

SALESFORCE PROJECT READY

ENTRY LEVEL TRAINING AND PROFESSIONAL DEVELOPMENT PROGRAM

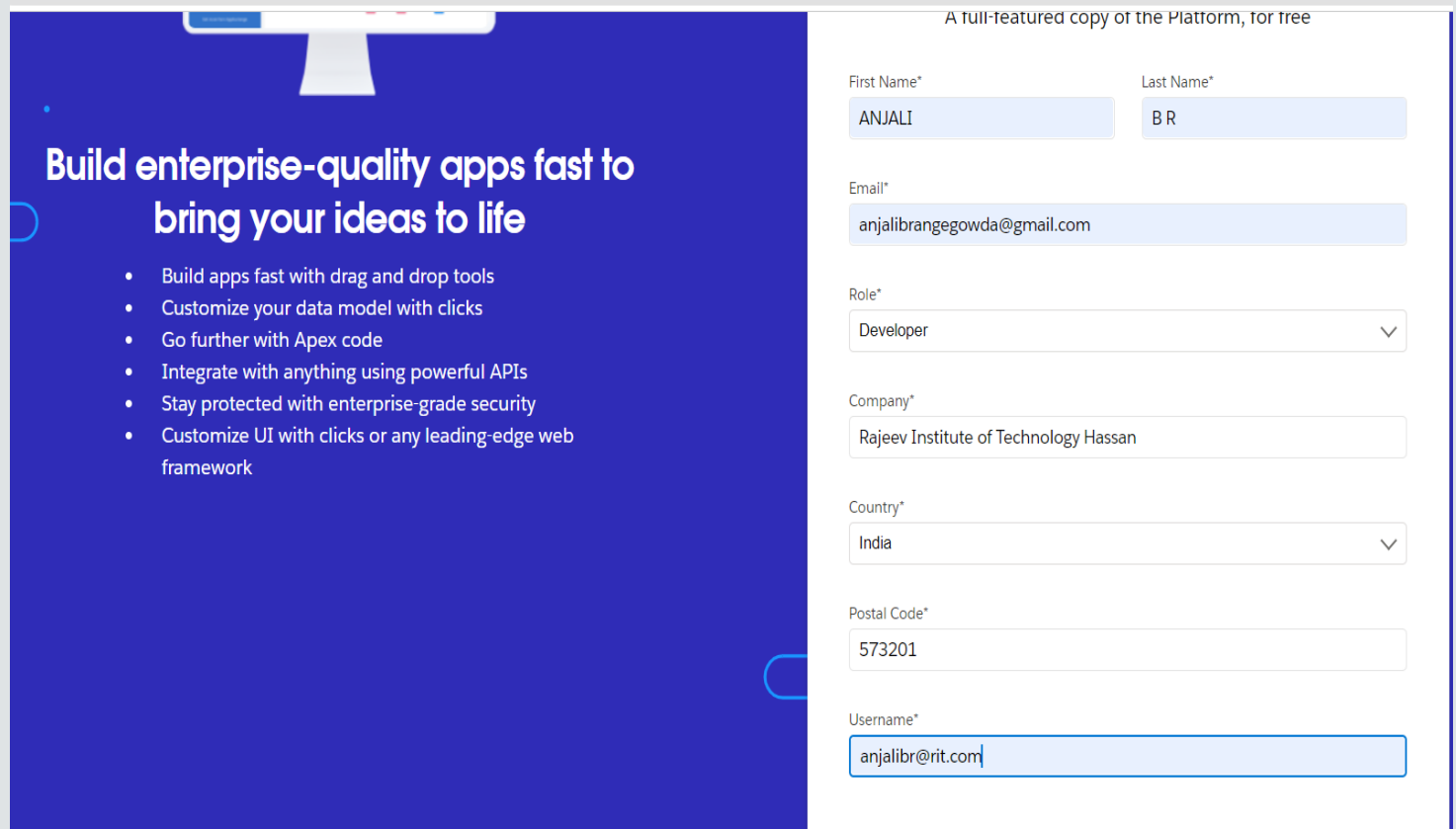
DAY 1: **Salesforce Overview**

DATE:03/01/2022

TOPIC: Introduction to Salesforce and Creation of Salesforce Developer Org

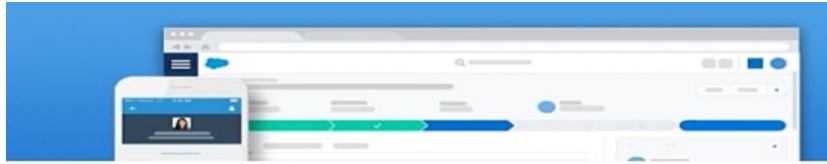
Salesforce is a customer success platform, designed to help on sell, service, market, analyze and connect with your customers. Salesforce has everything we need to run our business from anywhere using standard products and features, we can manage relationship with prospects and customers, collaboration, and with employees and partners, and store our data securely in the cloud.

ACTIVITY 1: Creating the Salesforce Developer Account



The image shows a screenshot of the Salesforce Developer Org creation form. On the left, there is a blue sidebar with the text 'Build enterprise-quality apps fast to bring your ideas to life' and a list of features: 'Build apps fast with drag and drop tools', 'Customize your data model with clicks', 'Go further with Apex code', 'Integrate with anything using powerful APIs', 'Stay protected with enterprise-grade security', and 'Customize UI with clicks or any leading-edge web framework'. On the right, there is a white form titled 'A full-featured copy of the Platform, for free'. The form contains several input fields: 'First Name*' (ANJALI), 'Last Name*' (B R), 'Email*' (anjaliibrangegowda@gmail.com), 'Role*' (Developer), 'Company*' (Rajeev Institute of Technology Hassan), 'Country*' (India), 'Postal Code*' (573201), and 'Username*' (anjaliibr@rit.com).

2. Account Activation



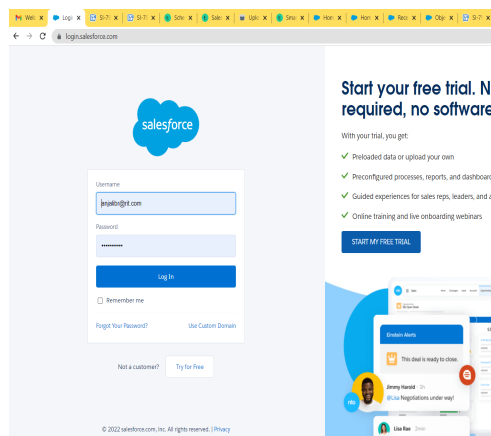
Click below to verify your account.

Verify Account

To easily log in later, save this URL:
<https://rajeevinstituteoftechnolo53-dev-ed.my.salesforce.com>

Username:
anjalibr@rit.com

3. Login to your salesforce account



Day -02 : Setting Up First App, Custom Object and Field Creation

Date:04/01/2022

Topic: Custom Object Creation

Activity 02 : Create the custom Objects For The Application

1. College
2. Application Form

3. Students
4. Object

01: Create Fields on College Object

Setup > OBJECT MANAGER > College

Fields & Relationships
9 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
College Email	College_Email__c	Picklist	College Name	
College Fees	College_Fees__c	Currency(7, 2)		
College Name	College_Name__c	Picklist		
Created By	CreatedById	Lookup(User)		
Hostel Fees	Hostel_Fees__c	Currency(6, 2)		
Last Modified By	LastModifiedById	Lookup(User)		
No. of Seats	No_of_Seats__c	Picklist	College Name	
Owner	OwnerId	Lookup(User/Group)		✓
Record Info	Name	Text(80)		✓

02: Create Fields on Application Form Object

Setup > OBJECT MANAGER > Application Form

Fields & Relationships
14 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text(255)		
Application Form ID	Application_Form_ID__c	Auto Number		
College	College__c	Master-Detail(College)		✓
College Fees	College_Fees__c	Formula (Currency)		
Created By	CreatedById	Lookup(User)		
Date of Birth	Date_of_Birth__c	Date		
Email	Email__c	Email (Unique)		✓
Form Number	Name	Auto Number		✓
Guardian Name	Guardian_Name__c	Text(30)		

03: Create Fields on Student Object

Setup Home Object Manager

SEARCH Setup

SETUP > OBJECT MANAGER
Student

Details

Fields & Relationships
10 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text(255)		
Application Form	Application_Form__c	Lookup(Application Form)		✓
College Name	College_Name__c	Formula (Text)		
Created By	CreatedById	Lookup(User)		
Date of Birth	Date_of_Birth__c	Date		
Guardian Name	Guardian_Name__c	Text(30)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone	Phone__c	Phone		
Student Name	Name	Text(80)		✓

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

04 : Create Fields on Subject Object

Setup Home Object Manager

SEARCH Setup

SETUP > OBJECT MANAGER
Subject

Details

Fields & Relationships
8 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Paper1	Paper1__c	Picklist		
Paper2	Paper2__c	Picklist		
Student	Student__c	Lookup(Student)		✓
Subject ID	Subject_ID__c	Auto Number		
Subject Name	Name	Text(80)		✓

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

Day: 03: Profiles Record Types and Automation

Date:05/01/2022

Topic :Adding Business Logic to Application

Business logic is the brain of the application.it makes sense of the data and response to the button clicks in the UI


Activity 03 :

01: Creating Global Picklist Value Sets

1. Create the following global picklist value sets for the application.

a)College

Picklist Value Name	Values
Colllege	MIT-HYD MIT-BLR MIT-MUM MIT-MAA MIT-DEL MIT-CCU



SETUP

Picklist Value Sets

Description

Picklist Values Used

Active and inactive picklist values

6 (1,000 max)

Edit

Delete

Values

New

Reorder

Replace

Printable View

Chart Colors

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	MIT-BLR	MIT-BLR	<input type="checkbox"/>	Assigned dynamically	ANJALI B.R. , 1/5/2022, 2:38 AM
Edit Del Deactivate	MIT-MAA	MIT-MAA	<input type="checkbox"/>	Assigned dynamically	ANJALI B.R. , 1/5/2022, 2:38 AM
Edit Del Deactivate	MIT-HYD	MIT-HYD	<input type="checkbox"/>	Assigned dynamically	ANJALI B.R. , 1/5/2022, 2:38 AM
Edit Del Deactivate	MIT-DEL	MIT-DEL	<input type="checkbox"/>	Assigned dynamically	ANJALI B.R. , 1/5/2022, 2:38 AM
Edit Del Deactivate	MIT-MUM	MIT-MUM	<input type="checkbox"/>	Assigned dynamically	ANJALI B.R. , 1/5/2022, 2:38 AM
Edit Del Deactivate	MIT-CCU	MIT-CCU	<input type="checkbox"/>	Assigned dynamically	ANJALI B.R. , 1/5/2022, 2:38 AM

Inactive Values


No Inactive Values values defined.

Fields Where Used

Field Label	Object	Data Type	Controlling Field
College Name	College	Picklist	

b)Paper1

Picklist Value Name	Values
Paper 1	APEX JAVA C C++



SETUP

Picklist Value Sets

Global Value Set Details

▼ Information

Label

Paper1

Name

Paper1

Description

Picklist Values Used

Active and inactive picklist values

4 (1,000 max)

Edit

Delete

Values

New

Reorder

Replace

Printable View

Chart Colors ▼

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	APEX	APEX	<input type="checkbox"/>	Assigned dynamically	ANJALI B R , 1/6/2022, 1:42 AM
Edit Del Deactivate	JAVA	JAVA	<input type="checkbox"/>	Assigned dynamically	ANJALI B R , 1/6/2022, 1:42 AM
Edit Del Deactivate	C	C	<input type="checkbox"/>	Assigned dynamically	ANJALI B R , 1/6/2022, 1:42 AM
Edit Del Deactivate	C++	C++	<input type="checkbox"/>	Assigned dynamically	ANJALI B R , 1/6/2022, 1:42 AM

Inactive Values


No Inactive Values values defined.

Fields Where Used

Field Label	Object	Data Type	Controlling Field
Paper1	Subject	Picklist	

c)Paper2

Picklist Value Name	Values
Paper2	MATHEMATICS ENGLISH STATISTICS


Picklist Value Sets

Information

Label

Paper2

Name

Paper2

Description

Picklist Values Used

Active and inactive picklist values 3 (1,000 max)

Edit

Delete

Values

New

Reorder

Replace

Printable View

Chart Colors

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	MATHAMATICS	MATHAMATICS	<input type="checkbox"/>	Assigned dynamically	ANJALI B.R. 1/6/2022, 1:44 AM
Edit Del Deactivate	STATISTICS	STATISTICS	<input type="checkbox"/>	Assigned dynamically	ANJALI B.R. 1/6/2022, 1:44 AM
Edit Del Deactivate	ENGLISH	ENGLISH	<input type="checkbox"/>	Assigned dynamically	ANJALI B.R. 1/6/2022, 1:44 AM


Inactive Values

No Inactive Values values defined.

Fields Where Used

Field Label	Object	Data Type	Controlling Field
Paper2	Subject	Picklist	

- ## 2. Create Field Dependencies:
1. Create field dependency between college Name and Email, where the controlling field is college Name and dependent field is Email. Select the email ids according to the college names.
 2. Create field dependency between college Name and capacity of students, where the controlling field is college Name and dependent field is Capacity of Students. Select the values according to your wish.



All

Search Colleges and more...

★

+

🔔

?

⚙️

🔔

👤

College management

Colleges

Application Forms

Students

Subjects

Colleges

All

New

Import

Change Owner

Printable View

6 items • Sorted by Record Info • Filtered by All colleges • Updated a few seconds ago

Search this list...

⚙️

🔍

🔄

✏️

🗑️

⌵

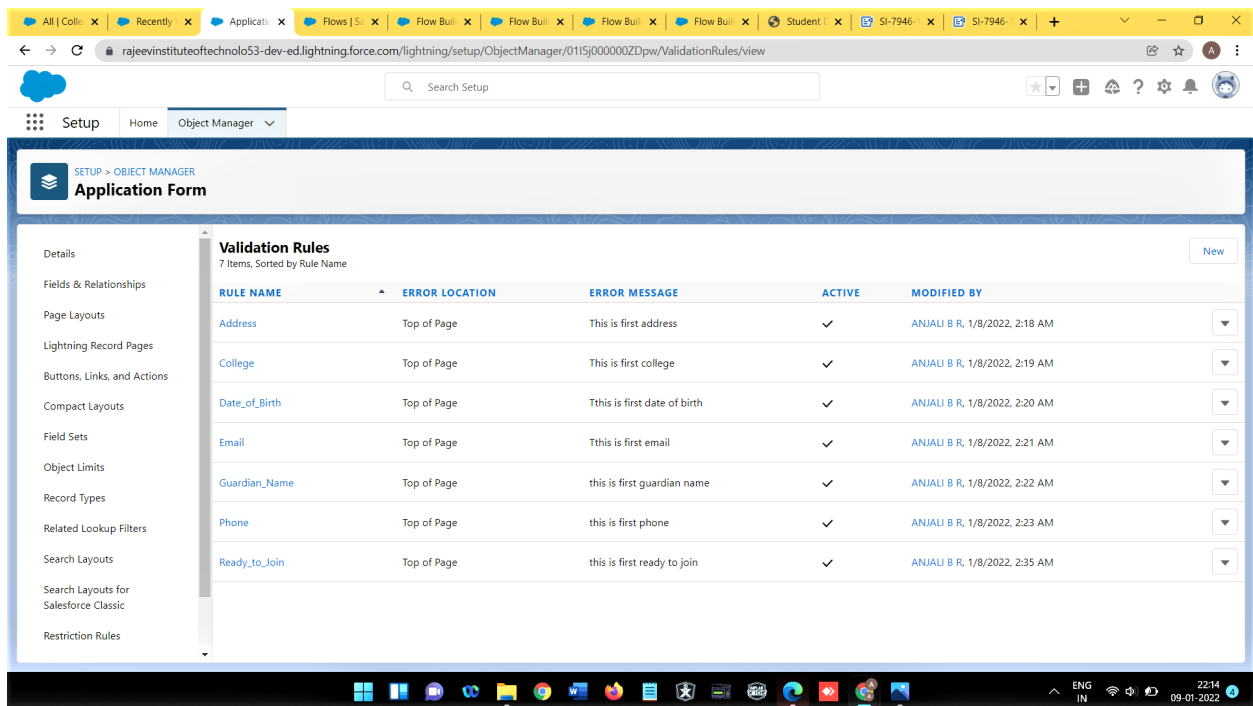
<input type="checkbox"/>	Record Info ↑	College Fees	College Name	Hostel Fees	College Email
1	<input type="checkbox"/> MIT-BLR	\$1,400,000.00	MIT-BLR	\$500,000.00	mit@blr.com
2	<input type="checkbox"/> MIT-CCU	\$1,100,000.00	MIT-CCU	\$150,000.00	
3	<input type="checkbox"/> MIT-DEL	\$1,100,000.00	MIT-DEL	\$250,000.00	mit@del.com
4	<input type="checkbox"/> MIT-HYD	\$1,000,000.00	MIT-HYD	\$150,000.00	mit@hyd.com
5	<input type="checkbox"/> MIT-MAA	\$100,000.00	MIT-MAA	\$200,000.00	mit@maa.com
6	<input type="checkbox"/> MIT-MUM	\$1,500,000.00	MIT-MUM	\$400,000.00	mit@mum.com

- ## 3. Creating Validation Rules :
1. Create a validation rule on the college object such that the college name and record info should have the same name.
TEXT(College_Name__c) <> Name.
 2. Create a validation rule on the application form object to stop any modification on the application form once a student record is created.

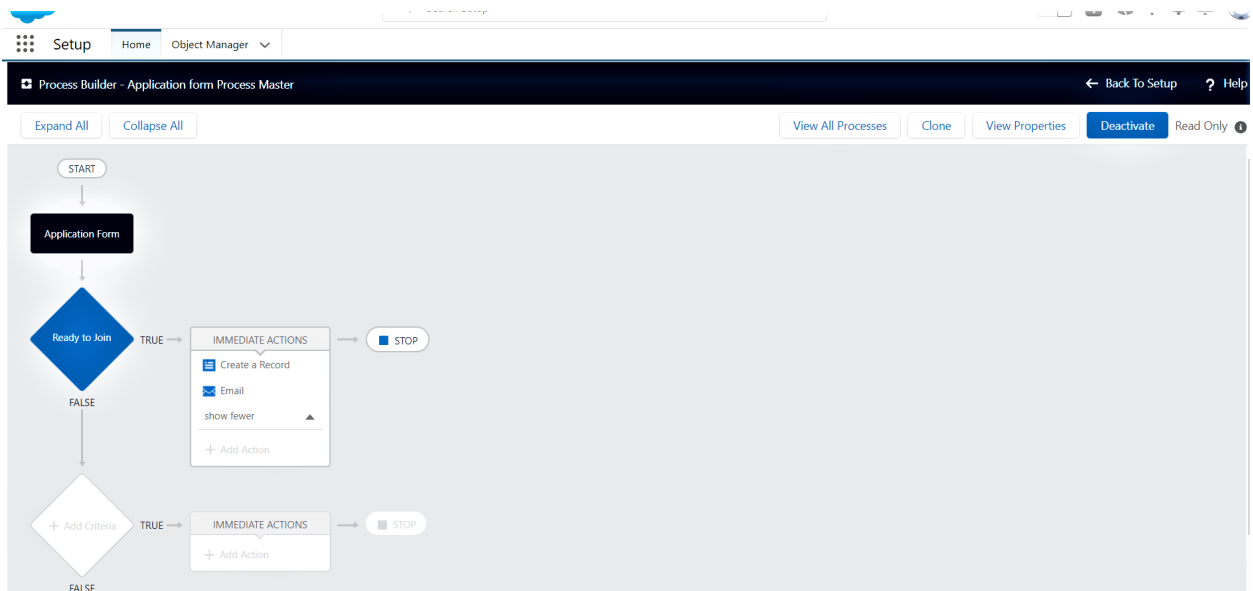

```

AND( Ready_To_Join__c == true,
OR(
ISCHANGED( Address__c ) ,
ISCHANGED( College__c ) ,
ISCHANGED( Date_Of_Birth__c),
ISCHANGED( Email__c ) ,
ISCHANGED( Guardian_Name__c ) ,
ISCHANGED( Phone__c )
)
)
)

```



- 4.Process Automation :**
1. Create an automation process such that when the "ready to join" field is checked on the application form object we need to create the student record automatically with the information specified in the application form record.
 2. Go to Setup -> select "Process Builder" from quick find. Create a Process Builder on the "Application Form" object with a condition as "When a record change". And select "When a record is created or edited".
 - a. In the diamond shape box(called nodes), select the criteria which trigger the Process builder to fire. In our example, it is "When Ready_to_Join field is checked."
 - b. Once the node is setup, click on the adjacent box called "Immediate action". And select create a record on the student object. Please follow the below screenshot.



5.Create The Student Record Using Flow :1. First deactivate the process builder which we created earlier.

2. Now search for flows and select new flow ->record triggered flow

3. For object select application form, in configure trigger select when the record is updated for entry criteria select as ready to join equals to true.

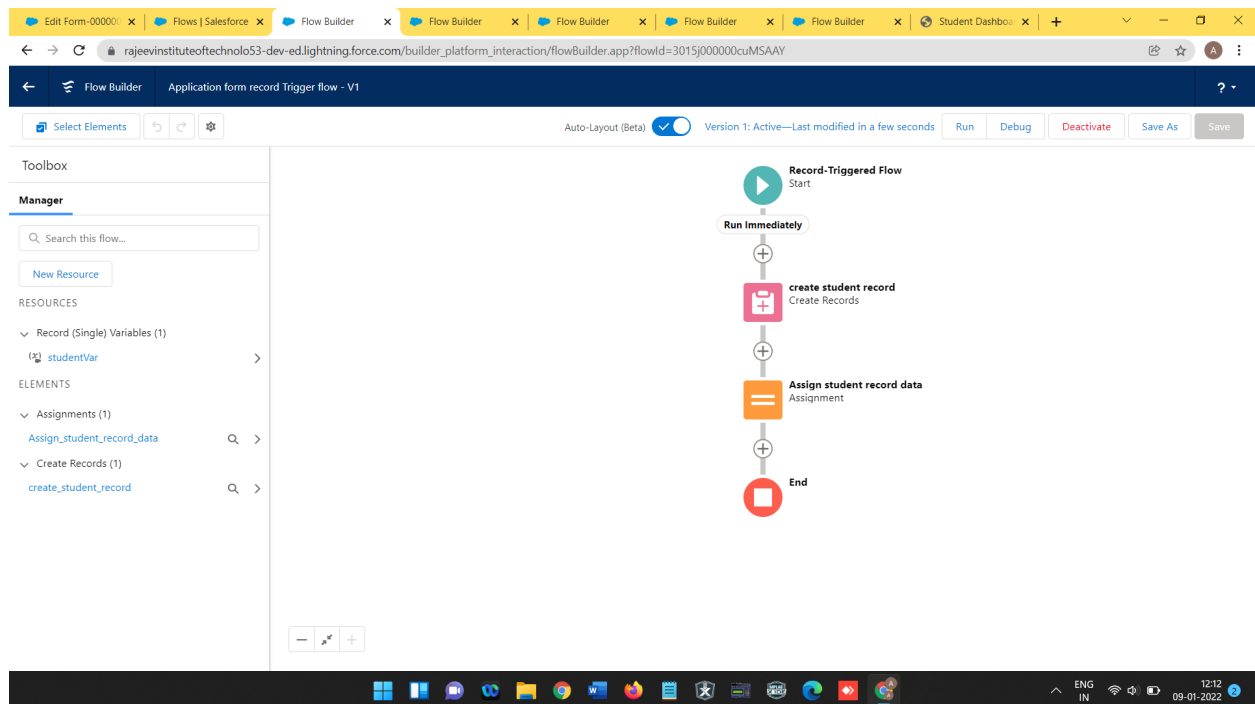
4. Now create a variable named student in the resource section.

5. Now add the assignment as follows.

The 'New Assignment' dialog box is shown, listing variable assignments for the 'studentVar' resource. The assignments are as follows:

Variable	Operator	Value
studentVar > Address	Equals	\$Record > Address
studentVar > Application Form	Equals	\$Record > Record ID
studentVar > College Name	Equals	\$Record > College
studentVar > Date of Birth	Equals	\$Record > Date of Birth
studentVar > Guardian Name	Equals	\$Record > Guardian Name
studentVar > Phone	Equals	\$Record > phone
studentVar > Student Name	Equals	\$Record > Student Name

The dialog box includes a 'Cancel' button and a 'Done' button at the bottom right.



DAY-4: Automation, Security and Sharing ,Overview of Apex

DATE : 06/01/2022

Activity-04

Security and Sharing :

- Security Model** :
1. Object level security
 2. Field level security
 3. Record level security

Record level security : A record that we are creating have to be private
profile :Test Profile

Setup

Home

Object Manager

Q profile

Users

Profiles

Didn't find what you're looking for?

Try using Global Search.

SETUP

Profiles

Profile

Test profile

Help for this profile

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Login IP Ranges (0) | Enabled Apex Class Access (0) | Enabled Visualforce Page Access (0) | Enabled External Data Source Access (0) | Enabled Named Credential Access (0) | Enabled Custom Metadata Type Access (0) | Enabled Custom Setting Definitions Access (0) | Enabled Flow Access (0) | Enabled Service Presence Status Access (0) | Enabled Custom Permissions (0)

Profile Detail

EditCloneDeleteView Users

Name	Test profile				
User License	Salesforce	Custom Profile	✓		
Description					
Created By	ANJALI B R	1/6/2022, 2:36 AM	Modified By	ANJALI B R	1/7/2022, 10:28 PM

Page Layouts

Standard Object Layouts

Global	Global Layout [View Assignment]	Legal Entity	Legal Entity Layout [View Assignment]
Email Application	Not Assigned [View Assignment]	Macro	Macro Layout [View Assignment]
Home Page Layout	DE Default [View Assignment]	Metric	Varies by Record Type [View Assignment]
Account	Varies by Record Type [View Assignment]	Metric Data Link	Metric Data Link Layout [View Assignment]
Alternative Payment Method	Alternative Payment Method Layout [View Assignment]	Object Milestone	Object Milestone Layout [View Assignment]
Asset	Asset Layout [View Assignment]	Operating Hours	Operating Hours Layout [View Assignment]

SETUP

Profiles

Article Type Layouts

Field-Level Security

Standard Field-Level Security

Account	[View]	Legal Entity	[View]
Alternative Payment Method	[View]	Location	[View]
API Anomaly Event Store	[View]	Location Group	[View]
Appointment Topic Time Slot	[View]	Location Group Assignment	[View]
Asset	[View]	Macro	[View]
Asset Action	[View]	Messaging Session	[View]
Asset Action Source	[View]	Messaging User	[View]
Asset Relationship	[View]	Metric	[View]
Asset State Period	[View]	Object Milestone	[View]
Assigned Resource	[View]	Operating Hours	[View]
Associated Location	[View]	Opportunity	[View]
Authorization Form	[View]	Opportunity Contact Role	[View]
Authorization Form Consent	[View]	Opportunity Product	[View]
Authorization Form Data Use	[View]	Order	[View]
Authorization Form Text	[View]	Order Product	[View]
Business Brand	[View]	Party Consent	[View]
Campaign	[View]	Payment	[View]
Campaign Member	[View]	Payment Authorization	[View]
Card Payment Method	[View]	Payment Authorization Adjustment	[View]
Case	[View]	Payment Gateway	[View]
Case Related Issue	[View]	Payment Group	[View]
Change Request	[View]	Payment Line Invoice	[View]
Change Request Related Issue	[View]	Performance Cycle	[View]

anolo53-dev-ed.lightning.force.com/lightning/setup/null/page?address=%2Fsetup%2Flayout%2Ffldsdetail.jsp%3Fid%3D00e5j000000hNBi%26type%3D0115j000000ZDpw%26retURL%...

Search Setup

anager

SETUP

Application Form Field-Level Security for profile

Test profile

Edit Back to Profile

Field Name	Field Type	Read Access	Edit Access
Address	Text	✓	✓
Application Form ID	Auto Number	✓	✗
College	Lookup	✓	✓
College Fees	Currency	✓	✗
Created By	Lookup	✓	✗
Date of Birth	Date	✓	✓
Email	Email	✓	✓
Form Number	Auto Number	✓	✗
Guardian Name	Text	✓	✓
Hostel fees	Currency	✓	✗
Last Modified By	Lookup	✓	✗
Looking For Hostel Stay	Checkbox	✓	✓
phone	Phone	✓	✓
Ready to Join	Text	✓	✓
Student Name	Text	✓	✓

Edit Back to Profile

2Ffldsdetail.jsp%3Fid%3D00e5j000000hNBi%26type%3D0115j000000ZDpw%26retURL%3D%252F00e5j000000hNBi%253FappLayout%253Dsetup%2526tour%253D%2526isdtp%253Dp1%2526sfclFrameOrigin%253Dhttps%25253A%...

Sharing Rules : sharing rules open up when owd is private or public read only.

public groups makes a one particular group

Debug Statement : 1.output statement

2.whether the statement is true or false

Query Editor : 1. vs code

2. LWC editor there apex

debug only:

```
17:27:53:019 USER_DEBUG [4]|DEBUG| list of applicatin form : (Application_Form__c:
{Student_Name__c=ANJALI B R, Guardian_Name__c=anjali, Date_of_Birth__c=2001-04-20
00:00:00, Id=a015j000008I9OIAA0},
Application_Form__c:{Student_Name__c=Anjali B R, Guardian_Name__c=ANJU,
Date_of_Birth__c=2001-04-20 00:00:00, Id=a015j000008I9OHAA0},
Application_Form__c:{Student_Name__c=Roshan, Guardian_Name__c=sudha,
Date_of_Birth__c=2004-01-14 00:00:00, Id=a015j000008I9aeAAC},
Application_Form__c:{Student_Name__c=Chaitra, Guardian_Name__c=gagana, Da
```

//quering from child parent

```
select      Student_Name__c,      Guardian_Name__c,      Date_of_Birth__c,
College__r,College_Name__c from Application_Forms__c
```

// query from parent to child

```
select
id,College_Name__c,(selectId,Student_Name__c,Guardian_Name__c,Date_of_Birth__c
from Application_Forms__c )
from College__c
```

Day-05: Primitive Data Types and Collections

Date: 07/01/2022

Primitive Data Types: 1. Boolean

2. string

3. date

4. data time

5. id

6. integer

7. double

primitive data types - id, integer, double, string,
collection - list, set, map

SOQL - salesforce Object Query Language

```
[select FIELDS from soBject]  
select * from table
```

Collections: collection variable can be of primitive data types

1. List
2. set
3. Map

Collection variable can be of subject type of below lists:

1. List<Subject>
2. List<account>
3. List<Account>
4. List<contact>

1. List<Application_form__c> afList = [select Student_Name__c,
Guardian_Name__c, Date_of_Birth__c from Application_Form__c];

```
//debug statements  
for(Application_form__c afp : afList){  
    system.debug(afp);  
}
```

2. List<Account> accountList = new List<Account>();

// add method insert an record of type account into accountList


```
account aa1= new account(Name ='Ram',SLA__c
='Gold',SLASerialNumber__c='123' );
account aa2= new account(Name ='Shyam',SLA__c
='Gold',SLASerialNumber__c='123' );
//insert aa1; insert aa2; -DML Statement (insert,update,delete)
```

```
accountList.add(aa1);
accountList.add(aa2);
```

```
insert accountList;
for(Account aa:accountList){
    system.debug('inserted records'+ aa);
}
```

```
3. set<Integer> intSet=new Set<integer>(); //add elements . add()
intset.add(1);
intset.add(2);
intset.add(1);
```

```
System.debug(intset);
```

```
4. //Map,key,value pair=
```

```
Map<Integer,string> mm =new Map<Integer,string>();
// Put(k,v);
```

```
mm.put(1,'One');
mm.put(2,'Two');
mm.put(3,'Three');
```

```
System.debug('all values :'+mm);
```

```
System.debug('all key values means left side of the map :'+mm.keySet());  
System.debug('all right side values:'+mm.values());
```

DAY -06 : Overview of Triggers, Batch Apex

10/01/2022

Create a BatchApex for Application form : it segregates the total no of records

Three methods :

1. start = it gathers the data for us
2. execute = it will process the data
3. finish = end of the operation

batch class

```
{  
start(),execute(),finish()  
}
```

default batch size =200

maximum size = 2000

Application form : At the end of the business hours they are basically want to trigger an email which will number of total application form ; number of application got converted to student.

It is a kind of a interface that is used number of records in process.

```
Public class ApplicationBatchTest implements Database.Batchable<sObject>{
```

```
//start(), execute(). finish()
```

```
public Integer totalForms = 0; // total no of application form
```

```

public Integer totalConvertedForms = 0; // total no of students
public Database.QueryLocator start(Database.BatchableContext bc){
    // gathers the data for you
    String applicationQuery = 'select id, Name, Ready_To_Join__c from
ApplicationForm__c';
    return Database.getQueryLocator(applicationQuery);
}
public void execute(Database.BatchableContext bc, List<ApplicationForm__c>
formList){
    // process the data
    for(ApplicationForm__c af : formList){
        totalForms++;
        if(af.Ready_To_Join__c){
            totalConvertedForms++;
        }
    }
}
public void finish(Database.BatchableContext bc){
    // emails ,
    Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
    // address, subject, content( data to sent to admins)
    mail.setSubject(' Application form and student record data as of today ');
    mail.setPlainTextBody(' Total no of application form records are : '+totalForms+ '
out of which no of students as per today : '+totalConvertedForms);
    String[] emailAddress = new String[]{ 'your email address' };
    mail.setToAddresses(emailAddress);
    Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail } );
}
}

ApplicationBatchTest abt = new ApplicationBatchTest();
Database.executeBatch(abt, 400); // 200 to 2000
string crn='20 10 23 1 1';
System.Schedulable('cronjob1,crn,new ApplicationSchedule ');

```

Apex Trigger : custom logic is that you worked to perform when you insert or update or delete or undelete

Order of execution :

1. Validation rule
2. Custom rule
3. Record triggered flow before trigger
4. after trigger
5. assignments
6. autom duplication rule
7. work flow
8. process or invokable flow

Trigger context variables :

1. trigger.new
2. trigger.old
3. trigger.newmap
4. trigger.oldmap

1. From the developer console create a new apex class and enter the following code.

```
public class applicationschedule implements Schedulable{

    public void execute(SchedulableContext sc){
        ApplicationBatchTest abt = new ApplicationBatchTest();
        Database.executeBatch(abt, 400); // 200 to 2000

    }

}

// database.executeBatch(instance of batch class,size of batch);

ApplicationBatchTest abt = new ApplicationBatchTest();
```

```
Database.executeBatch(abt, 400); // 200 to 2000
//secs,mins,hrs,day_of_moht,moth,day_of_week,year(option)
string crn='20 10 23 1 1';
System.Schedulable('cronjob1,crn,new ApplicationSchedule()');
```

Day - 08 : LWC Overview

12/01/2022

1.LWC is a new programming model to develop salesforce lightning components. it is a ui framework that is built using native html and modern java script . it uses core web components standard ,custom elements,templater,decorator,modules,shadow dam

Here we are going to create the college data table, for this we need to create an apex class, from which we are going to retrieve the data and Html, javascript files for UI .

Create an Apex Class With The Required Functionality

```
public class GetapplicationDetails
{
    @AuraEnabled(cacheable=true)
    public static List<ApplicationForm__c> getapplicationvalues(id CollegeId)
    {

        List<ApplicationForm__c> formlist = [SELECT ID, College_Fees__c,Name,
        Date_Of_Birth__c,Email__c,Hostel_Fees__c, Student_Name__c FROM
        ApplicationForm__c WHERE College__c =:CollegeId];

        return formList;
    }
}
```

DAY - 09 : LWC

12/01/2022

Create college data table components(HTML file) :

Create the component with the following files.

HTML:

```
<template>
```

```
    <h1> College and Application form list table </h1>
```

```
    <template if:true={recordList}>
```

```
        <lightning-datatable
```

```
            key-field="id"
```

```
            data={recordList}
```

```
            show-row-number-column
```

```
            hide-checkbox-column
```

```
            columns={columnsList}
```

```
        >
```

```
    </lightning-datatable>
```

```
</template>
```

```
    <template if:true={error}>
```

```
        {error}
```

```
    </template>
```

```
</template>
```

DAY -10 : Component Creation For Application

Date :13/01/2022

Create college data table components (Java script file) :

After the creation of the Html file create the following js file

```
import { LightningElement, api, wire } from 'lwc';
import getapplicationvalues
from '@salesforce/apex/GetapplicationDetails.getapplicationvalues';
export default class CollegeDataTable extends LightningElement {
  columnsList = [
    {label : 'Application Form' , fieldName : 'Name', type:'text' },
    {label : 'College Fees' , fieldName : 'College_Fees__c', type:'currency' },
    {label : 'Date Of Birth' , fieldName : 'Date_Of_Birth__c', type:'date' },
    {label : 'Email' , fieldName : 'Email__c', type:'email' },
    {label : 'Hostel Fees' , fieldName : 'Hostel_Fees__c', type:'Currency' },
    {label : 'StudentName' , fieldName : 'Student_Name__c', type:'text' }
  ];

  @api recordId;
  recordList;
  error;

  @wire(getapplicationvalues, {CollegelId : '$recordId'})
  wiredCollegeData({data, error}){
    if(data){
      this.recordList = data;
    }
    else if(error){
      this.error = error;
      this.recordList = undefined;
    }
  }
}
```

}

Date -10 : Life Cycle Hooks

15/01/2022

Create college data table components (META File) :

Change the Meta File as follows:

```
<?xml version="1.0"?>
<LightningComponentBundle
xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>51.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>lightning__RecordPage</target>
    <target>lightning__AppPage</target>
    <target>lightning__HomePage</target>
  </targets>
</LightningComponentBundle>
```