



Smart Internz

INTERNZ SMART _____

SALESFORCE PROJECT READY - ONLINE TRAINING MANAGEMENT DEVELOPER.

BY SANATH DM.

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 type to AutoNumber While creating course object changing recordType to course id)	Format(C-000001) Starting number=1
Course name	Text	
Course Duration	Picklist	1month,2months,3months
Course Fees	Currency	
Course Description	Text(Long)	

Course Category	Picklist	Data science, cloudcomputing, AI, ML
Course Trainer	Master-Detail(Trainer)	

Trainer Object:-

We need to add these fields in the object.

Field	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	Type	Values
Student	Autonumber	Format=S-00001,Starting Number=1

First Name	Text	
Last Name	Text	
Email	Email	
PhoneNumber	Phone	
ate Of Birth	Date	
Whatsapp	Number Phone	
Enrollment Student Enrollment	Date 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:-

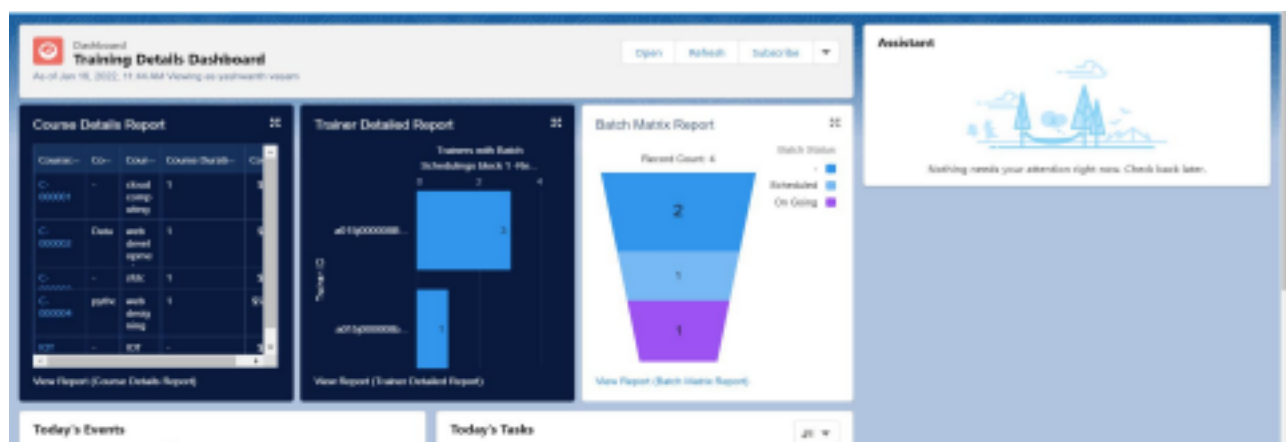
Field	Name Data	Type Values
Meeting	ID Autonumber	Format=M-00001, Starting N
Valid	From	Date
Valid	To Date	

Active	Checkbox	
Meeting	Link URL	

The data model of the above Customobjects is,



The Application Home Tab:



> 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 object.

> For the Course object :

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.

Course Validation Rule

[Back to Course](#)[Help for this Page](#)

Validation Rule Detail

EditClone

Rule Name	Course_fee_should_be_less_than_3000	Active	✓
Error Condition Formula	Course_Fees__c >=3000	Error Location	Course Fees
Error Message	Course Fees Should Be LESS than 3000\$		
Description			
Created By	Sanath DM, 1/6/2022, 7:35 AM	Modified By	Sanath DM, 1/6/2022, 7:35 AM

EditClone

> For the Trainer object :

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.

Trainer Validation Rule

[Back to Trainer](#)

Validation Rule Detail

EditClone

Rule Name	Specified_fields_should_not_be_blank	Active	<input checked="" type="checkbox"/>
Error Condition Formula	OR(ISBLANK(DateofBirth__c), ISBLANK(Full_Name__c), ISBLANK(PhoneNumber__c), ISBLANK(TEXT(Experience__c)))		
Error Message	Specified fields should not be null	Error Location	Top of Page
Description			
Created By	Sanath DM, 1/6/2022, 9:04 AM		Modified By Sanath DM, 1/6/2022, 9:12 AM

EditClone

Trainer Validation Rule

[Back to Trainer](#)

Validation Rule Detail

EditClone

Rule Name	Age_Should_Not_Be_Less_than_25_years	Active	<input checked="" type="checkbox"/>
Error Condition Formula	Age__c <25		
Error Message	age__c<25	Error Location	Top of Page
Description			
Created By	Sanath DM, 1/6/2022, 9:21 AM		Modified By Sanath DM, 1/6/2022, 9:21 AM

EditClone

[Help](#)

Trainer Validation Rule

[Back to Trainer](#)
[Help for this Page](#)

Validation Rule Detail

Edit

Clone

Rule Name	Phone_Number_and_Whatsapp_Number_should	Active	✓
Error Condition Formula	OR(NOT(REGEX(PhoneNumber__c , "[0-9]{10,15}")),NOT(REGEX(Whatsapp_Number__c , "[0-9]{10,15}")))		
Error Message	Phone_Number_and_Whatsapp_Number_should	Error Location	Top of Page
Description			
Created By	Sanath DM, 1/6/2022, 9:33 AM	Modified By	Sanath DM, 1/6/2022, 9:33 AM

Edit

Clone

> For the Student object:

1. Last name should not be Blank: enter the following error condition formula "ISBLANK(Fieldname)" 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.

Student Validation Rule

[Back to Student](#)

Validation Rule Detail

Edit

Clone

Rule Name	Lastname_should_not_be_Blank	Active	✓
Error Condition Formula	ISBLANK(Last_Name__c)		
Error Message	Lastname should not be Blank	Error Location	Last Name
Description			
Created By	Sanath DM, 1/6/2022, 9:51 AM	Modified By	Sanath DM, 1/6/2022, 9:51 AM

Edit

Clone

Student Validation Rule

[Back to Student](#)

Validation Rule Detail Edit Clone

Rule Name	Minimum_16_years_of_age	Active	<input checked="" type="checkbox"/>
Error Condition Formula	Student_Enrollment__r.Trainer_id__r.Age__c <=16		
Error Message	Minimum_16_years_of_age	Error Location	Top of Page
Description			
Created By	Sanath DM , 1/6/2022, 9:53 AM	Modified By	Sanath DM , 1/6/2022, 9:53 AM

Edit Clone

Student Validation Rule

[Back to Student](#)

Validation Rule Detail Edit Clone

Rule Name	Ph_and_whatapp_no_should_be_min_10_dig	Active	<input checked="" type="checkbox"/>
Error Condition Formula	OR(NOT(REGEX(PhoneNumber__c , "[0-9]{10,15}")),NOT(REGEX(Whatsapp_Number__c , "[0-9]{10}")))		
Error Message	Ph_and_whatapp_no_should_be_min_10_dig	Error Location	Top of Page
Description			
Created By	Sanath DM , 1/6/2022, 10:11 AM	Modified By	Sanath DM , 1/6/2022, 10:11 AM

Edit Clone

> For the Meeting object :

1. Valid From should not be less than current date: Click on validation rules from meetingobject, 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.

Meeting Validation Rule

[Back to Meeting](#)

Validation Rule Detail

[Edit](#) [Clone](#)

Rule Name	Valid_From_should_not_less_than_current	Active	✓
Error Condition Formula	Valid_From__c < TODAY()		
Error Message	Valid_From_should_l<_less_than_current	Error Location	Valid From
Description			
Created By	Sanath DM , 1/6/2022, 10:13 AM	Modified By	Sanath DM , 1/6/2022, 10:13 AM

[Edit](#) [Clone](#)

Meeting Validation Rule

[Back to Meeting](#)

Validation Rule Detail

[Edit](#) [Clone](#)

Rule Name	Valid_From_should_not_less_than_current	Active	✓
Error Condition Formula	Valid_From__c < TODAY()		
Error Message	Valid_From_should_l<_less_than_current	Error Location	Valid From
Description			
Created By	Sanath DM , 1/6/2022, 10:13 AM	Modified By	Sanath DM , 1/6/2022, 10:13 AM

[Edit](#) [Clone](#)

Meeting Validation Rule

[Back to Meeting](#)

Validation Rule Detail

[Edit](#) [Clone](#)

Rule Name	min_14_diff_btwn_valid_from_and_valid_to	Active	✓
Error Condition Formula	Valid_To__c - Valid_From__c < 14		
Error Message	14_diff_btwn_valid_from_and_valid_to	Error Location	Top of Page
Description			
Created By	Sanath DM , 1/6/2022, 10:16 AM	Modified By	Sanath DM , 1/8/2022, 10:53 AM

[Edit](#) [Clone](#)

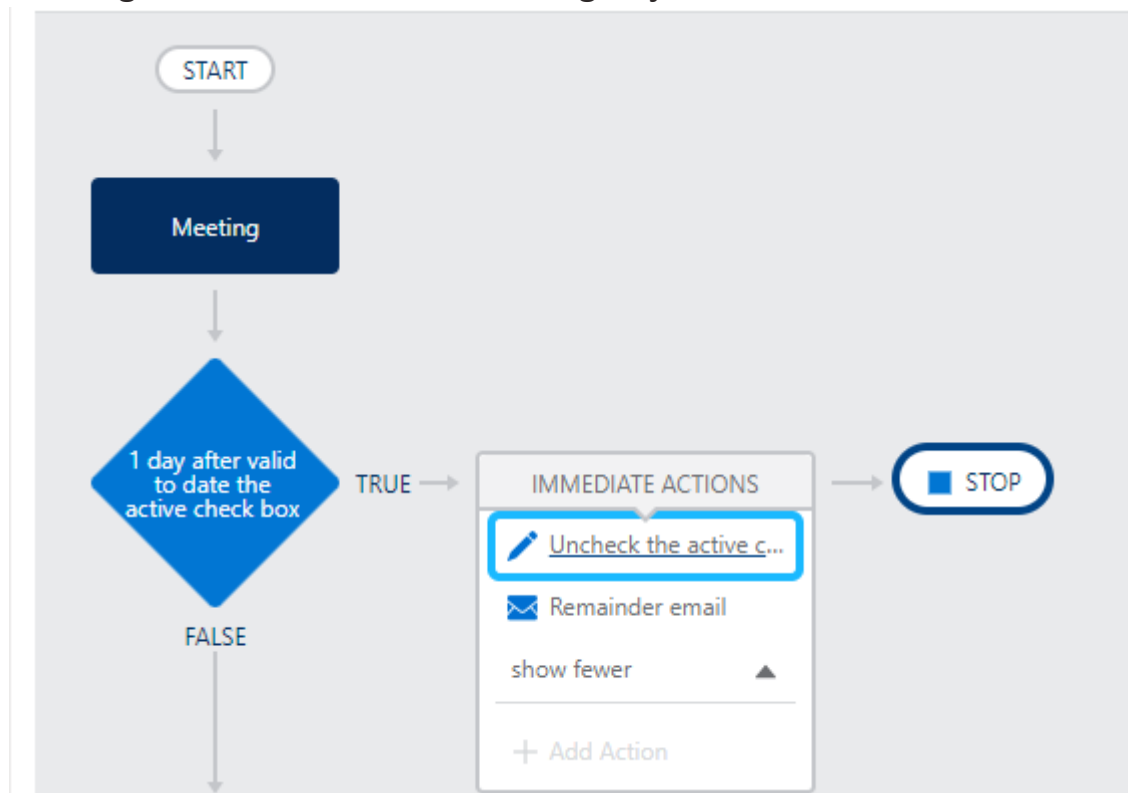
> Process Automation:-

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.

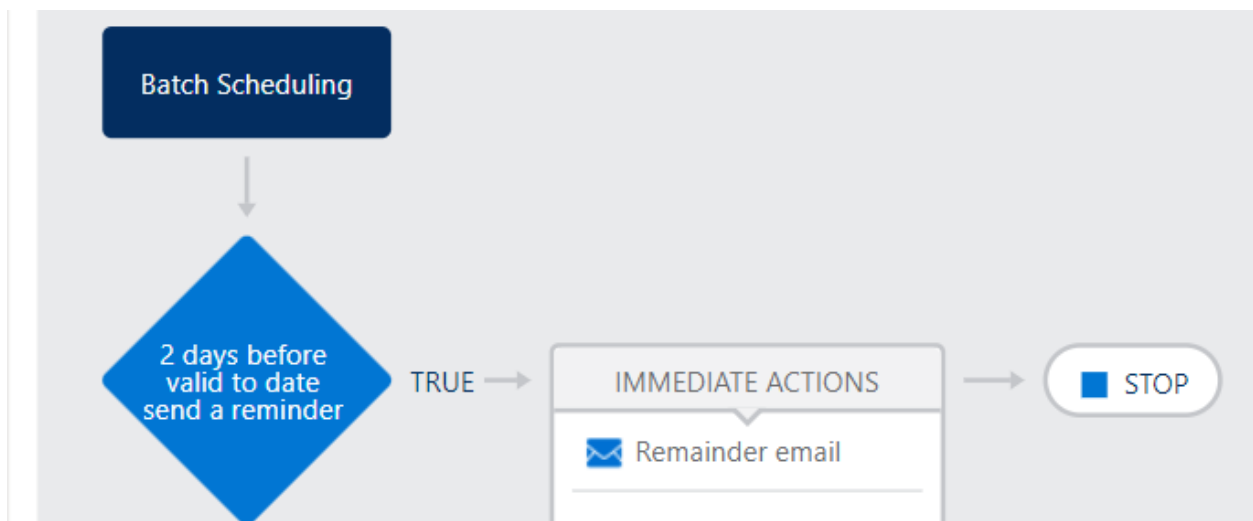
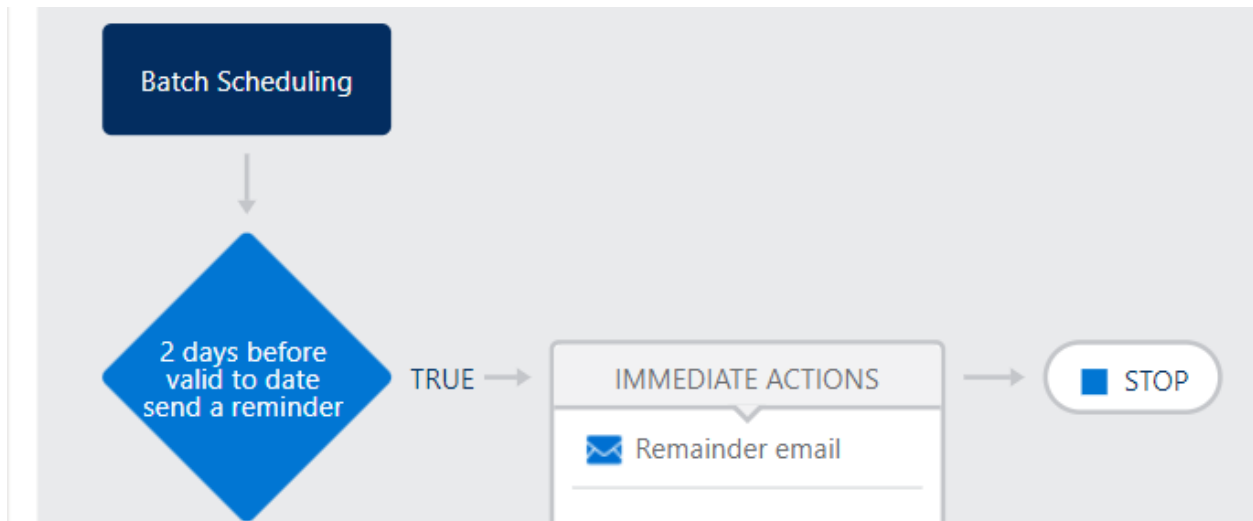
There are four automation tools in Salesforce:

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

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 Successsfully'));
            cr=new Course__c();
        }
    }
}
```

The screenshot shows a 'New Course' form. At the top is a header 'New Course'. Below it is a sub-header 'New Information'. The form contains several input fields: 'Course Name' (text), 'Course Category' (dropdown menu), 'Course Fee' (text), 'Course Duration (in Months)' (text), and 'Course Full Name' (text). Below these fields is a section titled 'Course Description' which contains a rich text editor with various formatting options like bold, italic, underline, and link.

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 InsertedSuccessfully'));
tr=new Trainer__c();

}

}

```

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>

```


New Student

First Name *
Sanath

Last Name *
DM

Email *
sanathdm81@gmail.com

PhoneNumber *
07676047839

Date Of Birth *
1/2/2002 [1/9/2022]

Whatsapp Number *
07676047839

Save

Students

<apex:form>

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

<apex:pageBlock>

<apex:pageBlockSection>

<apex:inputField value="{!Student__c.First_Name__c}"/>

<apex:inputField value="{!Student__c.Last_Name__c}" required="true"/>

<apex:inputField value="{!Student__c.Email__c}"/>

<apex:inputField value="{!Student__c.PhoneNumber__c}"/>

<apex:inputField value="{!Student__c.Date_Of_Birth__c}"/>

<apex:inputField value="{!Student__c.Whatsapp_Number__c}"/>

</apex:pageBlockSection>

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

</apex:pageBlock>

</apex:form>

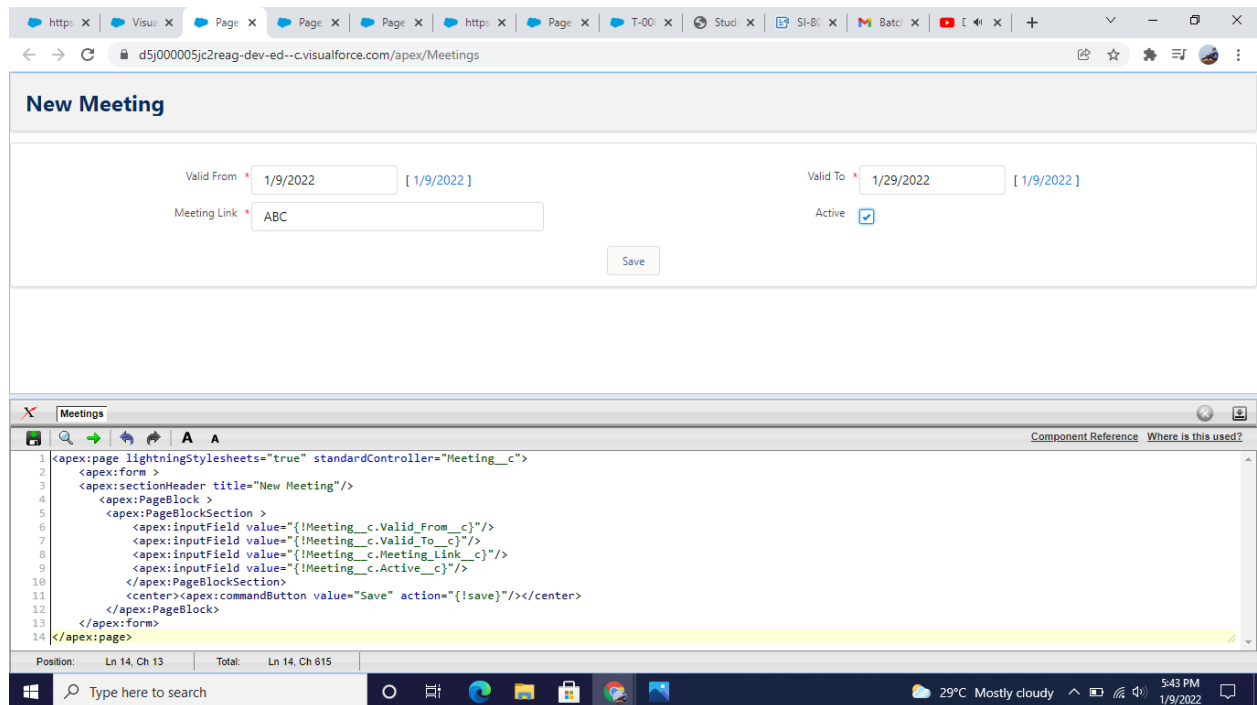
</apex:page>

Component Reference Where is this used?

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>

```

New Batch

Batch Type * Weekday

Start Date * 1/9/2022 [1/9/2022]

End Date * 1/29/2022 [1/9/2022]

Course Id * 1234

Meeting id * ABC

Trainer id * 1111

Status Scheduled

Save

Batch

```
3 <apex:form>
4   <apex:sectionHeader title="New Batch"/>
5   <apex:pageBlock>
6     <apex:pageBlockSection>
7       <apex:inputField value="{!Class_Scheduling__c.Batch_Type__c}"/>
8       <apex:inputField value="{!Class_Scheduling__c.Start_Date__c}"/>
9       <apex:inputField value="{!Class_Scheduling__c.End_Date__c}"/>
10      <apex:inputField value="{!Class_Scheduling__c.Course_Id__c}"/>
11      <apex:inputField value="{!Class_Scheduling__c.Meeting_id__c}"/>
12      <apex:inputField value="{!Class_Scheduling__c.Trainer_id__c}"/>
13      <apex:inputField value="{!Class_Scheduling__c.Status__c}"/>
14    </apex:pageBlockSection>
15    <center><apex:commandButton value="Save" action="{!save}"/></center>
16  </apex:pageBlock>
17</apex: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 system support triggers, Apex provides trigger support for managing records.

Trigger On Trainer Object:-

```
Apex Trigger  Version Settings  Trace Flags

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 }
```

Trigger On Student Object:-

```
Apex Trigger  Version Settings  Trace Flags

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<25){
9             std.Date_of_Birth__c.adderror('Age should be more than 16');
10        }
11    }
12 }
13 }
14 }
```

Trigger On Meeting Object:-

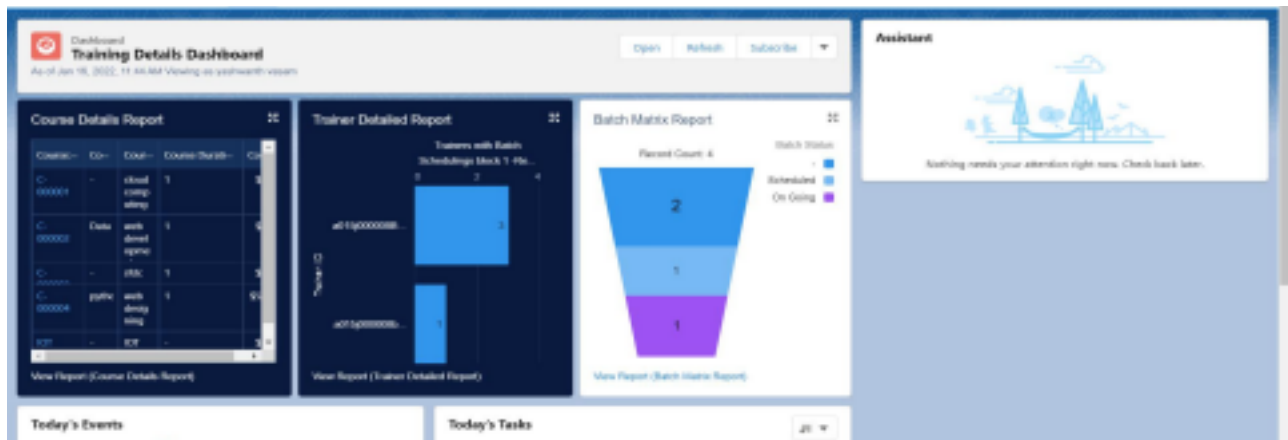
```
Apex Trigger    Version Settings    Trace Flags
1 trigger Datecheck on Meeting__c (before insert)
2 {
3     for(Meeting__c mt :trigger.new)
4     {
5         if(mt.Valid_From__c < Date.today() )
6         {
7             mt.Valid_From__c.adderror('Valid From Date Should Be Greater Than Today');
8         }
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 }
```

> 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 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 real time experience in how to work in a company organisation.

Thank you.