

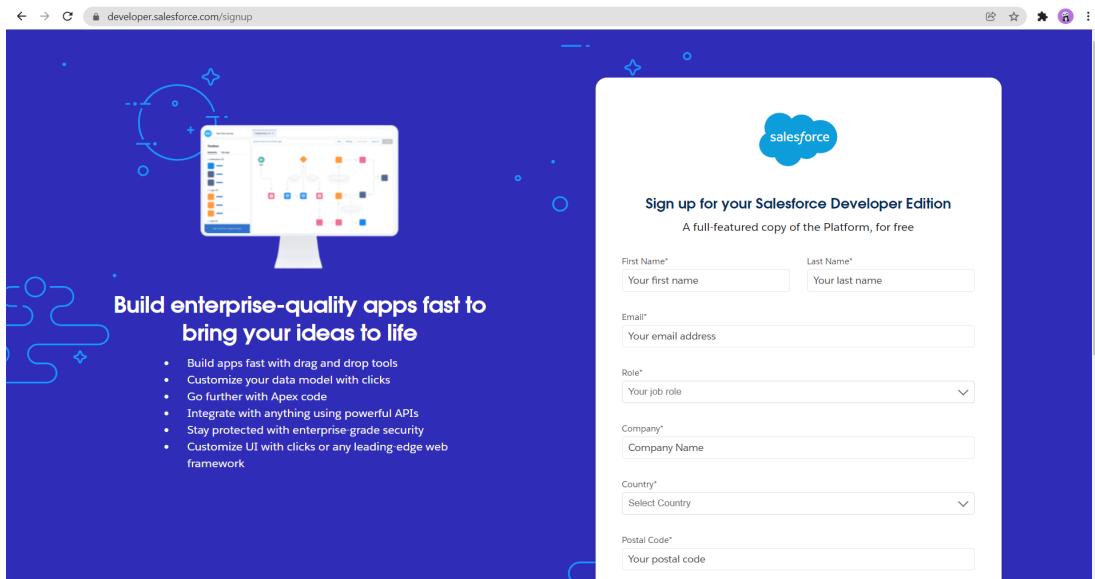
Day 1 Project Report

Topic: Introduction to lightning application.

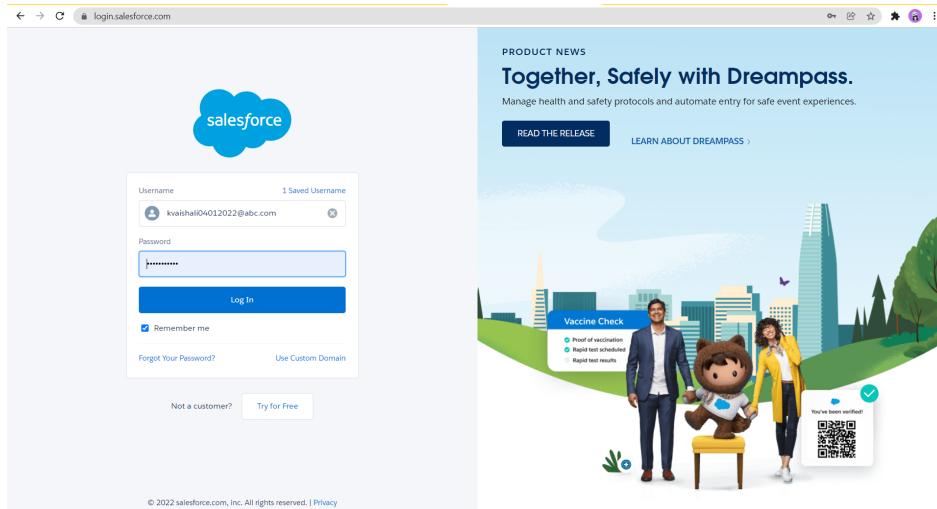
Activities: Created developer account using "developer.salesforce.com" site with my details .

And I received a mail from developer salesforce for verification.

Description And Screenshots:

A screenshot of a Gmail inbox. The inbox shows several messages, with one from "developer@salesforce.com" titled "Welcome to Salesforce: Verify your account". The message body contains a "Verify Account" button and a URL: <https://d5j000004nbvveaq-dev-ed.my.salesforce.com>. The rest of the screen shows the standard Gmail interface with navigation bars and other messages.

I loved into my salesforce account by salesforce.com/login



then I returned this home page:

A screenshot of the Salesforce Setup Home page at d5j000004nbdveaq-dev-ed.lightning.force.com/lightning/setup/SetupOneHome/home. The page has a light blue header with the word 'Setup' and a 'Home' button. On the left is a sidebar with navigation links like 'Service Setup Assistant', 'Release Updates', and 'Lightning Experience Transition Assistant'. The main content area features three cards: 'Get Started with Einstein Bots' (with a 'Get Started' button), 'Mobile Publisher' (with a 'Learn More' button), and 'Real-time Collaborative Docs' (with a 'Get Started' button). Below these cards is a section titled 'Most Recently Used' with a table showing 10 items. The table has columns for 'NAME', 'TYPE', and 'OBJECT'. Examples include 'Marketing' (Custom App), 'Phone_number_rule' (Validation Rule), and 'Trainer' (Custom Object Definition).

DAY 2

Topic: Creating lightning application and adding its objects and fields

Activities and description: 1.created a "online training management "app using following steps

app manager>>new lightning app>>filling the details as required
thus created a lightning app

2.created following custom objects for the app :Course ,Trainer
setup>>object manager>>create custom object

3.created fields for course object ,they are

Courses ID - Auto number (C-000001)

Course Name: Text

Course Duration (In Months) – Picklist (1,2,3,4,5,6)

Course Category: Picklist (Cloud Computing, Data Management, Programming, Web Designing)

Course Fee: Currency

Course Description:

i.e object manager>>course>>fields and relationship>>new>>filling details as required (mentioned above)

4.created fields for trainer object ,they are

Trainer Id: Auto-number (T-00001)

Full Name: Text

Email: Email - No Duplicate

Phone Number: Phone

WhatsApp Number: Phone

Date Of Birth: Date

Age : Formula

Experience: Picklist

LinkedIn Profile: URL

Expertise : Text (CRM, BI)

this time I used different method for addend the fields i.e

object manager>>schema builder>>course>>and adding the appropriate fields as mentioned above as respective type

4.created validation rule for course object

->>rule for course fee field should be less than 3000 or grater than 0

course>>validation rules>>formula is "OR(Course_Fee__c<0,
Course_Fee__c >3000)".

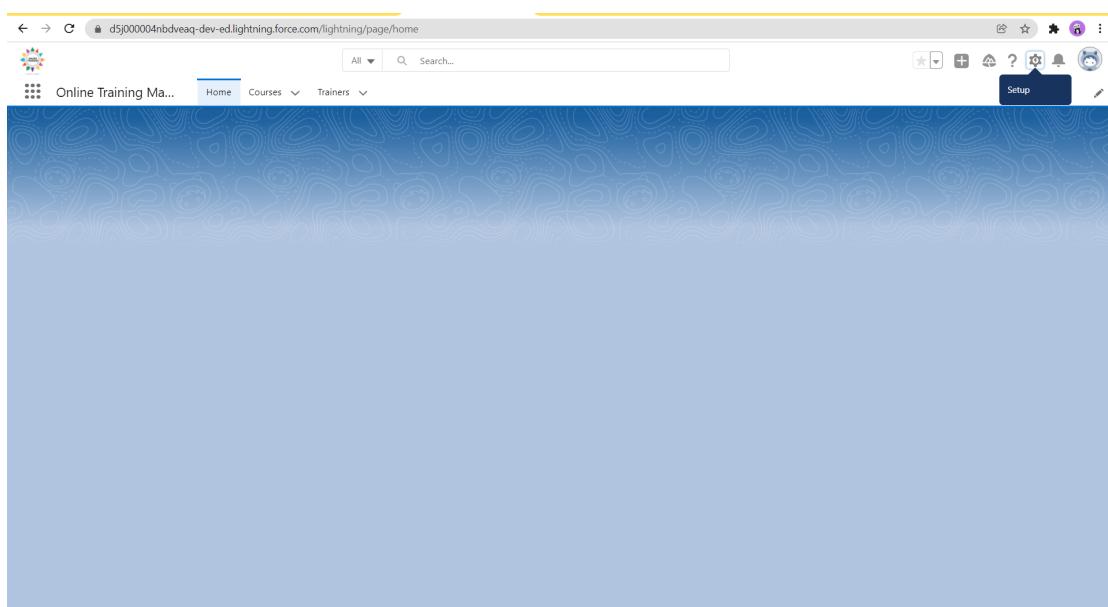
5.created validation rule for trainer object

->>rule for trainer age should be greater than 25

trainer>>validation rules>>formula is" Age__c <25".

Screenshots:

online training management lightning app



course object

The screenshot shows the Salesforce Object Manager interface. The search bar at the top contains 'course'. A table below lists one item:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Course	Courses__c	Custom Object		1/4/2022	✓

trainer object

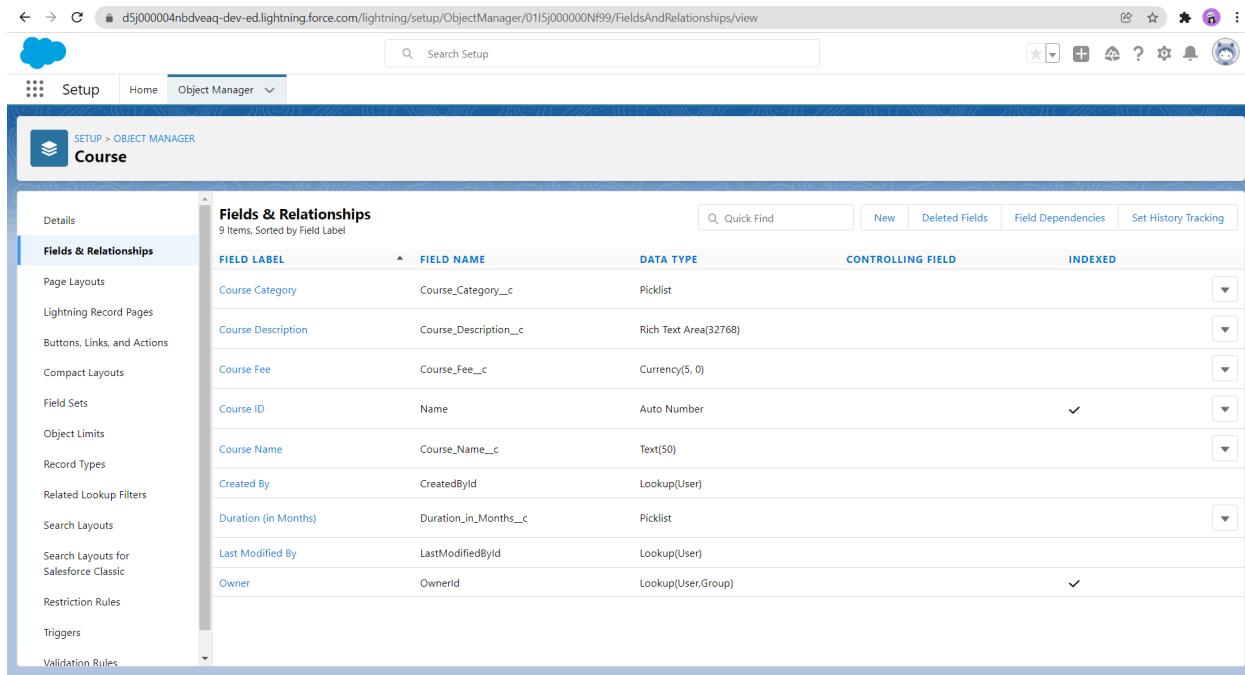
The screenshot shows the Salesforce Object Manager interface. The search bar at the top contains 'trainer'. A table below lists one item:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Trainer	Trainer__c	Custom Object		1/4/2022	✓

At the bottom left of the page, there is a small piece of JavaScript code:

```
javascript:void(0);
```

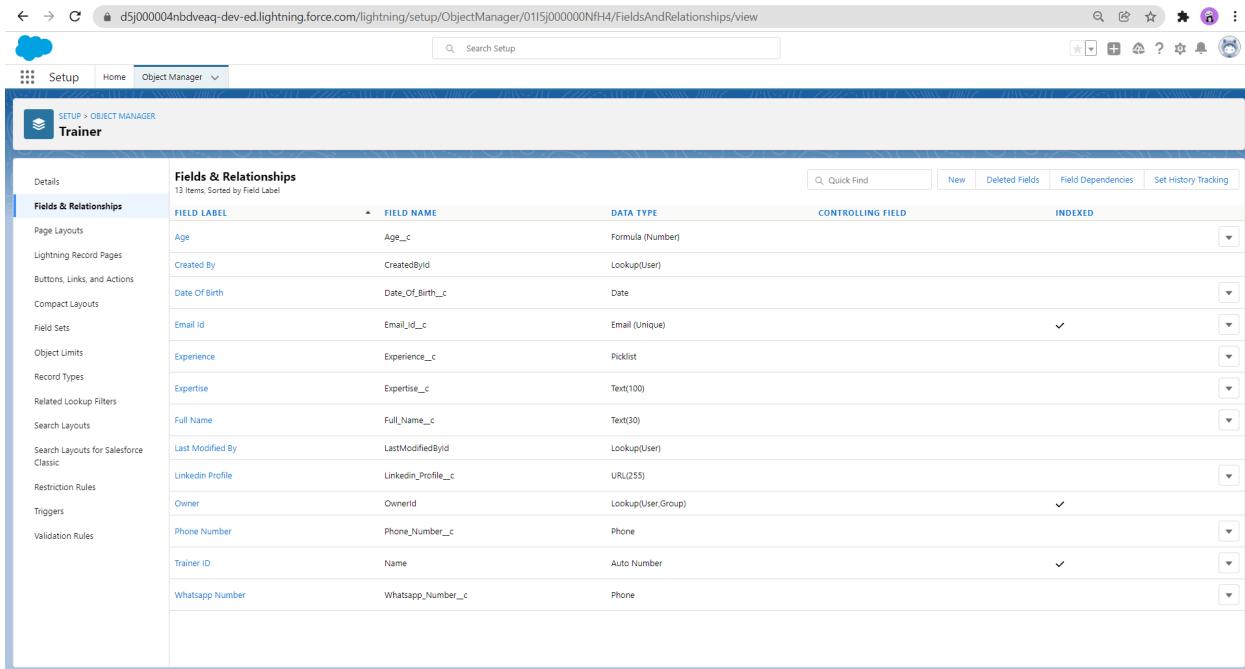
course fields



The screenshot shows the Salesforce Object Manager for the 'Course' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, and Field Sets. The main area displays a table titled 'Fields & Relationships' containing 9 items. The columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Course Category	Course_Category__c	Picklist		
Course Description	Course_Description__c	Rich Text Area(32768)		
Course Fee	Course_Fee__c	Currency(5, 0)		
Course ID	Name	Auto Number		
Course Name	Course_Name__c	Text(50)		
Created By	CreatedById	Lookup(User)		
Duration (in Months)	Duration_in_Months__c	Picklist		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		

trainer fields



The screenshot shows the Salesforce Object Manager for the 'Trainer' object. The left sidebar lists various setup options. The main area displays a table titled 'Fields & Relationships' containing 13 items. The columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Age	Age__c	Formula (Number)		
Created By	CreatedById	Lookup(User)		
Date Of Birth	DateOfBirth__c	Date		
Email Id	Email_Id__c	Email (Unique)		
Experience	Experience__c	Picklist		
Expertise	Expertise__c	Text(100)		
Full Name	Full_Name__c	Text(30)		
Last Modified By	LastModifiedById	Lookup(User)		
LinkedIn Profile	LinkedIn_Profile__c	URL(255)		
Owner	OwnerId	Lookup(User,Group)		
Phone Number	Phone_Number__c	Phone		
Trainer ID	Name	Auto Number		
Whatsapp Number	Whatsapp_Number__c	Phone		

course object validation rule for course fee field

The screenshot shows the Salesforce Setup interface for the 'Course' object. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The main content area displays the 'Course Validation Rule' detail page. The validation rule is named 'Course_Fee_Validation_Rule'. The error condition formula is set to OR(Course_Fee__c<0, Course_Fee__c>3000). The error message is 'Course fee cannot be less than 0 or greater than 3000'. The rule is active and was created by Vaishali Karnewar on 1/4/2022, 3:31 AM.

trainer object validation rule for trainer age field

The screenshot shows the Salesforce Setup interface for the 'Trainer' object. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The main content area displays the 'Trainer Validation Rule' detail page. The validation rule is named 'Trainer_Age_rule'. The error condition formula is set to Age__c < 25. The error message is 'age should be greater than 25'. The rule is active and was created by Vaishali Karnewar on 1/4/2022, 4:41 AM.

Day 3

Topic: Relationships between objects

Activities and Screenshots:

1.Creating custom objects and their fields: a)**Courses** - Course ID,Course Name,Course Duration,Course Fee,Course Description,Course Category.

b)**Trainer**-Trainer ID,Email Id, Phone Number ,Whatsapp Number, DateofBirth, Age, Experience, LinkedIn Profile, Expertise.

c)**Student**:First Name,Last Name,Email,Phone Number,Date Of Birth,Whatsapp Number,Enrollment Date,Student Enrollment.

d)**Meeting**:Meeting ID, Valid From, Valid To,Active, Meeting Link.

e)**Batch Scheduling**: Batch ID,Batch Type,Start Date, End Date, Course ID, Meeting id,Trainer id ,Status,Number Of Students Enrolled,Details.

The screenshot shows the Salesforce Schema Builder interface. At the top, there's a navigation bar with links for Setup, Home, Object Manager, and other system icons. Below the navigation is a search bar and a legend indicating field types: Lookup (blue line), Master (red line), and Required (purple line).

The main area displays five custom objects:

- Student**: Fields include Created By (Lookup), Date Of Birth (Date), Email Id (Email Unique), Enrollment Date (Formula Date), First Name (Text 50), Last Modified By (Lookup User), Last Name (Text 50), Owner (Lookup User+1), Phone Number (Phone), Student age (Formula Number), Student Id (Auto Number), and Whatsapp Number (Phone).
- Batch Scheduling**: Fields include Batch Id (Auto Number), Batch Status (Picklist), Batch Type (Picklist), Course Category (Formula Text), Course ID (Lookup Course), Course Name (Formula Text), Created By (Lookup User), End Date (Date), Last Modified By (Lookup User), Meeting ID (Lookup Meeting), Meeting Url (Formula Text), Meeting valid till (Formula Date), Owner (Lookup User+1), Start Date (Date), Trainer email (Email), Trainer ID (Lookup Trainer), and Trainer Name (Formula Text).
- Course**: Fields include Course Category (Picklist), Course Description (Rich Text Area 32768), Course Fee (Currency 3, 0), Course ID (Auto Number), Course Name (Text 50), Created By (Lookup User), Duration (in Months) (Formula Number), Last Modified By (Lookup User), Owner (Lookup User+1), and Total NO Of Trainer (Roll-Up Summary COUNT Course-Trainer).
- Meeting**: Fields include Active (Checkbox), Created By (Lookup User), Last Modified By (Lookup User), Meeting Id (Auto Number), Meeting URL (URL 255), Owner (Lookup User+1), Valid From (Date), and Valid To (Date).
- Course-Trainer**: Fields include Course ID (Master-Detail Course), Course Name (Formula Text), Course Trainer ID (Auto Number), Created By (Lookup User), Last Modified By (Lookup User), Trainer ID (Master-Detail Trainer), and Trainer Name (Formula Text).

On the left, a sidebar lists selected objects: Batch Scheduling, Course, Course-Trainer, Meeting, and Trainer. A legend on the right defines the colors for Lookup, Master, and Required relationships.

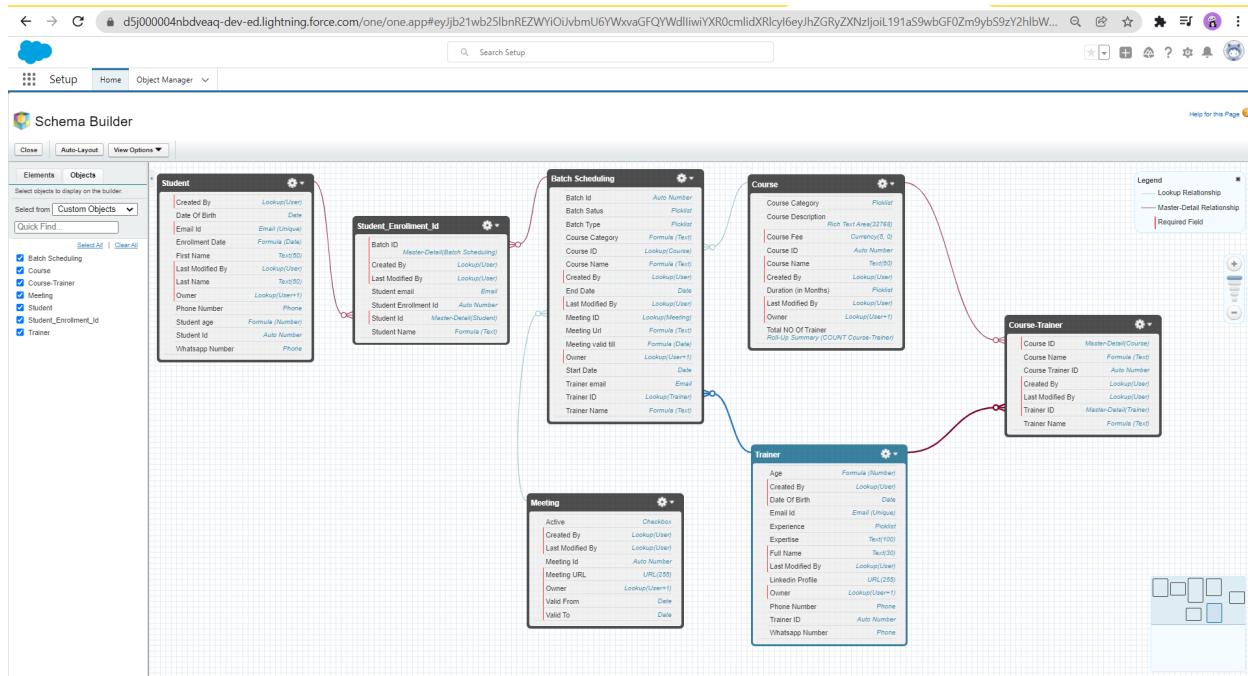
2.Relationship between objects:

-->Master Detail Relationship between {student and batch scheduling}:

Student_Enrollment_id},{Course and Trainer :Course-Trainer}.

-->Lookup Relation between {Course and Batch scheduling},{Trainer and Batch Scheduling},{meeting and Batch Scheduling}.

-->Roll up summary between {course and Trainer}.



3. Validation Rules of Custom Objects:

Trainer-

The screenshot shows the Salesforce Object Manager interface for the 'Trainer' object. On the left, a sidebar lists various setup categories like Page Layouts, Buttons, and Actions. The 'Validation Rules' section is selected and highlighted in blue. The main content area displays a table titled 'Validation Rules' with 6 items, sorted by Rule Name. The table columns are: RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. The data is as follows:

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Experience_fields_on_the_trainer_object	Experience	experience should not be empty	✓	Vaishali karnewar, 1/10/2022, 12:02 AM
Full_Name_blank	Full Name	trainer full name should not be empty	✓	Vaishali karnewar, 1/6/2022, 12:56 AM
Phone_DOB	Top of Page	PHONE NUMBER AND DATE OF BIRTH SHOULD NOT BE EMPTY	✓	Vaishali karnewar, 1/6/2022, 1:08 AM
Phone_number_rule	Phone Number	phone number should be min 10 digits and max 15 digits	✓	Vaishali karnewar, 1/6/2022, 1:16 AM
Trainer_Age_rule	Date Of Birth	age should be greater than 25	✓	Vaishali karnewar, 1/4/2022, 4:41 AM
Whatsapp_no_shouldnot_be_empty	Top of Page	whatsappno should be greter than 10 digits	✓	Vaishali karnewar, 1/6/2022, 1:17 AM

Student-

The screenshot shows the Salesforce Object Manager interface for the 'Student' object. The sidebar lists various setup categories. The 'Validation Rules' section is selected and highlighted in blue. The main content area displays a table titled 'Validation Rules' with 3 items, sorted by Rule Name. The table columns are: RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. The data is as follows:

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Lastname_should_not_be_Bank	Last Name	last name shouldnot be empty	✓	Vaishali karnewar, 1/6/2022, 1:28 AM
Minimum_16_years_of_age	Top of Page	Minimum 16 years of age	✓	Vaishali karnewar, 1/6/2022, 1:34 AM
Phone_and_whatsappNumber_should_be_minim	Top of Page	Phone and whatsappNumber should be minimum 10 digits	✓	Vaishali karnewar, 1/6/2022, 1:36 AM

Meeting:

The screenshot shows the Salesforce Setup interface for the 'Object Manager' section, specifically for the 'Meeting' object. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The main content area displays the 'Validation Rules' for the Meeting object. There are three validation rules listed:

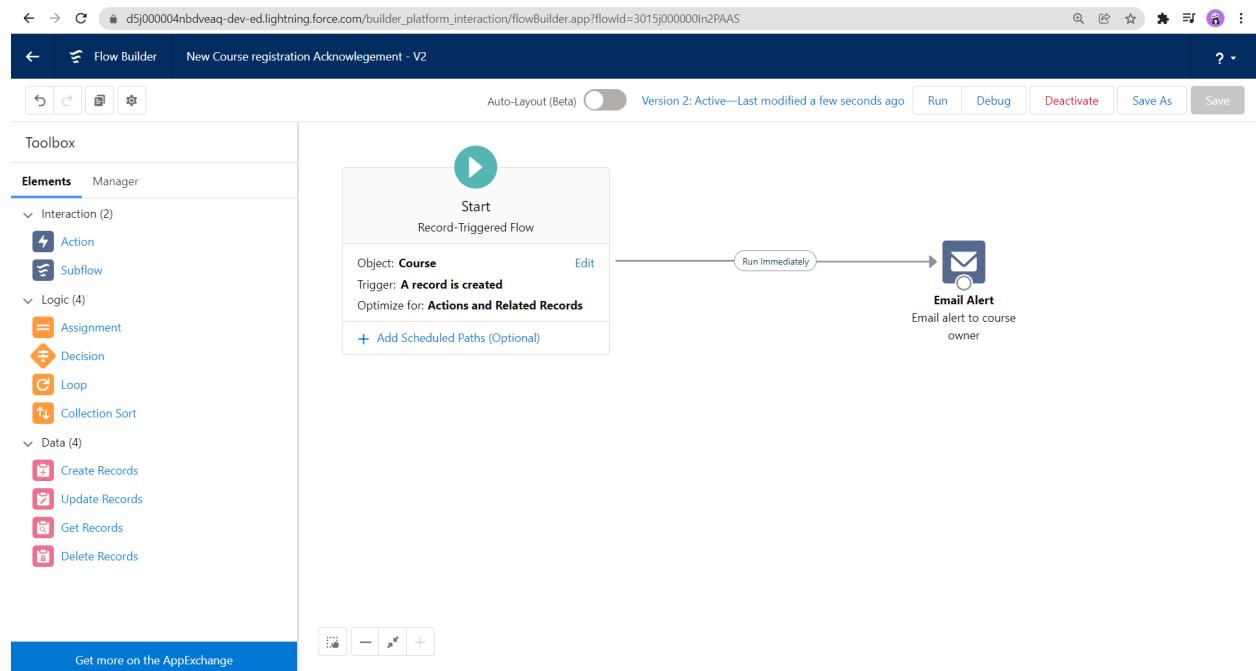
RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
MIN_14_DAYS_DIFF_BTN_VALID_TO_AND_FROM	Top of Page	Minimum 14 days difference should be in between valid from and valid to	✓	Vaishali karnewar, 1/6/2022, 1:40 AM
ValidTo_Cannot_be_less_than_Valid_from	Valid To	ValidTo Cannot be less than Valid from	✓	Vaishali karnewar, 1/6/2022, 1:41 AM
Valid_From_should_not_be_less_than_curreNT_DATE	Valid From	Valid_From_should_not_be_less_than_curreNT_DATE	✓	Vaishali karnewar, 1/6/2022, 1:41 AM

Day 4

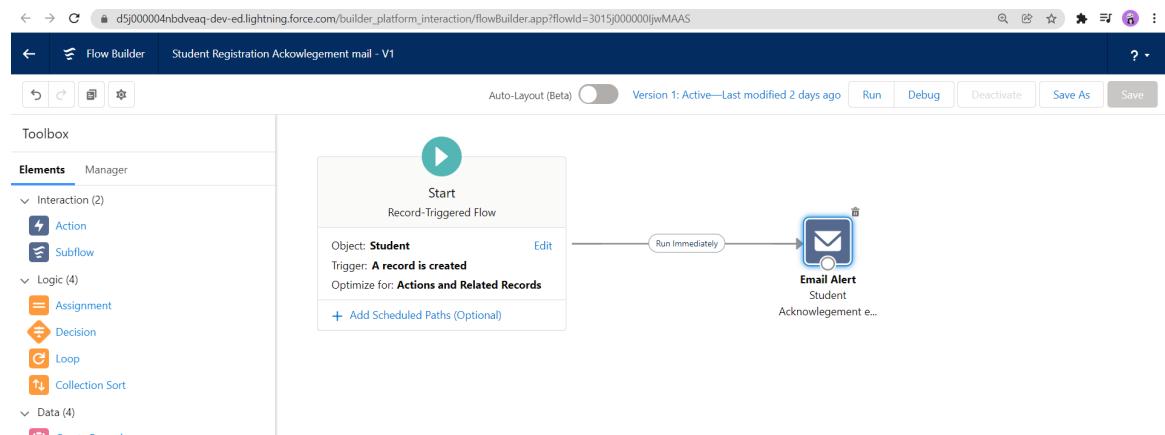
Topic: Automation on Custom Objects.

Activities:

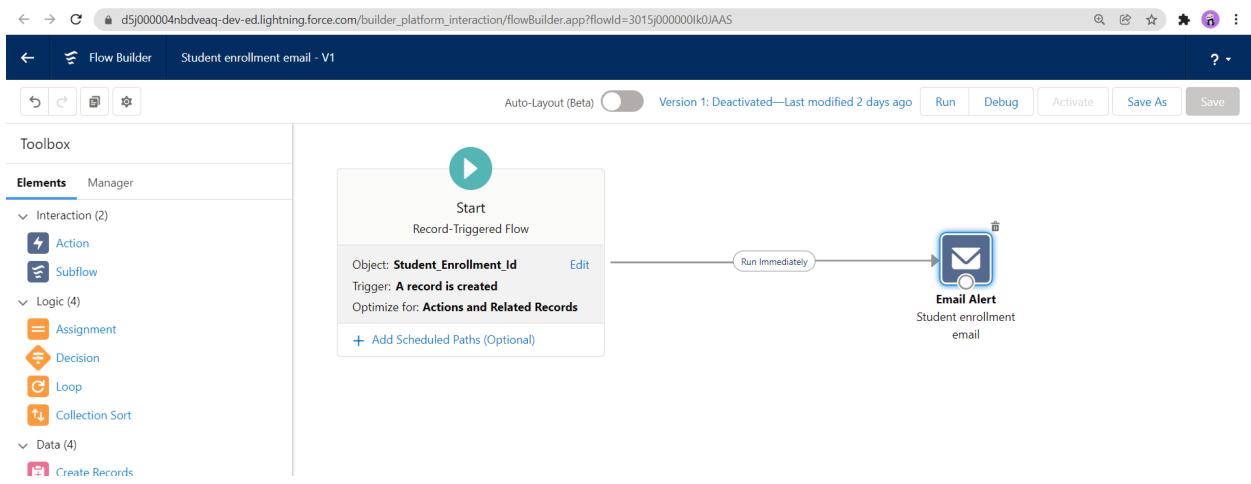
1. Automation Rule on Course object to send Email to owner every record gets updates using email templates and email alerts through flows:



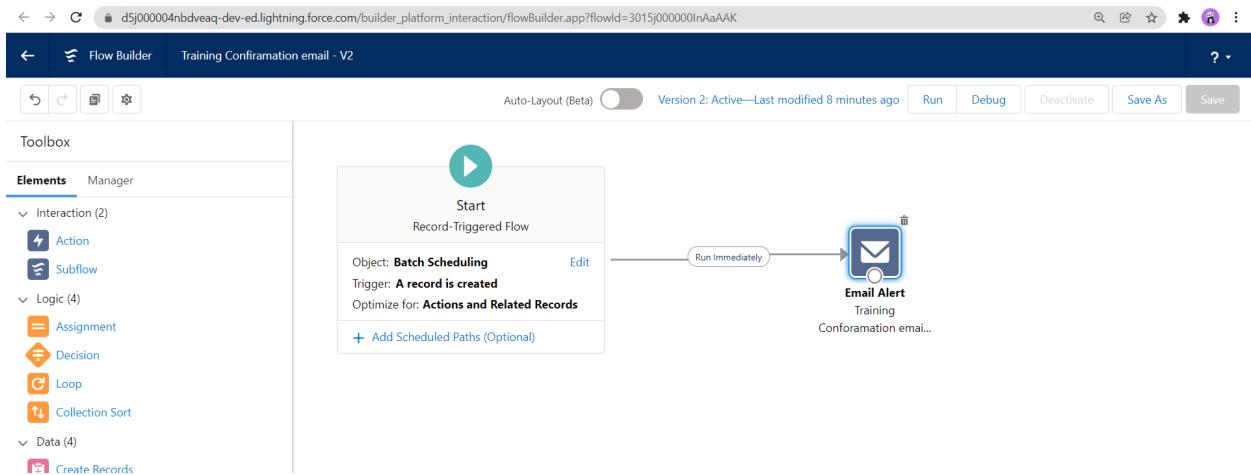
2. Email Automation on Student object to send Email to student every time he/she get Registered.



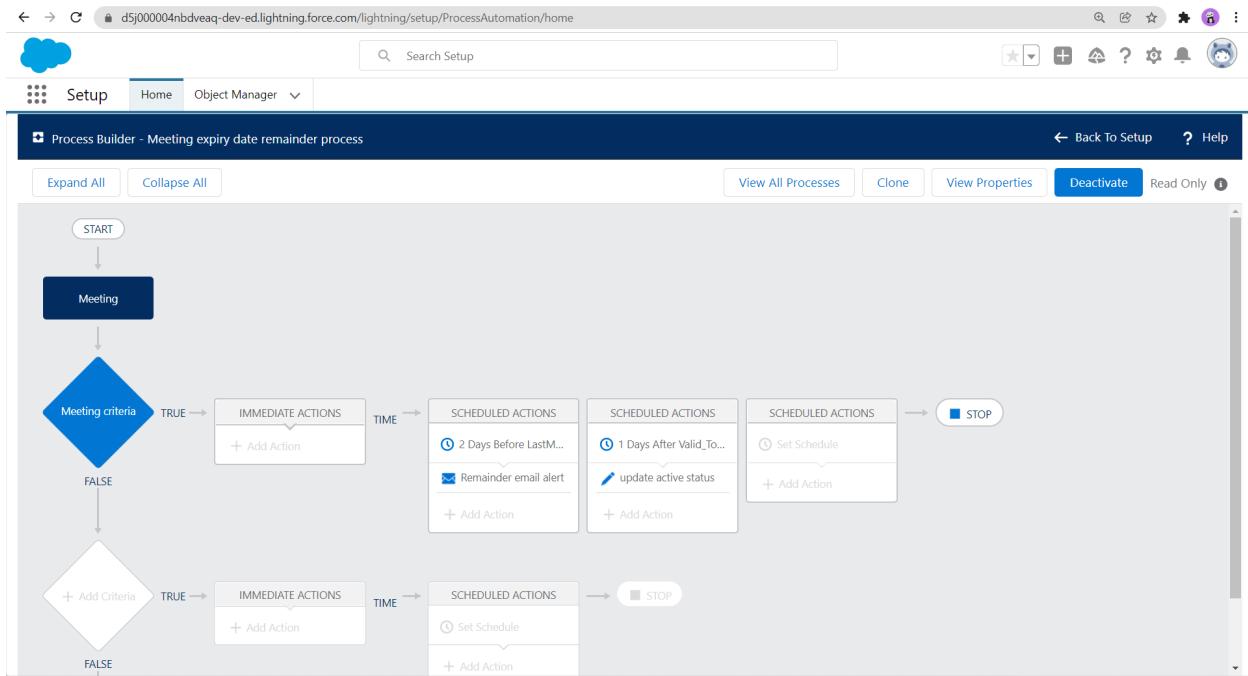
3. Email Automation on student Enrollment when ever student enrolled for a Batch.



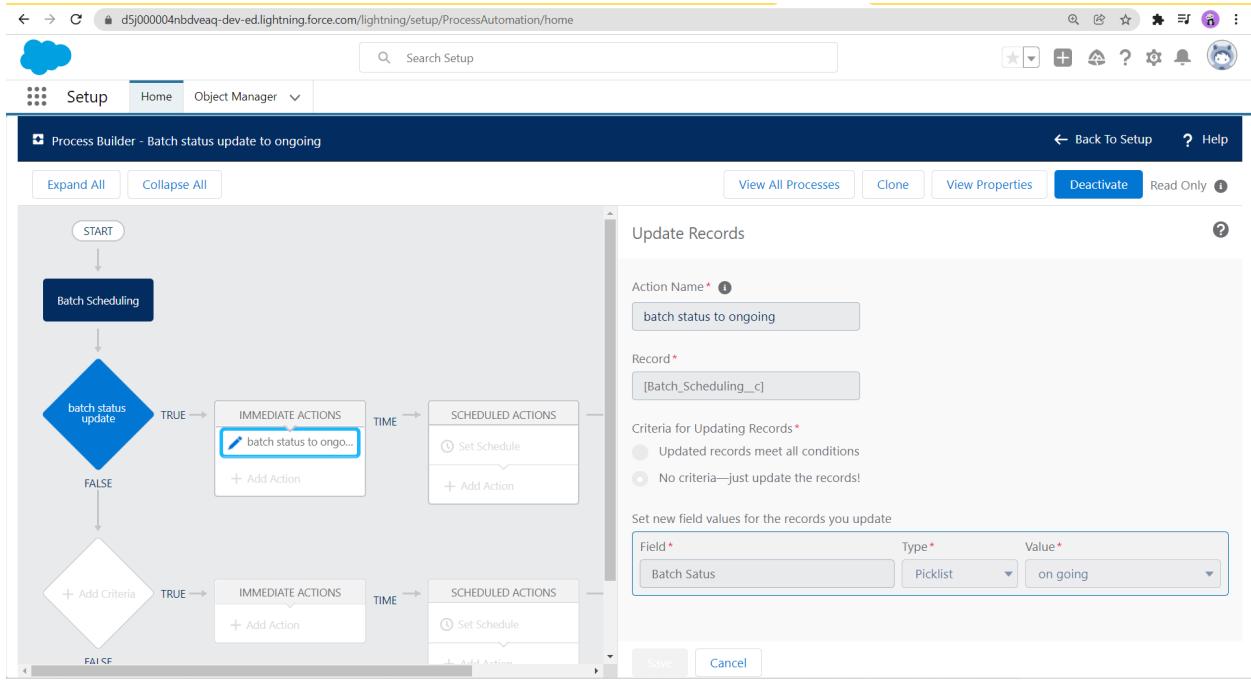
4. Email Automation on Trainer Object to send email to trainer every time he /she get registered



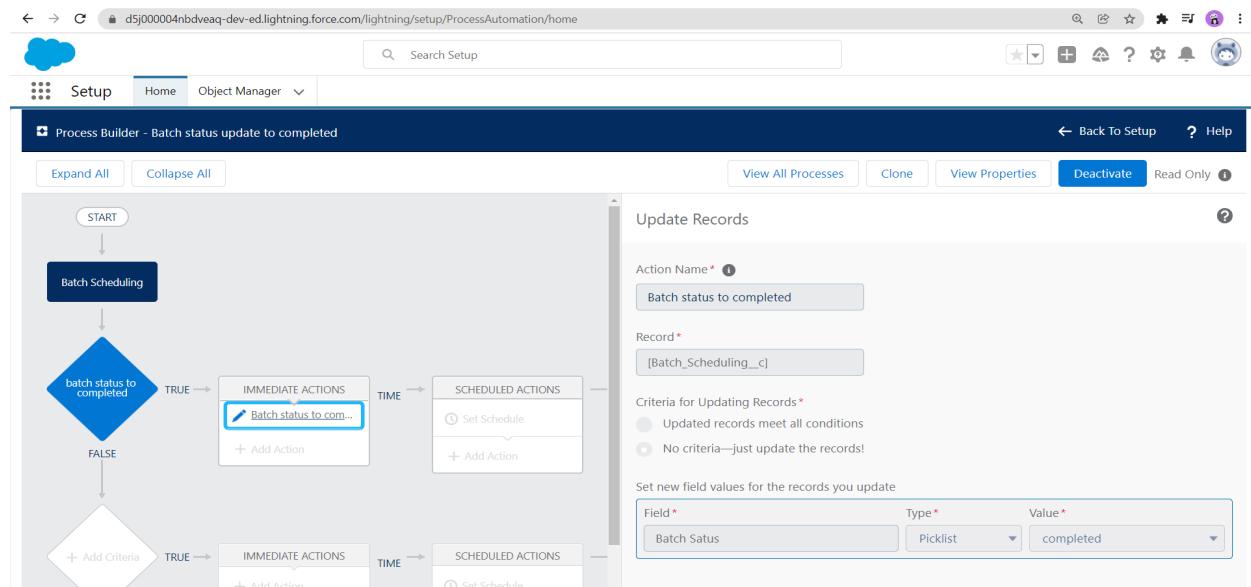
5.email alert to record owner that Meeting is going to end 2 days before or 1 day before



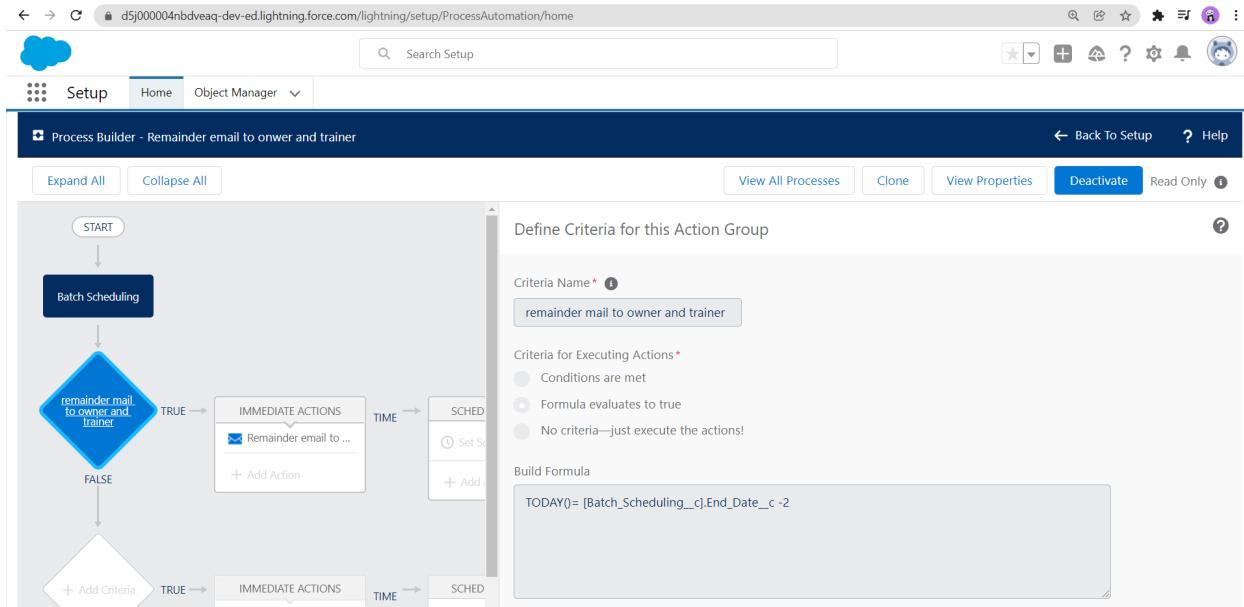
6. Batch status record update to on going when start day is the current date



7. Batch_status-record update to completed when end date is reached.



8.Remainder mail to owner and trainer before 2 days batch end date.



Day 5

Topic: Custom pages using Visual force Pages

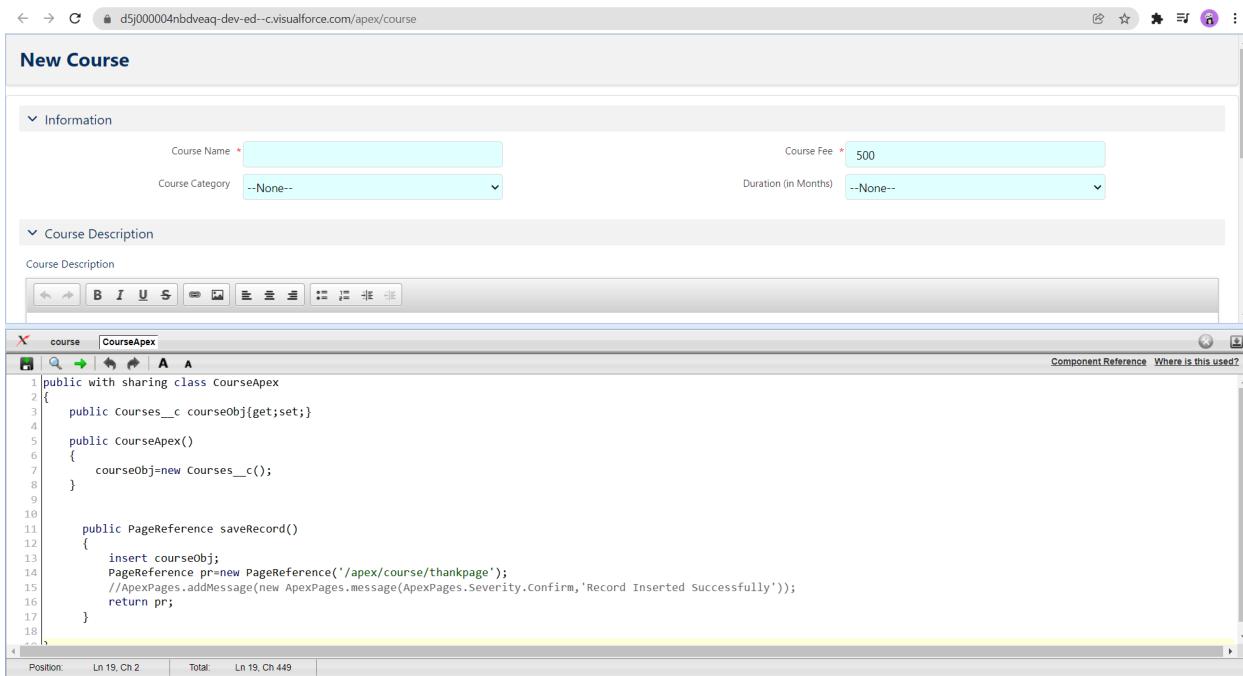
Activities and Screen shots:

1. Created a page online Training management VF and added a new course object using Visual force pages

VF page

The screenshot shows a Visualforce page titled "New Course". The URL in the browser is d5j000004nbdveaq-dev-ed.lightning.force.com/lightning/n/New_course. The page has a header with "Online Training Ma..." and "Home" buttons, and a search bar. The main content area is titled "New Course" and contains a section titled "Information". This section includes fields for "Course Name" (with a required asterisk), "Course Fee" (set to 500), "Course Category" (set to "--None--"), and "Duration (in Months)" (set to "--None--"). Below this is a "Course Description" section, which is currently collapsed. A "Save Record" button is located at the bottom right of the form.

CourseApex code:



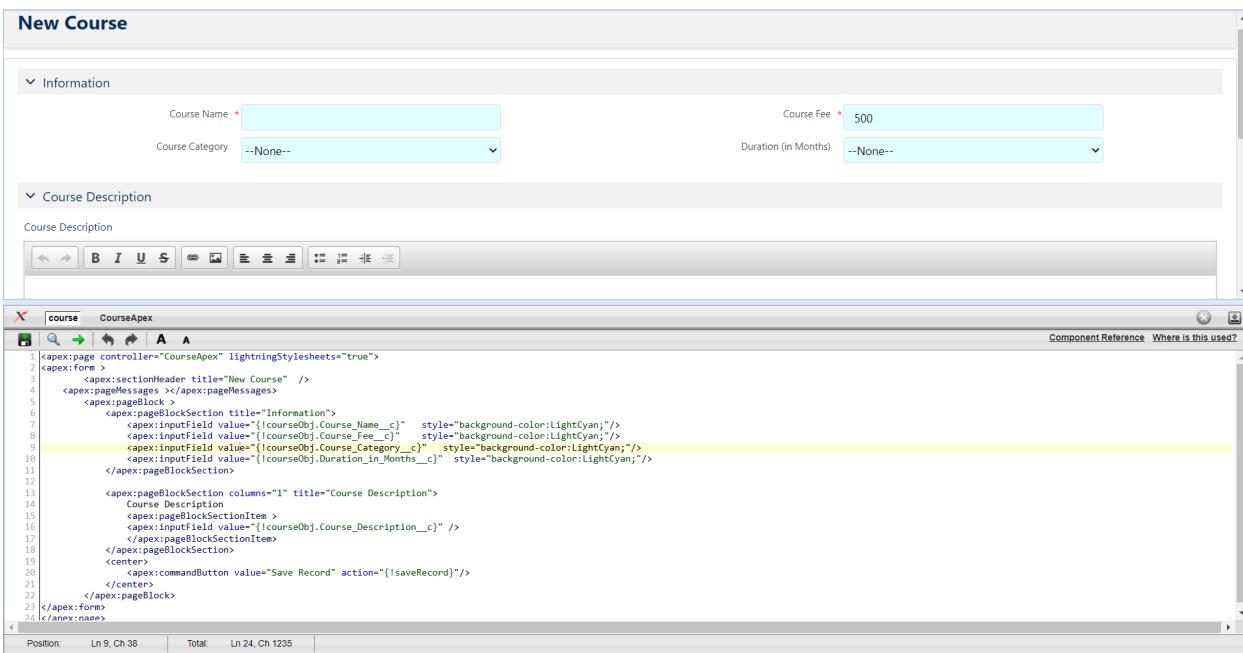
The screenshot shows the Salesforce developer console with the page title "New Course". The page has sections for "Information" and "Course Description". The "Information" section contains fields for "Course Name" (with value "500"), "Course Fee" (with value "500"), "Course Category" (with value "--None--"), and "Duration (in Months)" (with value "--None--"). The "Course Description" section contains a rich text editor with various styling options like bold, italic, and underline. Below the page content is the Apex code editor:

```
public with sharing class courseApex
{
    public Courses__c courseObj{get;set;}
    public CourseApex()
    {
        courseObj=new Courses__c();
    }

    public PageReference saveRecord()
    {
        insert courseObj;
        PageReference pr=new PageReference('/apex/course/thankpage');
        //ApexPages.addMessage(new ApexPages.message(ApexPages.Severity.Confirm,'Record Inserted Successfully'));
        return pr;
    }
}
```

The code editor shows the cursor at line 18, character 2. The status bar at the bottom indicates "Position: Ln 19, Ch 2" and "Total: Ln 19, Ch 449".

Course code:



The screenshot shows the Salesforce developer console with the page title "New Course". The page has sections for "Information" and "Course Description". The "Information" section contains fields for "Course Name" (with value "500"), "Course Fee" (with value "500"), "Course Category" (with value "--None--"), and "Duration (in Months)" (with value "--None--"). The "Course Description" section contains a rich text editor with various styling options like bold, italic, and underline. Below the page content is the Apex code editor:

```
<apex:page controller="CourseApex" lightningStylesheets="true">
<apex:form>
    <apex:sectionHeader title="New Course" />
    <apex:pageMessages /></apex:pageMessages>
    <apex:pageBlock>
        <apex:pageBlockSection title="Information">
            <apex:inputField value="{!courseObj.Course_Name__c}" style="background-color:LightCyan;" />
            <apex:inputField value="{!courseObj.Course_Fee__c}" style="background-color:LightCyan;" />
            <apex:inputField value="{!courseObj.Course_Category__c}" style="background-color:LightCyan;" />
            <apex:inputField value="{!courseObj.Duration_In_Months__c}" style="background-color:LightCyan;" />
        </apex:pageBlockSection>
        <apex:pageBlockSection columns="1" title="Course Description">
            Course Description
            <apex:pageBlockSectionItem>
                <apex:inputField value="{!courseObj.Course_Description__c}" />
            </apex:pageBlockSectionItem>
            <apex:pageBlockSectionItem>
                <center>
                    <apex:commandButton value="Save Record" action="{!saveRecord}" />
                </center>
            </apex:pageBlockSectionItem>
        </apex:pageBlockSection>
    </apex:pageBlock>
</apex:form>
</apex:page>
```

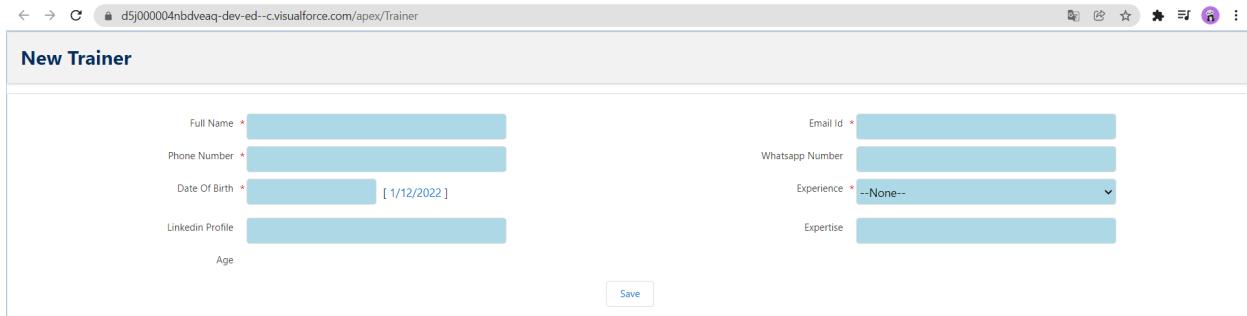
The code editor shows the cursor at line 9, character 38. The status bar at the bottom indicates "Position: Ln 9, Ch 38" and "Total: Ln 24, Ch 1235".

Day 6

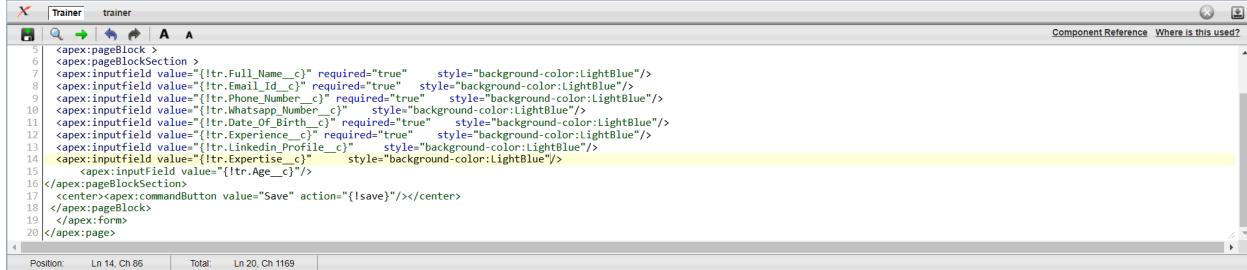
Topic: New Trainer Using Custom Controller (visual force).

Activities and Screen shots:

- 1.Creating New Trainer object on VF page using Custom controller.



The screenshot shows a Visualforce page titled "New Trainer". The page contains several input fields: "Full Name" (required), "Email Id" (required), "Phone Number" (required), "Whatsapp Number", "Date Of Birth" (with a value of "1/12/2022"), "Experience" (dropdown menu with "None" selected), "LinkedIn Profile", and "Expertise". A "Save" button is located at the bottom right of the form.



```
<apex:pageBlock>
<apex:pageBlockSection>
<apex:inputfield value="{!trr.Full_Name__c}" required="true" style="background-color:LightBlue"/>
<apex:inputfield value="{!trr.Email_Id__c}" required="true" style="background-color:LightBlue"/>
<apex:inputfield value="{!trr.Phone_Number__c}" required="true" style="background-color:LightBlue"/>
<apex:inputfield value="{!trr.Whatsapp_Number__c}" style="background-color:LightBlue"/>
<apex:inputfield value="{!trr.DateOfBirth__c}" required="true" style="background-color:LightBlue"/>
<apex:inputfield value="{!trr.Experience__c}" required="true" style="background-color:LightBlue"/>
<apex:inputfield value="{!trr.linkedin_Profile__c}" style="background-color:LightBlue"/>
<apex:inputfield value="{!trr.Expertise__c}" style="background-color:LightBlue"/>
<apex:inputfield value="{!trr.Age__c}"/>
</apex:pageBlockSection>
<center><apex:commandButton value="Save" action="{!save}"/></center>
</apex:pageBlock>
</apex:form>
</apex:page>
```

The screenshot shows the Salesforce IDE with the Apex code for the "Trainer" controller. The code includes a constructor, a getter for the Trainer object, and a save method that inserts the object. The "Expertise" field is highlighted with a yellow selection bar. The code editor shows line numbers from 1 to 20.

apex code:

The screenshot shows a Visualforce page titled "New Trainer" and its corresponding Apex code in the editor.

Visualforce Page (New Trainer):

Full Name *	Email Id *
Phone Number *	Whatsapp Number
Date Of Birth *	Experience * -None--
LinkedIn Profile	Expertise
Age	

Save

Apex Code Editor:

```
1 public with sharing class Trainer {
2     public Trainer__c tr{get; set;}
3     public Trainer__c()
4     {
5         tr=new Trainer__c();
6     }
7     public void save()
8     {
9         insert tr;
10        ApexPages.addmessage(new ApexPages.message(ApexPages.severity.Confirm, 'Record Inserted Successfully'));
11        tr=new Trainer__c();
12    }
13}
14}
```

Component Reference Where is this used?

Position: Ln 14, Ch 2 | Total: Ln 14, Ch 282

2.Similarly created visual force page for New Student ,New Batch, New meeting and added their tabs.

The screenshot shows the Salesforce Setup interface, specifically the "Tabs" configuration screen.

Setup Home

tabs

Tabs

Web Tabs

No Web Tabs have been defined

Visualforce Tabs

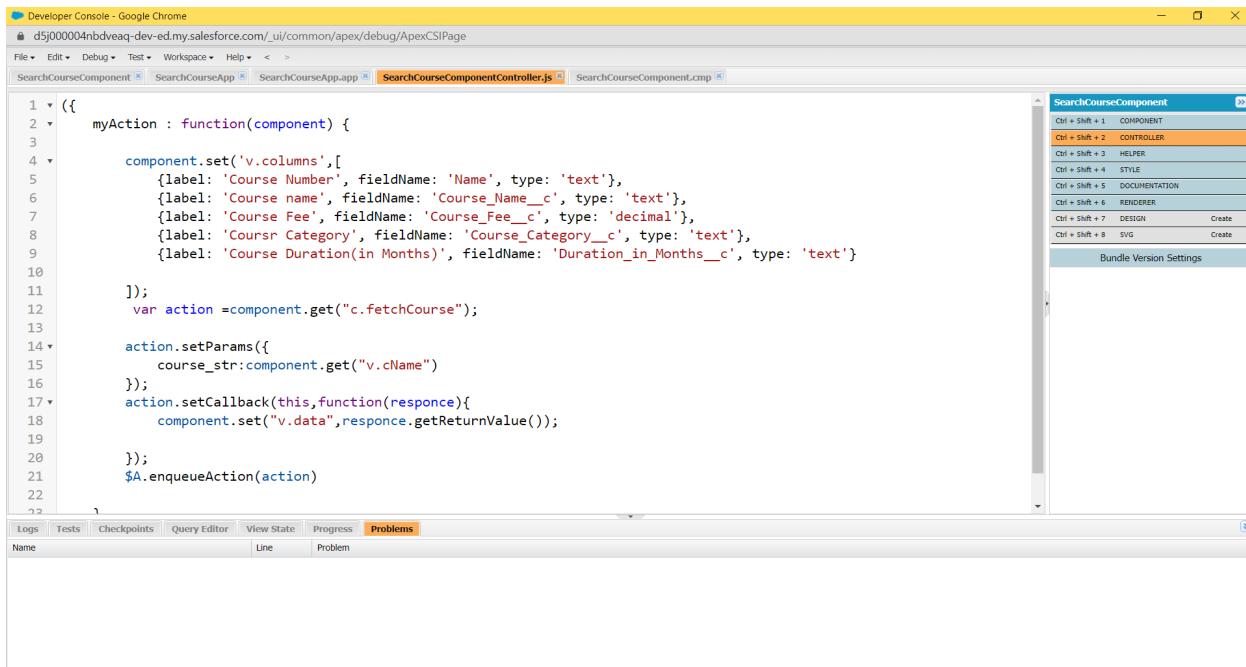
Action	Label	Tab Style	Description
Edit Del	New_Batch	Lightning	
Edit Del	New_Meeting	Hexagon	
Edit Del	New_course	Desk	
Edit Del	New_Student	People	
Edit Del	New_Trainer	Helicopter	
Edit Del	Search_Student	People	
Edit Del	Training_Calendar	Globe	

Day 7

Topic:Custom pages using lightning components.

Activities and Screenshots:

1.Create a lightning search course page in developer mode:



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is d5j000004nbvdeaq-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tabs at the top are File, Edit, Debug, Test, Workspace, Help, and < >. The active tab is SearchCourseComponentController.js. The code in the editor is as follows:

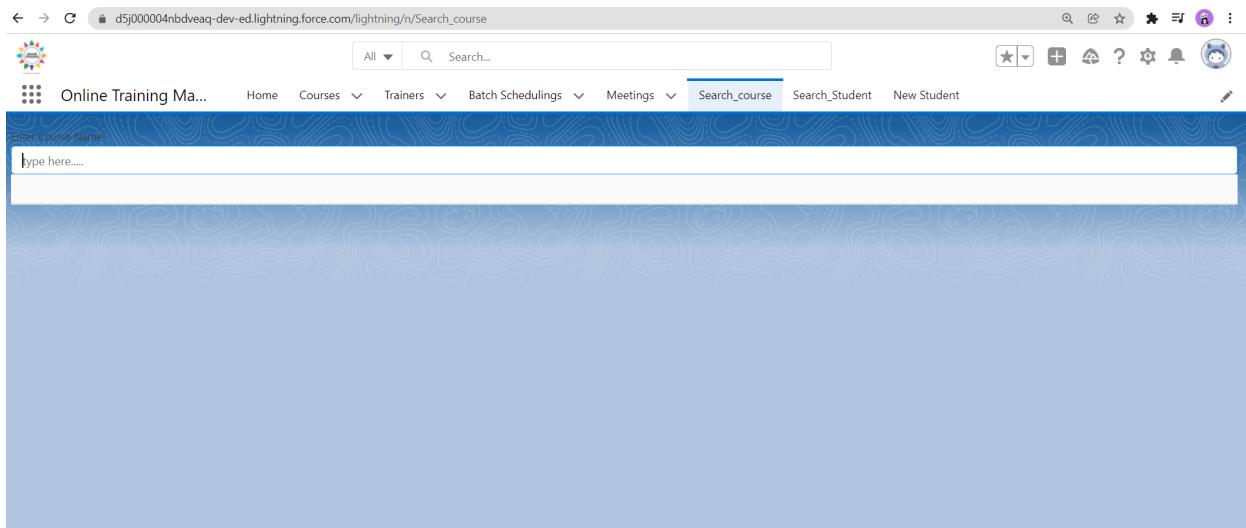
```
1  {{{
2   myAction : function(component) {
3
4     component.set('v.columns',[ 
5       {label: 'Course Number', fieldName: 'Name', type: 'text'},
6       {label: 'Course name', fieldName: 'Course_Name__c', type: 'text'},
7       {label: 'Course Fee', fieldName: 'Course_Fee__c', type: 'decimal'},
8       {label: 'Course Category', fieldName: 'Course_Category__c', type: 'text'},
9       {label: 'Course Duration(in Months)', fieldName: 'Duration_in_Months__c', type: 'text'}
10    ]);
11    var action =component.get("c.fetchCourse");
12
13    action.setParams({
14      course_str:component.get("v.cName")
15    });
16    action.setCallback(this,function(response){
17      component.set("v.data",response.getReturnValue());
18    });
19    $A.enqueueAction(action)
20  }
21
22 }};
```

To the right of the code editor is a sidebar titled "SearchCourseComponent" with the following key bindings:

Ctrl + Shift + 1	COMPONENT
Ctrl + Shift + 2	CONTROLLER
Ctrl + Shift + 3	HELPER
Ctrl + Shift + 4	STYLE
Ctrl + Shift + 5	DOCUMENTATION
Ctrl + Shift + 6	RENDERER
Ctrl + Shift + 7	DESIGN
Ctrl + Shift + 8	SVG

At the bottom of the sidebar are "Create" and "Bundle Version Settings" buttons.

And added lightning tab.



2.search student VF page :

using student email id ,we can check the details related to that student ,if the student id not found it shows an error.

The screenshot shows a web browser window with the URL d5j000004nbdveaq-dev-ed.lightning.force.com/lightning/n/Search_Student. The page title is "Search Student Details". The navigation bar includes links for Home, Courses, Trainers, Batch Schedulings, Meetings, Search_course, Search_Student (which is highlighted), and New Student. A search bar at the top has the placeholder "Search...". Below the search bar is a form with a text input field containing "vaishalikarnewar701@gmail.com" and a "Search" button. The main content area displays "Student Details" for a student with ID S-000005. The details listed are: Student Name: Pradeep Deepu, Contact No: 9859596876, Email Id: vaishalikarnewar701@gmail.com, Enrollment Date: Wed Jan 19 00:00:00 GMT 2022. Below this information is a table with the following data:

BATCH ID	BATCH TYPE	COURSE NAME	START DATE	TRAINER NAME	MEETING URL	BATCH STATUS
B-000005	Weekday	Salesforce developer	9/10/2022	priya	https://meet.google.com/?hs=197&pli=1&authuser=0	Scheduled

In this ,page we can check the batch related details to particular student also.

Apex-class of Search Student page:

Enter Student Email:

```

<apex:page controller="SearchStudent" lightningStylesheets="true">
<apex:form >
<apex:pageMessages >/apex/pageMessages>
<apex:sectionHeader title="Search Student Details"/>
<apex:pageBlock >
    Enter Student Email:&nbsp;&nbsp; <apex:inputText value="{!studentEmail}" />
    <apex:commandButton value="Search" action="{!searchRecord}" />
</apex:pageBlock>
<apex:pageBlock title="Student Details" rendered="{!display}">
    <p> Student Id:&nbsp; {!student[0].Name} </p>
    <p> Student Name:&nbsp; {!student[0].First_Name__c} {!student[0].Last_Name__c}</p>
    <p> Contact No:&nbsp; {!student[0].Phone_Number__c}</p>
    <p> Email Id:&nbsp; {!student[0].Email_Id__c}</p>
    <p> Enrollment Date:&nbsp; {!student[0].Enrollment_Date__c}</p>
</apex:pageBlock>
*****<apex:pageBlockTable value="{!batchList}" var="b">
    <apex:column value="{!b.Name}" />
    <apex:column value="{!b.Batch_Type__c}" />
    <apex:column value="{!b.Course_Name__c}" />
    <apex:column value="{!b.Meeting_Day__c}" />
    <apex:column value="{!b.Trainer_Name__c}" />
    <apex:column value="{!b.Meeting_Url__c}" />
    <apex:column value="{!b.Batch_Status__c}" />
</apex:pageBlockTable>
</apex:pageBlock>
</apex:form>
</apex:page>
```

Position: Ln 14, Ch 22 | Total: Ln 30, Ch 1423

Enter Student Email:

```

public with Sharing class SearchStudent {
    public String studentEmail {get;set;}
    public List<Student__c> student{get;set;}
    public boolean display{get;set;}
    public List<Student_Enrollment__c> stdEnroll {get;set;}
    public List<Batch_Scheduling__c> batchList{get;set;}
    public SearchStudent()
    {
        display=false;
        batchList =new List<Batch_Scheduling__c>();
    }
    public PageReference searchRecord()
    {
        if(studentEmail=='' || studentEmail==null)
        {
            ApexPages.addMessage(new ApexPages.message(ApexPages.severity.Error,'No Email Id Entered'));
            display=false;
        }
        else
        {
            student=[select Name,First_Name__c,Last_Name__c,Enrollment_Date__c,Email_Id__c,Phone_Number__c from Student__c where Email_Id__c=:studentEmail];
            if (student.size()>0)
            {
                display=true;
                stdEnroll=[select Name,Batch_ID__c from Student_Enrollment__c where Student_ID__c=:student[0].id];
                if (stdEnroll.size()>0)
```

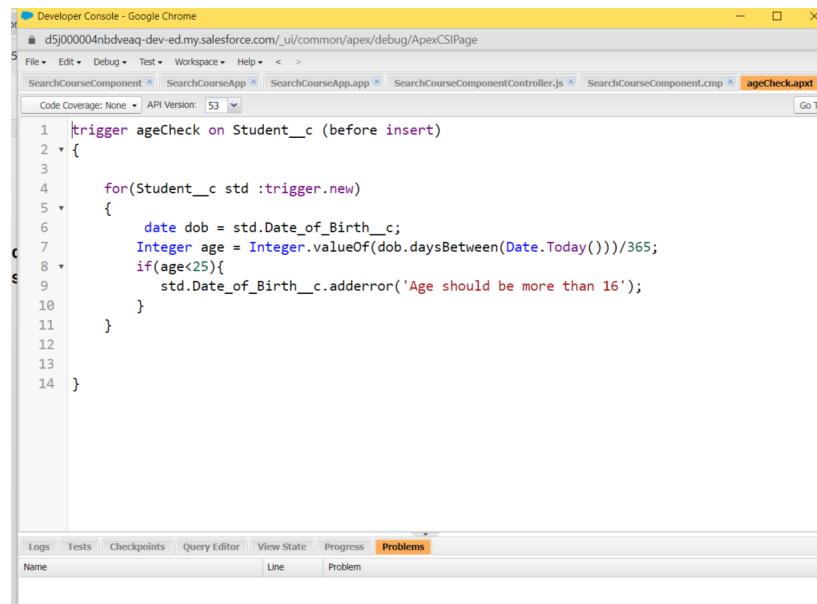
Position: Ln 63, Ch 4 | Total: Ln 63, Ch 1934

Day 8

Topic:Triggers on student ,Trainer,Meeting.

Activities and Screenshots:

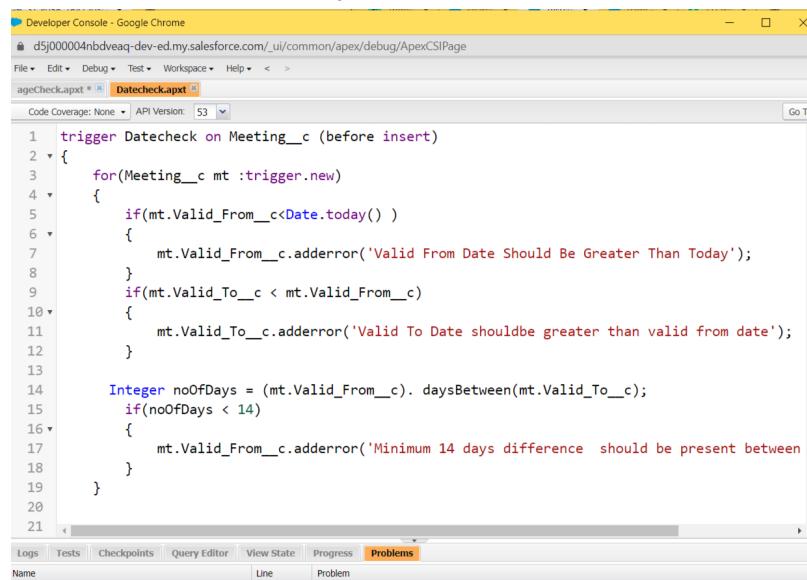
- 1.Trigger on Student object--the trigger gives error when the student age is below 16 year .



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is d5j000004nbvdeaq-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab bar includes SearchCourseComponent, SearchCourseApp, SearchCourseApp.app, SearchCourseComponentController.js, SearchCourseComponent.cmp, and ageCheck.apxt. The code editor contains a trigger named ageCheck on Student__c (before insert). The trigger logic checks if the student's age is less than 25 and adds an error message to the Date_of_Birth_c field if true. The code coverage is at 53%.

```
trigger ageCheck on Student__c (before insert)
{
    for(Student__c std :trigger.new)
    {
        date dob = std.Date_of_Birth__c;
        Integer age = Integer.valueOf(dob.daysBetween(Date.Today()))/365;
        if(age<25){
            std.Date_of_Birth__c.adderror('Age should be more than 16');
        }
    }
}
```

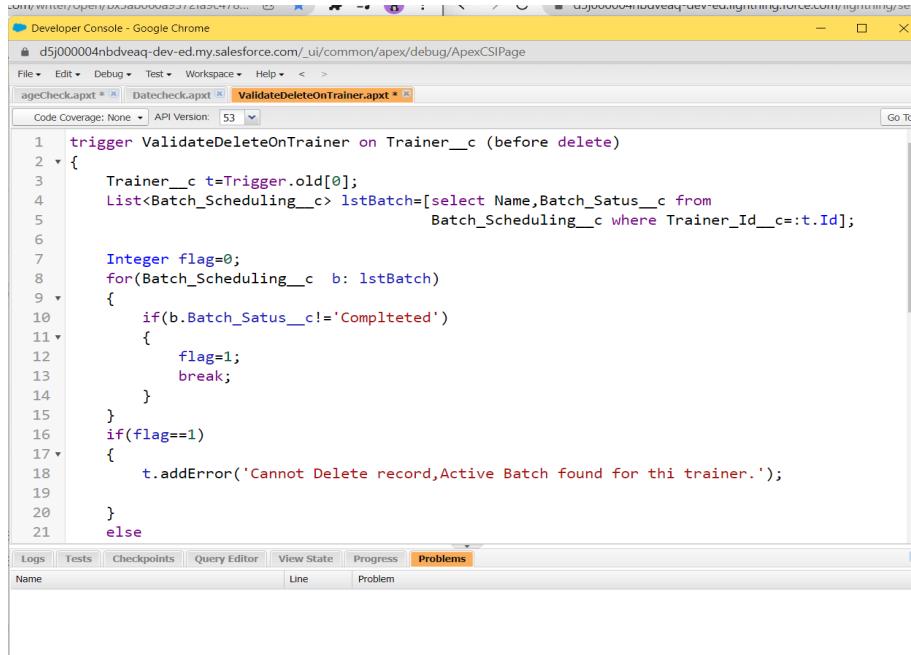
- 2.Trigger on Meeting object--that the valid from date should be greater than today and valid to should be greater than valid from.



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is d5j000004nbvdeaq-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab bar includes ageCheck.apxt and DatecheckKapxt. The code editor contains a trigger named Datecheck on Meeting__c (before insert). The trigger logic checks if the Valid_From__c date is less than today or if the Valid_To__c date is less than the Valid_From__c date. It also checks if the difference between the two dates is less than 14 days. Error messages are added to the Valid_From__c and Valid_To__c fields respectively. The code coverage is at 53%.

```
trigger Datecheck on Meeting__c (before insert)
{
    for(Meeting__c mt :trigger.new)
    {
        if(mt.Valid_From__c<Date.today() )
        {
            mt.Valid_From__c.adderror('Valid From Date Should Be Greater Than Today');
        }
        if(mt.Valid_To__c < mt.Valid_From__c)
        {
            mt.Valid_To__c.adderror('Valid To Date shouldbe greater than valid from date');
        }
        Integer noOfDays = (mt.Valid_From__c). daysBetween(mt.Valid_To__c);
        if(noOfDays < 14)
        {
            mt.Valid_From__c.adderror('Minimum 14 days difference should be present between');
        }
    }
}
```

3.Trigger on trainer field that if any active batch is ongoing to that trainer then the trigger will not allow the user to delete the trainer field.



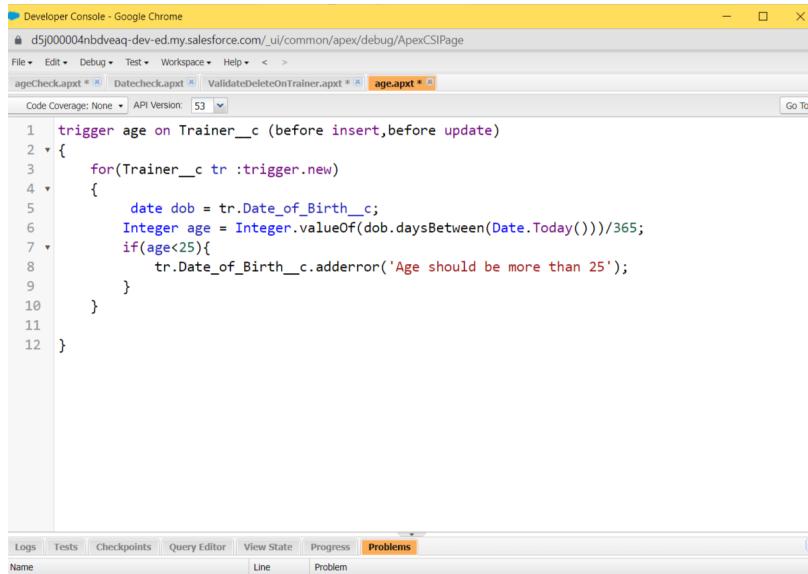
```

trigger ValidateDeleteOnTrainer on Trainer__c (before delete)
{
    Trainer__c t=Trigger.old[0];
    List<Batch_Scheduling__c> lstBatch=[select Name,Batch_Status__c from
                                         Batch_Scheduling__c where Trainer_Id__c=:t.Id];

    Integer flag=0;
    for(Batch_Scheduling__c b: lstBatch)
    {
        if(b.Batch_Status__c!='Complteded')
        {
            flag=1;
            break;
        }
    }
    if(flag==1)
    {
        t.addError('Cannot Delete record,Active Batch found for thi trainer.');
    }
    else
}

```

4.Trigger on Trainer object that the age of the Trainer should be above 25 years.



```

trigger age on Trainer__c (before insert,before update)
{
    for(Trainer__c tr :trigger.new)
    {
        date dob = tr.Date_of_Birth__c;
        Integer age = Integer.valueOf(dob.daysBetween(Date.Today()))/365;
        if(age<25){
            tr.Date_of_Birth__c.adderror('Age should be more than 25');
        }
    }
}

```

4.TrainerCalendar VF page :

this page will list out all the Scheduled batches and allow the user to register for particular batch.

The screenshot shows a browser window displaying a Visualforce page titled "Training List". The page lists three training batches in a table. Each row contains a "Register" button in the "ACTION" column. Below the table, the Visualforce page source code is shown in a code editor. The code uses Apex components like `<apex:page>`, `<apex:form>`, `<apex:sectionHeader>`, `<apex:pageBlock>`, `<apex:pageBlockTable>`, `<apex:column>`, and `<apex:commandButton>`. The code also includes JavaScript-like logic for rendering and parameter assignment.

KEY	BATCH NUMBER	COURSE NAME	BATCH TYPE	BATCH STATUS	ACTION
1	B-000004	Salesforce developer	Weekday	Scheduled	<button>Register</button>
2	B-000006	CRM training	Weekday	Scheduled	<button>Register</button>
3	B-000005	Salesforce developer	Weekday	Scheduled	<button>Register</button>

```
<apex:page controller="trainingCalendarApex" lightningStylesheets="true">
<apex:form >
    <apex:sectionHeader title="Training List"/>
    <apex:pageBlock id="bobj">
        <apex:pageBlockTable value="{!listWrap}" var="t">
            <apex:column value="{!t.key}" headerValue="Key"/>
            <apex:column value="{!t.batchInfo.Name}" headerValue="Batch Number"/>
            <apex:column value="{!t.batchInfo.Course_Name__c}" headerValue="Course Name"/>
            <apex:column value="{!t.batchInfo.Batch_Type__c}" headerValue="Batch Type"/>
            <apex:column value="{!t.batchInfo.Batch_Status__c }" headerValue="Batch Status"/>
            <apex:column headerValue="Action">
                <apex:commandButton value="Register" action="={!Register}" reRender="bobj">
                    <apex:param name="keyToRegister" value="{!t.key}" assignTo="!keyToRegister"/>
                </apex:commandButton>
            </apex:column>
        </apex:pageBlockTable>
    </apex:pageBlock>
</apex:form>
</apex:page>
```

Day 9

Topic:Custom App building.

Activities And Screen Shots:

1.Registration for training:

above page will navigates to this page ,where the student can able to register .

The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'Register for Training' and displays a form for registering a student. The form includes fields for 'Enter Student Email Id:' and a 'Search for Register' button. Below the form, there is a status message: 'Course Name :Salesforce developer'. The browser's address bar shows the URL: d5j000004nbvveaq-dev-ed-c.visualforce.com/apex/trainingCalendar. In the bottom right corner of the browser window, there is a small icon with a purple circle and a white dot.

The screenshot also shows an 'Apex Code' editor window below the browser. The code editor has tabs for 'trainingRegistration' and 'trainerregistrationApex'. The main pane displays the following Apex code:

```
12
13 </apex:pageBlock>
14
15 <apex:pageBlock rendered="{!studexist}" >
16   <apex:commandButton value="Click to Register" action="{!registerStudent}"/>
17 </apex:pageBlock>
18
19 <apex:pageBlock title="Student Status" rendered="{!display1}">
20   <h1 style="font-size:20px;">Student Not Registered for this batch </h1>
21   <apex:commandButton value="Click to Register for Current training" action="{!clicktoRegister}"/>
22 </apex:pageBlock>
23
24 </apex:form>
25 </apex:page>
```

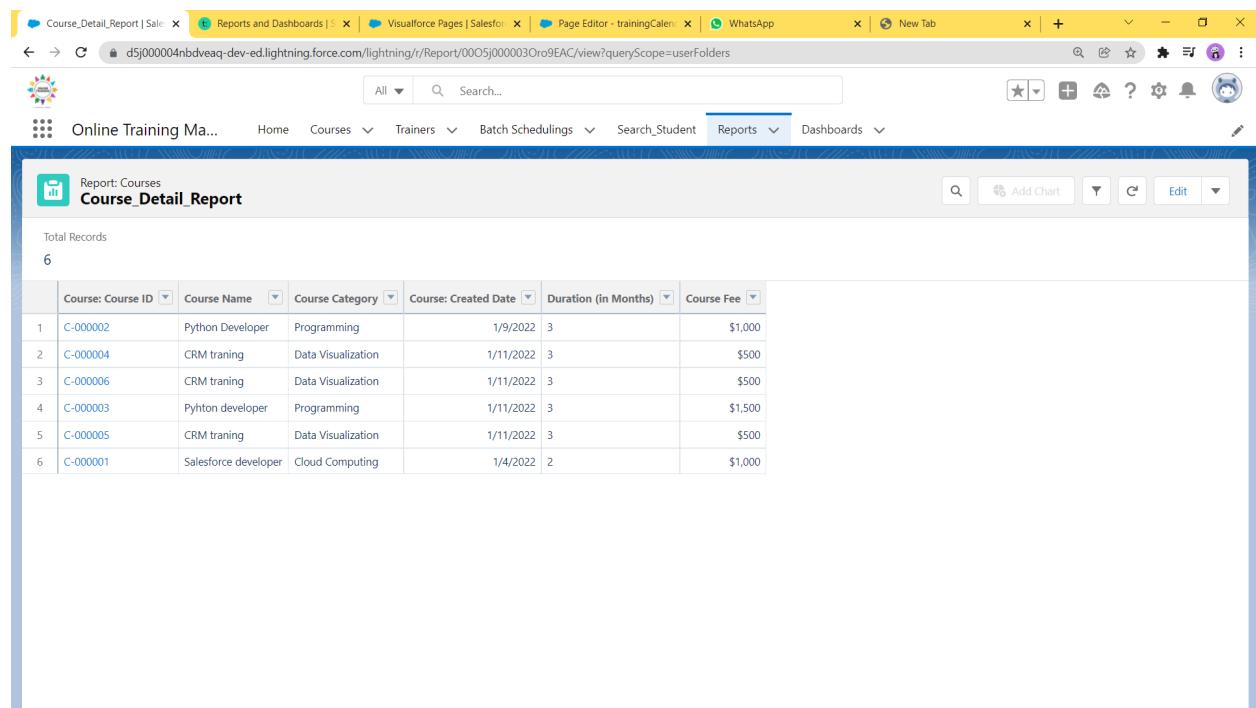
The code editor interface includes standard tools like search, copy, and paste, and a status bar at the bottom indicating 'Position: Ln 31, Ch 13' and 'Total: Ln 31, Ch 1110'.

Day 10

Topic:Reports and dashboards .

Activities and Screen shots:

1.Create Report on Course Detail.



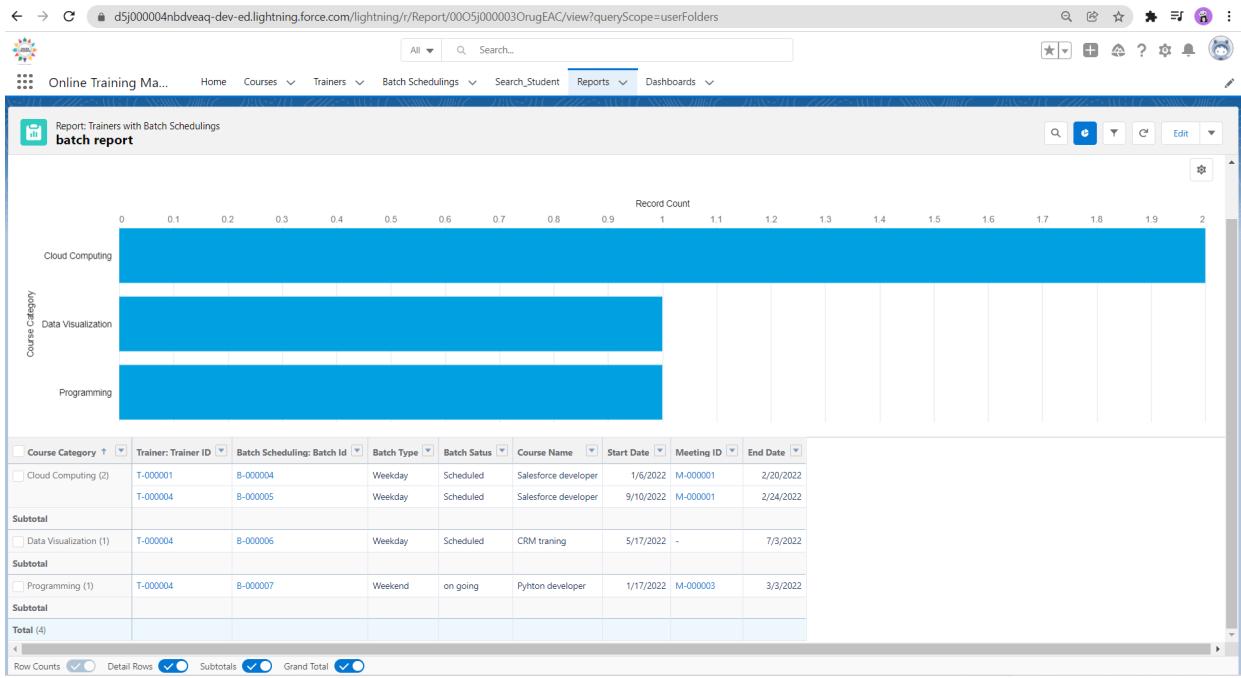
The screenshot shows a Salesforce Lightning interface with a report titled "Course_Detail_Report". The report displays 6 total records in a tabular format. The columns are: Course: Course ID, Course Name, Course Category, Course: Created Date, Duration (in Months), and Course Fee. The data is as follows:

	Course: Course ID	Course Name	Course Category	Course: Created Date	Duration (in Months)	Course Fee
1	C-000002	Python Developer	Programming	1/9/2022	3	\$1,000
2	C-000004	CRM traning	Data Visualization	1/11/2022	3	\$500
3	C-000006	CRM traning	Data Visualization	1/11/2022	3	\$500
4	C-000003	Pyhton developer	Programming	1/11/2022	3	\$1,500
5	C-000005	CRM traning	Data Visualization	1/11/2022	3	\$500
6	C-000001	Salesforce developer	Cloud Computing	1/4/2022	2	\$1,000

this report gives over all details of course fields in tabular form.

2. Report on Batch Scheduling object.

This report gives the bar graph representation of the batches for specific courses.



3. Report on Trainer object:

This report gives the representation of trainers fields and related batch and meeting information in pie chart representation.

Trainer: ID	Full Name	Trainer: Trainer ID	Batch Scheduling: Batch Id	Batch Type	Meeting ID	Course Category
a01500000898dm	Peter	T-000001	B-000004	Weekday	M-000001	Cloud Computing
a015000008AGFC	Mike	T-000002				
a015000008c0w1	priya	T-000004	B-000005	Weekday	M-000001	Cloud Computing
			B-000007	Weekend	M-000003	Programming
			B-000006	Weekday	-	Data Visualization
Total				Count: 3		
				Count: 4		

Row Counts Detail Rows Subtotals Grand Total

4. Created Training Detail report as the Dashboard page.
 Here, this dashboard gives entire information of the Online Training Management page.

Course: Course ID	Course Name	Course Category	Course: Created Date	Duration (in Months)
C-00001	Salesforce developer	Cloud Computing	1/4/2022	2
C-00002	Python Developer	Programming	1/5/2022	3
C-00003	Python developer	Programming	1/11/2022	3
C-00004	CRM traning	Data Visualization	1/11/2022	3
C-00005	CRM traning	Data Visualization	1/11/2022	3
C-00006	CRM traning	Data Visualization	1/11/2022	3

View Report (Trainer Detail Report)

View Report (batch report)

Day 11

Topic:Application building

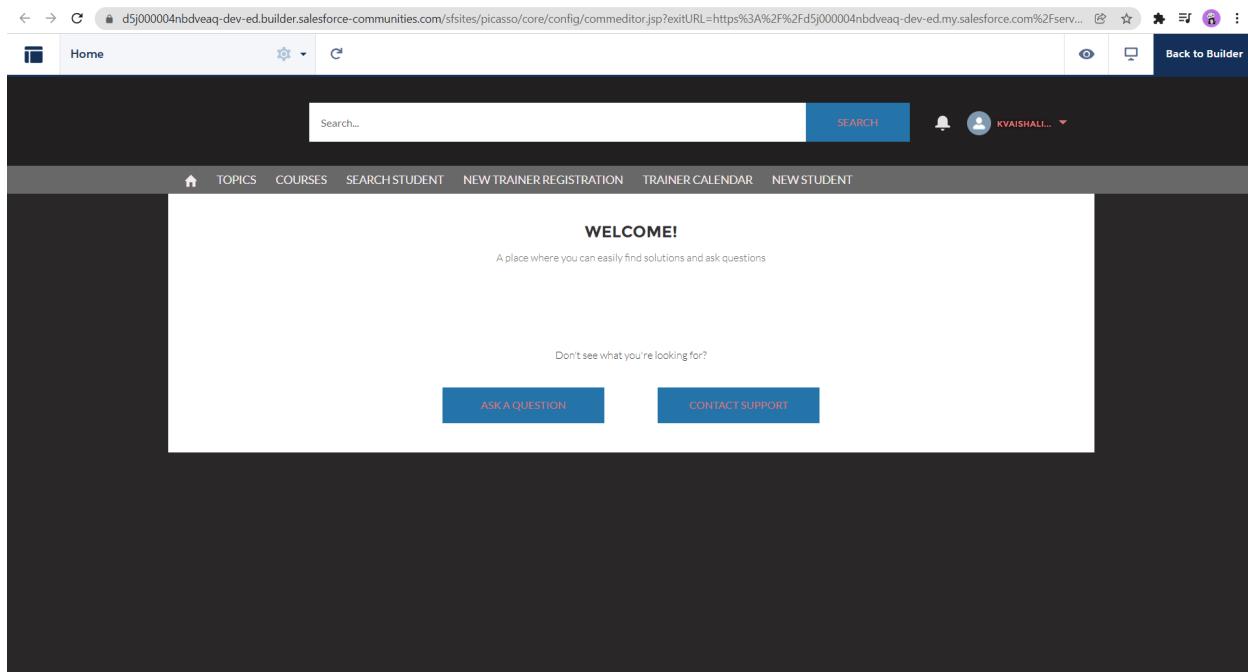
Activities and Screenshots:

1.Create a site using a unique domain name:

I created a Site Named:"Onlinetrainingedu".

Using builder we can able to create individual page of objects and add to the navigation bar and customize the application.

-->thus we can add different object as below.



Here ,i customized the page by dark color themes, and added courses search student,new trainer registration,trainercalendar,new student registration VF pages.

Day 12

Topic:Customizations the application.

Activities and Screenshots:

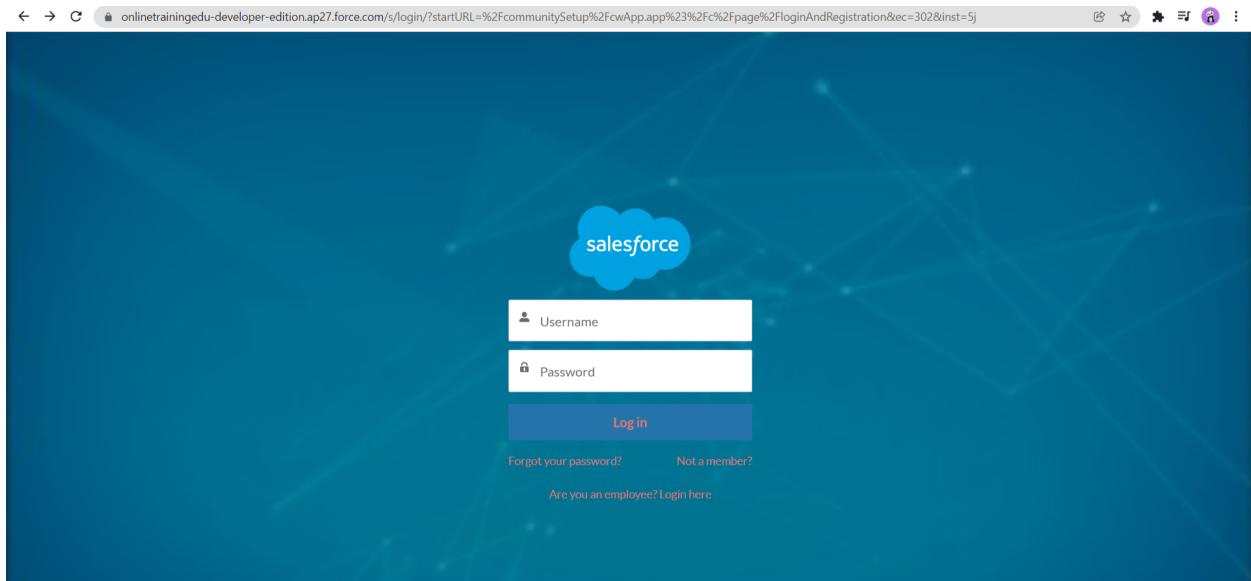
1.Security Customization on the application using Administration part by allowing system administrator and online custom plus user to authenticate.

The screenshot shows the Salesforce Administration interface for the 'Members' section. On the left, a sidebar lists various administration categories: Settings, Preferences, Members (which is selected and highlighted in blue), Contributors, Login & Registration, Emails, Pages, Rich Publisher Apps, and URL Redirects. The main content area is titled 'Members' and contains a note: 'All users with the selected profiles OR permission sets are members.' Below this is a 'Select Profiles' section with two panels: 'Available Profiles' (listing standard profiles like Analytics Cloud Integration User, Chatter Moderator User, Contract Manager, etc.) and 'Selected Profiles' (listing 'Online Custom plus User' and 'System Administrator'). There are 'Add' and 'Remove' buttons between the panels. Below this is a 'Select Permission Sets' section with similar 'Available' and 'Selected' panels, both currently empty. The bottom of the page shows the URL: <https://onlinetrainingedu-developer-edition.ap27.force.com/communitySetup/cwApp.app#/c/page/members>.

2.login and registration page settings:

The screenshot shows the Salesforce Administration interface with the URL onlinetrainingedu-developer-edition.ap27.force.com/communitySetup/cwApp.app#/c/page/loginAndRegistration. The left sidebar has a 'Login & Registration' section selected. The main content area is titled 'Login & Registration' and contains a 'Branding Options' section. It includes fields for 'Logo Type' (File), 'Logo File' (Choose file, No file chosen, JPG, GIF or PNG, 100 KB max), and a preview of the Salesforce logo. Other branding options include 'Background Type' (Color), 'Background' (#B1BAC1), 'Login Button' (#1797C0), 'Right Frame URL', and 'Footer Text' (salesforce.com). A 'Login Page Setup' section is also present.

3.The login page:



my link for this site is "<https://onlinetrainingedu-developer-edition.ap27.force.com/s/>"