

ONLINE TRAINING MANAGEMENT

Project Title: Online Training Management - Development

Day 1:

Topic 1: Creating Developer Account

Milestones / Activities: Creating a Developer org Account in salesforce

Detailed Description: We need to go to developers.salesforce.com/ site and need to sign in by entering required details like

1. First name & Last name
2. Email
3. Role
4. Company
5. Country
6. Postal Code
7. Username

Upload the Screenshot the Milestone / Activities :



The screenshot shows the sign-up page for the Salesforce Developer Edition. At the top, it says "Sign up for your Salesforce Developer Edition" and "A full-featured copy of the Platform, for free". Below this are several input fields: "First Name*" and "Last Name*", both with placeholder text "Your first name" and "Your last name". There is also a "Email*" field with placeholder "Your email address". The "Role*" field contains "Your job role" and has a dropdown arrow. The "Company*" field contains "Company Name". The "Country*" field contains "Select Country" and has a dropdown arrow. The "Postal Code*" field contains "Your postal code". Finally, there is a "Username*" field with placeholder "Ex. name@yourcompany.com".

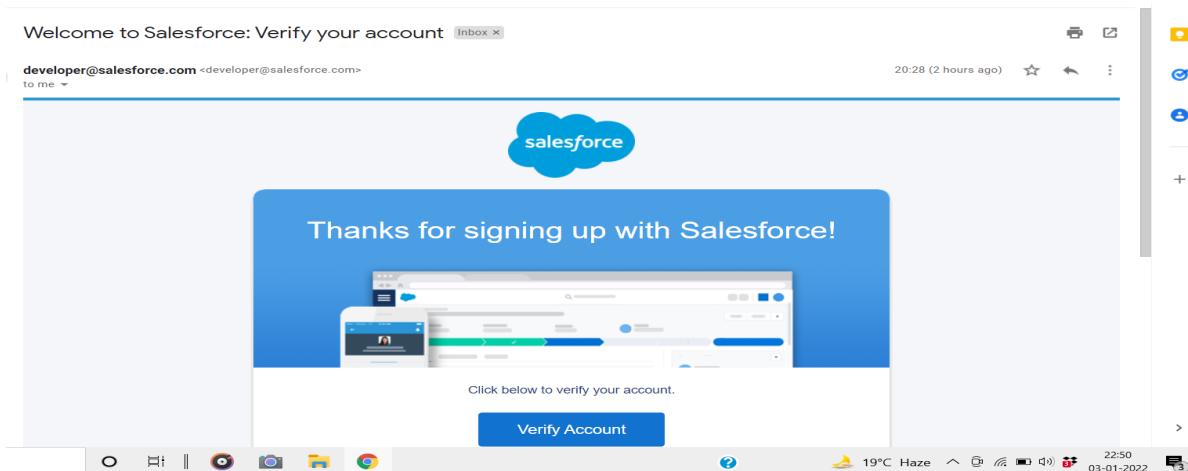
Topic 2: Account Activation

Milestones / Activities: Activating our developer org account.

Detailed Description: We need to activate our developer org account by verifying our email.

Upload the Screenshot the Milestone / Activities :

ONLINE TRAINING MANAGEMENT

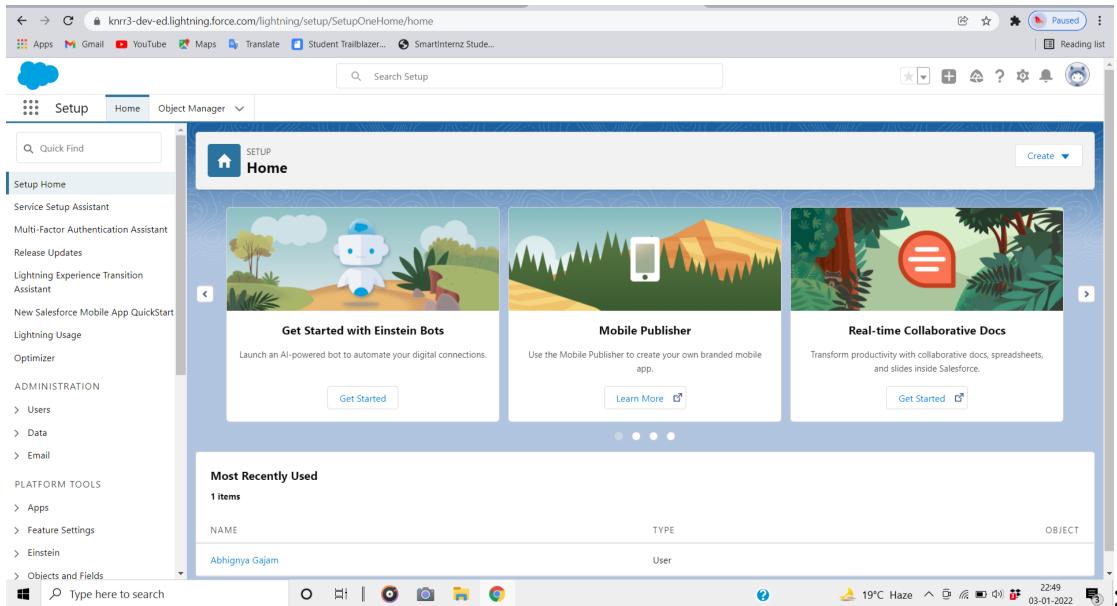


Topic 3: Login to your Salesforce Account

Milestones / Activities: Logging into our Salesforce Account.

Detailed Description: We have to go to salesforce.com and click on login. Enter the username and password that we just created. After logging in, this is the home page which we will see.

Upload the Screenshot the Milestone / Activities:



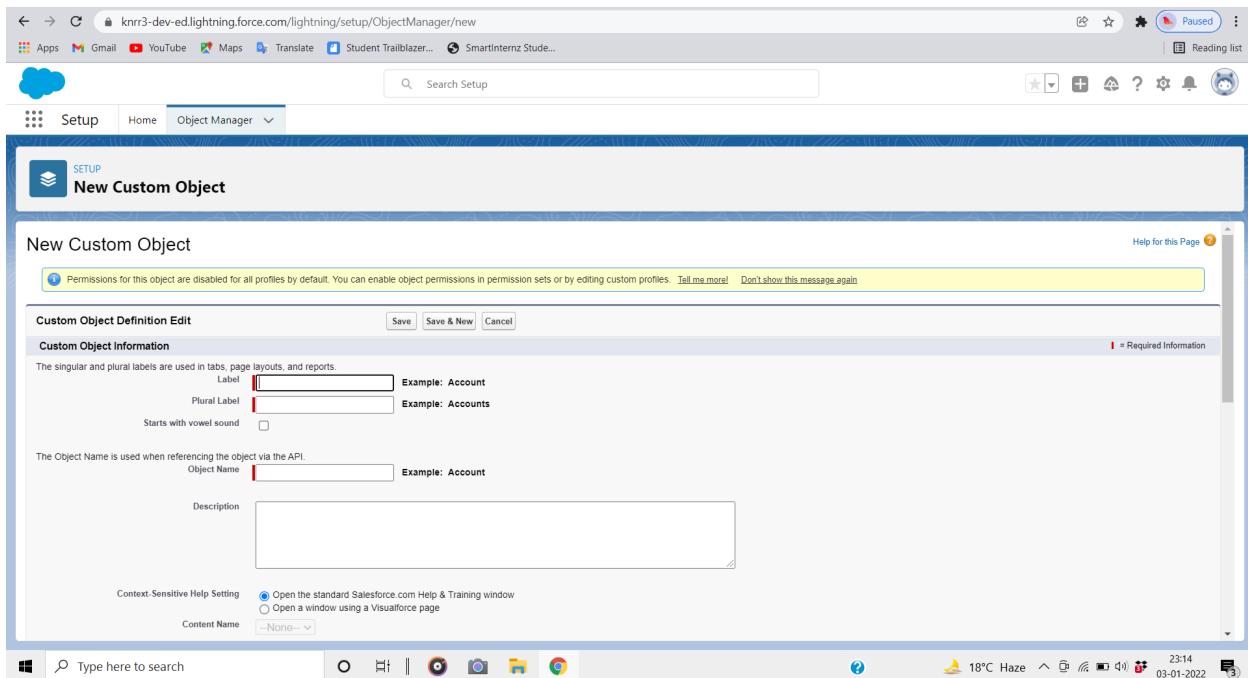
Topic 4: Create The Custom Objects For The Application

Milestones / Activities: Creating the custom objects for the application.

Detailed Description: The custom object I have created are: Courses, Trainer, Student, Meeting, Batch/Class Scheduling.

Upload the Screenshot the Milestone / Activities:

ONLINE TRAINING MANAGEMENT



Day 2:

Topic 1: Create Custom Fields On Course Object

Milestones / Activities: Creating Custom Fields On Course Object

Detailed Description: To create the 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-000000) Starting number=1
Course Name	Text	
Course Duration	Picklist	1month,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)	

ONLINE TRAINING MANAGEMENT

Topic 2: Create Custom Fields On Trainer Object

Milestones / Activities: Creating Custom Fields On Trainer Object

Detailed Description: To create the fields on the 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	

Topic 3: Create Custom Fields On Student Object

Milestones / Activities: Creating Custom Fields On Student Object

Detailed Description: To Create 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()

ONLINE TRAINING MANAGEMENT

Topic 4: Create Custom Fields On Meeting Object

Milestones / Activities: Creating Custom Fields On Meeting Object

Detailed Description: To Create 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	

Topic 5: Create Custom Fields On Batch Object

Milestones / Activities: Creating Custom Fields On Batch Object

Detailed Description: To Create Custom Fields On Batch 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

ONLINE TRAINING MANAGEMENT

Upload the Screenshot the Milestone / Activities : Inserting the Screenshot of schema builder where it shows all custom fields on each custom object clearly that we created today.

Topic 6: Create Validation Rules On Course Object

Milestones / Activities: Creating Validation Rules On Course Object

Detailed Description: In error condition formula enter ,

Course Fees Cannot be Negative: Enter "Course_Fee__c<0"

Course Fees Should Be Less than 3000\$: Enter "Course_Fee__c>=3000".

Upload the Screenshot the Milestone / Activities:

The screenshot shows the Salesforce Object Manager interface for the 'Course' object. The left sidebar lists various customization options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, Search Layouts for Salesforce Classic, and Restriction Rules. The main content area is titled 'Validation Rules' and displays one item: 'Course_Fee_Validation_Rule'. The table columns are RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. The rule details are: Rule Name: Course_Fee_Validation_Rule, Error Location: Course Fee, Error Message: Course Fee Cannot be less than 0 or Greater than 3000, Active: checked, Modified By: Abhignya Gajam, Date: 1/4/2022, 2:51 AM.

ONLINE TRAINING MANAGEMENT

Topic 7 : Create Validation Rules On Trainer Object

Milestones / Activities: Creating Validation Rules On Trainer Object

Detailed Description: Full Name, Phone Number, Dateof Birth , Experience fields on the trainer object should not be blank: In error condition formula enter "ISBLANK(Field Name)" and give the error message according to the rule.

Age Should Not Be Less than 25 years: Enter "age__c<25" in error condition formula and give the error message according to the rule.

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 give the error message according to the rule.

Upload the Screenshot the Milestone / Activities:

The screenshot shows the Salesforce Object Manager interface for the 'Trainer' object. The left sidebar lists various object settings like Details, Fields & Relationships, Page Layouts, etc. The main content area displays a table titled 'Validation Rules' with the following data:

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Age_cannot_be_blank	Date of Birth	Age cannot be blank	✓	Abhignya Gajam, 1/4/2022, 4:52 AM
Date_of_birth_cannot_be_blank	Date of Birth	Date of birth cannot be blank	✓	Abhignya Gajam, 1/4/2022, 4:48 AM
Full_Name_Validation_Rule	Full Name	Full Name Cannot be blank	✓	Abhignya Gajam, 1/4/2022, 4:46 AM
Phone_cannot_be_blank_Validation_Rule	Phone Number	Phone Number should accept minimum of 10 digits and maximum of 15 digits	✓	Abhignya Gajam, 1/4/2022, 4:54 AM
Phone_Number_Validation_Rule	Phone Number	Phone Number Cannot be blank	✓	Abhignya Gajam, 1/4/2022, 4:47 AM
Whatsapp_Number_Validation_Rule	Whatsapp Number	Whatsapp Number should accept a minimum of 10 digits and a maximum of 15 digits	✓	Abhignya Gajam, 1/4/2022, 4:55 AM

Topic 8 : Create Validation Rules On Student Object

Milestones / Activities: Creating Validation Rules On Student Object

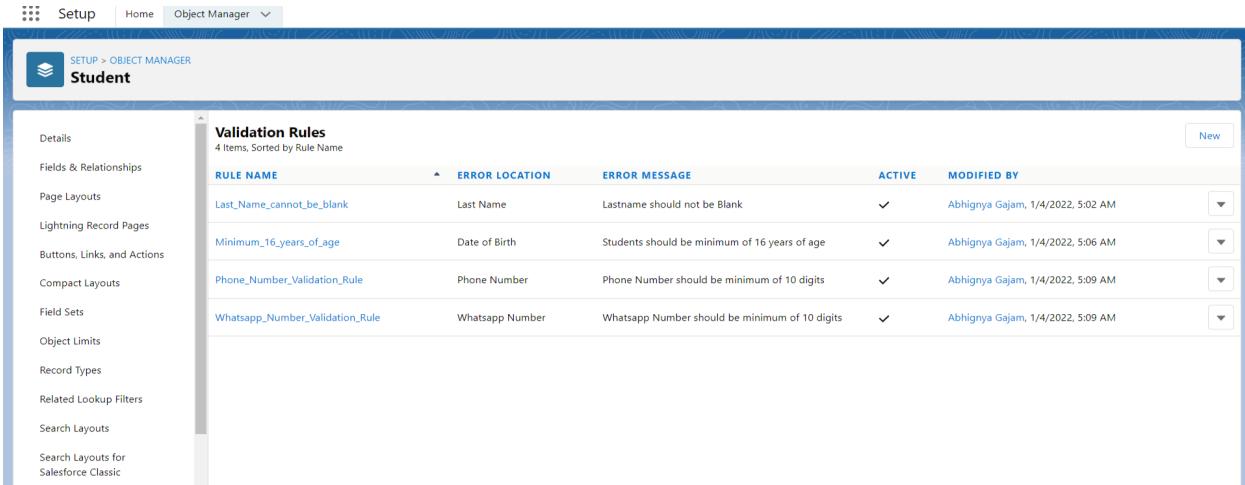
Detailed Description: Lastname should not be Blank: Enter error condition formula "ISBLANK(Field name)" and give the error message according to the rule.

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.

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

ONLINE TRAINING MANAGEMENT

Upload the Screenshot the Milestone / Activities:



The screenshot shows the Salesforce Object Manager interface for the 'Student' object. On the left, a sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Validation Rules' and displays four items, sorted by Rule Name. Each rule includes the rule name, error location, error message, active status, and modified by user (Abhignya Gajam) and date (1/4/2022). The rules are:

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Last_Name_cannot_be_blank	Last Name	Lastname should not be Blank	✓	Abhignya Gajam, 1/4/2022, 5:02 AM
Minimum_16_years_of_age	Date of Birth	Students should be minimum of 16 years of age	✓	Abhignya Gajam, 1/4/2022, 5:06 AM
Phone_Number_Validation_Rule	Phone Number	Phone Number should be minimum of 10 digits	✓	Abhignya Gajam, 1/4/2022, 5:09 AM
Whatsapp_Number_Validation_Rule	Whatsapp Number	Whatsapp Number should be minimum of 10 digits	✓	Abhignya Gajam, 1/4/2022, 5:09 AM

Topic 9: Create Validation Rules On Meeting Object

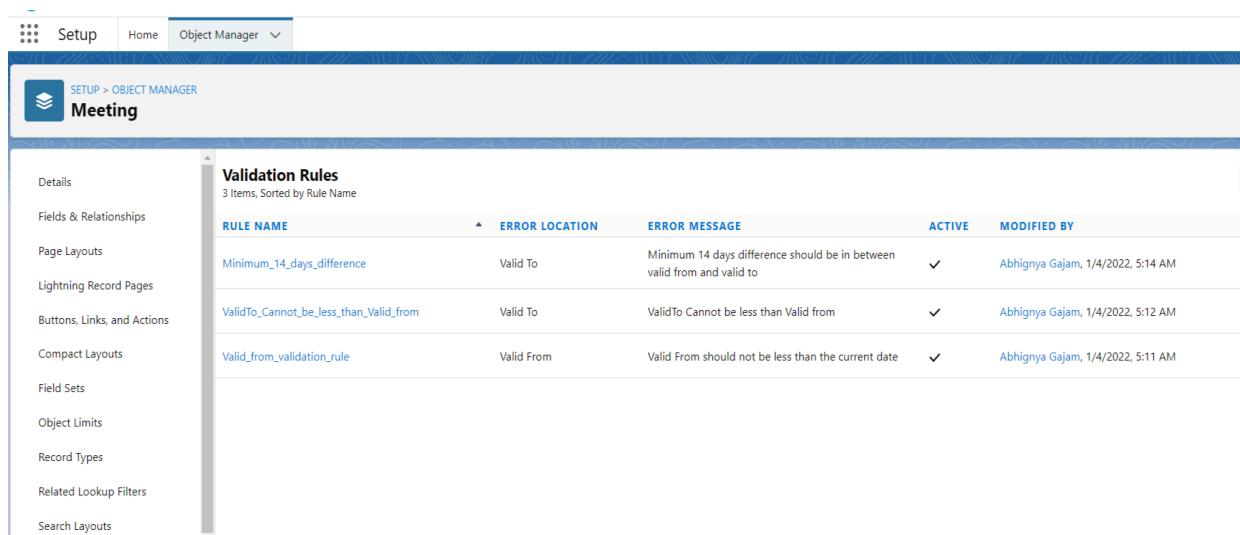
Milestones / Activities: Creating Validation Rules On Meeting Object

Detailed Description: Valid From should not be less than current date: 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 : 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 : In error condition formula enter "Valid_to__c - Valid_from__c < 14 " and enter the error message according to the rule.

Upload the Screenshot the Milestone / Activities:



The screenshot shows the Salesforce Object Manager interface for the 'Meeting' object. On the left, a sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Validation Rules' and displays three items, sorted by Rule Name. Each rule includes the rule name, error location, error message, active status, and modified by user (Abhignya Gajam) and date (1/4/2022). The rules are:

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Minimum_14_days_difference	Valid To	Minimum 14 days difference should be in between valid from and valid to	✓	Abhignya Gajam, 1/4/2022, 5:14 AM
ValidTo_Cannot_be_less_than_Valid_from	Valid To	ValidTo Cannot be less than Valid from	✓	Abhignya Gajam, 1/4/2022, 5:12 AM
Valid_from_validation_rule	Valid From	Valid From should not be less than the current date	✓	Abhignya Gajam, 1/4/2022, 5:11 AM

ONLINE TRAINING MANAGEMENT

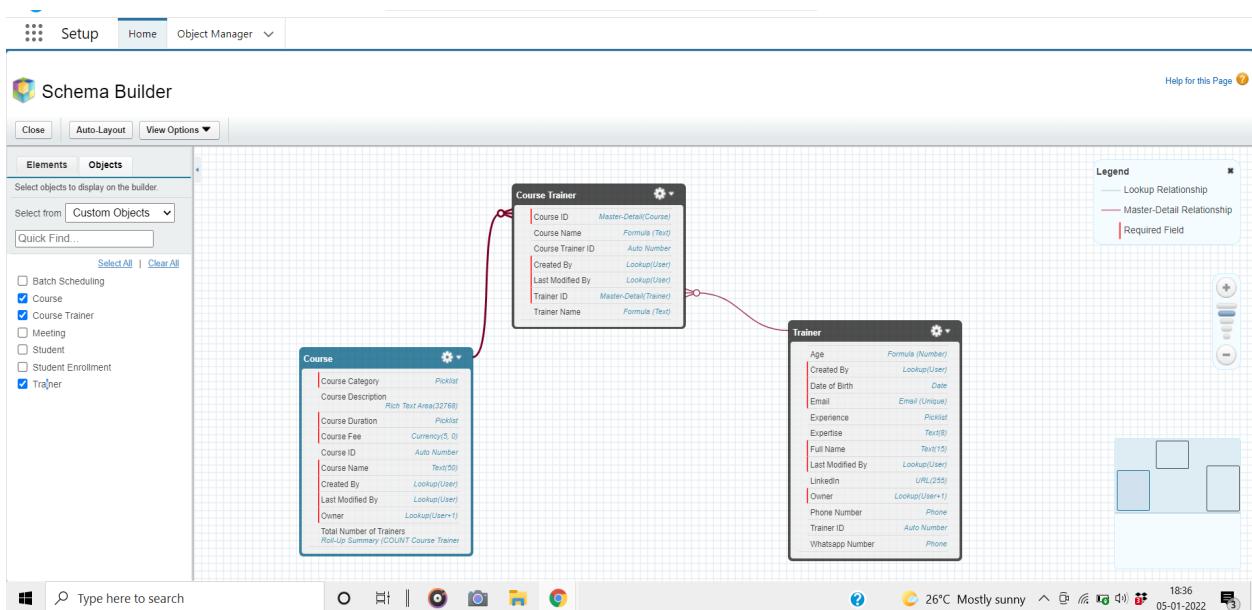
Day 3:

Topic 1: Create Relationship between Course and Trainer.

Milestones / Activities: Creating Master-Detail Relationship between Course and Trainer.

Detailed Description: To create relationship between Course and Trainer we are using a junction Object called "**Course Trainer**" using Master-Detail Relationship field type. Here, we are using Master-Detail Relationship field type because between the Course and Trainer objects there is many-to-many relationship.

Upload the Screenshot the Milestone / Activities:

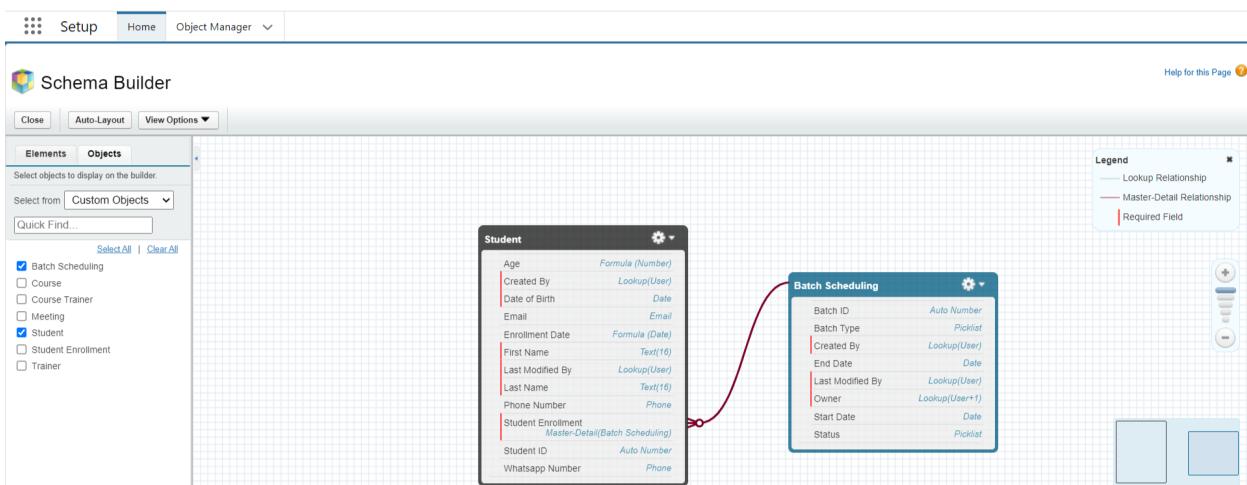


Topic 2: Create Relationship from Student to Batch Scheduling.

Milestones / Activities: Creating Relationship from Student to Batch Scheduling.

Detailed Description: To create relationship between Student and Batch Scheduling here we are using Master-Detail Relationship field type because between the Student and Batch Scheduling objects there is many-to-many relationship.

Upload the Screenshot the Milestone / Activities:



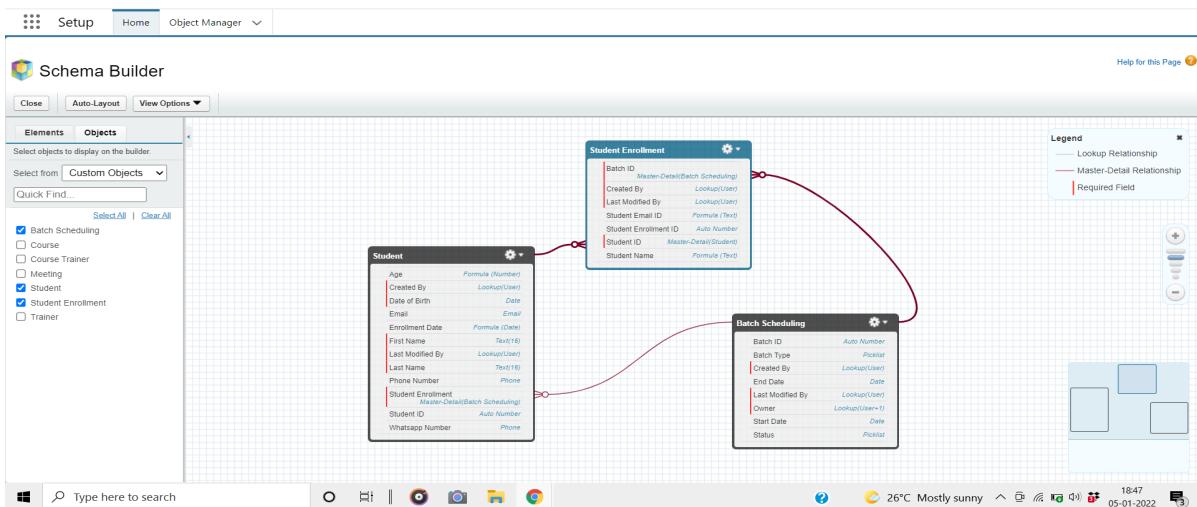
ONLINE TRAINING MANAGEMENT

Topic 3: Create Relationship between Student and Batch Scheduling.

Milestones / Activities: Creating Master-Detail Relationship between Student and Batch Scheduling.

Detailed Description: To create relationship between Course and Trainer we are using a junction Object called "**Student Enrollment**" using Master-Detail Relationship field type. Here, we are using Master-Detail Relationship field type because between the Student and Batch Scheduling objects there is many-to-many relationship.

Upload the Screenshot the Milestone / Activities:



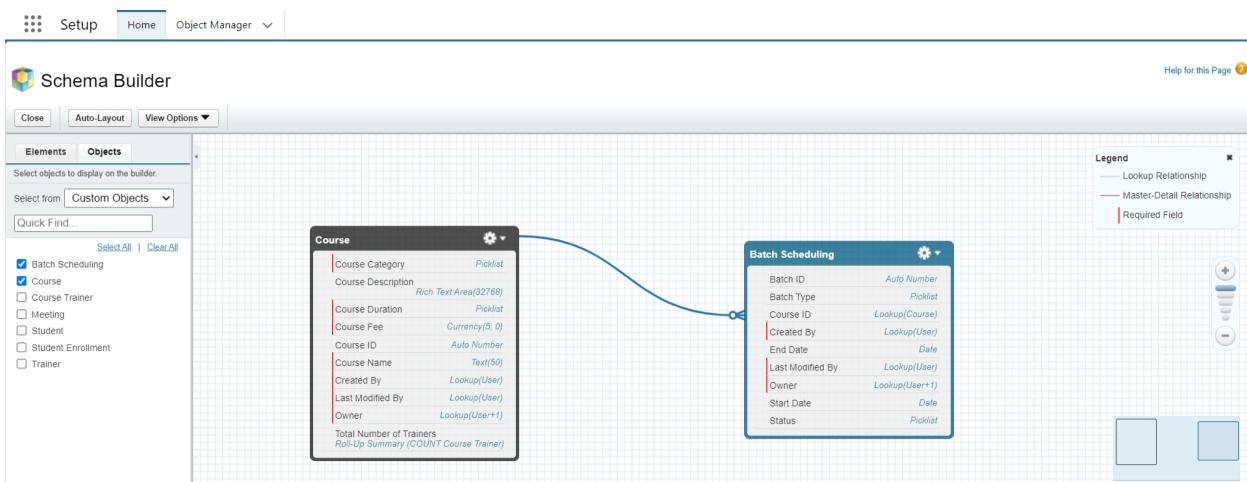
Day 4:

Topic 1: To create relationship between Batch Scheduling and Course

Milestones / Activities: Creating Look-up Relationship between Course and Batch Scheduling.

Detailed Description: To create relationship between Course and Batch Scheduling here we are using Look-up Relationship field type because between the Course and Batch Scheduling objects there is one-to-many relationship. In this case, the parent is Course and child is Batch Scheduling.

Upload the Screenshot the Milestone / Activities:



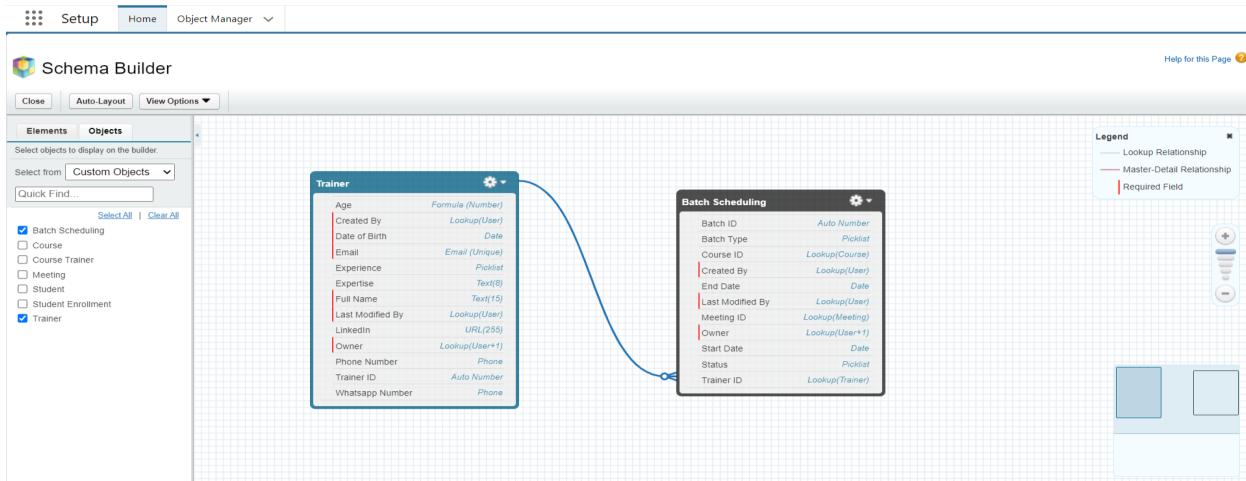
ONLINE TRAINING MANAGEMENT

Topic 2: To create relationship between Batch Scheduling and Trainer.

Milestones / Activities: Creating Look-up Relationship between Trainer and Batch Scheduling.

Detailed Description: To create relationship between Course and Batch Scheduling here we are using Look-up Relationship field type because between the Course and Batch Scheduling objects there is one-to-many relationship. In this case, the parent is Trainer and child is Batch Scheduling.

Upload the Screenshot the Milestone / Activities:

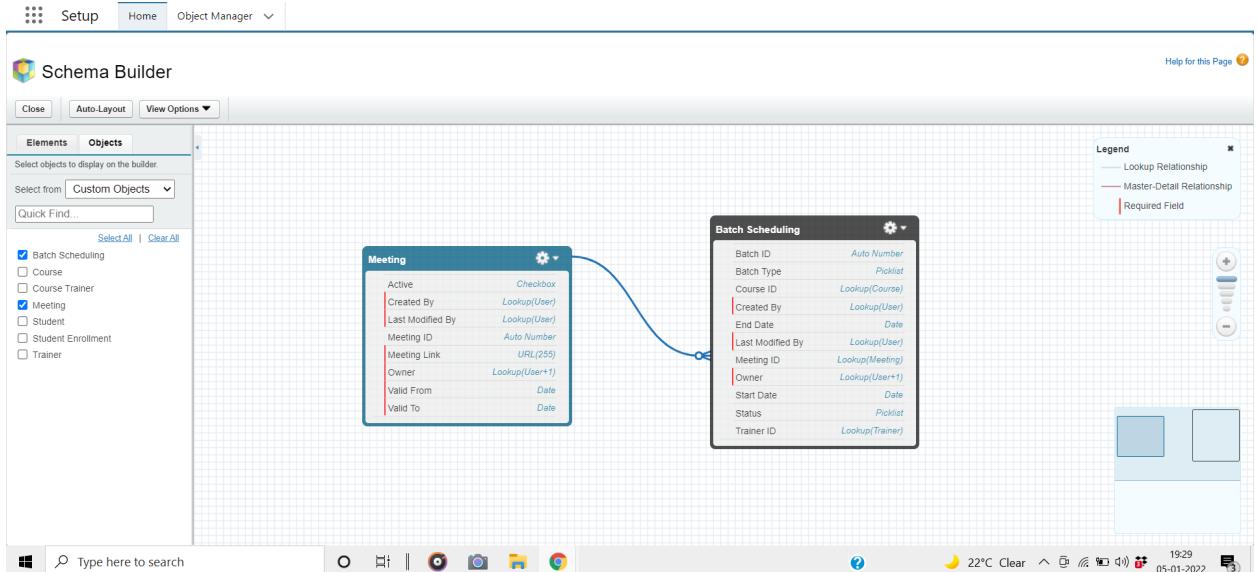


Topic 3: To create relationship between Batch Scheduling and Meeting.

Milestones / Activities: Creating Look-up Relationship between Meeting and Batch Scheduling.

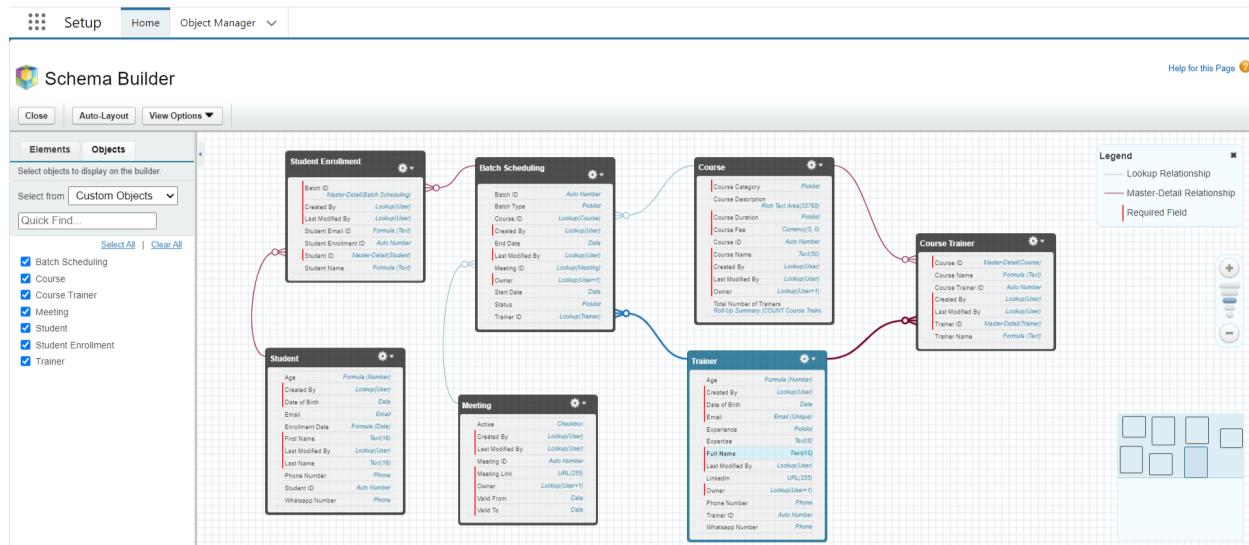
Detailed Description: To create relationship between Meeting and Batch Scheduling here we are using Look-up Relationship field type because between the Meeting and Batch Scheduling objects there is one-to-many relationship. In this case, the parent is Meeting and child is Batch Scheduling.

Upload the Screenshot the Milestone / Activities:



ONLINE TRAINING MANAGEMENT

Upload the Screenshot the Milestone / Activities Done Today: All the relationships that we created today are shown in below:

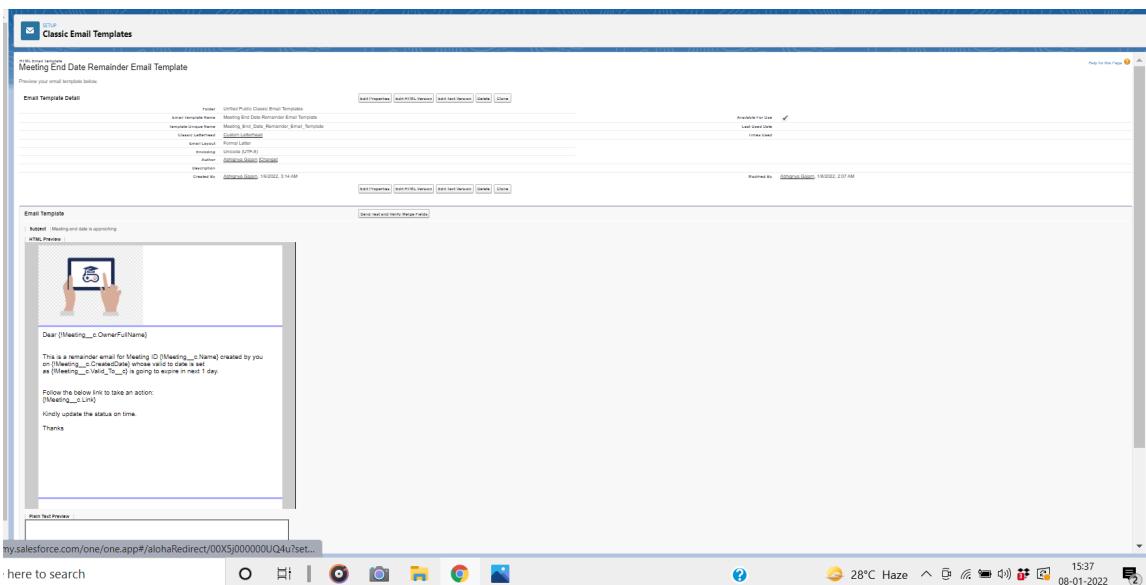


Day 5:

Topic 1: Create Work Flow Rules On Meeting Object

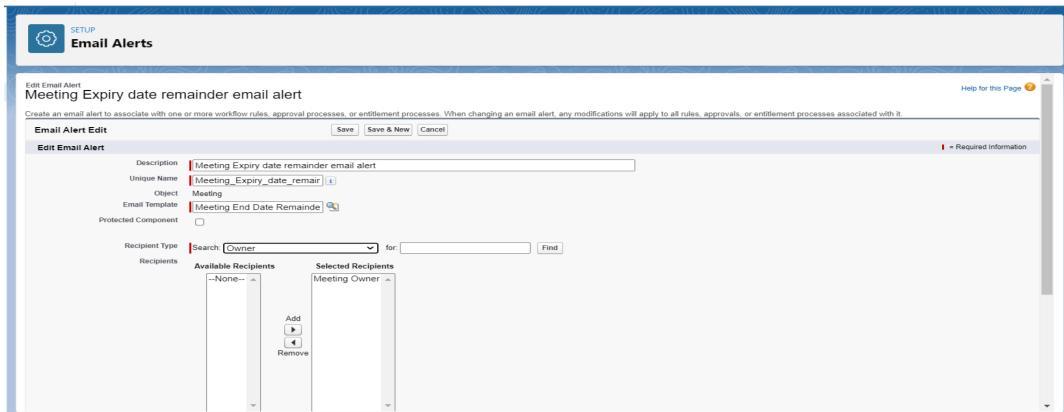
Milestones / Activities: Creating Automation Rules On Meeting Object by using Process Builder.

Detailed Description: To create the automation rule, we first need to create an email template to send 1 day before valid to date and 1 day after valid to date send a reminder email and set the active check box will uncheck automatically and inform owner by email.



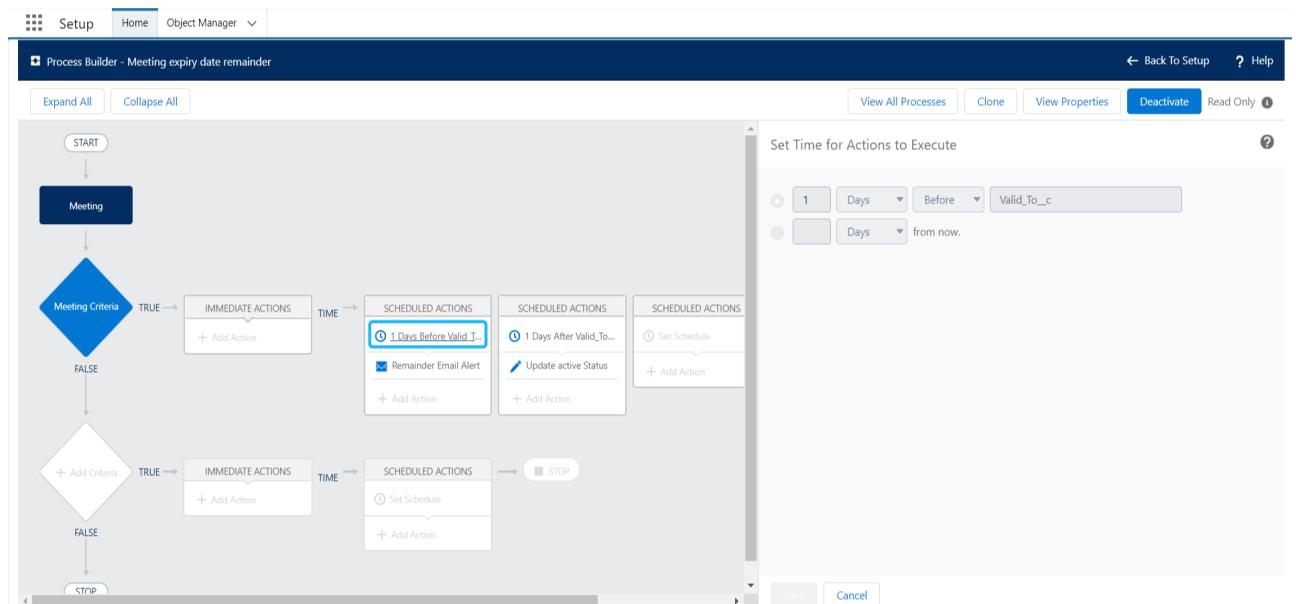
ONLINE TRAINING MANAGEMENT

After creating the email template we need to add an email alert on the Meeting object.



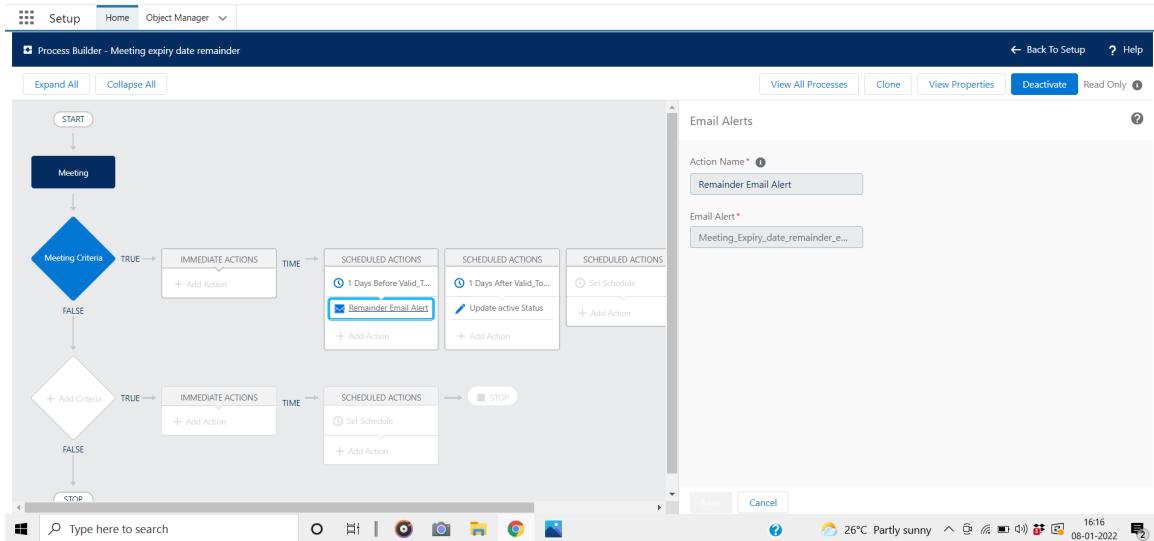
Now we need to create the automation rule to send the remainder email to inform the owner that the meeting will be expired and we need to add an action for the active check box to uncheck it automatically when the meeting is expired.

a) Creating automation rule 1 day before valid to date.

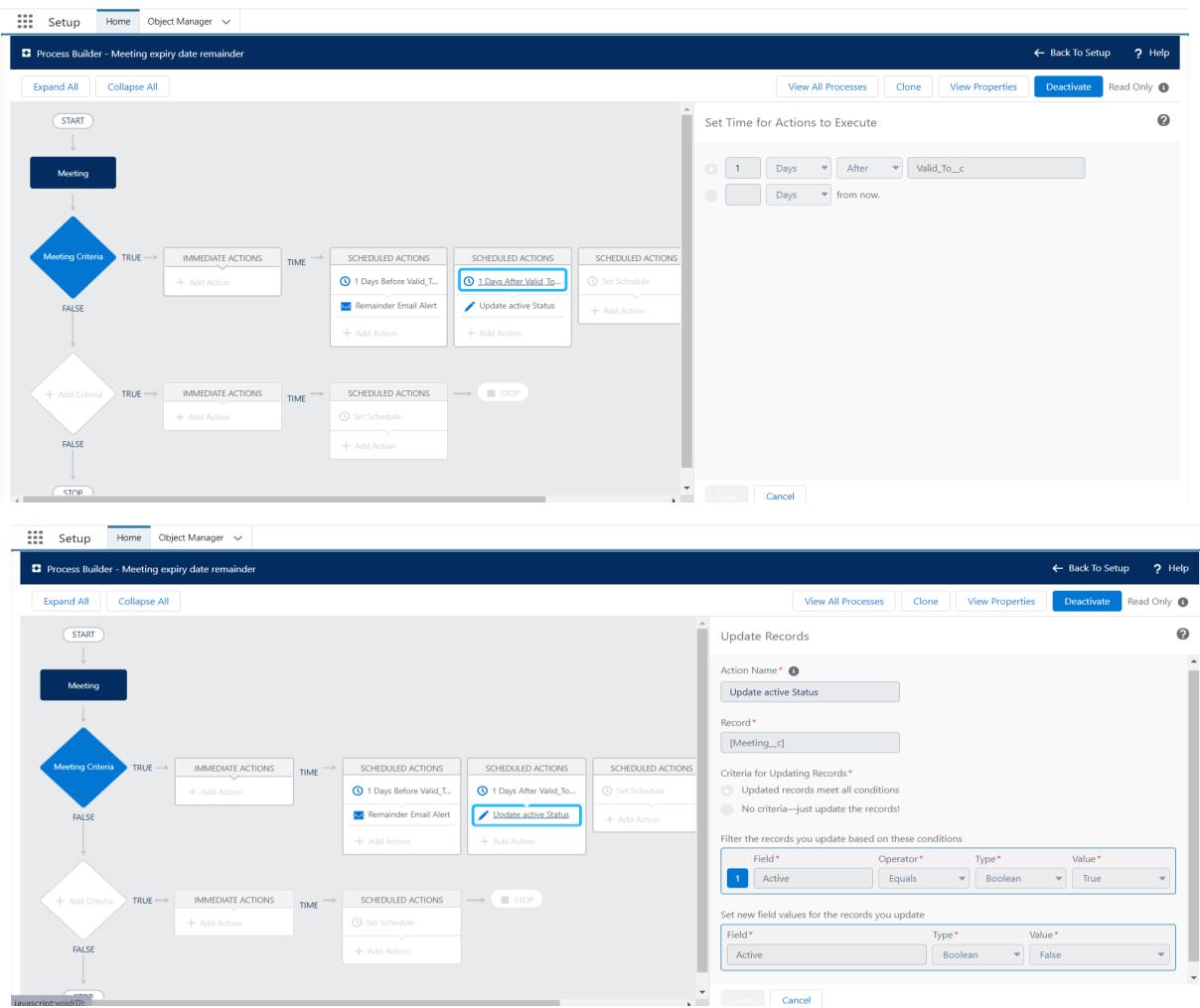


ONLINE TRAINING MANAGEMENT

b) Adding action i.e., adding email alert under the above rule.

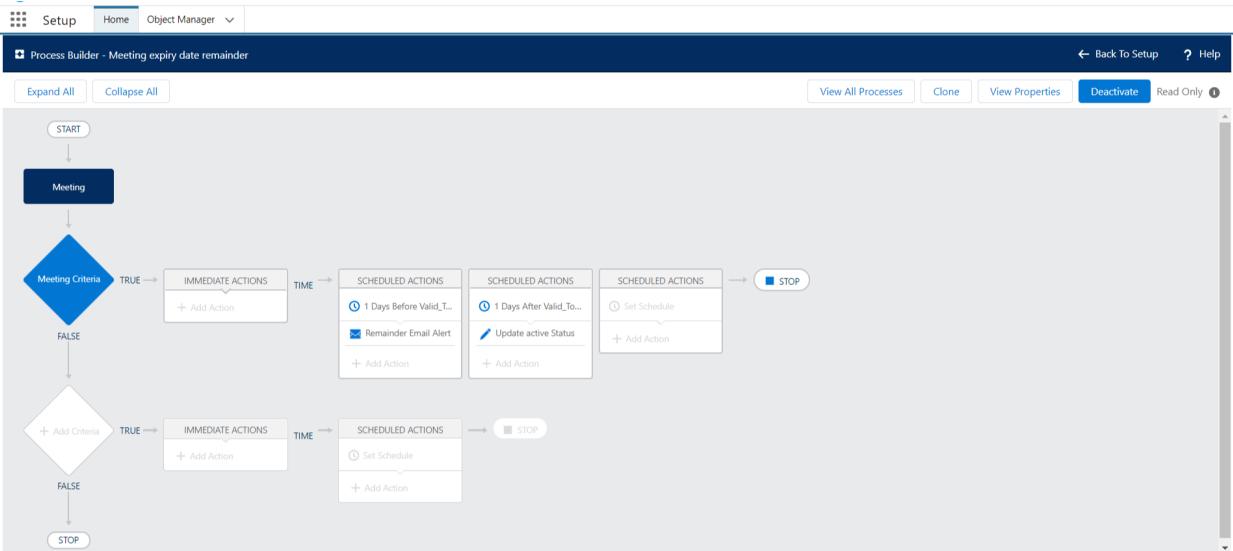


c) Creating rule 1 day after valid to date the active check box will uncheck automatically and inform owner.



ONLINE TRAINING MANAGEMENT

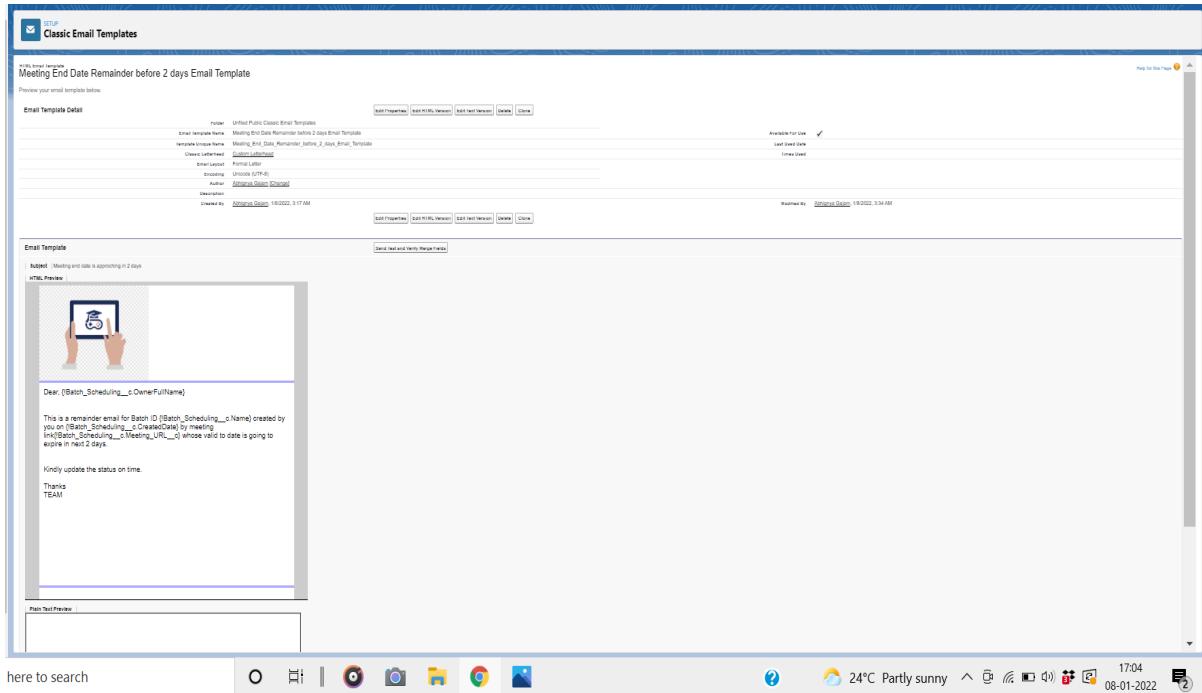
Upload the Screenshot the Milestone / Activities: The final outcome if this activity:



Topic 2: Create Work Flow Rules On Class/Batch Schedulings Object.

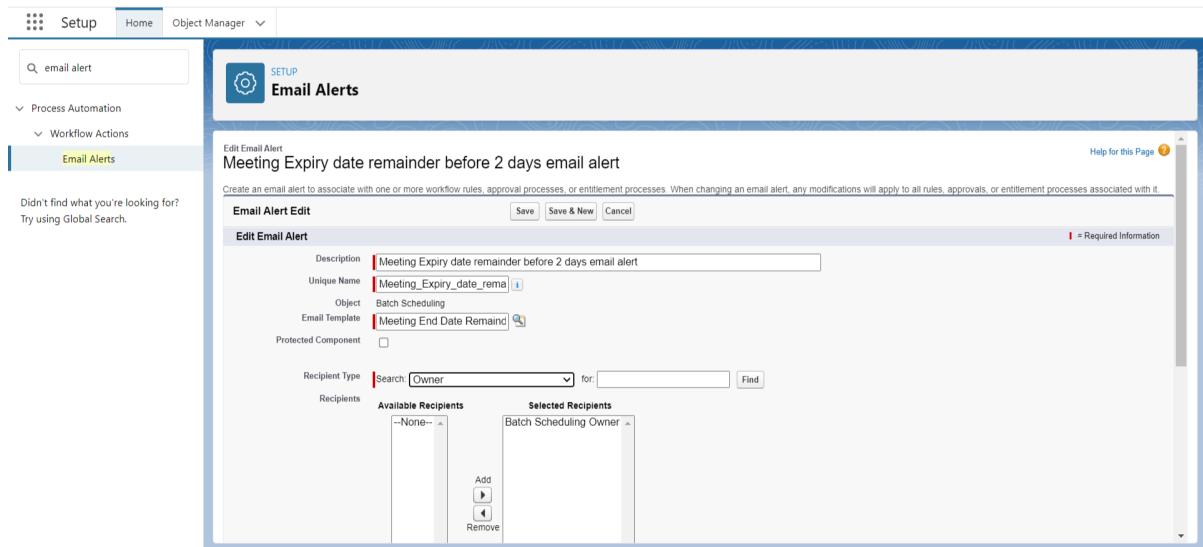
Milestones / Activities: Creating Work Flow Rules On Class/Batch Schedulings Object.

Detailed Description: To create the automation rule, we first need to create an email template to send 2 days before end date send a reminder email to inform owner by email.

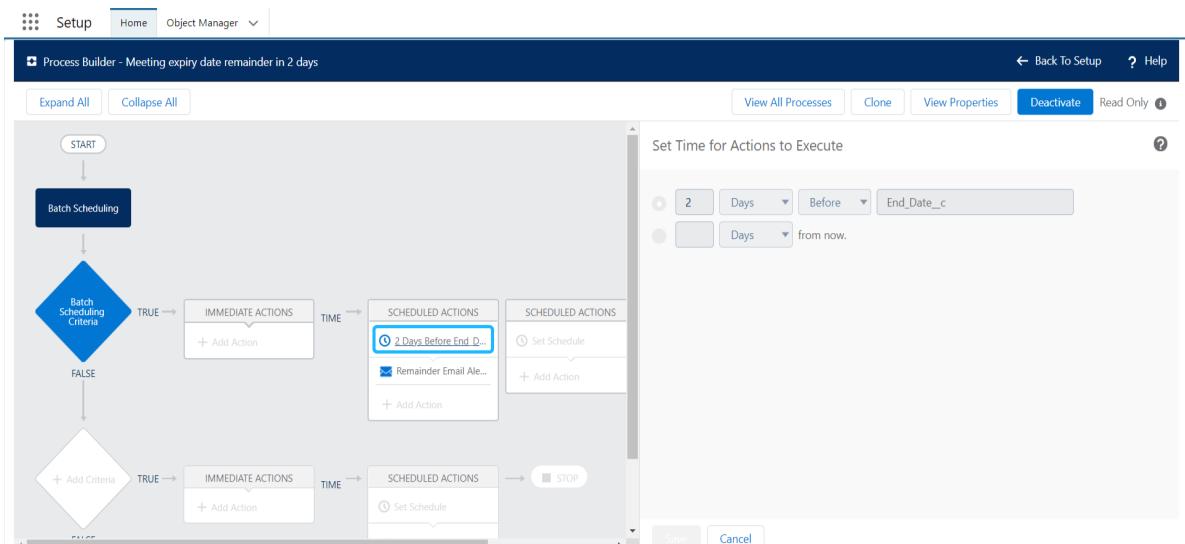


ONLINE TRAINING MANAGEMENT

After creating the email template we need to add an email alert on the Batch Scheduling object.

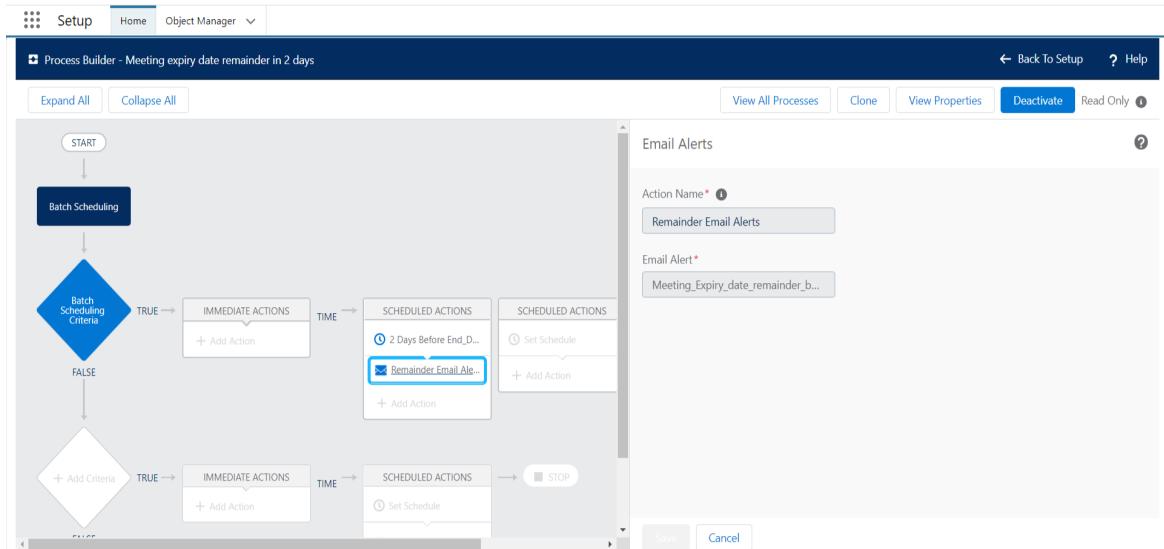


a) Creating automation rule 2 days before end date.

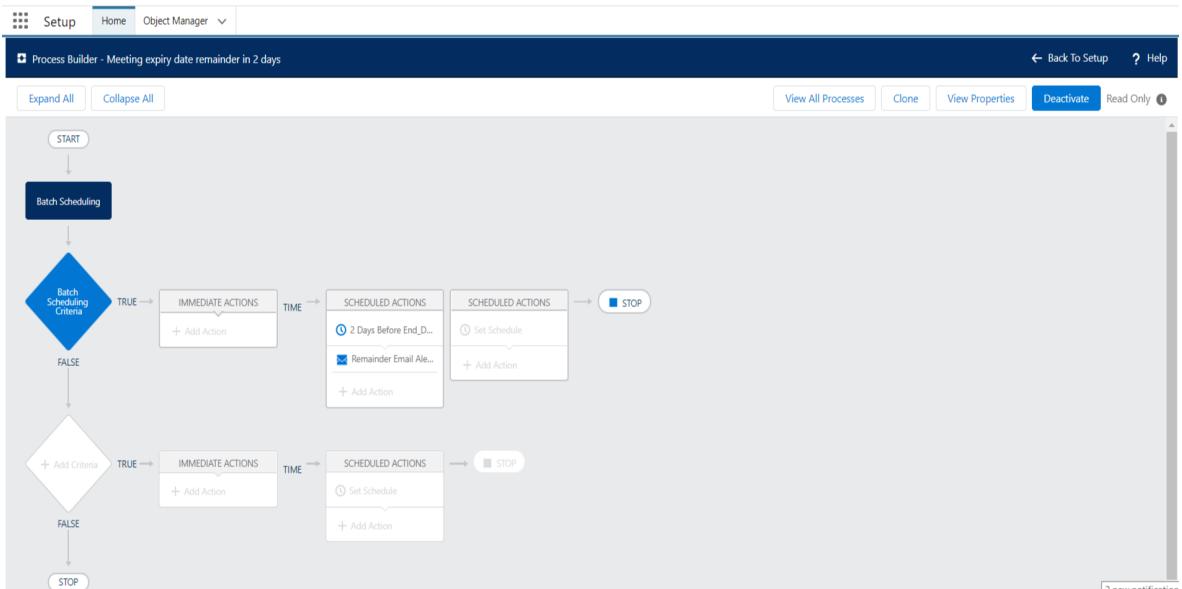


ONLINE TRAINING MANAGEMENT

b) Adding action i.e., adding email alert under the above rule.



Upload the Screenshot the Milestone / Activities:



ONLINE TRAINING MANAGEMENT

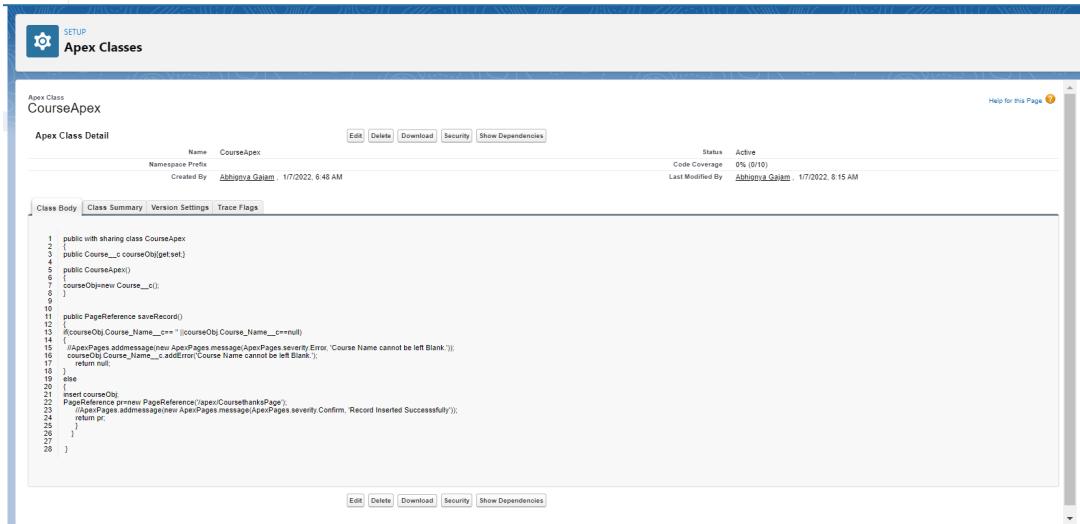
Day 6:

Topic 1: Create New Course Using Custom Controller.

Milestones / Activities: Creating New Course Using Custom Controller.

Detailed Description: To create a new course visualforce page, we need to write the visualforce page code.

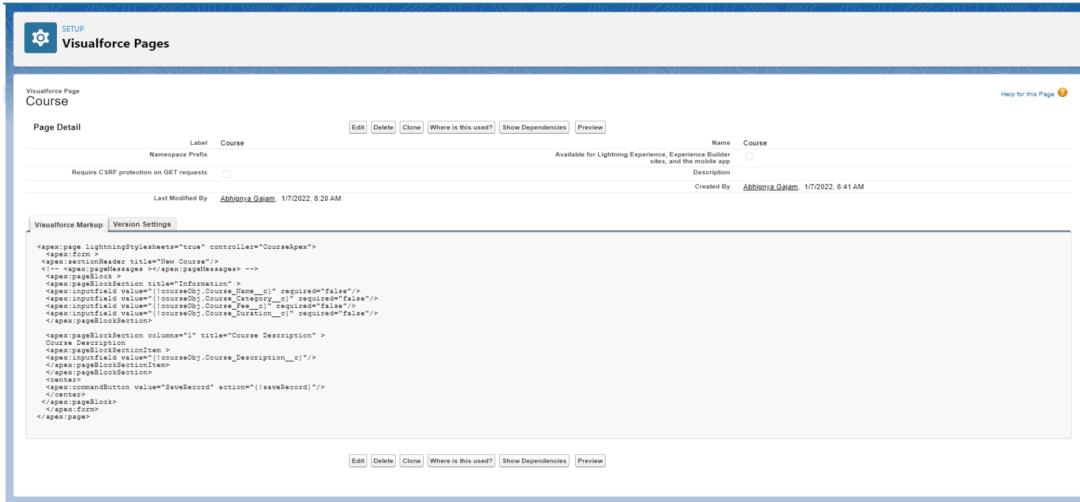
To write the visualforce page code for New Course we need to create a apex class. I named the apex class as "**CourseApex**".



The screenshot shows the Salesforce Apex Classes page. A single Apex Class named "CourseApex" is listed. The class body contains the following code:

```
1 public with sharing class CourseApex
2 {
3     public Course__c courseObj{get;}
4     public CourseApex()
5     {
6         courseObj=new Course__c();
7     }
8
9     public PageReference saveRecord()
10    {
11        if(courseObj.Course__Name__c == null)
12        {
13            ApexPages.addMessage(new ApexPages.message(ApexPages.severity.Error, 'Course Name cannot be left Blank.'));
14            courseObj.Course__Name__c.addError('Course Name cannot be left Blank.');
15            return null;
16        }
17        else
18        {
19            insert courseObj;
20            PageReference prnew = new ApexPages.currentPage();
21            prnew.setRedirect(true);
22            ApexPages.addMessage(new ApexPages.message(ApexPages.severity.Confirm, 'Record Inserted Successfully'));
23            return pr;
24        }
25    }
26 }
27 }
```

After creating the apex class, now its time for the visualforce page code.



The screenshot shows the Salesforce Visualforce Pages page. A single Visualforce Page named "COURSE" is listed. The Visualforce Markup section contains the following code:

```
<apex:page lightningStyleSheets="true" controller="CourseApex">
<apex:sectionHeader title="New Course">
<apex:pageMessages /><apex:pageBlock>
<apex:pageBlockSection id="courseSection" >
<apex:inputField value="!courseObj.Course_Name__c" required="false"/>
<apex:inputField value="!courseObj.Course_Category__c" required="false"/>
<apex:inputField value="!courseObj.Course_Duration__c" required="false"/>
<apex:pageBlockSection columns="1" title="Course Description" >
<apex:inputText value="!courseObj.Course_Description__c" />
<apex:inputField value="!courseObj.Course_Duration__c" />
</apex:pageBlockSection>
<apex:commandButton value="SaveRecord" action="!saveRecord"/>
</apex:pageBlock>
</apex:form>
</apex:page>
```

ONLINE TRAINING MANAGEMENT

After successfully adding the record in to the database, I want to redirect the page to course thanks page.

The screenshot shows the Visualforce Pages setup interface. The page title is "CourseThanksPage". The "Page Detail" section includes fields for Label (CourseThanksPage), Namespace Prefix, Name (CourseThanksPage), Available for Lightning Experience, Experience Builder sites, and the mobile app, Description, and Created By (Abhijnya.Gajam). It also shows the last modified date (1/7/2022, 7:31 AM) and the user who modified it (Abhijnya.Gajam). The "Visualforce Markup" tab is selected, displaying the following code:

```
<apex:page lightningStyleSheets="true">
<h1 style="font-size:20px;">Course Registered Successfully in Database.</h1>
</apex:page>
```

Upload the Screenshot the Milestone / Activities:

The screenshot shows the "New Course" form. The "Information" section contains fields for Course Name (empty), Course Category (dropdown menu showing "--None--"), Course Fee (500), and Course Duration (dropdown menu showing "--None--"). The "Course Description" section contains a rich text editor toolbar and a large text area. At the bottom right of the form is a "SaveRecord" button.

Course Registered Successfully in Database.

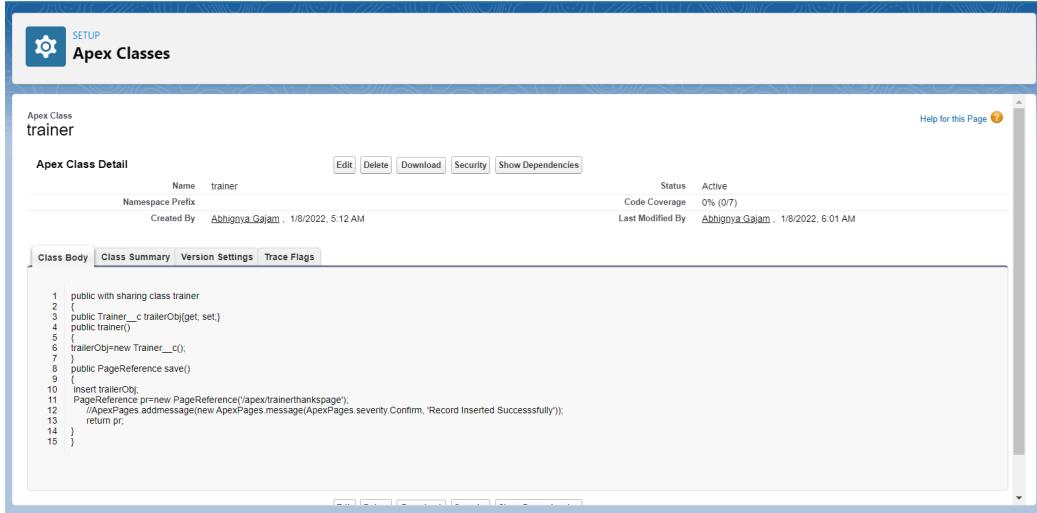
ONLINE TRAINING MANAGEMENT

Topic 2: Create New Trainer Using Custom Controller.

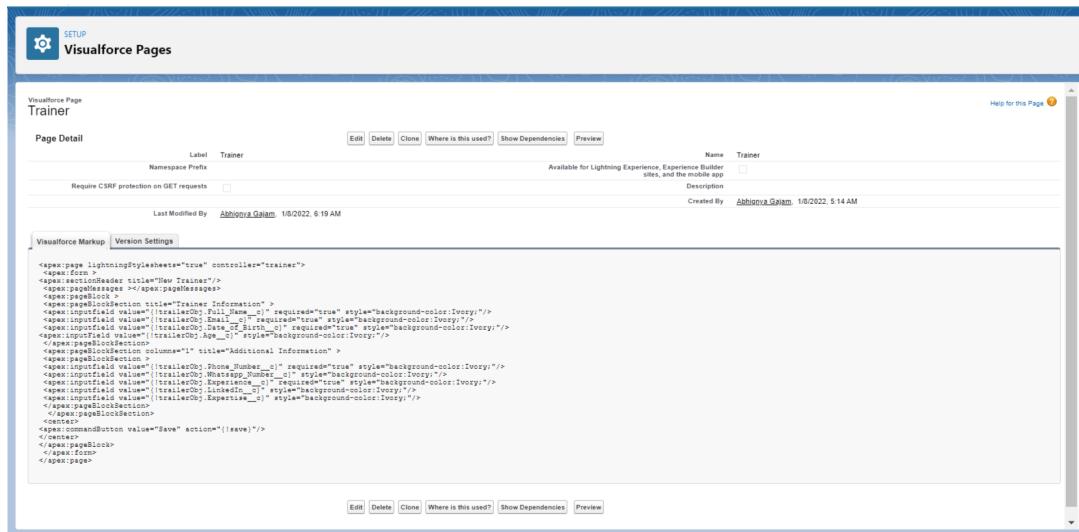
Milestones / Activities: Creating New Trainer Using Custom Controller.

Detailed Description: To create a new Trainer visualforce page, we need to write the visualforce page code. We need to create apex class first.

To write the visualforce page code for New Trainer we need to create apex class. I named the apex class as "**Trainer**".

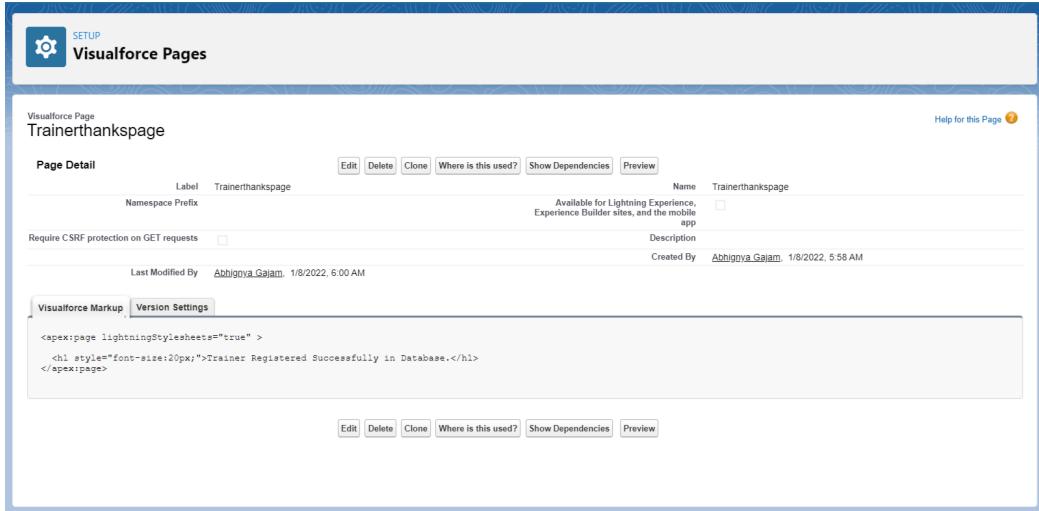


After creating the apex class, now its time for the visualforce page code.



ONLINE TRAINING MANAGEMENT

After successfully adding the record in to the database, I want to redirect the page to course thanks page.



Upload the Screenshot the Milestone / Activities:

A screenshot of the "New Trainer" form. The form has two sections: "Trainer Information" and "Additional Information". In the "Trainer Information" section, fields include "Full Name" (required), "Email" (required), "Date of Birth" (with value "1/8/2022"), and "Age". In the "Additional Information" section, fields include "Phone Number" (required), "Whatsapp Number", "Experience" (dropdown menu with "None" selected), "LinkedIn" (input field), and "Expertise" (input field). A "Save" button is at the bottom right. Below the form, a success message says "Trainer Registered Successfully in Database."

Trainer Registered Successfully in Database.

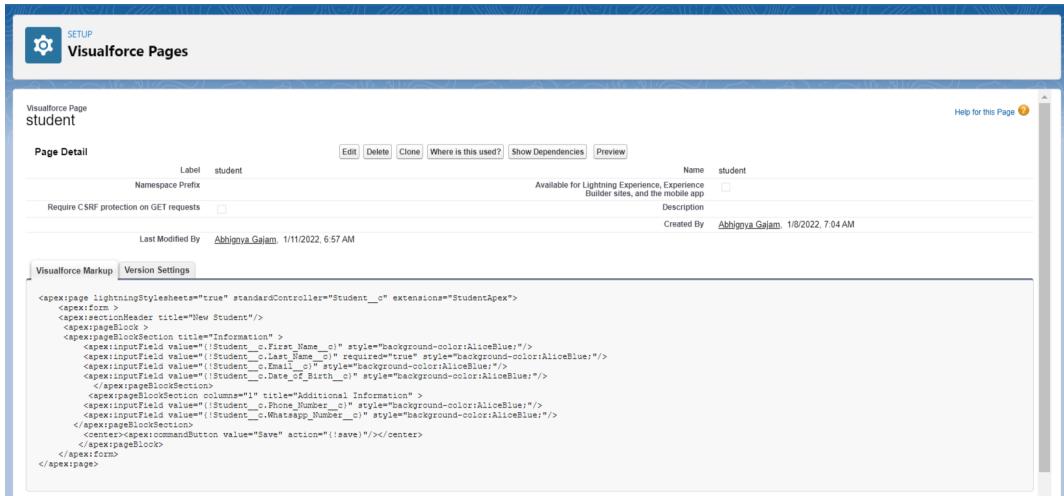
ONLINE TRAINING MANAGEMENT

Day 7:

Topic 1: Create New Student Using Standard Controller.

Milestones / Activities: Creating New Student Using Standard Controller.

Detailed Description: To create a new Student visualforce page, we need to write the visualforce code using Standard Controller.



Upload the Screenshot the Milestone / Activities:

The screenshot shows the 'New Student' visualforce page. It has two main sections: 'Information' and 'Additional Information'. The 'Information' section contains fields for First Name, Last Name, Email, and Date of Birth. The 'Additional Information' section contains fields for Phone Number and Whatsapp Number, along with a 'Save' button.

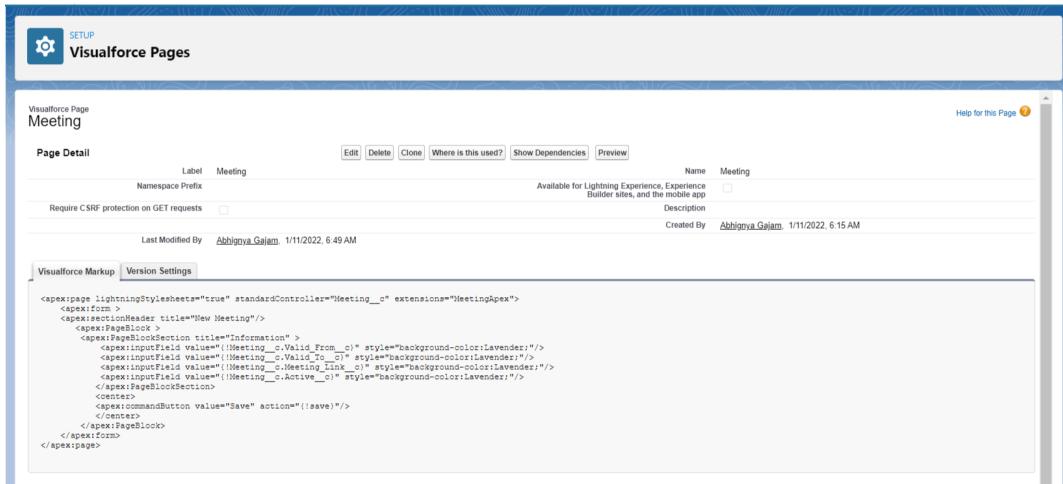
Section	Field	Type	Value
Information	First Name	Text Input	
	Last Name	Text Input	
	Email	Text Input	
	Date of Birth	Date Input	[1/14/2022]
Additional Information			
Phone Number		Text Input	
Whatsapp Number		Text Input	

ONLINE TRAINING MANAGEMENT

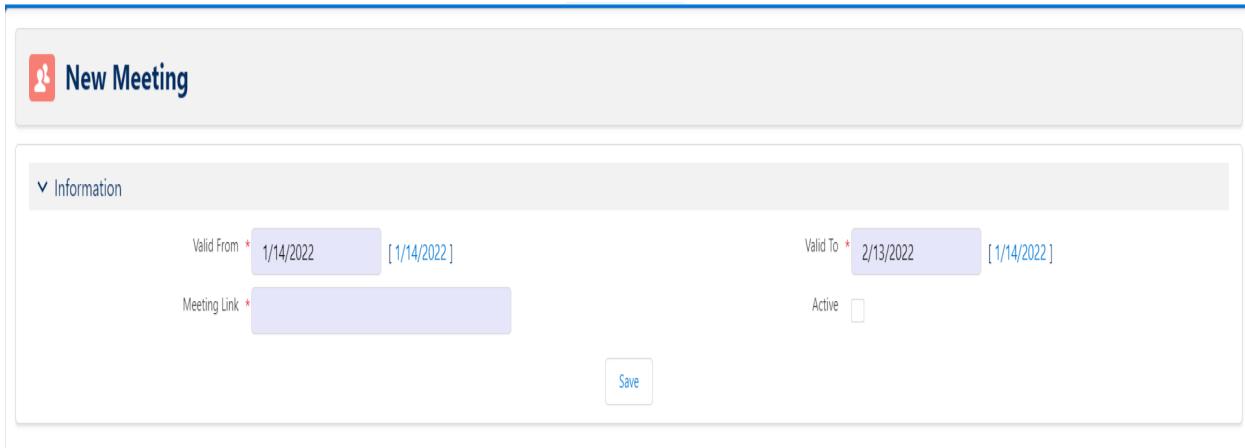
Topic 2: Create New Meeting Using Standard Controller.

Milestones / Activities: Creating New Meeting Using Standard Controller.

Detailed Description: To create a new Meeting visualforce page, we need to write the visualforce code using Standard Controller.



Upload the Screenshot the Milestone / Activities:

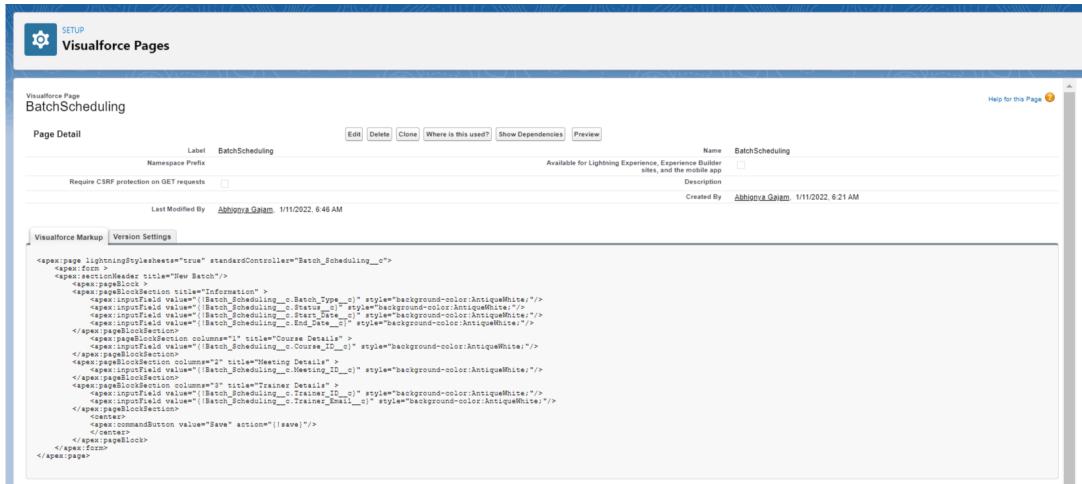


ONLINE TRAINING MANAGEMENT

Topic 3: Create New Batch Scheduling Using Standard Controller.

Milestones / Activities: Creating New Batch Scheduling Using Standard Controller.

Detailed Description: To create a new Batch Scheduling visualforce page, we need to write the visualforce code using Standard Controller.



Upload the Screenshot the Milestone / Activities:

A screenshot of the "New Batch" creation form. The "Information" section includes fields for "Batch Type" (dropdown, empty), "Status" (dropdown, empty), "Start Date" (1/14/2022) and "End Date" (2/28/2022). The "Course Details" section has a "Course ID" input field with a search icon. The "Meeting Details" section has a "Meeting ID" input field with a search icon. The "Trainer Details" section has "Trainer ID" and "Trainer Email" input fields, both with search icons. At the bottom right is a "Save" button.

ONLINE TRAINING MANAGEMENT

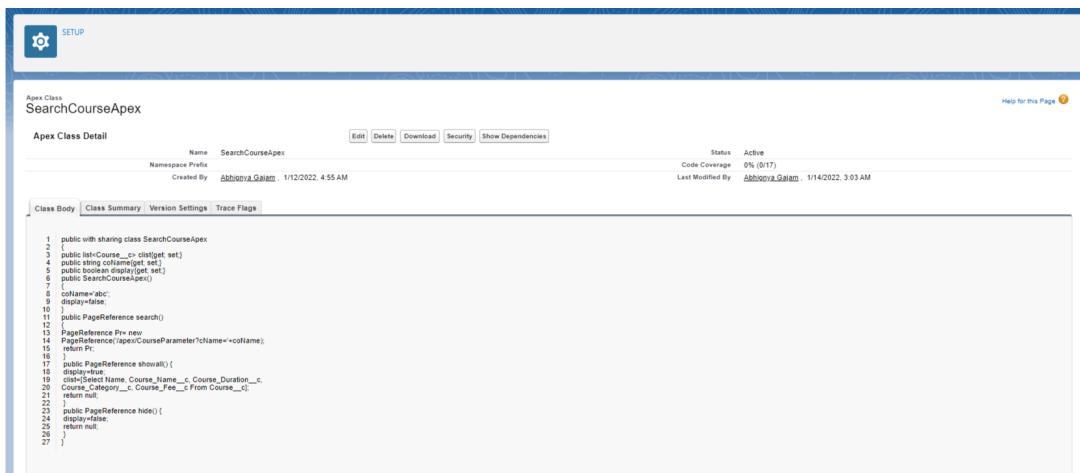
Day 8:

Topic 1: Create Search Course Page using custom Controller.

Milestones / Activities: Creating Search Course Page using custom Controller.

Detailed Description: To create a Search Course visualforce page, we need to write the visualforce page code.

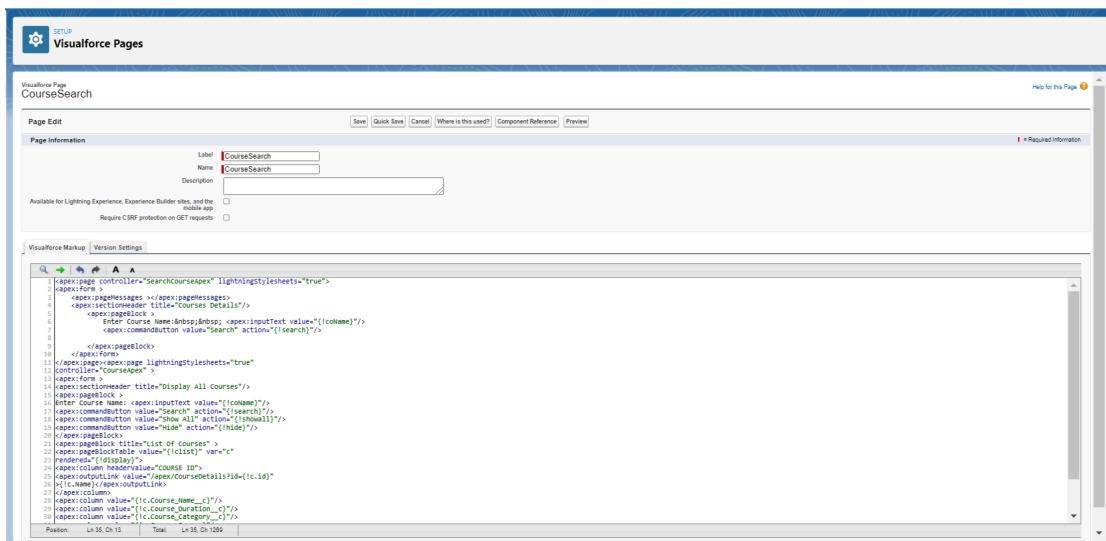
To write the visualforce page code for Search Course we need to create apex class. I named the apex class as "**SearchCourseApex**".



The screenshot shows the Apex Class Editor for the class `SearchCourseApex`. The class body contains the following Apex code:

```
1 public with sharing class SearchCourseApex {
2     public list<Course__c> cList{get;set;}
3     public string colName{get;set;}
4     public string courseID{get;set;}
5     public SearchCourseApex()
6     {
7         colName='abc';
8         display=false;
9     }
10    public PageReference search()
11    {
12        PageReference pr = new PageReference('/apex/SearchCourseParameter?cName=' + colName);
13        return pr;
14    }
15    public PageReference showAll()
16    {
17        list<Select Name__c, Course__Duration__c,
18        Course__Category__c, Course__Fee__c> From Course__c;
19        return null;
20    }
21    public PageReference hide()
22    {
23        display=false;
24        return null;
25    }
26}
27}
```

After creating the apex class, now its time for the visualforce page code.



The screenshot shows the Visualforce Page Editor for the page `CourseSearch`. The visualforce markup section contains the following code:

```
<apex:page controller="SearchCourseApex" lightningStylesheets="true">
<apex:messages />
<apex:pageMessages title="Courses Details"/>
<apex:pageBlock >
    Enter Course Name &nbsp;&nbsp;<apex:inputText value="{!colName}"/>
    <apex:commandButton value="Search" action="{!search}"/>
</apex:pageBlock>
<apex:form >
    <apex:selectList >
        Available for Lightning Experience, Experience Builder sites, and the mobile app <input checked="" type="checkbox"/>
        Requires CSRF protection on GET requests <input type="checkbox"/>
    </apex:selectList>
</apex:form>
<apex:pageBlock >
    <apex:pageHeader title="Display All Courses"/>
    <apex:pageBlock >
        <apex:inputText value="{!courseName}"/>
        <apex:commandButton value="Search" action="{!search}"/>
        <apex:commandButton value="Hide" action="{!showAll}"/>
    </apex:pageBlock>
    <apex:pageBlock >
        <apex:pageHeader title="List of courses" />
        <apex:pageBlock >
            <apex:selectTable value="{!cList}" var="c" />
            <apex:column headerValue="Course ID" />
            <apex:column outputValue="/apex/CourseDetails?id=[!c.id]" />
            <apex:column />
            <apex:column value="{!c.Course_Name__c}"/>
            <apex:column value="{!c.Course_Duration__c}"/>
            <apex:column value="{!c.Course_Category__c}"/>
        </apex:pageBlock>
    </apex:pageBlock>
</apex:pageBlock>
</apex:page>
```

ONLINE TRAINING MANAGEMENT

Upload the Screenshot the Milestone / Activities:

The screenshot shows a web-based application interface. At the top, there's a navigation bar with icons for Home, New Course, Trainer Registration, Student Registration, Meeting Registration, Batch Scheduling Registration, Search Course, Search Student, and Training Calendar. Below the navigation bar, a search bar allows users to enter a course name and search. The main content area is titled "Courses Details".

Topic 2: Create Search Student Course Page using custom Controller.

Milestones / Activities: Creating Search Student Page using custom Controller.

Detailed Description: To create a Search Student visualforce page, we need to write the visualforce page code.

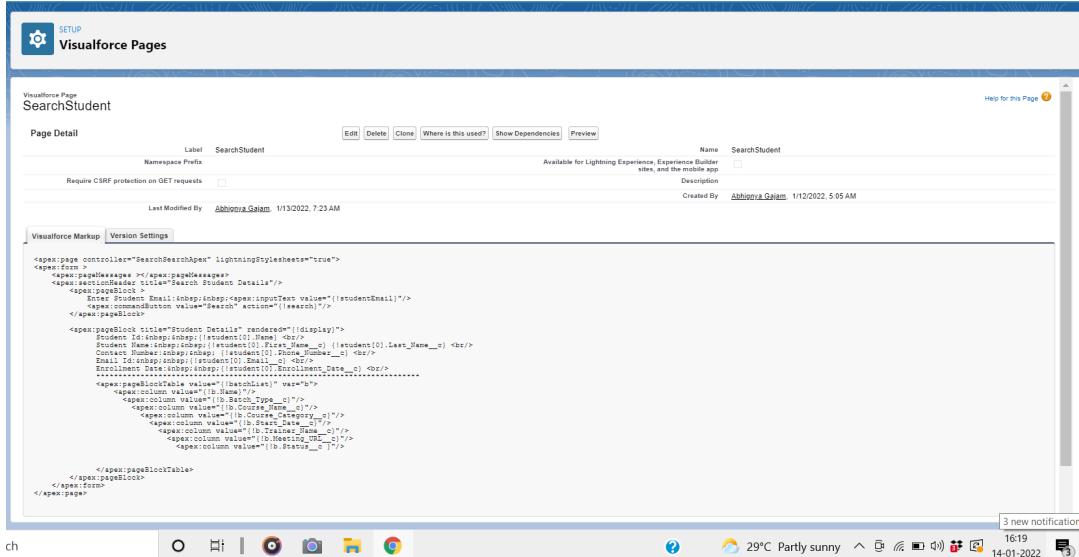
To write the visualforce page code for Search Student we need to create apex class. I named the apex class as "**SearchStudentApex**".

The screenshot shows the Apex Classes page in Salesforce. The class is named "SearchStudentApex". The code is as follows:

```
1 public with sharing class SearchStudentApex {
2     public String getEnvironment() {
3         return Environment;
4     }
5     public void setEnvironment(String environment) {
6         Environment = environment;
7     }
8     public List<Batch> getBatches() {
9         return Batches;
10    }
11 }
12
13
14
15 public PageReference search() {
16     EnvUtil.setEnvironment();
17     EnvUtil.setBatch();
18     EnvUtil.setMeeting();
19     EnvUtil.setBatchScheduling();
20     EnvUtil.setBatch();
21
22     if (Environment == null) {
23         EnvUtil.setEnvironment();
24     }
25     if (Batch == null) {
26         EnvUtil.setBatch();
27     }
28     if (Meeting == null) {
29         EnvUtil.setMeeting();
30     }
31     if (BatchScheduling == null) {
32         EnvUtil.setBatchScheduling();
33     }
34
35     if (Batch != null) {
36         Batch.Batch_ID__c = Batch_ID__c;
37     }
38
39     if (Meeting != null) {
40         Meeting.Meeting_ID__c = Meeting_ID__c;
41     }
42
43     if (BatchScheduling != null) {
44         BatchScheduling.Batch_Schedule_ID__c = Batch_Schedule_ID__c;
45     }
46
47     ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.Info, 'Student didn\'t register for any program'));
48
49     if (Batch != null) {
50         EnvUtil.setBatch();
51     }
52
53     if (Meeting != null) {
54         EnvUtil.setMeeting();
55     }
56
57     if (BatchScheduling != null) {
58         EnvUtil.setBatchScheduling();
59     }
60
61     return null;
62 }
```

ONLINE TRAINING MANAGEMENT

After creating the apex class, now its time for the visualforce page code.



Upload the Screenshot the Milestone / Activities:

A screenshot of the 'Online Training Ma...' application. The top navigation bar includes links for Home, New Course, Trainer Registration, Student Registration, Meeting Registration, Batch Scheduling Registration, Search Course, Search Student (which is highlighted), and Training Calendar. Below the navigation is a search bar with placeholder 'Search...'. The main content area is titled 'Search Student Details' and features a search form with an 'Enter Student Email:' input field and a 'Search' button.

Day 9:

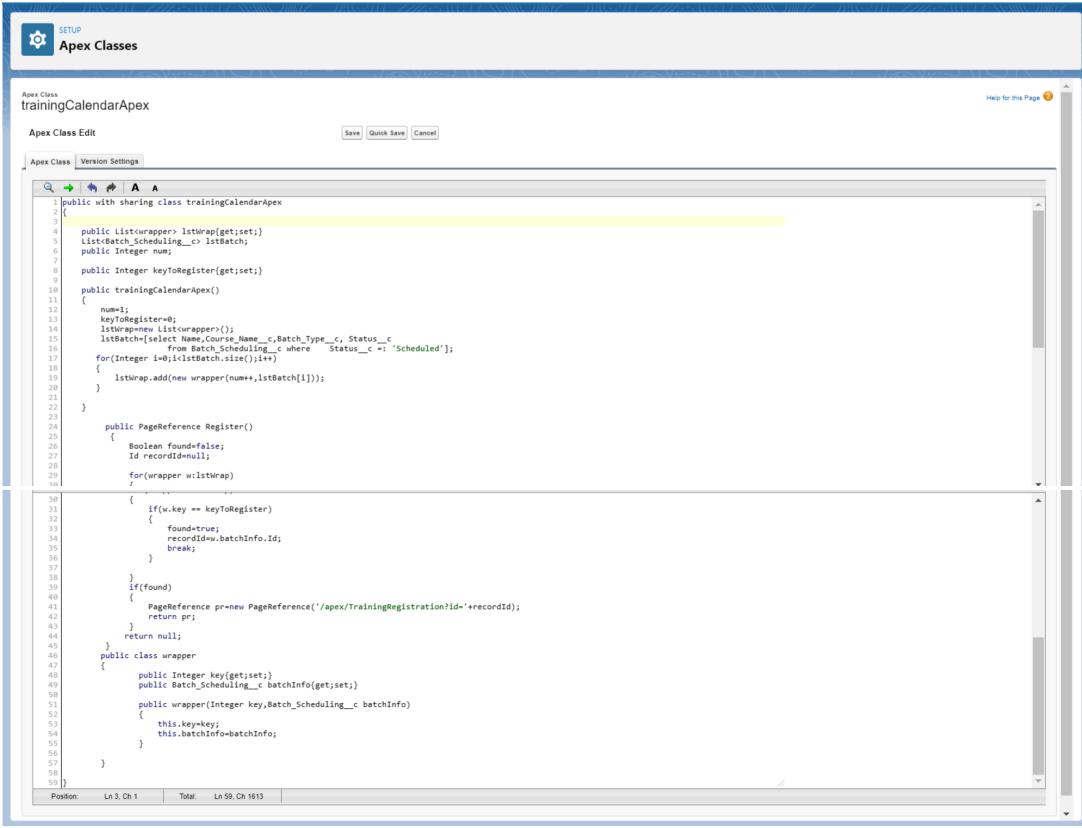
Topic 1: Create Training Calendar Page using custom Controller.

Milestones / Activities: Creating Training Calendar Page using custom Controller.

Detailed Description: To create a Training Calendar visualforce page, we need to write the visualforce page code.

To write the visualforce page code for Training Calendar we need to create apex class. I named the apex class as "TrainingCalendarApex".

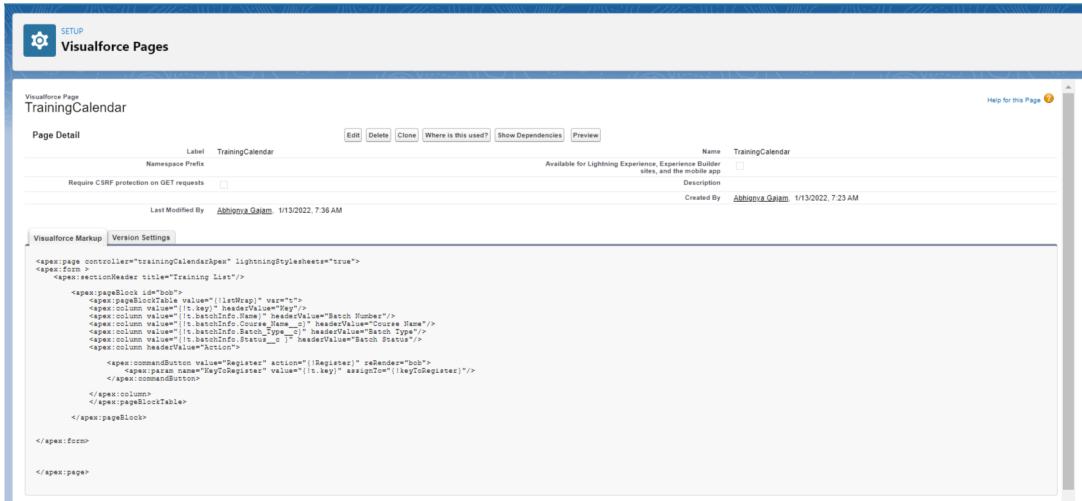
ONLINE TRAINING MANAGEMENT



The screenshot shows the Apex Classes page in the Salesforce setup. The class name is 'trainingCalendarApex'. The code implements a wrapper class for the 'Batch_Scheduling__c' object. It includes methods to register new entries and to find a record by key. The code uses a list of wrappers to store batch scheduling information.

```
1 public with sharing class trainingCalendarApex
2 {
3     public List<Wrapper> lstWrap{get;set;}
4     List<Batch_Scheduling__c> lstBatch;
5     public Integer num;
6     public Integer keyToRegister{get;set;}
7     public trainingCalendarApex()
8     {
9         num=1;
10        keyToRegister=0;
11        lstWrap=new List<Wrapper>();
12        lstBatch=[select Name__c, Course_Name__c, Batch_Type__c, Status__c from Batch_Scheduling__c where Status__c =: 'Scheduled'];
13        for(Integer i=0;i<lstBatch.size();i++)
14        {
15            lstWrap.add(new wrapper(num++,lstBatch[i]));
16        }
17    }
18
19    public PageReference Register()
20    {
21        Boolean found=false;
22        Id recordId=null;
23        for(wrapper w:lstWrap)
24        {
25            if(w.key == keyToRegister)
26            {
27                found=true;
28                recordId=w.batchInfo.Id;
29                break;
30            }
31        }
32        if(found)
33        {
34            PageReference pr=new PageReference('/apex/TrainingRegistration?id='+recordId);
35            return pr;
36        }
37        return null;
38    }
39
40    public class wrapper
41    {
42        public Integer key{get;set;}
43        public Batch_Scheduling__c batchInfo{get;set;}
44        public wrapper(Integer key,Batch_Scheduling__c batchInfo)
45        {
46            this.key=key;
47            this.batchInfo=batchInfo;
48        }
49    }
50
51 }
52
53 }
```

After creating the apex class, now its time for the visualforce page code.



The screenshot shows the Visualforce Pages page detail for 'TrainingCalendar'. The page is labeled 'TrainingCalendar' and has a namespace prefix of 'TrainingCalendar'. It is available for Lightning Experience, Experience sites, and the mobile app. The page was created by 'Abhipnya Galam' on 1/13/2022 at 7:23 AM. The page markup includes a section header for 'Training List', a table block with columns for Batch Number, Course Name, and Batch Type, and a command button for registration.

```
<apex:page controller="trainingCalendarApex" lightningStylesheets="true">
<apex:sectionHeader title="Training List"/>
<apex:pageBlock id="body">
<apex:form>
<apex:pageBlock value="{!lstWrap}" var="a">
<apex:column value="{!a.key}" headerValue="Key"/>
<apex:column value="{!a.batchInfo.Course_Name__c}" headerValue="Batch Number"/>
<apex:column value="{!a.batchInfo.Batch_Type__c}" headerValue="Course Name"/>
<apex:column value="{!a.batchInfo.Status__c}" headerValue="Batch Status"/>
<apex:column headerValue="Action">
<apex:commandButton value="PageGet" action="!key=a.key" reRender="body">
<apex:param name="key" value="{!a.key}" assignTo="(!keyToRegister)"/>
</apex:commandButton>
</apex:column>
</apex:pageBlock>
</apex:form>
</apex:page>
```

ONLINE TRAINING MANAGEMENT

Upload the Screenshot the Milestone / Activities:

The screenshot shows a web-based application interface for managing training batches. At the top, there's a navigation bar with links like Home, New Course, Trainer Registration, Student Registration, Meeting Registration, Batch Scheduling Registration, Search Course, Search Student, and Training Calendar. Below the navigation is a search bar and a toolbar with various icons. The main content area is titled 'Training List' and displays a table of training batches. The columns are KEY, BATCH NUMBER, COURSE NAME, BATCH TYPE, BATCH STATUS, and ACTION. There are three entries:

KEY	BATCH NUMBER	COURSE NAME	BATCH TYPE	BATCH STATUS	ACTION
1	B-000001	Python Developer	Weekend	Scheduled	<button>Register</button>
2	B-000004	PHP development	Weekend	Scheduled	<button>Register</button>
3	B-000005	Salesforce Development	Weekday	Scheduled	<button>Register</button>

Topic 2: Create validate delete Trigger on Trainer object.

Milestones / Activities: Creating validate delete Trigger on Trainer object.

Detailed Description: Here we are creating a apex trigger to validate to delete the trainer on trainer object.

Upload the Screenshot the Milestone / Activities:

The screenshot shows the Apex trigger code for validating the deletion of a Trainer record. The trigger is named 'ValidateDeleteOnTrainer' and is defined on the 'Trainer__c' object. It checks if there are any active batches associated with the trainer. If there are, it adds an error message and prevents the deletion. If there are no active batches, it sends an email notification to the trainer's email address.

```
trigger ValidateDeleteOnTrainer on Trainer__c (before delete)
{
    for(Trainer__c t:Trigger.old)
    {
        List<Batch_Scheduling__c> lstBatch=[select Name,Status__c from Batch_Scheduling__c where Trainer_Id__c =: t.Id];
        Integer flag=0;
        for(Batch_Scheduling__c b: lstBatch)
        {
            if(b.Status__c!= 'Completed')
            {
                flag=1;
                break;
            }
        }
        if(flag==1)
        {
            t.addError('Cannot Delete record, Active Batch found for this trainer');
        }
        else
        {
            Messaging.SingleEmailMessage lstEmail=new List<Messaging.SingleEmailMessage>();
            Messaging.SingleEmailMessage mail= new Messaging.SingleEmailMessage();
            mail.setToAddress(new String[] {t.Email__c});
            mail.setReplyTo('abhiinya3003@gmail.com');
            mail.setPlainTextBody('Your details removed successfully from our database');
            mail.setSenderDisplayName('Team Smart Internz');
            mail.setSubject('Successfully Deleted');
            lstEmail.add(mail);
        }
    }
}
```

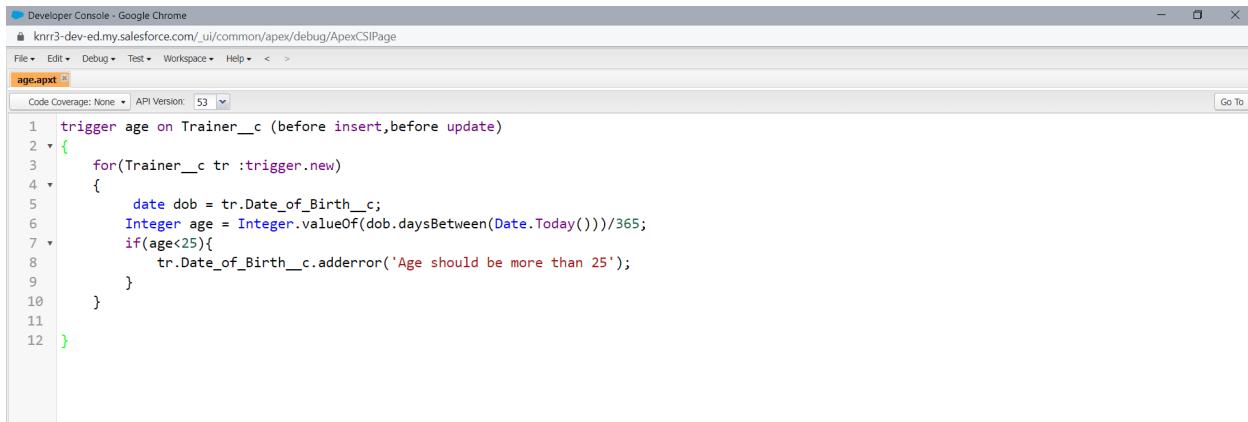
ONLINE TRAINING MANAGEMENT

Topic 3: Create the trigger on trainer object.

Milestones / Activities: Creating the trigger on trainer object.

Detailed Description: Here we are creating a apex trigger to validate the trainer age on trainer object.

Upload the Screenshot the Milestone / Activities:



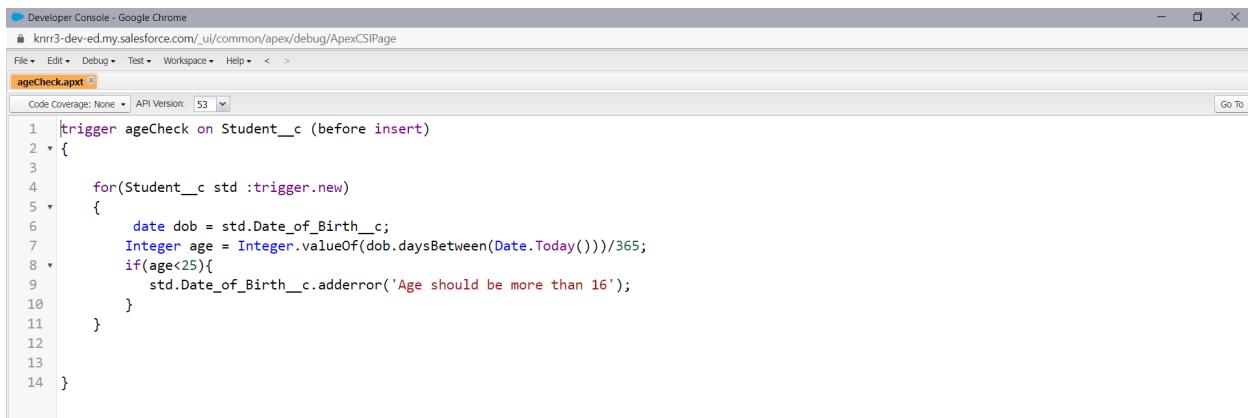
```
Developer Console - Google Chrome
knnr3-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
age.apex
Code Coverage: None ▾ API Version: 53 ▾ Go To
1 trigger age on Trainer__c (before insert,before update)
2 {
3     for(Trainer__c tr :trigger.new)
4     {
5         date dob = tr.Date_of_Birth__c;
6         Integer age = Integer.valueOf(dob.daysBetween(Date.Today()))/365;
7         if(age<25){
8             tr.Date_of_Birth__c.adderror('Age should be more than 25');
9         }
10    }
11 }
12 }
```

Topic 4: Create the trigger on student object.

Milestones / Activities: Creating the student on trainer object.

Detailed Description: Here we are creating a apex trigger to validate the student age on student object.

Upload the Screenshot the Milestone / Activities:



```
Developer Console - Google Chrome
knnr3-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
ageCheck.apex
Code Coverage: None ▾ API Version: 53 ▾ Go To
1 trigger ageCheck on Student__c (before insert)
2 {
3
4     for(Student__c std :trigger.new)
5     {
6         date dob = std.Date_of_Birth__c;
7         Integer age = Integer.valueOf(dob.daysBetween(Date.Today()))/365;
8         if(age<16){
9             std.Date_of_Birth__c.adderror('Age should be more than 16');
10        }
11    }
12 }
13 }
14 }
```

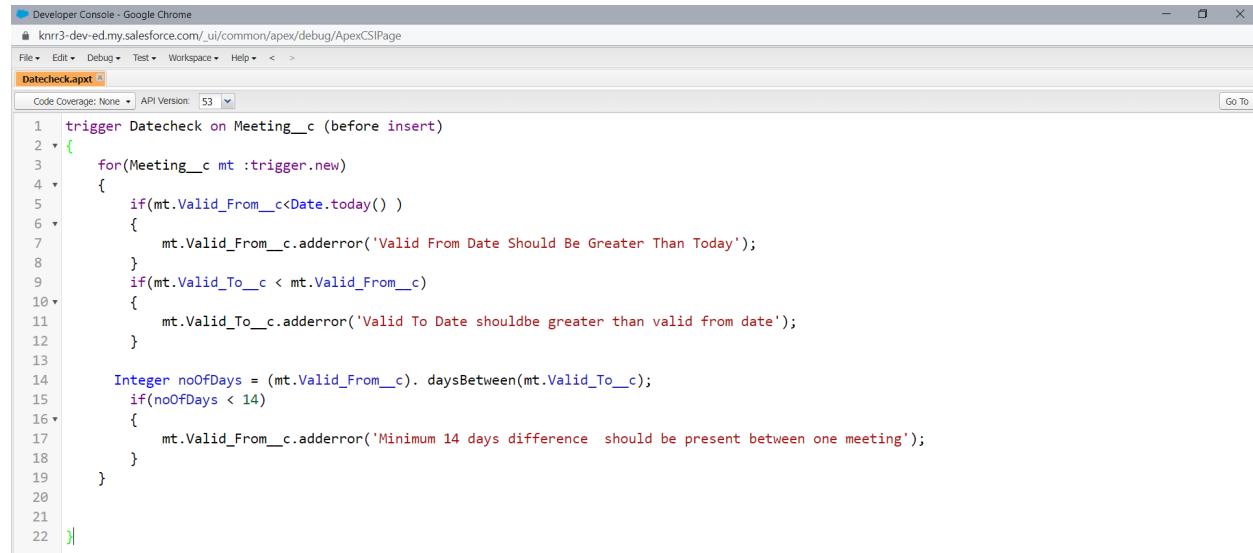
ONLINE TRAINING MANAGEMENT

Topic 5: Create the trigger on Meeting object.

Milestones / Activities: Creating the Meeting on trainer object.

Detailed Description: Here we are creating a apex trigger to validate the date check on meeting object.

Upload the Screenshot the Milestone / Activities:



The screenshot shows the Salesforce Developer Console interface. The title bar reads "Developer Console - Google Chrome" and the URL is "knnr3-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage". The menu bar includes File, Edit, Debug, Test, Workspace, Help, and a Go To button. The main area is titled "Datecheck.apxt" and contains the following Apex trigger code:

```
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');
        }
    }
}
```

Day 10:

Topic 1: Create Training Registrarion Page using custom Controller.

Milestones / Activities: Creating Training Registration Page using custom Controller.

Detailed Description: The use of this Training Registration Page is for the students to register for their training for particular courses.

To create a Training Registration visualforce page, we need to write the visualforce page code.

To write the visualforce page code for Training Registration we need to create apex class. I named the apex class as "TrainingRegistationApex".

ONLINE TRAINING MANAGEMENT

The screenshot shows the Apex Class Detail page for the class `TrainerRegistrationApex`. The page includes tabs for Apex Class Detail, Class Body, Class Summary, Version Settings, and Trace Flags. The Class Body tab is active, displaying the following Apex code:

```
1 public with sharing class TrainerRegistrationApex {
2     public Id recId{get;set;}
3     public Trainer_Enrollment__c batchInfo{get;set;}
4     public String studEmail{get;set;}
5     public Boolean studentExist{get;set;}
6     public Boolean display{get;set;}
7     public Student_Environment__c studEnv;
8
9     @Future
10    public void TrainerRegistrationApex();
11
12    public TrainerRegistrationApex() {
13        //fetch parameter value from url and store it in variable
14        recId=ApexPages.currentPage.getParameters().get('id');
15        batchInfo = [SELECT Status__c,Course_Type__c,Course_Name__c,End_Date__c,Category__c,Course_URL__c FROM Trainer_Enrollment__c WHERE Id = :recId];
16        studEnv = [SELECT Environment__c FROM Student_Environment__c WHERE Id = :recId];
17        display=false;
18    }
19
20    @future
21    void searchStudent();
22
23    public PageReference searchStudent() {
24        Integer found=0;
25        if(studEmail!=null) {
26            ApexPages.addMessage(new ApexPages.message(ApexPages.severity.Error,'No Email Id Provided for Search'));
27            studentExist=false;
28        }
29        else {
30            studList=[SELECT Date_of_Birth__c,c_Email__c,First_Name__c,Last_Name__c,Phone_Number__c FROM Student__c WHERE Email__c = :studEmail];
31            if(studList.size()!=0) {
32                studEnvInfo=[SELECT Batch__c,Student_Email__c FROM Student_Environment__c WHERE Student_Id__c = studList[0].id];
33                if(studEnvInfo.size()!=0) {
34                    for(Student_Environment__c c:studEnvInfo) {
35                        if(c.Batch__c == batchInfo.id) {
36                            ApexPages.addMessage(new ApexPages.message(ApexPages.severity.Warning,'Student Already Registered in this batch'));
37                            display=true;
38                        }
39                    }
40                }
41                else {
42                    ApexPages.addMessage(new ApexPages.message(ApexPages.severity.Info,'Student Not Registered in any batch'));
43                    display=true;
44                }
45            }
46            else {
47                ApexPages.addMessage(new ApexPages.message(ApexPages.severity.Error,'No Student Registered with this email id'));
48                display=false;
49            }
50        }
51        studentExist=false;
52    }
53
54    public void clickToRegister() {
55        Student_Environment__c s=new Student_Environment__c();
56        s.Student_ID__c=sstudEnv.id;
57        s.Batch__c=batchInfo.id;
58        ApexPages.addMessage(new ApexPages.message(ApexPages.severity.Confirm,'Student Enrollment Done Successfully'));
59        studentExist=true;
60    }
61
62    //Custom code to send a acknowledgement email to student
63    @future
64    void sendEmail() {
65        List<Messaging.SingleEmailMessage> lstEmail=new List<Messaging.SingleEmailMessage>();
66        Messaging.SingleEmailMessage mail=new Messaging.SingleEmailMessage();
67        mail.setTo('sumitgoyal@gmail.com');
68        mail.setSubject('Success for Batch Number'+batchInfo[0].Name);
69        mail.setPlainText('Thank you for your interest');
70        mail.setSubject('Registered Successfully');
71        if(studEnvInfo.size()!=0) {
72            for(Student_Environment__c c:studEnvInfo) {
73                try {
74                    Messaging.SingleEmailMessage mail1=new Messaging.SingleEmailMessage();
75                    mail1.setTo(c.Student_Email__c);
76                    mail1.setSubject('Success for Batch Number'+batchInfo[0].Name);
77                    mail1.setPlainText('Thank you for your interest');
78                    mail1.setSubject('Registered Successfully');
79                    lstEmail.add(mail1);
80                }
81            }
82        }
83        Messaging.sendEmail(lstEmail);
84        System.debug('Process Completed Successfully');
85    }
86
87    public PageReference registerStudent() {
88        return null;
89    }
90
91 }
92
93 }
```

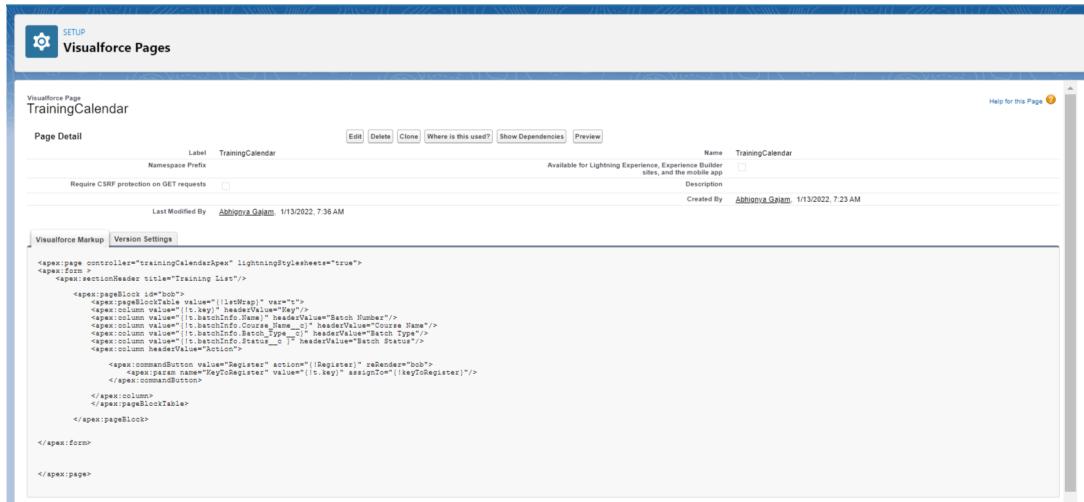
The page also displays the following metadata information:

Name	Value	Status	Active
Created By	Ananya Goyal	Code Coverage	0% (0/6)
Last Modified By	Ananya Goyal	Last Modified By	1/14/2022, 4:44 AM

The browser status bar at the bottom right shows the time as 18:14, temperature as 27°C, and date as 14-01-2022.

ONLINE TRAINING MANAGEMENT

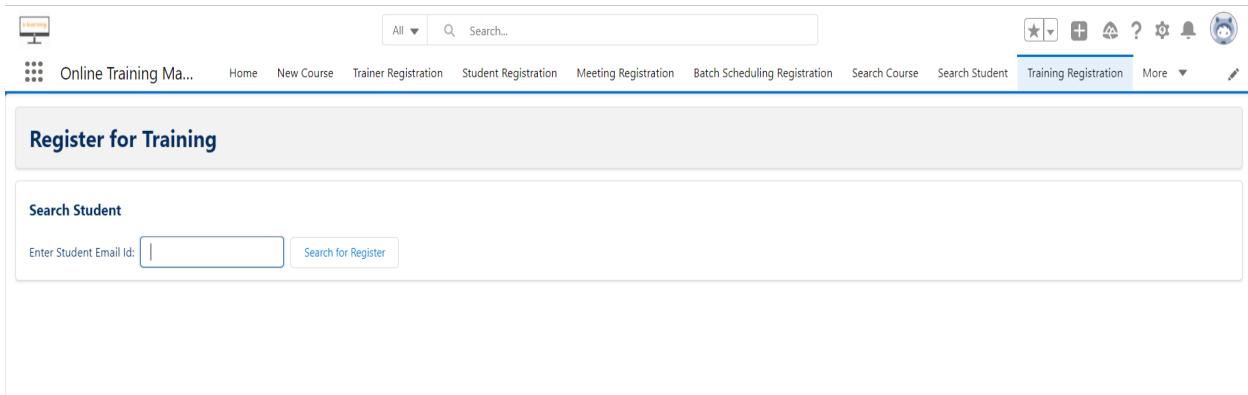
After creating the apex class, now its time for the visualforce page code.



The screenshot shows the Salesforce Visualforce Pages setup interface. A page named "TrainingCalendar" is selected. The "Page Detail" section includes fields for Label ("TrainingCalendar"), Namespace Prefix, and Description. It also shows the page is available for Lightning Experience, Experience Builder sites, and the mobile app. The "Visualforce Markup" tab is active, displaying the Apex code for the page:

```
<apex:page controller="TrainingCalendarController" lightningStyleSheets="true">
<apex:sectionHeader title="Training List"/>
<apex:pageBlock id="body">
<apex:column value="{:batchKey}" headerValue="Key"/>
<apex:column value="{:batchInfo.Course_Name__c}" headerValue="Batch Number"/>
<apex:column value="{:batchInfo.Course_Name__c}" headerValue="Course Name"/>
<apex:column value="{:batchInfo.Batch_Type__c}" headerValue="Batch Type"/>
<apex:column value="{:batchInfo.Batch_Status__c}" headerValue="Batch Status"/>
<apex:column headerValue="Action"/>
<apex:commandButton value="DeRegister" action="{!DeRegister}" reRender="body"/>
</apex:column>
</apex:pageBlock>
</apex:form>
</apex:page>
```

Upload the Screenshot the Milestone / Activities:



The screenshot shows the "Online Training Management" application interface. The top navigation bar includes links for Home, New Course, Trainer Registration, Student Registration, Meeting Registration, Batch Scheduling Registration, Search Course, Search Student, Training Registration, and More. The main content area features two sections: "Register for Training" and "Search Student". The "Search Student" section contains a search input field labeled "Enter Student Email Id:" and a "Search for Register" button.