

# Project: Online Training Management - Developer

## Day-1:

Topic: CREATE YOUR SALESFORCE DEVELOPER ORG TO GET STARTED

Milestone: Creating Developer Account, Account Activation, Login to your Salesforce Account

### Description:

Creating a developer org in salesforce.

Go to [developers.salesforce.com/](https://developers.salesforce.com/)

Click on sign up.

On the sign up form, enter the following details :

First name & Last name

Email

Role : Developer

Company : College Name

Country : India

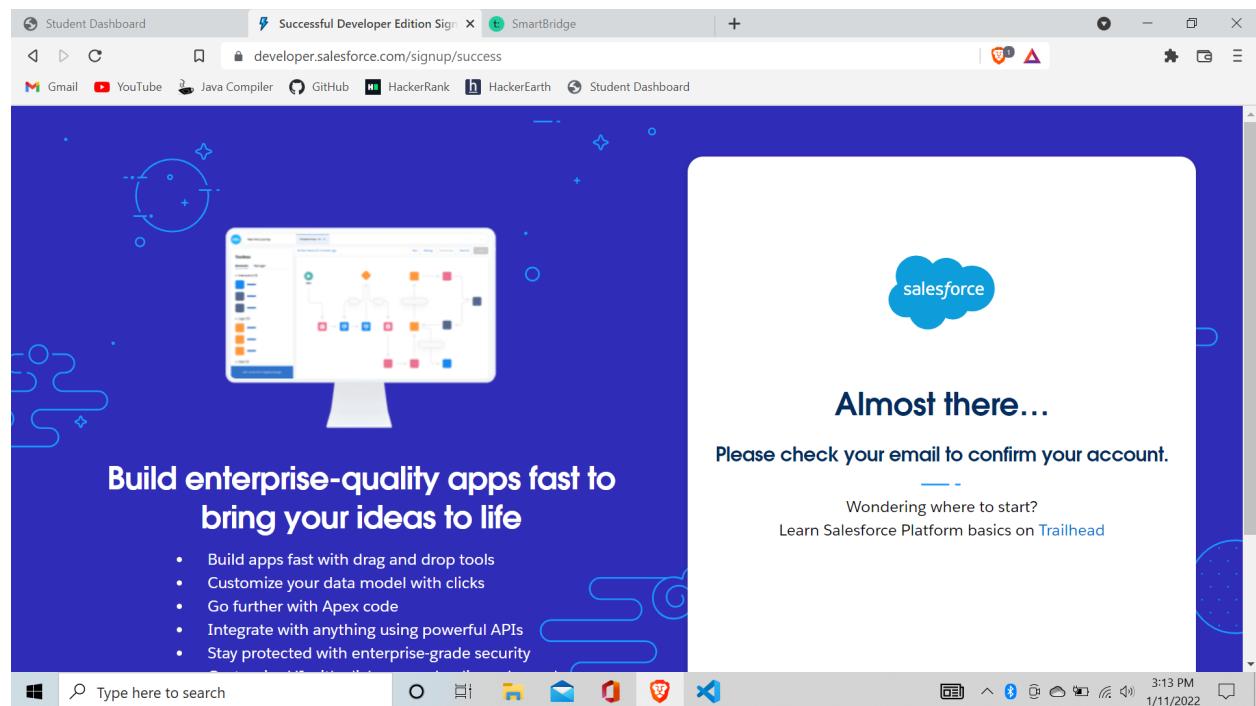
Postal Code : pin code

Username : should be a combination of your name and company

This need not be an actual email id, you can give anything in the format :

[username@organization.com](mailto:username@organization.com)

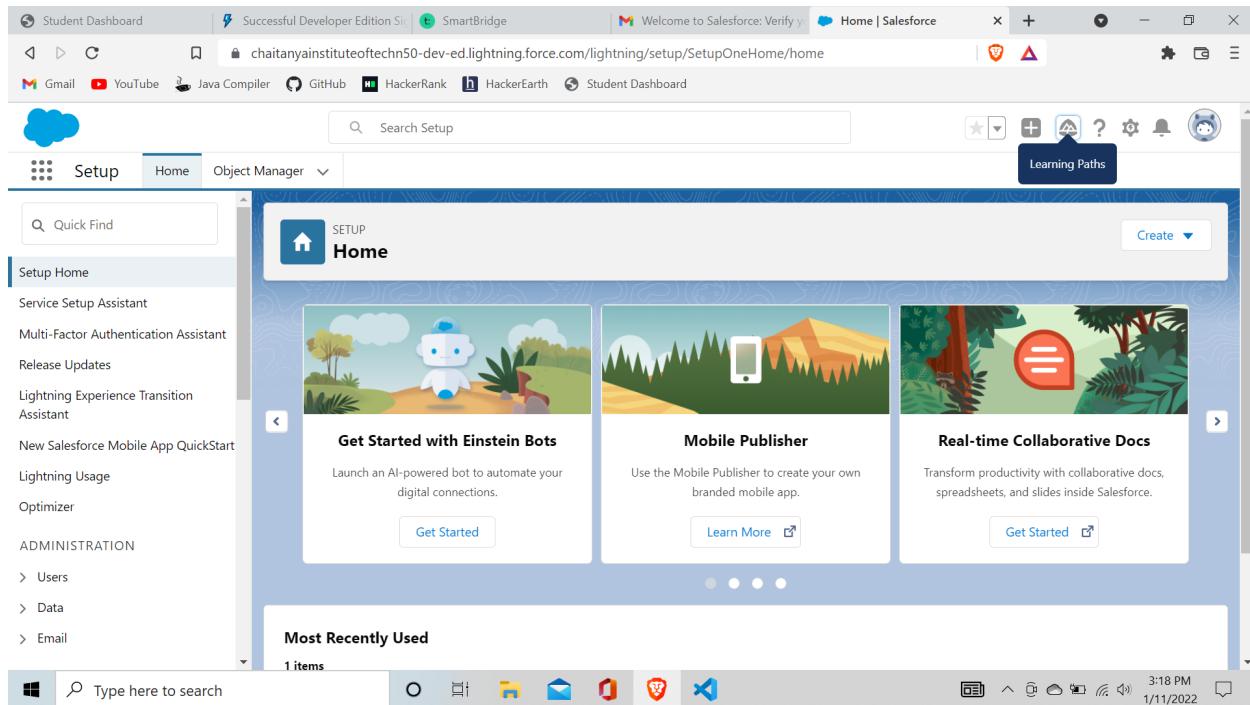
Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account.



Go to salesforce.com and click on login.

Enter the username and password that you just created.

After login this is the home page which you will see.



## Day-2:

Topic: CUSTOM OBJECT CREATION

Milestone: Create the custom objects for the application, Create custom fields on Course Object, Trainer Object, Student Object, Meeting Object and Batch Object.

### Description:

Custom Objects: Custom objects are the objects which are created manually by the users. Below is the video which shows how custom objects are created.

1. After you log in to your salesforce org, create the following custom objects as mentioned above in the video.
2. The custom objects are:
  - a) Courses
  - b) Trainer
  - c) Student
  - d) Meeting
  - e) Batch/Class Scheduling

The screenshot shows a Salesforce Lightning interface with three tabs open: 'Home | Salesforce', 'Batch | Salesforce', and the current tab, 'chaitanyainstituteoftechn50-dev-ed.lightning.force.com/lightning/page/home'. The main content area displays a 'Quarterly Performance' dashboard. It includes a line chart showing revenue trends from January to March. The chart has three data series: 'Closed' (orange), 'Goal' (green), and 'Closed + Open (>70%)' (blue). The blue line starts at 0 in Jan, rises sharply to about 105k in Feb, and remains flat at 105k in Mar. The 'Closed' series is at 105k in Feb and Mar. The 'Goal' series is at 0 in all three months. The 'Closed + Open (>70%)' series is at 105k in all three months. Below the chart, the URL is https://chaitanyainstituteoftechn50-dev-ed.lightning.force.com/lightning/page/home.

**Assistant**

- New lead assigned to you today Mike Braund
- New lead assigned to you today Violet Macleod
- New lead assigned to you today Andy Young
- New lead assigned to you today Phyllis Cotton
- New lead assigned to you today Pamela Owenby
- New lead assigned to you today Bill Dadio Jr
- New lead assigned to you today Carolyn Crenshaw
- New lead assigned to you today Pat Stumiller

After Creating the course object follow the above-mentioned video to create the following fields on the course object.

Field Name	Data Type	Values
Course ID	Auto Number (Change the data type to AutoNumber) While creating course object and changing record Type to course id)	Format(C-000001) Starting number=1
Course Name	Text	
Course Duration	Picklist	1 month, 2 months, 3 months
Course Fees	Currency	
Course Description	Text(Long)	
Course Category	Picklist	Data science, cloud computing, AI, ML
Course Trainer	Master-Detail(Trainer)	

The screenshot shows the Salesforce Setup interface with the following details:

- Page Title:** Home | Salesforce
- Current Page:** Course | Salesforce
- URL:** chaitanyainstituteoftechn50-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01I5j000000xUHZ/FieldsAndRelationships/view
- Header:** Search Setup, Apps, Gmail, YouTube, Maps, Translate, Reading list
- Left Sidebar:** Setup, Home, Object Manager
- Section:** SETUP > OBJECT MANAGER
- Object:** Course
- Section:** Fields & Relationships
- Table Headers:** FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, INDEXED
- Table Data:**

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Course Category	Course_Category__c	Picklist		
Course Description	Course_Description__c	Long Text Area(32768)		
Course Duration	Course_Duration__c	Picklist		
Course Fees	Course_Fees__c	Currency(18, 0)		
Course ID	Name	Auto Number		✓
Course Name	Course_Name__c	Text(50)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Trainer	Trainer__c	Master-Detail(Trainer)		✓
- Bottom Bar:** Type here to search, Taskbar icons (File, Print, Mail, etc.), System status (28°C, 6:20 PM, 1/11/2022)

Create the following fields on trainer object.

Field Name	Data Type	Values
Trainer ID	Autonumber	Format=T-00001,Starting number=1
Full Name	Text	
Email	Email	
PhoneNumber	Phone	
Whatsapp Number	Phone	
DateofBirth	Date	
Age	Formula	TODAY()-Date_of_Birth__c
Experience	Picklist	1year,2years,3years,.....
LinkedIn Profile	URL	
Expertise	Text	

The screenshot shows the Salesforce Setup interface with the URL <https://chaitanya instituteoftechn50-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01I5j000000xUHo/FieldsAndRelationships/view>. The page is titled "Fields & Relationships" for the "Trainer" object. It displays 13 items sorted by Field Label. The columns include FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. Fields listed include Age, Created By, DateofBirth, Email, Experience, Expertise, Full Name, Last Modified, and LinkedIn Profile.

Create the following Custom fields on student object.

Field Name	Data Type	Values
Student ID	Autonumber	Format=S-00001,Starting Number=1
First Name	Text	
Last Name	Text	
Email	Email	
PhoneNumber	Phone	
Date Of Birth	Date	
Whatsapp Number	Phone	
Enrollment Date	Formula	TODAY()
Student Enrollment	Master Detail(Batch)	

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Date of Birth	DateOfBirth_c	Date		
Email	Email_c	Email		
Enrollment Date	EnrollmentDate_c	Formula (Date)		
First Name	FirstName_c	Text(50)		
Last Modified By	LastModifiedById	Lookup(User)		
Last Name	LastName_c	Text(50)		
Phone Number	PhoneNumber_c	Phone		
Student Enrollment	StudentEnrollment_c	Master-Detail(Batch)		

Create the following custom fields on meeting object

Field Name	Data Type	Values
Meeting ID	Autonumber	Format=M-00001, Starting Number=1
Valid From	Date	
Valid To	Date	
Active	Checkbox	
Meeting Link	URL	

The screenshot shows the Salesforce Object Manager interface. The left sidebar is collapsed, and the main area displays the 'Fields & Relationships' section for the 'Meeting' object. The table lists the following fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Active	Active__c	Checkbox		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Meeting ID	Name	Auto Number		✓
Meeting Link	Meeting_Link__c	URL(255)		
Owner	OwnerId	Lookup(User,Group)		✓
Valid From	Valid_From__c	Date		
Valid To	Valid_To__c	Date		

Create the following custom fields on the batch/class scheduling object.

Field Name	Data Type	Values
Batch ID	Auto Number	Format=B-00001 Starting Number=1
Batch Type	Picklist	Weekday, Weekend
Start Date	Date	
End Date	Date	
Course ID	Lookup(Course)	
Meeting id	Lookup(Meeting)	
Trainer id	Lookup(Trainer)	
Status	Picklist	Scheduled/Started/Completed
Number Of Students Enrolled	Rollup Summary(Count)	
Details	Formula(Text)	Course_ID__r.Course_Name__c & Trainer_ID__r.FullName__c & Meeting_ID__r.Meeting_Link__c

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. The main content area displays the 'Fields & Relationships' section for the 'Batch' object. The table lists the following fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Batch ID	Name	Auto Number		✓
Batch Type	Batch_Type__c	Picklist		✗
Course ID	Course_ID__c	Lookup(Course)		✓
Created By	CreatedById	Lookup(User)		✗
Details	Details__c	Formula (Text)		✗
End Date	End_Date__c	Date		✗
Last Modified By	LastModifiedById	Lookup(User)		✗
Meeting Id	Meeting_Id__c	Lookup(Meeting)		✓
Number Of Students Enrolled	Number_of_Students_Enrolled__c	Roll-Up Summary (COUNT Student)		✗

## Day-3:

Topic: CREATE VALIDATION RULES

Milestone: Creating Validation Rules on Course Object, Trainer Object, Student Object, and Meeting Object

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of "True" or "False".

1. Course Fees Cannot be Negative: From Object Manager select the course object and click on validation rules, give your rule a name and in error condition formula enter "Course\_Fee\_\_c<0", give error message according to the condition.

2. Course Fees Should Be Less than 3000\$: From Object Manager select the course object and click on validation rules, give your rule a name and in error condition formula enter "Course\_Fee\_\_c>=3000", give error message according to the condition.

1. Full Name, Phone Number, Date of Birth, Experience fields on the Trainer object should not be blank:  
 Search for Trainer Object from the object manager, click on validation rules, click new and give your rule a name, in error condition formula enter "ISBLANK(Field Name)" and enter the error message according to the validation rule.

2. Age Should Not Be Less than 25 years: Enter "age\_\_c<25" in error condition formula and enter the error message according to the rule.

3. Phone Number and Whatsapp Number should accept minimum 10 digits and maximum 15 digits: In error condition formula enter "NOT(REGEX(FIELDNAME, "[0-9]{10,15}"))" and error message according to the validation rule.

The screenshot shows two screenshots of the Salesforce Lightning Experience interface. The top screenshot displays a Trainer record (T-00003) with various fields like Trainer ID, Full Name, Date of Birth, Experience, and Contact information. The bottom screenshot shows the 'Validation Rules' section for the Trainer object, listing three rules: 'NumberSize' (Error Location: Top of Page, Message: number size should be appropriate), 'TrainerAgeRule' (Error Location: Date of Birth, Message: Trainer Age should not be less than 25), and 'TrainerRule1' (Error Location: Top of Page, Message: Full Name, Phone Number, Date of Birth and Experience should not be null).

Validation Rule	Error Location	Error Message	Active	Modified By
NumberSize	Top of Page	number size should be appropriate	✓	Vinay Diwan, 1/16/2022, 8:26 AM
TrainerAgeRule	Date of Birth	Trainer Age should not be less than 25	✓	Vinay Diwan, 1/16/2022, 8:16 AM
TrainerRule1	Top of Page	Full Name, Phone Number, Date of Birth and Experience should not be null.	✓	Vinay Diwan, 1/16/2022, 8:12 AM

1. Lastname should not be Blank: enter the following error condition formula "ISBLANK(Field name)" and give the error message according to the rule.

2. Minimum 16 years of age: student should be minimum 16 years of age, in error condition formula enter "age\_c<16" and give the error message according to the rule.

3. Phone and whatsappNumber should be minimum 10 digits : in error condition formula enter "NOT(REGEX(Fieldname, "[0-9]{10}"))" and give the error message according to the validation rule.

The screenshot shows a web browser with two tabs open:

- S-00001 | Salesforce**: This tab displays a Student record for 'S-00001'. The 'Details' tab is selected, showing the following data:
  - Student ID: S-00001
  - First Name: Diwan
  - Last Name: Diwan
  - Enrollment Date: 1/16/2022
  - Student Enrollment: B-00001
  - Date Of Birth: 1/16/2000
  - Phone Number: (123) 456-7890
  - Email: (not visible)
  - Created By: Vinay Diwan, 1/16/2022, 8:52 AM
  - Last Modified By: Vinay Diwan, 1/16/2022, 8:55 AM
- Meeting | Salesforce**: This tab is currently active, showing a search bar and various navigation links.

Below the browser windows, the operating system taskbar is visible, showing the date and time (1/16/2022, 10:40 PM), weather (19°C), and system icons.

**Validation Rules Page:**

The 'Validation Rules' page for the 'Student' object is displayed. It lists two validation rules:

Rule Name	Error Location	Error Message	Active	Modified By
lastNameAndAge	Top of Page	Last Name should not be empty and age should be greater than 16	✓	Vinay Diwan, 1/16/2022, 8:49 AM
numberSize	Top of Page	Number size should be appropriate.	✓	Vinay Diwan, 1/16/2022, 8:55 AM

1. Valid From should not be less than current date: Click on validation rules from meeting object, give your rule a name and in error condition formula enter "Valid\_from\_\_c < TODAY()" and enter the error message according to the rule.

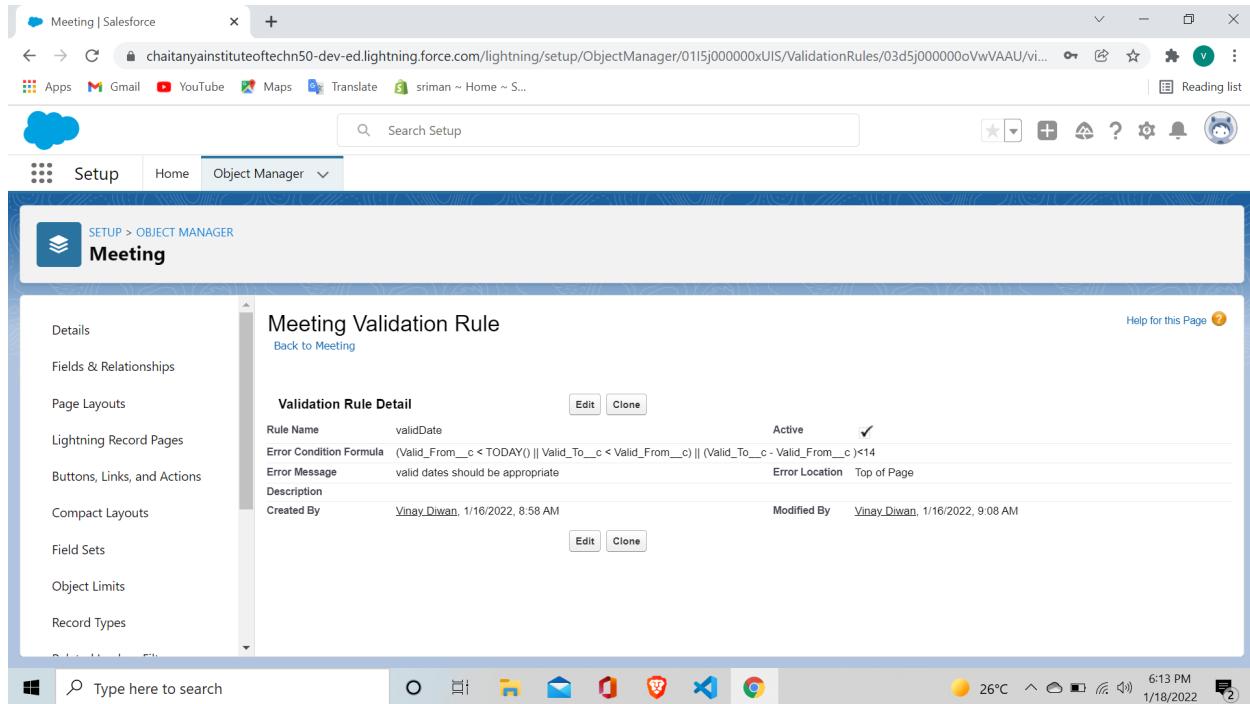
2. ValidTo Cannot be less than Valid from : Click on validation rules from meeting object, give your rule a name and in error condition formula enter "Valid\_to\_\_c < Valid\_from\_\_c " and enter the error message according to the rule.

3. Minimum 14 days difference should be in between valid from and valid to : Click on validation rules from meeting object, give your rule a name and in error condition formula enter "Valid\_to\_\_c - Valid\_from\_\_c < 14 " and enter the error message according to the rule.

The screenshot shows a web browser window with three tabs: 'M-00001 | Salesforce', 'Meeting | Salesforce', and 'New Tab'. The main content area displays a Salesforce Lightning component for a meeting record. The record ID is M-00001. The 'Details' tab is selected. The fields shown are:

- Meeting ID: M-00001
- Active:
- Meeting Link: sdfgsfsdf
- Valid From: 1/16/2022
- Valid To: 1/30/2022
- Created By: Vinay Diwan, 1/16/2022, 9:04 AM
- Owner: Vinay Diwan
- Last Modified By: Vinay Diwan, 1/16/2022, 9:08 AM

The browser's address bar shows the URL: chaitanyainstiuteoftechn50-dev-ed.lightning.force.com/lightning/r/Meeting\_\_c/a085j000001SMdKAAW/view. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time as 1/16/2022 at 10:40 PM.



## Day-4:

Topic: PROCESS AUTOMATION

Milestone: Creating Work Flow Rules on Meeting Object and Batch Object.

Instead of relying on your users to perform each part of a business process, automate it! The benefits are two-fold: your users can spend their time on other tasks, and you can trust that the process is always done just so. Salesforce offers tools to automate several kinds of business processes: guided visual experiences, behind-the-scenes automation, and approval automation. You'll be happy to know that these tools don't require you to write code—they're all point-and-click.

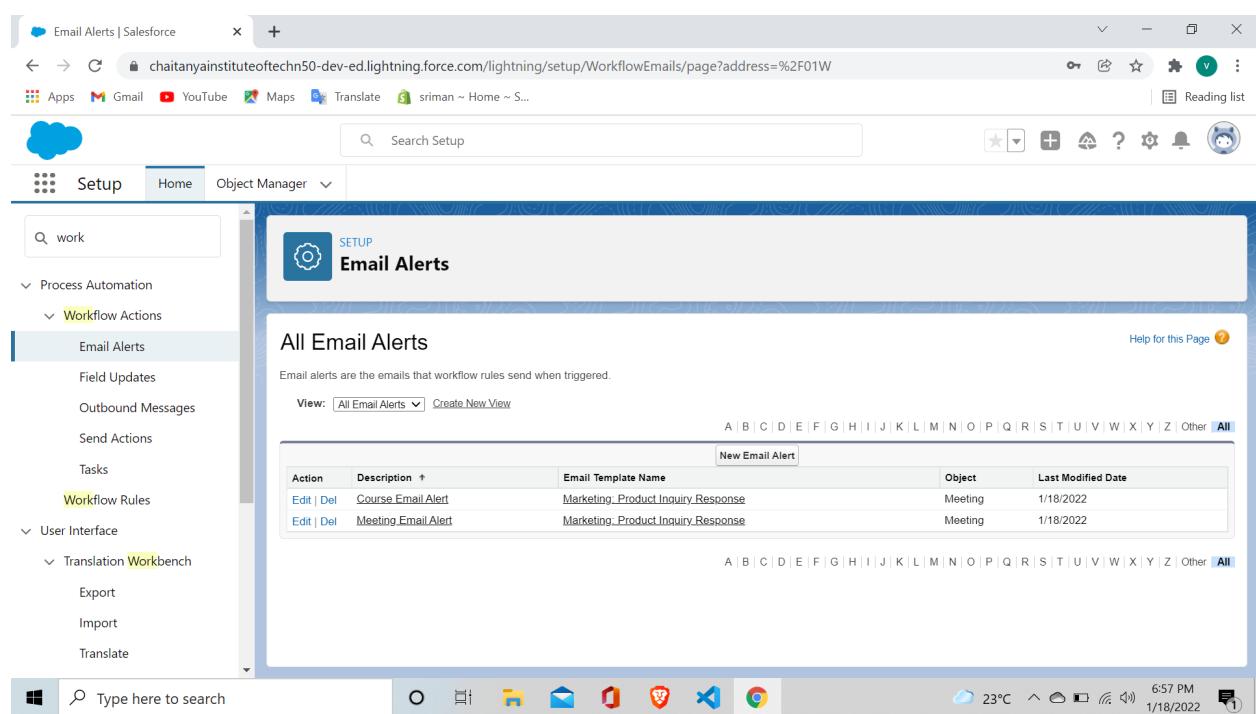
There are four automation tools in salesforce:

1. Approval Process
2. Process builder
3. Flows
4. Workflows

1. create work flow rule 1 day after valid to date the active check box will uncheck automatically and inform owner:

- a) from set up search for work flow rules, click on new rule select meeting object click next.
  - b) enter rule name , in evaluation criteria select created, and any time it's edited to subsequently meet criteria, in rule criteria select formula evaluates to true.
  - c) in formula section " TODAY()- VALID\_TO\_\_C= 1" click save.
  - d) from work flow action select field update.
  - e) give a name for your work flow action, select active field and for specify new field value select false.click save.
- 2.create work flow rule 1 day before valid to date send a reminder email to owner:

- a) from set up search for work flow rules, click on new rule select meeting object click next.
  - b) enter rule name , in evaluation criteria select created, and any time it's edited to subsequently meet criteria, in rule criteria select formula evaluates to true.
  - c) in formula section " VALID\_TO\_\_C - TODAY () = 1" click save.
  - d) from work flow action select email alert.
  - e) enter all mandatory details , select an email template, search for owner in recipient type and select the owner from the list click save
- Make sure to activate the rules.



The screenshot shows the Salesforce Setup interface with the following details:

- Page Title:** Email Alerts | Salesforce
- URL:** chaitanyainsteinstituteoftechn50-dev-ed.lightning.force.com/lightning/setup/WorkflowEmails/page?address=%2F01W
- Header:** Includes standard browser controls, a search bar, and a reading list icon.
- Left Navigation Bar:**
  - Setup (selected)
  - Home
  - Object Manager
  - Q: work
  - Process Automation
    - Workflow Actions (selected)
    - Email Alerts
    - Field Updates
    - Outbound Messages
    - Send Actions
    - Tasks
    - Workflow Rules
  - User Interface
    - Translation Workbench
    - Export
    - Import
    - Translate
- Content Area:**

## SETUP Email Alerts

### All Email Alerts

Email alerts are the emails that workflow rules send when triggered.

View: All Email Alerts | Create New View

Action	Description	Email Template Name	Object	Last Modified Date
Edit   Del	Course Email Alert	Marketing_Product Inquiry Response	Meeting	1/18/2022
Edit   Del	Meeting Email Alert	Marketing_Product Inquiry Response	Meeting	1/18/2022

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All
- Bottom:** Windows taskbar with search bar, pinned icons (File Explorer, Mail, Microsoft Edge, etc.), and system status indicators (temperature, battery, date/time).

The screenshot shows the Salesforce Setup interface with the following details:

- Left Sidebar:** Shows the navigation tree under "Process Automation" with "Workflow Actions" expanded, listing Email Alerts, Field Updates, Outbound Messages, Send Actions, and Tasks.
- Main Content Area:**
  - Section Header:** SETUP Workflow Rules
  - Section:** All Workflow Rules
  - Description:** Configure your organization's workflow by creating workflow rules. Each workflow rule consists of:
    - Criteria that cause the workflow rule to run
    - Immediate actions that execute when a record matches the criteria. For example, Salesforce can automatically send an email that notifies the account team when a new high-value opportunity is created.
    - Time-dependent actions that queue when a record matches the criteria, and execute according to time triggers. For example, Salesforce can automatically send an email reminder to the account team if a high-value opportunity is still open ten days before the close date.
  - Quick Tips:**
    - Useful Sample Workflow Rules
    - Video Tutorial (English Only)
    - Troubleshooting Workflow
  - Table:** Shows the list of workflow rules with columns: Action, Rule Name, Description, Object, and Active. Two rules are listed:
 

Action	Rule Name	Description	Object	Active
Edit   Del   Deactivate	Meeting WorkFlow		Meeting	✓
Edit   Del   Deactivate	Meeting WorkFlow.2		Meeting	✓
- Bottom:** A search bar with "Type here to search" and a toolbar with various icons.

1.create work flow rule 2 days before valid to date send a reminder email to owner:

- from set up search for work flow rules, click on new rule select meeting object click next.
- enter rule name , in evaluation criteria select created, and any time it's edited to subsequently meet criteria, in rule criteria select formula evaluates to true.
- in formula section " VALID\_TO\_\_C - TODAY () = 2" click save.
- from work flow action select email alert.
- enter all mandatory details , select an email template, search for owner in recipient type and select the owner from the list click save

Make sure to activate the rules.

The screenshot shows the Salesforce Setup interface with the 'Workflow Rules' page open. The sidebar on the left is collapsed, showing categories like Process Automation, Workflow Actions, User Interface, etc. The main content area has a title 'Workflow Rules' with a 'SETUP' icon. Below it is a section titled 'All WORKFLOW RULES' with a brief description of what workflow rules consist of. A 'Quick Tips' box contains links to 'Useful Sample Workflow Rules', 'Video Tutorial (English Only)', and 'Troubleshooting Workflow'. A table lists three workflow rules:

Action	Rule Name	Description	Object	Active
Edit   Del   Deactivate	BatchWorkFlow		Meeting	✓
Edit   Del   Deactivate	Meeting WorkFlow		Meeting	✓
Edit   Del   Deactivate	Meeting WorkFlow_2		Meeting	✓

## Day-5:

Topic: CREATING CUSTOM PAGES USING VISUALFORCE

**Milestone:** Creating New Course Using Custom Controller, Creating New Trainer Using Custom Controller, Creating New Student Using Standard Controller, Creating New Meeting Using Standard Controller, Creating New Batch Using Standard Controller.

**Visualforce:** Visualforce is a framework that allows developers to build sophisticated, custom user interfaces that can be hosted natively on the Lightning platform. The Visualforce framework includes a tag-based markup language, similar to HTML, and a set of server-side “standard controllers” that make basic database operations, such as queries and saves, very simple to perform.

In the Visualforce markup language, each Visualforce tag corresponds to a coarse or fine-grained user interface component, such as a section of a page, a related list, or a field. The behavior of Visualforce components can either be controlled by the same logic that is used in standard Salesforce pages, or developers can associate their own logic with a controller class written in Apex.

course visualforce page code

```
<apex:page lightningStylesheets="true" controller="course">
<apex:form>
<apex:sectionHeader title="New Course"/>
<apex:pageMessages></apex:pageMessages>
<apex:pageBlock>
<apex:pageBlockSection>
<apex:inputfield value="{!cr.Course_Name__c}"/>
```

```

<apex:inputfield value="{!!cr.Course_Category__c}" />
<apex:inputfield value="{!!cr.Course_Fee__c}" />
<apex:inputfield value="{!!cr.Course_Duration__c}" />
<apex:inputfield value="{!!cr.Course_Description__c}" />
</apex:pageBlockSection>
<center><apex:commandButton value="Save" action="{!!save}" /></center>
</apex:pageBlock>
</apex:form>
</apex:page>

```

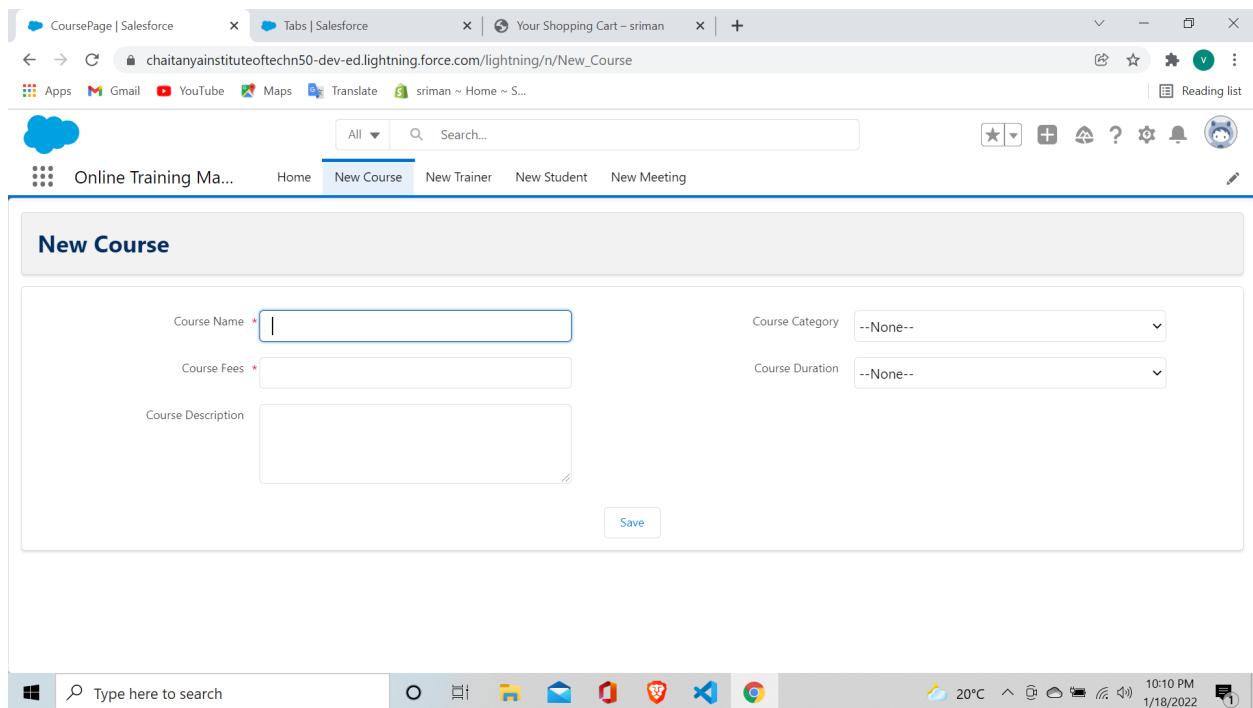
course custom controller code:

```

public with sharing class course
{
    public Course__c cr{get;set;}
    public course()
    {
        cr=new Course__c();
    }

    public void save() {
        if(cr.Course_Fee__c <0 || cr.Course_Fee__c>3000 )
        {
            ApexPages.addmessage(new ApexPages.message(ApexPages.severity.Error, 'Course Fee Should be
less Than 3000 and Cannot Be Negative'));
        }
        else{
            insert cr;
            ApexPages.addmessage(new ApexPages.message(ApexPages.severity.Confirm, 'Record Inserted
Successsfully'));
            cr=new Course__c();
        }
    }
}

```



Trainer visualforce code:

```
<apex:page lightningStylesheets="true" controller="trainer">
<apex:form>
<apex:sectionHeader title="New Trainer"/>
<apex:pageMessages></apex:pageMessages>
<apex:pageBlock>
<apex:pageBlockSection>
<apex:inputfield value="{!tr.FullName__c}" required="true"/>
<apex:inputfield value="{!tr.Email__c}" required="true"/>
<apex:inputfield value="{!tr.Phone_Number__c}" required="true"/>
<apex:inputfield value="{!tr.Whatsapp_Number__c}"/>
<apex:inputfield value="{!tr.Date_Of_Birth__c}" required="true"/>
<apex:inputfield value="{!tr.Experience__c}" required="true"/>
<apex:inputfield value="{!tr.LinkedIN_Profile__c}"/>
<apex:inputfield value="{!tr.Expertise__c}"/>
<apex:inputField value="{!tr.Age__c}"/>
</apex:pageBlockSection>
<center><apex:commandButton value="Save" action="{!save}"/></center>
</apex:pageBlock>
</apex:form>
</apex:page>
```

Trainer controller code:

```

public with sharing class Trainer {
    public Trainer__c tr{get; set;}
    public trainer()
    {
        tr=new Trainer__c();
    }
    public void save()
    {
        insert tr;
        ApexPages.addmessage(new ApexPages.message(ApexPages.severity.Confirm, 'Record Inserted
Successsfully'));
        tr=new Trainer__c();
    }
}

```

The screenshot shows a Salesforce Lightning page titled "New Trainer". The page contains the following fields:

- Full Name \*
- Email \*
- PhoneNumber \*
- Whatsapp Number
- DateofBirth \*
- Experience \* --None--
- LinkedIn Profile
- Expertise
- Age

A "Save" button is located at the bottom right of the form.

### Student visualforce code

```

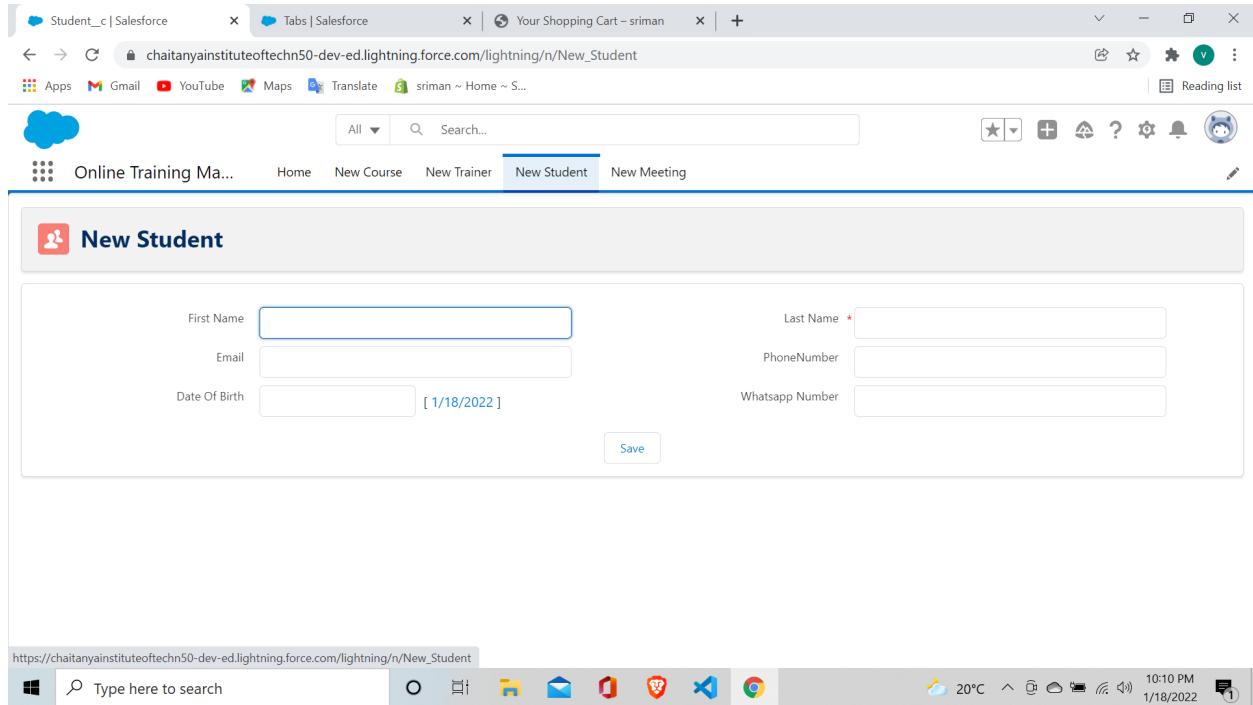
<apex:page lightningStylesheets="true" standardController="Students__c">
    <apex:form>
        <apex:sectionHeader title="New Student"/>
        <apex:pageBlock>
            <apex:pageBlockSection>
                <apex:inputField value="{!!Students__c.First_Name__c}"/>
                <apex:inputField value="{!!Students__c.Last_Name__c}" required="true"/>

```

```

<apex:inputField value="{!!Students__c.Email__c}" />
<apex:inputField value="{!!Students__c.Phone_Number__c}" />
<apex:inputField value="{!!Students__c.Date_Of_Birth__c}" />
<apex:inputField value="{!!Students__c.Whatsapp_Number__c}" />
</apex:pageBlockSection>
<center><apex:commandButton value="Save" action="{!!save}" /></center>
</apex:pageBlock>
</apex:form>
</apex:page>

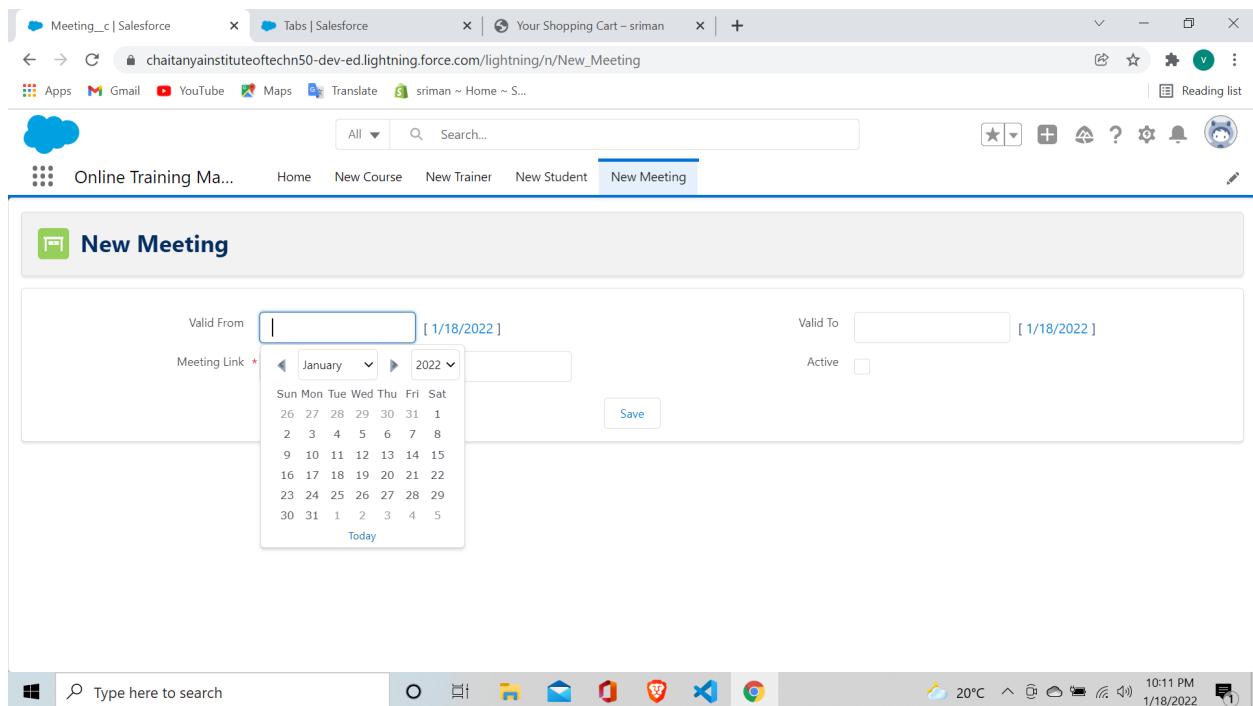
```



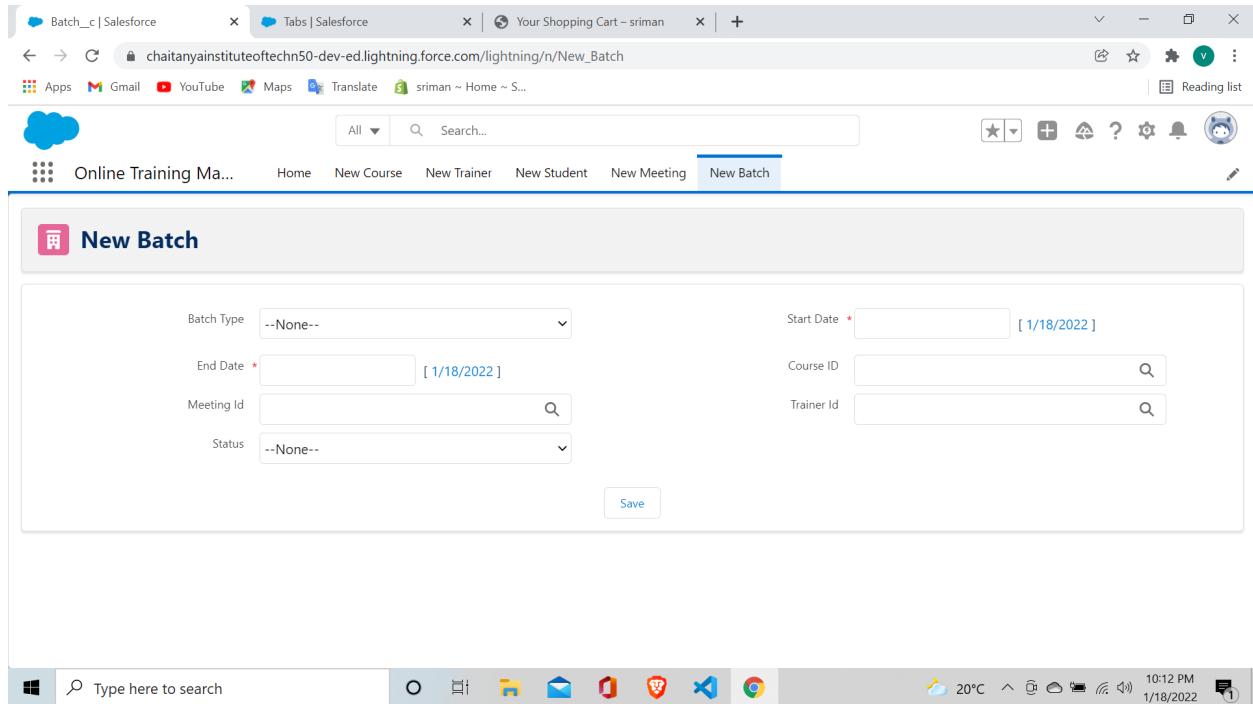
```

<apex:page lightningStylesheets="true" standardController="Meeting__c">
<apex:form>
<apex:sectionHeader title="New Meeting"/>
<apex:PageBlock>
<apex:PageBlockSection>
<apex:inputField value="{!!Meeting__c.Valid_From__c}" />
<apex:inputField value="{!!Meeting__c.Valid_To__c}" />
<apex:inputField value="{!!Meeting__c.Meeting_Link__c}" />
<apex:inputField value="{!!Meeting__c.Active__c}" />
</apex:PageBlockSection>
<center><apex:commandButton value="Save" action="{!!save}" /></center>
</apex:PageBlock>
</apex:form>
</apex:page>

```



```
<apex:page lightningStylesheets="true" standardController="Batch__c">
    <apex:form>
        <apex:sectionHeader title="New Batch"/>
        <apex:pageBlock>
            <apex:pageBlockSection>
                <apex:inputField value="={!Batch__c.Batch_Type__c}" />
                <apex:inputField value=" {!Batch__c.Start_Date__c}" />
                <apex:inputField value=" {!Batch__c.End_Date__c}" />
                <apex:inputField value=" {!Batch__c.Course_ID__c}" />
                <apex:inputField value=" {!Batch__c.Meeting_ID__c}" />
                <apex:inputField value=" {!Batch__c.Trainer_ID__c}" />
                <apex:inputField value=" {!Batch__c.Status__c}" />
            </apex:pageBlockSection>
            <center><apex:commandButton value="Save" action=" {!save}" /></center>
        </apex:pageBlock>
    </apex:form>
</apex:page>
```



## Day-6:

Topic: TRIGGERS

Milestone: Trigger on Trainer Object, Student Object, New Meeting Object.

A trigger is an Apex script that executes before or after data manipulation language (DML) events occur. Apex triggers enable you to perform custom actions before or after events to record in Salesforce, such as insertions, updates, or deletions. Just like database systems support triggers, Apex provides trigger support for managing records.

1. Create the following trigger on trainer object.

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');  
        }  
    }  
}
```

The screenshot shows a developer console window in Google Chrome. The URL is [chaitanya.instituteoftechn50-dev-ed.my.salesforce.com/\\_ui/common/apex/debug/ApexCSIPage](https://chaitanya.instituteoftechn50-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The trigger code is as follows:

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

The 'Problems' tab is selected in the bottom navigation bar. The status bar at the bottom right shows the date as 1/19/2022 and the time as 12:22 AM.

Create the following trigger on student object.

trigger ageCheck on Students\_\_c (before insert)

```
{  
    for(Students__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');  
        }  
    }  
}
```

```

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');
        }
    }
}

```

The screenshot shows a developer console window in Google Chrome. The URL is [chaitanya.instituteoftechn50-dev-ed.my.salesforce.com/\\_ui/common/apex/debug/ApexCSIPage](https://chaitanya.instituteoftechn50-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The trigger code is displayed in the editor. On the right side, there is a tooltip for the 'Bluetooth' icon with instructions: 'Turn on Bluetooth even faster' and 'To turn Bluetooth on or off without opening Settings, open action center and select the Bluetooth icon.' Below the tooltip, there are links for 'Related settings', 'Devices and printers', 'Sound settings', 'Display settings', 'More Bluetooth options', and 'Send or receive files via Bluetooth'. At the bottom of the screen, there is a Windows taskbar with various icons and a search bar.

Create the following trigger on meeting object.

```

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 one
meeting');
        }
    }
}

```

```
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 should be greater than valid from date');
        }
        Integer noOfDays = (mt.Valid_From__c).daysBetween(mt.Valid_To__c);
    }
}
```

### Conclusion:

In this project I have learnt many things like how to create an application, how to create the custom objects and aslo the various types to create an custom objects like by using schema builder, creating validation rules for the custom objects, process automation in the salesforce, creating custom pages in the visualforce by using standard and custom controllers and also triggers. I have got many errors while doing the tasks but with the help trainer and the mentor, errors were solved by the next session and the project has been done successfully in a short period of time. Now I have got experience to perform in various project without any errors. This project had given me the realtime experience in how to work in a company organization.

Thank You  
Vinay Diwan