**Day 1:**

Topic: Salesforce Overview.

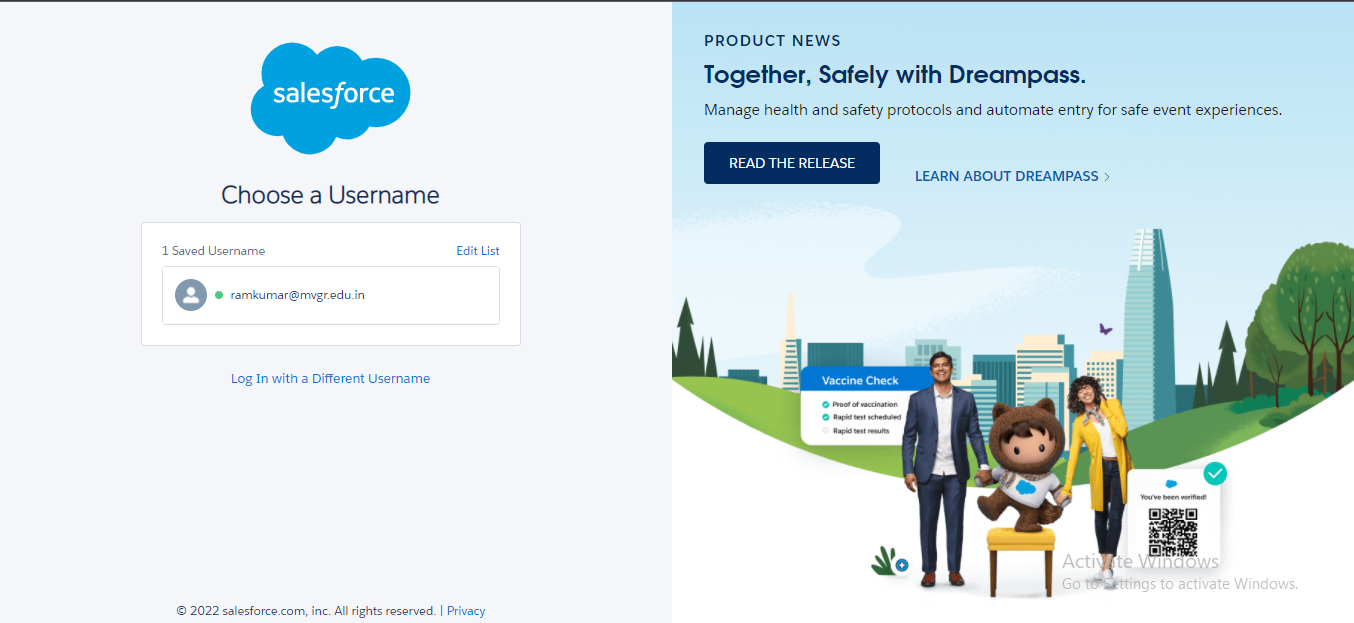
Milestone: Create Your Salesforce Developer Org to get Started.

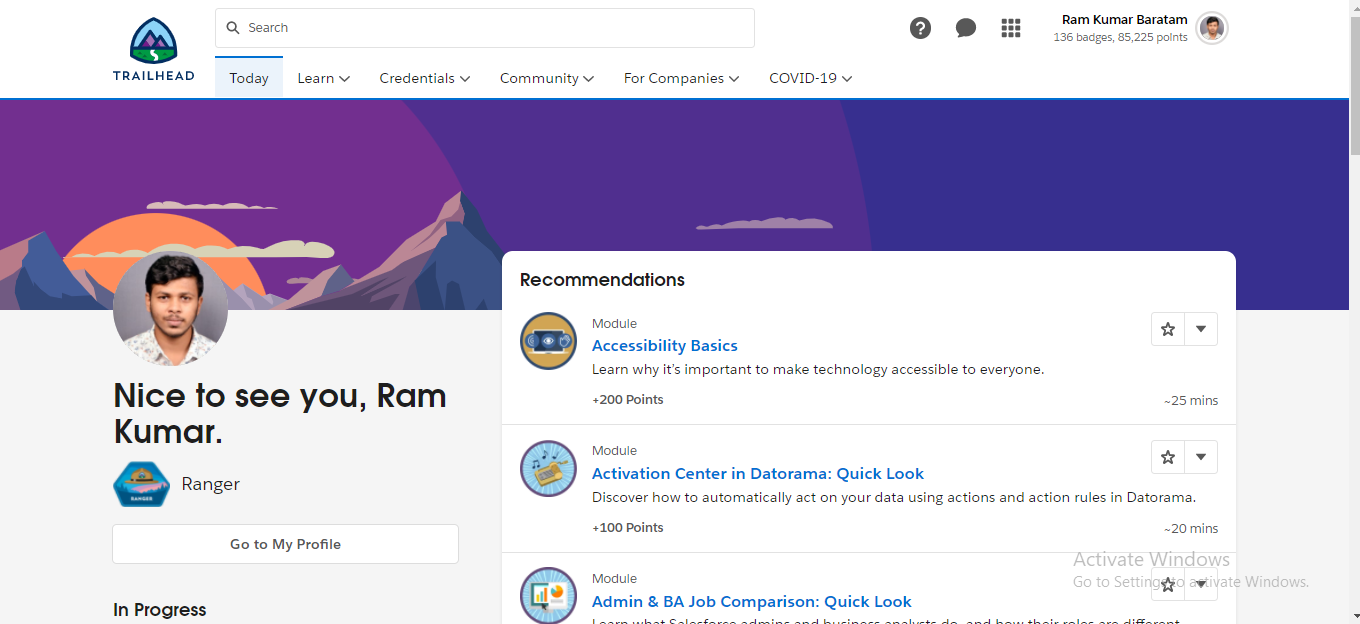
Detailed Description:

In this, to create a Salesforce Developer Org we have to visit developers.salesforce.com/ link and sign up there. We have to give all our details like Name, Email Id, Role, Company, Country, Postal Code, Username. Then click on Sign up. Now we get a Activation Mail to the Id which we have given while signing up for Developer Org. Now, in the mail Click on Activate Your Account. Then it takes 5-10 minutes to create a New Org.

Now, Login to your Salesforce Account by visiting salesforce.com login page and give the Username and Password which we have created just now.

Upload the Screenshot of MileStone:





**Day 2:**

Topic: Setting up First App, Custom Object and Fields Creation

Milestone: Custom Object Creation and Fields Creation

Detailed Description:

In this, First Create an App called College Management Application and then create the Objects which are required for the Application.

Create the Objects named

1. College
2. Application Form
3. Subjects
4. Students

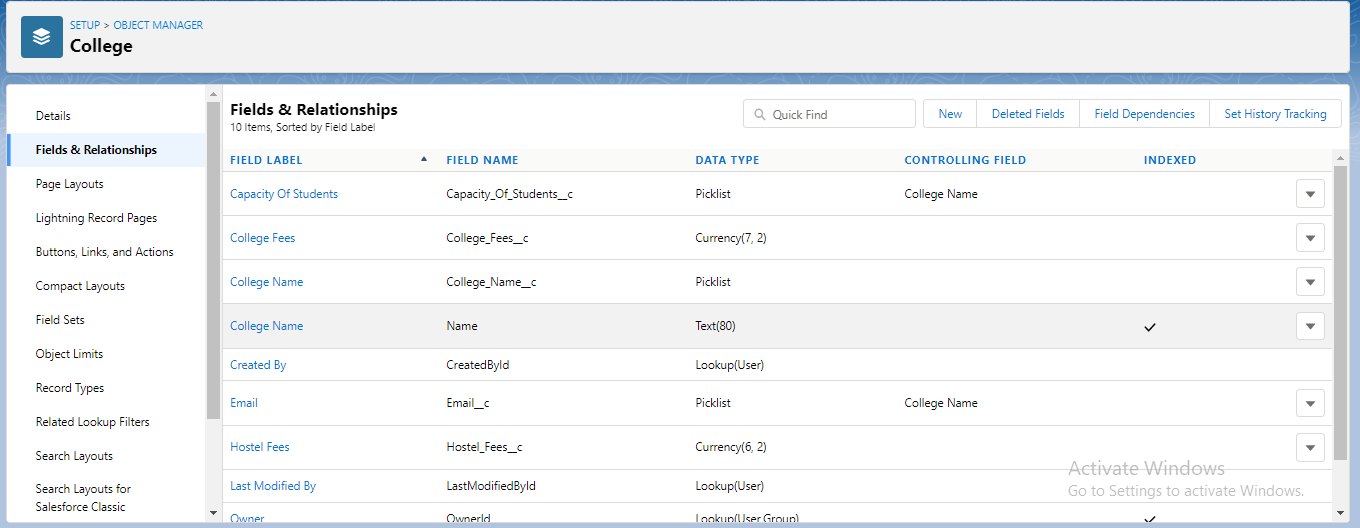
1. Create the fields for College Object like Record Info, College Fees, Hostel Fees, College Name, EMail, Capacity of Students.

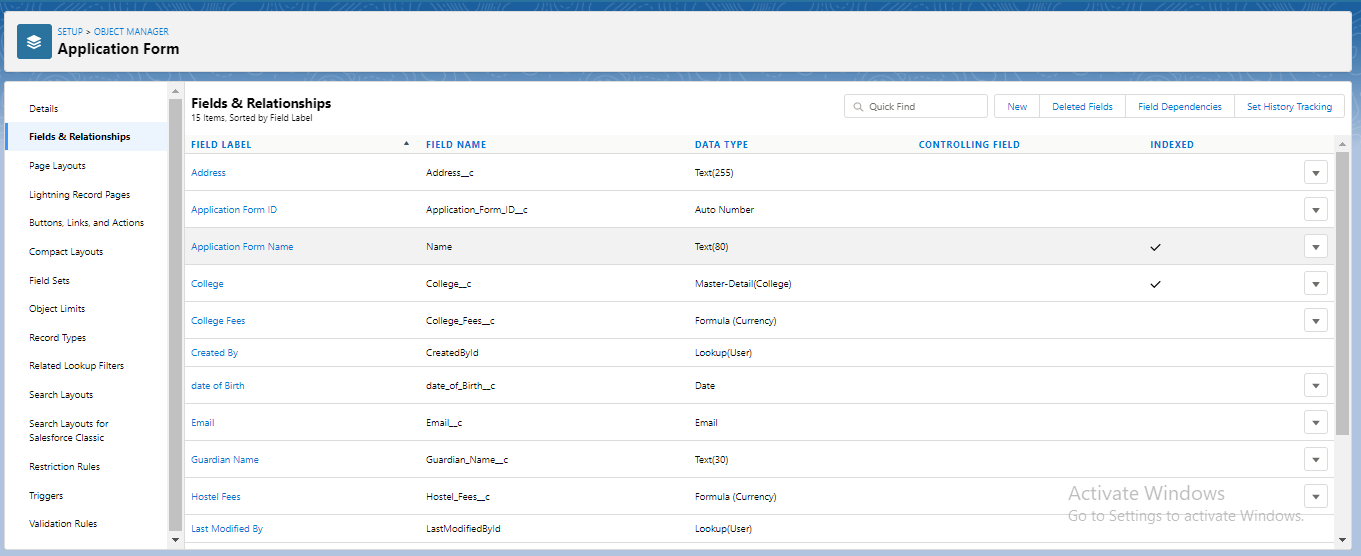
2. Create the fields for Application Form like Application Form ID, Address, College, College Fees, Hostel Fees, date of Birth, Email, Guardian Name, Looking For Hostel Stay, Ready To Join, Student Name.

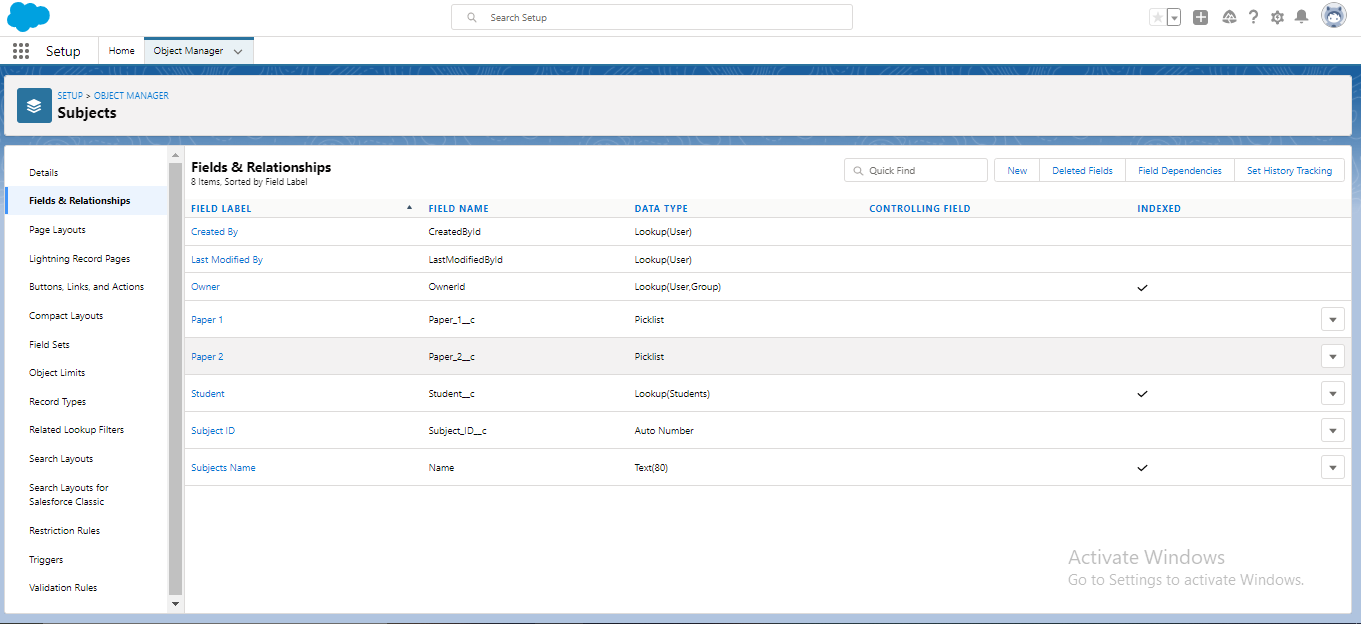
3. Create the Fields for Student Object like Student Name, Address, Application Form, College Name, Date Of Birth, Guardian Name, Phone.

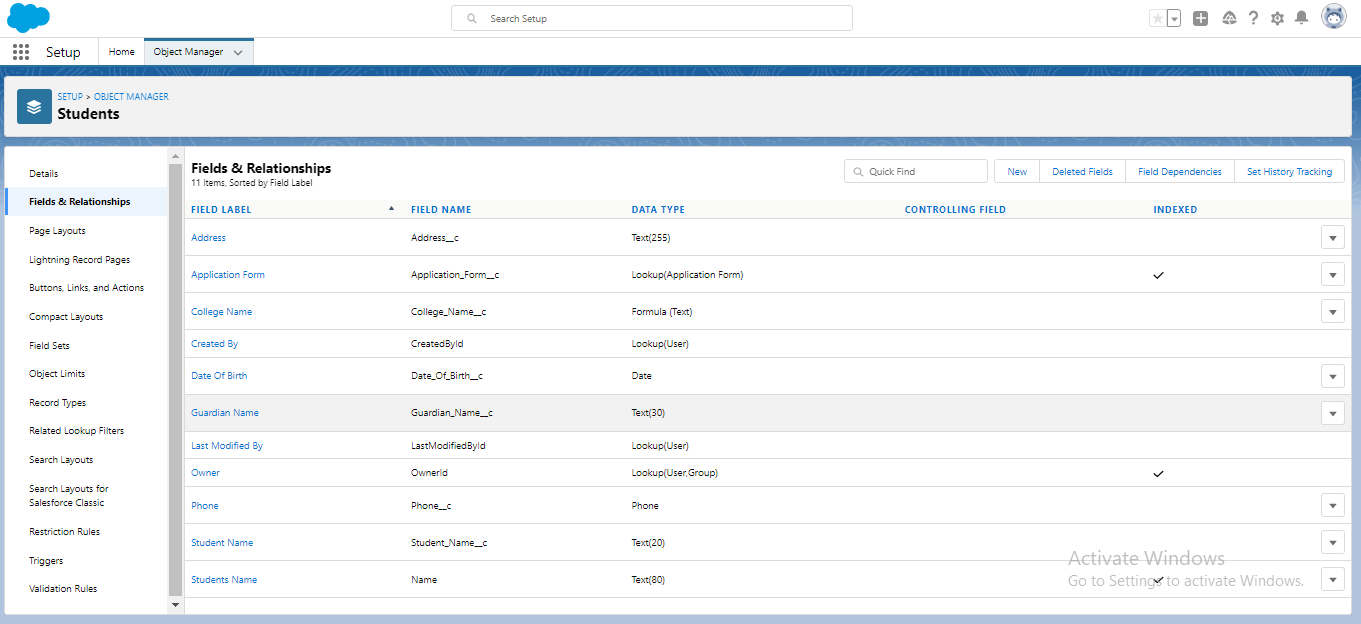
4. Create the Fields for Subject Object like Subject Id, Paper 1, Paper2, Student.

Upload the Screenshot of Milestone:









**Day 3:**

Topic: Adding Business Logic to Application

MileStone: Creating Global Picklist Values, Field Dependencies, Validation Rules, Process Automation and Student Record using Flow.

Detailed Description:

First, Create the Global Picklist Values for the Application.

a.) College:-

Picklist Value Name:- College

Values:- MIT-HYD

MIT-BLR

MIT-MUM

MIT-MAA

MIT-DEL

MIT-CCU

b.) Paper 1:-

Picklist Value Name:- Paper1

Values:- APEX

JAVA

C

C++

c.) Paper 2:-

Picklist Value Name:- Paper2

Values:- MATHEMATICS

ENGLISH

STATISTICS

Now, Create the Field Dependencies.

1.) Create Field Dependency for College Object where Controlling Field is college Name and Dependent Field is E-Mail. Select E-Mail Ids according to Colleges.

2.)Create Field Dependency for College Object where Controlling Field is college Name and Dependent Field is Capacity of Students. Select values according to your wish.

Now, Create Validation Rules.

1.) Create a validation rule on the college object such that the college name and record information 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 )  
 )  
 )

Now, 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 info specified in the application form record.

2.)Go to Setup-> select "Process Builder" from quick find.

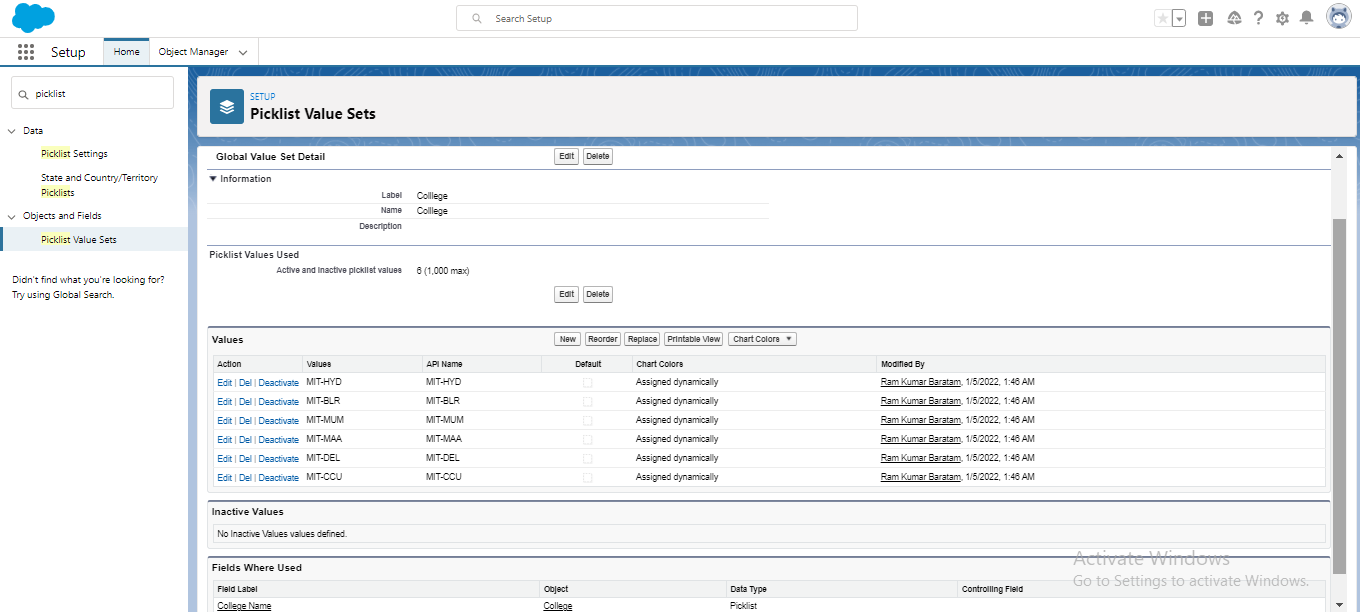
Create a Process Builder on the "Application Form" object with a condition as "When a record change". And select "When a record is selected or edited".

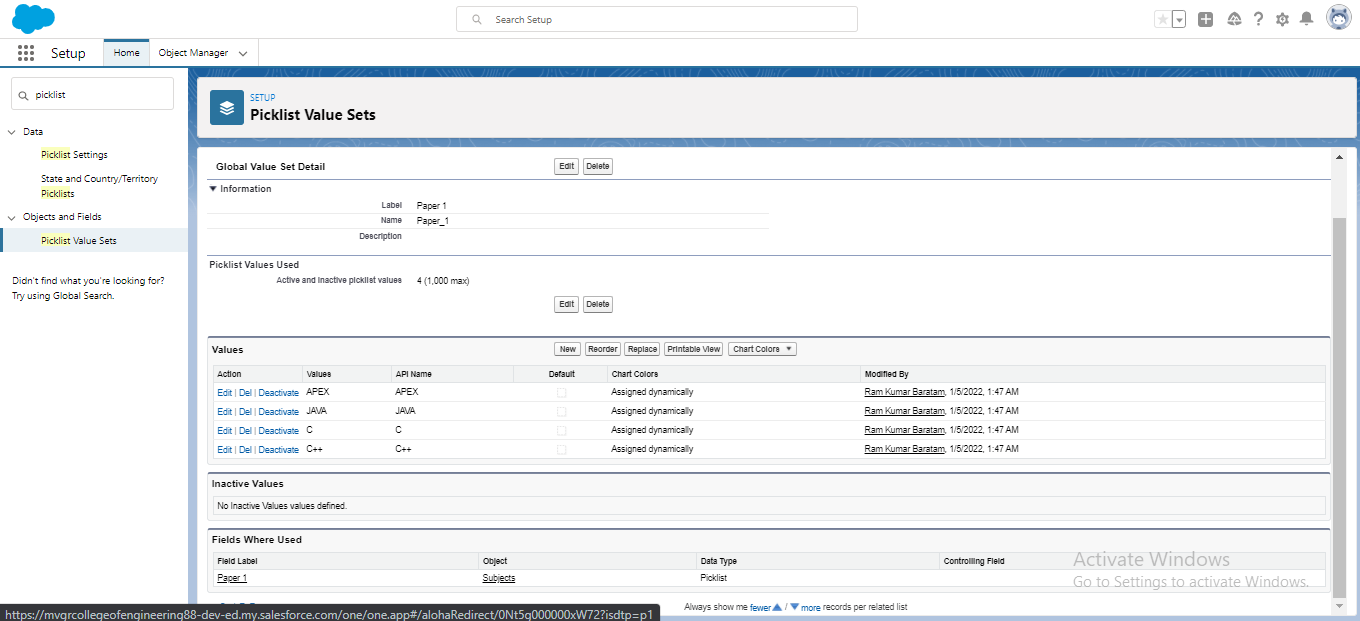
a.) In the diamond shape box, select the criteria which trigger the process builder to fire. In our example, it is "When Ready to Join field is checked."

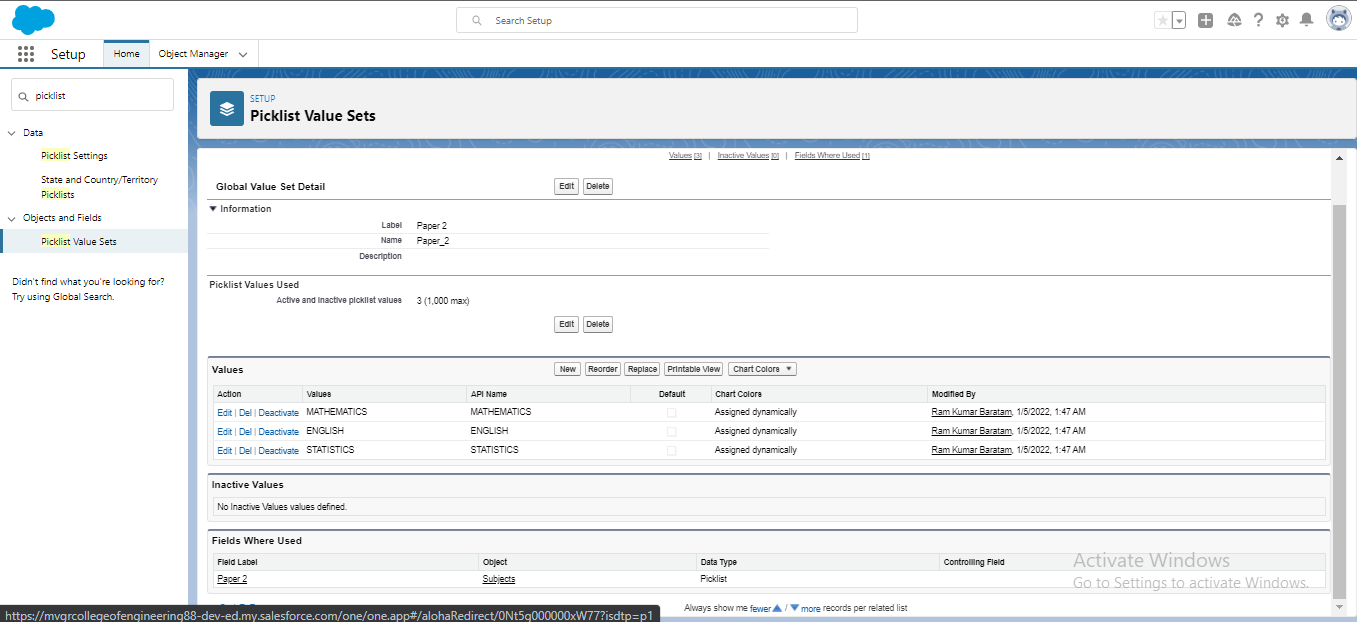
b.) Once the node is setup, click on the adjacent box called "Immediate Action". And select create a record on the student object.

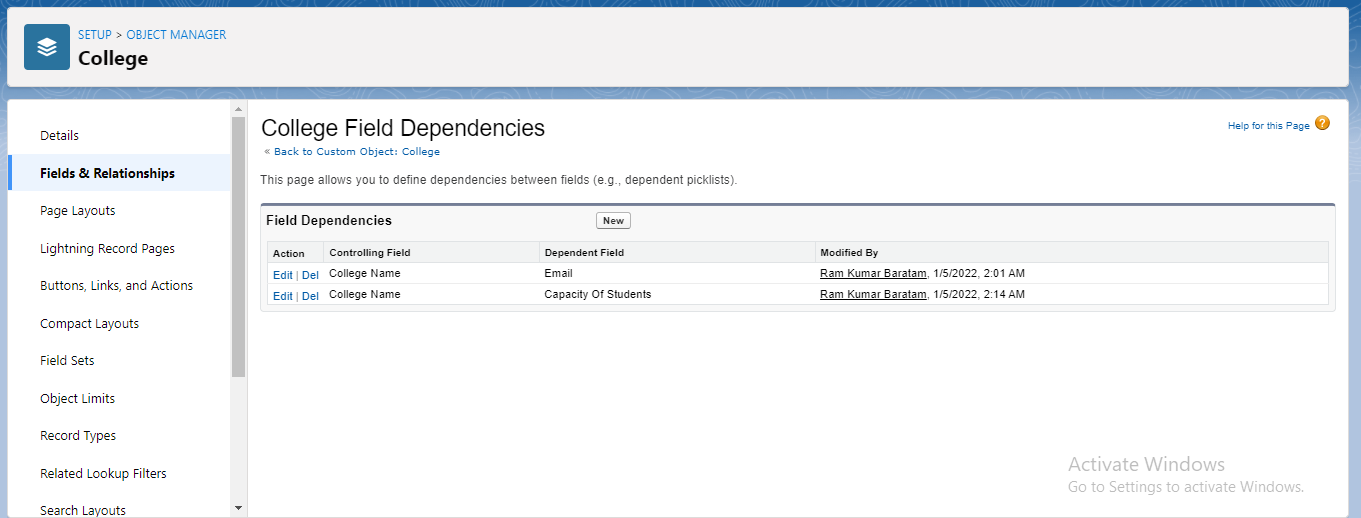
Now, Student Record Using Flow.

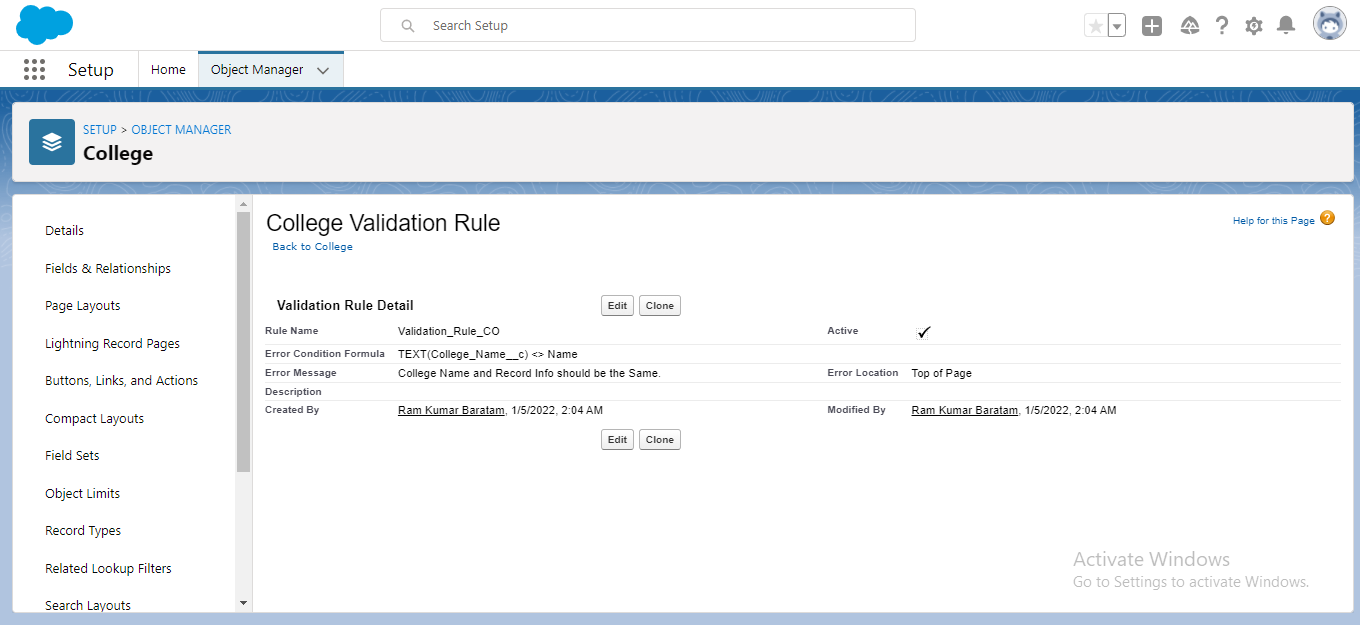
1. First deactivate the process builder which we created earlier.
2. Now search for flows and select new flow->record triggered flow.
3. For object select application form, in configure trigger select when the record is updated for entry criteria select as ready to join equals to true.
4. Now create a variable named student in the resource section.
5. Now add the assignment.
6. Now add Create Records.

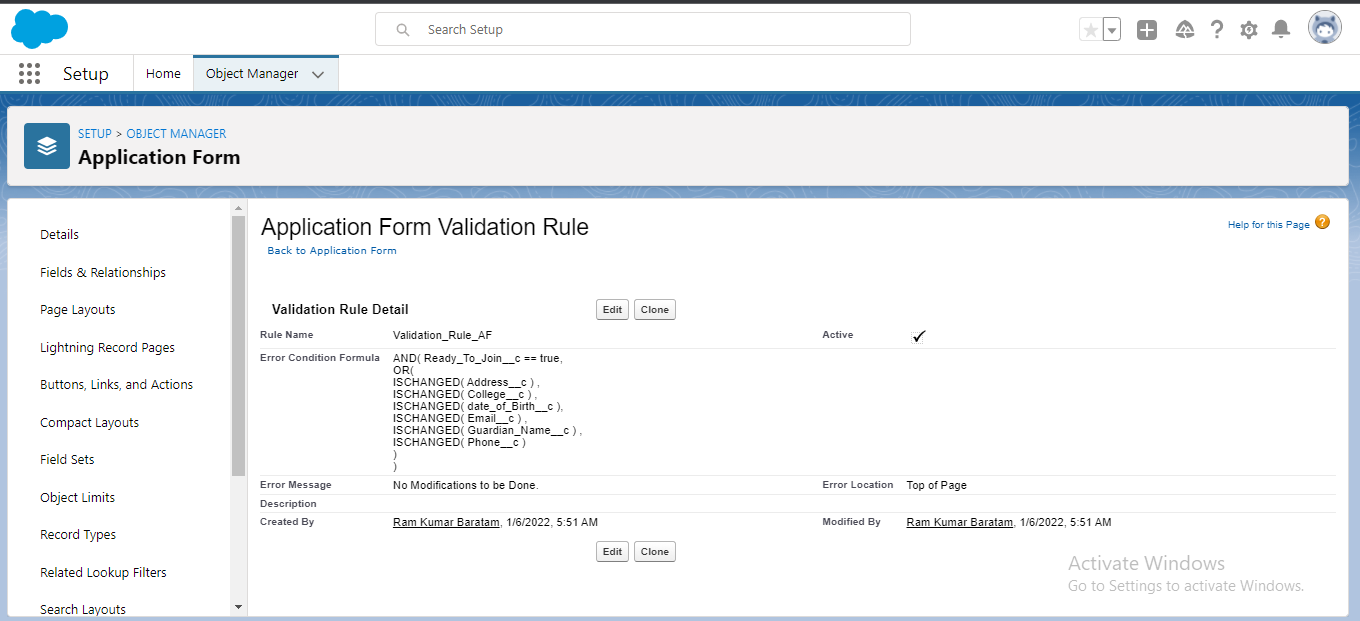


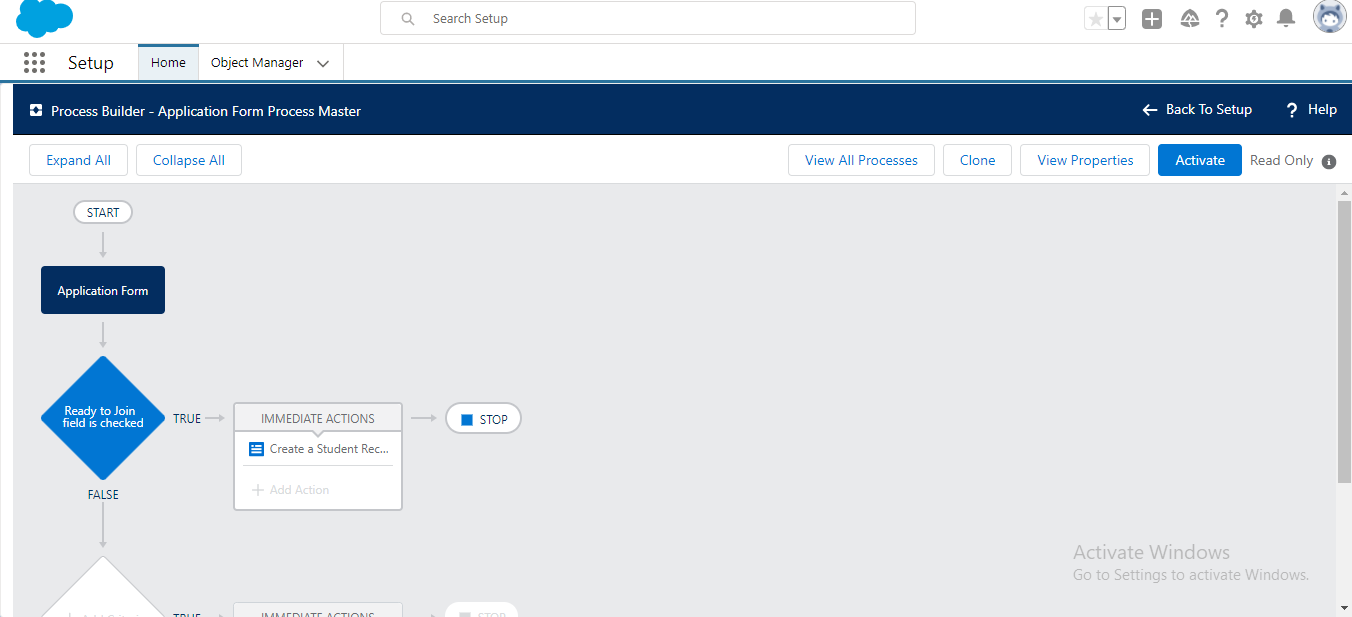


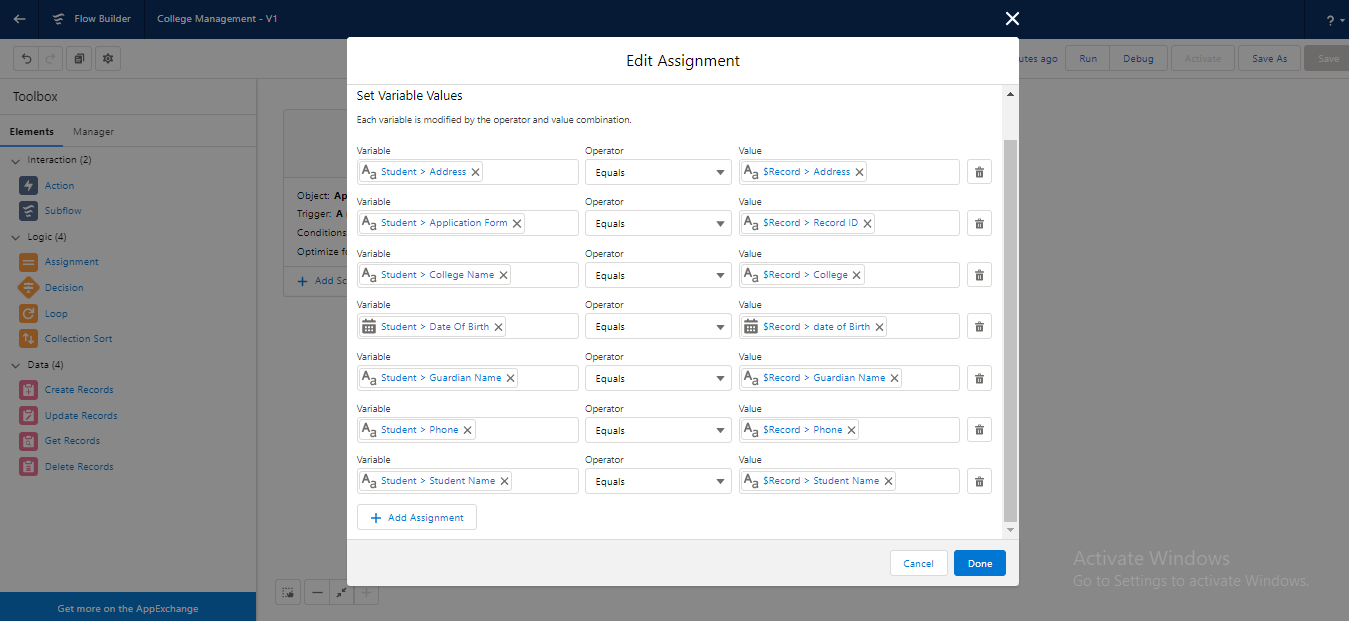












**Day 4:**

Topic: Batch Apex

Milestone: Create a Batch-apex for Application Form

Detailed Description:

First, Create a new Apex Class in Developer Console and enter the following code:

Code:-

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[] emailAddess = new String[]{'your email address'};

mail.setToAddresses(emailAddess);

Messaging.sendEmail(new Messaging.SingleEmailMessage[]{ mail } );

}

}

Now, change the Application Form API name to the name which we got while we have created the Application Form Object in College Management Application.

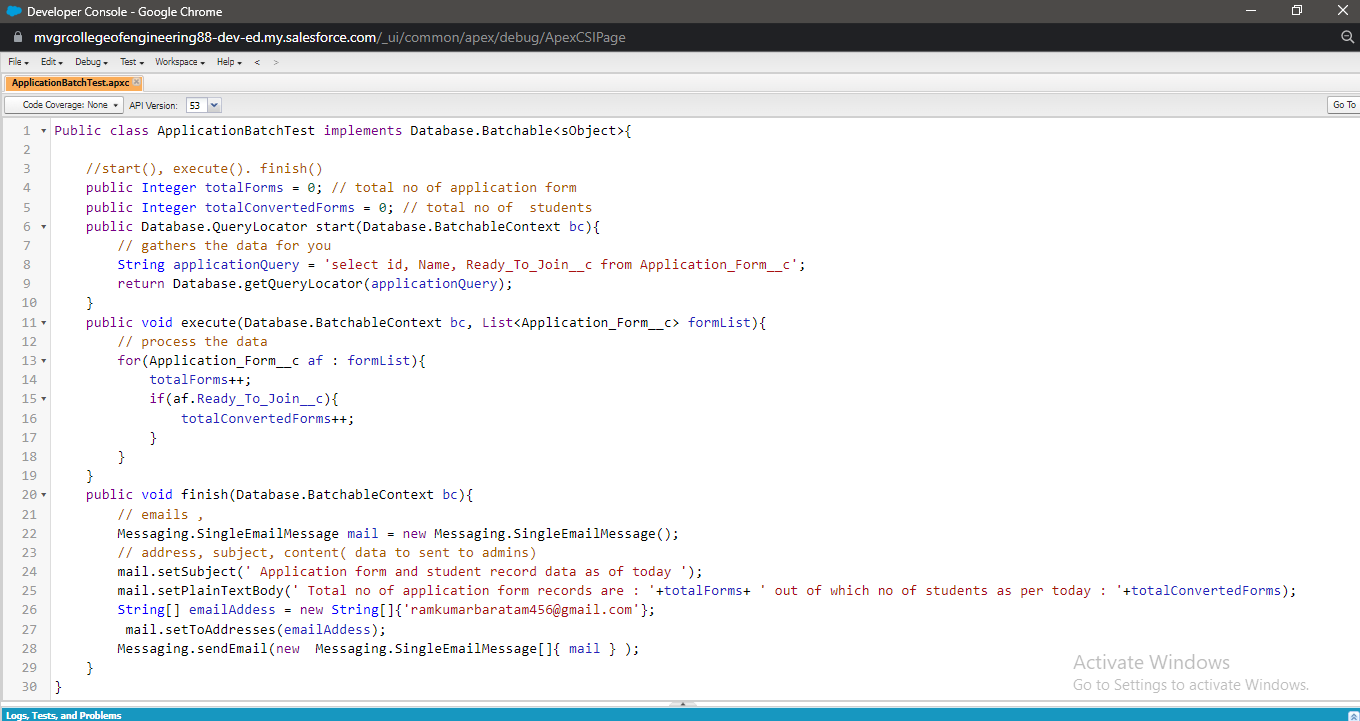
Save the File Name as ApplicationBatchTest and press Ctrl+E.

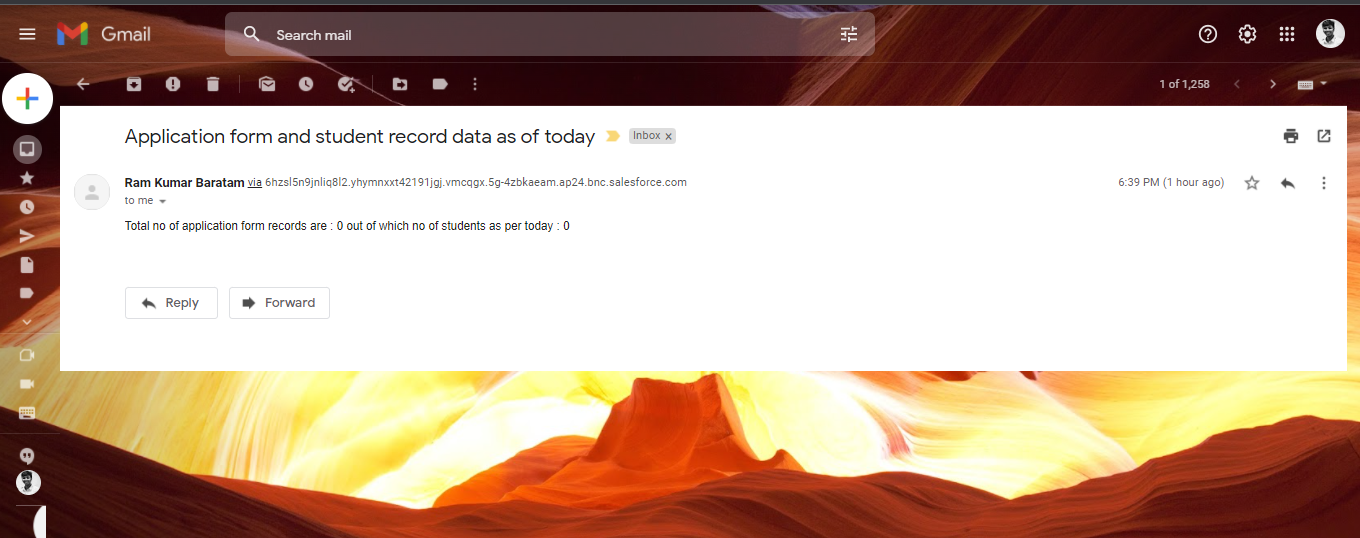
Then we get a Console. In it enter the Apex Code and execute it.

Now, remove "your E-Mail Address" text and put your personal Mail ID. Press Execute.

Then we get a Mail regarding the number of records which we have created in Application Form Object till date.

Upload the Screenshot:





Milestone: Create a Scheduler Class

Detailed Description:

From the Developer Console create a new Apex Class and enter the following code:

Code:

public class applicationschedule implements Schedulable{

public void execute(SchedulableContext sc){

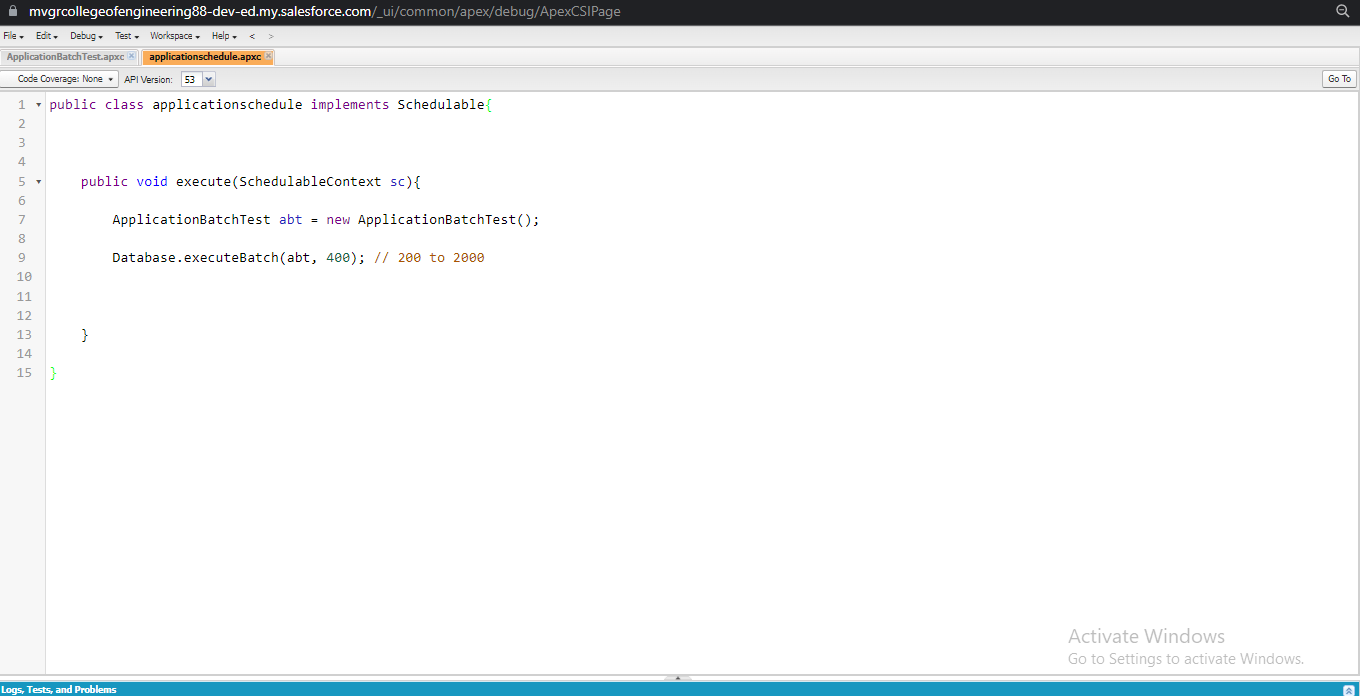
ApplicationBatchTest abt = new ApplicationBatchTest();

Database.executeBatch(abt, 400); // 200 to 2000

}

}

Upload the Screenshot:



**Day 5:**

Topic: Lightning Web Components

Milestone: College Data Table Component(APEX CLASS, HTML FILE, JAVA SCRIPT FILE, META FILE).

Detailed Description:

APEX CLASS:-

Create a 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 following 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;

}

}

HTML File:-

Create the component with following files:

<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>

JAVA SCRIPT 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, {CollegeId : '$recordId'})

wiredCollegeData({data, error}){

if(data){

this.recordList = data;

}

else if(error){

this.error = error;

this.recordList = undefined;

}

}

}

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>

Upload the Screenshot:

