

Day 1: 03/01/2022, Monday

Topic: Create your salesforce developer org to get started

Milestone / Activities: Creating Developer Account

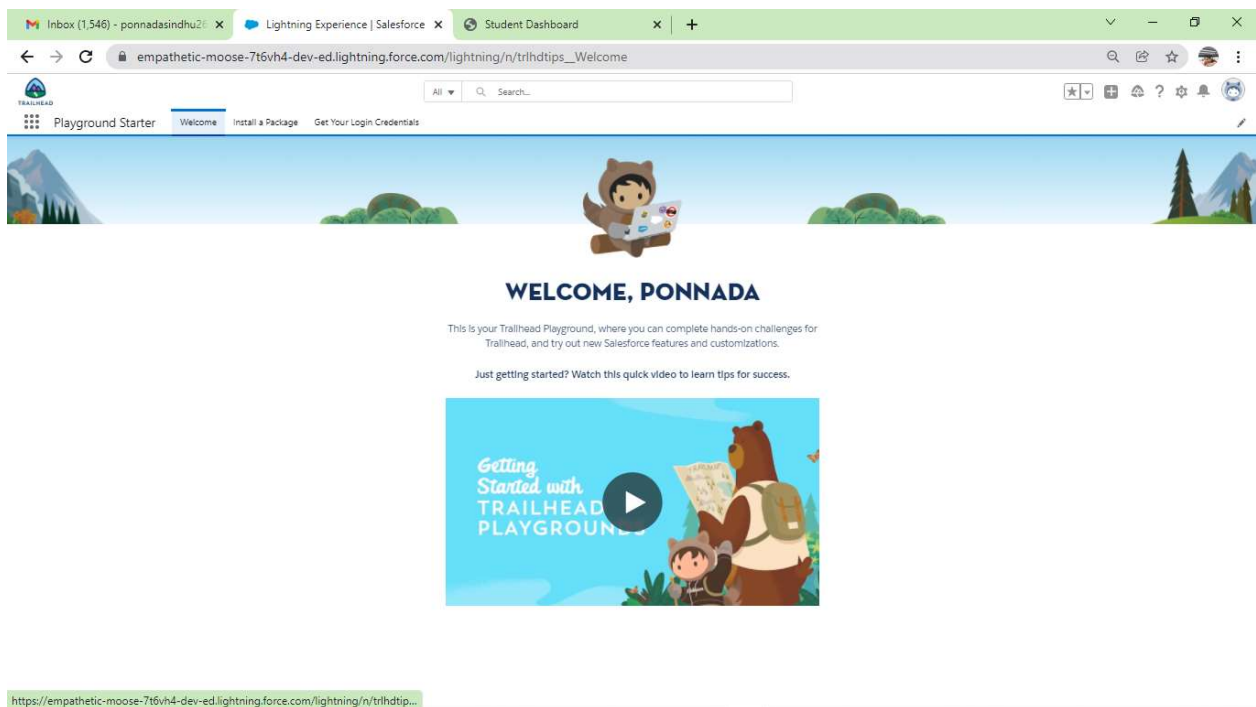
Account Activation

Login to your Salesforce account

Detailed Description:

First sign up the developers.salesforce.com with all the required information, after clicking the sign up button I will get an email for activating the developer account, then I verify the account, then finally I login to the salesforce account

Upload the Screenshot of Milestone / Activity:



Day 2: 04/01/2022, Tuesday

Topic: Custom Object creation

Milestone / Activities: Create The Custom Objects For The Application

Create Custom Fields On Course Object

Create Custom Fields On Trainer Object

Create Custom Fields On Student Object

Create Custom Fields On Meeting Object

Create Custom Fields On Batch Object

Detailed Description: After login to the salesforce account create a custom objects on object manager. The Custom Objects are Courses, Trainer, Student, Meeting and Batch/Class Scheduling.

After Creating a Course Object the following are the fields we are created. Course ID using the data type 'Auto Number', 'Course Name' using the data type 'Text', 'Course Duration' using data type 'Picklist', 'Course Fees' using data type 'Currency', 'Course Description' using data type 'Text', 'Course Category' using data type 'Picklist' and finally 'Course Trainer' using 'Master Detail Relationship'.

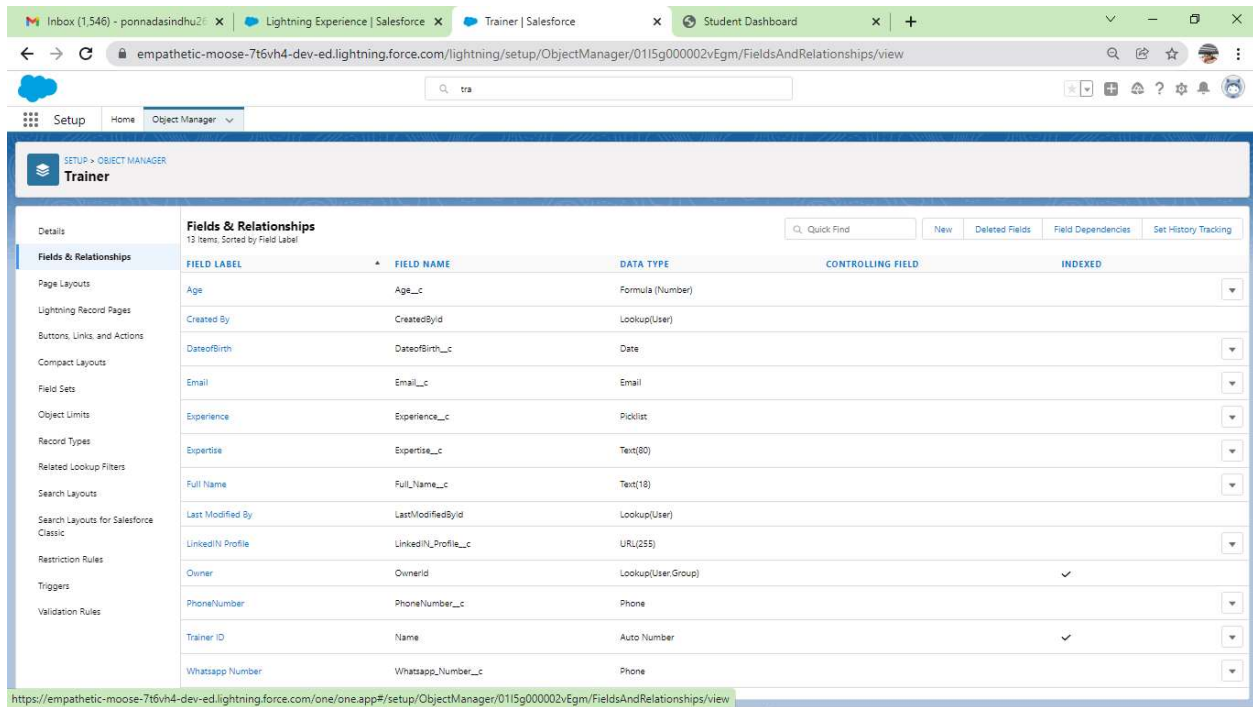
Upload the Screenshot of Milestone / Activity: (Create Custom Fields On Course Object)

The screenshot shows the Salesforce Object Manager interface for the 'Course' object. The 'Fields & Relationships' tab is active, displaying a table of 9 fields. The table has columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed. The fields are: Course Category (Picklist), Course Description (Long Text Area), Course Duration (Picklist), Course Fees (Currency), Course ID (Auto Number), Course Name (Text), Course Trainer (Master-Detail), Created By (Lookup), and Last Modified By (Lookup).

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(6 2)		
Course ID	Name	Auto Number		✓
Course Name	Course_Name__c	Text(18)		
Course Trainer	Course_Trainer__c	Master-Detail(Trainer)		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		

After Creating a Trainer Object the following are the fields we are created. Trainer ID, Full Name, Email, Phone Number, Whatsapp Number, Date of Birth, Age, Experience, LinkedIn Profile and Expertise.

Upload the Screenshot of Milestone / Activity: (Create Custom Fields On Trainer Object)



The screenshot shows the Salesforce Object Manager interface for the 'Trainer' object. The 'Fields & Relationships' tab is selected, displaying a list of 13 custom fields. The fields are sorted by Field Label. The table includes columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed status.

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	Email__c	Email		
Experience	Experience__c	Picklist		
Expertise	Expertise__c	Text(80)		
Full Name	Full_Name__c	Text(18)		
Last Modified By	LastModifiedById	Lookup(User)		
LinkedIn Profile	LinkedIn_Profile__c	URL(255)		
Owner	OwnerId	Lookup(User/Group)		✓
PhoneNumber	PhoneNumber__c	Phone		
Trainer ID	Name	Auto Number		✓
Whatsapp Number	Whatsapp_Number__c	Phone		

The following are the fields that are created on Student Object that are Student ID, First Name, Last Name, Email, Phone Number, Date of Birth, Whatsapp Number, Enrollment Date, Student and Enrollment.

Upload the Screenshot of Milestone / Activity: (Create Custom Fields On Student Object)

The screenshot shows the Salesforce Object Manager for the 'Student' object. The 'Fields & Relationships' tab is selected, displaying a list of 11 fields. The fields are sorted by Field Label. The table includes columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed status.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Date Of Birth	Date_Of_Birth__c	Date		
Email	Email__c	Email		
Enrollment Date	Enrollment_Date__c	Formula (Date)		
First Name	First_Name__c	Text(18)		
Last Modified By	LastModifiedById	Lookup(User)		
Last Name	Last_Name__c	Text(18)		
PhoneNumber	PhoneNumber__c	Phone		
Student Enrollment	Student_Enrollment__c	Master-Detail(Batch/Class Scheduling)		✓
Student ID	Name	Auto Number		✓
Whatsapp Number	Whatsapp_Number__c	Phone		

The following are the fields we are created on Meeting Object that are Meeting ID, Valid From, Valid To, Active and Meeting link.

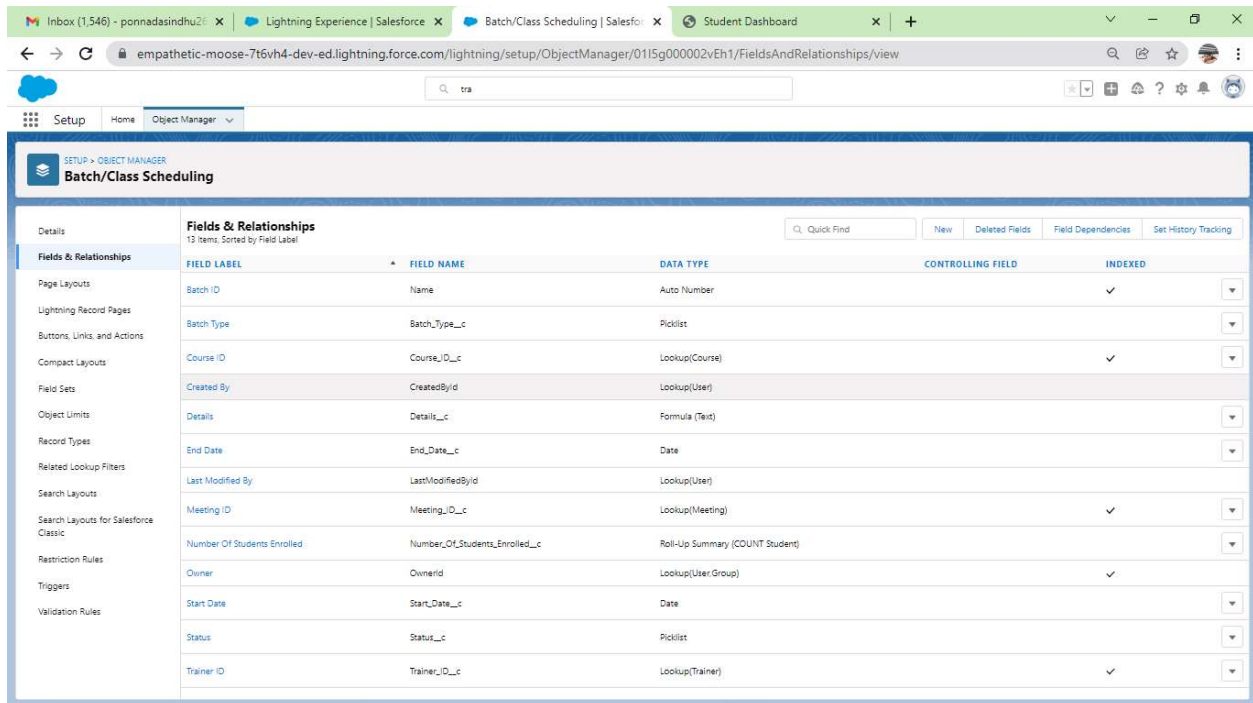
Upload the Screenshot of Milestone / Activity: (Create Custom Fields On Meeting Object)

The screenshot shows the Salesforce Object Manager for the 'Meeting' object. The 'Fields & Relationships' tab is selected, displaying a list of 8 fields. The fields are sorted by Field Label. The table includes columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed status.

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		

The following are the fields we are created on Batch Object that are Batch ID, Batch Type, Start Date, End Date, Course ID, Meeting ID, Trainer ID, Status, Number Of Students Enrolled and Details.

Upload the Screenshot of Milestone / Activity: (Create Custom Fields On Batch Object)



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)		
Owner	OwnerId	Lookup(User/Group)		✓
Start Date	Start_Date__c	Date		
Status	Status__c	Picklist		
Trainer ID	Trainer_ID__c	Lookup(Trainer)		✓

Day 3: 05/01/2022, Wednesday

Topic: Create Validation Rules

Milestone / Activities: Create Validation Rules on Course Object

Create Validation Rules on Trainer Object

Create Validation Rules on Student Object

Create Validation Rules on Meeting Object

Detailed Description:

From Object Manager find out the Course Object and create a validation rules on Course Object. The first validation rule are Course Fees cannot be negative by using the formula $\text{Course_Fees_c} < 0$. The second validation rule are Course Fees should be less than or equal to 3000 by using the formula $\text{Course_Fees_c} \leq 3000$.

Upload the Screenshot of Milestone / Activity: (Create Validation Rules on Course Object)

The screenshot shows the Salesforce Setup interface for the 'Course' object. The left sidebar lists various setup options, with 'Validation Rules' selected. The main content area displays a table of validation rules for the 'Course' object. The table has columns for Rule Name, Error Location, Error Message, Active status, and Modified By. Two rules are listed: 'new2' and 'newone'.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
new2	Top of Page	should be greater	✓	Ponnada Sindhu, 1/19/2022, 10:39 AM
newone	Top of Page	fees can't be negative	✓	Ponnada Sindhu, 1/19/2022, 10:37 AM

Create a validation rules on Trainer Object. The validation rules are FullName, Phone Number, Date of Birth and Experience fields should not be blank, age should not be less than 25 years and Phone Number and Whatsapp Number should have min 10 digits and max 15 digits.

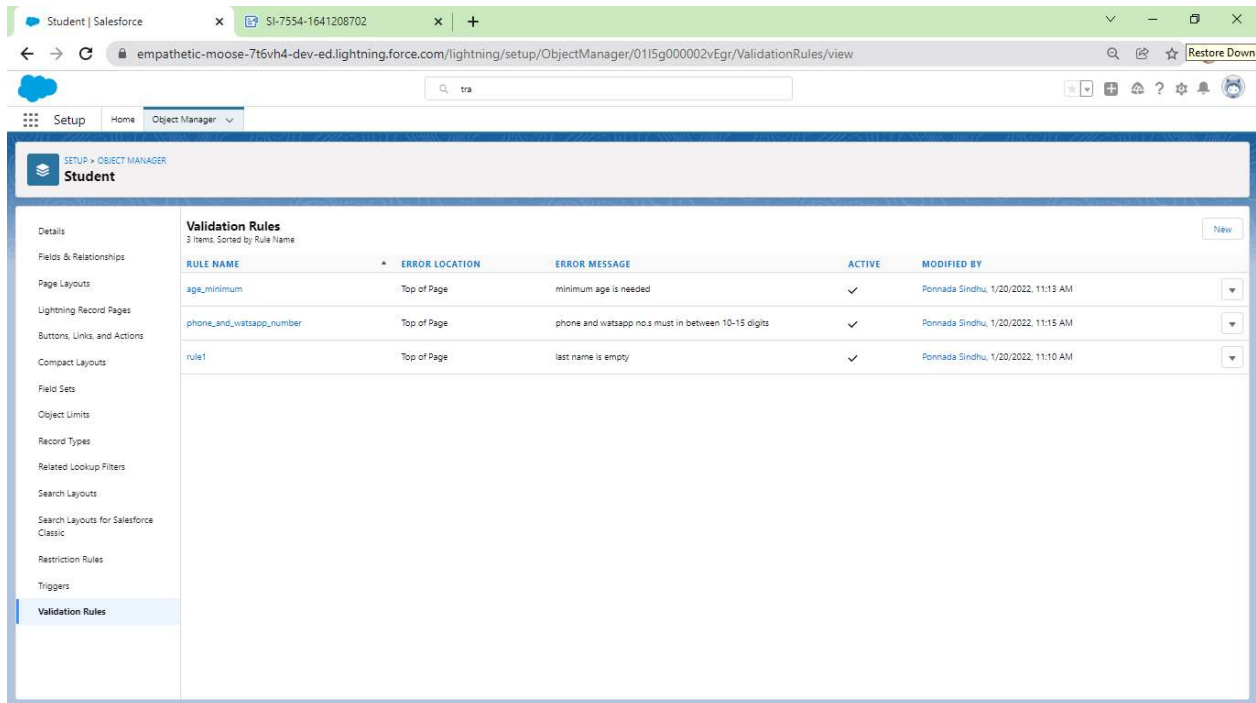
Upload the Screenshot of Milestone / Activity: (Create Validation Rules on Trainer Object)

The screenshot shows the Salesforce Setup interface for the 'Trainer' object. The left sidebar lists various setup options, with 'Validation Rules' selected. The main content area displays a table of validation rules for the 'Trainer' object. The table has columns for Rule Name, Error Location, Error Message, Active status, and Modified By. Three rules are listed: 'rule1', 'rule2', and 'rule3'.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
rule1	Top of Page	fill the blank fields	✓	Ponnada Sindhu, 1/19/2022, 10:55 AM
rule2	Top of Page	age limit	✓	Ponnada Sindhu, 1/19/2022, 11:10 AM
rule3	Top of Page	must be in between 10 and 15 digits	✓	Ponnada Sindhu, 1/19/2022, 11:19 AM

Create a validation rules on Student Object. The validation rules are Last Name should not be blank, Minimum 16 years of age and PhoneNumber and Whatsapp Number should have minimum 10 digit.

Upload the Screenshot of Milestone / Activity: (Create Validation Rules on Student Object)

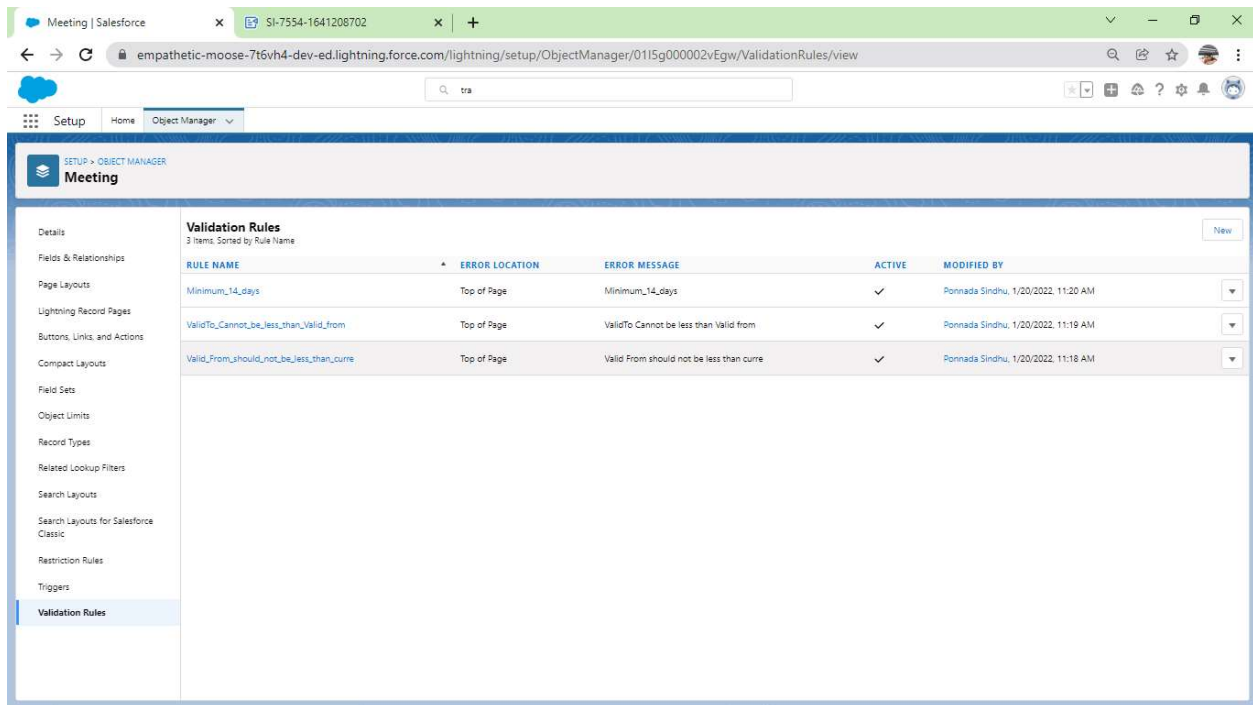


The screenshot shows the Salesforce Object Manager interface for the 'Student' object. The 'Validation Rules' section is active, displaying a table with three rules. The table has columns for Rule Name, Error Location, Error Message, Active status, and Modified By. The rules are: 'age_minimum' (minimum age is needed), 'phone_and_watsapp_number' (phone and whatsapp nos must in between 10-15 digits), and 'rule1' (last name is empty). All rules are active and were modified by Ponnada Sindhu on 1/20/2022.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
age_minimum	Top of Page	minimum age is needed	✓	Ponnada Sindhu, 1/20/2022, 11:13 AM
phone_and_watsapp_number	Top of Page	phone and whatsapp nos must in between 10-15 digits	✓	Ponnada Sindhu, 1/20/2022, 11:15 AM
rule1	Top of Page	last name is empty	✓	Ponnada Sindhu, 1/20/2022, 11:10 AM

Create a Validation rules on Meeting Object. The validation rules are Valid from should not be less than current date, Valid to cannot be less than valid from and minimum 14 days difference in between valid from and valid to.

Upload the Screenshot of Milestone / Activity: (Create Validation Rules on Meeting Object)



Day 5: 07/01/2022, Friday

Topic: Process Automation

Milestone / Activities: Creating Work Flow Rules On Meeting Object

Creating Work Flow Rules On Class/Batch Object

Detailed Description:

Search For Work Flow Rules on Object Manager. Create a Work Flow Rules on Meeting Object and enter a Rule name and in the formula section `TODAY() - Valid_to__c = 1` and select active field in workflow action.

Create another Work Flow on Meeting Object and enter a rule name and in formula evaluates enter `TODAY() - Valid_to__c = 1` and enter the details in Email Template in workflow action. Finally active the workflow actions.

Upload the Screenshot of Milestone / Activity: (Creating Work Flow Rules On Meeting Object)

Workflow Rules | Salesforce x SI-7554-1641208702 x +

empathetic-moose-7t6vh4-dev-ed.lightning.force.com/lightning/setup/WorkflowRules/page?address=%2F01Q5g000000MBPP&nodeId=WorkflowRules

Setup Home Object Manager

Workflow Rules

Workflow Rule rule 1

Workflow Rule Detail

Rule Name	rule 1	Object	Meeting
Active	✓	Evaluation Criteria	Evaluate the rule when a record is created, and any time it's edited to subsequently meet criteria
Description			
Rule Criteria	TODAY() < Valid_To__c = 1		
Created By	Ponnada Sindhu	Modified By	Ponnada Sindhu
	1/20/2022, 8:42 AM		1/20/2022, 8:51 AM

Workflow Actions

Immediate Workflow Actions

Type	Description
Field Update	reminder

Time-Dependent Workflow Actions [See an example](#)

You cannot add new time triggers to an active rule. [Deactivate This Rule](#)

Workflow Rules | Salesforce x SI-7554-1641208702 x +

empathetic-moose-7t6vh4-dev-ed.lightning.force.com/lightning/setup/WorkflowRules/page?address=%2F01Q5g000000MBPU&nodeId=WorkflowRules

Setup Home Object Manager

Workflow Rules

Workflow Rule email alert

Workflow Rule Detail

Rule Name	email alert	Object	Meeting
Active	✓	Evaluation Criteria	Evaluate the rule when a record is created, and any time it's edited to subsequently meet criteria
Description			
Rule Criteria	Valid_To__c - TODAY () = 1		
Created By	Ponnada Sindhu	Modified By	Ponnada Sindhu
	1/20/2022, 8:47 AM		1/20/2022, 8:51 AM

Workflow Actions

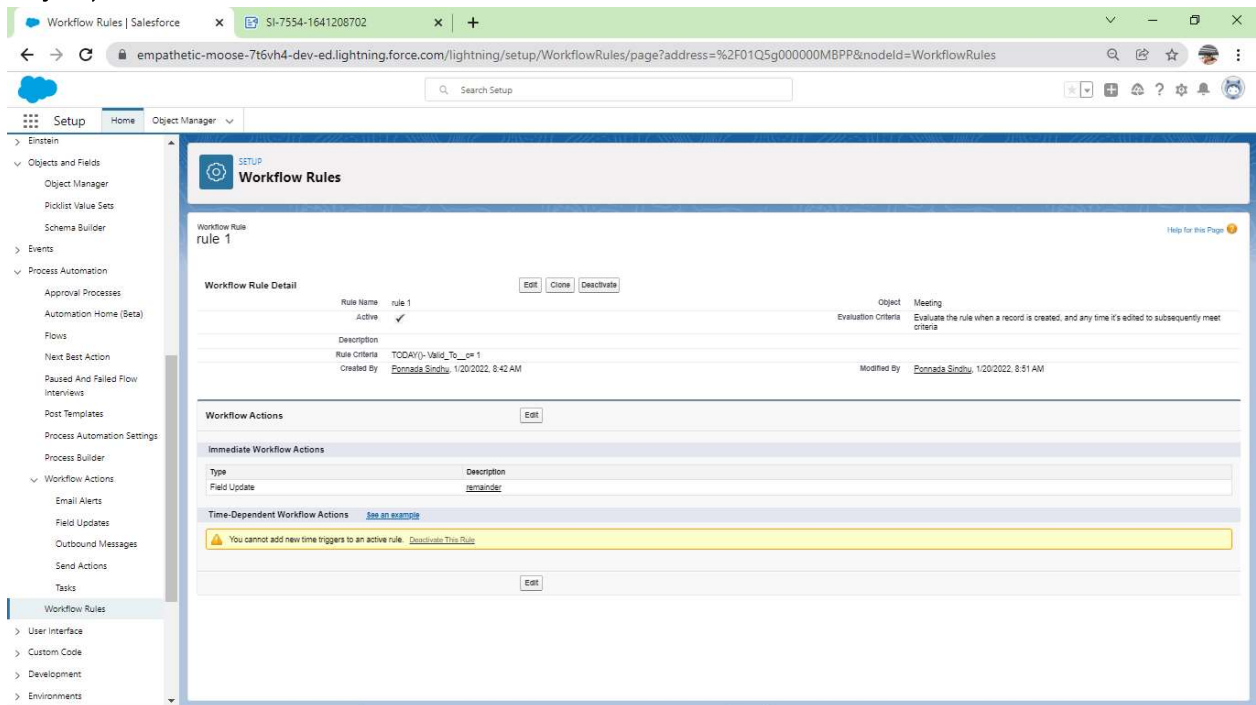
Immediate Workflow Actions

Type	Description
Email Alert	new mail

Time-Dependent Workflow Actions [See an example](#)

You cannot add new time triggers to an active rule. [Deactivate This Rule](#)

Upload the Screenshot of Milestone / Activity: (Creating Work Flow Rules On Class/Batch Object)



Day 6: 08/01/2022, Saturday

Topic: Creating Custom Pages Using VisualForce

Milestone / Activities: Creating New Course Using Custom Controller

Create New Trainer Using Custom Controller

Create New Student Using Standard Controller

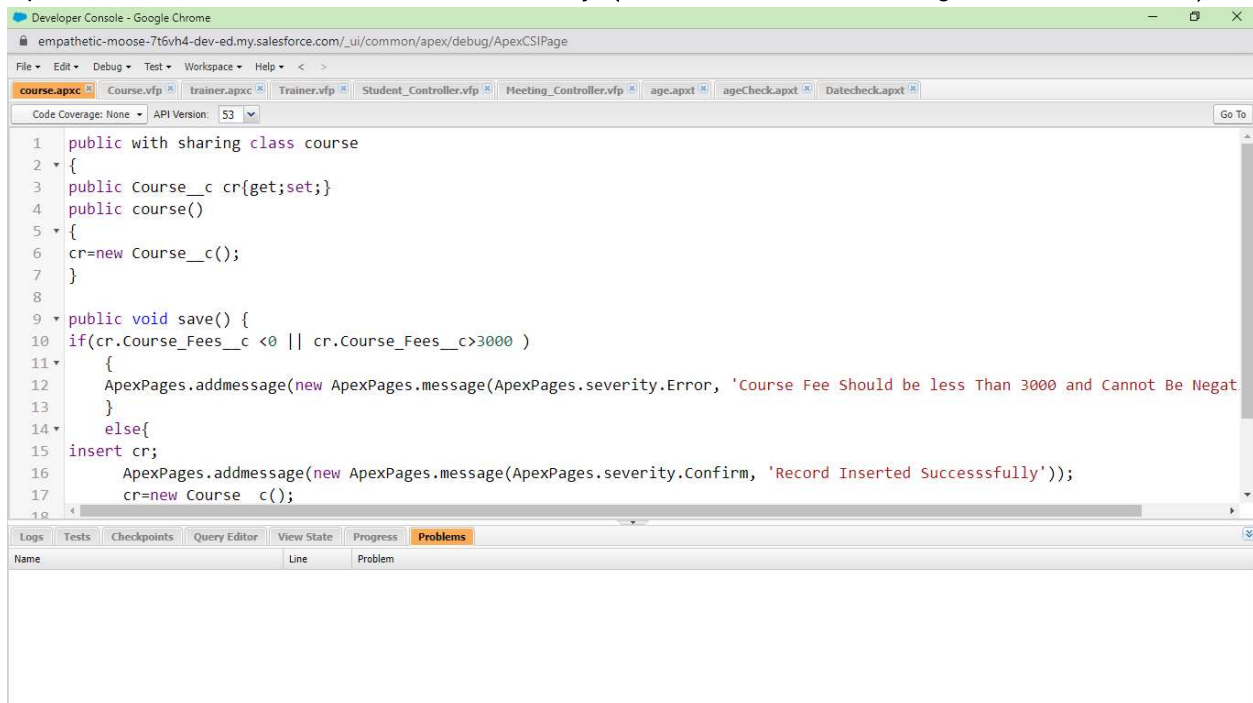
Create New Meeting Using Standard Controller

Create New Batch Using Standard Controller

Detailed Description:

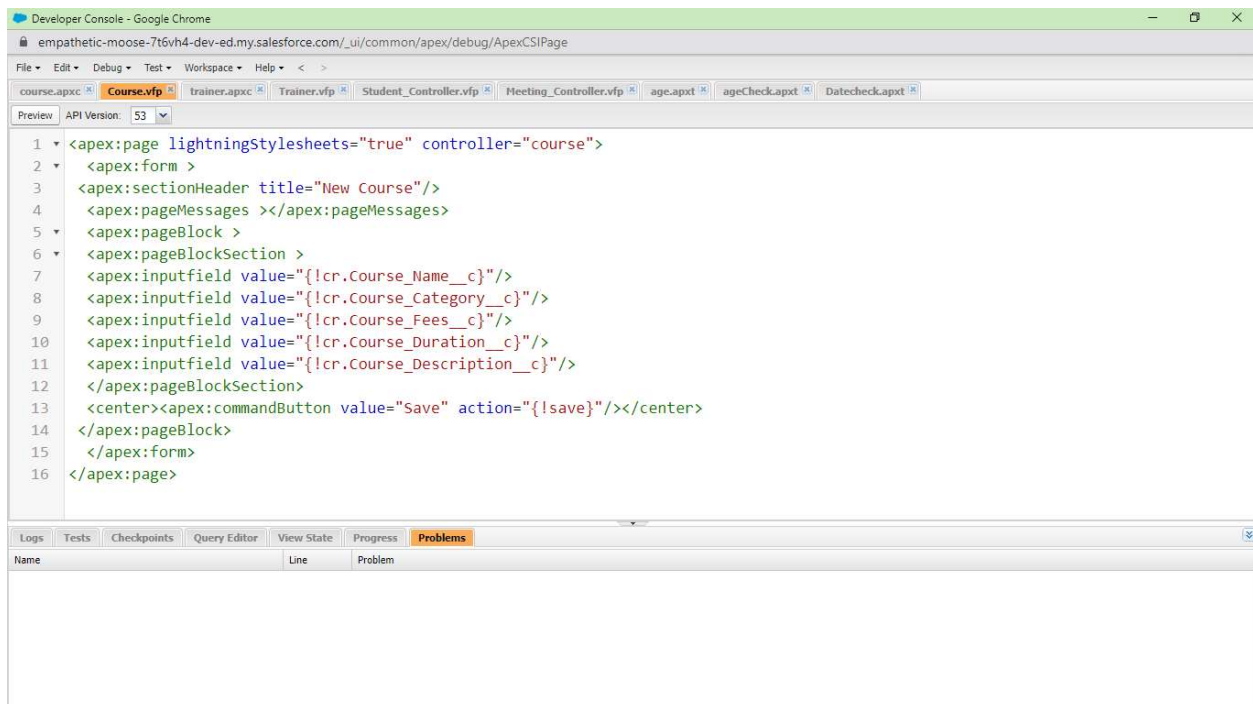
Open Developer Console and select File and then select new Visualforce page and paste the Course Visualforce code given in the task of "Creating New Course Using Custom Controller", and then select new apex class and paste the Course Controller code.

Upload the Screenshot of Milestone / Activity: (Create New Course Using Custom Controller)



The screenshot shows the Salesforce Developer Console with the Apex code for the 'course' controller. The code is as follows:

```
1 public with sharing class course
2 {
3     public Course__c cr{get;set;}
4     public course()
5     {
6         cr=new Course__c();
7     }
8
9     public void save() {
10        if(cr.Course_Fees__c <0 || cr.Course_Fees__c>3000 )
11        {
12            ApexPages.addmessage(new ApexPages.message(ApexPages.severity.Error, 'Course Fee Should be less Than 3000 and Cannot Be Negat
13        )
14        }
15        else{
16            insert cr;
17            ApexPages.addmessage(new ApexPages.message(ApexPages.severity.Confirm, 'Record Inserted Successsfully'));
18            cr=new Course__c();
19        }
20    }
```

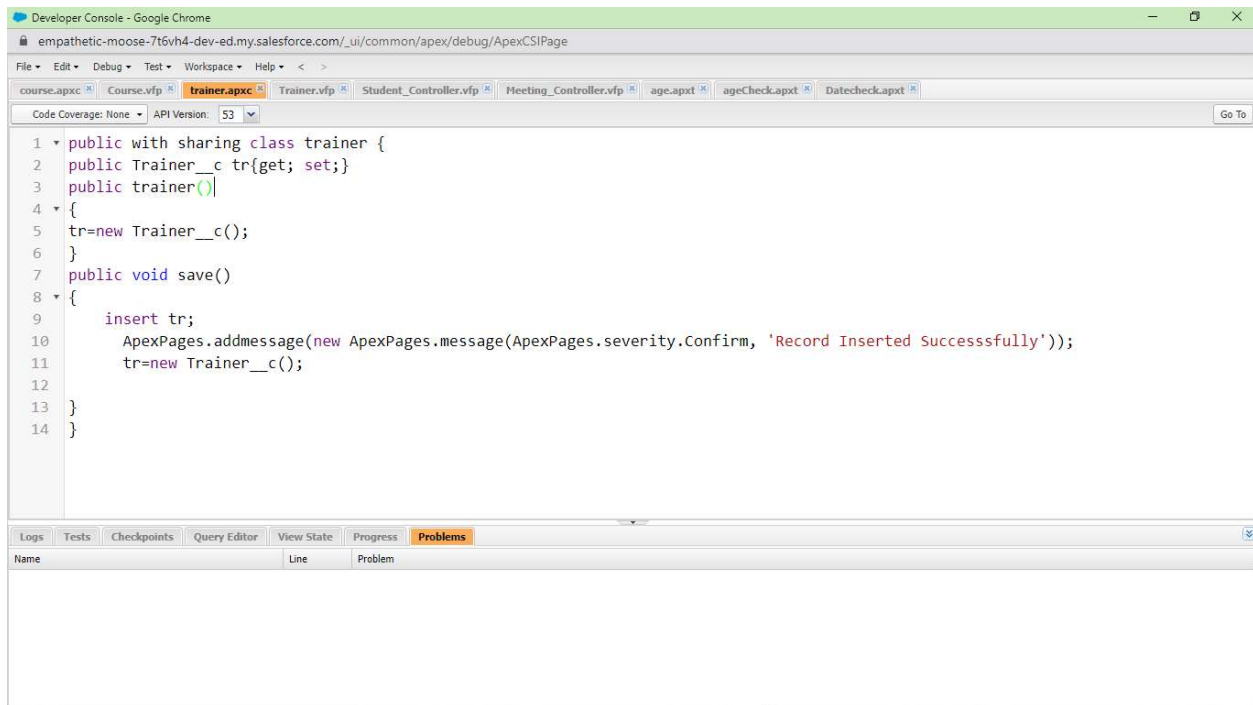


The screenshot shows the Salesforce Developer Console with the Visualforce code for the 'course' page. The code is as follows:

```
1 <apex:page lightningStylesheets="true" controller="course">
2     <apex:form >
3         <apex:sectionHeader title="New Course"/>
4         <apex:pageMessages ></apex:pageMessages>
5         <apex:pageBlock >
6             <apex:pageBlockSection >
7                 <apex:inputfield value="{!cr.Course_Name__c}"/>
8                 <apex:inputfield value="{!cr.Course_Category__c}"/>
9                 <apex:inputfield value="{!cr.Course_Fees__c}"/>
10                <apex:inputfield value="{!cr.Course_Duration__c}"/>
11                <apex:inputfield value="{!cr.Course_Description__c}"/>
12            </apex:pageBlockSection>
13            <center><apex:commandButton value="Save" action="{!save}"/></center>
14        </apex:pageBlock>
15    </apex:form>
16</apex:page>
```

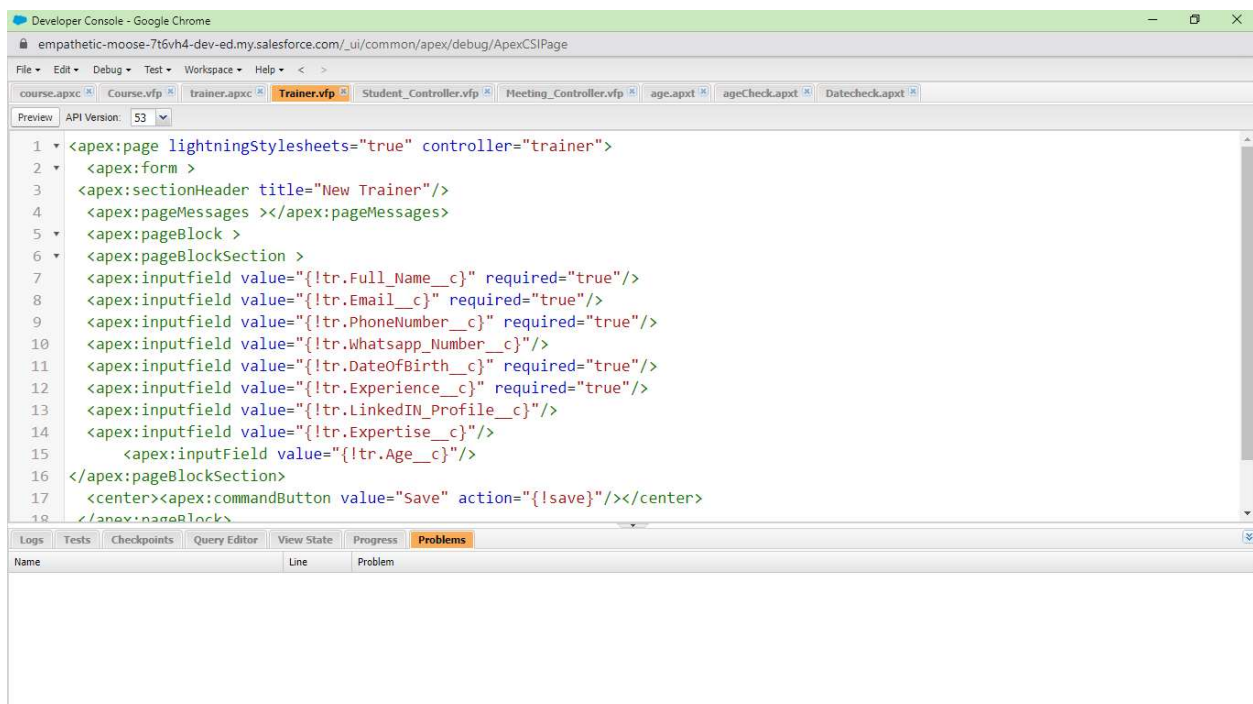
From File and then select new Visualforce page and paste the Trainer Visualforce code given in the task of "Creating New Trainer Using Custom Controller", and then select new apex class and paste the Trainer Controller code.

Upload the Screenshot of Milestone / Activity: (Create New Trainer Using Custom Controller)



The screenshot shows the Salesforce Developer Console with the Apex code for the `Trainer` class. The code is as follows:

```
1 public with sharing class trainer {  
2     public Trainer__c tr{get; set;}  
3     public trainer()  
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 Successssfully'));  
11        tr=new Trainer__c();  
12    }  
13 }  
14 }
```

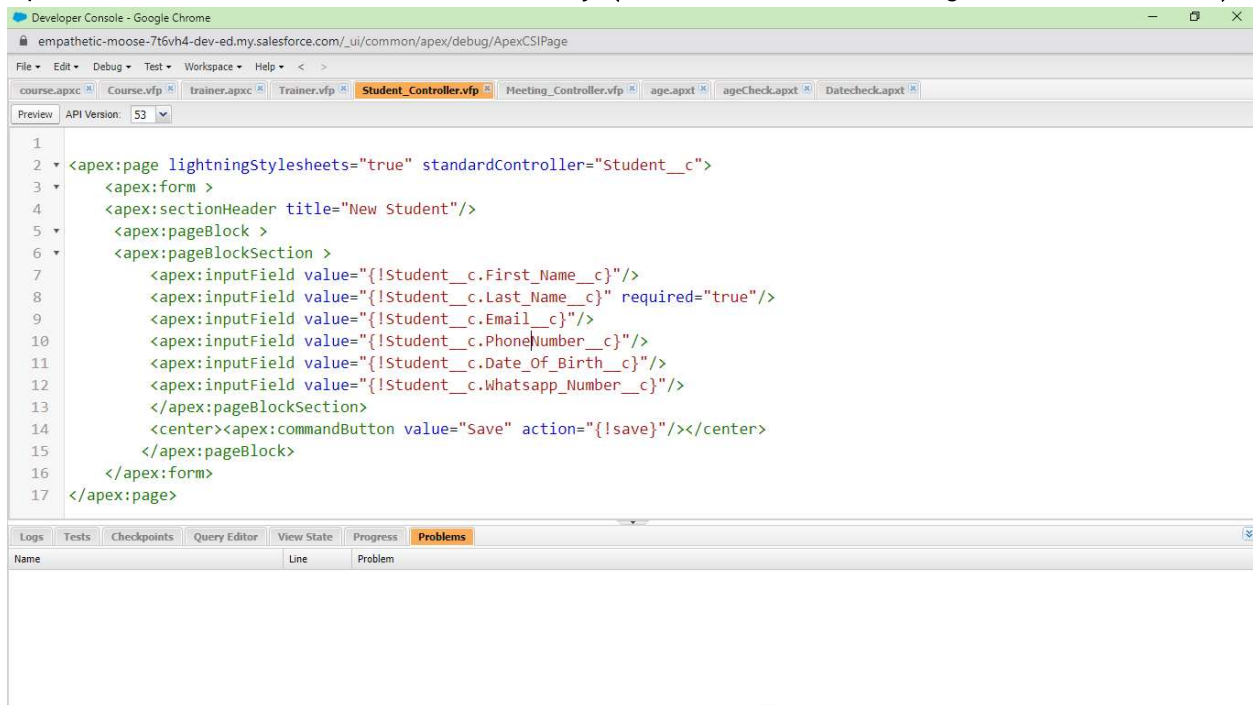


The screenshot shows the Salesforce Developer Console with the Visualforce page code for the `New Trainer` page. The code is as follows:

```
1 <apex:page lightningStyleSheets="true" controller="trainer">  
2     <apex:form >  
3         <apex:sectionHeader title="New Trainer"/>  
4         <apex:pageMessages ></apex:pageMessages>  
5         <apex:pageBlock >  
6             <apex:pageBlockSection >  
7                 <apex:inputfield value="{!tr.Full_Name__c}" required="true"/>  
8                 <apex:inputfield value="{!tr.Email__c}" required="true"/>  
9                 <apex:inputfield value="{!tr.PhoneNumber__c}" required="true"/>  
10                <apex:inputfield value="{!tr.Whatsapp_Number__c}" />  
11                <apex:inputfield value="{!tr.DateOfBirth__c}" required="true"/>  
12                <apex:inputfield value="{!tr.Experience__c}" required="true"/>  
13                <apex:inputfield value="{!tr.LinkedIN_Profile__c}" />  
14                <apex:inputfield value="{!tr.Expertise__c}" />  
15                <apex:inputField value="{!tr.Age__c}" />  
16            </apex:pageBlockSection>  
17            <center><apex:commandButton value="Save" action="{!save}" /></center>  
18        </apex:pageBlock>  
19    </apex:form>  
20 </apex:page>
```

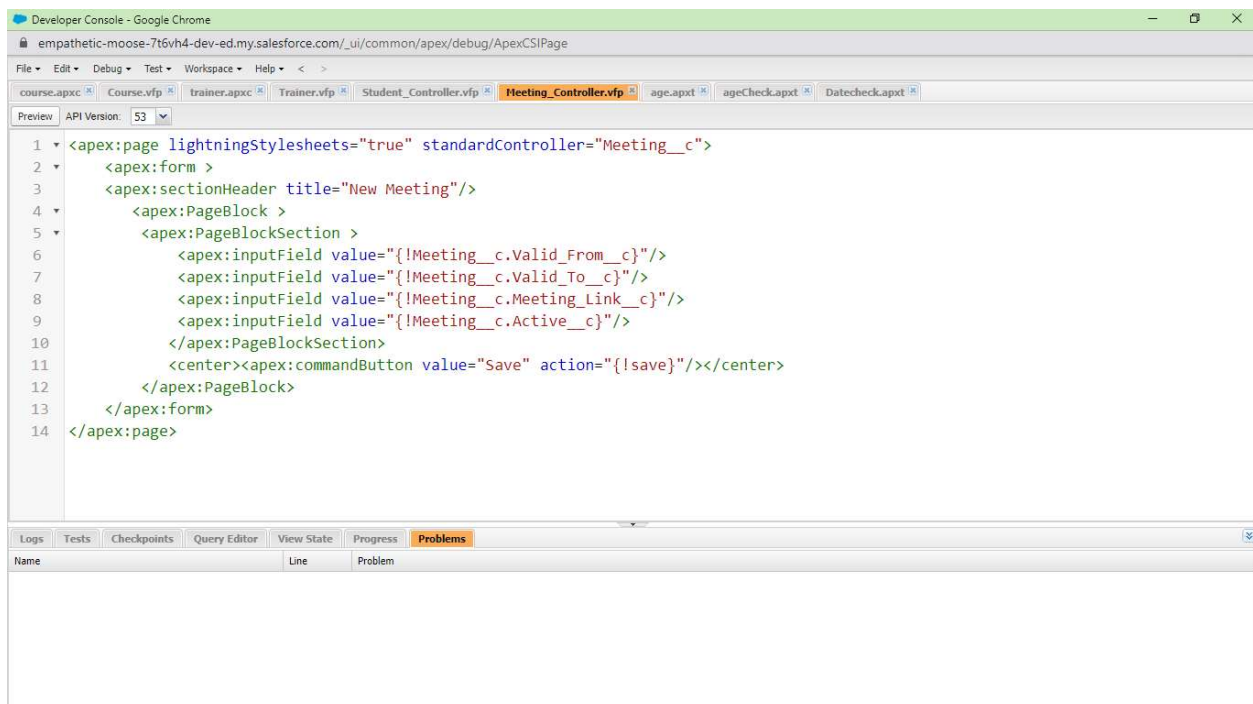
From developer console create a new VisualForce page and paste the Student Visualforce code in the visualforce page.

Upload the Screenshot of Milestone / Activity: (Create New Student Using Standard Controller)



```
1 <apex:page lightningStyleSheets="true" standardController="Student__c">
2   <apex:form >
3     <apex:sectionHeader title="New Student"/>
4     <apex:pageBlock >
5       <apex:pageBlockSection >
6         <apex:inputField value="{!Student__c.First_Name__c}"/>
7         <apex:inputField value="{!Student__c.Last_Name__c}" required="true"/>
8         <apex:inputField value="{!Student__c.Email__c}"/>
9         <apex:inputField value="{!Student__c.PhoneNumber__c}"/>
10        <apex:inputField value="{!Student__c.Date_Of_Birth__c}"/>
11        <apex:inputField value="{!Student__c.Whatsapp_Number__c}"/>
12      </apex:pageBlockSection>
13      <center><apex:commandButton value="Save" action="{!save}"/></center>
14    </apex:pageBlock>
15  </apex:form>
16</apex:page>
```

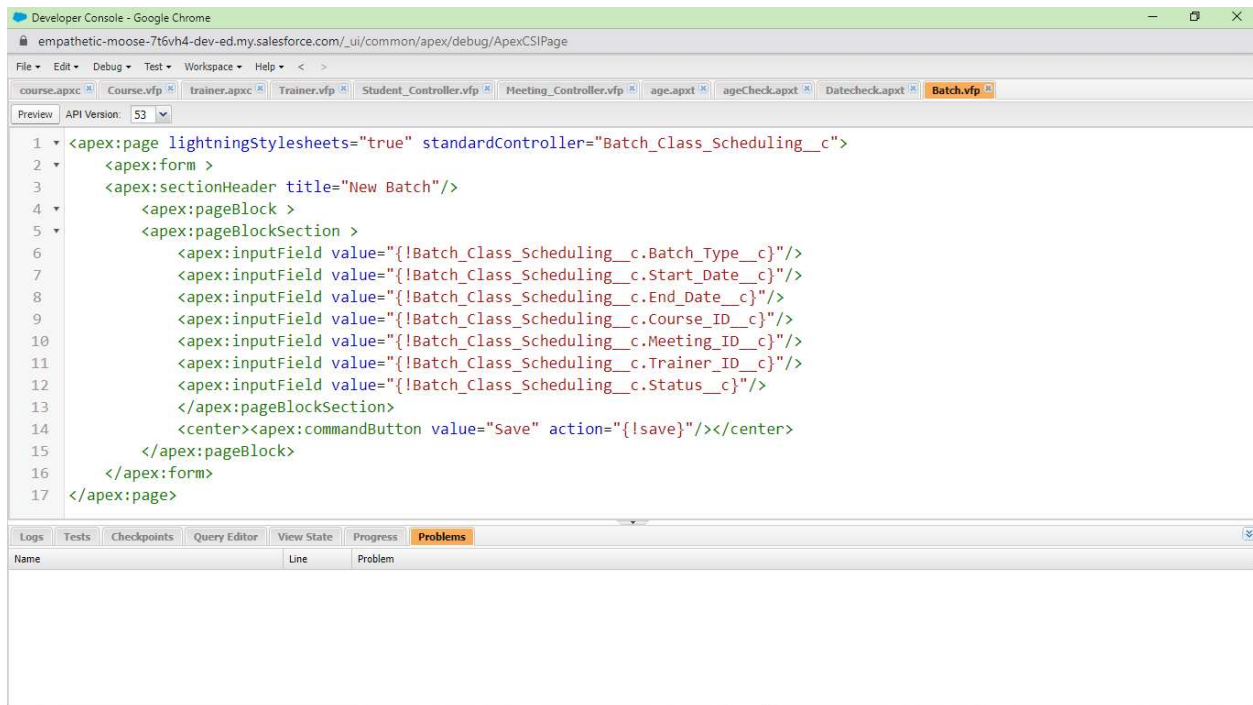
Upload the Screenshot of Milestone / Activity: (Create New Meeting Using Standard Controller)



```
1 <apex:page lightningStyleSheets="true" standardController="Meeting__c">
2   <apex:form >
3     <apex:sectionHeader title="New Meeting"/>
4     <apex:PageBlock >
5       <apex:PageBlockSection >
6         <apex:inputField value="{!Meeting__c.Valid_From__c}"/>
7         <apex:inputField value="{!Meeting__c.Valid_To__c}"/>
8         <apex:inputField value="{!Meeting__c.Meeting_Link__c}"/>
9         <apex:inputField value="{!Meeting__c.Active__c}"/>
10      </apex:PageBlockSection>
11      <center><apex:commandButton value="Save" action="{!save}"/></center>
12    </apex:PageBlock>
13  </apex:form>
14</apex:page>
```

From developer console create a new VisualForce page and paste the Batch Visualforce code in the visualforce page.

Upload the Screenshot of Milestone / Activity: (Create New Batch Using Standard Controller)



```
1 <apex:page lightningStylesheets="true" standardController="Batch_Class_Scheduling__c">
2   <apex:form >
3     <apex:sectionHeader title="New Batch"/>
4     <apex:pageBlock >
5       <apex:pageBlockSection >
6         <apex:inputField value="{!Batch_Class_Scheduling__c.Batch_Type__c}"/>
7         <apex:inputField value="{!Batch_Class_Scheduling__c.Start_Date__c}"/>
8         <apex:inputField value="{!Batch_Class_Scheduling__c.End_Date__c}"/>
9         <apex:inputField value="{!Batch_Class_Scheduling__c.Course_ID__c}"/>
10        <apex:inputField value="{!Batch_Class_Scheduling__c.Meeting_ID__c}"/>
11        <apex:inputField value="{!Batch_Class_Scheduling__c.Trainer_ID__c}"/>
12        <apex:inputField value="{!Batch_Class_Scheduling__c.Status__c}"/>
13      </apex:pageBlockSection>
14      <center><apex:commandButton value="Save" action="{!save}"/></center>
15    </apex:pageBlock>
16  </apex:form>
17</apex:page>
```

Day 9: 12/01/2022, Wednesday

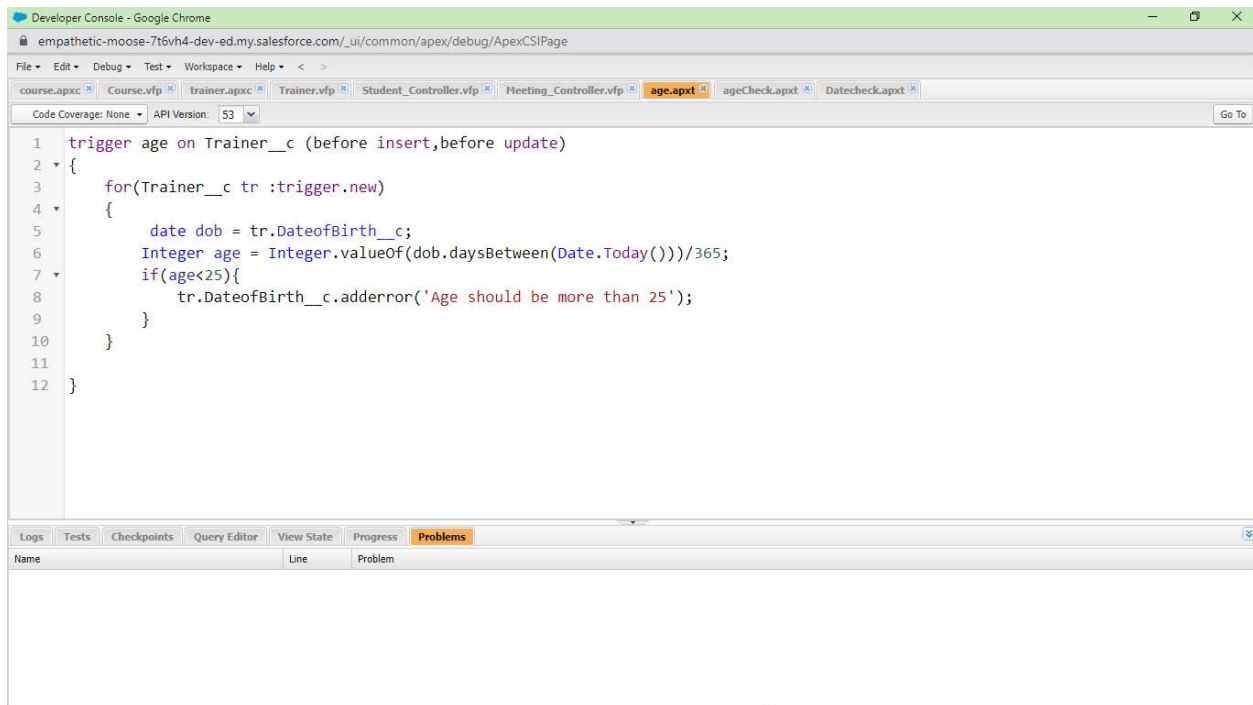
Topic: Triggers

Milestone / Activities: Trigger On Trainer Object
Trigger On Student Object
Trigger On Meeting Object

Detailed Description:

Create New Apex Triggers in Developer Console of Trainer, Student and Meeting. Paste the Trigger of Trainer, Student and Meeting code in the files of Trigger.

Upload the Screenshot of Milestone / Activity: (Trigger On Trainer Object)

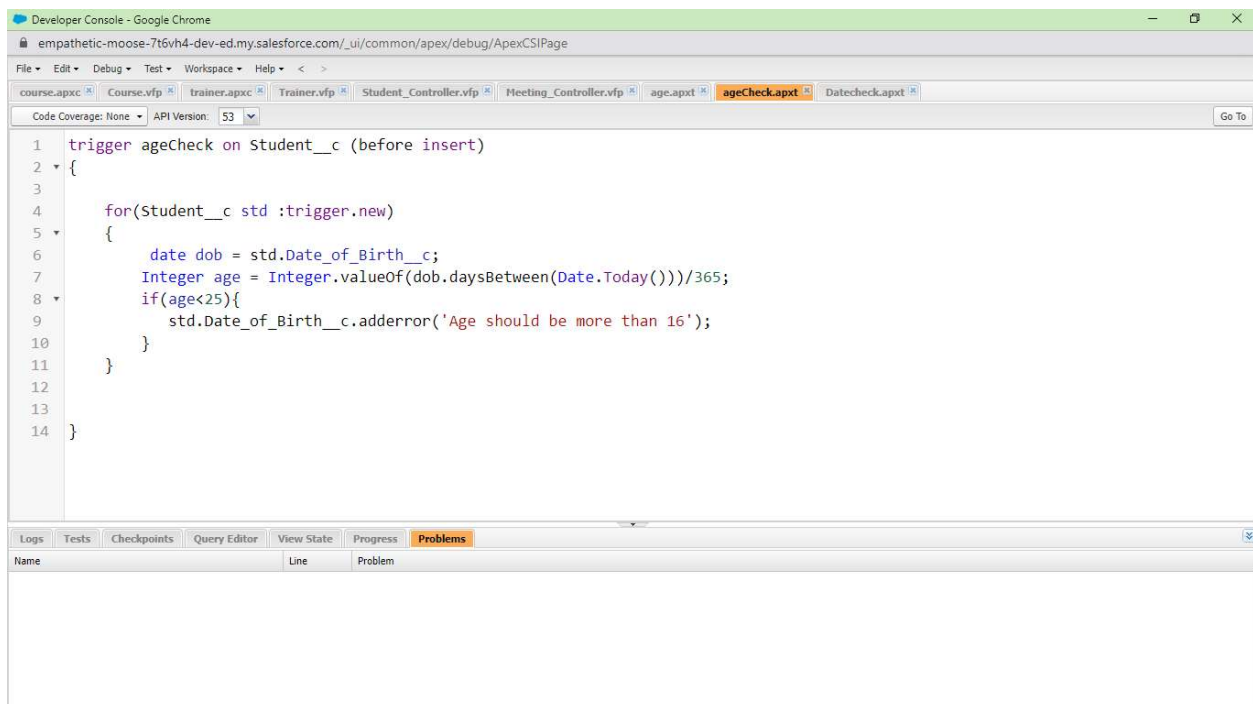


The screenshot shows the Salesforce Developer Console with the Apex trigger code for the Trainer__c object. The trigger is named 'trigger age on Trainer__c (before insert,before update)' and is located at line 1. The code is as follows:

```
1 trigger age on Trainer__c (before insert,before update)
2 {
3     for(Trainer__c tr :trigger.new)
4     {
5         date dob = tr.DateofBirth__c;
6         Integer age = Integer.valueOf(dob.daysBetween(Date.Today()))/365;
7         if(age<25){
8             tr.DateofBirth__c.adderror('Age should be more than 25');
9         }
10    }
11 }
12 }
```

The bottom of the console shows the 'Problems' tab, which is currently empty.

Upload the Screenshot of Milestone / Activity: (Trigger On Student Object)



The screenshot shows the Salesforce Developer Console with the Apex trigger code for the Student__c object. The trigger is named 'trigger ageCheck on Student__c (before insert)' and is located at line 1. The code is as follows:

```
1 trigger ageCheck on Student__c (before insert)
2 {
3     for(Student__c std :trigger.new)
4     {
5         date dob = std.Date_of_Birth__c;
6         Integer age = Integer.valueOf(dob.daysBetween(Date.Today()))/365;
7         if(age<25){
8             std.Date_of_Birth__c.adderror('Age should be more than 16');
9         }
10    }
11 }
12 }
13 }
14 }
```

The bottom of the console shows the 'Problems' tab, which is currently empty.

Upload the Screenshot of Milestone / Activity: (Trigger On Meeting Object)

