



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

**INDEX:**

#### BY ABINASH PUHAN.

SALESFORCE PROJECT READY - ONLINE TRAINING MANAGEMENT DEVELOPER.

* **Create Your Salesforce Developer Org To GetStarted.**

**>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 GetStarted:-

Creating a developer org in salesforce.

* 1. Go to[developers.salesforce.com/](https://developer.salesforce.com/)2.Click on signup.

1. On the sign up form, enter the following details: A.First name & Lastname
   1. Email
   2. Role :Developer
   3. Company : CollegeName
   4. County :India
   5. Postal Code : pincode

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 onlogin.
2. Enter the username and password that you justcreated. 3.After login this is the home page which you willsee.

* **Custom ObjectCreation:-**

**Creating The Custom Objects For The Application.**

The required custom objects are:

* 1. Courses. b)Trainer. c)Student. d)Meeting.

e)Batch/Class Scheduling.

Course Object:-

We need to add these fields in the object.

|  |  |  |
| --- | --- | --- |
| FieldName | DataType | Values |
| Course ID | AutoNumber(ChangethedatatypetoAutoNumber  Whilecreatingcourseobjectand  changingrecordTypetocourse id) | Format(C-000001) Startingnumber=1 |
| CourseName | Text |  |
| Course  Duration | Picklist | 1month,2 months,3  months |
| Course Fees | Currency |  |
| Course  Description | Text(Long) |  |
| Course Category  CourseTrainer | Picklist  Master-Detail(Trainer) | Datascience,cloudcomputing,AI,ML |

Trainer Object:-

We need to add these fields in the object.

|  |  |  |
| --- | --- | --- |
| FieldName | DataType | Values |

|  |  |  |
| --- | --- | --- |
| TrainerID | Autonumber | Format=T-00001,Startingnumber=1 |
| FullName | Text |  |
| Email | Email |  |
| PhoneNumber | Phone |  |
| WhatsappNumber | Phone |  |
| DateofBirth | Date |  |
| Age | Formula | TODAY()-Date\_of\_Birthc |
| Experience | Picklist | 1year,2years,3years,……… |
| LinkedIN Profile | URL |  |
| Expertise | Text |  |

Student Object:-

We need to add these fields in the object.

|  |  |  |
| --- | --- | --- |
| FieldName | DataType | Values |
| Student ID | Autonumber | Format=S-00001,Starting  Number=1 |
| FirstName | Text |  |
| LastName | Text |  |
| Email | Email |  |
| PhoneNumber | Phone |  |
| DateOfBirth | Date |  |
| WhatsappNumber | Phone |  |
| EnrollmentDate  Student Enrollment | Formula  Master Detail(Batch) | TODAY() |
|  |

**Meeting Object:-**

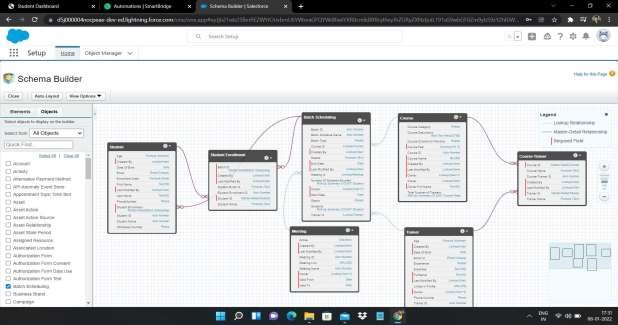
We need to add these fields in the object.

|  |  |  |
| --- | --- | --- |
| FieldName | DataType | Values |

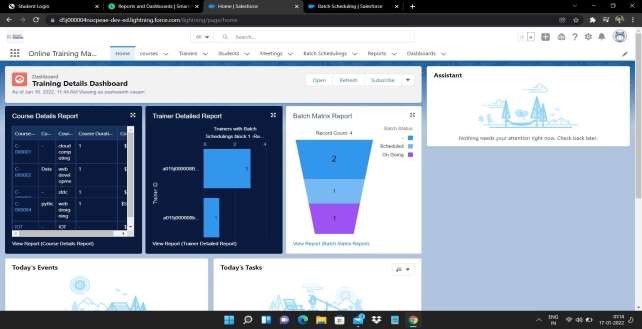
|  |  |  |
| --- | --- | --- |
| Meeting ID | Autonumber | Format=M-00001,StartingNumber=1 |
| ValidFrom | Date |  |
| ValidTo | Date |  |
| Active | Checkbox |  |
| Meeting Link | URL |  |

Batch Scheduling Object:-

|  |  |  |
| --- | --- | --- |
| FieldName | DataType | Values |
| Meeting ID | Autonumber | Format=M-00001,StartingNumber=1 |
| ValidFrom | Date |  |
| ValidTo | Date |  |
| Active | Checkbox |  |
| Meeting Link | URL |  |

The data model of the above Custom objects is,

The Application Home Tab looks like,

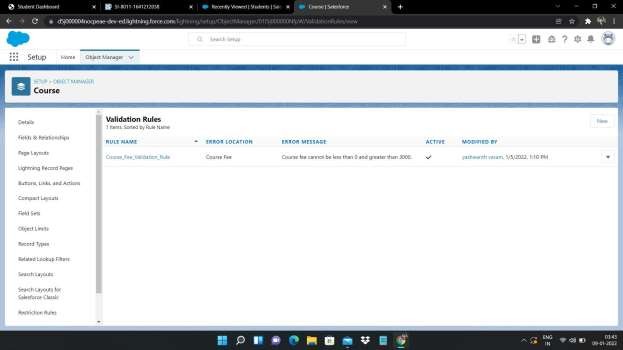


* **Create ValidationRules:-**

The Validation rules verifys 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.

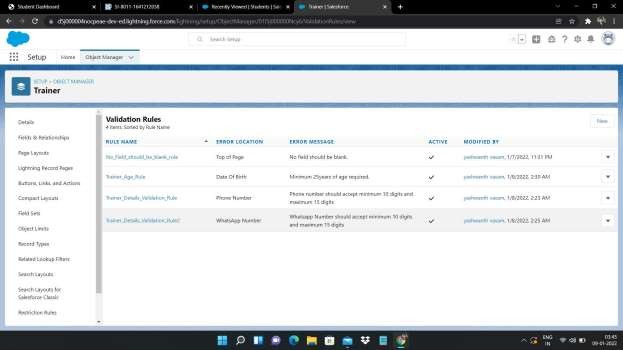
* For the Course object the validation rules we need to addis,

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 thecondition.
2. CourseFeesShould BeLessthan3000$:FromObjectManagerselectthecourseobjectand click on validation rules, give your rule a name and in error condition formula enter "Course\_Fee c>=3000",give error message according to thecondition.



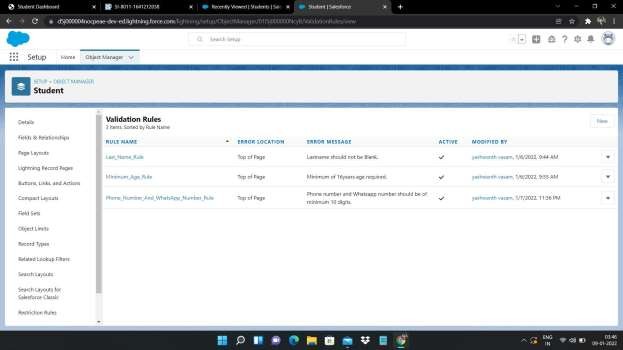
* + For the Trainer object the validation rules we need to addis,

1. FullName,PhoneNumber,DateofBirth,Experiencefieldsonthetrainerobjectshould notbe 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 validationrule.
2. Age Should Not Be Less than 25 years: Enter "age c<25" in error condition formula and enter the error message according to therule.
3. Phone Number and Whatsapp Number should accept minimum 10 digits andmaximum 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 addis,

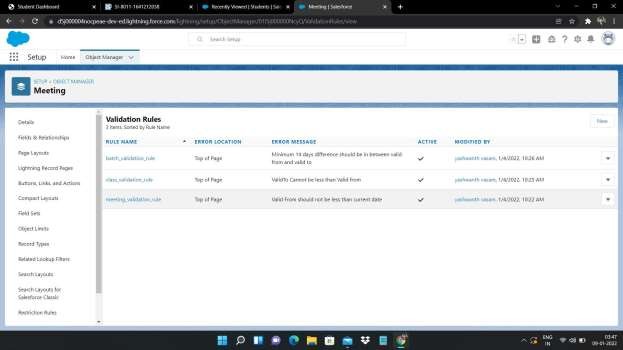
1. Lastname should not be Blank: enter the following error conditionformula "ISBLANK(Field name)" and give the error message according to therule.
2. Minimum 16 years of age: student should be minimum 16 years of age, in error condition formula enter "agec<16" and give the error message according to the rule. 3.PhoneandwhatsappNumbershouldbeminimum10digits:inerrorconditionformulaenter "NOT(REGEX(Fieldname, "[0-9]{10}"))" and give the error message accoding to the validation rule.



* + For the Meeting object the validation rules we need to addis,

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 therule.
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 therule.
3. Minimum14daysdifferenceshouldbeinbetweenvalidfromandvalidto:Clickon validation rules from meeting object, give your rule a name and in error condition formulaenter

"Valid\_toc - Valid\_fromc < 14 " and enter the error message according to the rule.



# ProcessAutomation:-

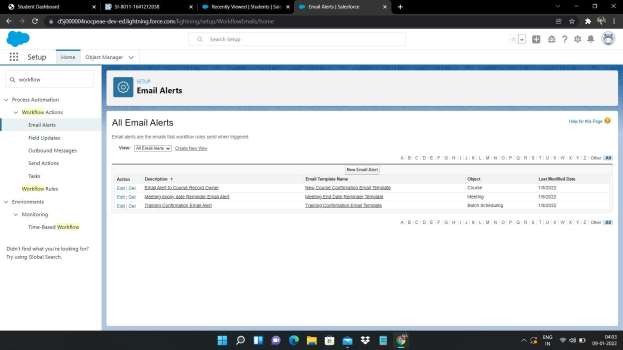
Salesforce offers tools to automate several kinds of business processes: guided visual experiences, behind-the-scenes automation, and approval automation. You'll be happy to know that these tools don't require you to write code—they're all point-and-click.

There are four automation tools in salesforce:

1. ApprovalProcess
2. Processbuilder
3. Flows
4. Workflows

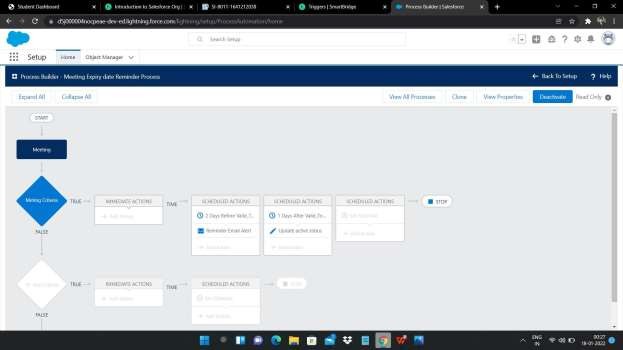
For Creating an automation first we need to add the email alerts. We added email alerts for,

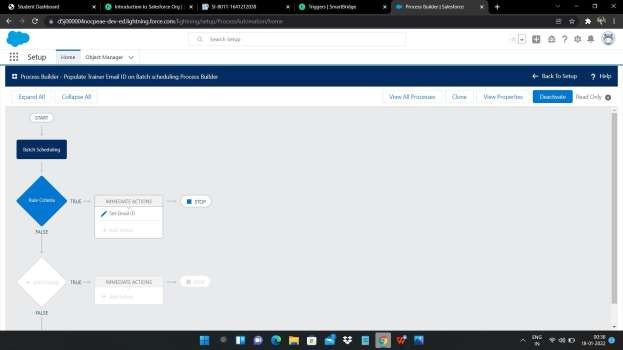
1)Email Alert to Course Record owner 2)Meeting Expiry Date Reminder Alert 3)Training Confirmation Email Alert



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:inputfieldvalue="{!cr.Course\_Description

</apex:pageBlockSection>

c}"/>

<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 publiccourse()

{

cr=newCourse

}

c cr{get;set;}

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 Successsfully'));

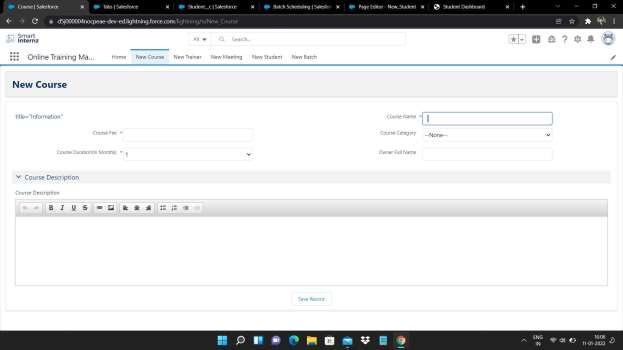
cr=new Course

}

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.FullNamec}" required="true"/>

<apex:inputfield value="{!tr.Emailc}" required="true"/>

<apex:inputfield value="{!tr.Phone\_Numberc}" required="true"/>

<apex:inputfield value="{!tr.Whatsapp\_Numberc}"/>

<apex:inputfield value="{!tr.Date\_Of\_Birthc}" required="true"/>

<apex:inputfield value="{!tr.Experiencec}" required="true"/>

<apex:inputfield value="{!tr.LinkedIN\_Profilec}"/>

<apex:inputfield value="{!tr.Expertisec}"/>

<apex:inputField value="{!tr.Agec}"/>

</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 Trainerc tr{get; set;} public trainer()

{

tr=new Trainerc();

}

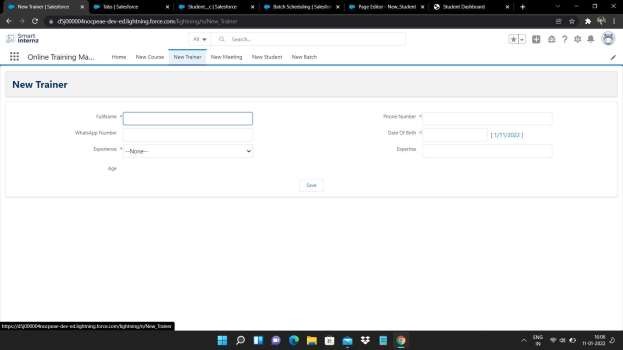
public void save()

{ insert tr;

ApexPages.addmessage(new ApexPages.message(ApexPages.severity.Confirm, 'Record Inserted Successsfully')); tr=new

Trainerc();

}

}

## 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="Studentsc">

<apex:form >

<apex:sectionHeader title="New Student"/>

<apex:pageBlock >

<apex:pageBlockSection >

<apex:inputField value="{!Studentsc.First\_Namec}"/>

<apex:inputField value="{!Studentsc.Last\_Namec}" required="true"/>

<apex:inputField value="{!Studentsc.Emailc}"/>

<apex:inputField value="{!Studentsc.Phone\_Numberc}"/>

<apex:inputField value="{!Studentsc.Date\_Of\_Birthc}"/>

<apex:inputField value="{!Studentsc.Whatsapp\_Numberc}"/>

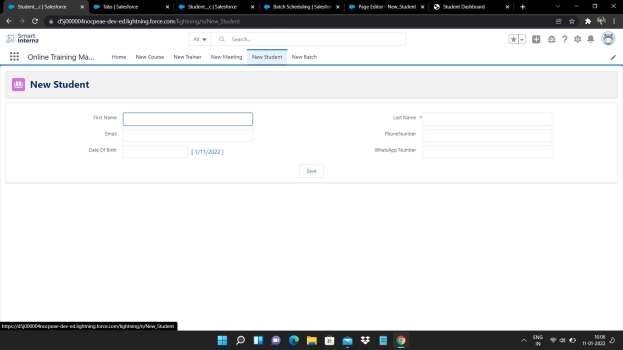
</apex:pageBlockSection>

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

</apex:pageBlock>

</apex:form>

</apex:page>



#### 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="Meetingc">

<apex:form >

<apex:sectionHeader title="New Meeting"/>

<apex:PageBlock >

<apex:PageBlockSection >

<apex:inputField value="{!Meetingc.Valid\_Fromc}"/>

<apex:inputField value="{!Meetingc.Valid\_Toc}"/>

<apex:inputField value="{!Meetingc.Meeting\_Linkc}"/>

<apex:inputField value="{!Meetingc.Activec}"/>

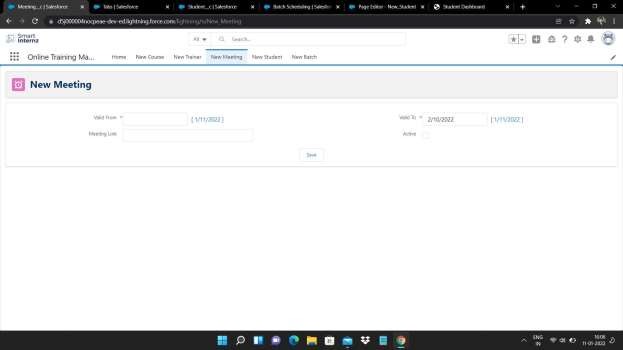
</apex:PageBlockSection>

<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="Batchc">

<apex:form >

<apex:sectionHeader title="New Batch"/>

<apex:pageBlock >

<apex:pageBlockSection >

<apex:inputField value="{!Batchc.Batch\_Typec}"/>

<apex:inputField value="{!Batchc.Start\_Datec}"/>

<apex:inputField value="{!Batchc.End\_Datec}"/>

<apex:inputField value="{!Batchc.Course\_IDc}"/>

<apex:inputField value="{!Batchc.Meeting\_IDc}"/>

<apex:inputField value="{!Batchc.Trainer\_IDc}"/>

<apex:inputField value="{!Batchc.Statusc}"/>

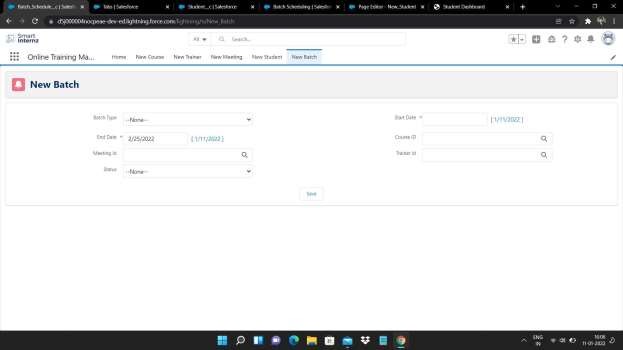
</apex:pageBlockSection>

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

</apex:pageBlock>

</apex:form>

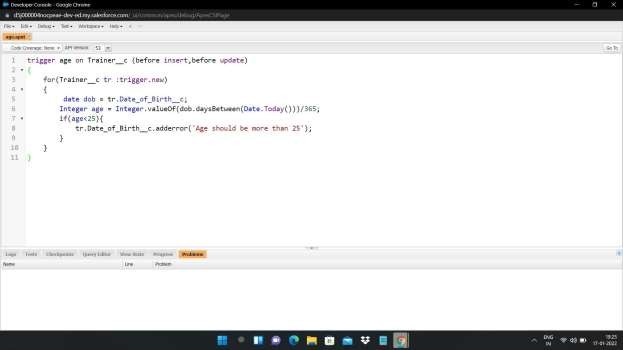
</apex:page>

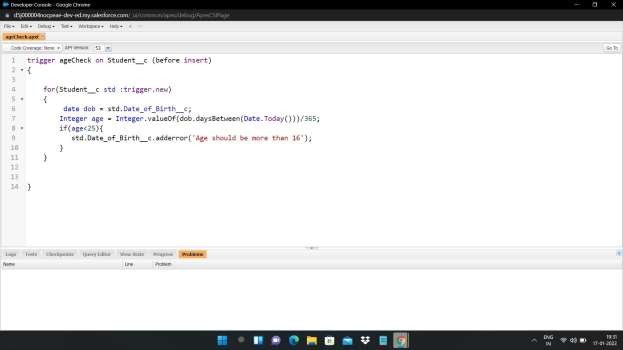


# Triggers:-

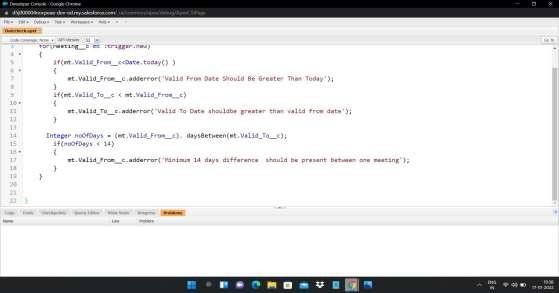
#### A **trigger is** an **Apex script** that executes before or after data manipulation language (**DML**) events occur. Apex triggers enable you to perform custom actions before or after events to record in Salesforce, such as insertions, updates, or deletions. Just like database systems support triggers,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

Trigger On Trainer Object:-



Trigger On Student Object:-

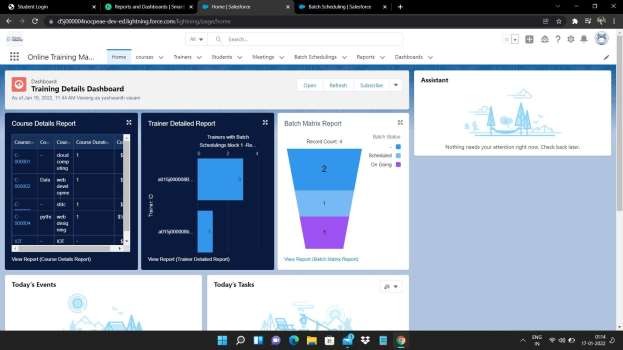
Trigger On Meeting Object:-



* **DashBoards withReports:-**

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.

Submitted by: ABINASH PUHAN