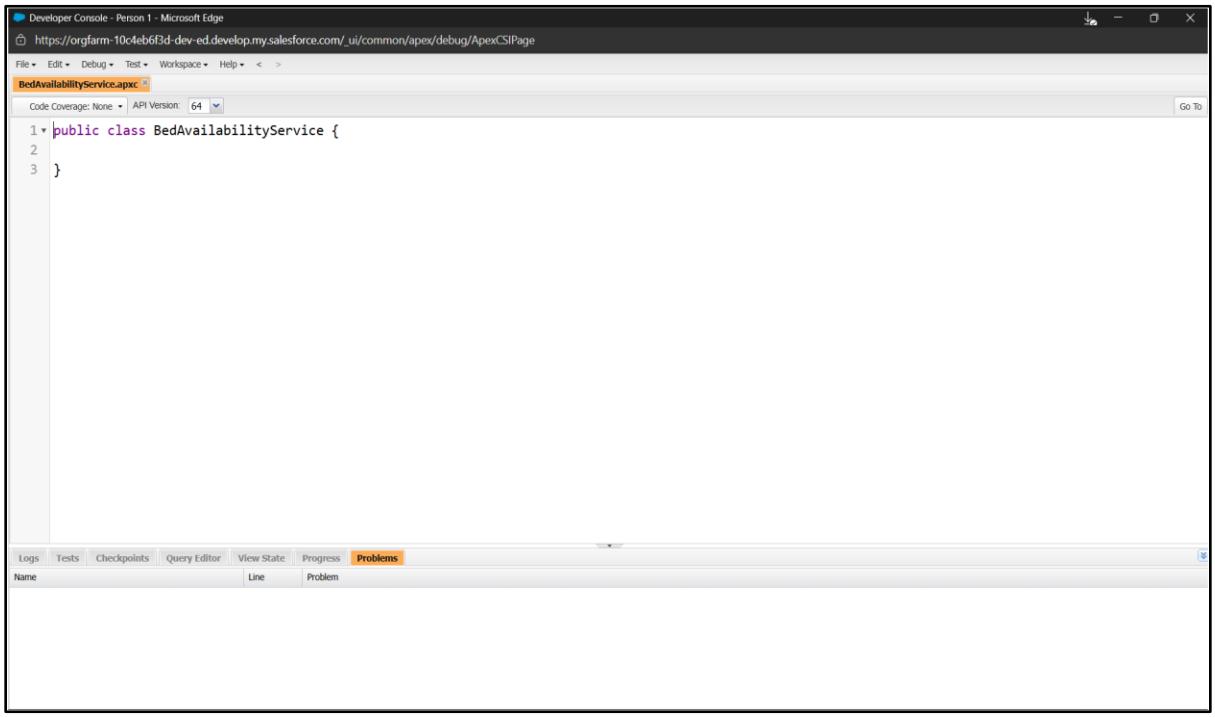
- **PatientService.apxc**



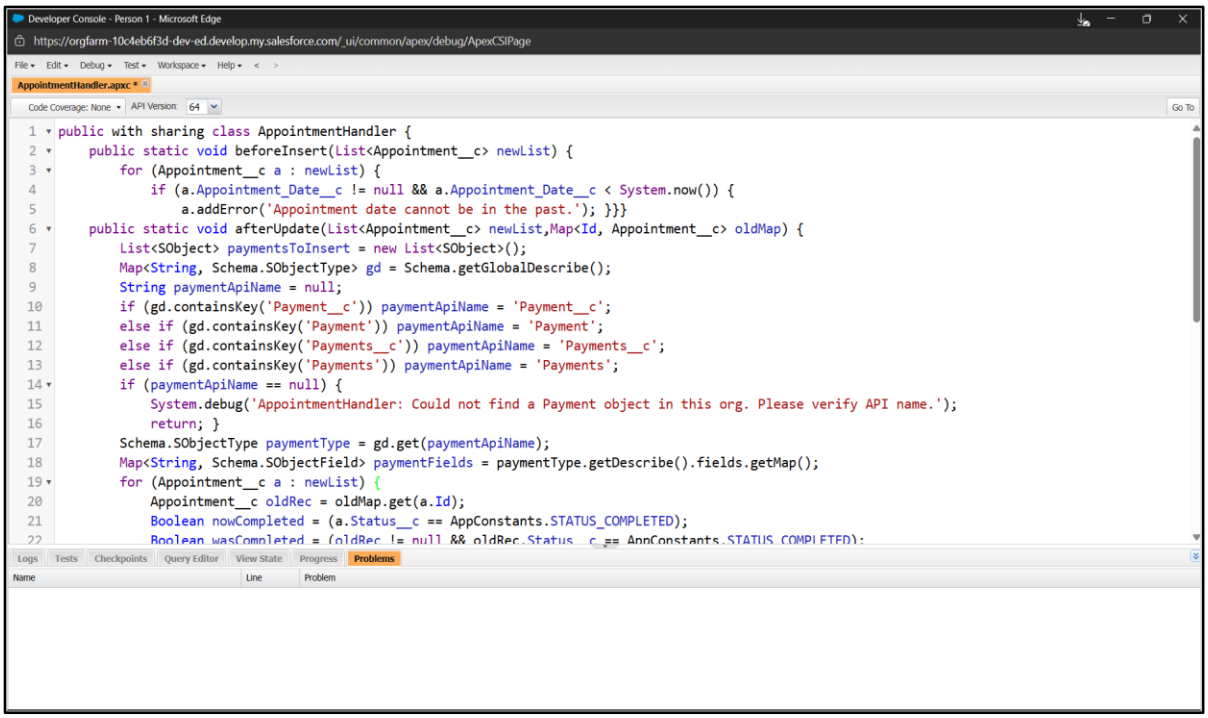
- **PaymentService.apxc**



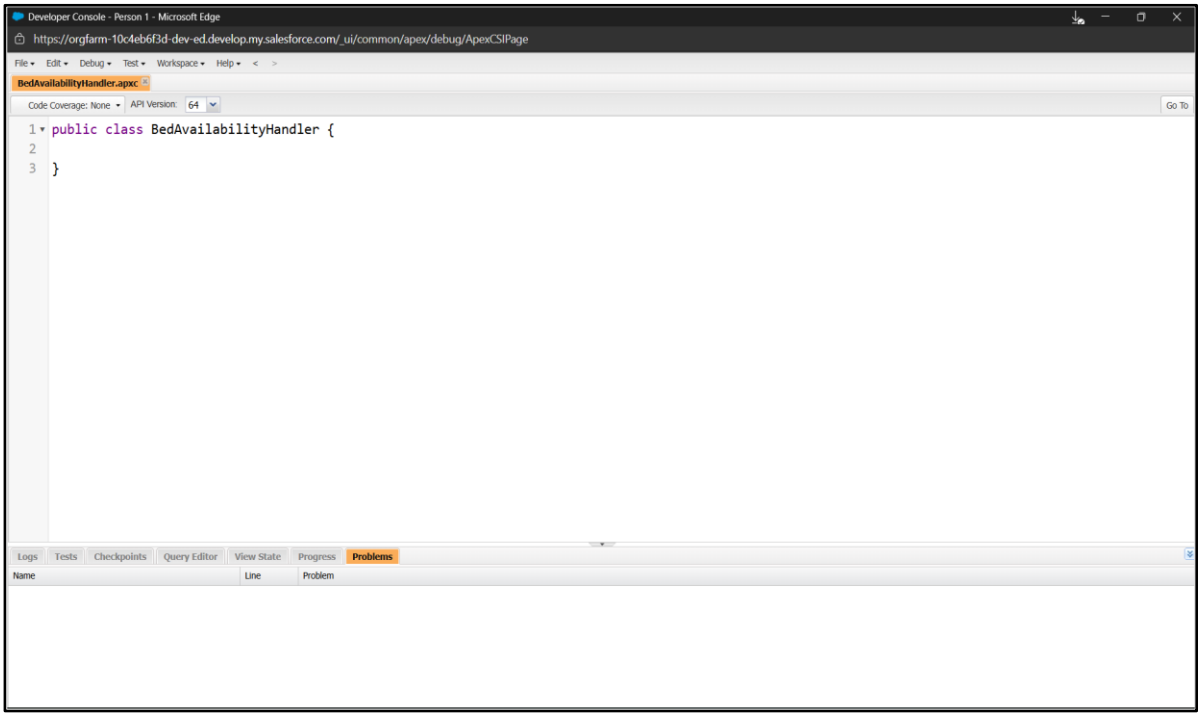
- **BedAvailabilityService.apxc**



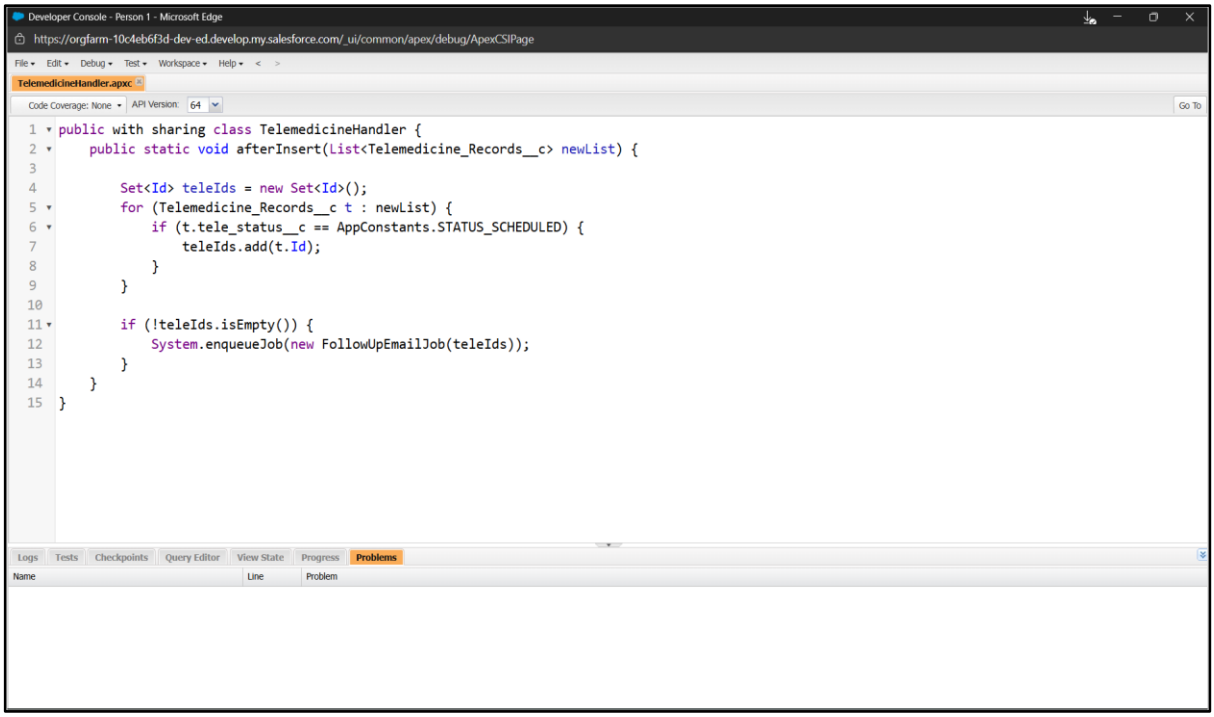
- **AppointmentHandler.apxc**



- **BedAvailabilityHandler.apxc**

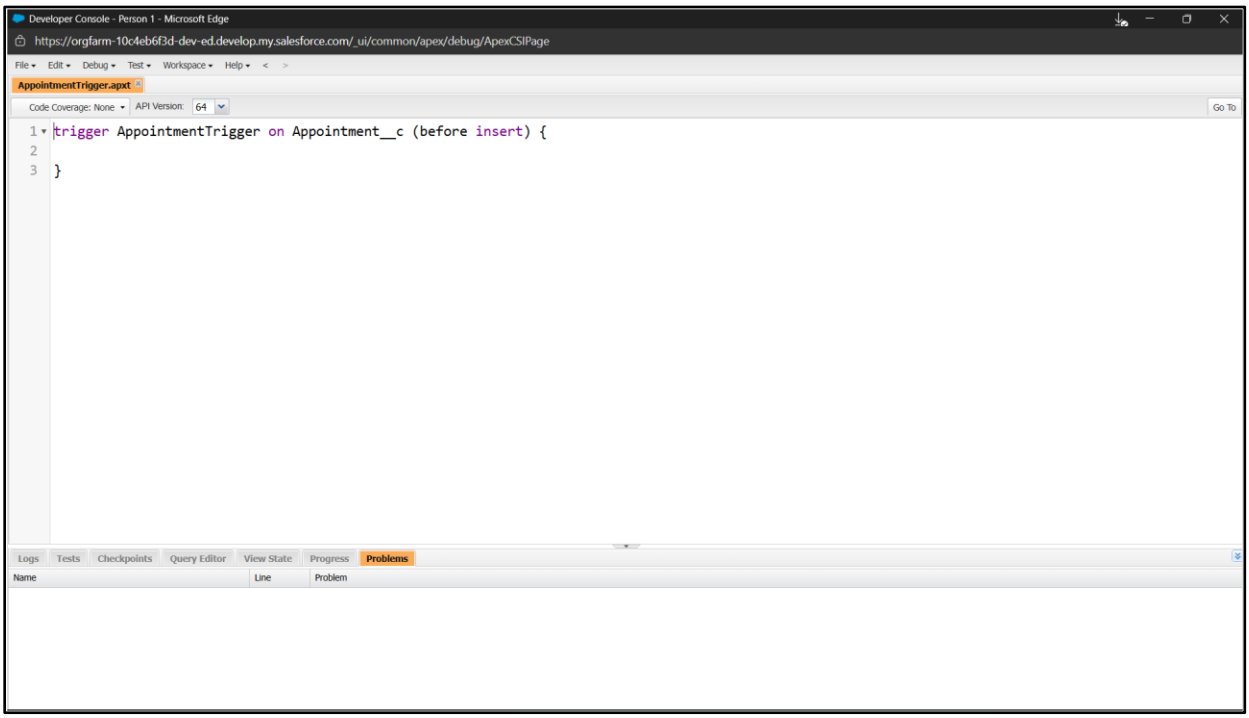


- **TelemedicineHandler.apxc**

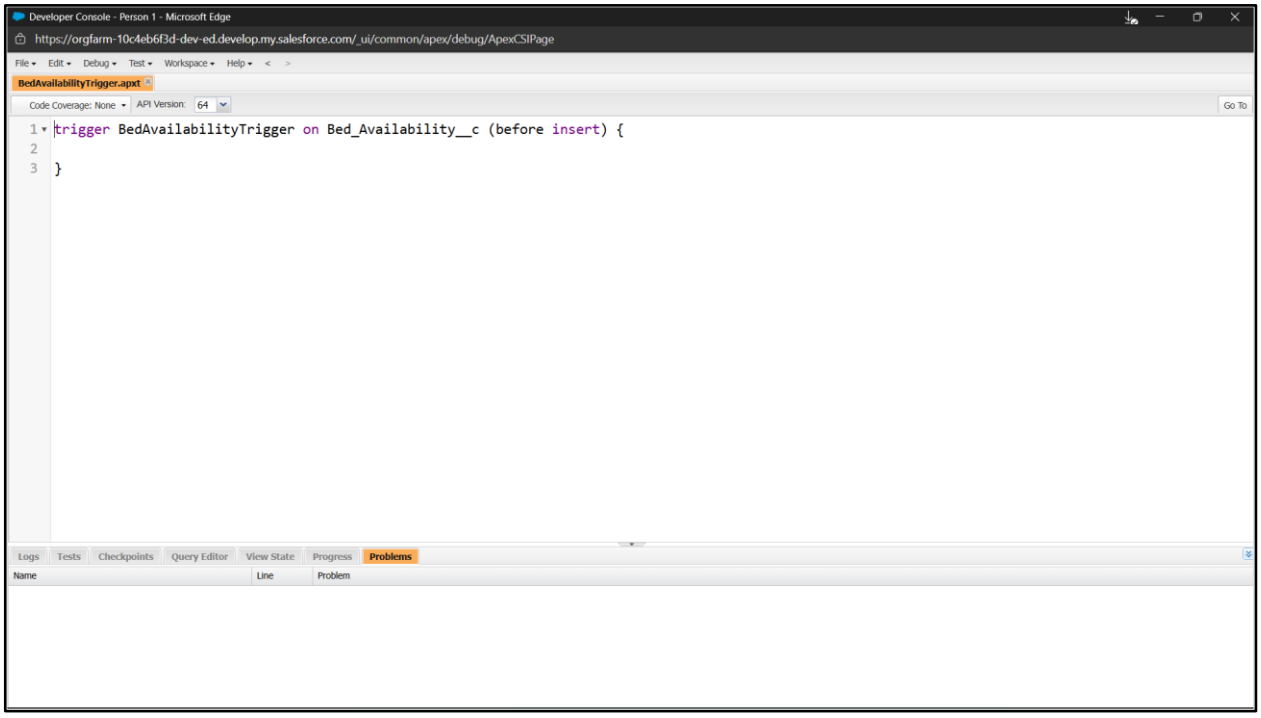


❖ Apex Triggers (before/after insert/update/delete):

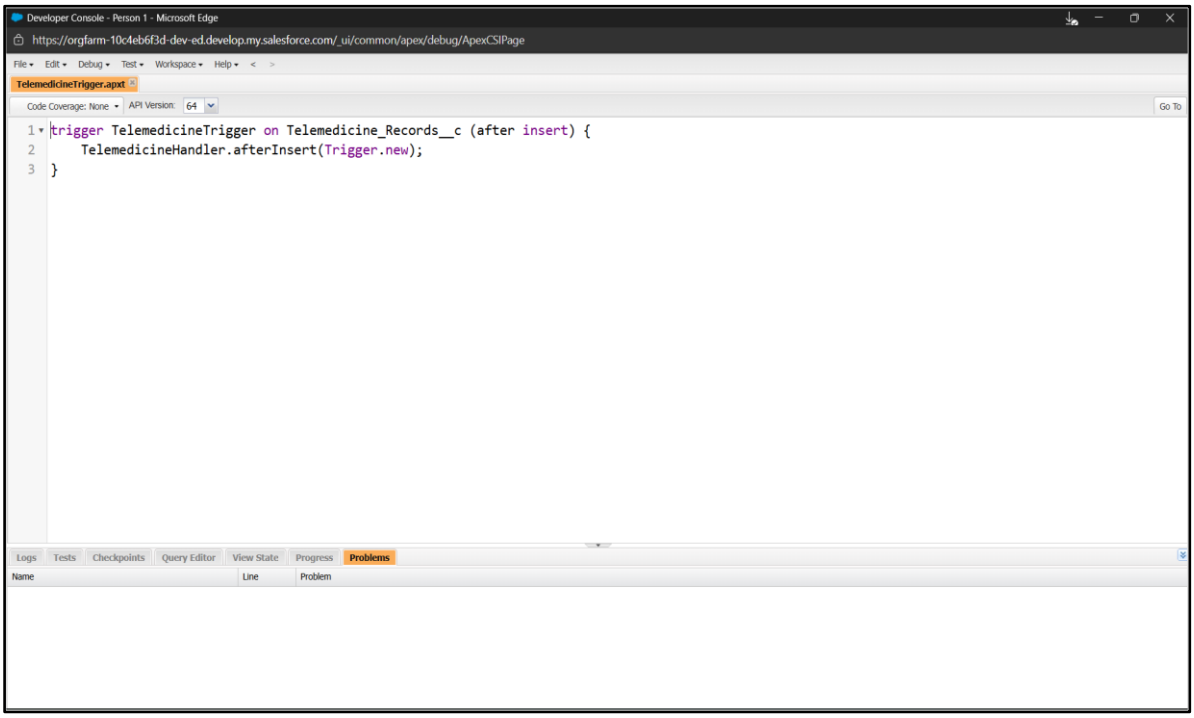
➢ AppointmentTrigger.apxt



➢ BedAvailabilityTrigger.apxt



- **TelemedicineTrigger.apxt**



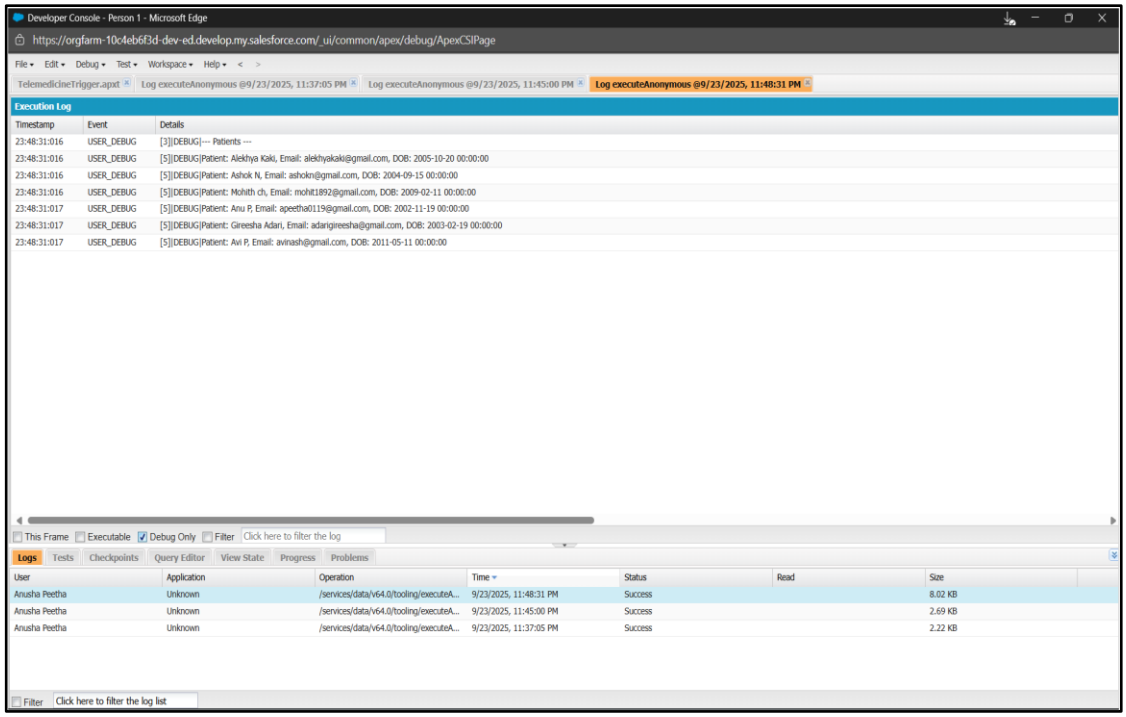
❖ Trigger Design Pattern:

We are already done this in the previous steps. So, no need of doing this.

❖ SOQL & SOSL:

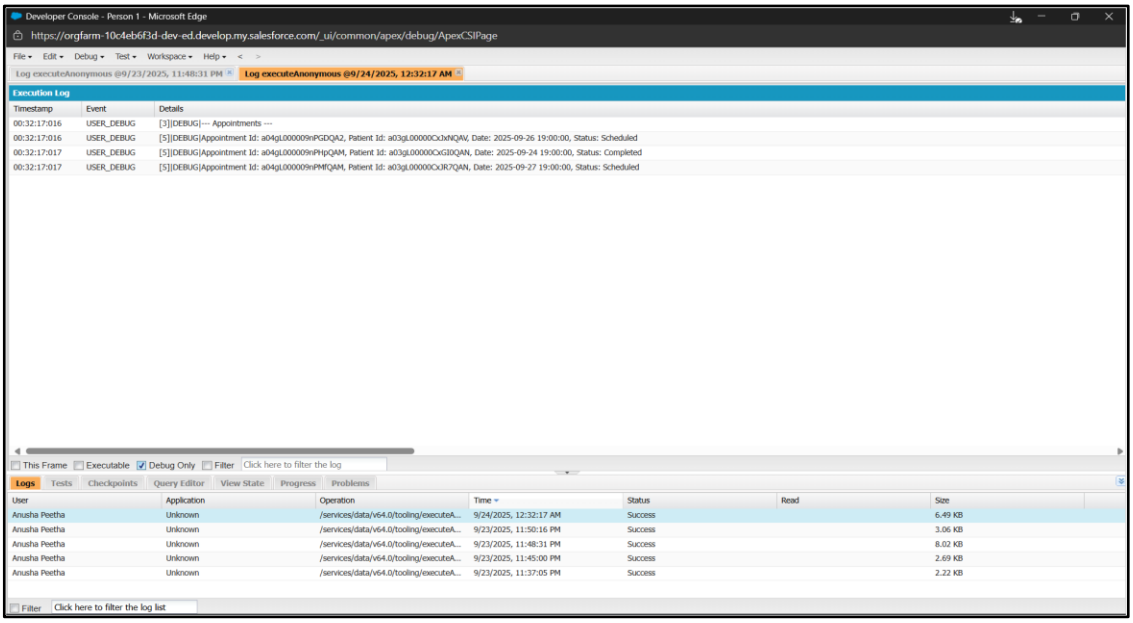
- Go to the Developer Console.
- Click on the Debug and then select the Execute Anonymous Window.
- Type the code and Execute
- Patients:

```
List<Patients__c> patients = [SELECT Id, First_Name__c, Last_Name__c,
Email__c, Date_of_Birth__c
                        FROM Patients__c];
System.debug('--- Patients ---');
for(Patients__c p : patients){
   System.debug('Patient: ' + p.First_Name__c + ' ' + p.Last_Name__c + ',
Email: ' + p.Email__c + ', DOB: ' + p.Date_of_Birth__c);
}
```



- Appointments:

```
List<Appointment__c> appointments = [SELECT Id, Patient__c,
Appointment_Date__c, Status__c
                                    FROM Appointment__c];
System.debug('--- Appointments ---');
for(Appointment__c a : appointments){
    System.debug('Appointment Id: ' + a.Id + ', Patient Id: ' + a.Patient__c + ',
Date: ' + a.Appointment_Date__c + ', Status: ' + a.Status__c);
}
```

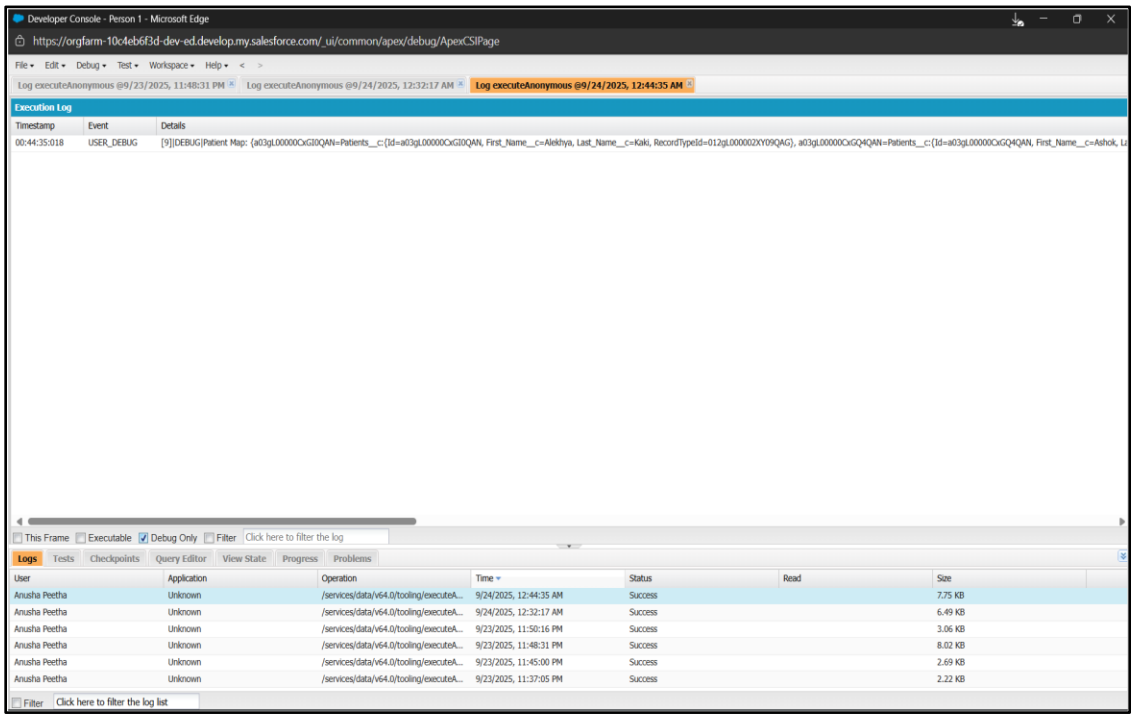


❖ **Collections: List, Set, Map**

- ➢ Go to Developer Console.
- ➢ Click on the Debug and then select the Execute Anonymous Window.
- ➢ Type the code and Execute

```
List<Patients__c> patientList = [SELECT Id, First_Name__c, Last_Name__c
FROM Patients__c];
Set<Id> patientIds = new Set<Id>();
Map<Id, Patients__c> patientMap = new Map<Id, Patients__c>();

for(Patients__c p : patientList){
   patientIds.add(p.Id);
   patientMap.put(p.Id, p);
}
System.debug('Patient Map: ' + patientMap);
```

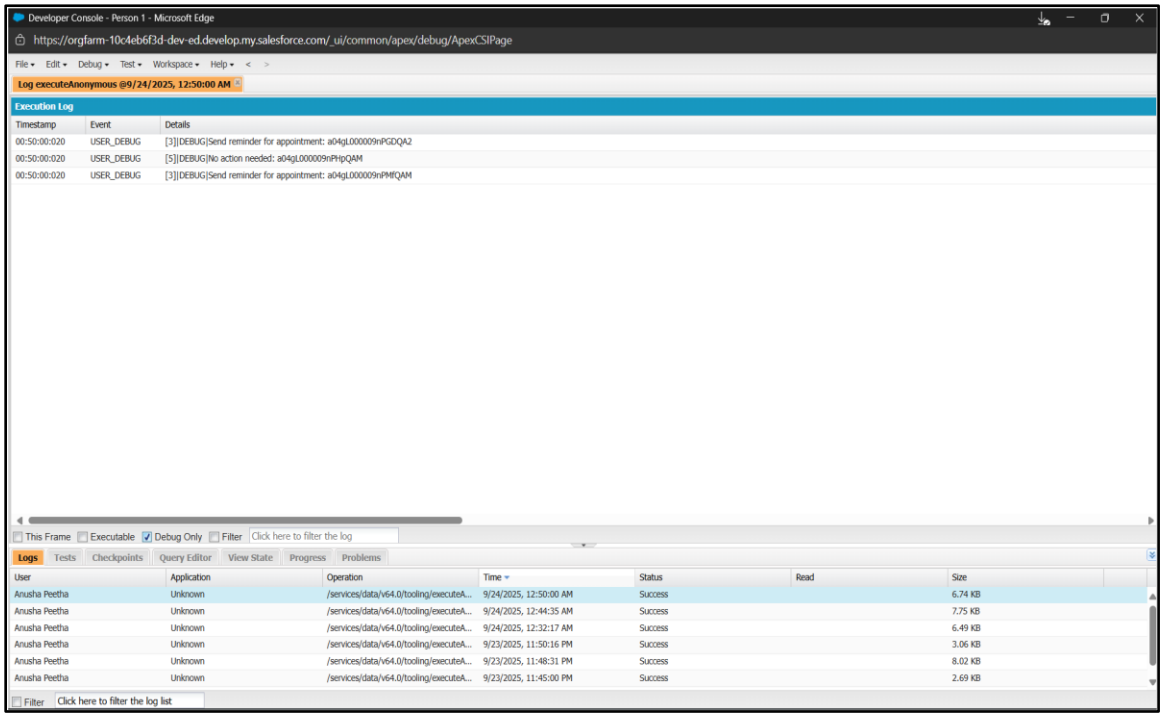


❖ **Control Statements:**

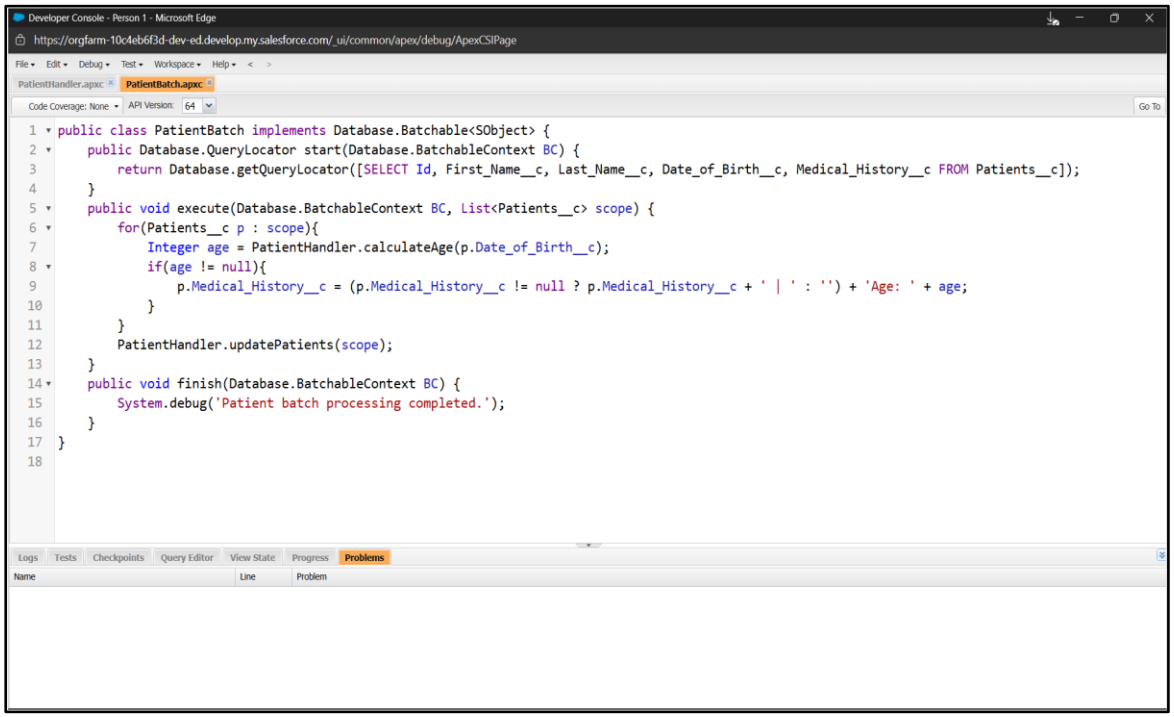
- ➢ Go to Developer Console.
- ➢ Click on the Debug and then select the Execute Anonymous Window.
- ➢ Type the code and Execute

```
for(Appointment__c a : [SELECT Id, Status__c FROM Appointment__c]){
   if(a.Status__c == 'Scheduled'){
      System.debug('Send reminder for appointment: ' + a.Id);
   } else {
```

```
            System.debug('No action needed: ' + a.Id);
        }}
```



- **Batch Apex:**
  - Go to Developer Console
  - Click on the File and select the New and then Apex class.



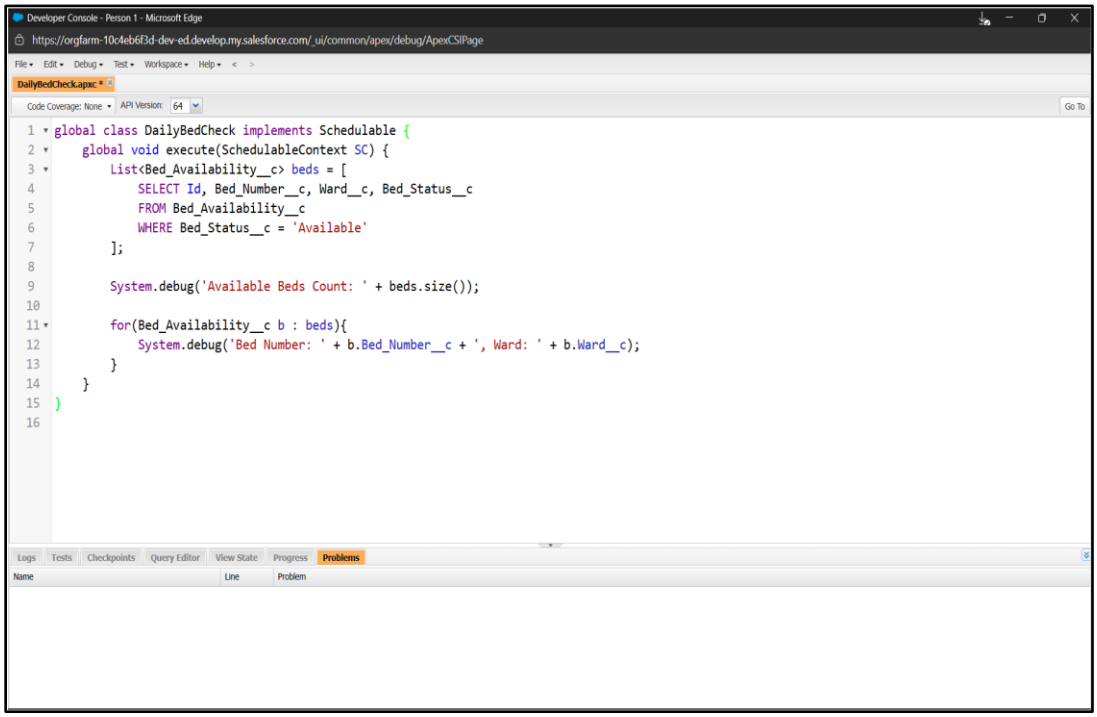
  - Click on the Debug and then select the Execute Anonymous Window.

- Type the code and then Execute.

```
Database.executeBatch(new PatientBatch(), 200);
```

❖ **Scheduled Apex:**

- Go to Developer Console.
- Click on File and New and then select the Apex class.
- Enter the File Name DailyBedCheck
- Type the code,

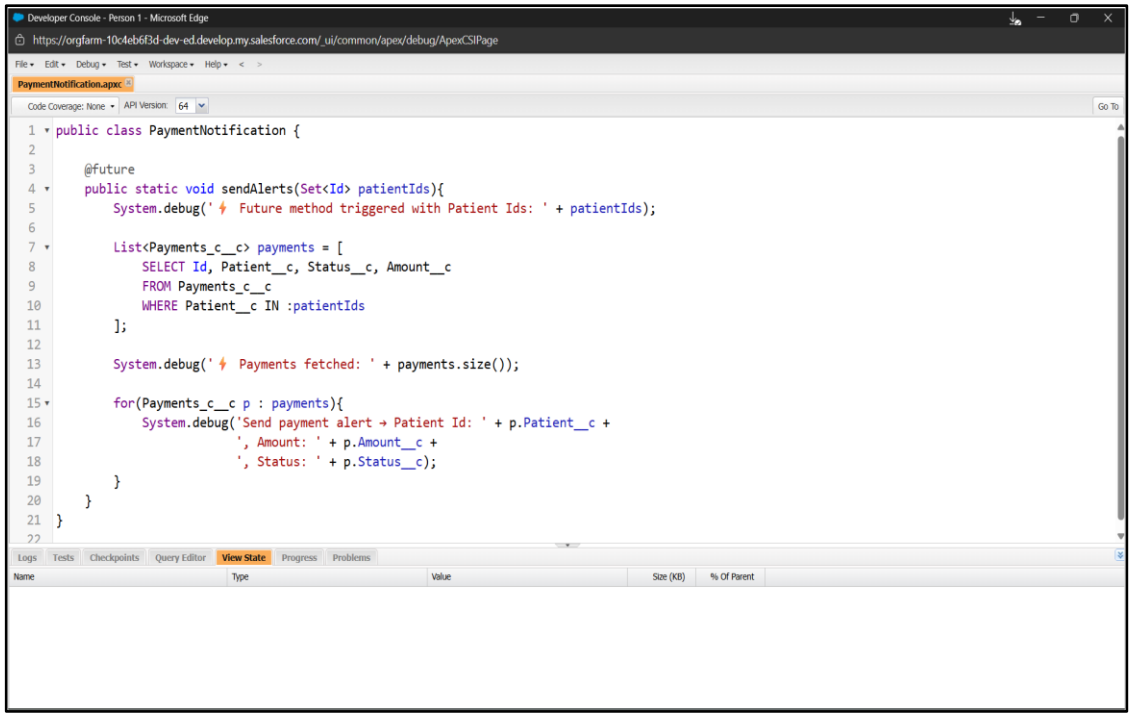


- Save the file.
- Now click on the Debug and select the Execute Anonymous Window.
- Type the code and then Execute.

```
System.schedule('Daily Bed Check', '0 0 8 * * ?', new DailyBedCheck());
```

❖ **Future Methods:**

- Go to Developer Console.
- Click on the File and New and then select the Apex class.
- Enter the File Name PaymentNotification.
- Type the code,



- Save the file.
- Now click on the Debug and select the Execute Anonymous Window.
- Type the code and Execute

```
List<Patients__c> patientList = [SELECT Id FROM Patients__c LIMIT 5];
Set<Id> patientIds = new Set<Id>();
for(Patients__c p : patientList){
   patientIds.add(p.Id);
}
PaymentNotification.sendAlerts(patientIds);
```

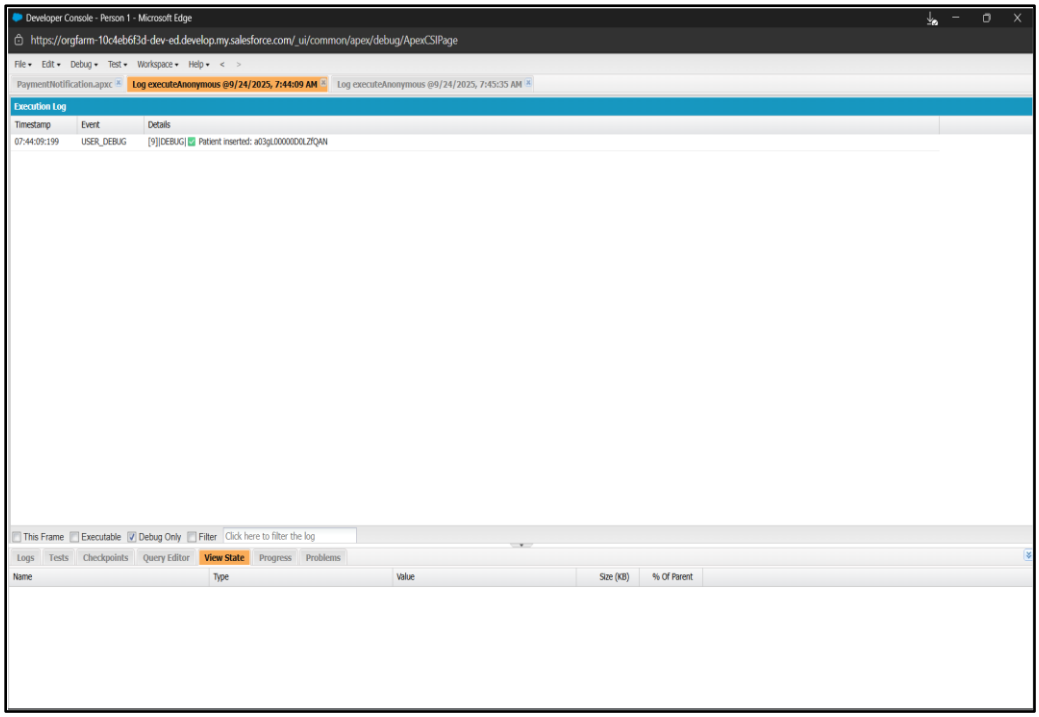
❖ **Exception Handling:**

- Go to Developer Console
- Click on Debug and select the Execute Anonymous Window.
- Type the code and Excute

Patients:

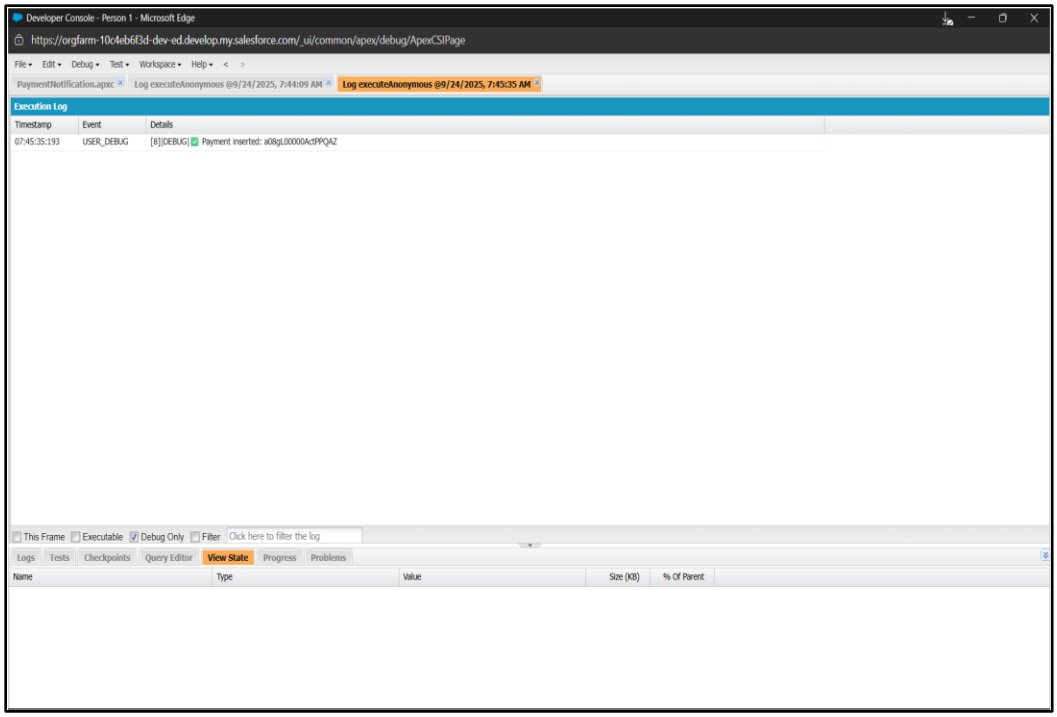
```
try {
   Patients__c p = new Patients__c(
      First_Name__c = 'John'
   );
```

```
    insert p;
 System.debug('✅ Patient inserted: ' + p.Id);
} catch(DmlException e) {
    System.debug('⚠ Error inserting patient: ' + e.getMessage());
}
```

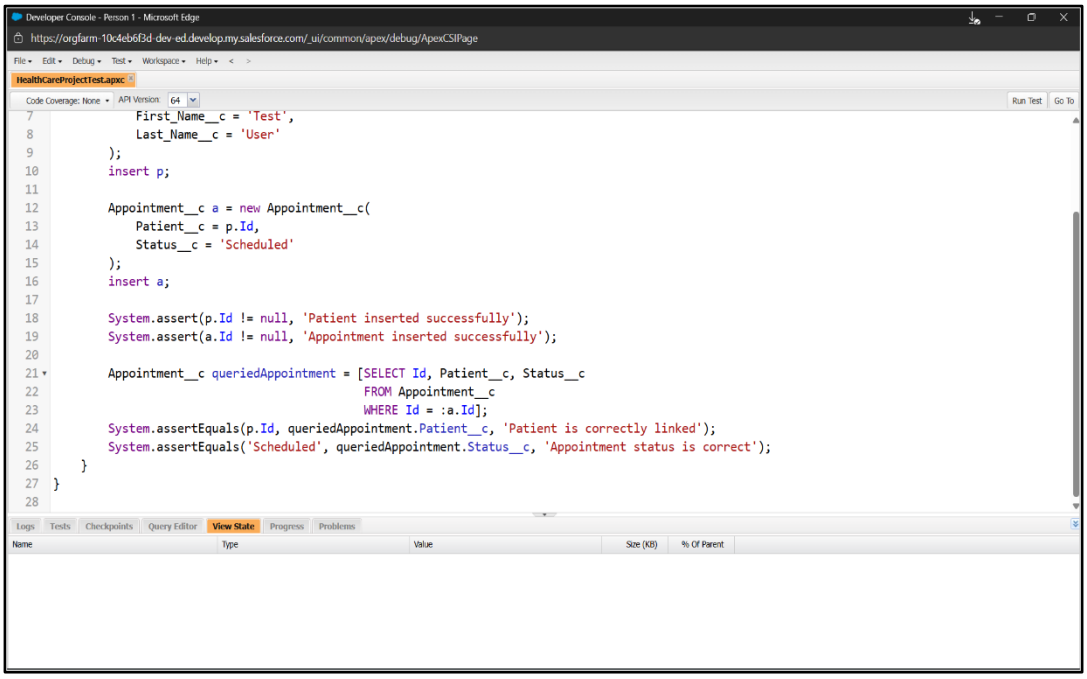


Payments:

```
    try {
         Payments__c pay = new Payments__c(
        Patient__c = null,
        Amount__c = 1000
          );
          insert pay;
        System.debug('✅ Payment inserted: ' + pay.Id);
      } catch(DmlException e) {
    System.debug('⚠ Error inserting payment: ' + e.getMessage());
          }
```



- **Test Classes:**
  - Go to Debug Console.
  - Click on File and New and then select the Execute Anonymous Window.
  - Enter the File Name HealthCareProjectTest
  - Type the code and save it.



- Click on the Test and select the New Run.
- The output will be Success.

Salesforce Project Phase-6

Health Patient Management and Telemedicine CRM

Phase 6: User Interface Development

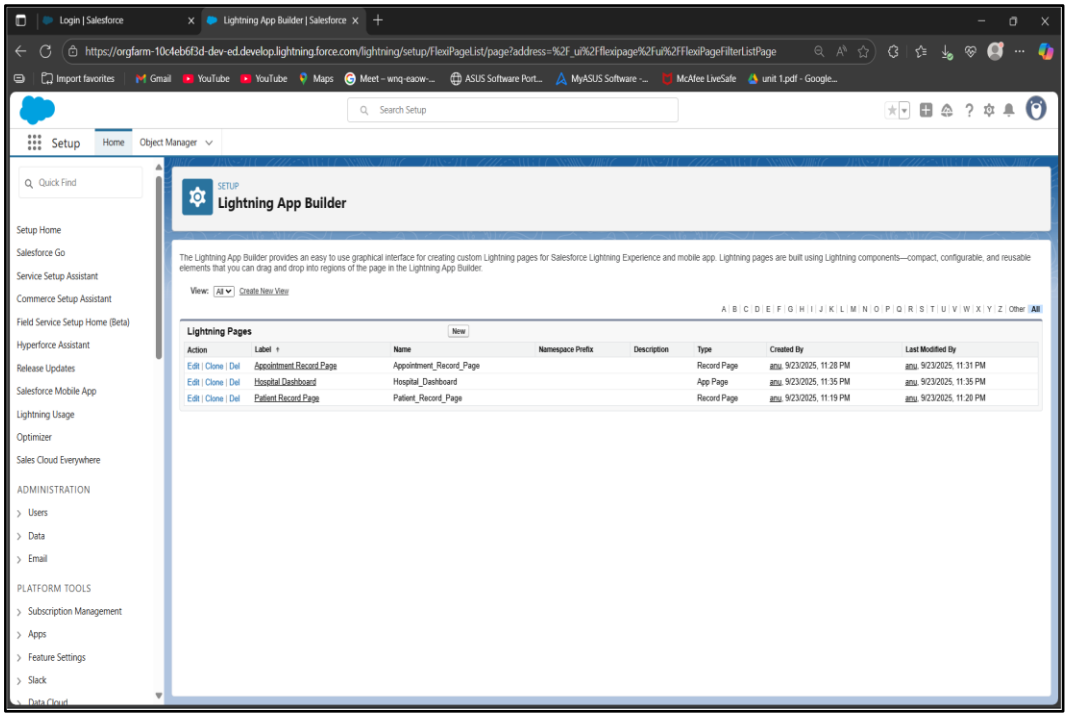
- **Lightning App Builder and Record Pages:**
  - Go to Setup.
  - In the Quick Find box, type the Lightning App Builder and click on it.
  - Click on New and select Record Page.
  - Patient:
    Page Name: Patient Record Page
    Object: Patient _c
    Next
    Choose Grouped header Template
    Drag components into Layout: Record detail, Field Section, Related List – Single – Appointment _c, Related List – Single – Bed Availability _c
    Select Set Dynamic Visibility: Field: Status, Operator: =, Value: Admitted
    Save and then Activate and close

  - Appointment:
    Page Name: Appointment Record Page
    Object: Appointment _c
    Next
    Choose Grouped header Template
    Drag components into Layout: Record detail, Field Section, Related Record
    Save and then Activate and close

  - Hospital Dashboard App Page:
    Instead of selecting Record Page select the App page
    Page Name: Hospital Dashboard
    Next
    Choose 2 Regions
    Drag components into Layout:

1. Left column:
   Report Column, List View
2. Right column:
   List View

Save and then Activate and close

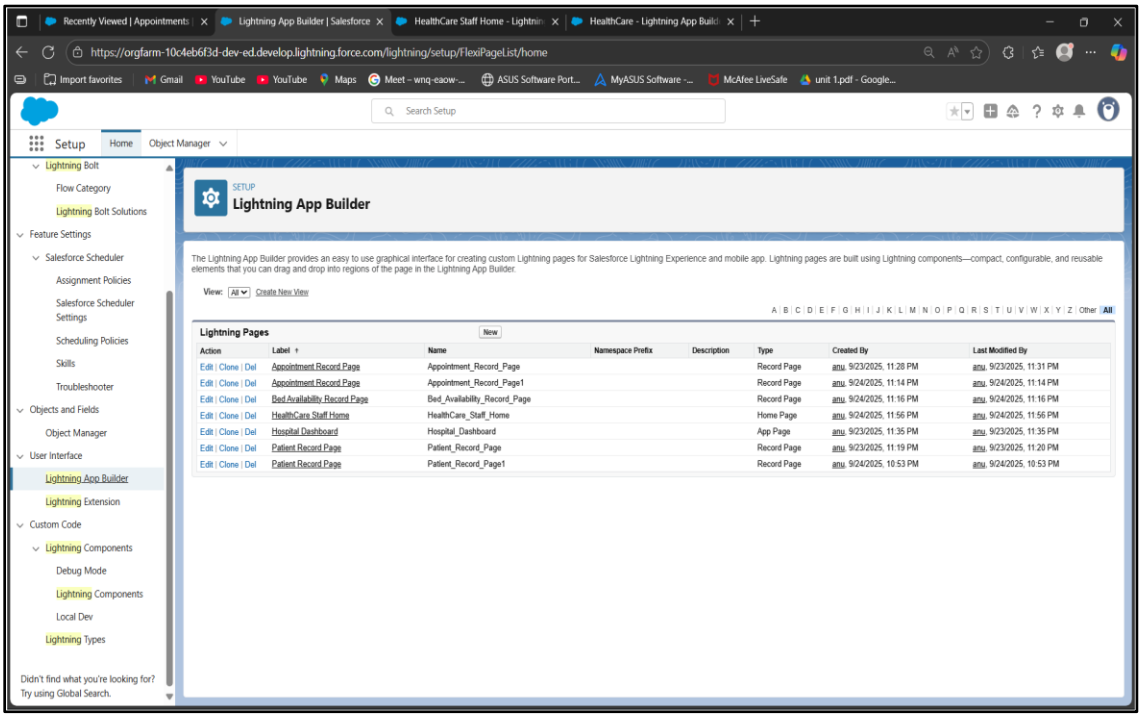


❖ **Tabs:**

- ➢ We already created the required tabs in the previous steps.
- ➢ Now we have to add to the Lightning Builder.
- ➢ Go to the Setup.
- ➢ In the Quick Find box, type the App Manager.
- ➢ Near HealthCare in the drop menu click on Edit.
- ➢ Go to Navigation Items
- ➢ Select the Dashboard to the selected items and arrange them in required way.
- ➢ Click on save.

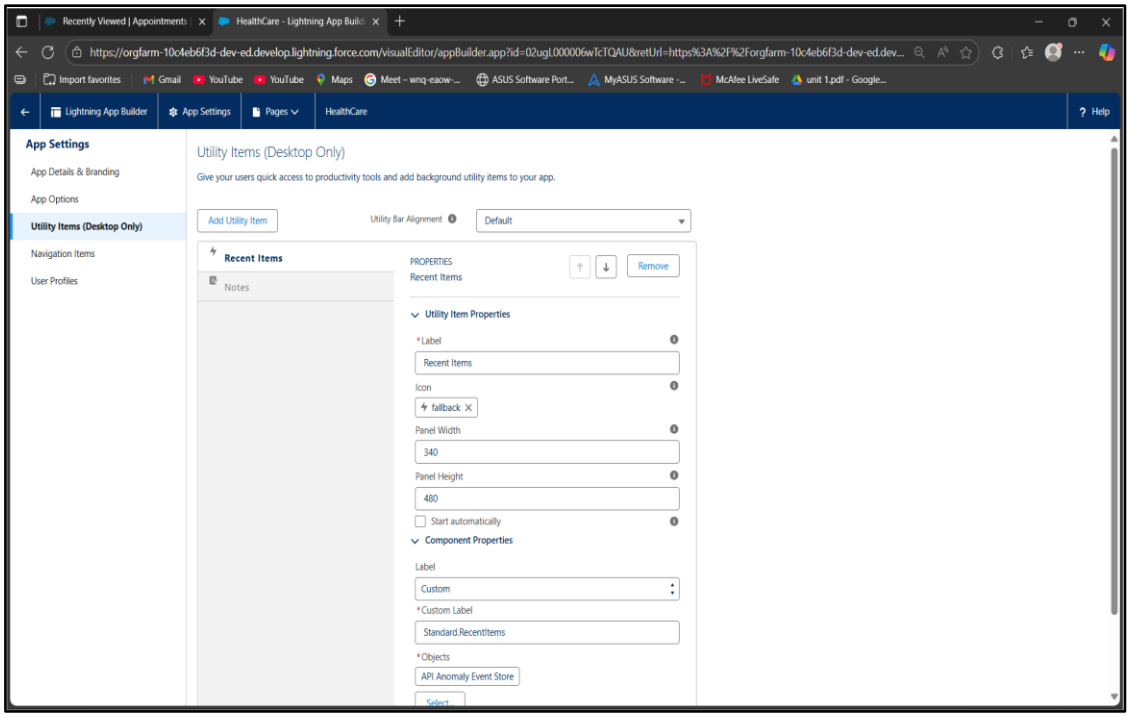
❖ **Home Page Layouts:**

- Go to Setup.
- In the Quick Find box, type the lightning app Builder and click it.
- Click on New.
- Select the Home Page and click Next.
- Label: HealthCare staff Home
- Choose template: Header and three regions
- Click Next.
- Drag the components,
  Left: List View, Middle: Report chart, Right: List View
- Click on save and Activate
- Select the App Default and then Assign Apps
- Select the HealthCare and save.



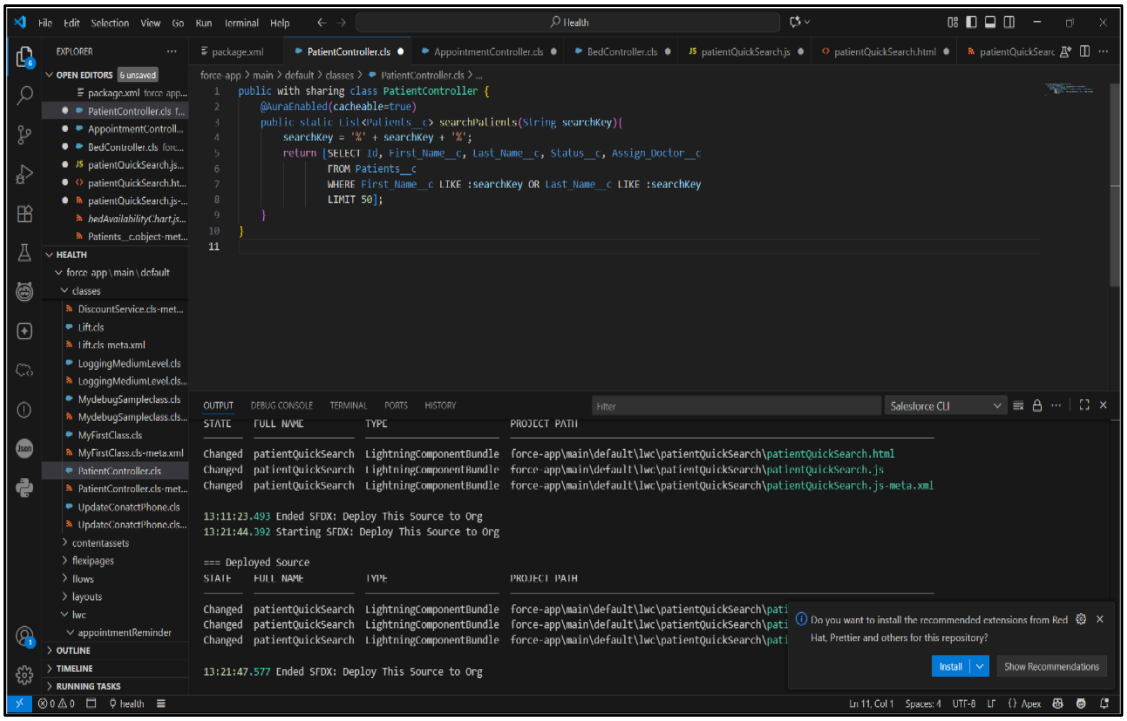
❖ **Utility Bar:**

- Go to Setup.
- In the Quick Find box, type the App Manager and click it.
- Search the HealthCare and click on edit.
- Click the Utility Items.
- Click on Add Utility Items and select the Notes and Recent Items.
- Click on save.

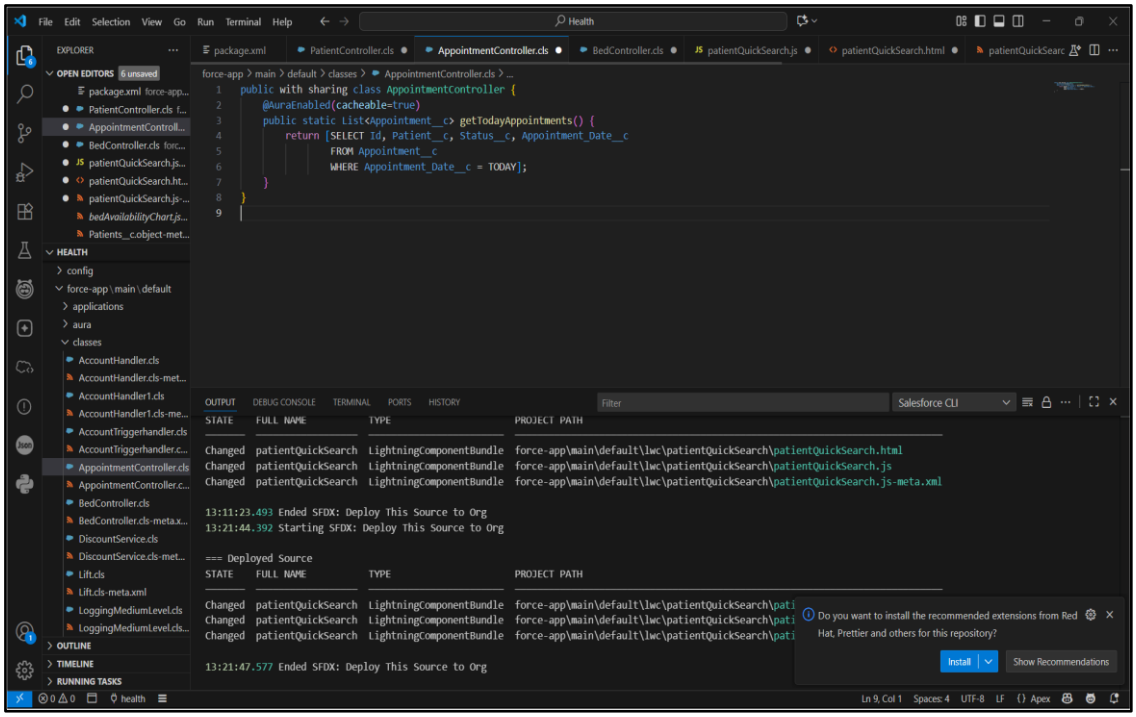


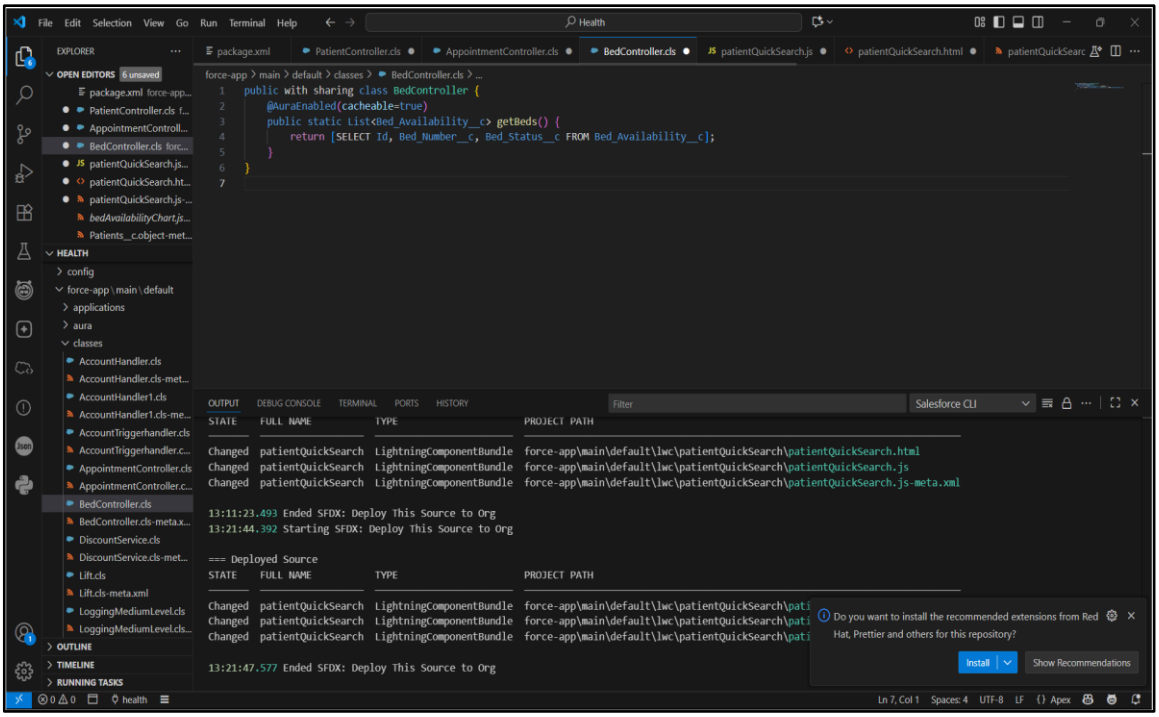
- **LWC (Lightning Web Components):**
  - Open VS code with Salesforce Extension Pack.
  - Press ctrl + shift + P and select the SFDX: Create Lightning Web Component
  - Name it as patientQuickSearch
  - Right click on the file select the SFDX: Deploy this source to org
  - Repeat the steps for AppointmentReminder and BedAvailabilityChart

- **Apex with LWC:**
  - Open the VS code with Salesforce Extension Pack
  - Press ctrl + shift + P and select SFDX: Create Apex class
  - Name the class PatientController
  - Type the code:



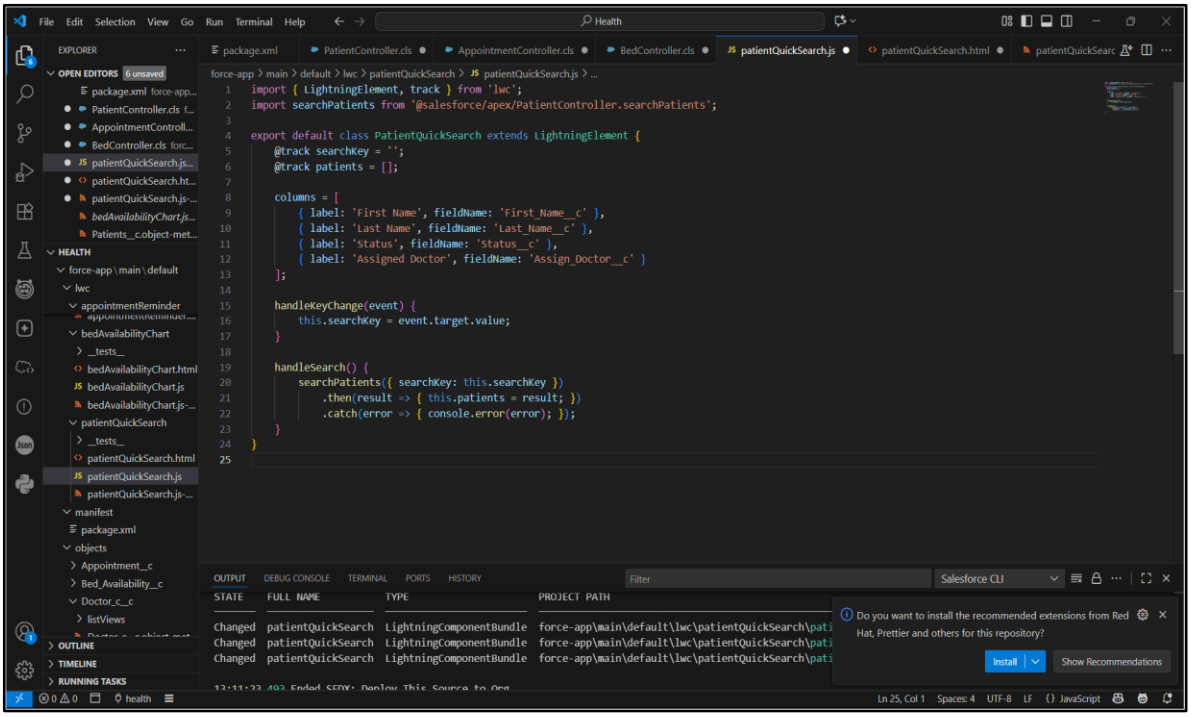
- Right click on the file and Select the SFDX: Depoly this source to Org
- Repeat for the AppointmentController and BedController

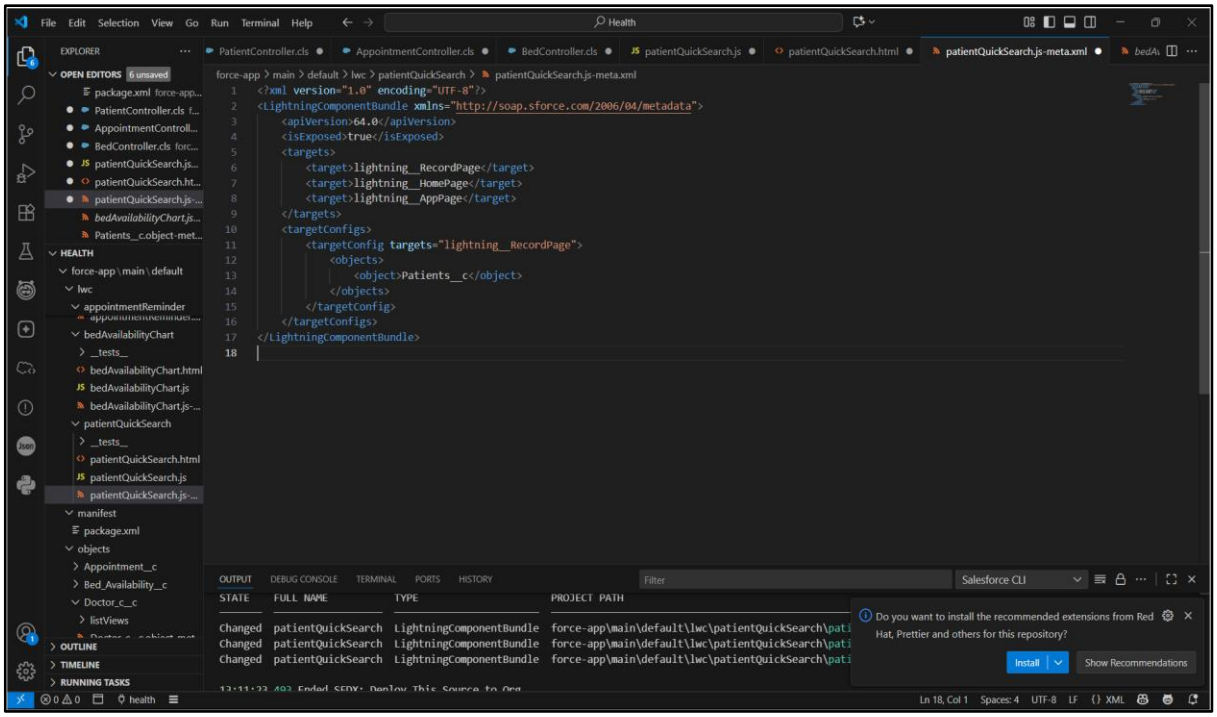


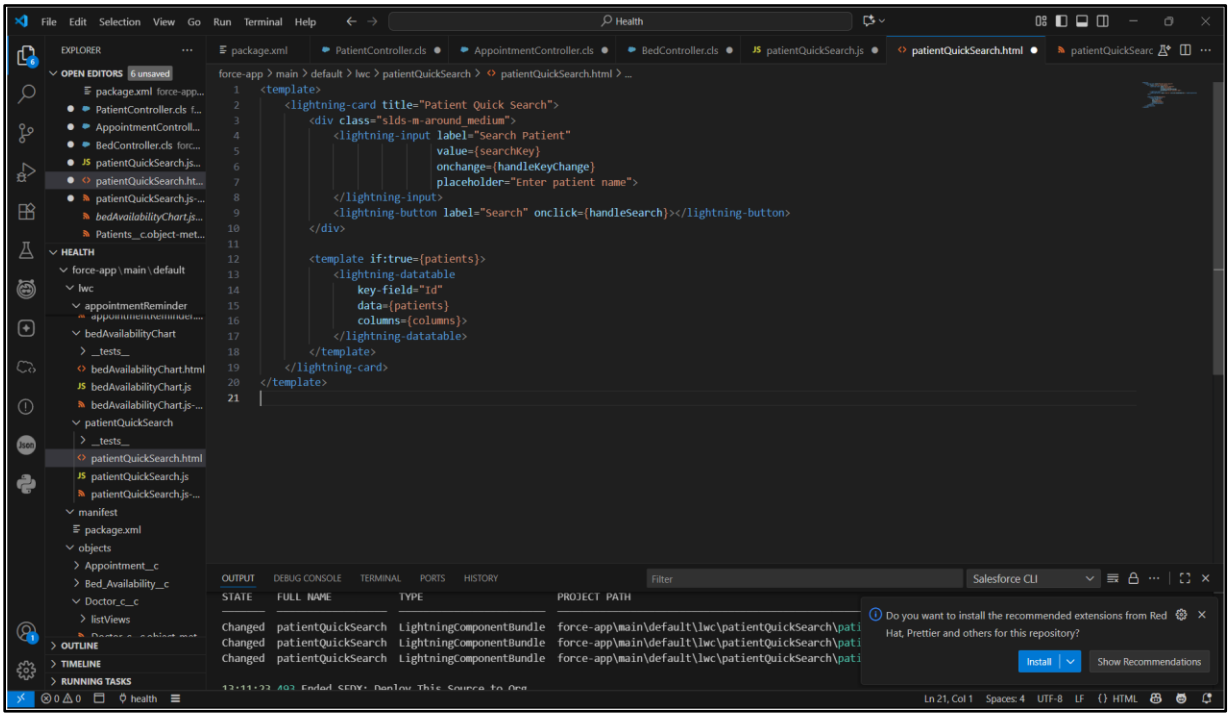


- Connect Apex to the LWC:

  Update the codes for patientQuickSearch.js, patinetQuickSearch.html, patientQuicksearch.js-meta.xml and right click on the file and select the SFDX: Deploy this source to Org




File Edit Selection View Go Run Terminal Help
Health
EXPLORER
OPEN EDITORS 6 unsaved
PatientController.cls
AppointmentController.cls
BedController.cls
patientQuickSearch.js
patientQuickSearch.html
patientQuickSearch.js-meta.xml
force-app > main > default > lwc > patientQuickSearch > patientQuickSearch.js-meta.xml
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
<apiVersion>64.0</apiVersion>
<isExposed>true</isExposed>
<targets>
<target>lightning__RecordPage</target>
<target>lightning__HomePage</target>
<target>lightning__AppPage</target>
</targets>
<targetConfigs>
<targetConfig targets="lightning__RecordPage">
<objects>
<object>Patients__c</object>
</objects>
</targetConfig>
</targetConfigs>
</LightningComponentBundle>
OUTPUT DEBUG CONSOLE TERMINAL PORTS HISTORY
Filter
Salesforce CLI
STATE FULL NAME TYPE PROJECT PATH
Changed patientQuickSearch LightningComponentBundle force-app\main\default\lwc\patientQuickSearch\pati
Changed patientQuickSearch LightningComponentBundle force-app\main\default\lwc\patientQuickSearch\pati
Changed patientQuickSearch LightningComponentBundle force-app\main\default\lwc\patientQuickSearch\pati
Do you want to install the recommended extensions from Red Hat, Prettier and others for this repository?
Install
Show Recommendations
OUTLINE
TIMELINE
RUNNING TASKS
Ln 18, Col 1 Spaces: 4 UTF-8 LF {} XML


File Edit Selection View Go Run Terminal Help
Health
EXPLORER
OPEN EDITORS 6 unsaved
package.xml
PatientController.cls
AppointmentController.cls
BedController.cls
patientQuickSearch.js
patientQuickSearch.html
force-app > main > default > lwc > patientQuickSearch > patientQuickSearch.html > ...
<template>
<lightning-card title="Patient Quick Search">

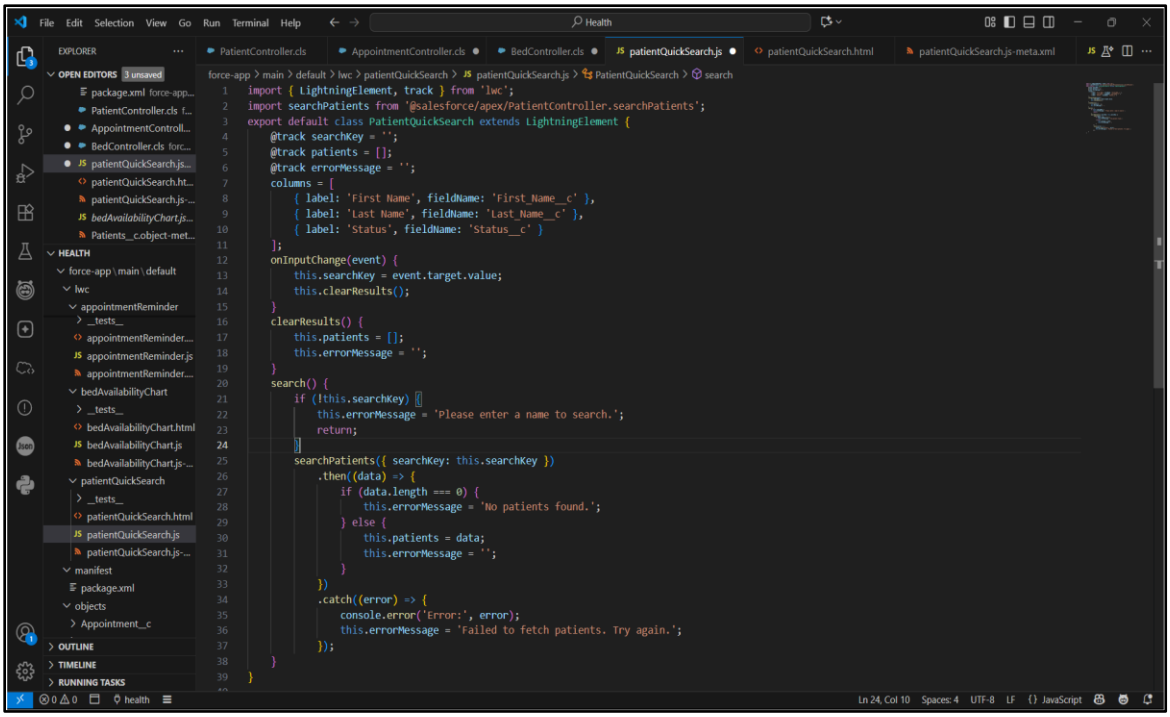
<lightning-input label="Search Patient"
value={searchKey}
onchange={handleKeyChange}
placeholder="Enter patient name">
</lightning-input>
<lightning-button label="Search" onclick={handleSearch}></lightning-button>

<template if:true={patients}>
<lightning-datatable
key-field="Id"
data={patients}
columns={columns}>
</lightning-datatable>
</template>
</lightning-card>
</template>
OUTPUT DEBUG CONSOLE TERMINAL PORTS HISTORY
Filter
Salesforce CLI
STATE FULL NAME TYPE PROJECT PATH
Changed patientQuickSearch LightningComponentBundle force-app\main\default\lwc\patientQuickSearch\pati
Changed patientQuickSearch LightningComponentBundle force-app\main\default\lwc\patientQuickSearch\pati
Changed patientQuickSearch LightningComponentBundle force-app\main\default\lwc\patientQuickSearch\pati
Do you want to install the recommended extensions from Red Hat, Prettier and others for this repository?
Install
Show Recommendations
OUTLINE
TIMELINE
RUNNING TASKS
Ln 21, Col 1 Spaces: 4 UTF-8 LF {} HTML

❖ **Events in LWC and Wire Adapters:**
These are already done no need do now.

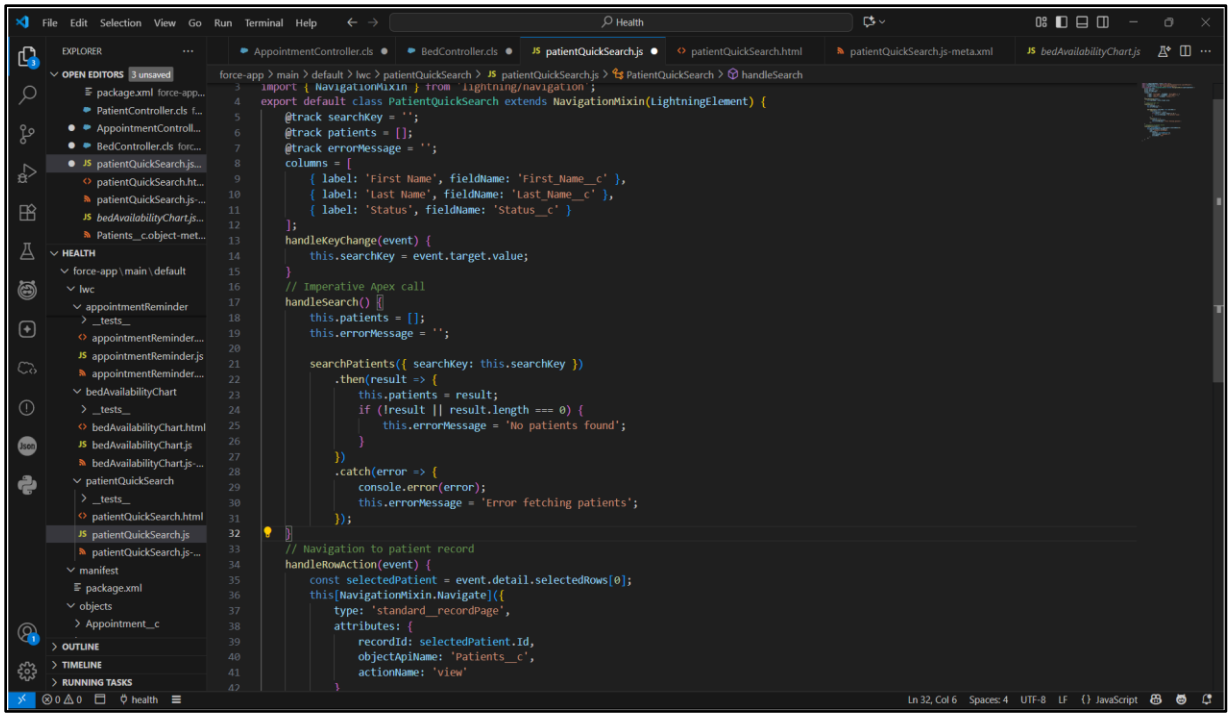
❖ **Imperative Apex Calls:**

- ➢ Update the patientQuickSearch.js code in VS code
- ➢ Right click on the code and select the Deploy this source to Org



❖ **Navigation Service:**

- ➢ Update the patientQuickSearch.js code in VS code
- ➢ After updating right click on the file and select SFDX: Deploy this source to Org
- ➢ Wait until the successfully message done

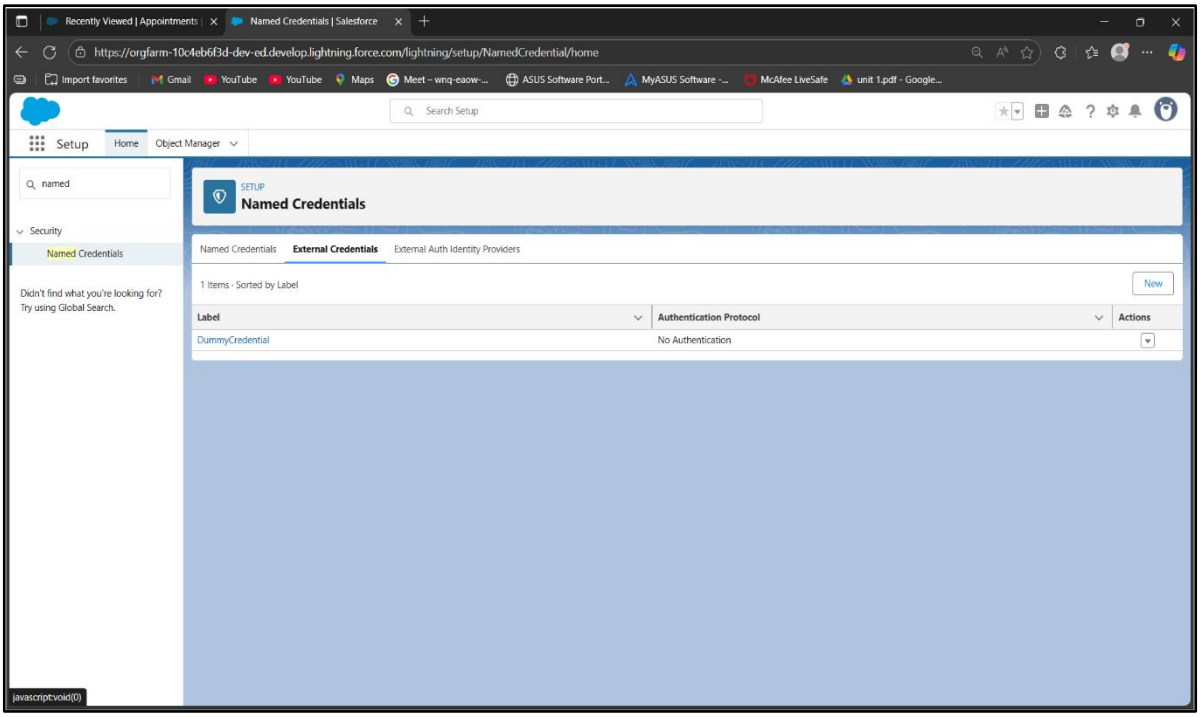

File Edit Selection View Go Run Terminal Help
Health
EXPLORER
OPEN EDITORS 3 unsaved
HEALTH
force-app \ main \ default
lwc
appointmentReminder
bedAvailabilityChart
patientQuickSearch
manifest
objects
OUTLINE
TIMELINE
RUNNING TASKS
AppointmentController.cls
BedController.cls
patientQuickSearch.js
patientQuickSearch.html
patientQuickSearch.js-meta.xml
bedAvailabilityChart.js
force-app > main > default > lwc > patientQuickSearch > patientQuickSearch.js > PatientQuickSearch > handleSearch
import { NavigationMixin } from 'lightning/navigation';
export default class PatientQuickSearch extends NavigationMixin(LightningElement) {
@track searchKey = '';
@track patients = [];
@track errorMessage = '';
columns = [
{ label: 'First Name', fieldName: 'First_Name__c' },
{ label: 'Last Name', fieldName: 'Last_Name__c' },
{ label: 'Status', fieldName: 'Status__c' }
];
handleKeyChange(event) {
this.searchKey = event.target.value;
}
// Imperative Apex call
handleSearch() {
this.patients = [];
this.errorMessage = '';
searchPatients({ searchKey: this.searchKey })
.then(result => {
this.patients = result;
if (!result || result.length === 0) {
this.errorMessage = 'No patients found';
}
})
.catch(error => {
console.error(error);
this.errorMessage = 'Error fetching patients';
});
}
// Navigation to patient record
handleRowAction(event) {
const selectedPatient = event.detail.selectedRows[0];
this[NavigationMixin.Navigate]({
type: 'standard__recordPage',
attributes: {
recordId: selectedPatient.Id,
objectApiName: 'Patients__c',
actionName: 'view'
}
Ln 32, Col 6 Spaces: 4 UTF-8 LF JavaScript

Salesforce Project Phase-7

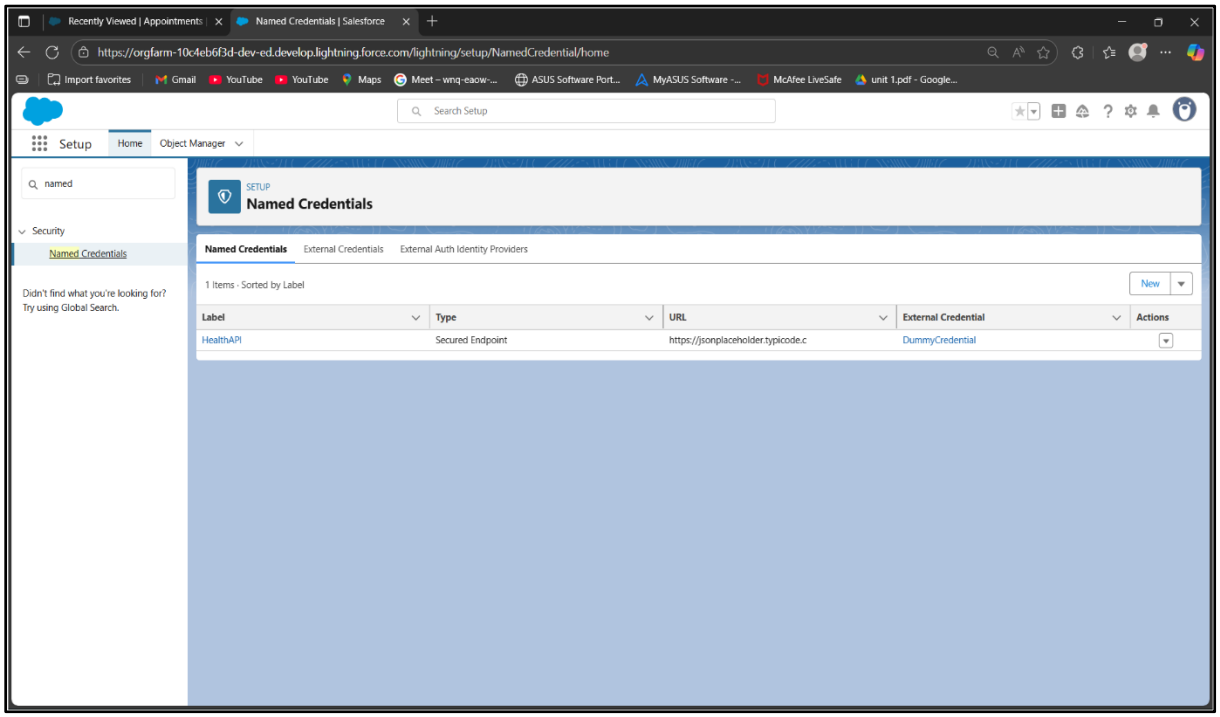
Health Patient Management and Telemedicine CRM

Phase 7: Integration & External Access

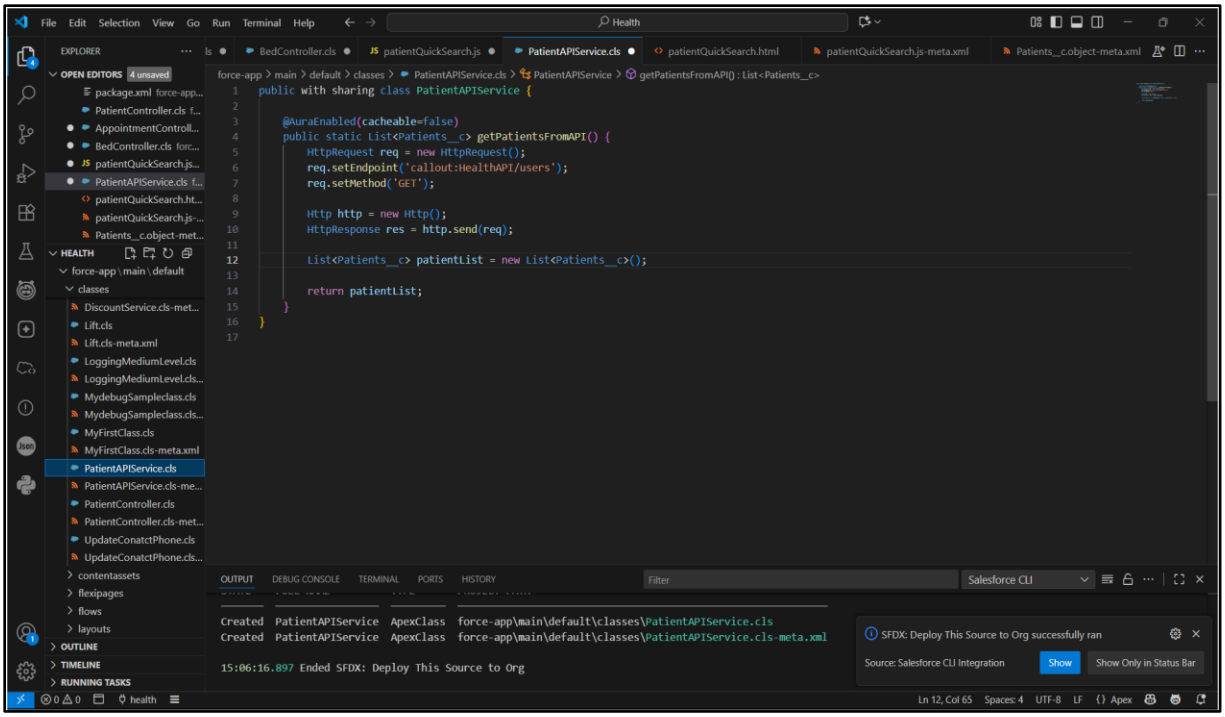
- **Named Credentials:**
  - Go to the Setup.
  - In the Quick Find box, type the Named Credentials.
  - Before creating the Named Credential you need to create the External Credential.
  - Click on the External Credential and click on New
  - Fill the details,
    Label and Name: DummyCredential
    Authentication Protocol: No Authentication
  - Click Save.



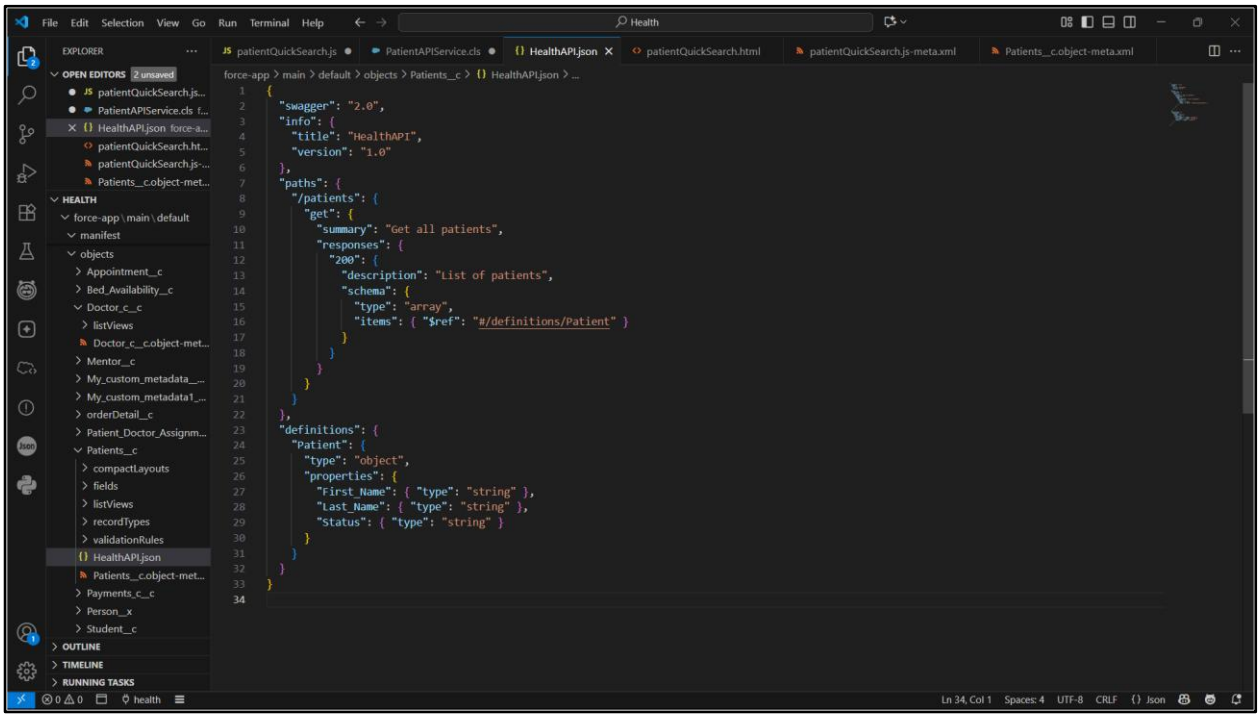
- Now, click on the Named Credential and click on New
- Fill the details:
  Label and Name: HealthAPI
  URL: https://jsonplaceholder.typicode.com
  External Credential: DummyCredential
  Generate Authorization Header leave unchecked
- Click on Save



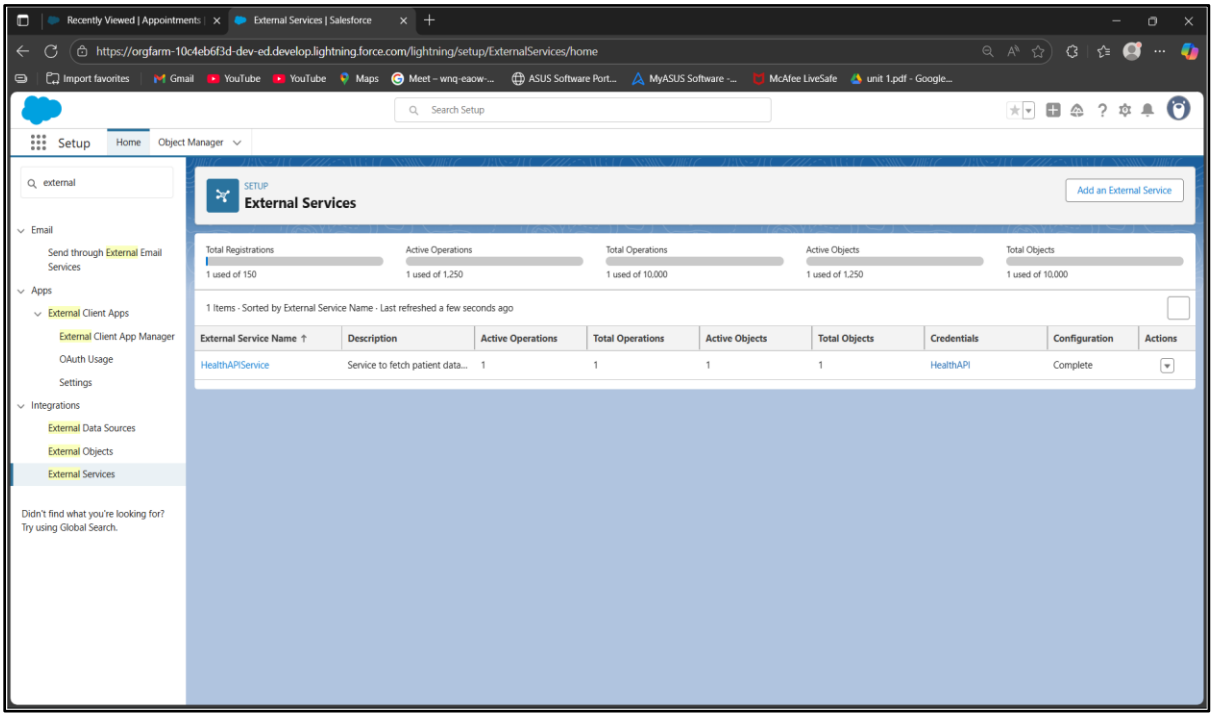
- Now create the Apex class with named PatientAPIService
- Type the code and save.
- Right click on the file and select the SFDX: Deploy this source to Org
- Wait until the successful message is displayed.



- **External Services:**
  - Open VS Code.
  - Create a HealthAPI.json file in patient object.



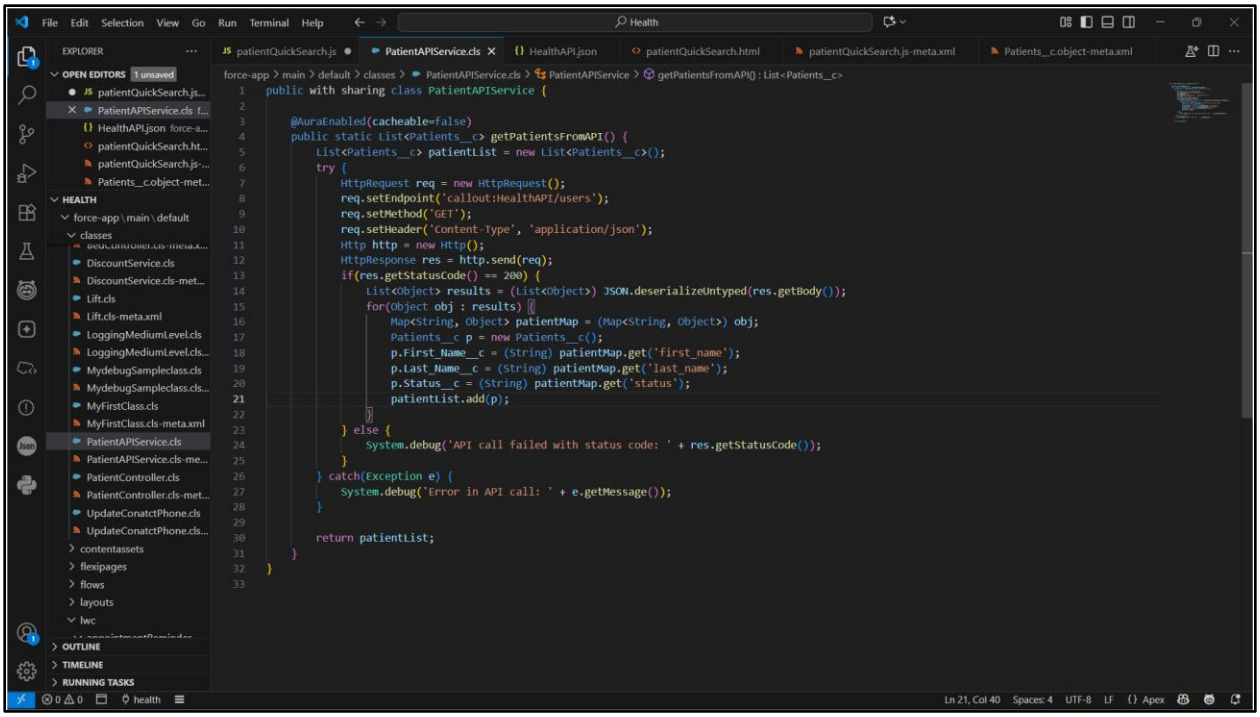
- Go to Setup.
- In the Quick Find box, type the External Services and click it.
- Now, click on the Add an External Service.
- Select the From API Specification and click on Next.
- Fill the details:
  External Service Name: HealthAPIService
  Description: Service to fetch patient data from exter Health API
  Service Schema: Upload the created JSON file
  Relative URL: Leave blank
  Named Credential: HealthAPI
- Click on save
- Select the GetPatients and click Next.
- Click on Finish.



❖ **Web Services (REST/SOAP):**
Already done this we need to do this.

❖ **Callouts:**

- Open the VS Code.
- Open the patientAPIService file and update the code as required
- Right click on the file and select the SFDX: Deploy this Source to Org




```
public with sharing class PatientAPIService {

    @AuraEnabled(cacheable=false)
    public static List<Patients__c> getPatientsFromAPI() {
        List<Patients__c> patientList = new List<Patients__c>();
        try {
            HttpRequest req = new HttpRequest();
            req.setEndpoint('callout:HealthAPI/users');
            req.setMethod('GET');
            req.setHeader('Content-Type', 'application/json');
            Http http = new Http();
            HttpResponse res = http.send(req);
            if(res.getStatusCode() == 200) {
                List<Object> results = (List<Object>) JSON.deserializeUntyped(res.getBody());
                for(Object obj : results) {
                    Map<String, Object> patientMap = (Map<String, Object>) obj;
                    Patients__c p = new Patients__c();
                    p.First_Name__c = (String) patientMap.get('first_name');
                    p.Last_Name__c = (String) patientMap.get('last_name');
                    p.Status__c = (String) patientMap.get('status');
                    patientList.add(p);
                }
            } else {
                System.debug('API call failed with status code: ' + res.getStatusCode());
            }
        } catch(Exception e) {
            System.debug('Error in API call: ' + e.getMessage());
        }

        return patientList;
    }
}
```

- **Platform Events:**
  - Go to the Setup.
  - In the Quick Find box, type the Platform Event and click it.
  - Now click on the New Platform Event
  - Fill the details:
    Label: PatientUpdateEvent
    Plural Label: PatientUpdateEvent
    Publish Behaviour: Publish After Commit
  - Click Save
  - Scroll down to see the Custom Fields and Relationships
  - Click New
  - Select the Text, Text and Text Area(Long) respectively
  - Fill the details:

    Patients
    Field Label: PatientId_c
    Length: 18

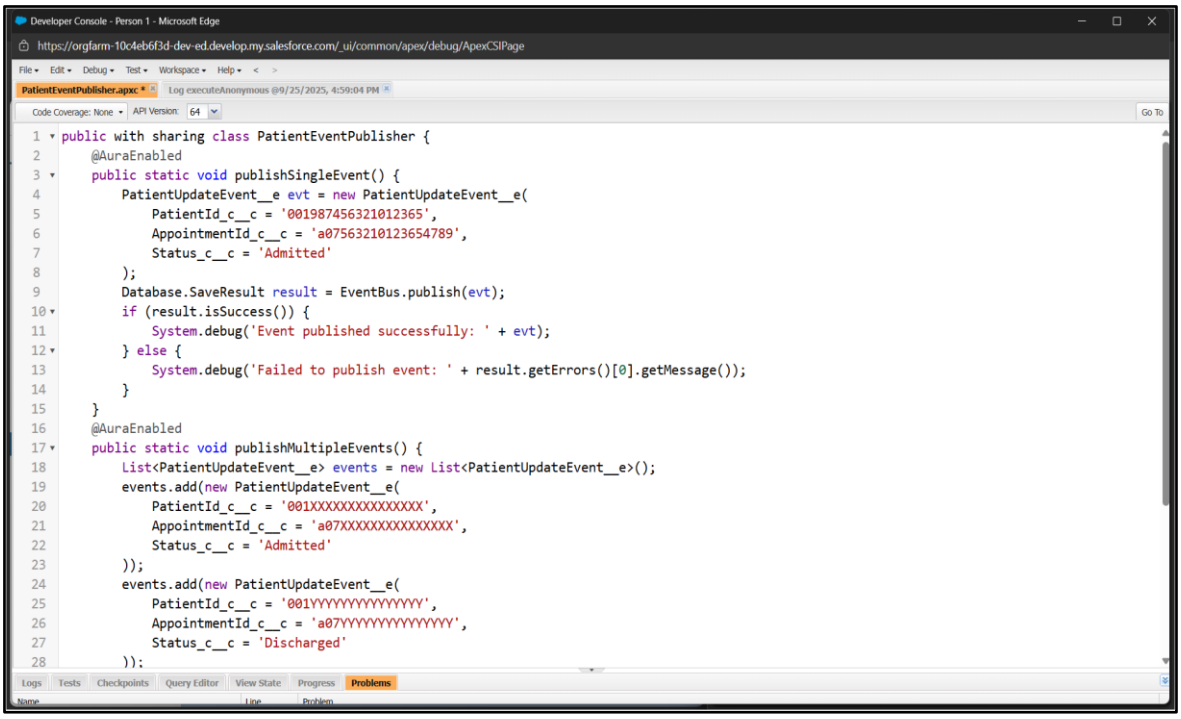
    Appointments
    Field Label: AppointmentId_c
    Length: 50

    Status

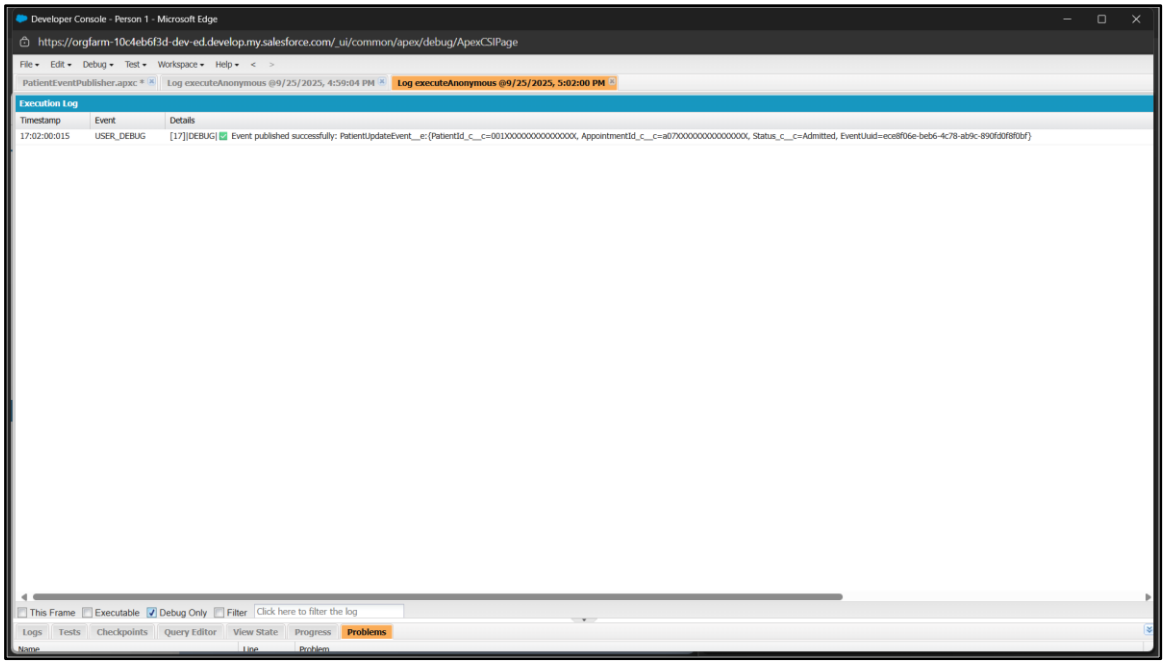
Field Label: Status_c

Length: 32768

- Click Save
- Now go to Developer Console
- Click on File and select New Apex class
- Enter File Name PatientEventPublisher
- Type the code and save it

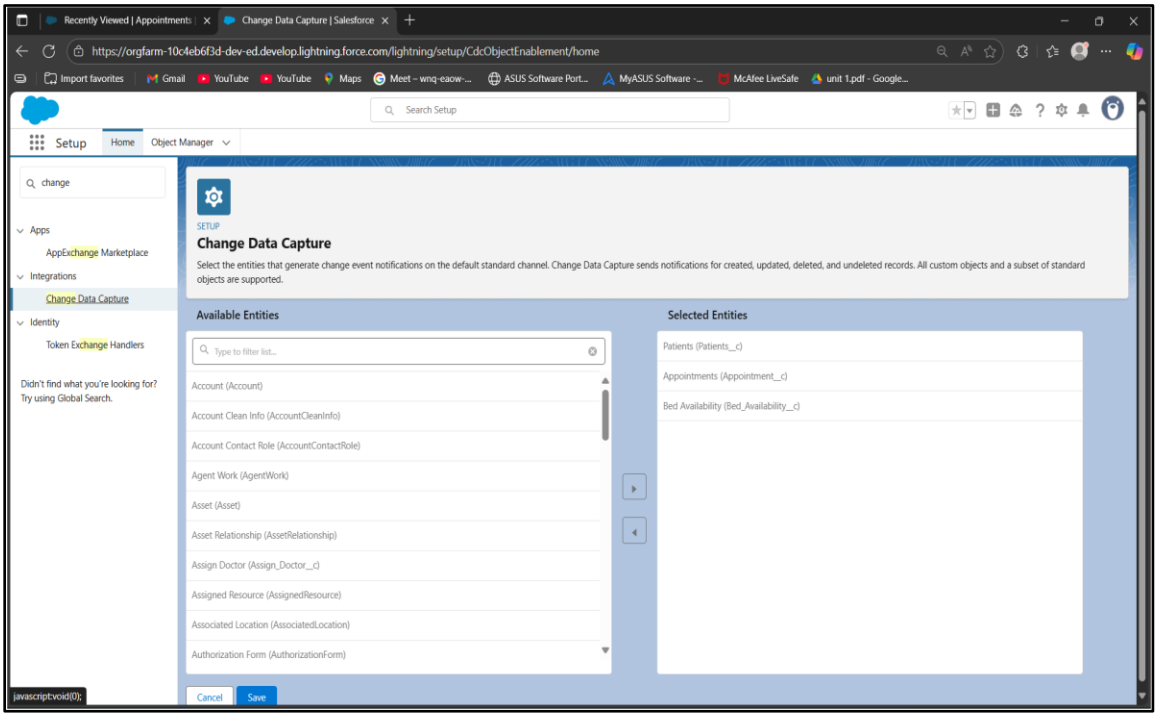


- Now click the Debug and select the Execute the Anonymous window
- PatientEventPublisher.publishSingleEvent();
- Now Execute

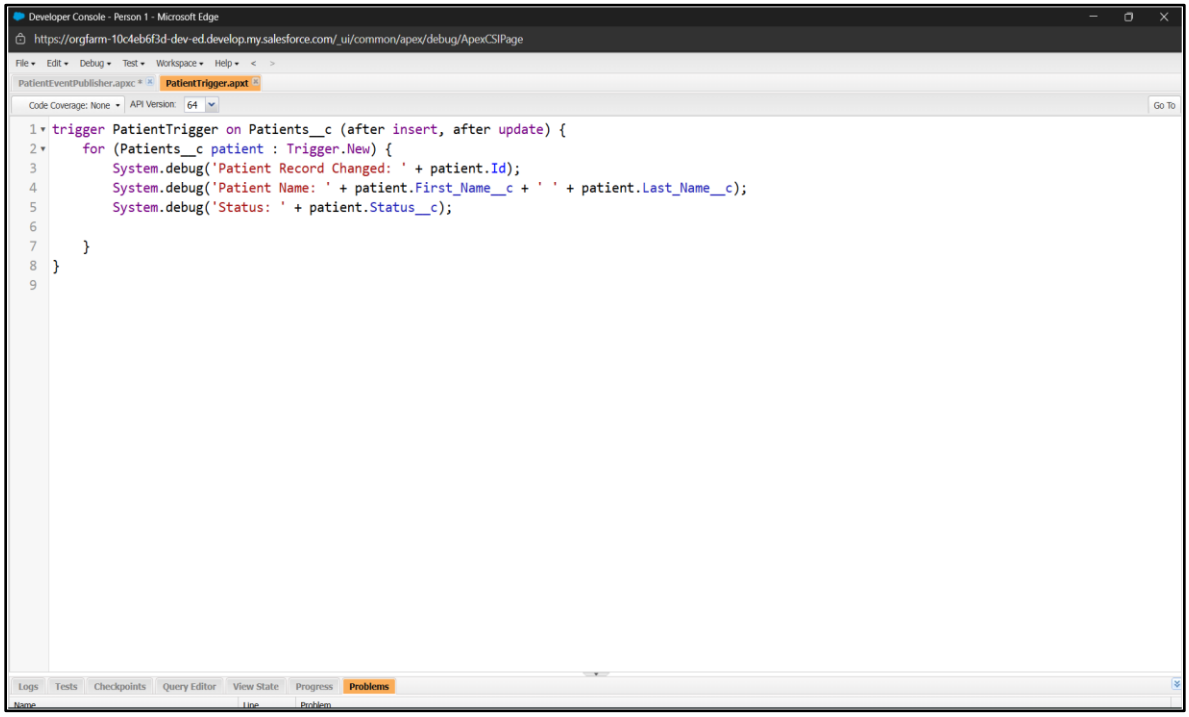


❖ Change Data Capture:

- ➢ Go to the Setup.
- ➢ In the Quick Find box, type the Change Data Capture and click it.
- ➢ From the list, select Patients_c, Appointment_c, Bed_Availability_c
- ➢ Click save.

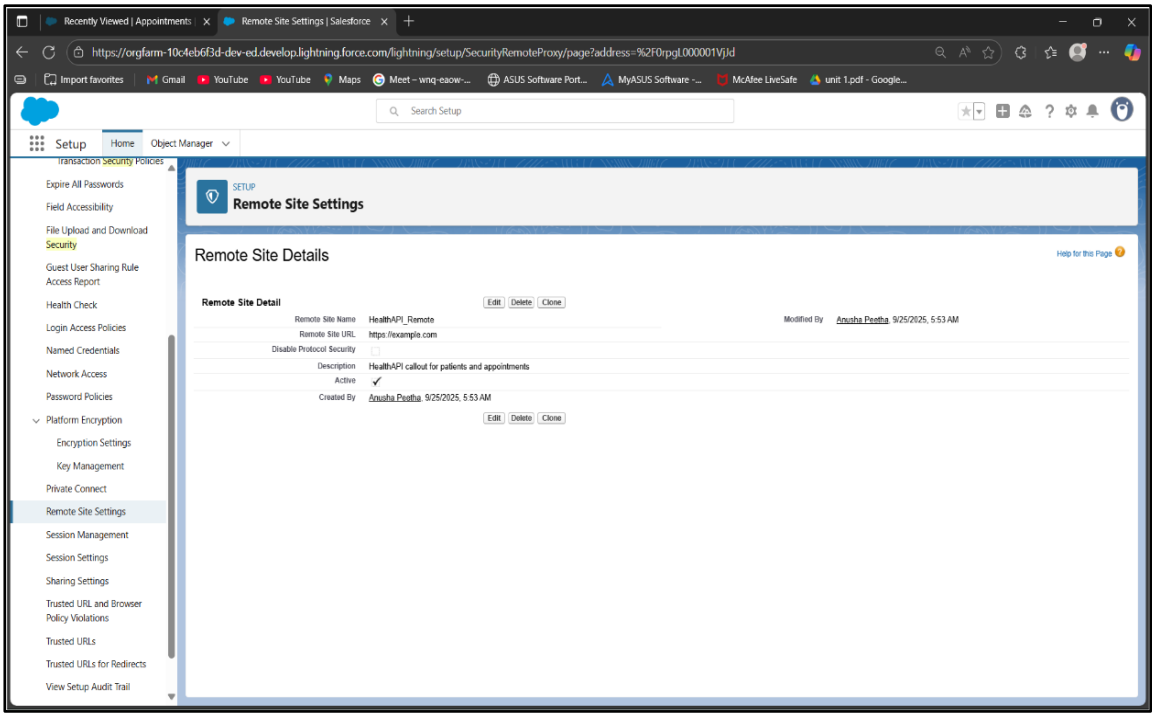


- Now go to Develop Console.
- Click the File and select New Apex trigger.
- Name the trigger as PatientCDCTrigger and select sObject Patient_c
- Type the code and save it.

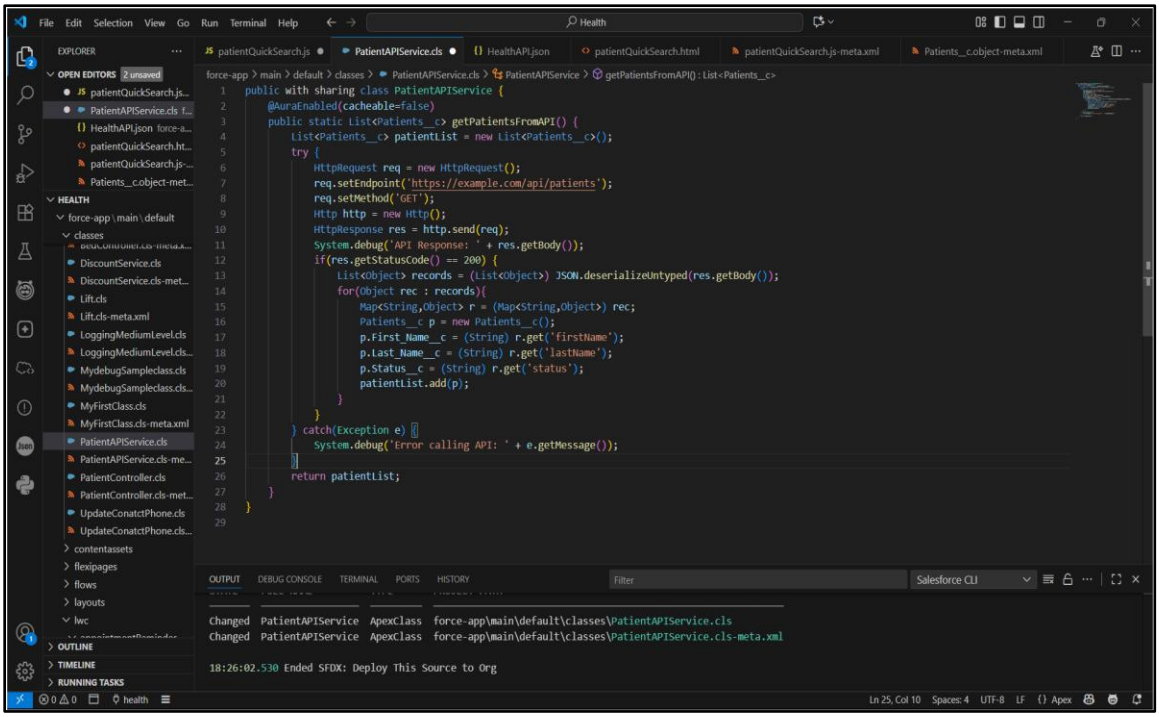


❖ **Remote site settings:**

- Go to Setup.
- In the Quick Find box, type the Remote site settings and click it.
- Click on the New Remote Site.
- Fill the details:
  Remote site Name: HealthAPI_Remote
  Remote Site URL: https://example.com/api
  Description: HealthAPI callout for patients and appointments
  Enable Active
- Click Save.

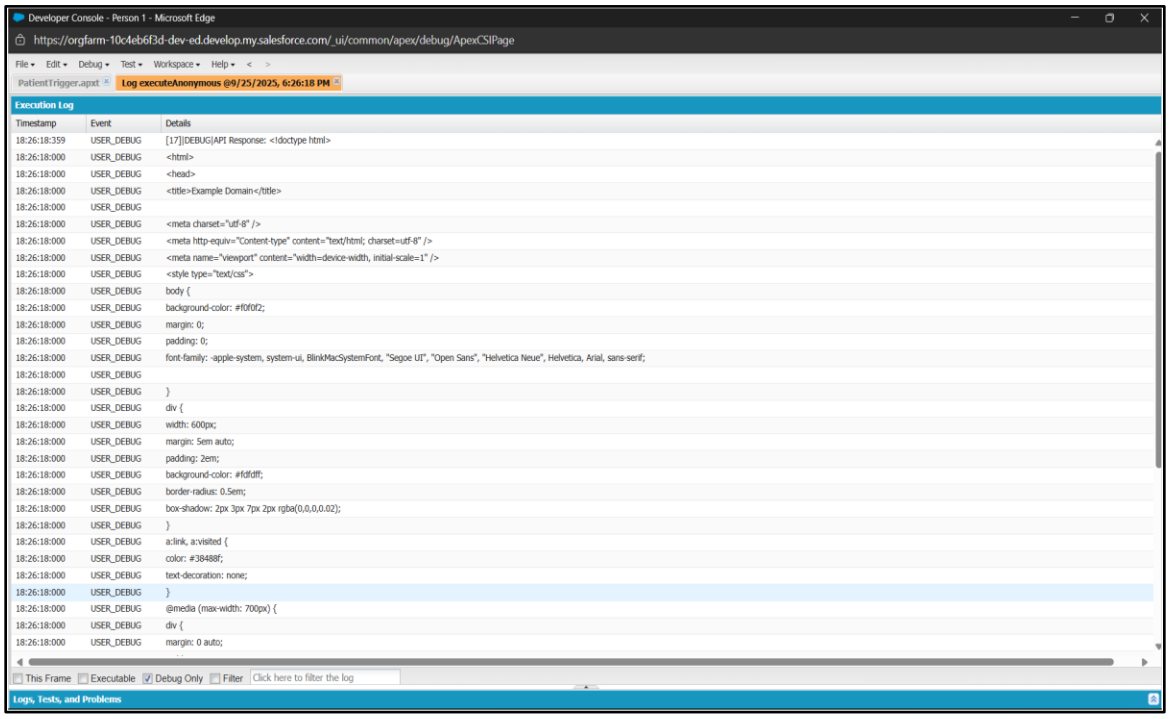


- Open VS Code.
- Open the PatientAPIService and update the code.
- After updating right click on the File Name and select the SFDX: Deploy this source to Org



- Now click on the Debug and select the Execute Anonymous window.
- Type the code,

```
List<Patients__c> patients = PatientAPIService.getPatientsFromAPI();
System.debug(patients);
```

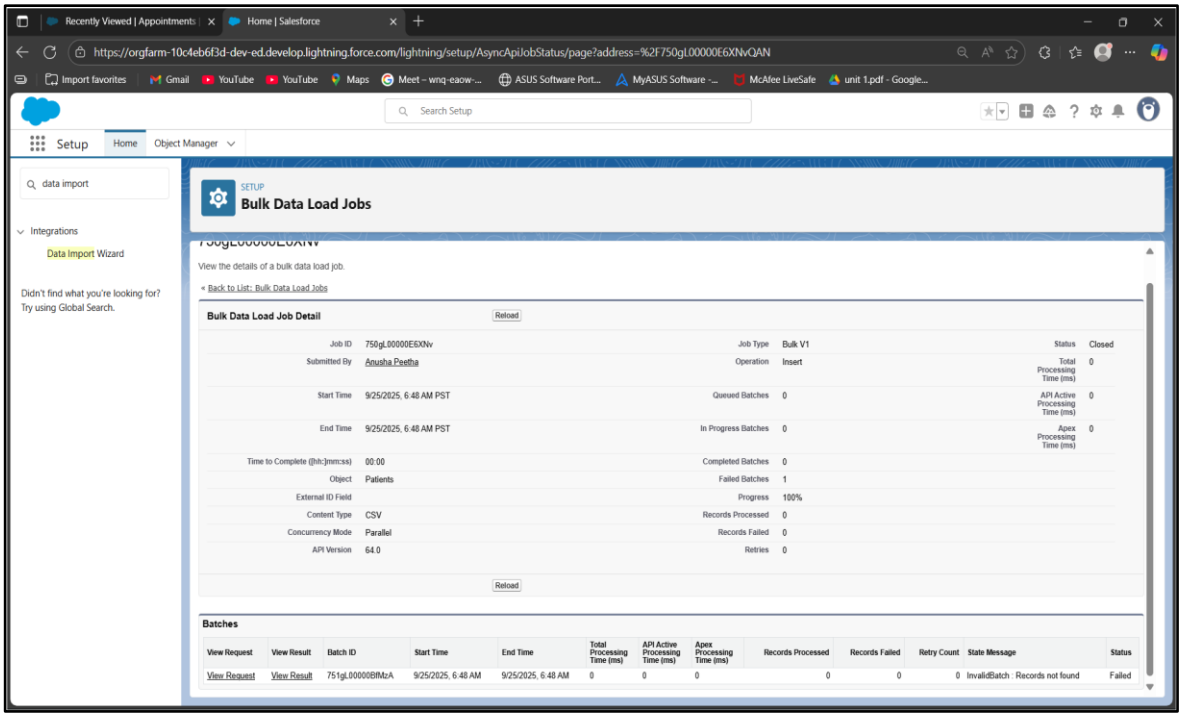


Salesforce Project Phase-8

Health Patient Management and Telemedicine CRM

Phase 8: Data Management & Deployment

- **Data Import Wizard:**
  - Go to the Google sheets or Microsoft Excel.
  - Create the patient csv file with the existing details
  - Save the file.
  - Now go to the Setup.
  - In the Quick Find box, type the Data Import Wizard and click it.
  - Click on the Launch Wizard.
  - Select the Object called Patients.
  - Click choose csv file
  - Click Next.
  - The Salesforce will auto-map csv columns if it shows unmapped click on the Map and select the required fields
  - Click on save and then finish.



- **Data Loader:**
  - Go to Setup.
  - In the Quick Find box, type Data Loader and click it.
  - Click on Downloads and click on Confirm
  - Launch the install.bat
  - Click on Allow
  - In command prompt if asked yes/no type yes
  - Choose operation insert and choose object patient
  - Browse the existing CSV file
  - Create on Create or Edit Map and then click ok
  - Click Next and Finish

- **Duplicate Rules:**
  - Go to the Setup.
  - In the Quick Find box, type the Duplicate rules and click it.
  - Click on New Rule
  - Select the Patinets Object
  - Fill the details:

    Rule Name: Patients Dupliacte Rule

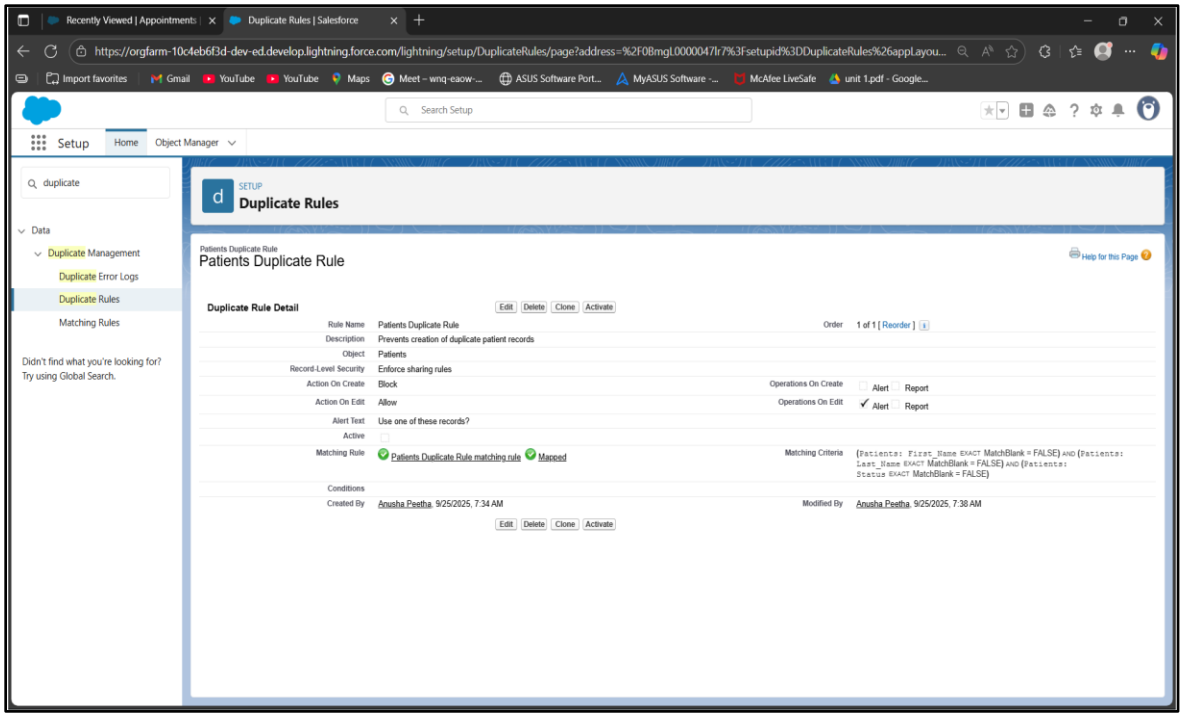
    Description: Prevents craetion of duplicate patient records

    Enable the Enforce sharing rules

    Select the Matching Rule, if don't have the matching rule click on the New Matching Rule.

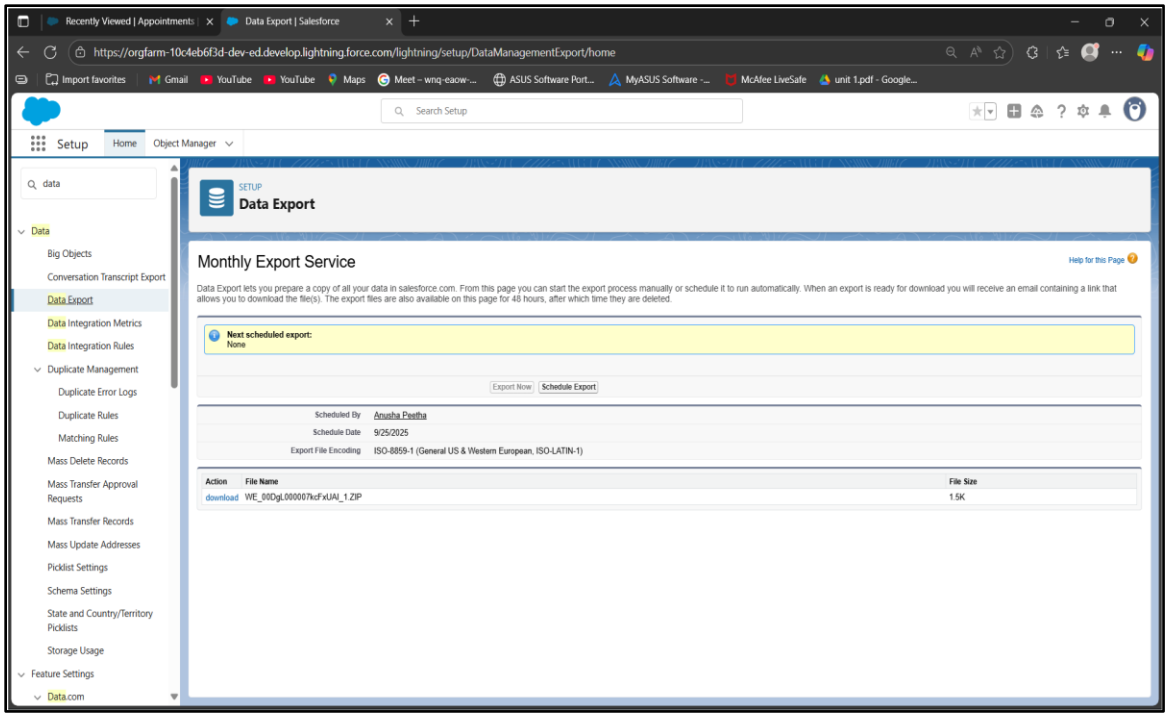
Define Matching Rule: Field: First_Name, Method: Exact
Field: Last_Name, Method: Exact
Field: Status, Method: Exact

- Click on save and activate.



❖ Data Export & Backup:

- Go to the Setup.
- In the Quick Find box, type the Data and click the Data Export.
- Click on the Export Now.
- Select the Patients and Appointments Object.
- Click on the Start Export.
- Now it will Scheduled to give it will notify through the Email.
- Click Download next to completed export.
- Extract the .zip file.
- Save the exported CSV files in a secure location.
- Now you have a backup of all patients and appointment records.



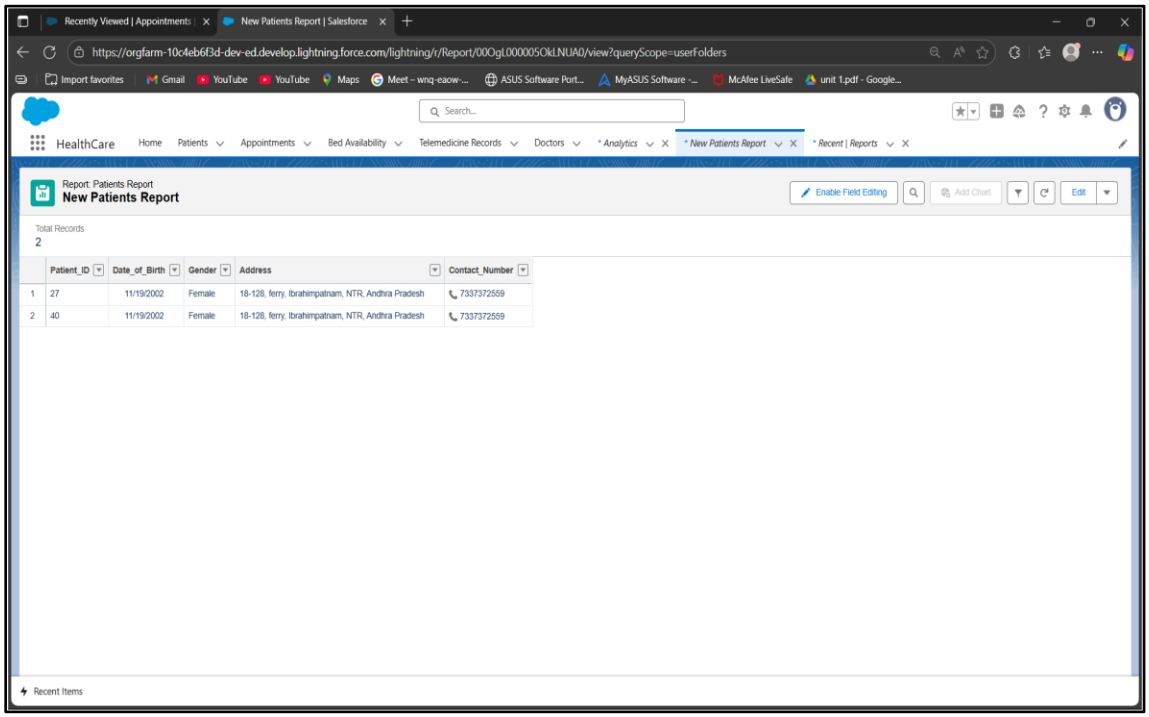
- **VS Code & SFDX:**
  - Install the VS Code: https://code.visualstudio.com/
  - Intall the Salesforce CLI: https://developer.salesforce.com/tools/sfdxcli
  - In VS Code install the Salesforce Extension Pack in the Extensions
  - Setup the project:
    - Open VS Code
    - Press the ctrl + shift + p and type SFDX: Create Project
    - Choose Standard Project
    - Enter the project name: Health
  - Connect the Salesforce Org:
    - Press ctrl + shift + p and type SFDX: Authorize an Org
    - Choose Production
    - Log in to the Org in the browser
  - Retrieve Metadata:
    - Press ctrl + shift +p and type SFDX: Retrieve source from Org
  - Deploy Metadata to Org:
    - Press ctrl + shift +p and type SFDX: Deploy this source to Org

Salesforce Project Phase-9

Health Patient Management and Telemedicine CRM

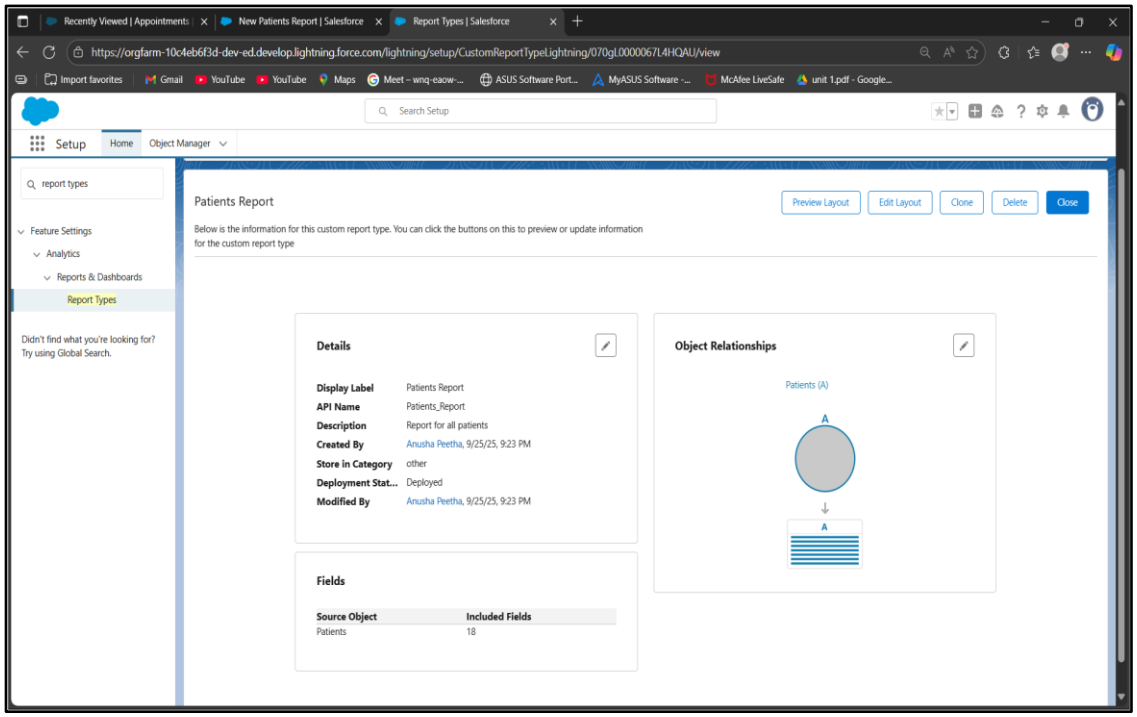
Phase 9: Reporting, Dashboards & Security Review

- **Reports (Tabular, Summary, Matrix, Joined):**
  - Go to Setup.
  - In the App Launcher type the Reports and click it.
  - Click on create.
  - Select the Patient Object.
  - In the Outline, select the required rows and columns
  - In the filter, give the required conditions
  - Click on Save & Run



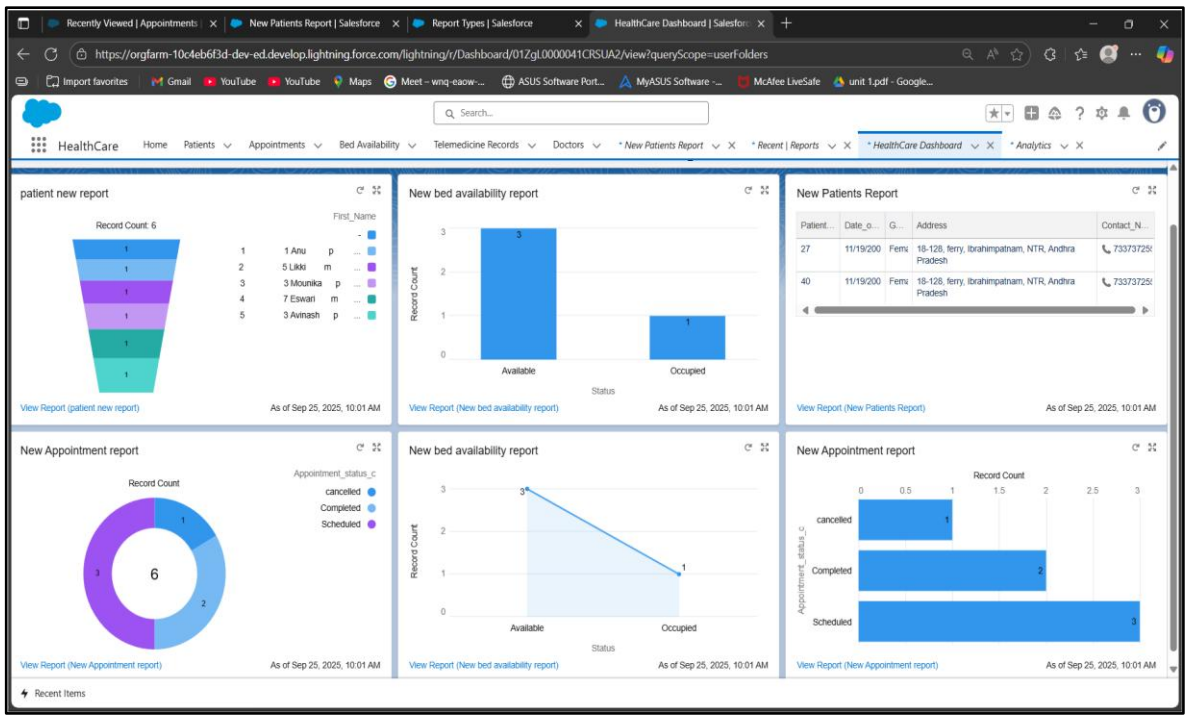
❖ **Report Types:**

- Go to Setup.
- In the Quick Find box, type the Report Types and click it.
- Click on New Custom Report Type.
- Primary Object: Patients
- Report Type Label: Patients Report
- Click to relate another object Appointments
- Choose relationship type, each patient may or may not have related appointments
- Click Save



❖ **Dashboards:**

- Go to Setup
- Click on App Launcher, type the Dashboards and click it
- Click on New Dashboard
- Name it as HealthCare Dashboard
- Choose the Private folder and click Next
- Click on +Widget and select the required reports
- Now arrange them in required format
- Click on Save and then Done.



- **Dynamic Dashboards:**
  - Go to Setup
  - In the Quick Find box, type the Dashboards and click it
  - Click on New Dashboard
  - Name it as HealthCare dynamic dashboard
  - Choose the Public folder and Click Next
  - Click on +Widget and select required reports and click on Add
  - In the top right click on Gear icon and select the View dashboard as The dashboard Viewer and Save.
  - Click Save and Done.

- **Session Settings:**
  - Go to Setup
  - In the Quick Find box, type the security and then select the Session Settings
  - Check:
    Timeout Value: 2 hours
    Enable force logout on Session Timeout
  - Save changes

- **Login IP Ranges:**
  - Go to Setup
  - In the Quick Find box, type the Profiles and them Select the Doctor profile

- Scroll down to see Login IP Ranges and click on New
- Fill the details:
  StartIp: 49.37.130.127
  EndIp: 49.37.130.127
- Click on Save

❖ **Audit Trail:**

- Go to Setup
- In the Quick Find box, type the Security and then select the view Setup Audit Trail
- You can see last 180 days of changes

Salesforce Project Phase-10

Health Patient Management and Telemedicine CRM

Phase 10: Final Presentation & Demo Day

- **Pitch Presentation:**
  - **Problem:** Inefficient tracking of patients, appointments, and bed availability using disconnected spreadsheets or manual records, leading to errors and delayed decision-making.
  - **Solution:** A centralized HealthCare Management System on Salesforce to manage patients, appointments, bed availability, and analytics in one platform.
  - **Benefits:**
    - Streamlined patient and appointment management
    - Accurate bed tracking and allocation
    - Quick search and navigation with Lightning Web Components
    - Data-driven insights via dynamic dashboards and reports
    - API integration for external patient data

- **Demo Walkthrough:**
  - Start on the Home Page Dashboard to get an overview of patient status, appointments, and bed availability.
  - Open a Patient record and demonstrate the Patient Quick Search LWC.
  - Navigate to Appointment records: create a new appointment, update its status, and show the workflow in action.
  - Navigate to Bed Availability records: update bed status and demonstrate tracking of occupied vs. available beds.
  - Show Reports & Charts:
    - Pie chart for bed occupancy
    - Bar chart for daily appointments
    - Summary of patient statuses

- Demonstrate API Callouts using PatientAPIService to fetch external patient data.

❖ **Handoff Documentation:**

User Guide:

- How to create a new patient record
- How to schedule or update appointments
- How to track bed availability
- How to use quick search and navigation features

Admin Guide:

- Overview of custom objects: Patients__c, Appointment__c, Bed_Availability__c
- Lightning Web Components used
- Apex classes for API integration and imperative calls
- Reports and dashboards setup
- Sharing settings, field-level security, and login IP ranges

❖ **Portfolio Project Showcase:**

Professional summary for LinkedIn/Portfolio:

- Title: Health Patient Management and Telemedicine CRM
- Description:
  "Designed, developed, and deployed a custom HealthCare Management System on Salesforce to efficiently manage patients, appointments, and bed availability. The system provides real-time analytics through dashboards, enables quick searches with LWC components, integrates external patient data via API callouts, and ensures accurate reporting for hospital operations."
- Skills: Salesforce Administration, Apex, Lightning Web Components (LWC), SOQL, Flows & Process Automation, Data Modeling, API Integration, Reports & Dashboards
- Link: https://github.com/anu379/Healthcare-Patient-Management-and-Telemedicine-CRM