



## **SMART INTERNZ**

**SALESFORCE PROJECT READY - ONLINE TRAINING  
MANAGEMENT DEVELOPER.**

BY YASHWANTH KUMAR VASAM.

# **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/](https://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](mailto: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	DataType	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 Student Enrollment	Formula Master Detail(Batch)	TODAY()

### Meeting Object:-

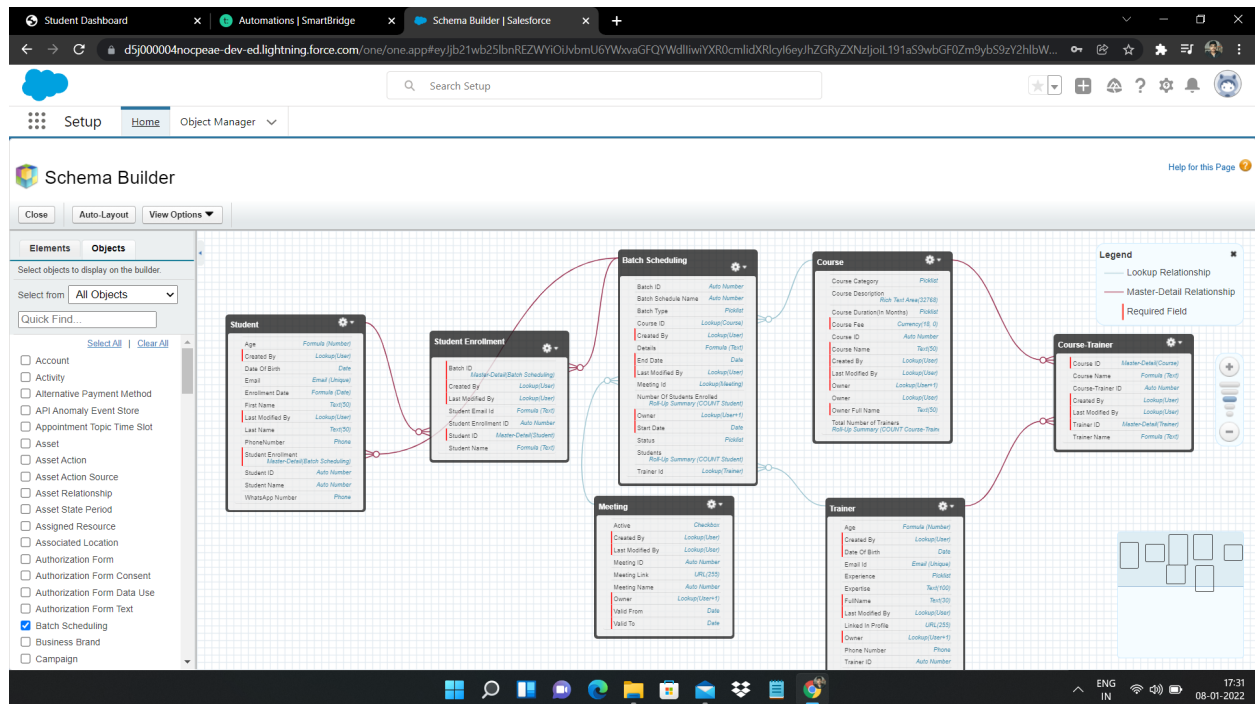
We need to add these fields in the object.

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	

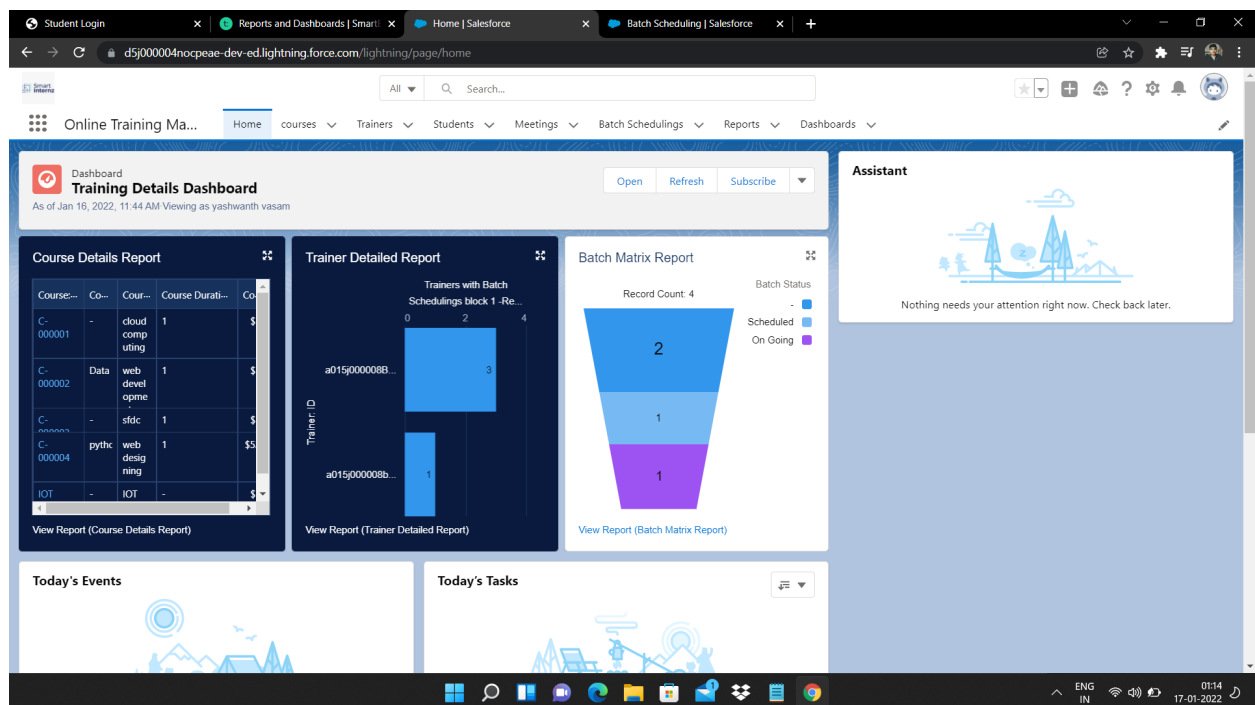
### Batch Scheduling Object:-

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,





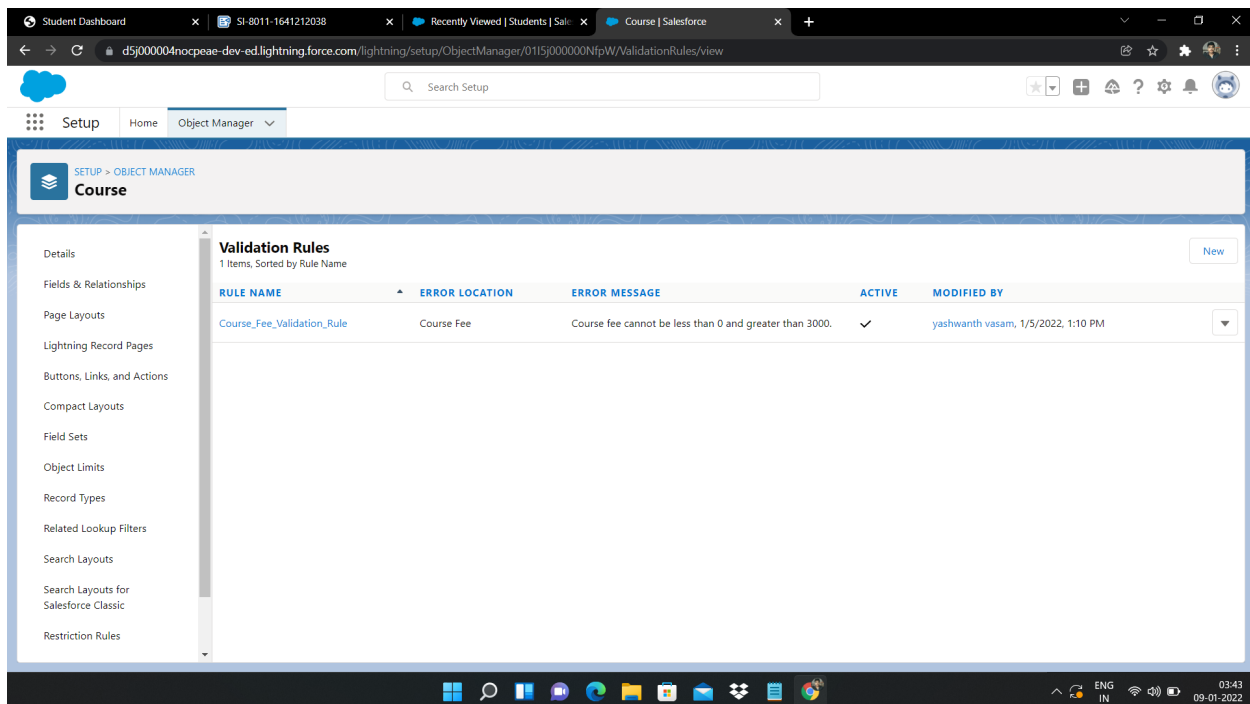
## > Create Validation Rules:-

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.



The screenshot shows the Salesforce Object Manager interface for the 'Course' object. The 'Validation Rules' section is active, displaying a table with one rule: 'Course\_Fee\_Validation\_Rule'. The rule is active and was modified by 'yashwanth vasam' on 1/5/2022 at 1:10 PM. The error message is 'Course fee cannot be less than 0 and greater than 3000.'.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Course_Fee_Validation_Rule	Course Fee	Course fee cannot be less than 0 and greater than 3000.	✓	yashwanth vasam, 1/5/2022, 1:10 PM

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

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

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

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

The screenshot shows the Salesforce Setup interface for the Trainer object. The left sidebar contains a navigation menu with 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 shows a list of 4 items, sorted by Rule Name. The table below lists the validation rules:

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
No_Field_should_be_blank_rule	Top of Page	No field should be blank.	✓	yashwanth vasam, 1/7/2022, 11:31 PM
Trainer_Age_Rule	Date Of Birth	Minimum 25years of age required.	✓	yashwanth vasam, 1/8/2022, 2:30 AM
Trainer_Details_Validation_Rule	Phone Number	Phone number should accept minimum 10 digits and maximum 15 digits	✓	yashwanth vasam, 1/8/2022, 2:23 AM
Trainer_Details_Validation_Rule2	WhatsApp Number	Whatsapp Number should accept minimum 10 digits and maximum 15 digits	✓	yashwanth vasam, 1/8/2022, 2:25 AM

> 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.

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: Last\_Name\_Rule, Minimum\_Age\_Rule, and Phone\_Number\_And\_WhatsApp\_Number\_Rule. Each rule is active and has a dropdown arrow for more options.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Last_Name_Rule	Top of Page	Lastname should not be Blank.	✓	yashwanth vasam, 1/6/2022, 9:44 AM
Minimum_Age_Rule	Top of Page	Minimum of 16years age required.	✓	yashwanth vasam, 1/6/2022, 9:55 AM
Phone_Number_And_WhatsApp_Number_Rule	Top of Page	Phone number and Whatsapp number should be of minimum 10 digits.	✓	yashwanth vasam, 1/7/2022, 11:36 PM

> 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. ValidTo Cannot be less than Valid from : Click on validation rules from meeting object, give your rule a name and in error condition formula enter "Valid\_to\_\_c < Valid\_from\_\_c " and enter the error message according to the rule.
3. Minimum 14 days difference should be in between valid from and valid to : Click on validation rules from meeting object, give your rule a name and in error condition formula enter "Valid\_to\_\_c - Valid\_from\_\_c < 14 " and enter the error message according to the rule.

The screenshot shows the Salesforce Setup interface for the Meeting object. 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, and Restriction Rules. The main content area is titled 'Validation Rules' and shows 3 items, sorted by Rule Name. A table lists the validation rules:

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
batch_validation_rule	Top of Page	Minimum 14 days difference should be in between valid from and valid to	✓	yashwanth vasam, 1/4/2022, 10:26 AM
class_validation_rule	Top of Page	ValidTo Cannot be less than Valid from	✓	yashwanth vasam, 1/4/2022, 10:25 AM
meeting_validation_rule	Top of Page	Valid From should not be less than current date	✓	yashwanth vasam, 1/4/2022, 10:22 AM

The bottom of the screen shows a Windows taskbar with various application icons and a system tray with the date and time (03:47, 09-01-2022).

## > 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,

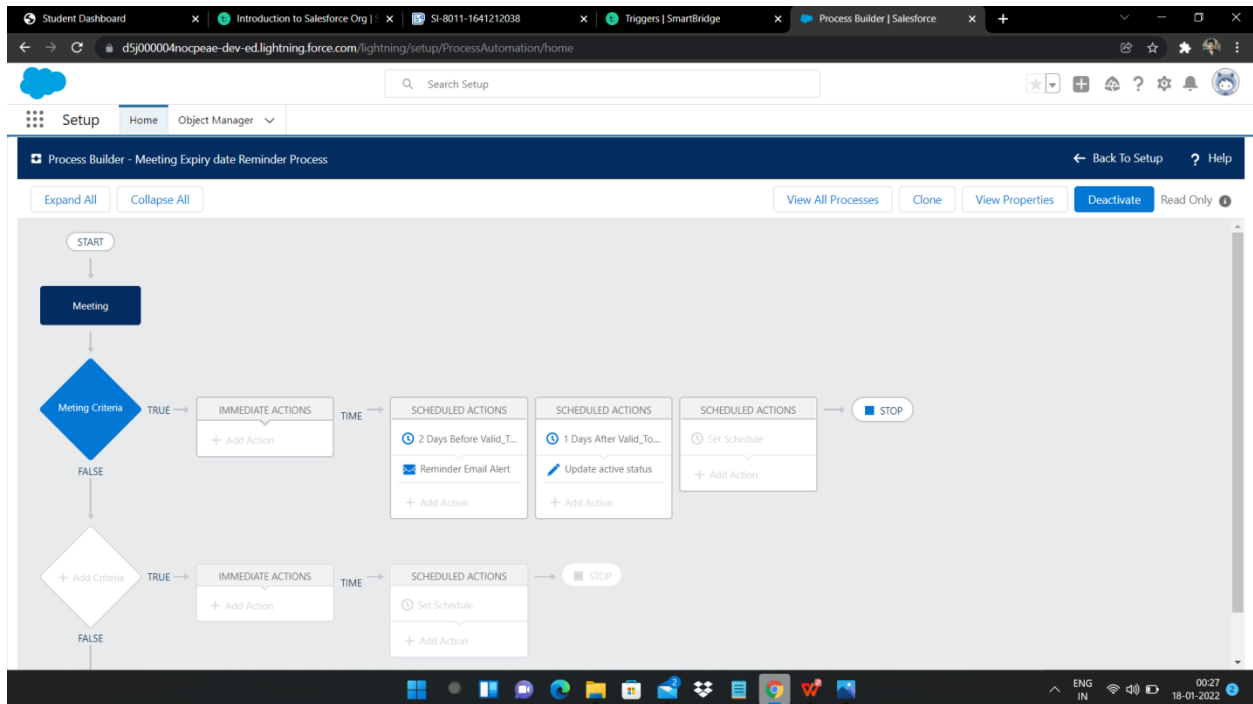
- 1)Email Alert to Course Record owner
- 2)Meeting Expiry Date Reminder Alert
- 3)Training Confirmation Email Alert

The screenshot shows the Salesforce Setup interface for Email Alerts. 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 is titled 'Email Alerts' and displays a table of all email alerts. The table has columns for Action, Description, Email Template Name, Object, and Last Modified Date. Three alerts are listed: 'Email Alert to Course Record Owner', 'Meeting Expiry Date Reminder Email Alert', and 'Training Confirmation Email Alert'. Each alert has an 'Edit' and 'Del' link. The table also includes a 'New Email Alert' button and a 'View' dropdown menu.

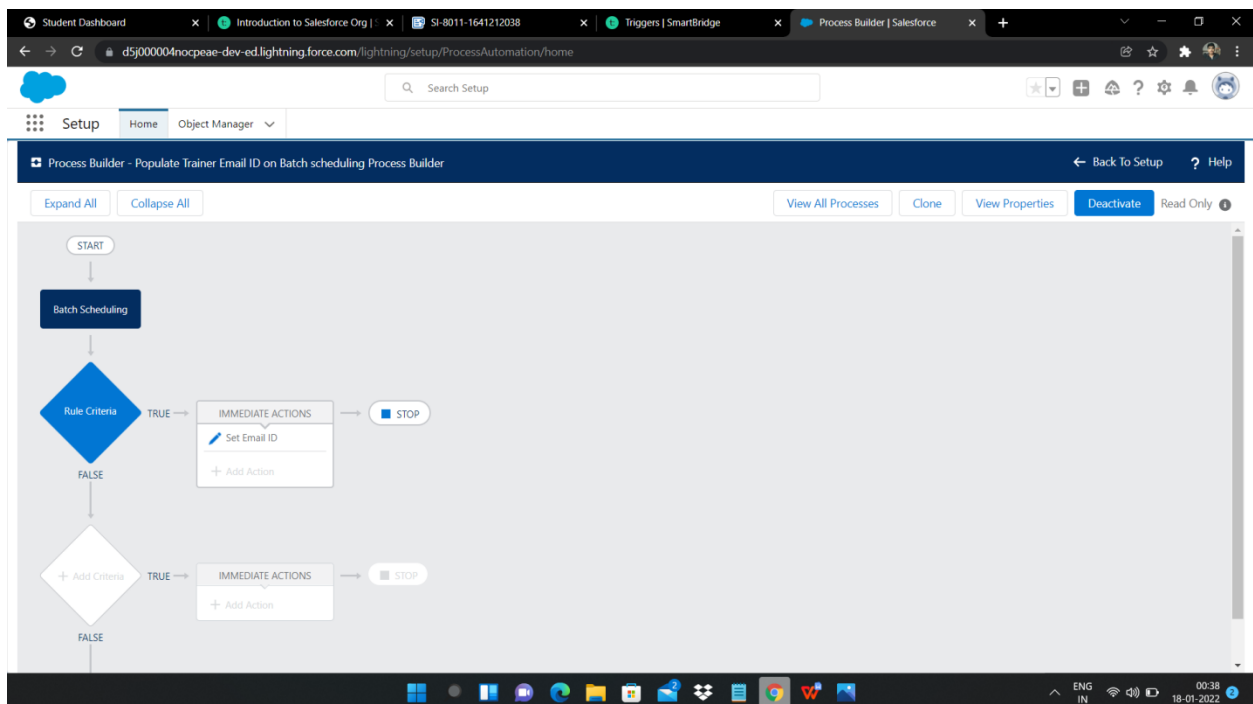
Action	Description	Email Template Name	Object	Last Modified Date
<a href="#">Edit</a>   <a href="#">Del</a>	Email Alert to Course Record Owner	New Course Confirmation Email Template	Course	1/8/2022
<a href="#">Edit</a>   <a href="#">Del</a>	Meeting expiry date Reminder Email Alert	Meeting End Date Reminder Template	Meeting	1/8/2022
<a href="#">Edit</a>   <a href="#">Del</a>	Training Confirmation Email Alert	Training Confirmation Email Template	Batch Scheduling	1/8/2022

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 server-side “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.

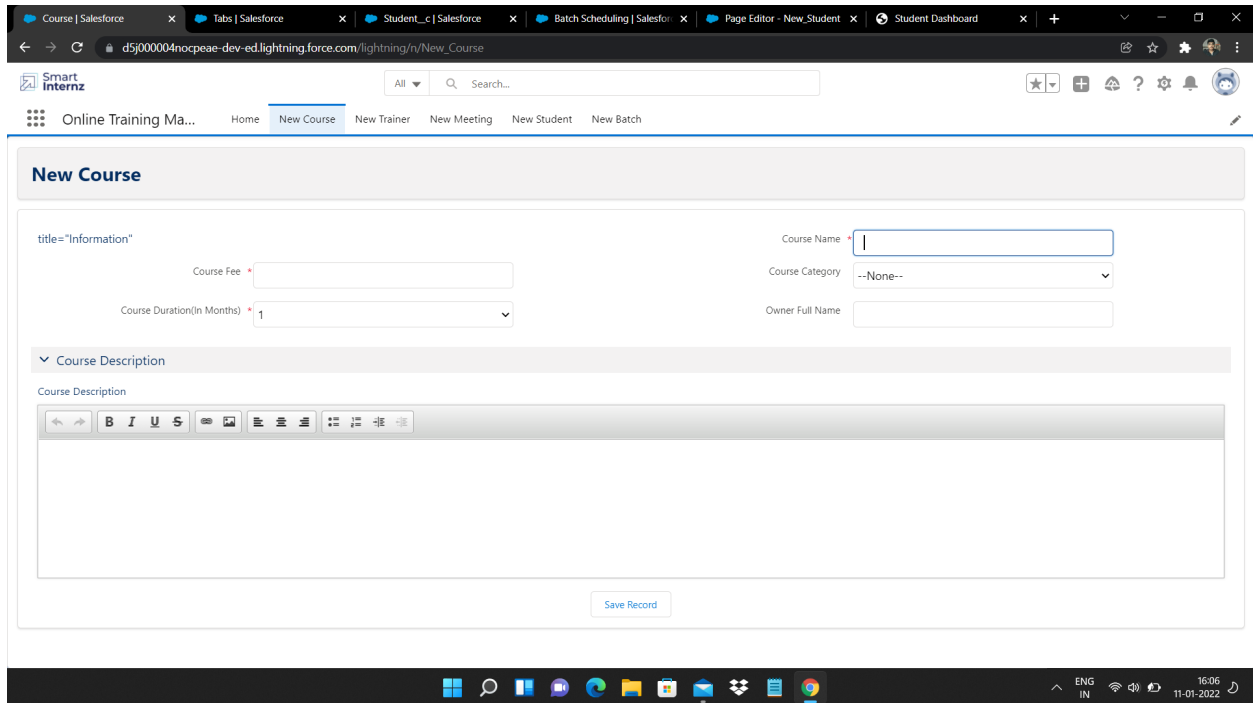
course visualforce page code

```
<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>
```



```
</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
        Successssfully'));
        tr=new Trainer__c();
    }
}
```

The screenshot shows a web browser window with multiple tabs open, including 'New Trainer | Salesforce', 'Student\_\_c | Salesforce', 'Batch Scheduling | Salesforce', 'Page Editor - New Student', and 'Student Dashboard'. The active tab is 'New Trainer | Salesforce', displaying a Lightning page titled 'New Trainer'. The page has a navigation bar with links: 'Online Training Ma...', 'Home', 'New Course', 'New Trainer' (active), 'New Meeting', 'New Student', and 'New Batch'. The main content area contains a form with the following fields: 'FullName' (text input), 'WhatsApp Number' (text input), 'Experience' (dropdown menu with '--None--' selected), 'Age' (text input), 'Phone Number' (text input), 'Date Of Birth' (date picker showing '1/11/2022'), and 'Expertise' (text input). A 'Save' button is located at the bottom right of the form. The browser's address bar shows the URL 'https://d5j000004nocpeae-dev-ed.lightning.force.com/lightning/r/New\_Trainer'. The Windows taskbar at the bottom shows the time as 16:06 on 11-01-2022.

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

The screenshot shows a web browser window displaying a Salesforce Lightning page. The page title is "New Student". The form contains the following fields:

- First Name:
- Last Name:
- Email:
- Phone Number:
- Date Of Birth:
- WhatsApp Number:

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

## Create New Meeting Using Standard Controller:-

Using meeting visual force code we can obtain the New Meeting.

Meeting visualforce code:

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

# 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 displays the Salesforce Lightning interface for creating a new batch. The browser window shows the URL `d5j000004nocpeae-dev-ed.lightning.force.com/lightning/r/New_Batch`. The page title is "New Batch". The form contains the following fields:

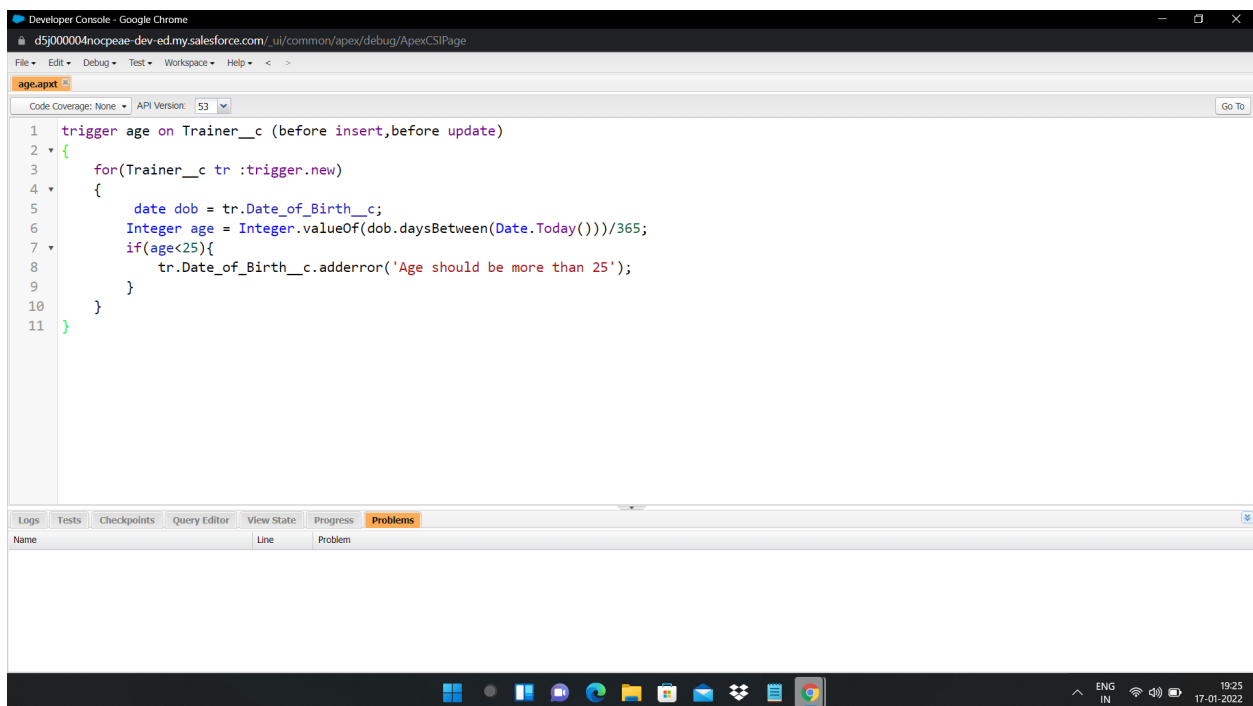
- Batch Type: A dropdown menu with "--None--" selected.
- Start Date: A date field with a red asterisk, showing "1/11/2022".
- End Date: A date field with a red asterisk, showing "2/25/2022" and a calendar icon.
- Course ID: A text field with a search icon.
- Meeting Id: A text field with a search icon.
- Trainer Id: A text field with a search icon.
- Status: A dropdown menu with "--None--" selected.

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

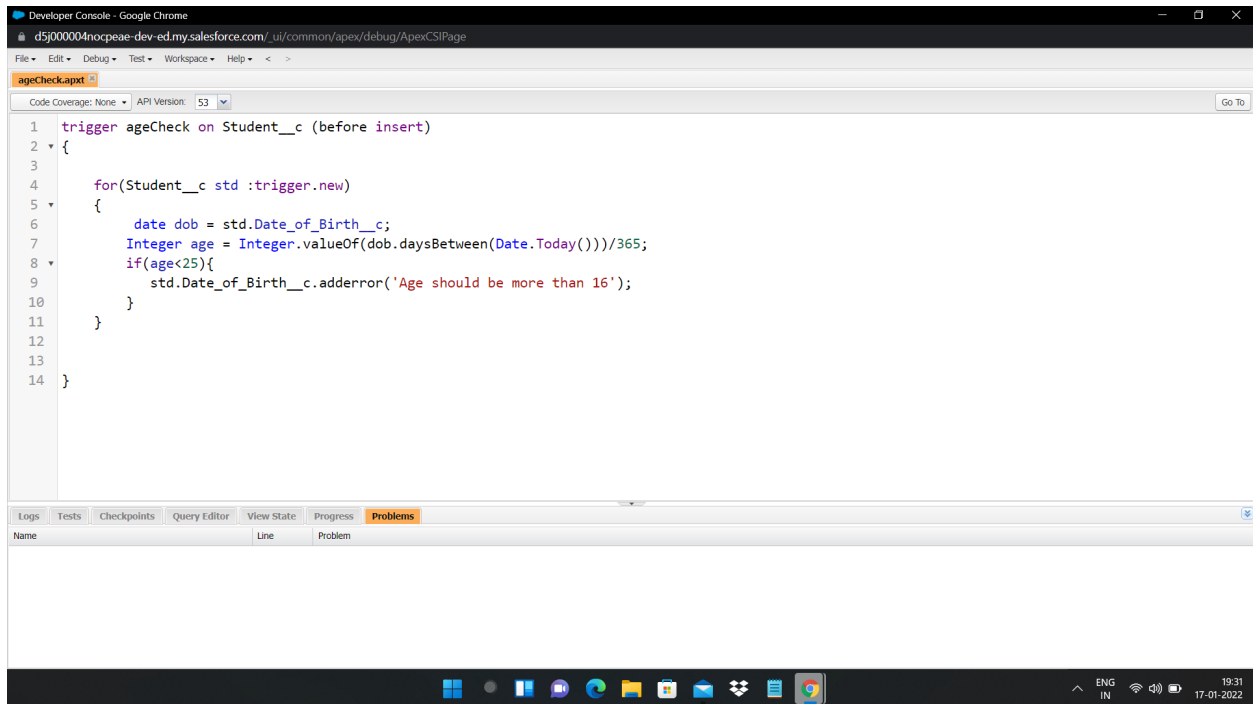
## > Triggers:-

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.

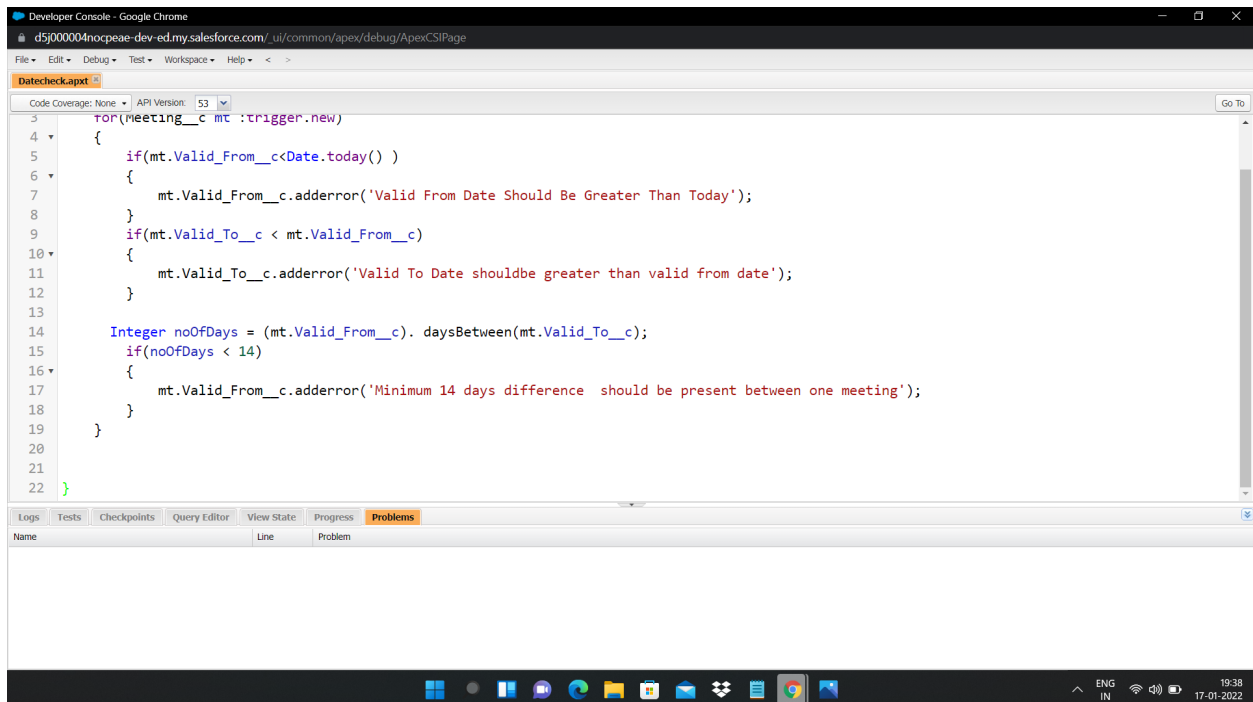
### Trigger On Trainer Object:-



## Trigger On Student Object:-



## Trigger On Meeting Object:-

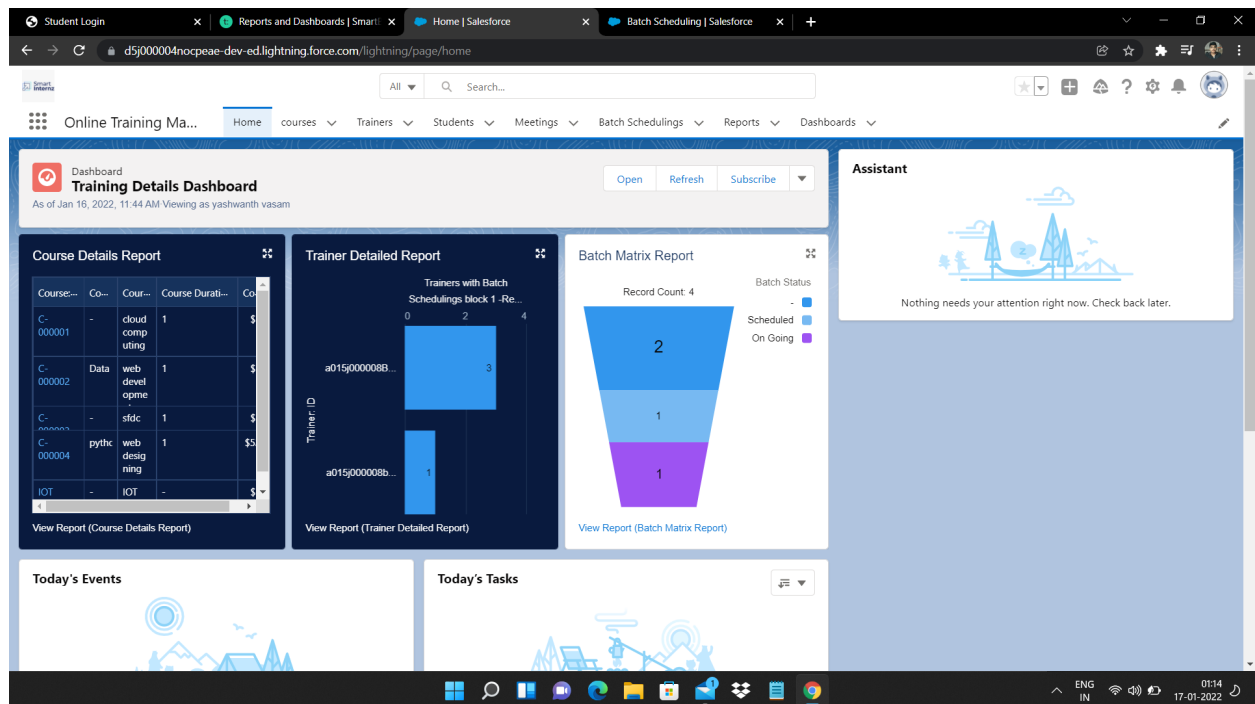


## > DashBoards with Reports:-

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 an 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 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.

