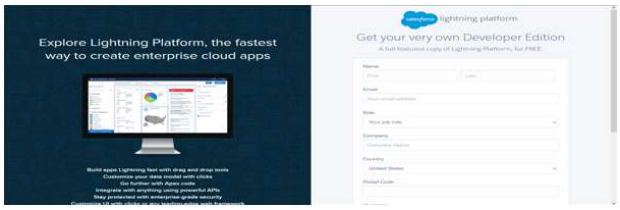
Day-1:

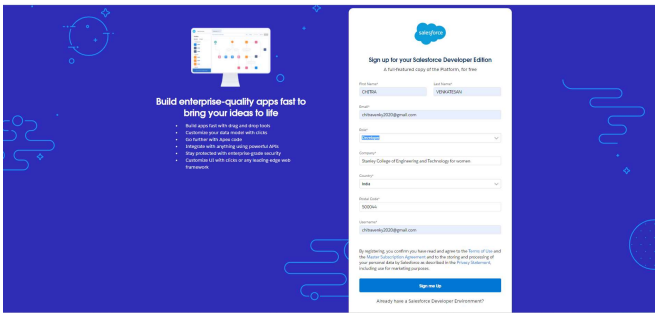
Create Your Salesforce Developer Org To Get Started

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 :
    1. First name & Last name
    2. Email
    3. Role : Developer
    4. Company : College Name
    5. County : India
    6. Postal Code : pin code
    7. 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

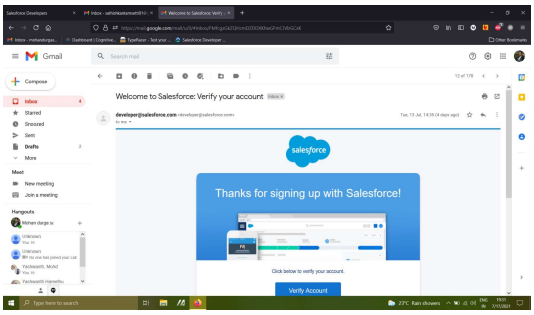
Click on sign up after filling these.





Account Activation

Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins, as allocating a developer org will take time.

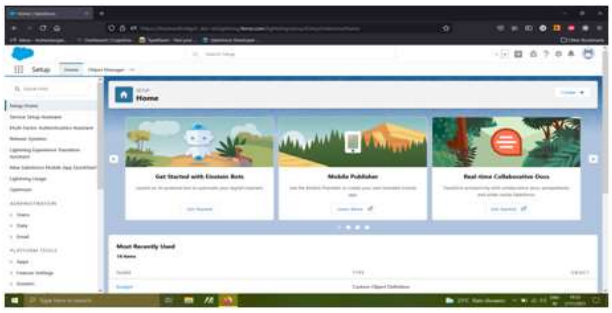


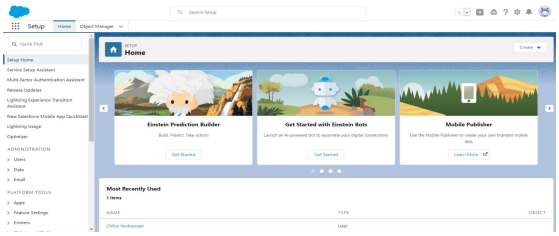
Login To Your Salesforce Account

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.





Day 2:

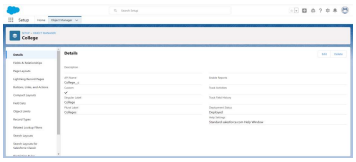
1. (i)Custom Object Creation:

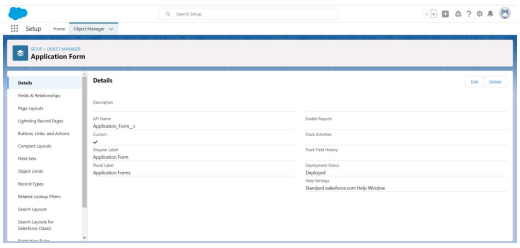
Custom Objects: Custom objects are the objects which are created manually by the users.

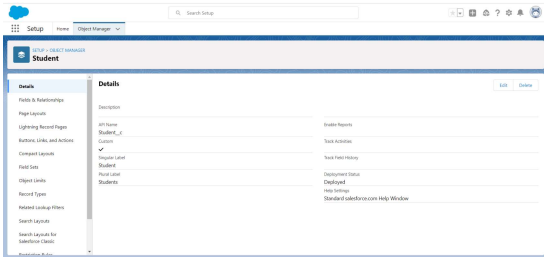
Create The Custom Objects For The Application

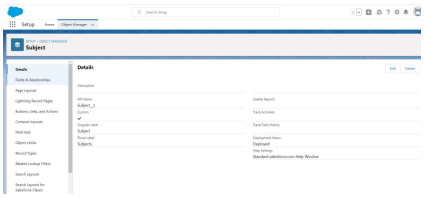
Create the following Custom Objects For the Application.

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





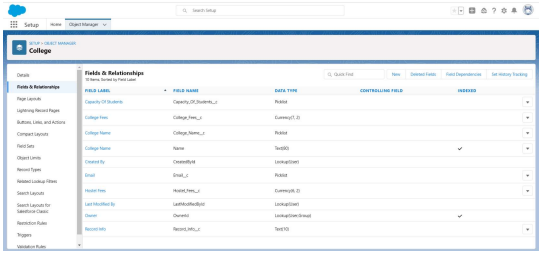




(ii)Create Fields On College Object

Create the following fields on the college object.

Field Name	Data Type	Required	Values
Record Info	Text		
College Fees	Currency(7,2)	Yes	
Hostel Fees	Currency(6,2)	Yes	
College Name	Picklist(Refer Business Logic In Milestones)	Yes	
Email	Picklist		blr@mit.co.in hyd@mit.co.in mum@mit.co.in maa@mit.co.in ccu@mit.co.in del@mit.co.in
Capacity Of Students	Picklist		500-1000, 1000-2500, 2500-6000, 6000-10000

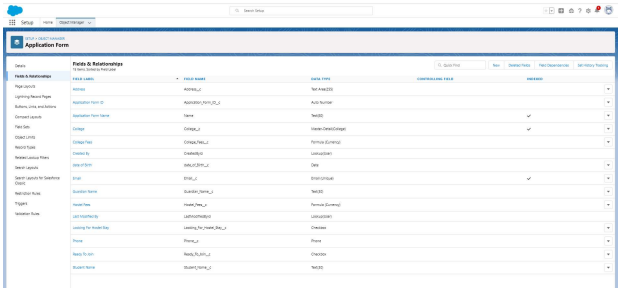


(iii)Create Fields On Application Form Object

Create the following fields on the application form object.

Field Name	Data Type	Required	Values
Application Form ID	Auto number		F-{000000} Starting Number=1
Address	Text(255)	Yes	
College	Master-Detail(College)	Yes	

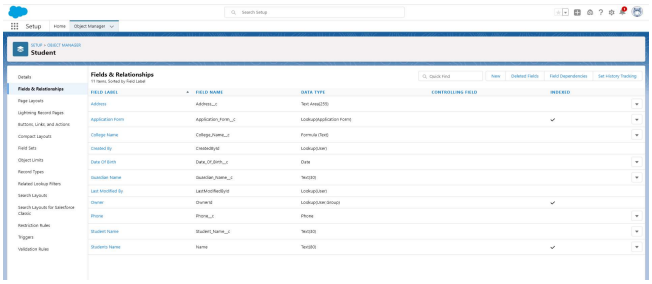
College Fees	Formula(Currency)		
Hostel Fees	Formula(Currency)		
date of Birth	Date	Yes	
Email	Email(Unique)	Yes	
Guardian Name	Text(30)	Yes	
Looking For Hostel Stay	Checkbox(default=Uncheck)		
Ready To Join	Checkbox(default=Uncheck)		
Student Name	Text(30)	Yes	
Phone	Phone	Yes	



(iv)Create Fields On Student Object

Create the following fields on the student object.

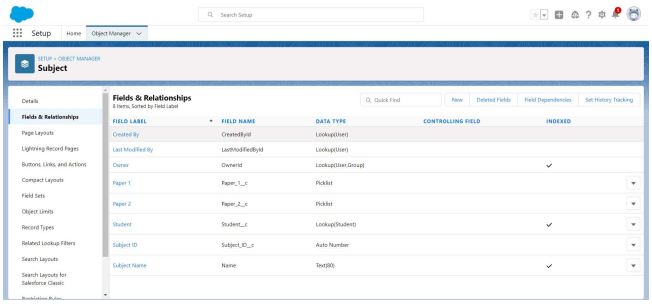
Field Name	Data Type	Required	Values
Student Name	Text		
Address	Text(255)	Yes	
Application Form	Lookup(Application Form)		
College Name	Formula(Text)		
Date Of Birth	Date	Yes	
Guardian Name	Text(30)	Yes	
Phone	Phone	Yes	



(v)Create Fields On Subject Object

Create the following fields On the Subject object.

Field Name	Data Type	Required	Values
Subject ID	AutoNumber		S-{000000} Starting Number=1
Paper 1	Picklist(Refer Business Logic Milestone)		
Paper2	Picklist(Refer Business logic Milestone)		
Student	Lookup(Student)		



2.Adding Business Logic To Application

(i)Creating Global Picklist Value Sets

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

a)College

Picklist Value	Values

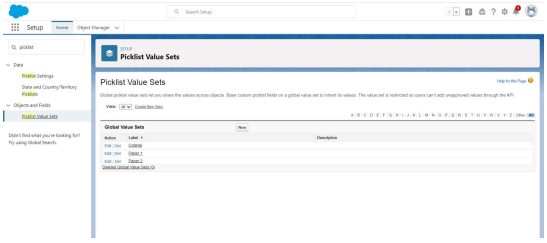
Name	
College	MIT-HYD MIT-BLR MIT-MUM MIT-MAA MIT-DEL MIT-CCU

b)Paper1

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

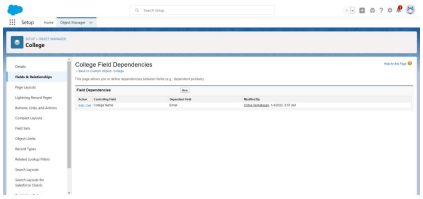
c)Paper2

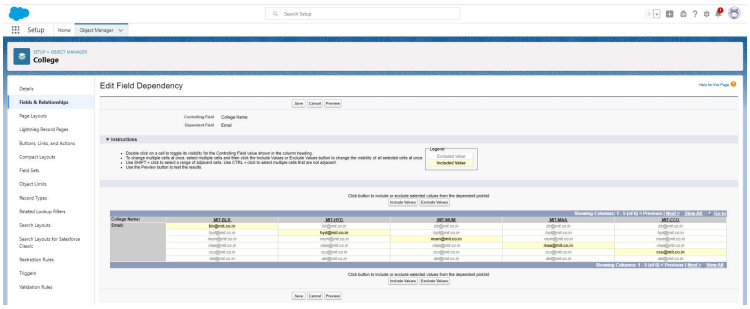
Picklist Value Name	Values
Paper2	MATHEMATICS ENGLISH STATISTICS



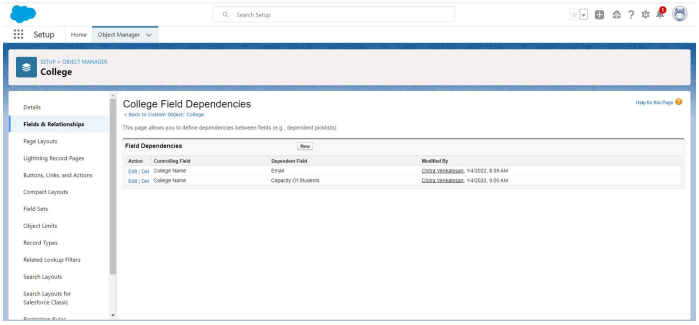
(ii)Creating Field Dependencies

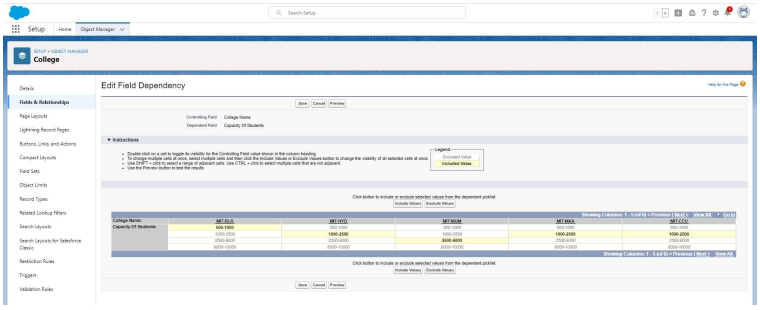
1. Create field dependency between college Name and Email, where the controlling field is college Name and dependent field is Email. Select the email ids according to the college names.





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.



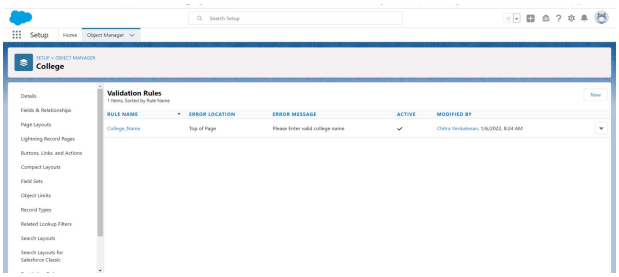


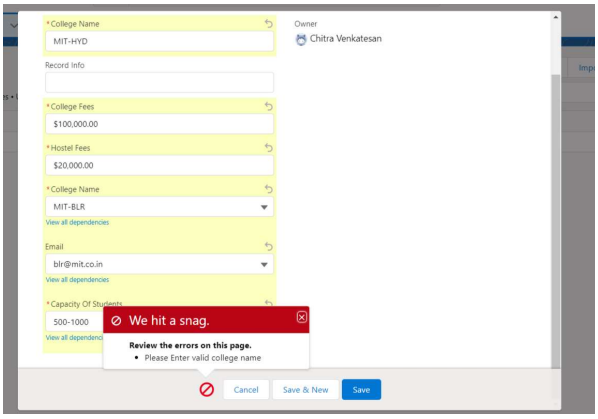
Day 3:

(iii)Create Validation Rule On College Object

1. Create a validation rule such that the college name and record info should have the same name.

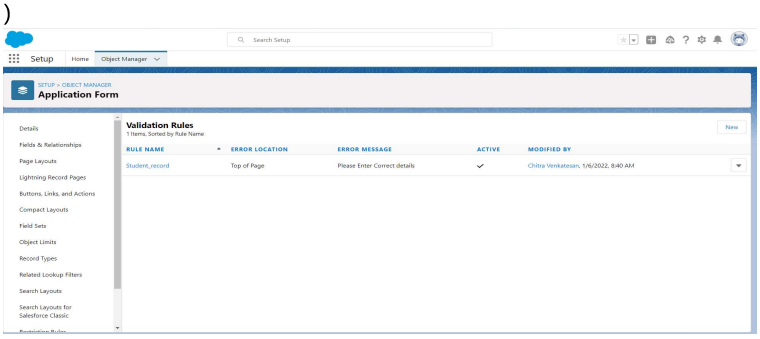
TEXT(College_Name__c) <> Name

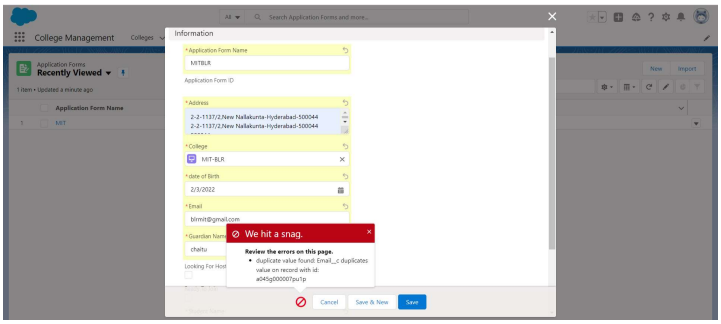




2.. Create a validation rule on the application form object to stop any modification on the application form once a student record is created.

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





Day 4:

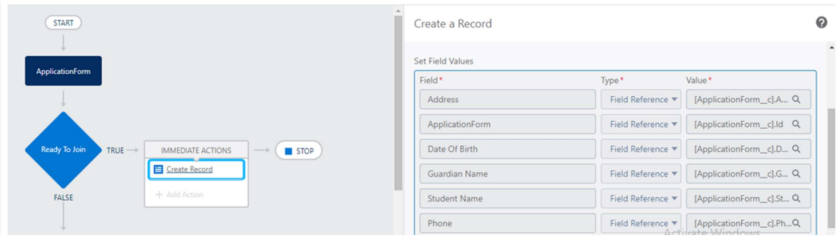
(iv)Process Automation

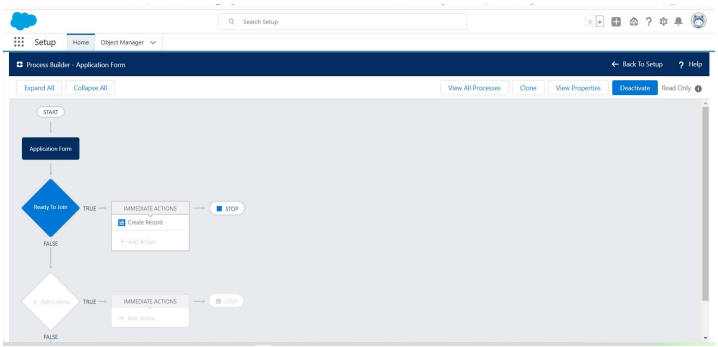
1. Create an automation process such that when the "ready to join" field is checked on the application form object we need to create the student record automatically with the information specified in the application form record.

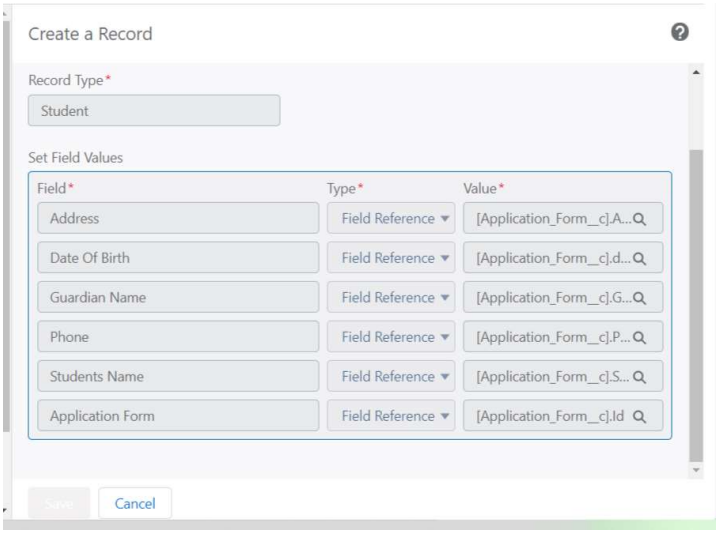
2. Go to Setup -> select “Process Builder” from quick find. Create a Process Builder on the “Application Form” object with a condition as “When a record change”. And select “When a record is created or edited”.

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.



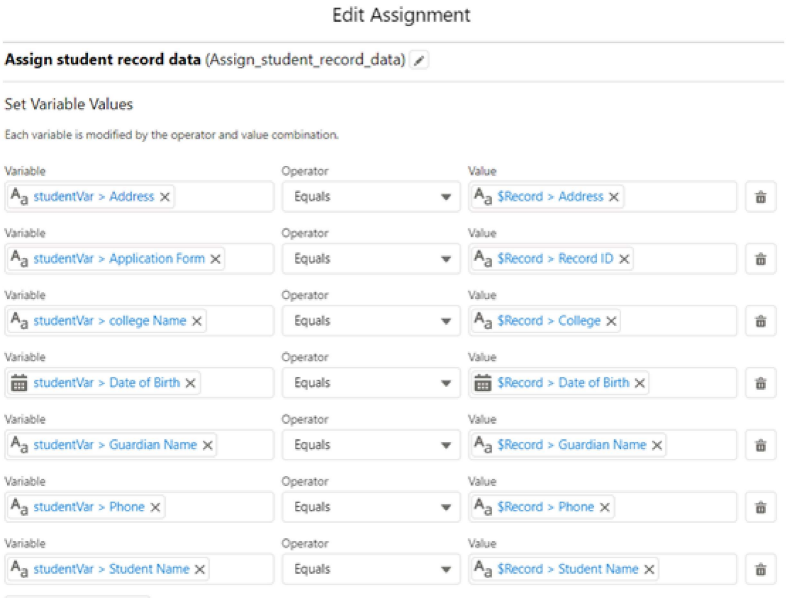




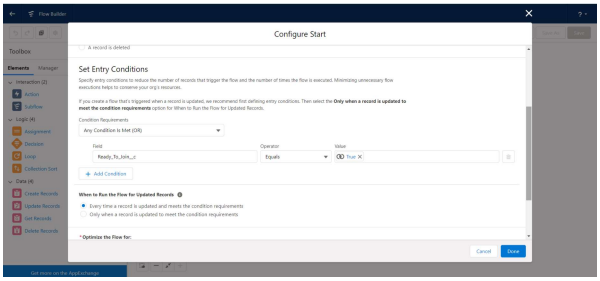
Day 5:

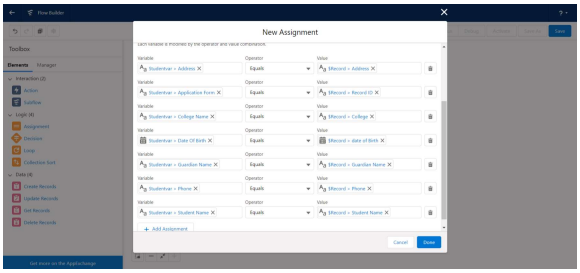
Create The Student Record Using Flow:

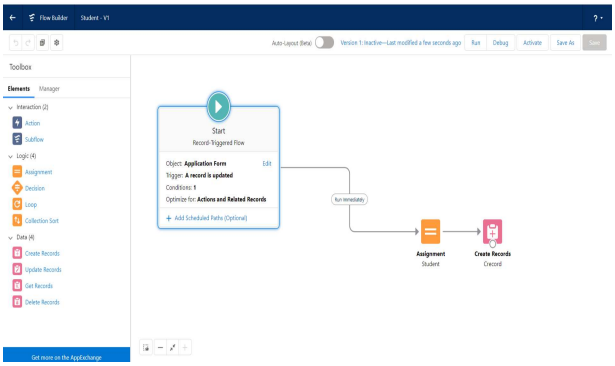
1. First deactivate the process builder which we created earlier.
2. Now search for flows and select new flow ->record triggered flow
3. For object select application form, in configure trigger select when the record is updated for entry criteria select as ready to join equals to true.
4. Now create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



6. Now add create records.







Day 6:

Create A Batchapex For Application Form

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

```
public class ApplicationBatchTest implements Database.Batchable<sobject> {
    public Integer totalForms = 0;
    public Integer totalConvertedforms = 0;
    public Database.QueryLocator start(Database.BatchableContext bc){

    // gathers the data for you
```

```
    String applicationQuery= 'select id, name, Ready_to_join_c from Application_Form__c';
    return Database.getQueryLocator(applicationQuery);
  }
  public void execute(Database.BatchableContext bc, List<Application_Form__c> formList){
    // process the data
    for(Application_Form__c af : formList) {
      totalForms++;
      if(af.Ready_to_Join__c){
        totalConvertedForms++;
      }
    }
  }
  public void finish(Database.BatchableContext bc){
    Messaging.SingleEmailMessage mail =  new Messaging.SingleEmailMessage();
    // address, sveject, content aata to sent to animi.
    mail.setSubject('Application from and student recond data as of today ');
      mail.setPlainTextBody('Totel no of application form records are : ' +totalForms+ ' out of
which no of students as per today :' +totalConvertedForms);
    String[] emailAddess = new String[]{'chitravenky2020@gmail.com'};
    mail.setToAddresses(emailAddess);
    Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail } );

  }
}
```

Create A Schedular Class

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

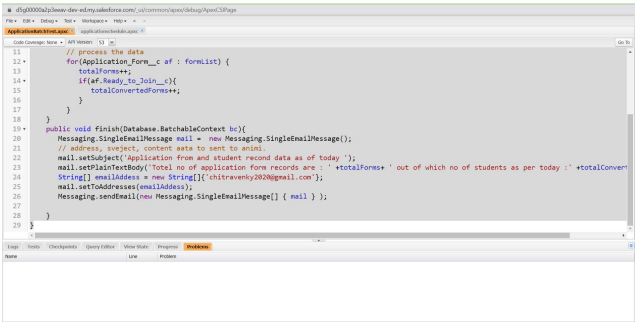
```
public class applicationschedule implements Schedulable{

    public void execute(SchedulableContext sc){
```

```
        ApplicationBatchTest abt = new ApplicationBatchTest();
        Database.executeBatch(abt, 400); // 200 to 2000

    }
}
```

From setup search for apex class and click on schedule jobs and fill the details as per your requirements



Day 7:

Lightning Web Components

Lightning Web Components (LWC) is a stack of modern lightweight frameworks built on the latest web standards. It is a DOM (Document Object Model), element created through reusable code and is used to generate a dynamic interface without using JavaScript or building a Library. This feasibility makes it quick and seamless, saving the developers a ton of time and effort on the Web Stack. Let's look at some of its remarkable features:

- Improved performance of the component as most of the code is recognized by the native web browser engine and web stack
- Ability to compose applications using smaller chunks of code since the crucial elements that are required to create a component is part of the native web browser engine and web stack
- Increase in the robustness of the applications built using LWCs as they are inclusive of

the said modern web standards.

- Parallel interoperability and feasibility to use both Lightning Web Components and Aura components together in the applications with no visible differentiation to the end-users

(i)Create College DataTable Component(APEX CLASS)

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 .

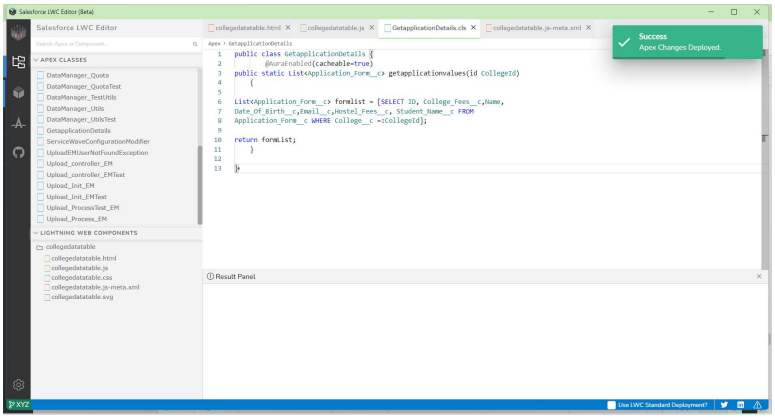
Create an Apex Class With The Required Functionality

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

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

return formList;
   }

}
```



Day 8:

(ii)Create College DataTable Component (HTML FILE)

Create the component with the following files.

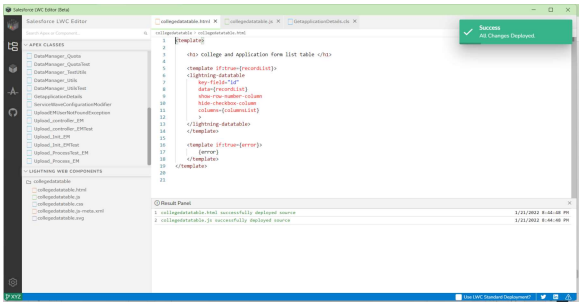
HTML:

```
<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 9:

(ii)

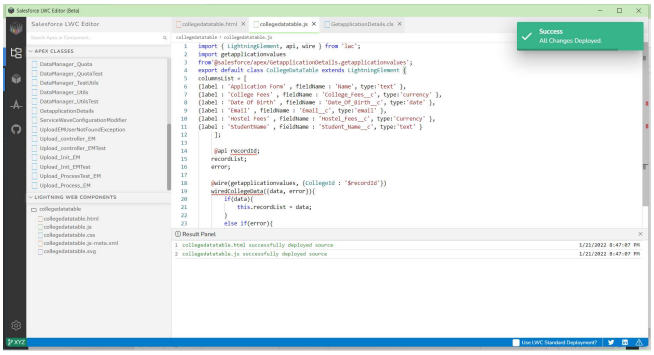
Create College DataTable Component(JAVA SCRIPT FILE)

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, {CollegeId : '$recordId'})
    wiredCollegeData({data, error}){
        if(data){
            this.recordList = data;
        }
        else if(error){
            this.error = error;
```

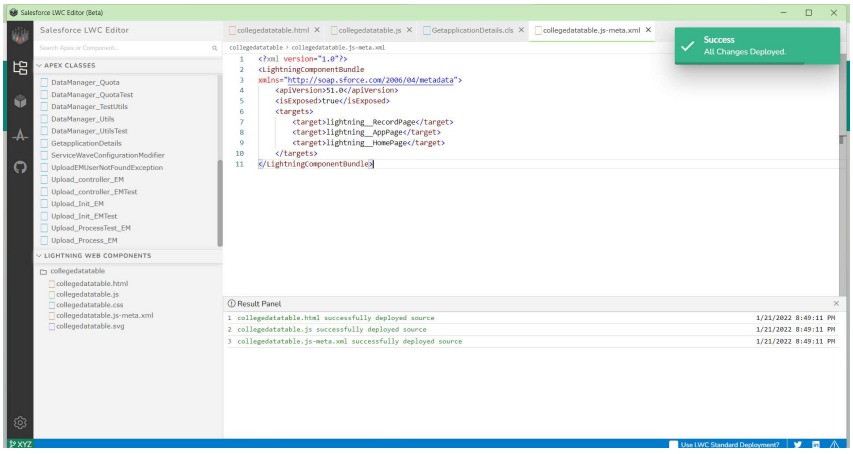


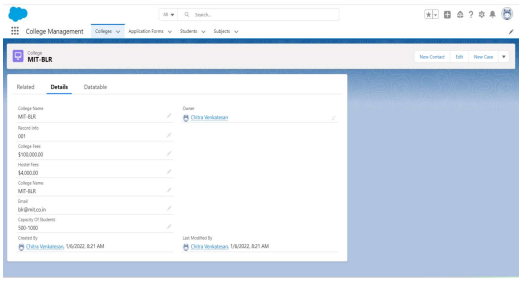
Day 10:

(iii)Create College DataTable Component(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>
```




College Management
Colleges
Application Forms
Students
Subjects
College
MIT-BLR
New Contact
Edit
New Case
Related
Details
Datatable
College Name
MIT-BLR
Owner
Chitra Venkatesan
Record Info
001
College Fees
$100,000.00
Hostel Fees
$4,000.00
College Name
MIT-BLR
Email
Mr@mit.co.in
Capacity Of Students
500-1000
Created By
Chitra Venkatesan, 1/6/2022, 8:21 AM
Last Modified By
Chitra Venkatesan, 1/6/2022, 8:21 AM