

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

Day 1: 03/01/2022, Monday

Topic: Create your salesforce developer org to get started

Milestone / Activities: Creating Developer Account

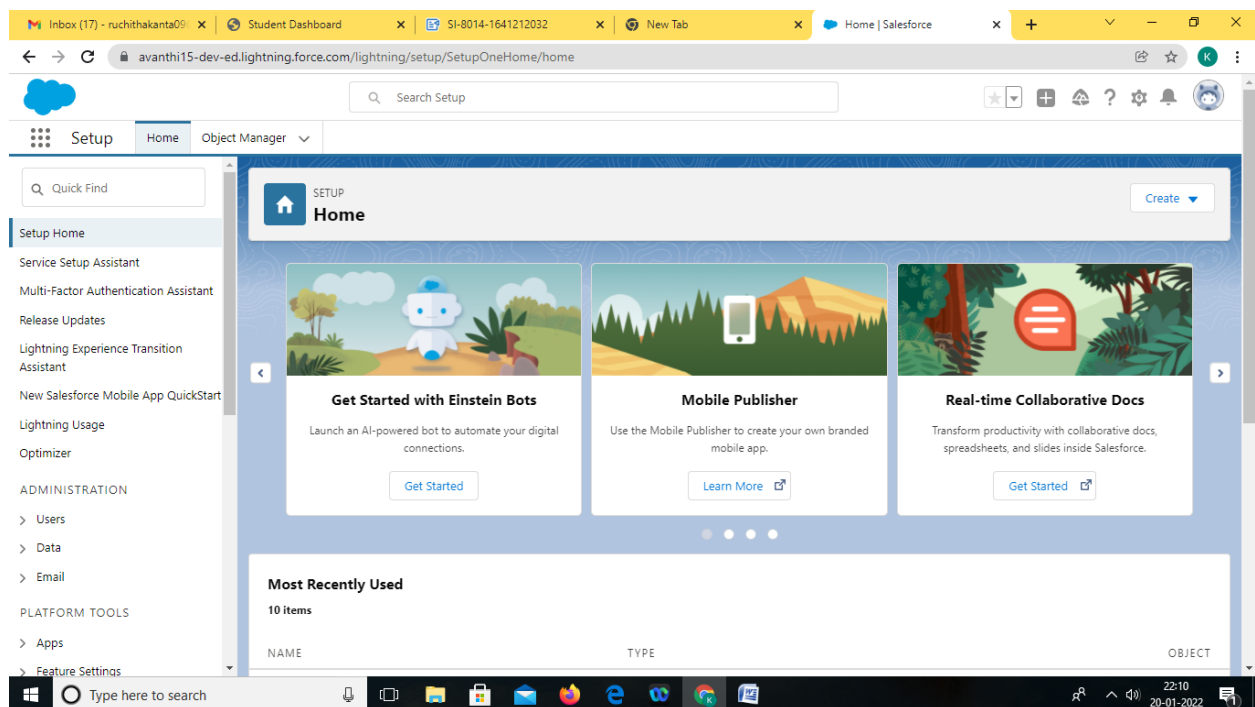
Account Activation

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

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

Milestone / Activities: Create The Custom Objects For The Application

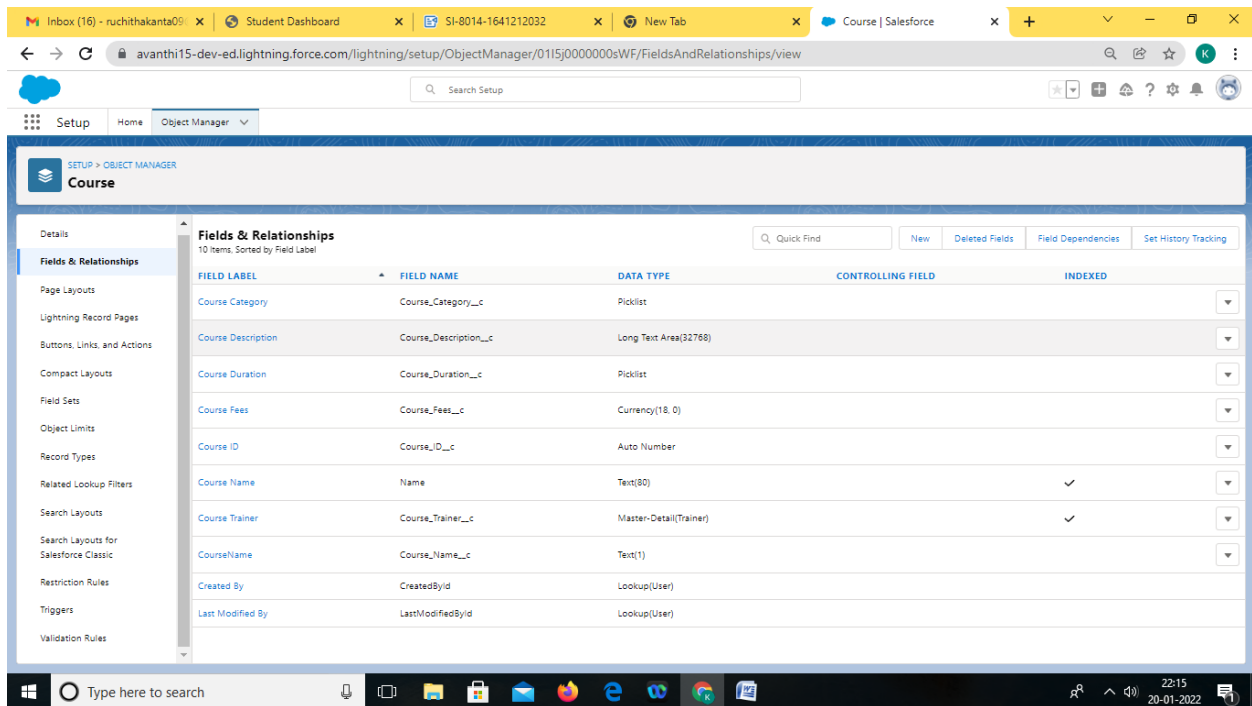
Create Custom Fields On Course Object
Create Custom Fields On Trainer Object
Create Custom Fields On Student Object
Create Custom Fields On Meeting Object
Create Custom Fields On Batch Object

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)

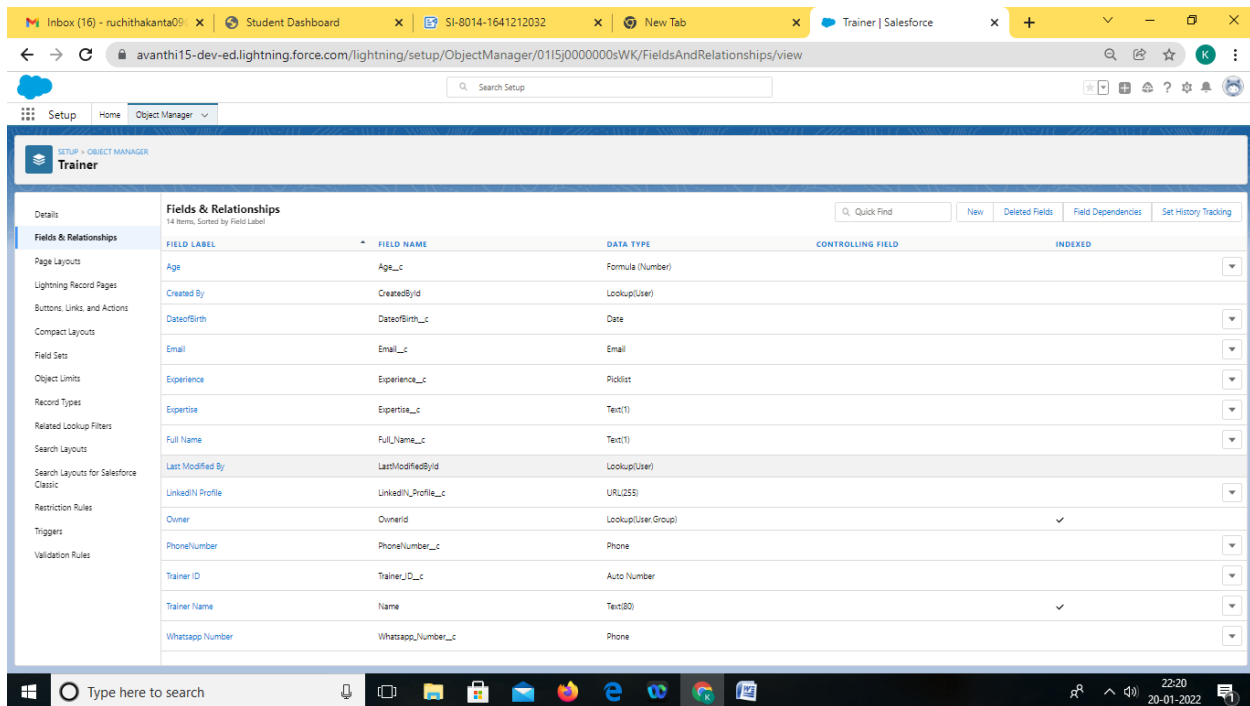


FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Course Category	Course_Category__c	Picklist		
Course Description	Course_Description__c	Long Text Area(32768)		
Course Duration	Course_Duration__c	Picklist		
Course Fees	Course_Fees__c	Currency(18, 0)		
Course ID	Course_ID__c	Auto Number		
Course Name	Name	Text(80)		✓
Course Trainer	Course_Trainer__c	Master-Detail(Trainer)		✓
CourseName	Course_Name__c	Text(1)		
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)

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

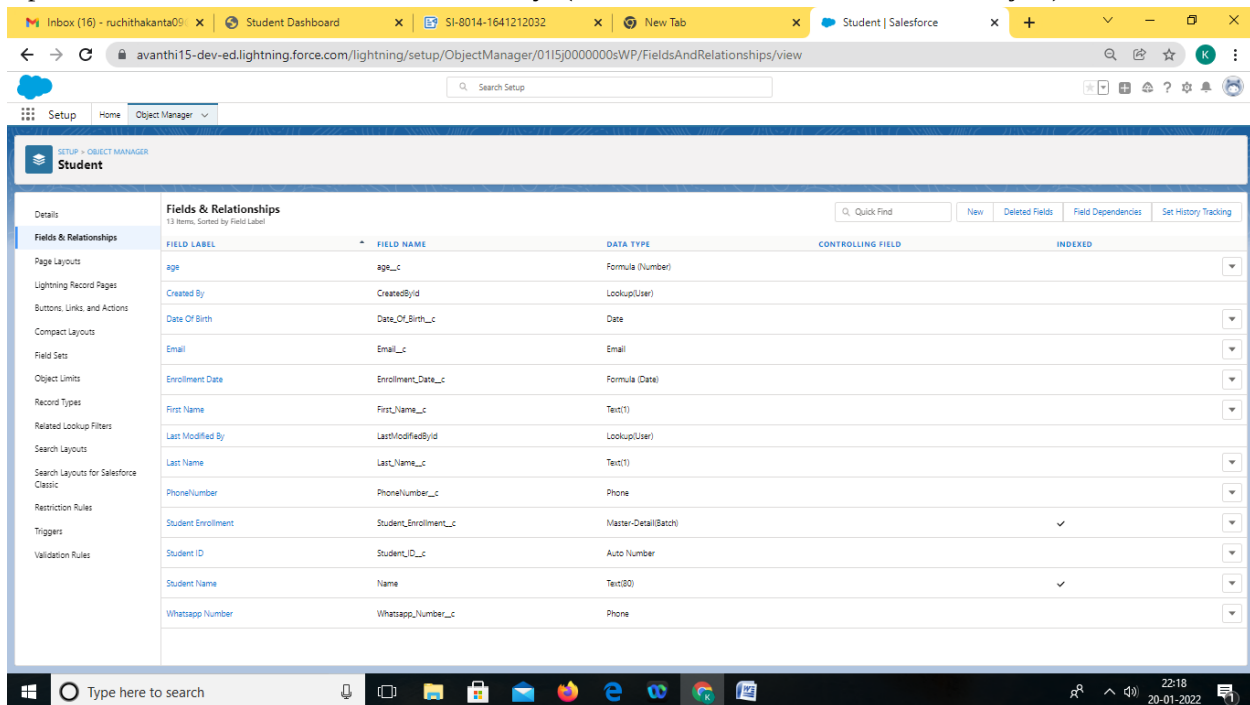


The screenshot shows the Salesforce Object Manager interface for the 'Trainer' object. The 'Fields & Relationships' tab is selected, displaying a list of 14 fields. The fields are sorted by Field Label. The table below lists the fields and their properties:

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(1)		
Full Name	FullName__c	Text(1)		
Last Modified By	LastModifiedById	Lookup(User)		
LinkedIn Profile	LinkedInProfile__c	URL(255)		
Owner	OwnerId	Lookup(User/Group)		✓
Phone Number	PhoneNumber__c	Phone		
Trainer ID	TrainerID__c	Auto Number		
Trainer Name	Name	Text(80)		✓
Whatsapp Number	WhatsappNumber__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 interface for the 'Student' object. The 'Fields & Relationships' tab is selected, displaying a list of 13 fields. The fields are sorted by Field Label. The table below lists the fields and their properties:

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		
Enrollment Date	EnrollmentDate__c	Formula (Date)		
First Name	FirstName__c	Text(1)		
Last Modified By	LastModifiedById	Lookup(User)		
Last Name	LastName__c	Text(1)		
Phone Number	PhoneNumber__c	Phone		
Student Enrollment	StudentEnrollment__c	Master-Detail(Batch)		✓
Student ID	StudentID__c	Auto Number		
Student Name	Name	Text(80)		✓
Whatsapp Number	WhatsappNumber__c	Phone		

The following are the fields we are created on Meeting Object that are Meeting ID, Valid From,

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

Valid

To,Active and Meeting link.

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

The screenshot shows the Salesforce Setup interface for the 'Meeting' object. The 'Fields & Relationships' section is active, displaying a list of fields. The table below represents the data shown in the screenshot:

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	Meeting_ID__c	Auto Number		
Meeting Link	Meeting_Link__c	URL(255)		
Meeting Name	Name	Text(80)		✓
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)

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

The screenshot shows the Salesforce Object Manager interface for the 'Batch' object. The 'Fields & Relationships' tab is selected, displaying a list of 14 fields. The fields are sorted by Field Label. The table includes columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed. The fields listed are: Batch ID (Auto Number), Batch Type (Picklist), Batch Class Scheduling Name (Text(20)), Course ID (Lookup(Course)), Created By (Lookup(User)), Details (Formula (Text)), End Date (Date), Last Modified By (Lookup(User)), Meeting Id (Lookup(Meeting)), Number Of Students Enrolled (Roll-Up Summary (COUNT Student)), Owner (Lookup(User Group)), Start Date (Date), Status (Picklist), and Trainer Id (Lookup(Trainer)).

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Batch ID	BatchId__c	Auto Number		
Batch Type	BatchType__c	Picklist		
Batch Class Scheduling Name	Name	Text(20)		✓
Course ID	CourseId__c	Lookup(Course)		✓
Created By	CreatedById	Lookup(User)		
Details	Details__c	Formula (Text)		
End Date	EndDate__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
Meeting Id	MeetingId__c	Lookup(Meeting)		✓
Number Of Students Enrolled	NumberOfStudentsEnrolled__c	Roll-Up Summary (COUNT Student)		
Owner	OwnerId	Lookup(User Group)		✓
Start Date	StartDate__c	Date		
Status	Status__c	Picklist		
Trainer Id	TrainerId__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

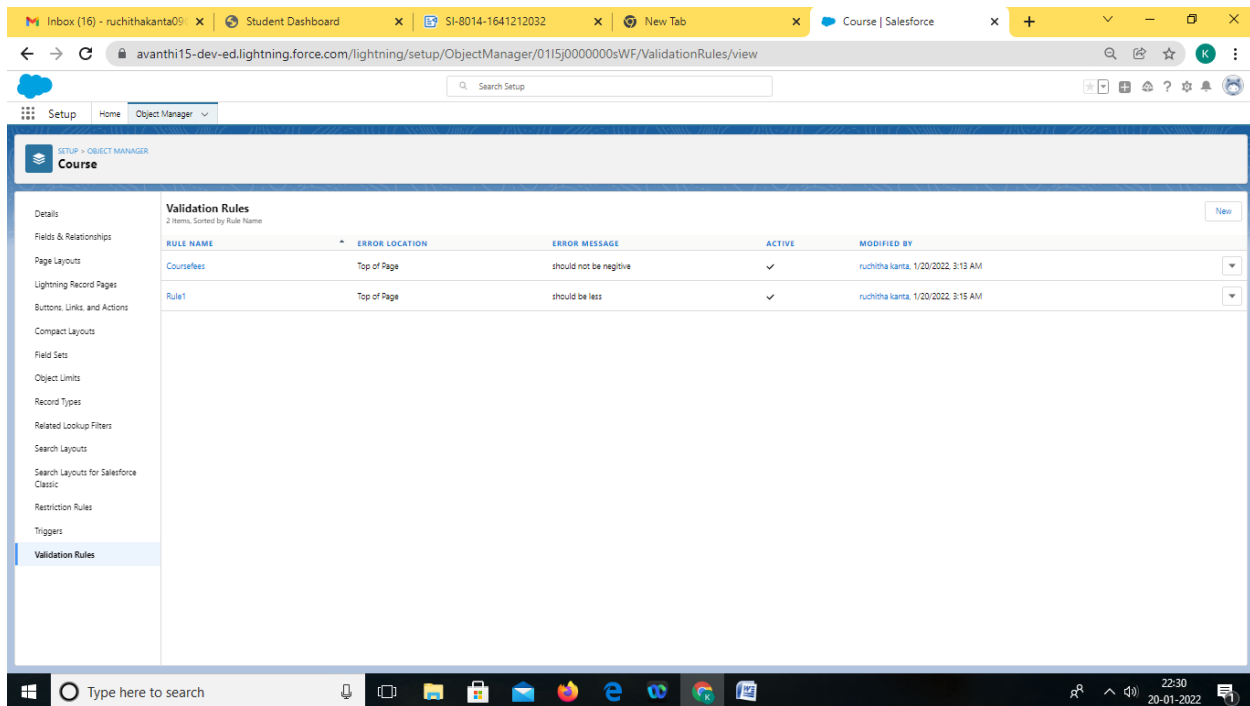
Create Validation Rules on Meeting Object

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)

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

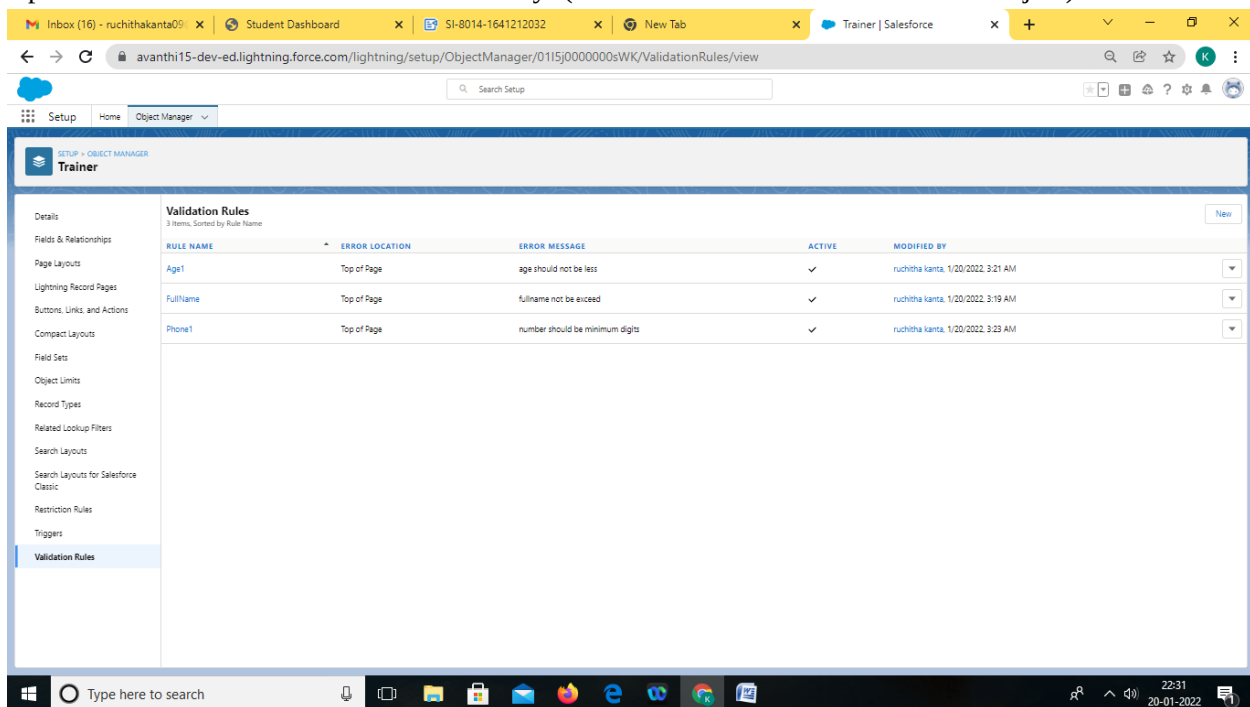


The screenshot shows the Salesforce Setup interface for the 'Course' object. The left sidebar lists various setup options, with 'Validation Rules' selected. The main content area displays a table of validation rules for the 'Course' object. The table has columns for Rule Name, Error Location, Error Message, Active status, and Modified By. Two rules are listed: 'Courseless' and 'Rule1'.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Courseless	Top of Page	should not be negative	✓	ruchitha kanta, 1/20/2022, 3:13 AM
Rule1	Top of Page	should be less	✓	ruchitha kanta, 1/20/2022, 3:15 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: 'Age1', 'FullName', and 'Phone1'.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Age1	Top of Page	age should not be less	✓	ruchitha kanta, 1/20/2022, 3:21 AM
FullName	Top of Page	fullname not be exceed	✓	ruchitha kanta, 1/20/2022, 3:19 AM
Phone1	Top of Page	number should be minimum digits	✓	ruchitha kanta, 1/20/2022, 3:23 AM

Create a validation rules on Student Object. The validation rules are Last Name should not be

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

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 Setup interface for the 'Student' object. The 'Validation Rules' section is active, displaying a table with 3 items. The table columns are: RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. The rules listed are: Age2 (should be minimum 16 years of age), LastName1 (lastname should not be blank), and WhatsappNumber (should be minimum 10 digits). All rules are active and were modified by ruchitha.kanta on 1/20/2022.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Age2	Top of Page	should be minimum 16 years of age	✓	ruchitha.kanta, 1/20/2022, 3:33 AM
LastName1	Top of Page	lastname should not be blank	✓	ruchitha.kanta, 1/20/2022, 3:26 AM
WhatsappNumber	Top of Page	should be minimum 10 digits	✓	ruchitha.kanta, 1/20/2022, 3:36 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)

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

The screenshot shows the Salesforce Setup interface. The left sidebar contains a navigation menu with options like 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, Restriction Rules, Triggers, and Validation Rules. The main content area is titled 'Validation Rules' and shows a table with 3 items, sorted by Rule Name. The table has columns for Rule Name, Error Location, Error Message, Active status, and Modified By. The rules listed are ObjectName, Rule3, and Rule5, all of which are active and were modified by ruchitha.kanta on 1/20/2022 at 3:41 AM, 3:38 AM, and 3:38 AM respectively.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
ObjectName	Top of Page	should be in between valid from and valid to	✓	ruchitha.kanta, 1/20/2022, 3:41 AM
Rule3	Top of Page	should not be current date	✓	ruchitha.kanta, 1/20/2022, 3:38 AM
Rule5	Top of Page	should not be less than valid from	✓	ruchitha.kanta, 1/20/2022, 3:38 AM

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)

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

This screenshot shows the Salesforce Setup page for Workflow Rules. The left sidebar contains navigation links for Setup, Home, Object Manager, and a search bar. The main content area is titled 'Workflow Rules' and shows details for a rule named 'Name3'. The rule is active and has a description 'Valid To: - TODAY () = 1'. The 'Workflow Actions' section shows an immediate action of type 'Email Alert' with the description 'Owner recipient type'. A warning message states: 'You cannot add new time triggers to an active rule. Deactivate This Rule'. The bottom of the page shows the Windows taskbar with various application icons and the system clock displaying 22:38 on 20-01-2022.

Workflow Rule Name3

Workflow Rule Detail

Rule Name	Active	Object	Meeting
Name3	<input checked="" type="checkbox"/>	Meeting	

Rule Criteria: Valid To: - TODAY () = 1

Created By: ruchithakanta 1/20/2022, 3:51 AM

Modified By: ruchithakanta 1/20/2022, 9:07 AM

Workflow Actions

Immediate Workflow Actions

Type	Description
Email Alert	Owner recipient type

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

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

This screenshot shows the Salesforce Setup page for Workflow Rules, similar to the first one but for a rule named 'Name6'. The rule is active and has a description 'Valid To: - TODAY () = 2'. The 'Workflow Actions' section shows an immediate action of type 'Email Alert' with the description 'Owner recipient type'. A warning message states: 'You cannot add new time triggers to an active rule. Deactivate This Rule'. The bottom of the page shows the Windows taskbar with various application icons and the system clock displaying 22:38 on 20-01-2022.

Workflow Rule Name6

Workflow Rule Detail

Rule Name	Active	Object	Meeting
Name6	<input checked="" type="checkbox"/>	Meeting	

Rule Criteria: Valid To: - TODAY () = 2

Created By: ruchithakanta 1/20/2022, 3:58 AM

Modified By: ruchithakanta 1/20/2022, 9:08 AM

Workflow Actions

Immediate Workflow Actions

Type	Description
Email Alert	Owner recipient type

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: (Create New Meeting Using Standard Controller)

The screenshot shows the Salesforce Setup page for Workflow Rules. The left sidebar contains a navigation menu with options like Process Automation, Workflow Actions, Email Alerts, Field Updates, Outbound Messages, Send Actions, Tasks, Workflow Rules, Environments, Monitoring, and Time-Based Workflow. The main content area displays the details for a specific Workflow Rule named 'Rule1'. The 'Workflow Rule Detail' section shows the rule name, status (Active), description, and evaluation criteria. The 'Workflow Actions' section lists immediate and time-dependent actions. A warning message states: 'You cannot add new time triggers to an active rule. Consider This Rule'.

Workflow Rule Detail

Field	Value
Rule Name	Rule1
Status	Active
Description	
Rule Criteria	TODAY() < Valid_To__c + 1
Created By	ruchitha.kanta
Created Date	1/20/2022, 3:45 AM
Modified By	ruchitha.kanta
Modified Date	1/20/2022, 9:09 AM

Workflow Actions

Type	Description
Field Update	

Time-Dependent Workflow Actions

You cannot add new time triggers to an active rule. Consider This Rule

The screenshot shows the Salesforce Setup page for Workflow Rules, displaying a list of all workflow rules. The left sidebar is the same as the previous screenshot. The main content area is titled 'All Workflow Rules' and provides instructions on how to configure workflow rules. A table lists the existing workflow rules, including their names, descriptions, objects, and active status. A 'Quick Tip' box on the right provides additional guidance.

All Workflow Rules

Configure your organization's workflow by creating workflow rules. Each workflow rule consists of:

- Criteria that causes 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.

View: All Workflow Rules | Create New Rule

Action	Rule Name	Description	Object	Active
Edit Deactivate	Rule1		Meeting	✓
Edit Deactivate	Rule2		Meeting	✓
Edit Deactivate	Rule3		Meeting	✓

Quick Tip

- Useful Sample Workflow Rules
- Video Tutorial: Creating a Rule
- Understanding Workflow Rules

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

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

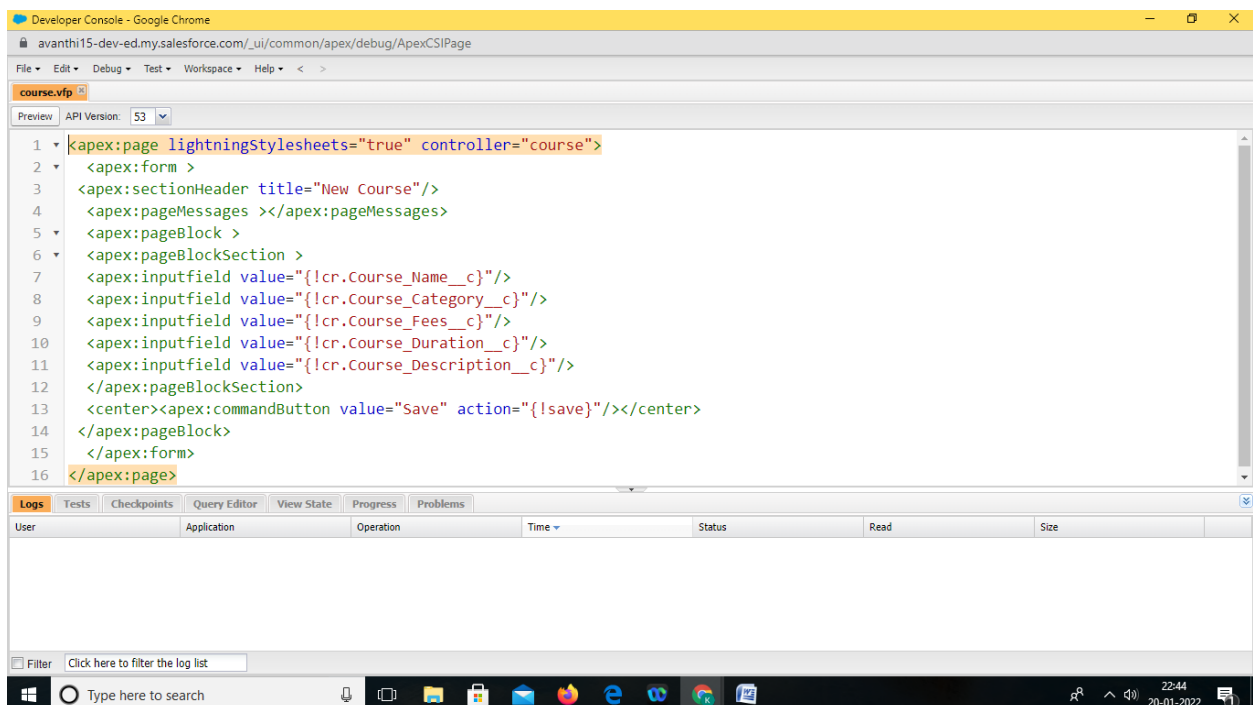
Create New Meeting Using Standard Controller

Create New Batch Using Standard Controller

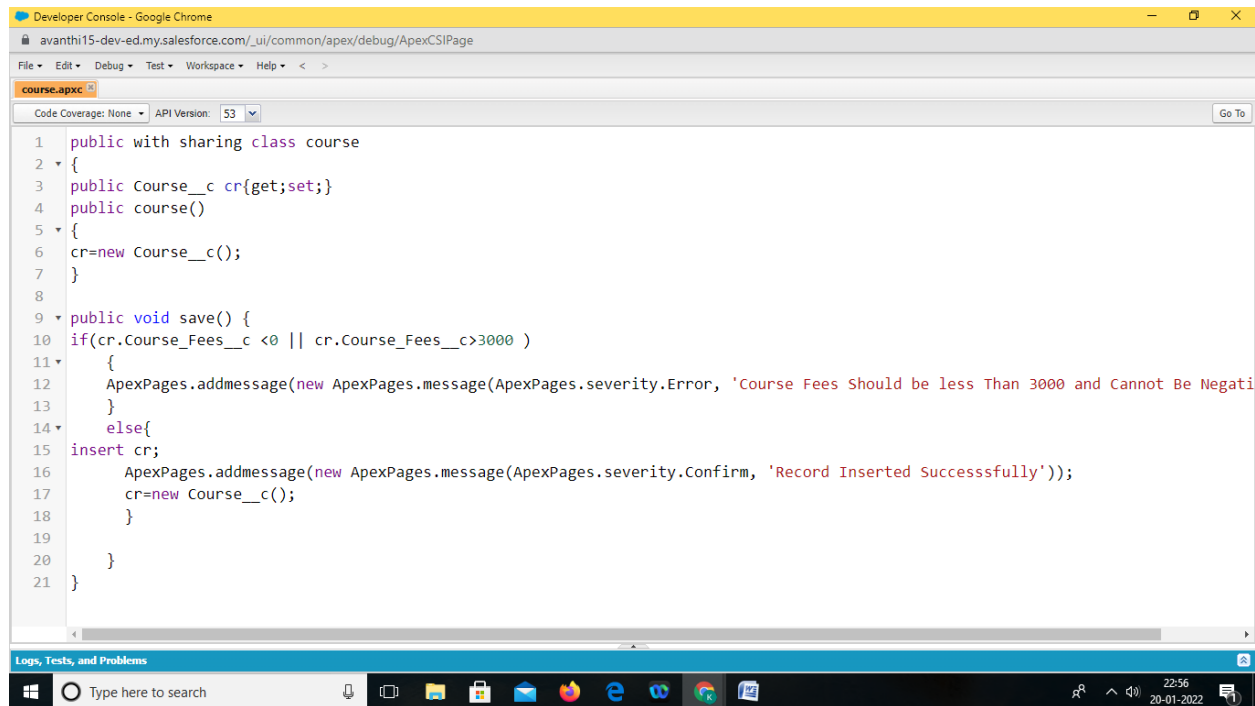
Detailed Description:

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)



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

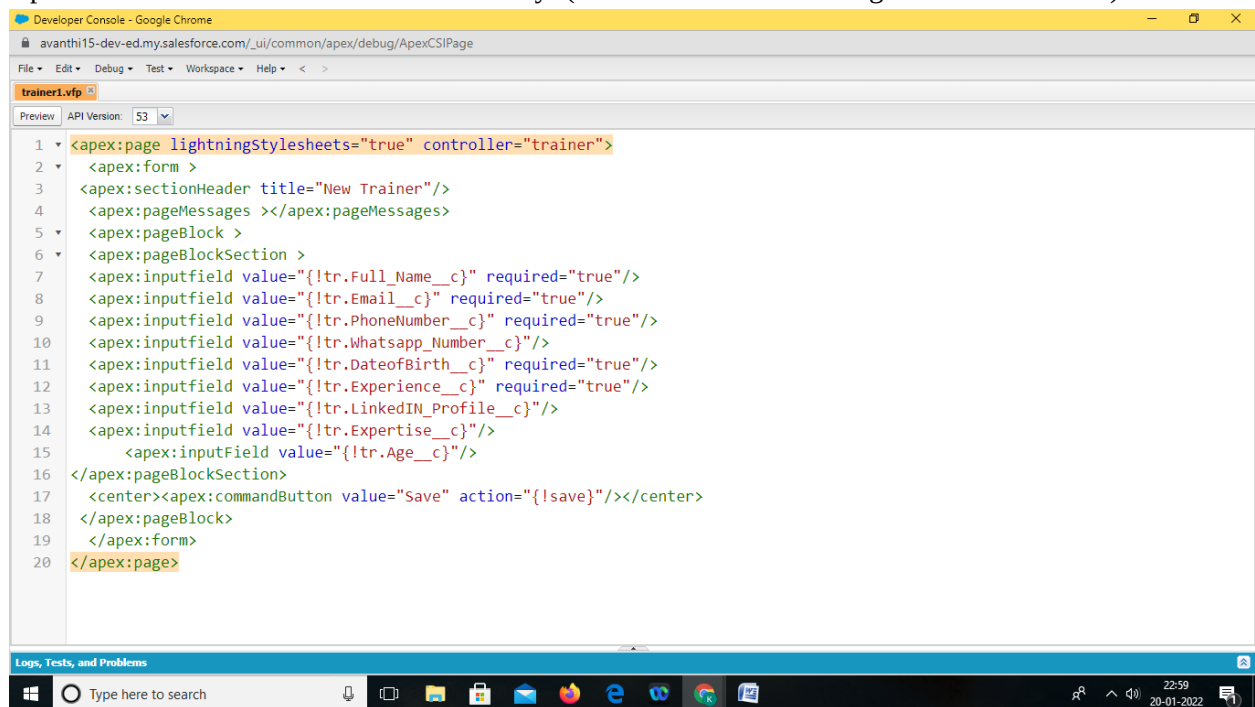


The screenshot shows the Salesforce Developer Console with the 'course.apxc' file open. The code defines a 'course' class with a 'save()' method that checks if the 'Course_Fees__c' is less than 3000 and not negative. If the condition is met, it inserts the record and displays a success message. Otherwise, it displays an error message.

```
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 Fees Should be less Than 3000 and Cannot Be Negati
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    }
21 }
```

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.

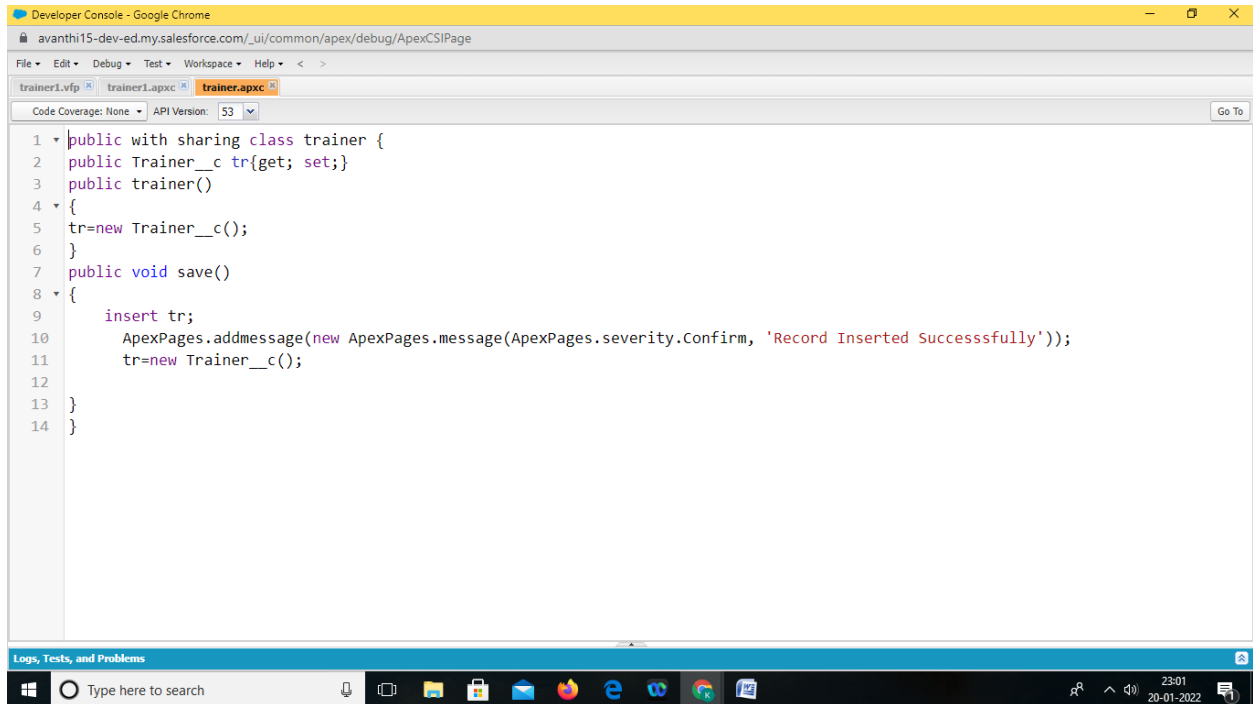
Upload the Screenshot of Milestone / Activity: (Create New Trainer Using Custom Controller)



The screenshot shows the Salesforce Developer Console with the 'trainer1.vfp' file open. The code is a Visualforce page that uses the 'Trainer' custom controller. It contains a form with input fields for 'Full_Name__c', 'Email__c', 'PhoneNumber__c', 'Whatsapp_Number__c', 'DateofBirth__c', 'Experience__c', 'LinkedIN_Profile__c', 'Expertise__c', and 'Age__c'. All fields are required. A 'Save' button is at the bottom of the form.

```
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>
```

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



The screenshot shows the Salesforce Developer Console in Google Chrome. The browser address bar displays the URL: `avanthi15-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The console's file explorer shows three files: `trainer1.vfp`, `trainer1.apxc`, and `trainer.apxc`, with `trainer.apxc` selected. The code editor displays the following Apex code:

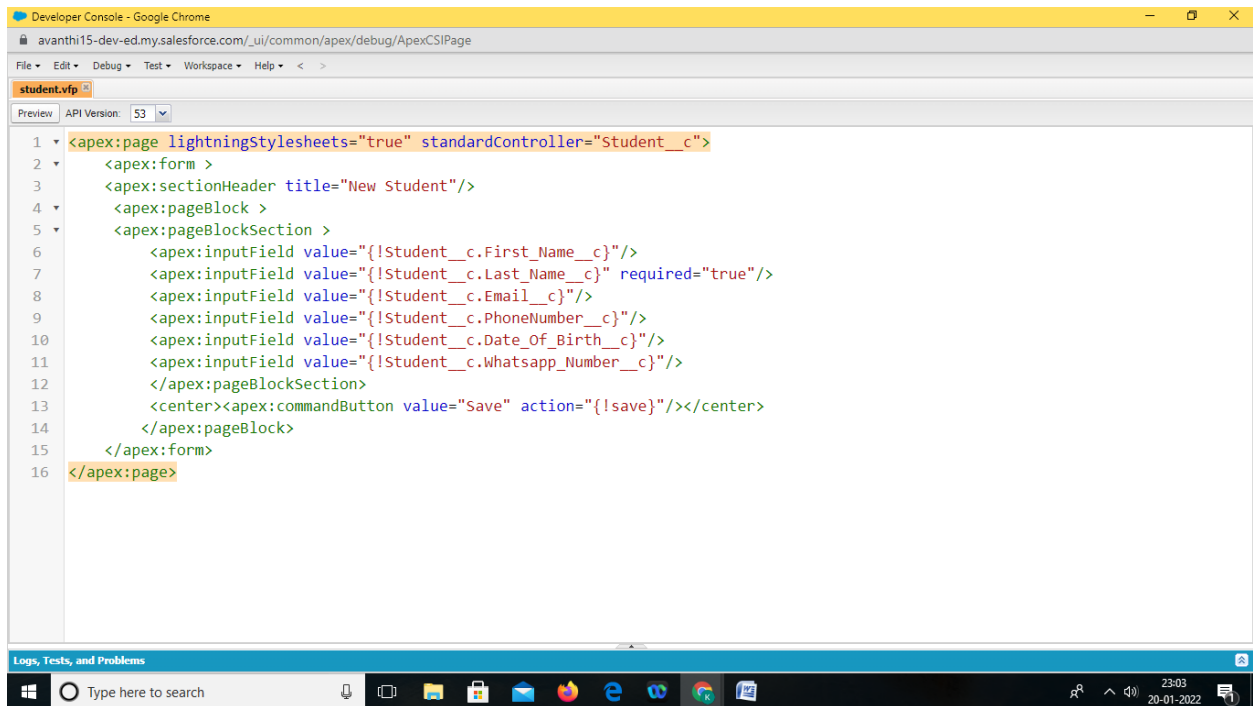
```
1 public with sharing class trainer {  
2     public Trainer__c tr{get; set;}  
3     public trainer()  
4 {  
5     tr=new Trainer__c();  
6 }  
7     public void save()  
8 {  
9         insert tr;  
10        ApexPages.addmessage(new ApexPages.message(ApexPages.severity.Confirm, 'Record Inserted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

The console's status bar at the bottom indicates 'Code Coverage: None' and 'API Version: 53'. The Windows taskbar is visible at the bottom of the screen, showing the search bar and various application icons.

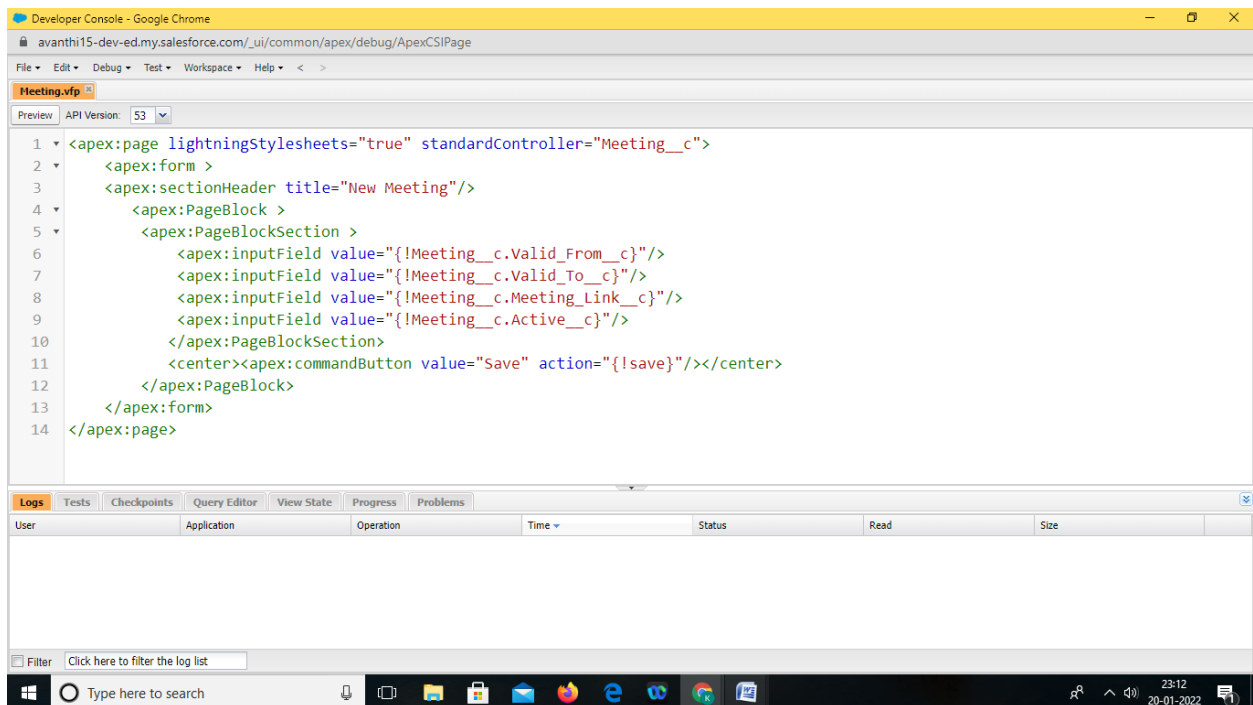
From developer console create a new VisualForce page and paste the Student Visualforce code in the visualforce page.

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

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



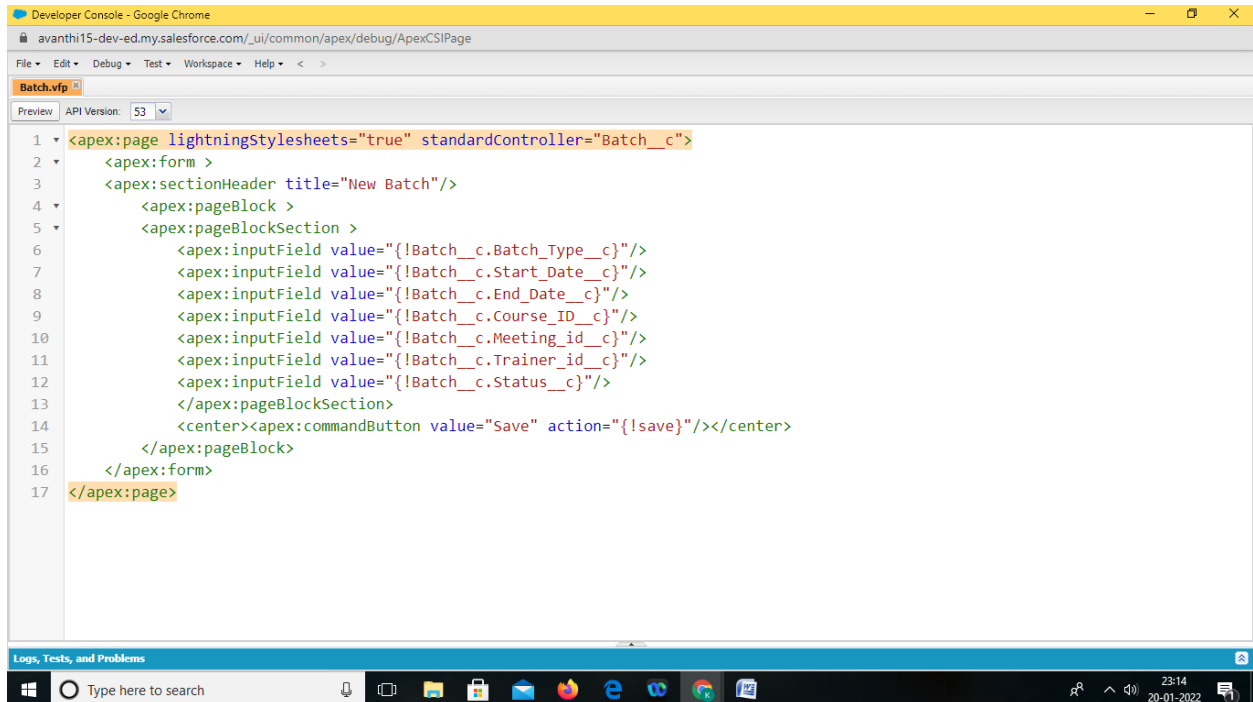
Upload the Screenshot of Milestone / Activity: (Create New Meeting Using Standard Controller)



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 Meeting Using Standard Controller)

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



The screenshot shows the Salesforce Developer Console with the Apex code for a new batch creation page. The code is as follows:

```
1 <apex:page lightningStylesheets="true" standardController="Batch_c">
2   <apex:form >
3     <apex:sectionHeader title="New Batch"/>
4     <apex:pageBlock >
5       <apex:pageBlockSection >
6         <apex:inputField value="{!Batch__c.Batch_Type__c}"/>
7         <apex:inputField value="{!Batch__c.Start_Date__c}"/>
8         <apex:inputField value="{!Batch__c.End_Date__c}"/>
9         <apex:inputField value="{!Batch__c.Course_ID__c}"/>
10        <apex:inputField value="{!Batch__c.Meeting_id__c}"/>
11        <apex:inputField value="{!Batch__c.Trainer_id__c}"/>
12        <apex:inputField value="{!Batch__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

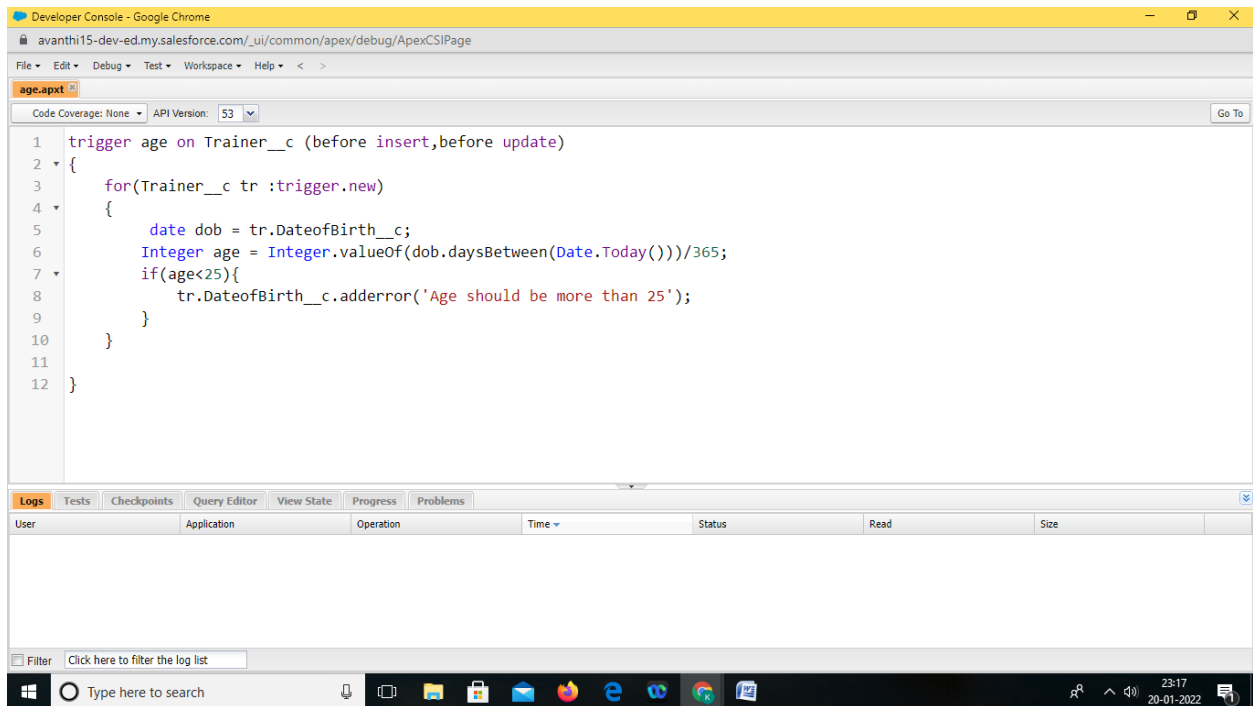
Trigger On Meeting Object

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)

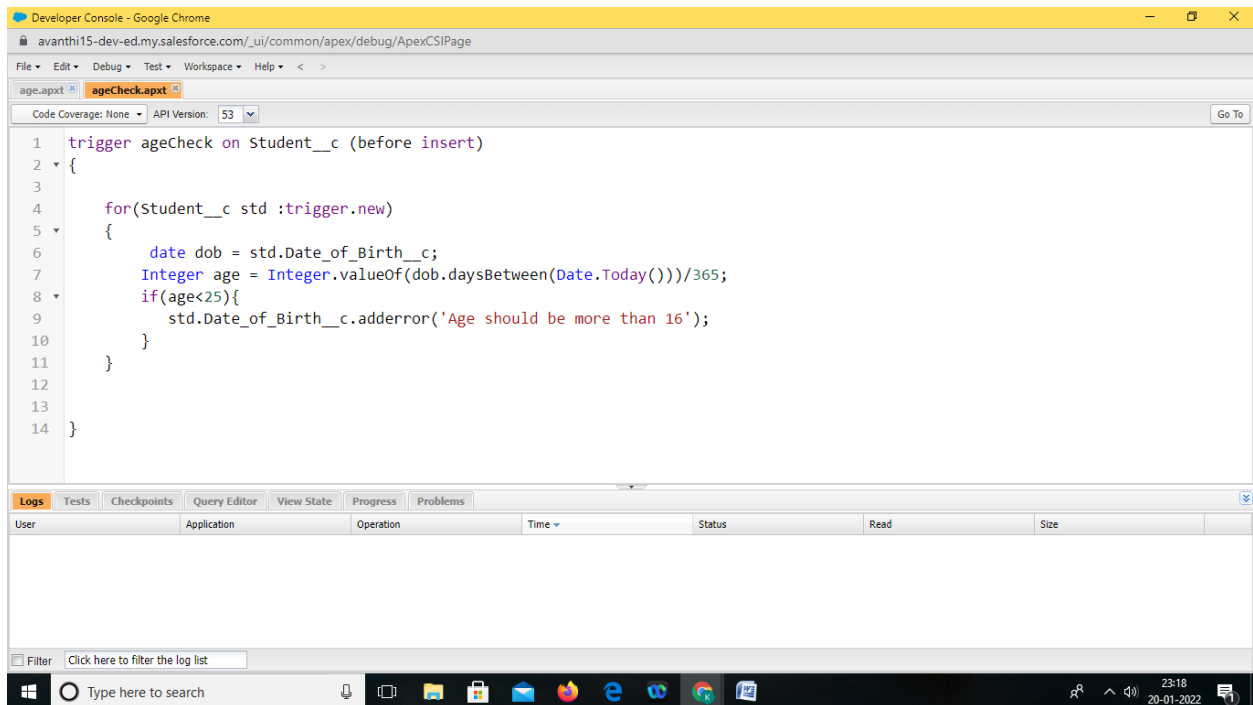
Upload the Screenshot of Milestone / Activity: (Create New Meeting Using Standard Controller)



```
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 }
```

User	Application	Operation	Time	Status	Read	Size
------	-------------	-----------	------	--------	------	------

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

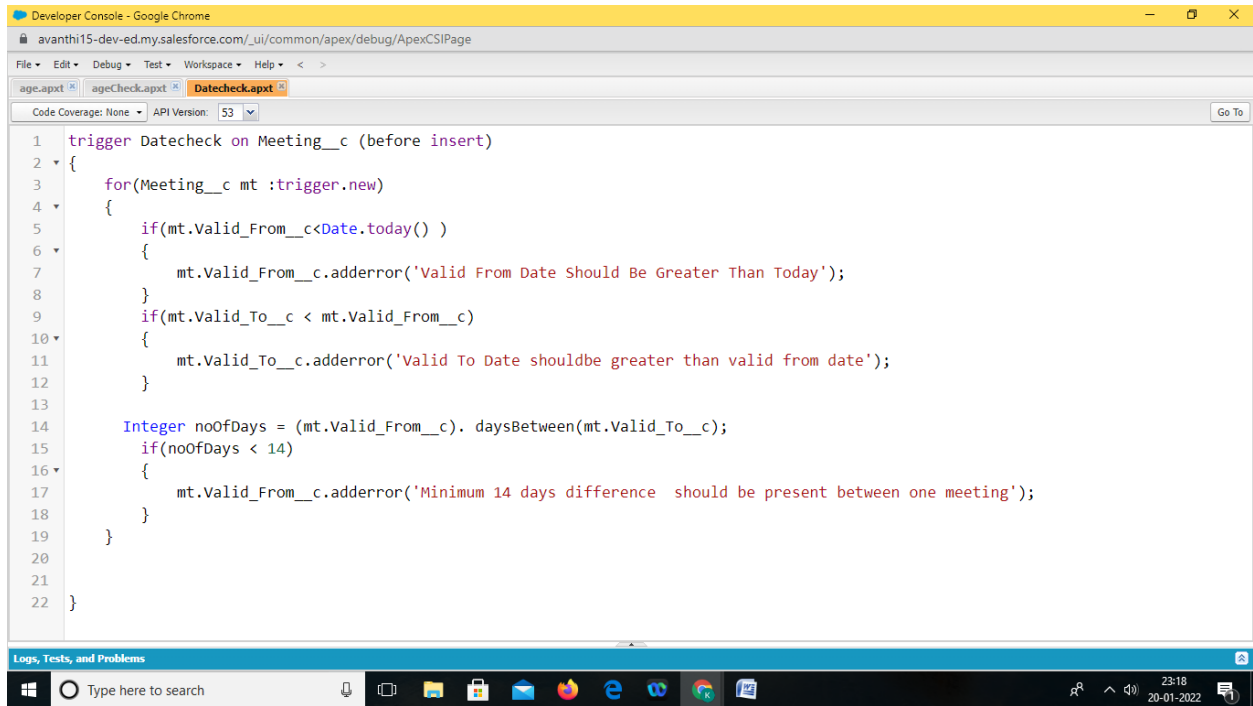


```
1 trigger ageCheck on Student__c (before insert)
2 {
3
4     for(Student__c std :trigger.new)
5     {
6         date dob = std.Date_of_Birth__c;
7         Integer age = Integer.valueOf(dob.daysBetween(Date.Today()))/365;
8         if(age<16){
9             std.Date_of_Birth__c.adderror('Age should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

User	Application	Operation	Time	Status	Read	Size
------	-------------	-----------	------	--------	------	------

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

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



```
1 trigger Datecheck on Meeting__c (before insert)
2 {
3     for(Meeting__c mt :trigger.new)
4     {
5         if(mt.Valid_From__c < Date.today() )
6         {
7             mt.Valid_From__c.adderror('Valid From Date Should Be Greater Than Today');
8         }
9         if(mt.Valid_To__c < mt.Valid_From__c)
10        {
11            mt.Valid_To__c.adderror('Valid To Date shouldbe greater than valid from date');
12        }
13
14        Integer noOfDays = (mt.Valid_From__c). daysBetween(mt.Valid_To__c);
15        if(noOfDays < 14)
16        {
17            mt.Valid_From__c.adderror('Minimum 14 days difference  should be present between one meeting');
18        }
19    }
20 }
21
22 }
```