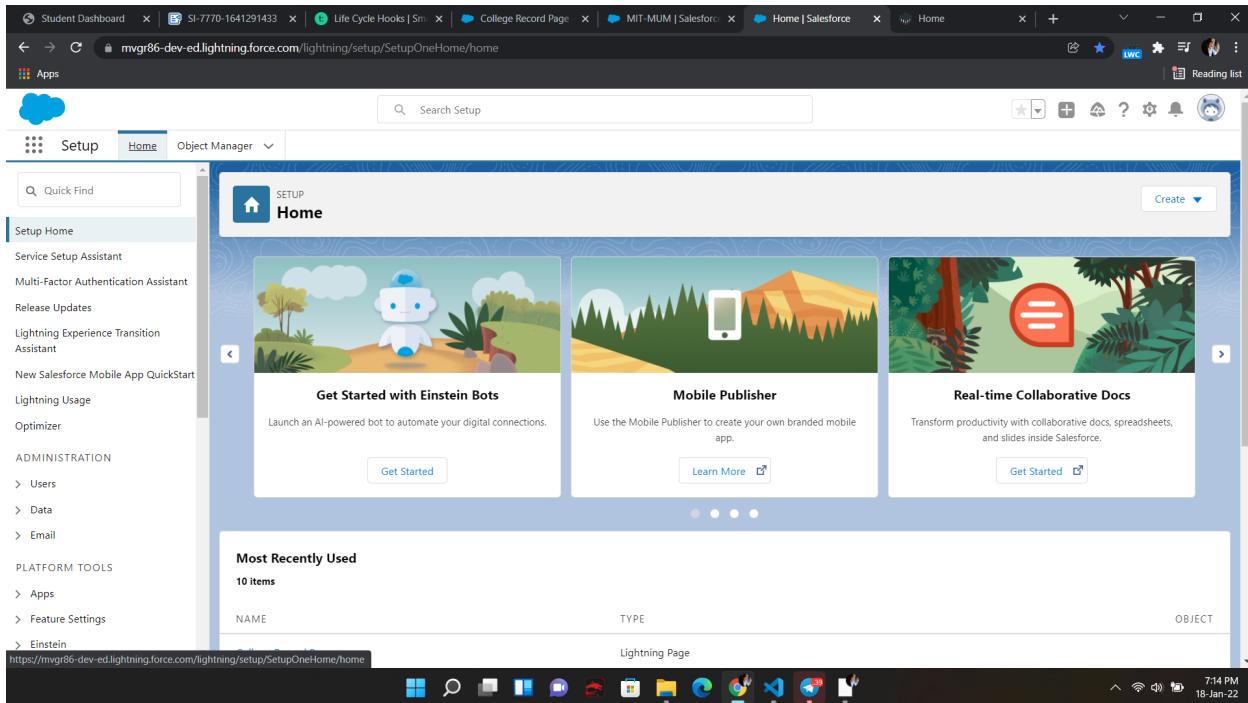


College Management Application

Day1-Activities :-

- 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.



Step1--> Creating a developer org in salesforce.

Step2--> Go to developers.salesforce.com/

Step3--> Click on sign up.

Step4--> On the sign up form, enter the following details :

Step5--> First name & Last name

Step6--> Email here give your email

Step7--> Role : Developer

Step8--> Company : College Name

Step9--> County : India

Step10--> Postal Code : pin code

Step11--> Username : should be a combine of your name and company

Step12--> This need not be an actual email id, you can give anything in the format

username@system.com

Day2-Activities :-

Topic: Milestone / Activities: Creating Objects, Fields and relationships

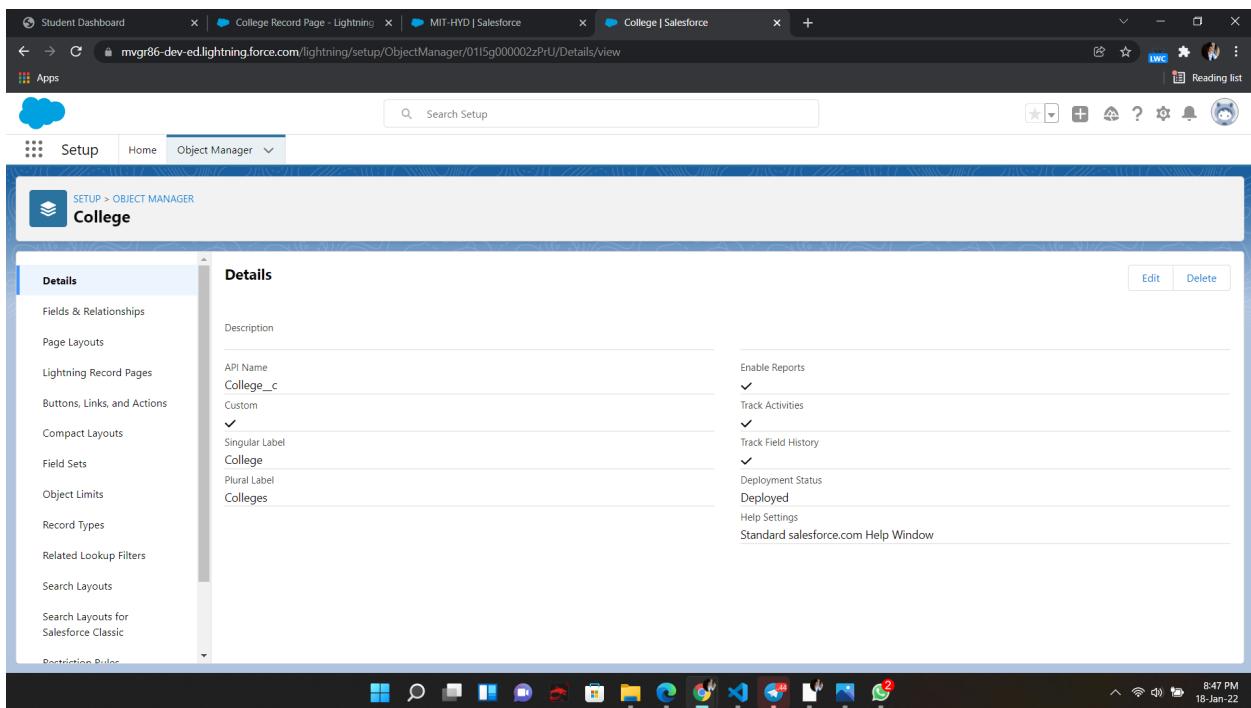
Custom Object Creation:

-->Created Custom Objects .

Here we have created College, Application Form, Students, Subjects

To create a custom object click on the gear button > click on set up >

There you can find the object manager > create > custom object



--> Similarly we have to create for the remaining objects

Fields & Relationships
11 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Capacity Of Students	Capacity__Of_Students__c	Picklist	College Name	
College Fees	College_Fees__c	Currency(7, 2)		
College ID	College_ID__c	Auto Number		
College Name	College_Name__c	Picklist		
College Name	Name	Text(80)		
Created By	CreatedById	Lookup(User)		
Email	Email__del__c	Picklist	College Name	
Hostel Fees	Hostel_Fees__c	Currency(6, 2)		
Last Modified By	LastModifiedBy	Lookup(User)		

-->Similarly created fields on Application Form object,Student object and in Subject object

--> Application Form Object Fields and Relationships

Fields & Relationships
15 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD
Application Form ID	Application_Form_ID__c	Auto Number	
Application Form Name	Name	Text(80)	
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)	
Guardian Name	Guardian_Name__c	Text(30)	
Hostel Fees	Hostel_Fees__c	Formula (Currency)	
Last Modified By	LastModifiedBy	Lookup(User)	

--> Student Object Fields and Relationships

The screenshot shows the Salesforce Setup interface for the 'Student' object. On the left, a sidebar lists various setup categories like Page Layouts, Lightning Record Pages, and Field Sets. The main area is titled 'Fields & Relationships' and displays a table of fields. The columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The table contains the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address_c	Text Area(255)		
Application Form	Application_Form__c	Lookup(Application Form)		
College Name	College_Name__c	Formula (Text)		
Created By	CreatedById	Lookup(User)		
Date Of Birth	DateOfBirth__c	Date		
Guardian Name	Guardian_Name__c	Text(30)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Phone	Phone__c	Phone		

-->Subject Object Fields and Relationships

The screenshot shows the Salesforce Setup interface for the 'Subject' object. On the left, a sidebar lists various setup categories. The main area is titled 'Fields & Relationships' and displays a table of fields. The columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The table contains the following data:

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)		

-->Now creating Tabs to the objects which we have created

click on the gear button> there we can find the quick find box >

search for tabs> click on tabs

here in tabs we can find different types like:

Custom Object Tabs

Visual force Tabs

Lightning Componenet Tabs

Lightning Page Tabs

we need to create in Custom Object Tabs because we have created custom objects for the app manager

The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page open. The page title is 'Custom Tabs'. It includes a search bar and a help link. Below the title, there's a brief description of what custom tabs are and how they differ from standard tabs. The main content area is divided into three sections: 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. The 'Custom Object Tabs' section contains four entries:

Action	Label	Tab Style	Description
Edit Del	Application Forms	Computer	
Edit Del	Colleges	Castle	
Edit Del	Students	Books	
Edit Del	Subjects	Box	

The 'Web Tabs' and 'Visualforce Tabs' sections both show a message indicating no tabs have been defined.

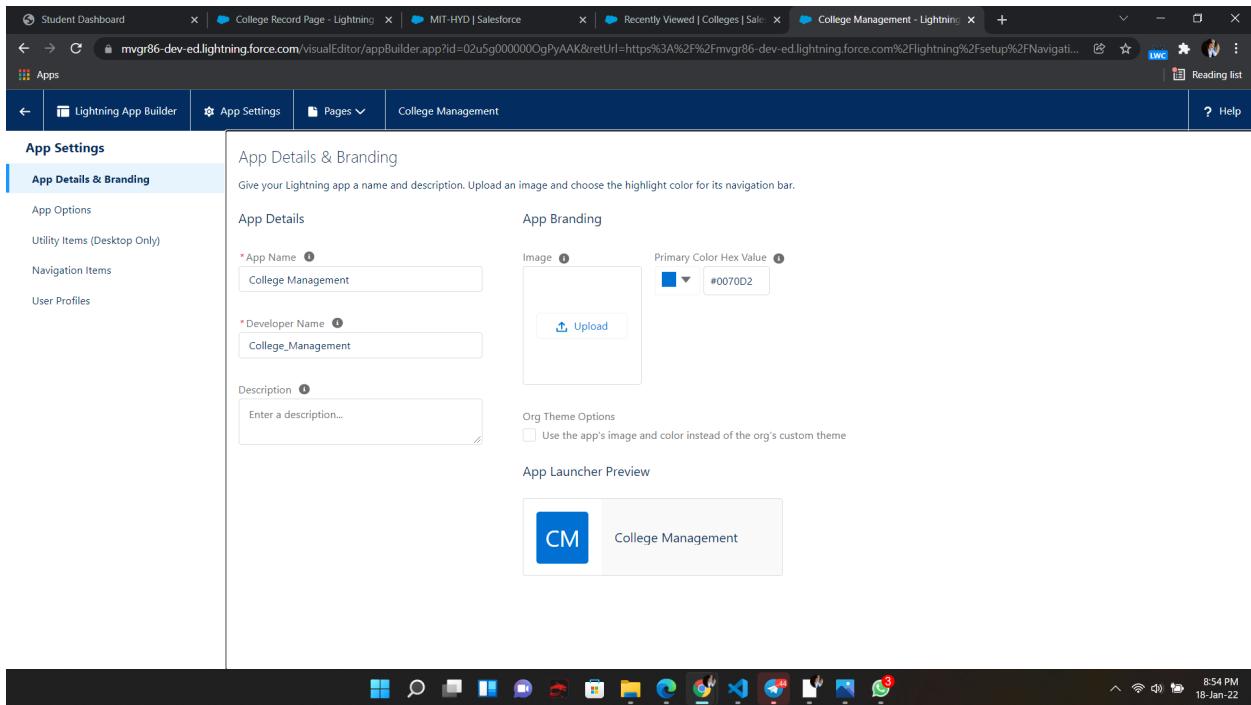
DAY 3:

Topic: Milestone / Activities: Create an app manager and creating global picklist

Detailed Description:

Create an App manager for College Management Application:

we have to create an App for the college management application so
Navigate to Quick find box> search for App manager >click on app manager



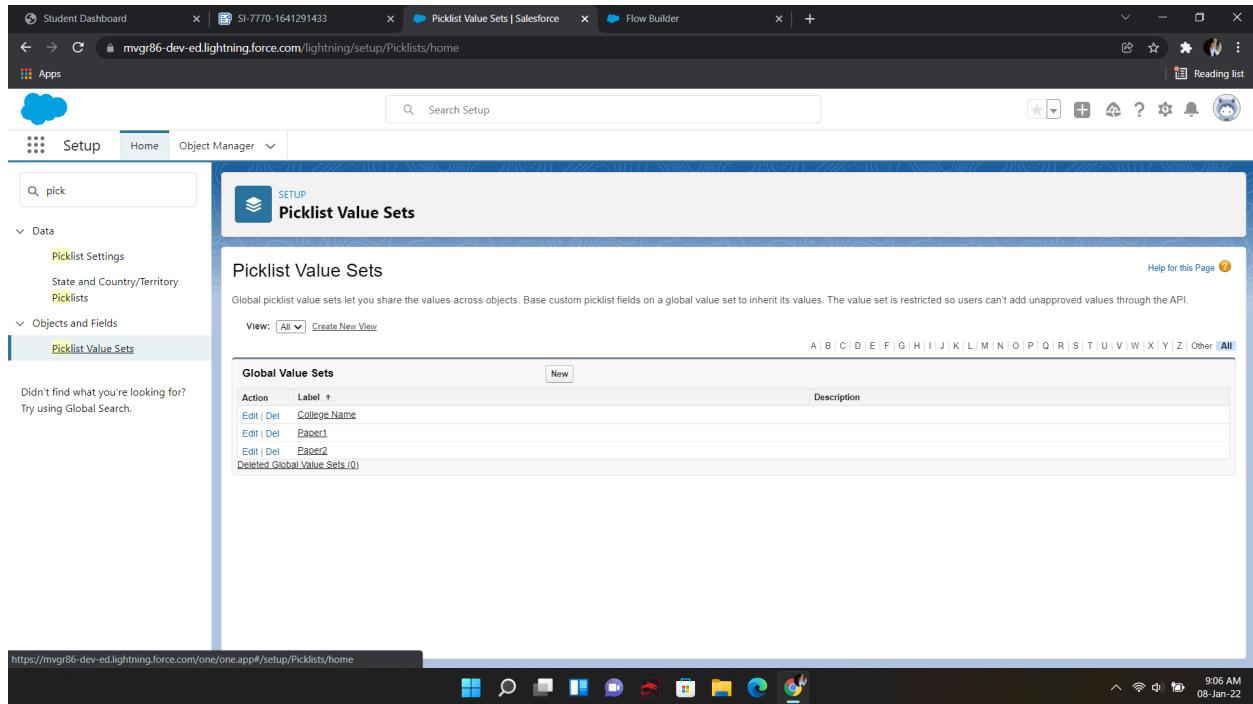
-->Creating global picklist values

when we declaring a field as picklist we can choose if we want custom picklist or global picklist

we need to create the global picklist values before using this

so head to quick find box in the setup>picklist value sets

click on new and enter the name and values



DAY 4:

Topic: Milestone / Activities: Creating fields for objects, and creating field dependencies

Detailed Description:

Creating fields for the objects which we have created (College, students, Subjects and Application form)

to create the fields in the object

click on the gear button > select setup > here we can see the object manager here we find the all objects which are present

--> Now we have to Create fields on College Object as show below

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Capacity Of Students	Capacity_of_Students__c	Picklist	College Name	
College Fees	College_Fees__c	Currency(7, 2)		
College ID	College_ID__c	Auto Number		
College Name	College_Name__c	Picklist		
College Name	Name	Text(80)		✓
Created By	CreatedById	Lookup(User)		
Email	Email_del__c	Picklist	College Name	
Hostel Fees	Hostel_Fees__c	Currency(6, 2)		
Last Modified By	LastModifiedBy	Lookup(User)		

-->Now we have to Create fields on Application Form Object as show below

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Application Form ID	Application_Form_ID__c	Auto Number		
Application Form Name	Name	Text(80)		✓
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)		✓
Guardian Name	Guardian_Name__c	Text(30)		
Hostel Fees	Hostel_Fees__c	Formula (Currency)		
Last Modified By	LastModifiedBy	Lookup/User		

-->Now we have to Create fields on Student Object as show below

The screenshot shows the Salesforce Setup interface with the URL mvgr86-dev-ed.lightning.force.com/lightning/setup/ObjectManager/0115g000002PrZ/FieldsAndRelationships/view. The page title is "Student | Salesforce". The left sidebar under "Fields & Relationships" includes links for 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, and Restriction Rules. The main content area is titled "Fields & Relationships" and lists 10 items, sorted by Field Label. The table columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The listed fields are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text Area(255)		
Application Form	Application_Form__c	Lookup(Application Form)		
College Name	College_Name__c	Formula (Text)		
Created By	CreatedById	Lookup(User)		
Date Of Birth	DateOfBirth__c	Date		
Guardian Name	Guardian_Name__c	Text(30)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User;Group)		
Phone	Phone__c	Phone		

-->Now we have to Create fields on Subject Object as show below

The screenshot shows the Salesforce Setup interface with the URL mvgr86-dev-ed.lightning.force.com/lightning/setup/ObjectManager/0115g000002PrZ/FieldsAndRelationships/view. The page title is "Subject | Salesforce". The left sidebar under "Fields & Relationships" includes links for 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, and Restriction Rules. The main content area is titled "Fields & Relationships" and lists 8 items, sorted by Field Label. The table columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The listed fields are:

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	Name		

Creating Field Dependencies

We have created some fields in every object so now we need to give some field dependencies

-> Created Field Dependency between two Objects

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

The screenshot shows the Salesforce Setup interface for the 'College' object. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, and Lightning Record Pages. The main content area is titled 'College Field Dependencies' and shows a table of existing dependencies:

Action	Controlling Field	Dependent Field	Modified By
Edit Del	College Name	Email	tharun kumar kolla 1/4/2022, 4:15 AM
Edit Del	College Name	Capacity Of Students	tharun kumar kolla 1/5/2022, 2:49 AM

DAY 5:

Topic: Milestone / Activities: Creating Validation rules, and creating process automation

Detailed Description:

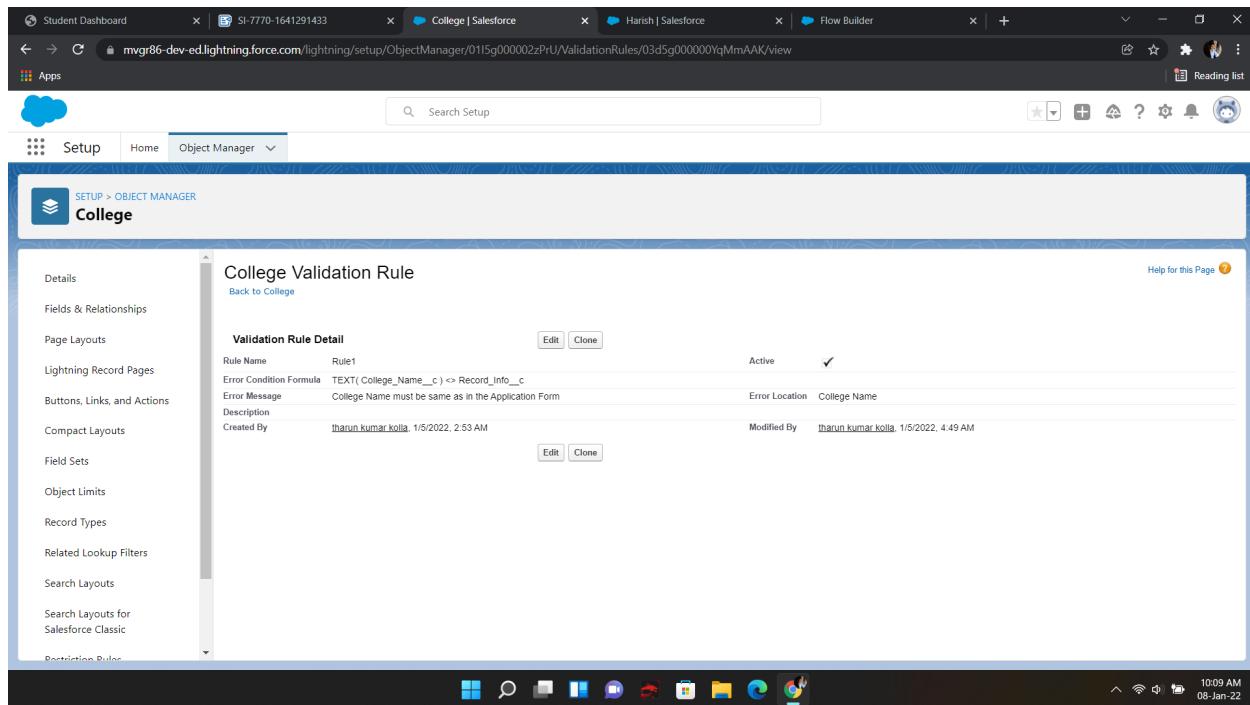
Creating Validation Rules

-> Created Validation Rule on College Object

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.



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".

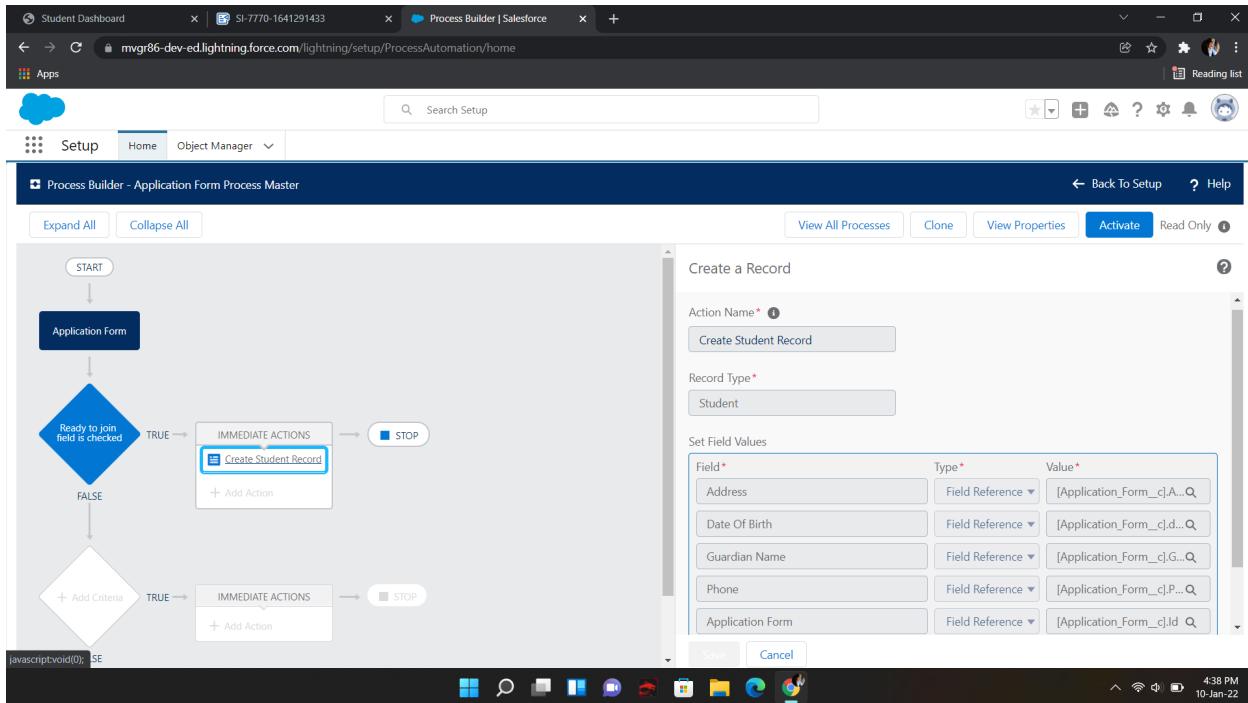
a. In the diamond shape box(called nodes), select the criteria which trigger the Process builder to fire. In our example, it is "When Ready to Join field is checked."

B Once the node is set up, click on the adjacent box called "Immediate action". And select create a record on the student object. Please follow the below screenshot.

To create a Process Automation

In quick find box search for process builder

-->Created the connection between two objects in a way such that if there is any change in field of one object it has to be reflected in the other object. This is done using Process Builder with the condition of "ready to join" field is checked on the application form object then we need to create student record with those details automatically.



Day6-Activities :-

Topic: Milestone / Activities: Create the student record using flow

Detailed Description:

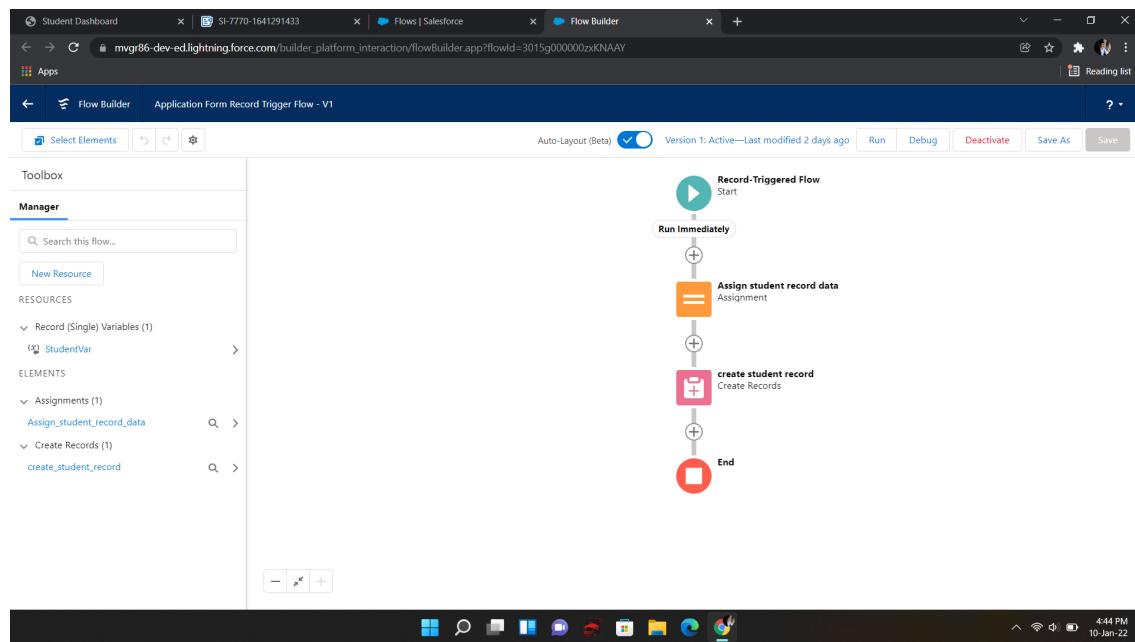
Create the Student Record using Flow

Skill Tags:

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 above built Process is deactivated and then created a flow for the same conditions using Flow Builder. Click setup and then search for flow builder in quick find and select flow builder and click on new flow and then select auto layout and created the flow



Day7-Activities :-

Topic: Milestone / Activities: Intro to Apex proframming

Detailed Description:

What is Apex programming language?

Apex is strongly typed object-oriented, on-demand programming language. It is compiled, stored, and run entirely on the Force.com platform (multi-tenant environment and is very controlled in its invocations and limits).

Apex syntax looks mostly like Java and acts like stored procedures.

Apex allows developers to attach business logic to the record save process

Apex has built-in support for unit test creation and execution.

As a language apex is Integrated, Easy to use, Data focused, Rigorous, Hosted, Multi-tenant aware, automatically up-gradable, easy to test and versioned.

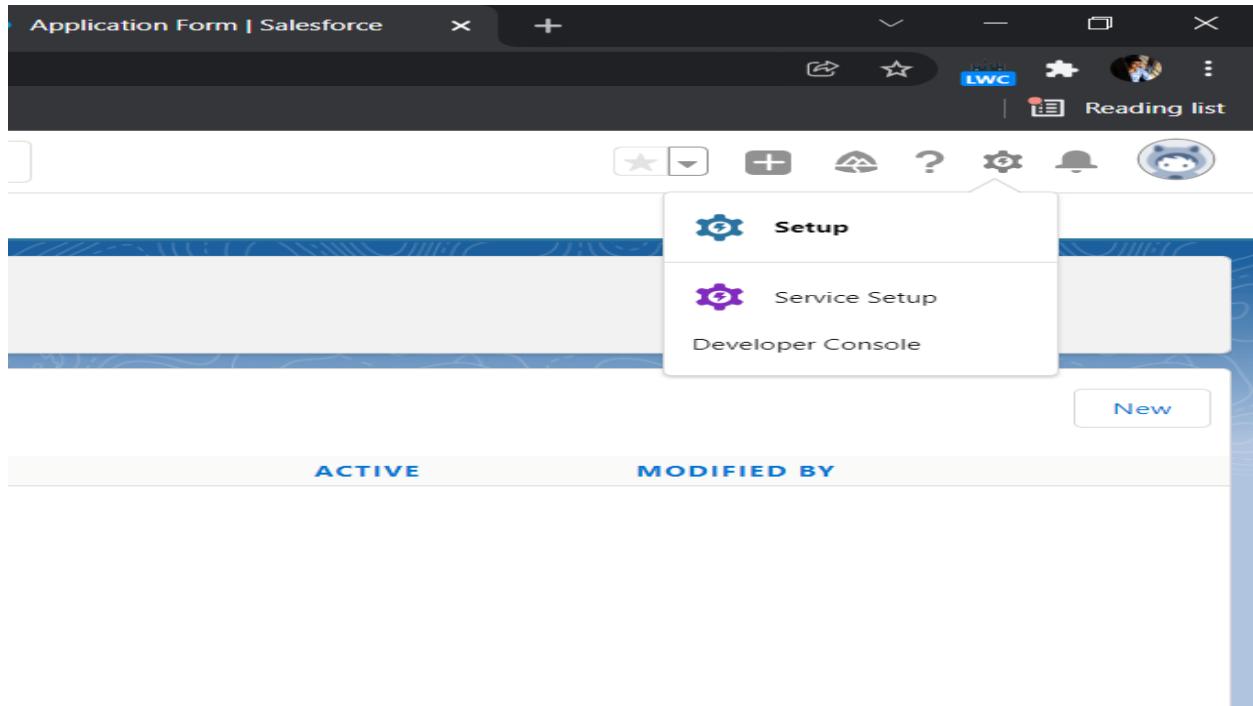
There are only two places you can write code in Salesforce:

Sandbox – most people will want to code here! Sandboxes are developer-friendly copies of your normal org and they're easy to create.

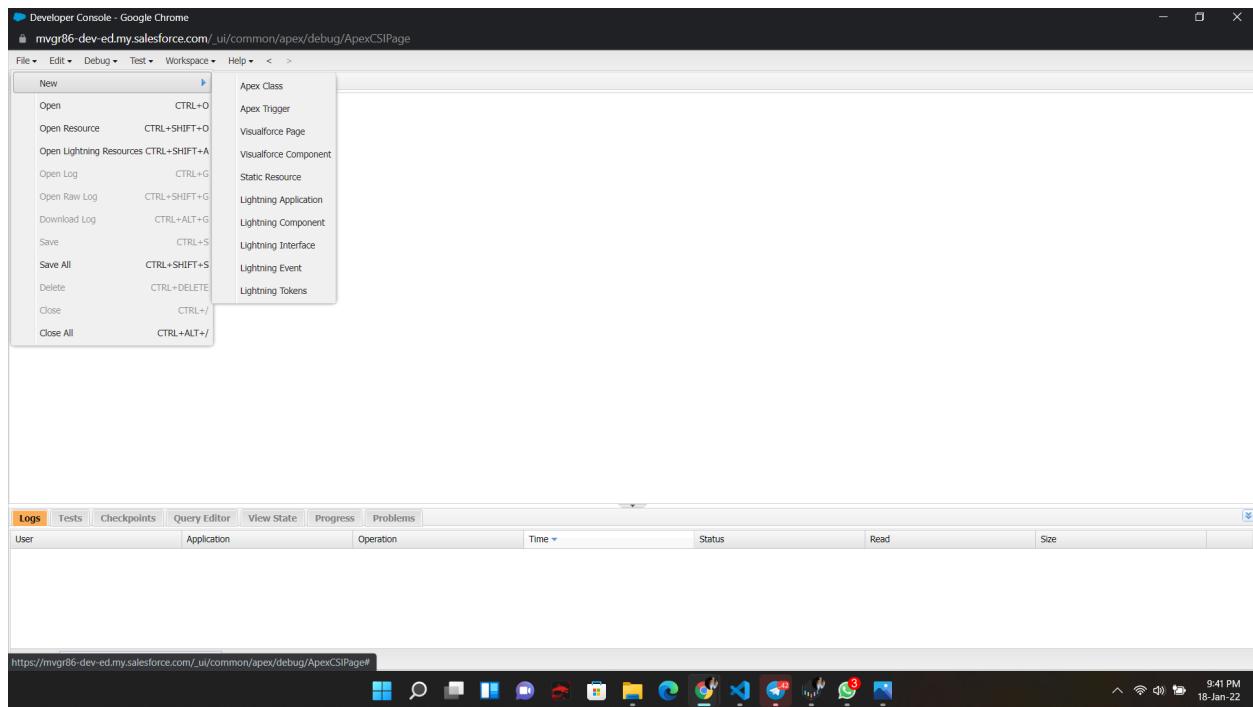
Developer Edition – if you don't have an instance of Salesforce to call home, you can practice here.

You can follow along in either version! If you want to eventually deploy into your normal "production" org though, you'll want to use a sandbox.

To open up the trigger coding UI in either version, navigate here:



and select developer console where we can write our apex classes and triggers and all programming part



Day8-Activities :-

```
1 Public class ApplicationBatchTest implements Database.Batchable<sObject>{
2
3     public Integer totalForms = 0; // total no of application form
4     public Integer totalConvertedForms = 0; // total no of students
5     public Database.QueryLocator start(Database.BatchableContext bc){
6
7         String applicationQuery = 'select id, Name, Ready_To_Join__c from Application_Form__c';
8         return Database.getQueryLocator(applicationQuery);
9     }
10    public void execute(Database.BatchableContext bc, List<Application_Form__c> formList){
11
12        for(Application_Form__c af : formList){
13            totalForms++;
14            if(af.Ready_To_Join__c){
15                totalConvertedForms++;
16            }
17        }
18    }
19 }
20 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

User Application Operation Time Status Read Size

Filter Click here to filter the log list

2--> Created a Scheduler Class in the developer class

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `mvrgr86-dev-ed.my.salesforce.com/ui/common/apex/debug/ApexCSIPage`. The tab title is "Developer Console - Google Chrome". The code editor window displays an Apex class named `applicationschedule` which implements `Schedulable`. The class has a single method `execute` that creates a test object `ApplicationBatchTest abt = new ApplicationBatchTest();` and executes it with a batch size of 400. The API version is set to 53. Below the code editor is a navigation bar with tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Logs tab is selected. A table below the navigation bar lists logs, with columns for User, Application, Operation, Time, Status, Read, and Size. At the bottom of the screen is a Windows taskbar showing various application icons.

```
1 public class applicationschedule implements Schedulable{  
2  
3  
4  
5 public void execute(SchedulableContext sc){  
6  
7     ApplicationBatchTest abt = new ApplicationBatchTest();  
8  
9     Database.executeBatch(abt, 400); // 200 to 2000  
10  
11  
12 }  
13  
14 }  
15 }
```

Then went to Setup and searched for Apex Classes and click on Schedule Apex to schedule a Apex Class according to our requirements. After scheduling this scheduled class will be visible in Apex Jobs.

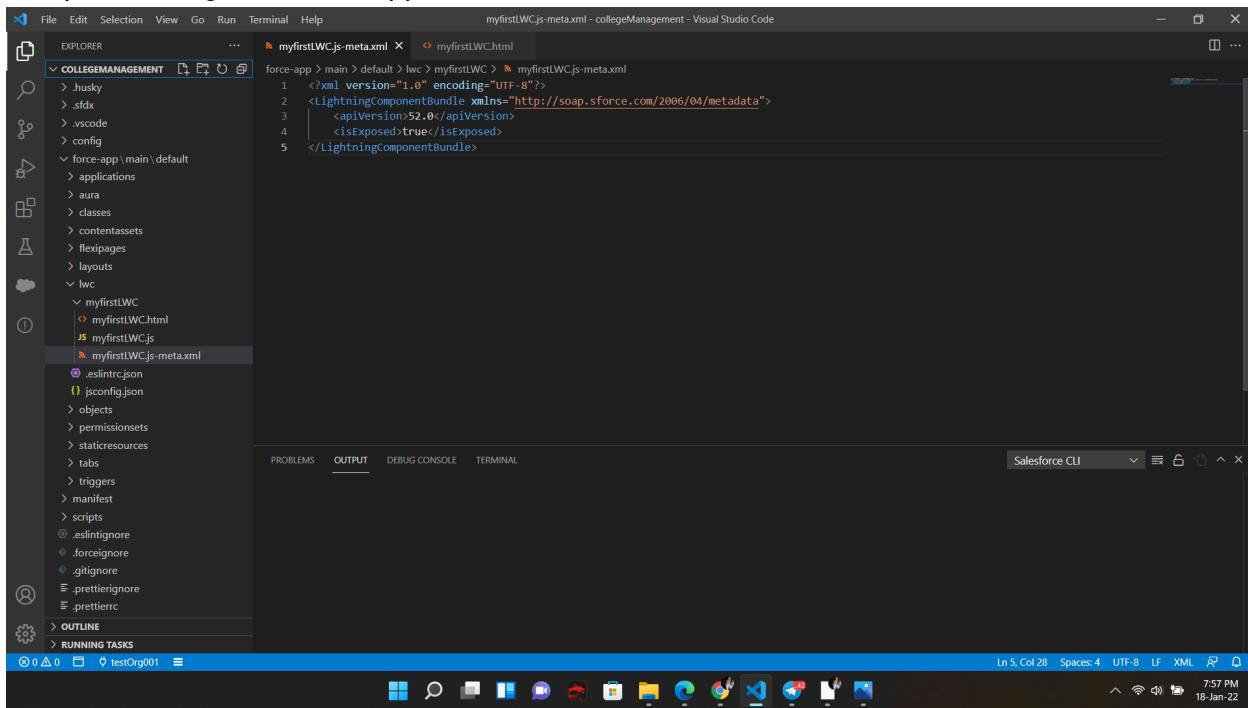
Day9-Activities :-

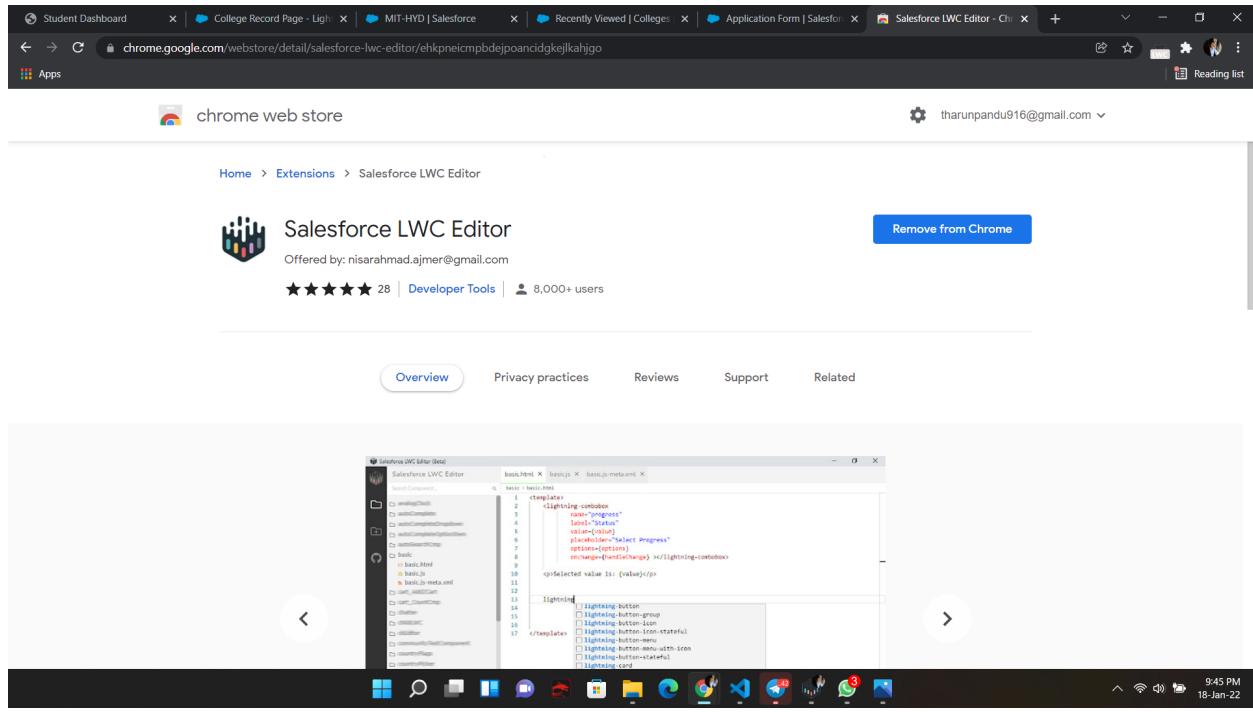
Topic: Milestone / Activities: overview of LWC and installation of visual studio

Detailed Description:

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





Day10-Activities :-

Topic: Milestone / Activities: create college data table components(APEX CLASS FILE,HTML FILE)

Detailed Description:

Create College DataTable Component(APEX CLASS FILE)

The screenshot shows the Salesforce LWC Editor (Beta) interface. On the left, there's a sidebar with icons for file operations, a search bar, and sections for 'APEX CLASSES' and 'LIGHTNING WEB COMPONENTS'. Under 'APEX CLASSES', there are three items: 'ApplicationBatchTest', 'applicationschedule', and 'getApplicationDetails'. Under 'LIGHTNING WEB COMPONENTS', there is one item: 'collegeDataTable'. The main area is a code editor with tabs for 'collegeDataTable.html', 'getApplicationDetails.cls', 'collegeDataTable.js', and 'collegeDataTable.js-meta.xml'. The currently selected tab is 'getApplicationDetails.cls', which contains the following Apex code:

```
1 public class getApplicationDetails {
2     @AuraEnabled(cacheable=true)
3     public static List<Application_Form__c> getapplicationvalues(id collegeId)
4     {
5         List<Application_Form__c> formlist = [SELECT ID, College_Fees__c, Application_Form_ID__c, Name,
6             Date_of_Birth__c, Email__c, Hostel_Fees__c, Student_Name__c, Phone__c FROM
7             Application_Form__c WHERE College__c =:CollegeId];
8
9         return formList;
10    }
11}
12
13
```

Create College DataTable Component(HTML FILE)

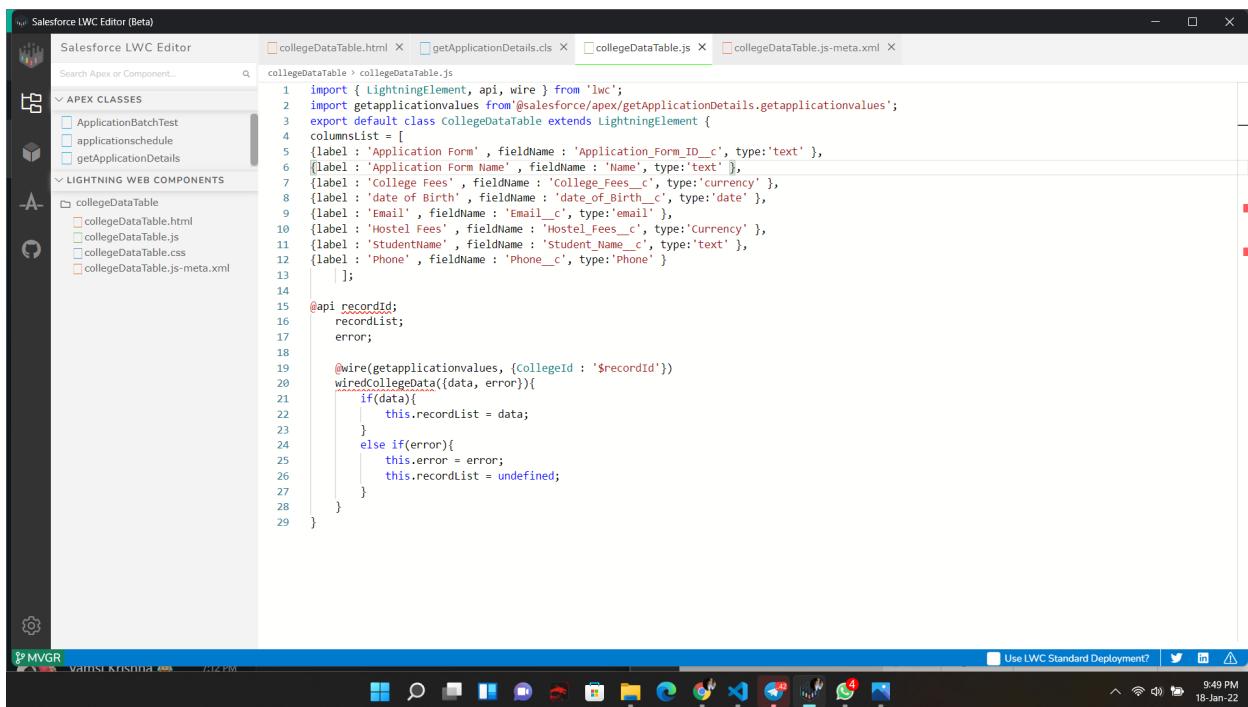
The screenshot shows the Salesforce LWC Editor (Beta) interface. The sidebar is identical to the previous screenshot, showing 'collegeDataTable.html' under 'LIGHTNING WEB COMPONENTS'. The main code editor has tabs for 'collegeDataTable.html', 'getApplicationDetails.cls', 'collegeDataTable.js', and 'collegeDataTable.js-meta.xml'. The currently selected tab is 'collegeDataTable.html', which contains the following HTML code:

```
<template>
1 <h1> College and Application form list table </h1>
2
3 <template if:true={recordList}>
4     <lightning-data-table
5         key-field="id"
6         data={recordList}
7         show-row-number-column
8         hide-checkbox-column
9         columns={columnsList}
10    >
11    </lightning-data-table>
12 </template>
13
14 <template if:true={error}>
15     | {error}
16 </template>
17 </template>
18
19 </template>
```

Day11-Activities :-

Topic: Milestone / Activities: create college data table components(JAVA SCRIPT FILE,XML FILE)

Detailed Description: Create College DataTable Component(JAVA SCRIPT FILE)



The screenshot shows the Salesforce LWC Editor (Beta) interface. On the left, there's a sidebar with icons for file operations and a search bar. Below it, a tree view shows the project structure: APEX CLASSES (ApplicationBatchTest, applicationschedule, getApplicationDetails) and LIGHTNING WEB COMPONENTS (collegeDataTable, collegeDataTable.html, collegeDataTable.js, collegeDataTable.css, collegeDataTable.js-meta.xml). The main area displays the code for collegeDataTable.js. The code is a JavaScript file with imports from 'lwc' and '@salesforce/apex/getApplicationDetails.getApplicationValues'. It defines a class CollegeDataTable extending LightningElement with a columnList property containing field definitions for Name, Application Form Name, Application Fees, Date of Birth, Email, Hostel Fees, Studentname, and Phone. It also includes a @wire annotation for getApplicationValues and a wiredCollegeData method to handle the response.

```
import { LightningElement, api, wire } from 'lwc';
import getApplicationValues from '@salesforce/apex/getApplicationDetails.getApplicationValues';
export default class CollegeDataTable extends LightningElement {
    columnList = [
        {label : 'Application Form' , fieldName : 'Application_Form_ID__c', type:'text' },
        {label : 'Application Form Name' , 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' },
        {label : 'Phone' , fieldName : 'Phone__c', type:'Phone' }
    ];
    @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;
        }
    }
}
```

Create College DataTable Component(META FILE)

The screenshot shows the Salesforce LWC Editor (Beta) interface. The left sidebar displays the project structure under 'APEX CLASSES' and 'LIGHTNING WEB COMPONENTS'. In the 'LIGHTNING WEB COMPONENTS' section, there is a folder named 'collegeDataTable' containing four files: 'collegeDataTable.html', 'collegeDataTable.js', 'collegeDataTable.css', and 'collegeDataTable.js-meta.xml'. The main editor area shows the XML content of 'collegeDataTable.js-meta.xml'. The code is as follows:

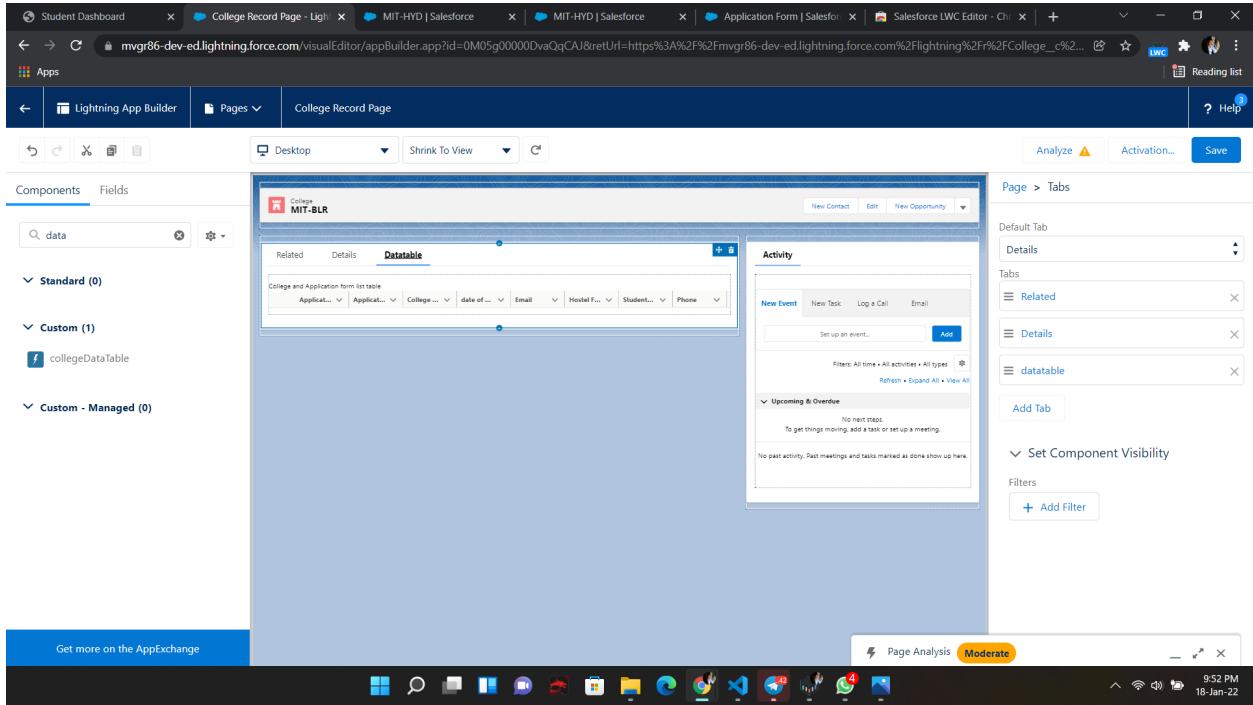
```
<?xml version="1.0"?>
<lightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
<apiVersion>51.0</apiVersion>
<isExposed>true</isExposed>
<targets>
<target>lightning__RecordPage</target>
<target>lightning__AppPage</target>
<target>lightning__HomePage</target>
</targets>
</lightningComponentBundle>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

The status bar at the bottom indicates the user is 'Vamsi Krishna' and shows system information like battery level, signal strength, and date/time.

Day12-Activities :-

As we have done all the tasks now we need to check in the college management application app which we have created go to the college management application from the app launcher now-->

click edit page then we have to create another tab for data table And then drag the college data table in that tab now you can see the data table



As we have completed the all things

Student Dashboard | College Record Page - Light | MIT-HYD | Salesforce | MIT-HYD | Salesforce | Application Form | Salesforce | Salesforce LWC Editor - Ch | +

mvgr86-dev-ed.lightning.force.com/lightning/r/College__c/a005g000031dUseAAE/view

All Search Colleges and more... Apps

College Management Colleges Application Forms Students Subjects

New Contact Edit New Opportunity

Related Details **Datable**

College and Application form list table

Application ID	Venu	College Fees	Date of Birth	Email	Hostel Fees	Student Name	Phone
1 F-000001	Venu	\$145,000.00	Dec 29, 2021	venu@venu...	155000	venu	2458369745

Activity

New Settings for Record Pages

Personalize your Activities view

You can choose to keep track of current and past tasks and events in the Open Activities and Activity History related lists or the Activity timeline.

Tell Me More Show Me How

javascript:void(0)

Windows taskbar icons: File Explorer, Task View, Search, Start, Taskbar settings, Edge, Google Chrome, Microsoft Edge, Microsoft Store, Mail, Photos, OneDrive, OneNote, Word, Excel, Powerpoint, Teams, WhatsApp, and a file icon.

9:52 PM 18-Jan-22

