

Smart Internz

SMART INTERNZ

SALESFORCE PROJECT READY - ONLINE TRAINING MANAGEMENT DEVELOPER.

BY AMIT YADAV.

INDEX:

SALESFORCE PROJECT READY - ONLINE TRAINING MANAGEMENT DEVELOPER.

> Create Your Salesforce Developer Org To Get Started.

>Custom Object Creation.

>Create Validation Rules.

>Process Automation.

>Creating Custom Pages Using Visualforce.

>Triggers.

Online Training Management - Developer

PROJECT INTRODUCTION:-

In this project we will be creating an online training management application that will store all the details of the training from courses offered, trainers, students enrolled, Meeting/training information, and many more. In the project, we will be creating objects to store these records and will also create various fields to store the values of these records. We will establish relationships between various objects and create custom pages using visualforce and also about the triggers.

> Create Your Salesforce Developer Org To Get Started:-

Creating a developer org in salesforce.

- 1.Go to developers.salesforce.com/
- 2.Click on sign up.
- 3.On the sign up form, enter the following details :
 - A.First name & Last name
 - B.Email
 - C.Role : Developer
 - D.Company : College Name
 - E.County : India
 - F.Postal Code : pin code
 - g.Username : should be a combination of your name and company.
This need not be an actual email id, you can give anything in the
format : [username@organization.com](#)

Login To Your Salesforce Account

1. Go to salesforce.com and click on login.
- 2.Enter the username and password that you just created.
- 3.After login this is the home page which you will see.

> Custom Object Creation:-

Creating The Custom Objects For The Application.

The required custom objects are:

- a)Courses.
- b)Trainer.
- c)Student.
- d)Meeting.
- e)Batch/Class Scheduling.

Course Object:-

We need to add these fields in the object.

Field Name	Data Type	Values
Course ID	Auto Number(Change the data type to AutoNumber While creating course object and changing record Type to course id)	Format(C-000001) Starting number=1
Course Name	Text	
Course Duration	Picklist	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)	

Trainer Object:-

We need to add these fields in the 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	

Student Object:-

We need to add these fields in the object.

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

Meeting Object:-

We need to add these fields in the object.

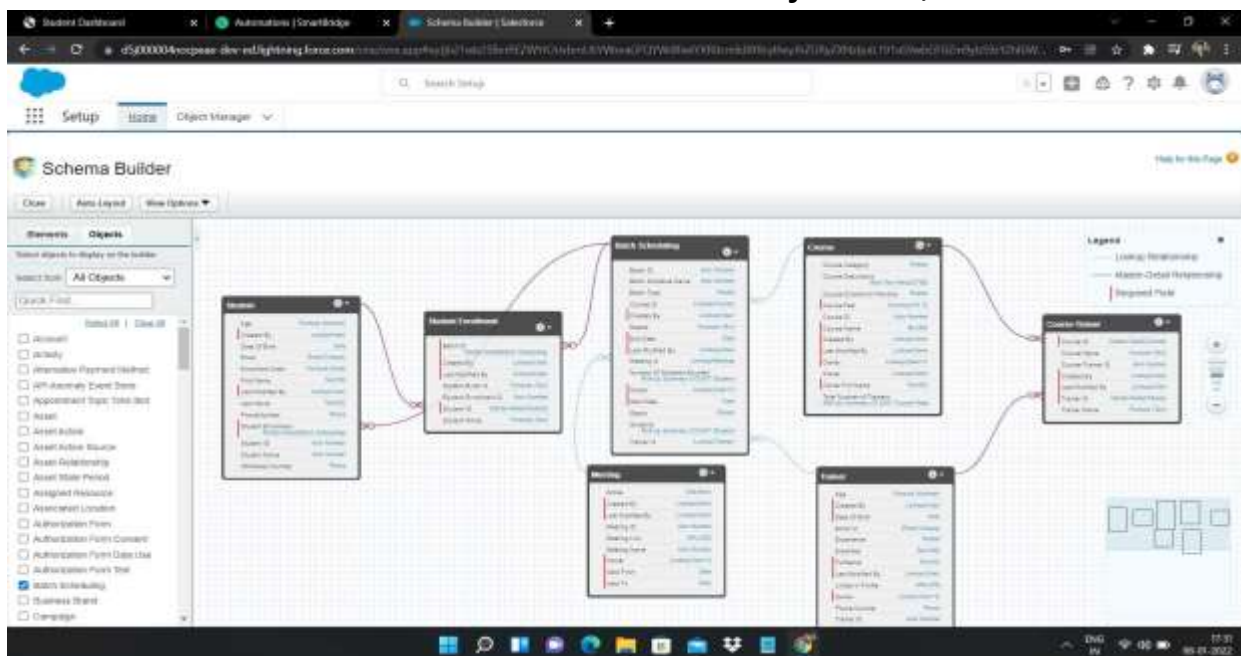
Field Name	Data Type	Values
------------	-----------	--------

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

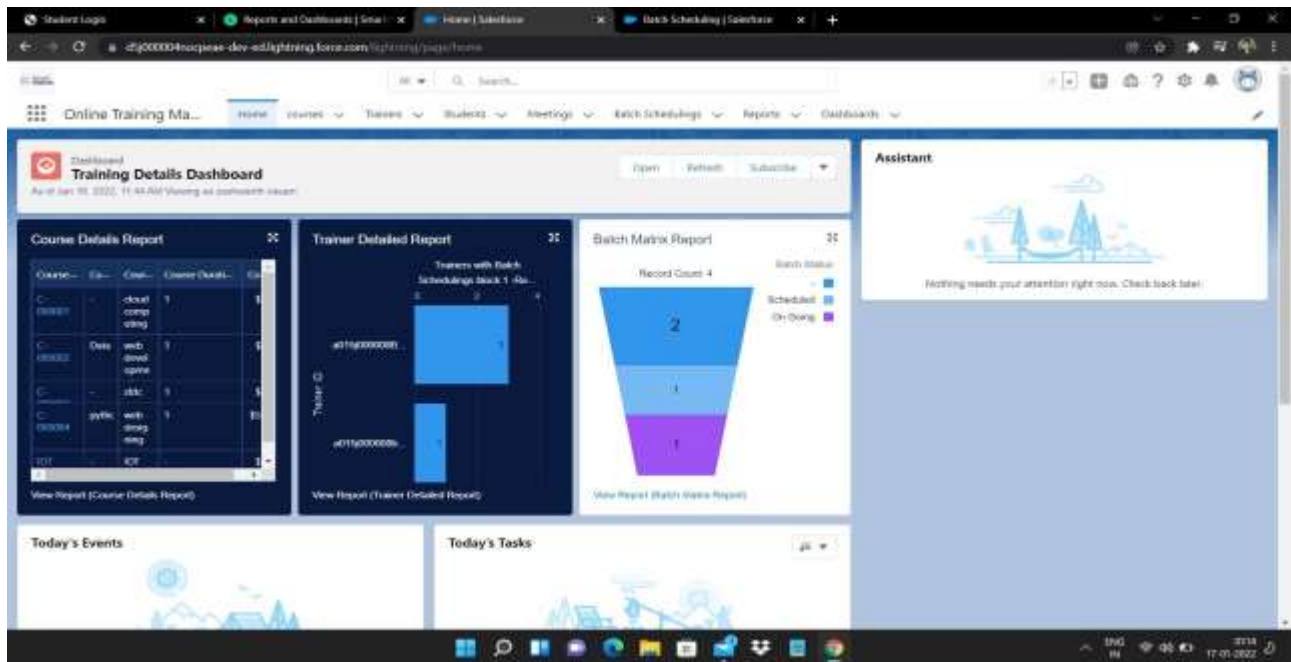
Batch Scheduling Object:-

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

The data model of the above Custom objects is,



The Application Home Tab looks like,



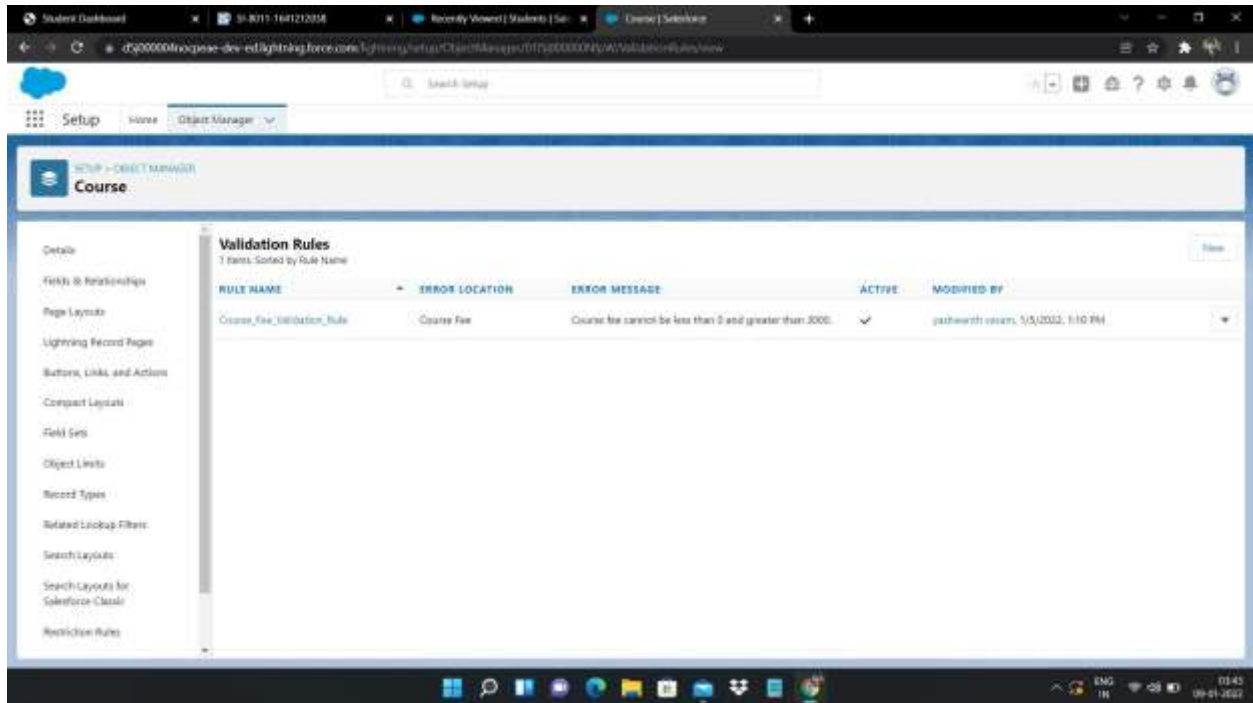
> Create Validation Rules:-

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

> For the Course object the validation rules we need to add is,

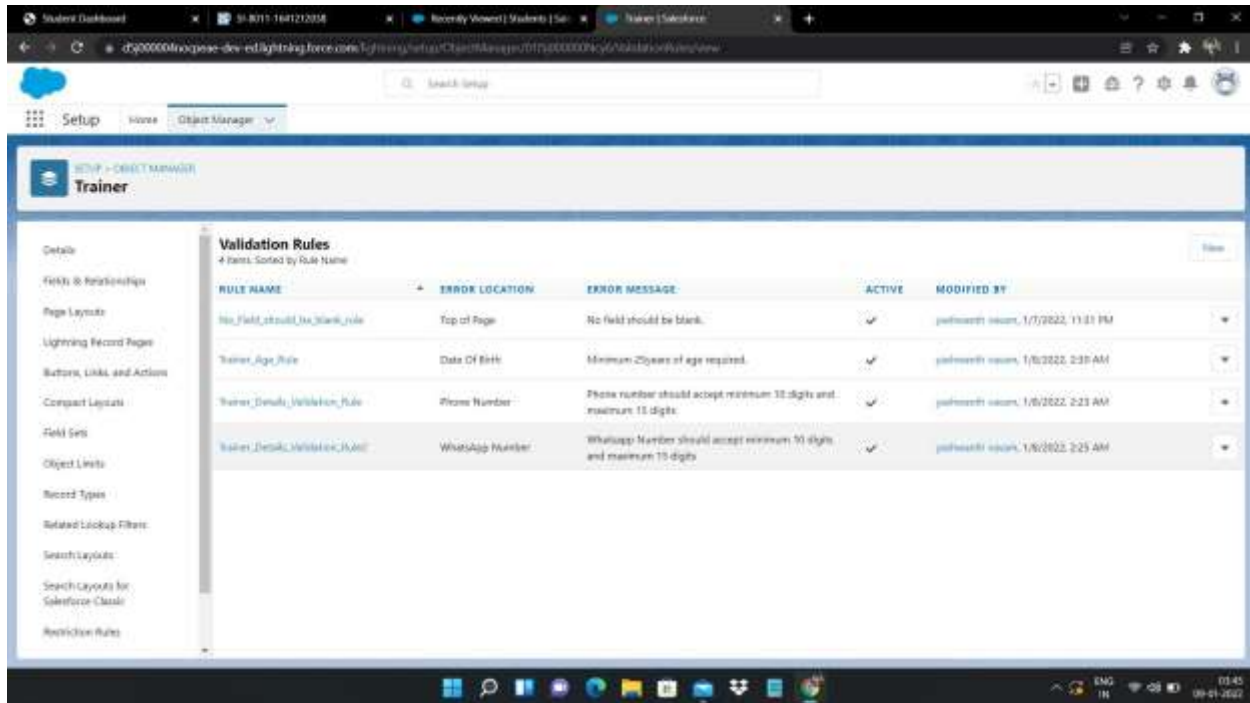
1. Course Fees Cannot be Negative: From Object Manager select the course object and click on validation rules, give your rule a name and in error condition formula enter "Course_Fee__c<0", give error message according to the condition.

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



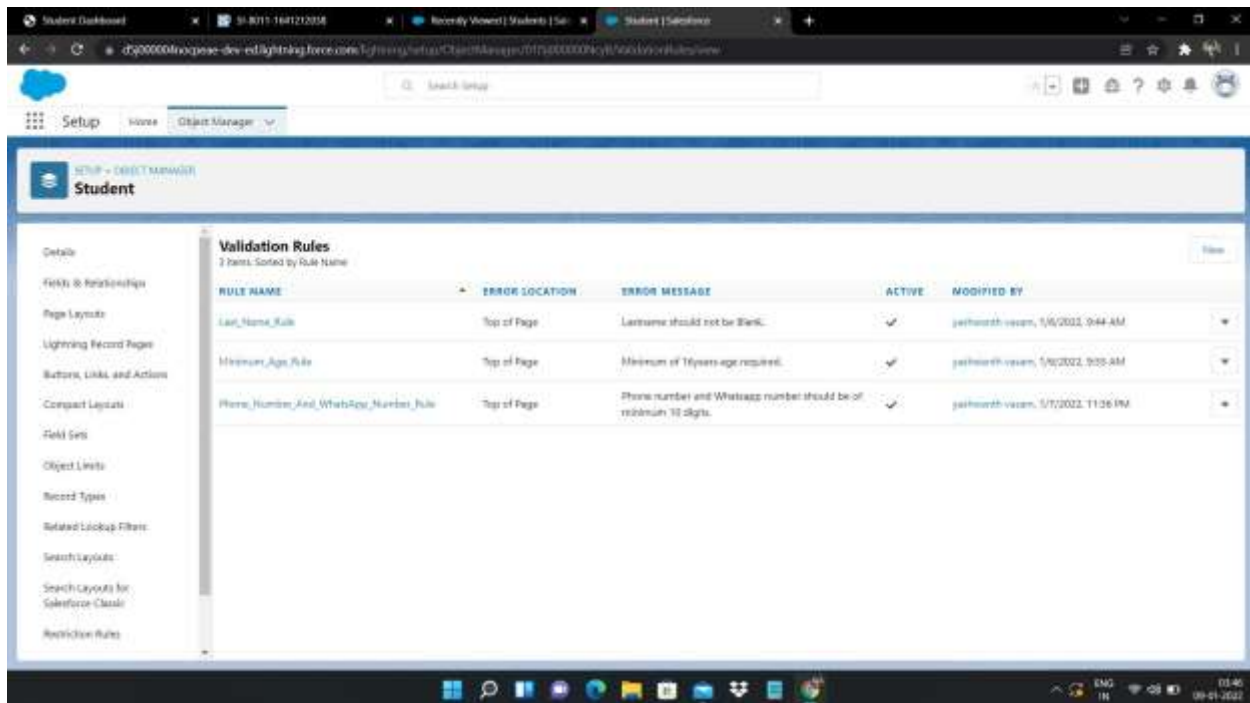
> For the Trainer object the validation rules we need to add is,

1. Full Name, Phone Number, Date of Birth, Experience fields on the trainer object should not be blank: Search for Trainer Object from the object manager, click on validation rules, click new and give your rule a name, in error condition formula enter "ISBLANK(Field Name)" and enter the error message according to the validation rule.
2. Age Should Not Be Less than 25 years: Enter "age__c<25" in error condition formula and enter the error message according to the rule.
3. Phone Number and Whatsapp Number should accept minimum 10 digits and maximum 15 digits: In error condition formula enter "NOT(REGEX(FIELDNAME, "[0-9]{10,15}"))" and error message according to the validation rule.



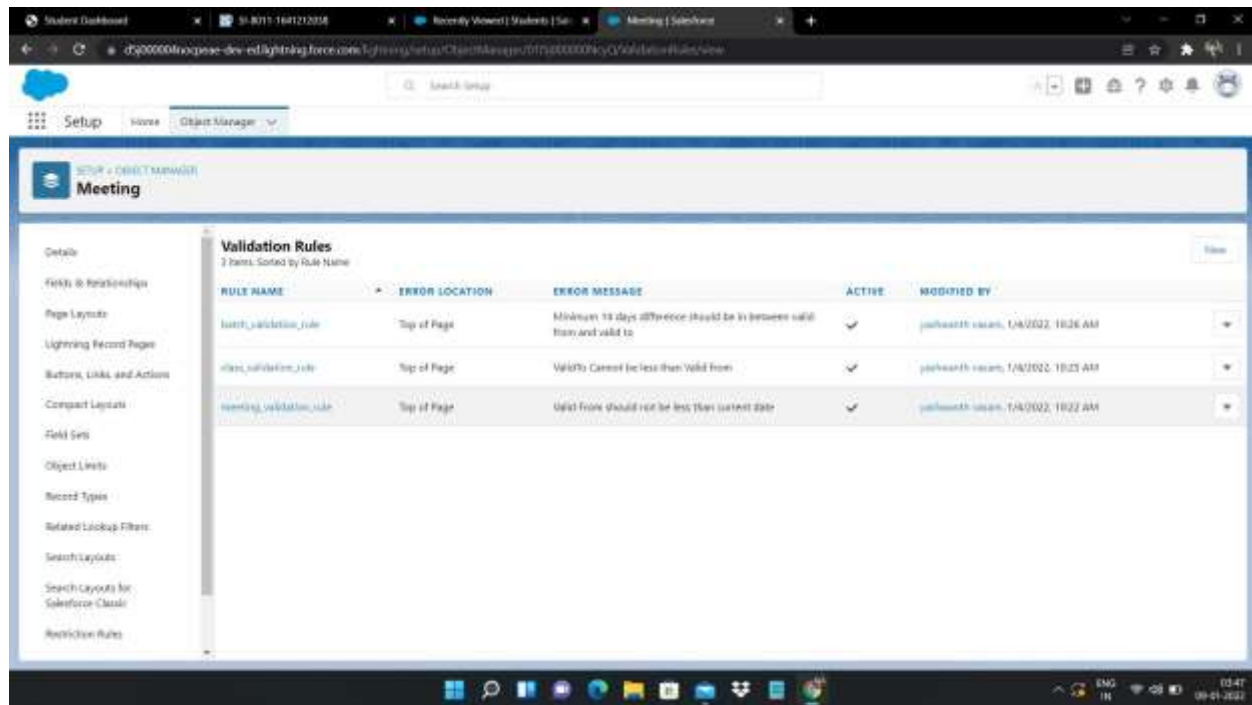
> For the Student object the validation rules we need to add is,

1. Lastname should not be Blank: enter the following error condition formula "ISBLANK(Field name)" and give the error message according to the rule.
2. Minimum 16 years of age: student should be minimum 16 years of age, in error condition formula enter "age__c<16" and give the error message according to the rule.
3. Phone and whatsappNumber should be minimum 10 digits : in error condition formula enter "NOT(REGEX(Fieldname, "[0-9]{10}"))" and give the error message according to the validation rule.



> For the Meeting object the validation rules we need to add is,

1. Valid From should not be less than current date: Click on validation rules from meeting object, give your rule a name and in error condition formula enter "`Valid_from__c < TODAY()`" and enter the error message according to the rule.
2. Valid To Cannot be less than Valid from : Click on validation rules from meeting object, give your rule a name and in error condition formula enter "`Valid_to__c < Valid_from__c`" and enter the error message according to the rule.
3. Minimum 14 days difference should be in between valid from and valid to : Click on validation rules from meeting object, give your rule a name and in error condition formula enter "`Valid_to__c - Valid_from__c < 14`" and enter the error message according to the rule.



> Process Automation:-

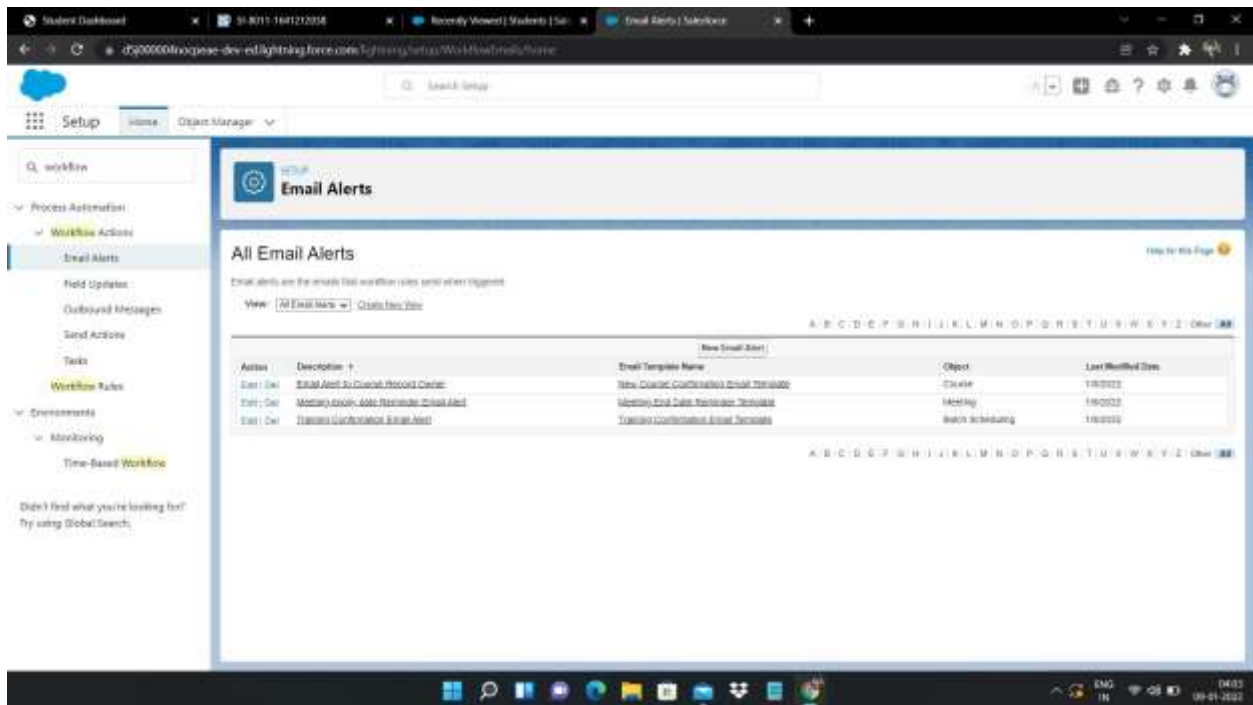
Salesforce offers tools to automate several kinds of business processes: guided visual experiences, behind-the-scenes automation, and approval automation. You'll be happy to know that these tools don't require you to write code—they're all point-and-click.

There are four automation tools in salesforce:

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

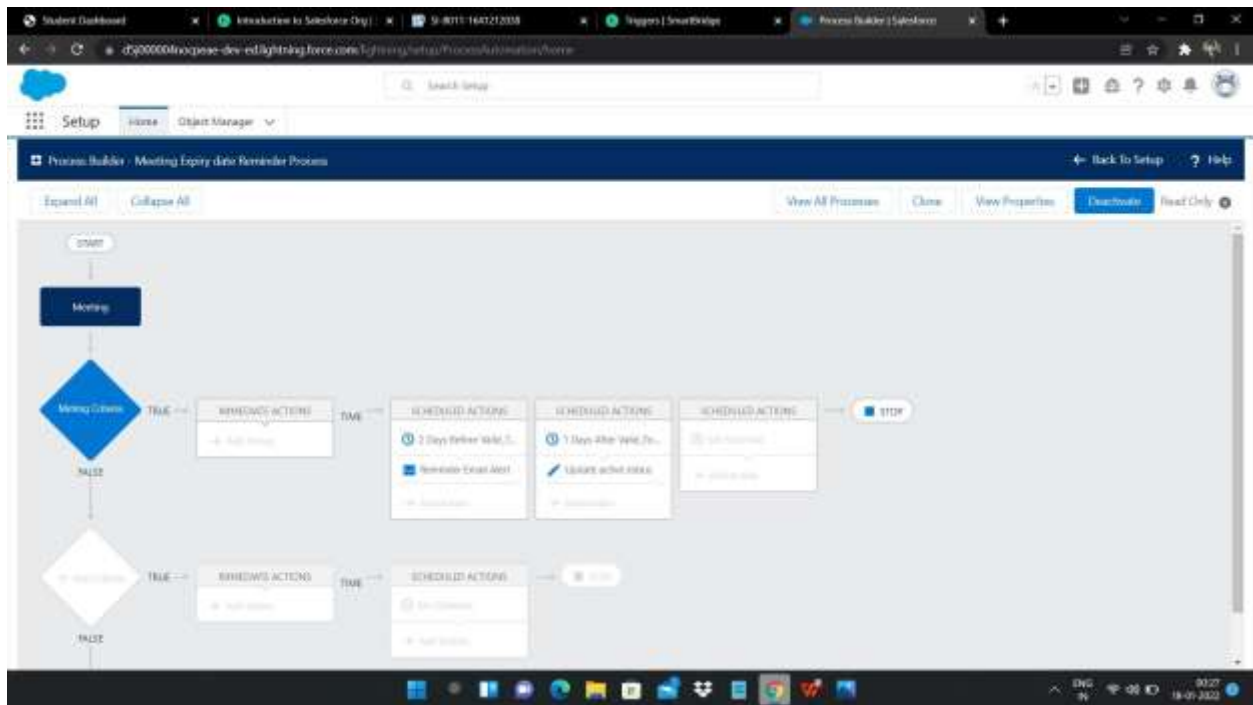
For Creating an automation first we need to add the email alerts. We added email alerts for,

- 1)Email Alert to Course Record owner
- 2)Meeting Expiry Date Reminder Alert
- 3)Training Confirmation Email Alert

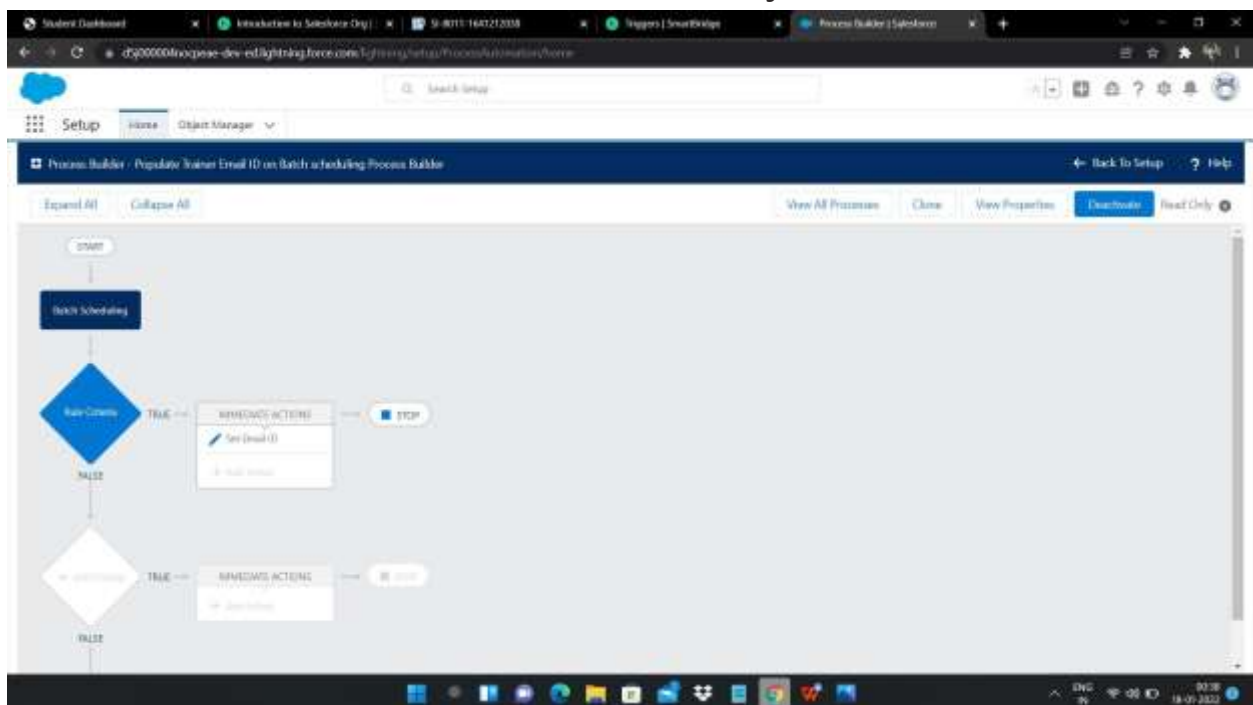


We can use the process builder for the automation,

Creating Work Flow Rules On Meeting Object:-



Create Work Flow Rules On Class/Batch Object:-



>Creating Custom Pages Using Visualforce:-

Visualforce is a framework that allows developers to build sophisticated, custom user interfaces that can be hosted natively on the Lightning platform. The Visualforce

framework includes a tag-based markup language, similar to HTML, and a set of serverside “standard controllers” that make basic database operations, such as queries and saves, very simple to perform.

Creating New Course Using Custom Controller:-

Using the visual force code and the custom controller code we obtain the new Course.

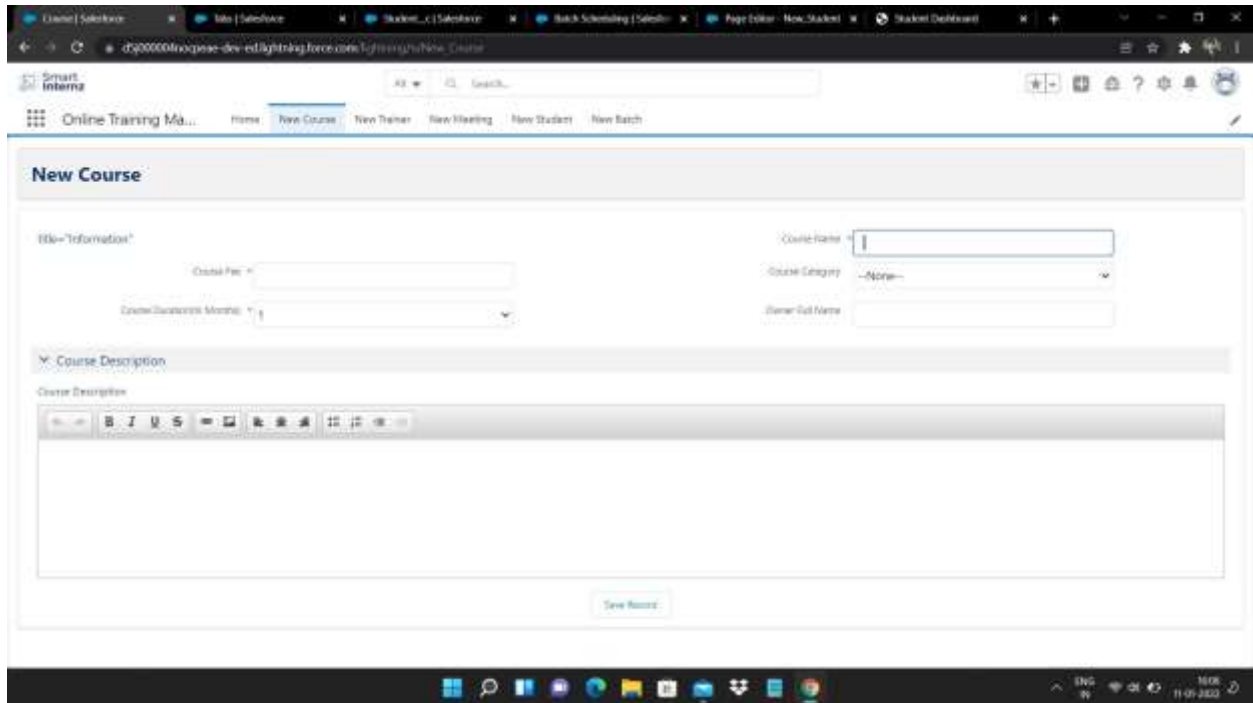
course visualforce page code

```
<apex:page lightningStylesheets="true" controller="course">
  <apex:form >
    <apex:sectionHeader title="New Course"/>
    <apex:pageMessages ></apex:pageMessages>
    <apex:pageBlock >
      <apex:pageBlockSection >
        <apex:inputfield value="{!cr.Course_Name__c}"/>
        <apex:inputfield value="{!cr.Course_Category__c}"/>
        <apex:inputfield value="{!cr.Course_Fee__c}"/>
        <apex:inputfield value="{!cr.Course_Duration__c}"/>
        <apex:inputfield value="{!cr.Course_Description__c}"/>
      </apex:pageBlockSection>
      <center><apex:commandButton value="Save" action="{!save}"/></center>
    </apex:pageBlock>
  </apex:form>
</apex:page> course custom
```

controller code:

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

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



Creating New Trainer Using Custom Controller:-

Using the visual force code and the custom controller code we obtain the new Trainer.

Trainer visualforce code:

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


</apex:page>

Trainer controller code:

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

The screenshot shows a web browser window with multiple tabs. The active tab is titled 'New Trainer | Salesforce'. The address bar shows a URL starting with 'ds000004...'. The page header includes a search bar and navigation links: 'Online Training Ma...', 'Home', 'New Course', 'New Trainer', 'New Training', 'New Student', and 'New Batch'. The main content area is titled 'New Trainer' and contains a form with the following fields: 'Full Name' (text input), 'Whatsapp Number' (text input), 'Experience' (dropdown menu with 'None' selected), 'Age' (text input), 'Phone Number' (text input), 'Date Of Birth' (calendar icon showing '1/11/2022'), and 'Expertise' (text input). A 'Save' button is located at the bottom right of the form. The browser's taskbar at the bottom shows various application icons and the system clock indicating 'MON 11:09:38.00'.

Creating New Student Using Standard Controller:-

Using student visual force code we can obtain the New Student.

Student visualforce code

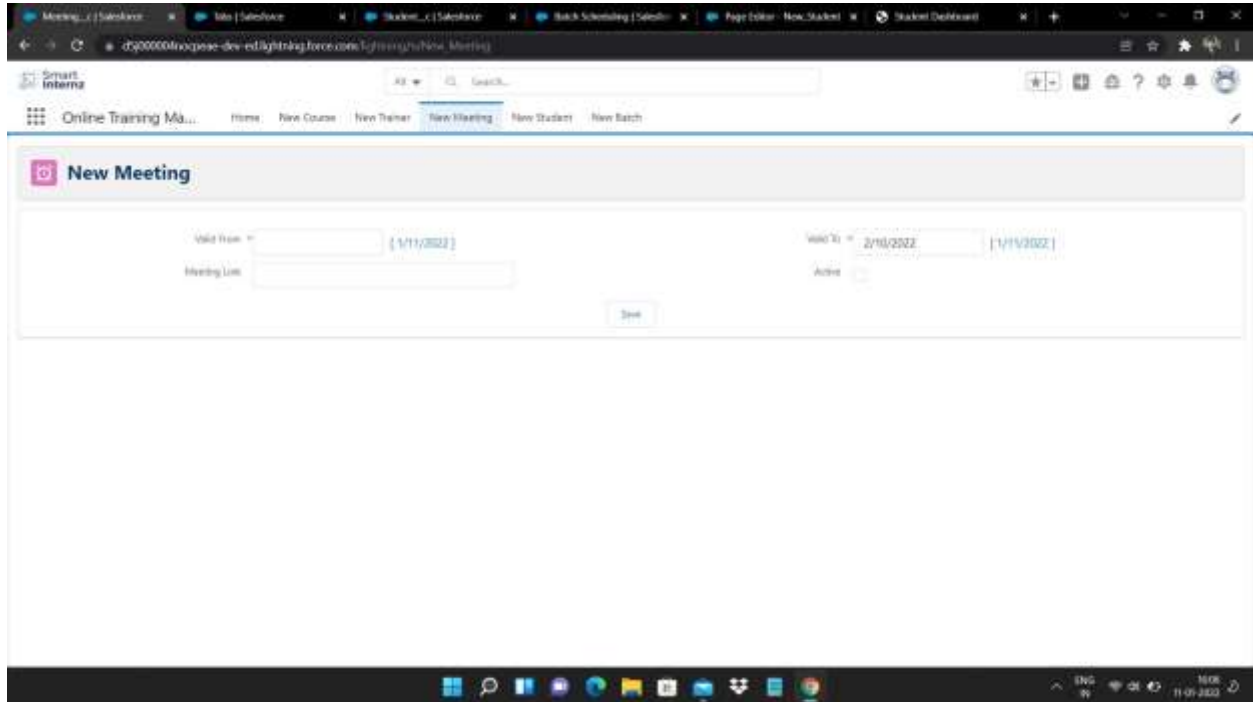
```
<apex:page lightningStyleSheets="true" standardController="Students__c">  
    <apex:form >  
        <apex:sectionHeader title="New Student"/>
```



```

        <apex:inputField value="{!Meeting__c.Active__c}"/>
    </apex:PageBlockSection>
    <center><apex:commandButton value="Save" action="{!save}"/></center>
</apex:PageBlock>
</apex:form>
</apex:page>

```



Create New Batch Using Standard Controller:-

Using Batch visual force code we can obtain the New Batch.

Batch visualforce code:

```

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

```

```

        <center><apex:commandButton value="Save" action="{!save}"/></center>
    </apex:pageBlock>
</apex:form>
</apex:page>

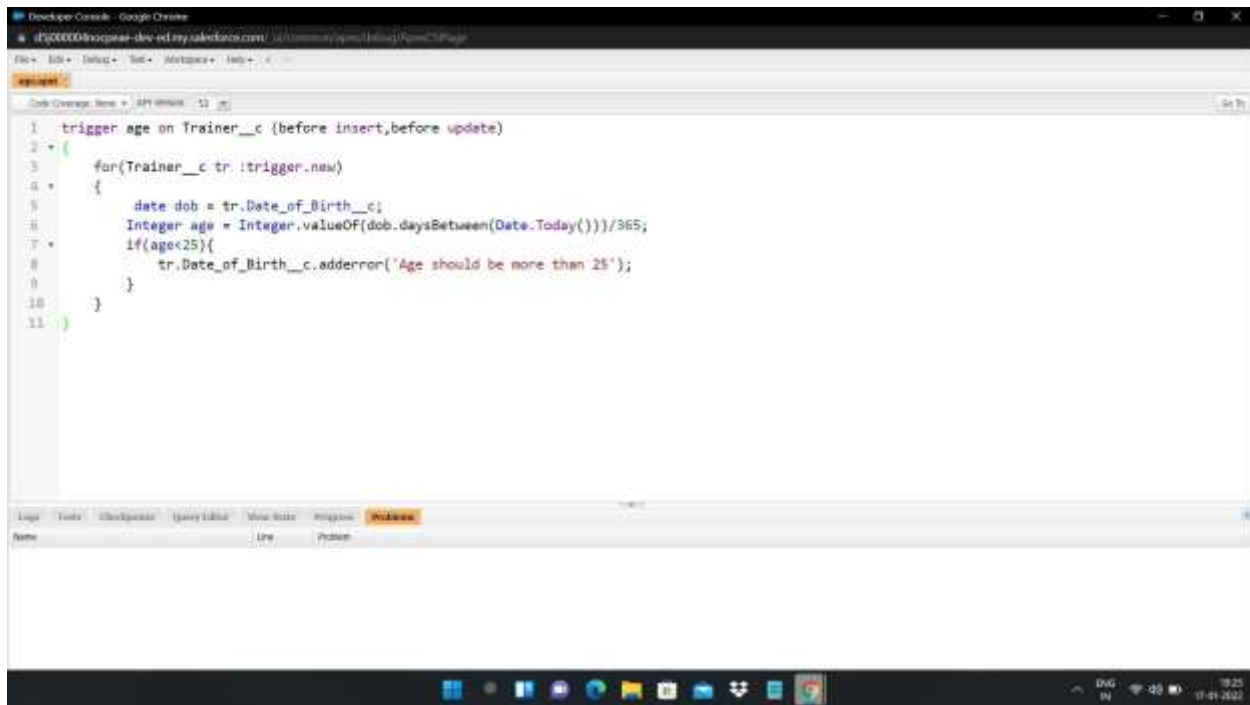
```

The screenshot shows a Salesforce Lightning interface for a 'New Batch' form. The header includes the 'Smart Internz' logo and navigation tabs: 'Online Training Ma...', 'Home', 'New Course', 'New Trainer', 'New Meeting', 'New Student', and 'New Batch'. The form itself is titled 'New Batch' and contains several input fields: 'Batch Type' (a dropdown menu), 'End Date' (set to 2/25/2022), 'Start Date' (set to 1/11/2022), 'Meeting ID' (with a search icon), 'Course ID' (with a search icon), 'Trainer ID' (with a search icon), and 'Status' (a dropdown menu). A 'Save' button is located at the bottom right of the form area. The browser's taskbar at the bottom shows various application icons and the system clock indicating 11:05 AM on 11/09/2022.

> Triggers:-

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

Trigger On Trainer Object:-

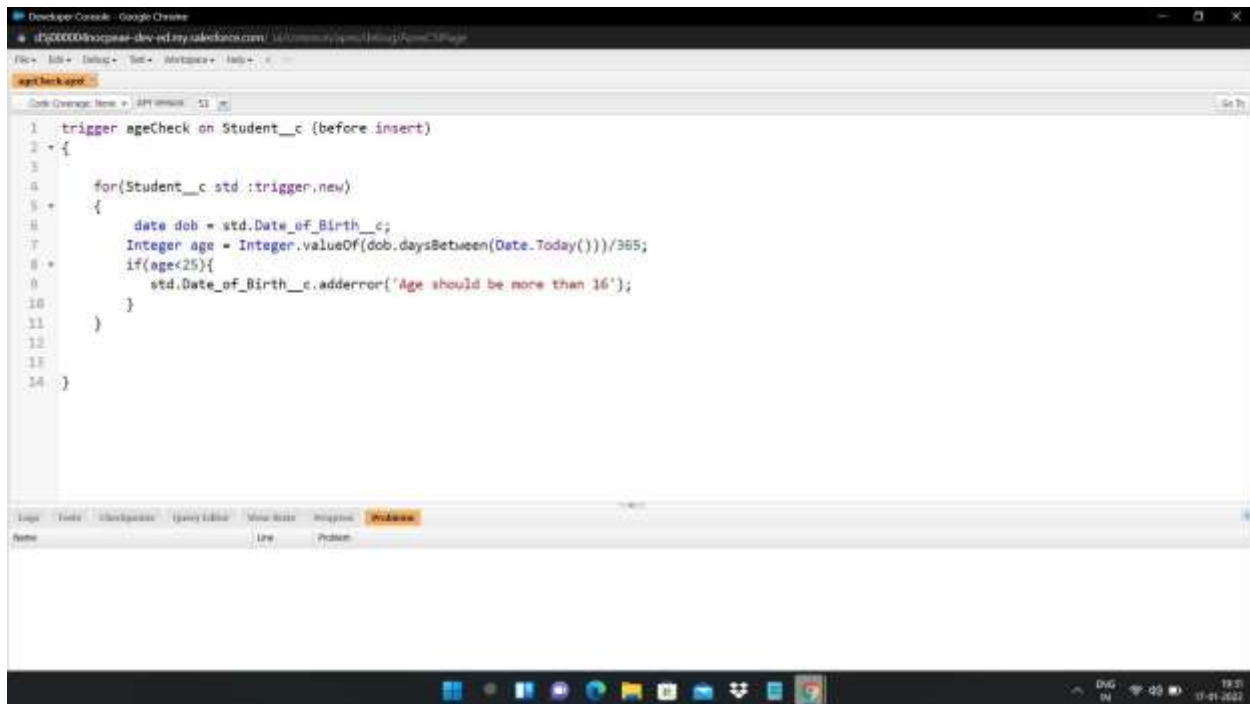


The screenshot shows a web browser window with the URL `https://www.salesforce.com/developer/console/`. The browser is displaying a page titled "ageTrigger". The page content is a code editor showing a trigger named "trigger age on Trainer__c (before insert,before update)". The code is as follows:

```
1 trigger age on Trainer__c (before insert,before update)
2 {
3     for(Trainer__c tr :trigger.new)
4     {
5         date dob = tr.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 }
```

The browser's status bar at the bottom shows the time as 19:25 on 07-01-2022.

Trigger On Student Object:-



The screenshot shows a web browser window with the URL `https://www.salesforce.com/developer/console/`. The browser is displaying a page titled "ageCheckAge". The page content is a code editor showing a trigger named "trigger ageCheck on Student__c (before insert)". 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 browser's status bar at the bottom shows the time as 19:31 on 07-01-2022.

Trigger On Meeting Object:-

```

1  for(register_c mt : trigger.new)
2  {
3      if(mt.Valid_From_c < Date.today())
4      {
5          mt.Valid_From_c.adderror('Valid From Date Should Be Greater Than Today');
6      }
7      if(mt.Valid_To_c < mt.Valid_From_c)
8      {
9          mt.Valid_To_c.adderror('Valid To Date should be greater than valid from date');
10     }
11     Integer noOfDays = (mt.Valid_From_c).daysBetween(mt.Valid_To_c);
12     if(noOfDays < 14)
13     {
14         mt.Valid_From_c.adderror('Minimum 14 days difference should be present between one meeting');
15     }
16 }

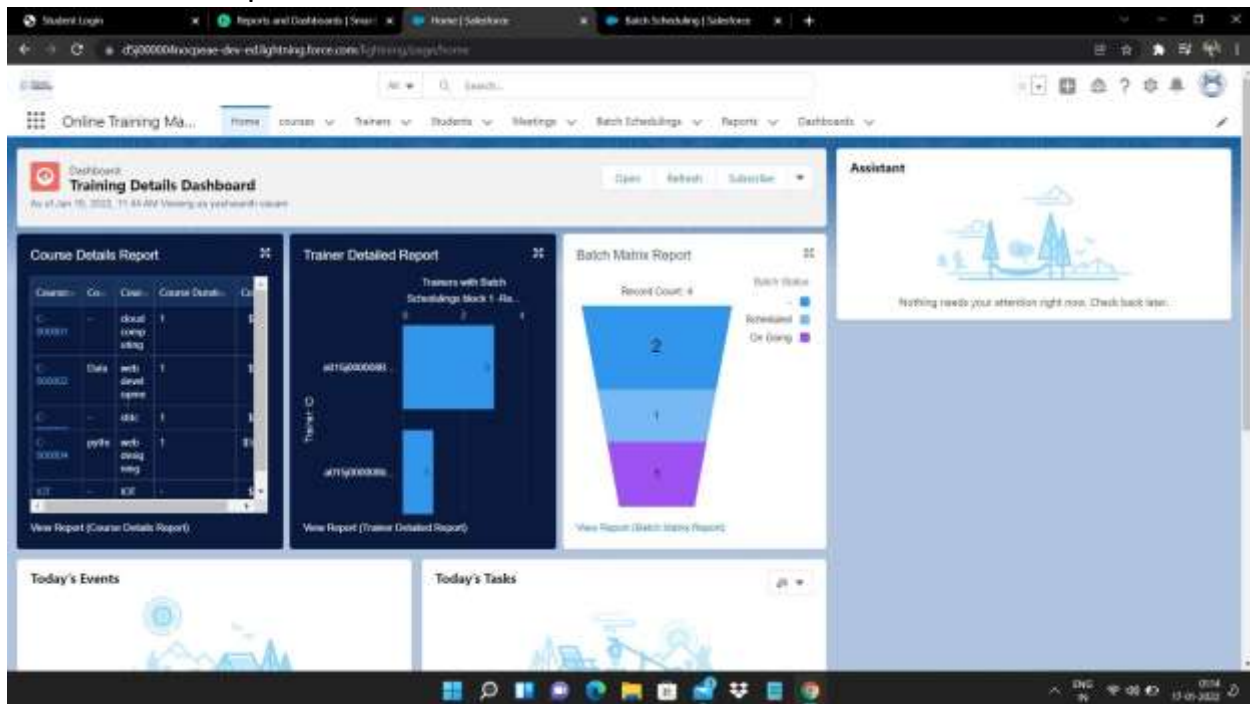
```

> Dashboards with Reports:-

We have Created a Training Details Dashboard using,
Course Details Report.

Trainer Detailed Report.

Batch Matrix Report.



Conclusion of the Project:-

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

Thank you.

