

Day 1: 03/01/2022, Monday

Topic : Create your salesforce developer org to get started

Milestone/Activities : Creating developer account Account activation Login to your salesforce account

Detailed Description : First I sign up the developers salesforce.com/with all required information , then I clicked sign up button after I got email for activating the developer account ,then I verified the account , after that I logged into the salesforce account ,after I created college management application in lightening app through app manager.

Upload the screenshot the Milestone / Activities :

The screenshot shows the Trailhead Salesforce user profile page at trailhead.salesforce.com/users/profiles/orgs. The user is Rikit Maheshwari, with 111 badges and 66,650 points. The page displays a list of connected developer editions:

- Connected Orgs (12)**
 - Developer Edition** (LAST USED)
 - Username: rikit@jcrc.com
 - Type: Developer Edition
 - Connected: 1/16/2022
 - Last Activity: Connected on 1/16/2022

[Rename](#) [Disconnect](#) [Launch](#)
 - New New**
 - Username: creative-panda-be40in.com
 - Type: Trailhead Playground
 - Created: 12/17/2021
 - Last Activity: Set Up and Manage Shield Platform Encryption on 12/19/2021

[Rename](#) [Disconnect](#) [Launch](#)
 - Creative Bear Playground**
 - Username: cunning-wolf-s53jw.com
 - Type: Trailhead Playground
 - Created: 10/19/2021
 - Last Activity: Created on 10/19/2021

[Rename](#) [Disconnect](#) [Launch](#)

The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'Recently Viewed | Colleges | Sales...' and displays a list of recently viewed college records. The URL is jeccuniversity86-dev-ed.lightning.force.com/lightning/o/College_c/list?filterName=Recent. The browser's address bar also shows other tabs like 'College | Salesforce' and 'Apex Classes | Salesforce'. The page header includes a cloud icon, the title 'College Management...', and navigation links for 'Colleges', 'Applications', 'Students', and 'Subjects'. A search bar at the top right says 'Search Colleges and more...'. The main content area is titled 'Recently Viewed' and shows a table with 6 items, all of which are 'Record Info' type. The items listed are:

		Record Info
1	<input type="checkbox"/>	MIT-CCU
2	<input type="checkbox"/>	MIT-DEL
3	<input type="checkbox"/>	MIT-MAA
4	<input type="checkbox"/>	MIT-MUM
5	<input type="checkbox"/>	MIT-BLR
6	<input type="checkbox"/>	MIT-HYD

At the bottom of the browser window, there is a taskbar with icons for various applications like File Explorer, Edge, and Google Chrome. The system tray shows the date (1/16/2022), time (8:16 PM), and weather (11°C, Cloudy).

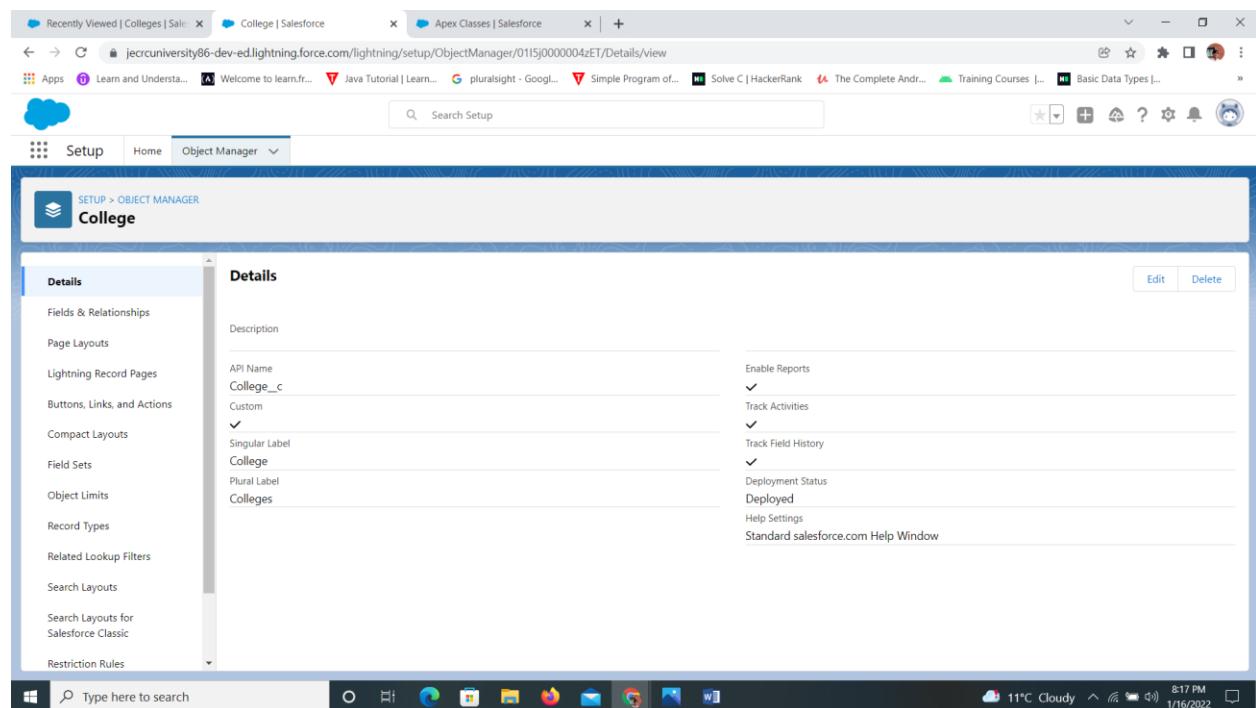
Day 2 : 04/01/2022

Topic : Custom object creation

Milestone / Activities : Create the custom object for the application Create fields on college object

Detailed Description : First I sign up the developers.salesforce.com/with all required information ,then I clicked setup and later object manager after I created custom objects .

Upload the screenshot the Milestone/Activities: COLLEGE



Screenshot of the Salesforce Object Manager interface for the 'College' object.

The page shows the 'Fields & Relationships' section with 9 items, sorted by Field Label.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Capacity Of Students	Capacity__c	Picklist	College Name	
College Fees	College_Fees__c	Currency(7, 2)		
College Name	College_Name__c	Picklist		
Created By	CreatedBy	Lookup(User)		
Email	Email__c	Picklist	College Name	
Hostel Fees	Hostel_Fees__c	Currency(6, 2)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User/Group)		✓
Record Info	Name	Text(80)		✓

Navigation pane on the left includes: Details, Fields & Relationships, 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.

System bar at the bottom shows: Type here to search, taskbar icons, and system status: 11°C Cloudy, 8:17 PM, 1/16/2022.

Day 3 : 05/01/2022

Topic : custom object creation

Milestone/Activities : create fields on application form object create fields on student object create fields on subject object

The screenshot shows the Salesforce Setup interface with the following details:

Object Manager > **Application Form**

Details tab selected.

Fields & Relationships sidebar:

- 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

Details section:

Description	Enable Reports
API Name Application_Form__c	✓
Custom ✓	Track Activities
Singular Label Application Form	Track Field History
Plural Label Applications	Deployment Status
	Deployed
	Help Settings
	Standard salesforce.com Help Window

Fields & Relationships tab selected.

Fields & Relationships table:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text(255)		
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)		
Hostel Fees	Hostel_Fees__c	Formula (Currency)		

The screenshot shows a Microsoft Edge browser window with three tabs open:

- Recently Viewed | Applications
- Application Form | Salesforce
- Apex Classes | Salesforce

The main content area displays the 'Recently Viewed' application list under the 'Applications' tab. The list includes:

Form Number
1 Form-00000003
2 Form-00000002
3 Form-00000001

Below this, there is a large, mostly blank white space.

The screenshot shows a Microsoft Edge browser window with three tabs open:

- Recently Viewed | Applications
- Students | Salesforce
- Apex Classes | Salesforce

The main content area displays the 'Students' object setup details under the 'Object Manager' tab. The left sidebar shows navigation options like Details, Fields & Relationships, Page Layouts, etc. The right panel shows the following details:

Details	
Description	
API Name	Students__c
Custom	✓
Singular Label	Students
Plural Label	Students
Enable Reports	✓
Track Activities	✓
Track Field History	✓
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

Below this, there is a large, mostly blank white space.

Screenshot of the Salesforce Object Manager interface showing the Fields & Relationships for the Students object.

Fields & Relationships

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	CreatedBy	Lookup(User)		
Date Of Birth	DateOfBirth__c	Date		
Guardian Name	Guardian_Name__c	Text(30)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User;Group)		✓
Phone	Phone__c	Phone		

Recently Viewed

3 items • Updated a few seconds ago

Students Name
1 Adish
2 Mohit
3 Ritik

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes tabs for Recently Viewed, Subjects, Apex Classes, and a new tab. Below the navigation is a search bar and a toolbar with various icons. The main content area has a header 'SETUP > OBJECT MANAGER Subjects'. On the left is a sidebar with a 'Details' section containing links to Fields & Relationships, 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 right side displays the 'Details' tab for the Subjects object, which includes fields for Description, API Name (Subjects__c), Singular Label (Subjects), Plural Label (Subjects), Enable Reports (checked), Track Activities (checked), Track Field History (checked), Deployment Status (Deployed), and Help Settings (Standard salesforce.com Help Window). There are 'Edit' and 'Delete' buttons at the top right.

This screenshot continues from the previous one, showing the 'Fields & Relationships' tab for the Subjects object. The sidebar remains the same. The main content area shows a table titled 'Fields & Relationships' with 8 items, sorted by Field Label. The columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data in the table is as follows:

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)		✓
Paper 1	Paper_1__c	Picklist		
Paper 2	Paper_2__c	Picklist		
Students	Students__c	Lookup(Students)		✓
Subject ID	Subject_ID__c	Auto Number		
Subjects Name	Name	Text(80)		✓

Day 4 : 06/01/2022

Topic : Adding Business Logic To Application

Milestone / Activities : Create Global Picklist Value Sets ,Create Feild Dependencies

Detailed Description :First I signed up the developers.salesforces.com/with all required information,then I created global picklist values college ,paper 1 and paper 2 in picklist value set then I created field dependencies ,after that I created validation rule on the college object .

Upload the screenshot the Milestone/Activities:Global picklist values

The screenshot shows the Salesforce Setup interface with the following details:

Page Title: Picklist Value Sets

Section: Global Value Sets

Table Headers: Action, Label, Description

Data Rows:

Action	Label	Description
Edit Del	College.Name	
Edit Del	Paper_1	
Edit Del	Paper_2	

Deleted Global Value Sets (0)

Page Title: Picklist Value Sets

Section: Global Value Set

Table Headers: Values, Inactive Values, Fields Where Used

Data Rows:

Label	Name
College Name	College_Name

Section: Picklist Values Used

Table Headers: Active and inactive picklist values, Edit, Delete

Data Rows:

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	MIT-HYD	MIT-HYD	<input type="checkbox"/>	Assigned dynamically	RITIK MAHESHWARI, 1/9/2022, 3:16 AM
Edit Del Deactivate	MIT-BLR	MIT-BLR	<input type="checkbox"/>	Assigned dynamically	RITIK MAHESHWARI, 1/9/2022, 3:16 AM
Edit Del Deactivate	MIT-MUM	MIT-MUM	<input type="checkbox"/>	Assigned dynamically	RITIK MAHESHWARI, 1/9/2022, 3:16 AM
Edit Del Deactivate	MIT-MAA	MIT-MAA	<input type="checkbox"/>	Assigned dynamically	RITIK MAHESHWARI, 1/9/2022, 3:16 AM
Edit Del Deactivate	MIT-DEL	MIT-DEL	<input type="checkbox"/>	Assigned dynamically	RITIK MAHESHWARI, 1/9/2022, 3:16 AM

Recently Viewed | Students | Sales | Picklist Value Sets | Salesforce | Apex Classes | Salesforce

jeccruniversity86-dev-ed.lightning.force.com/lightning/setup/Picklists/page?address=%2F0N15j000000L8vI

Apps Learn and Understand... Welcome to learn.fr... Java Tutorial | Learn... pluralsight - Google Simple Program of... Solve C | HackerRank The Complete Andr... Training Courses J... Basic Data Types J...

Setup Home Object Manager

Q Search Setup

Cloud icon

Q picklist

v Data

- Picklist Settings
- State and Country/Territory Picklists

v Objects and Fields

- Picklist Value Sets

Didn't find what you're looking for? Try using Global Search.

SETUP Picklist Value Sets

Global Value Set

< Back to List

Values (6) | Inactive Values (0) | Fields Where Used (0) | Help for this Page

Global Value Set Detail

Information

Label	Paper 1
Name	Paper_1
Description	

Edit Delete

Picklist Values Used

Active and inactive picklist values 4 (1,000 max)

Edit Delete

Values

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	APEX	APEX	<input type="checkbox"/>	Assigned dynamically	RITIK MAHESHWARI 1/9/2022, 8:11 AM
Edit Del Deactivate	C	C	<input type="checkbox"/>	Assigned dynamically	RITIK MAHESHWARI 1/9/2022, 8:11 AM
Edit Del Deactivate	C++	C++	<input type="checkbox"/>	Assigned dynamically	RITIK MAHESHWARI 1/9/2022, 8:11 AM
Edit Del Deactivate	JAVA	JAVA	<input type="checkbox"/>	Assigned dynamically	RITIK MAHESHWARI 1/9/2022, 8:11 AM

javascript:srcUp(%27%2F0N15j000000L8vI%3Fisdtpp%3Dp1%27);

11°C Cloudy 8:19 PM 1/16/2022

Recently Viewed | Students | Sales | Picklist Value Sets | Salesforce | Apex Classes | Salesforce

jeccruniversity86-dev-ed.lightning.force.com/lightning/setup/Picklists/page?address=%2F0N15j000000L8vI

Apps Learn and Understand... Welcome to learn.fr... Java Tutorial | Learn... pluralsight - Google Simple Program of... Solve C | HackerRank The Complete Andr... Training Courses J... Basic Data Types J...

Setup Home Object Manager

Q Search Setup

Cloud icon

Q picklist

v Data

- Picklist Settings
- State and Country/Territory Picklists

v Objects and Fields

- Picklist Value Sets

Didn't find what you're looking for? Try using Global Search.

SETUP Picklist Value Sets

Global Value Set

< Back to List

Values (3) | Inactive Values (0) | Fields Where Used (0) | Help for this Page

Global Value Set Detail

Information

Label	Paper2
Name	Paper2
Description	

Edit Delete

Picklist Values Used

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

Edit Delete

Values

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	ENGLISH	ENGLISH	<input type="checkbox"/>	Assigned dynamically	RITIK MAHESHWARI 1/9/2022, 8:11 AM
Edit Del Deactivate	MATHEMATICS	MATHEMATICS	<input type="checkbox"/>	Assigned dynamically	RITIK MAHESHWARI 1/9/2022, 8:11 AM
Edit Del Deactivate	STATISTICS	STATISTICS	<input type="checkbox"/>	Assigned dynamically	RITIK MAHESHWARI 1/9/2022, 8:11 AM

Inactive Values

javascript:srcUp(%27%2F0N15j000000L8vI%3Fisdtpp%3Dp1%27);

11°C Cloudy 8:19 PM 1/16/2022

Salesforce screenshot showing the 'College Field Dependencies' page.

The page title is 'College Field Dependencies' under 'College' in 'OBJECT MANAGER'. The sub-header says: 'This page allows you to define dependencies between fields (e.g., dependent picklists).'

The table displays the following data:

Action	Controlling Field	Dependent Field	Modified By
Edit Del	College Name	Email	RITIK MAHESHWARI, 1/9/2022, 3:41 AM
Edit Del	College Name	Capacity Of Students	RITIK MAHESHWARI, 1/9/2022, 7:16 AM

The left sidebar shows navigation links for 'Fields & Relationships' including: Details, 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.

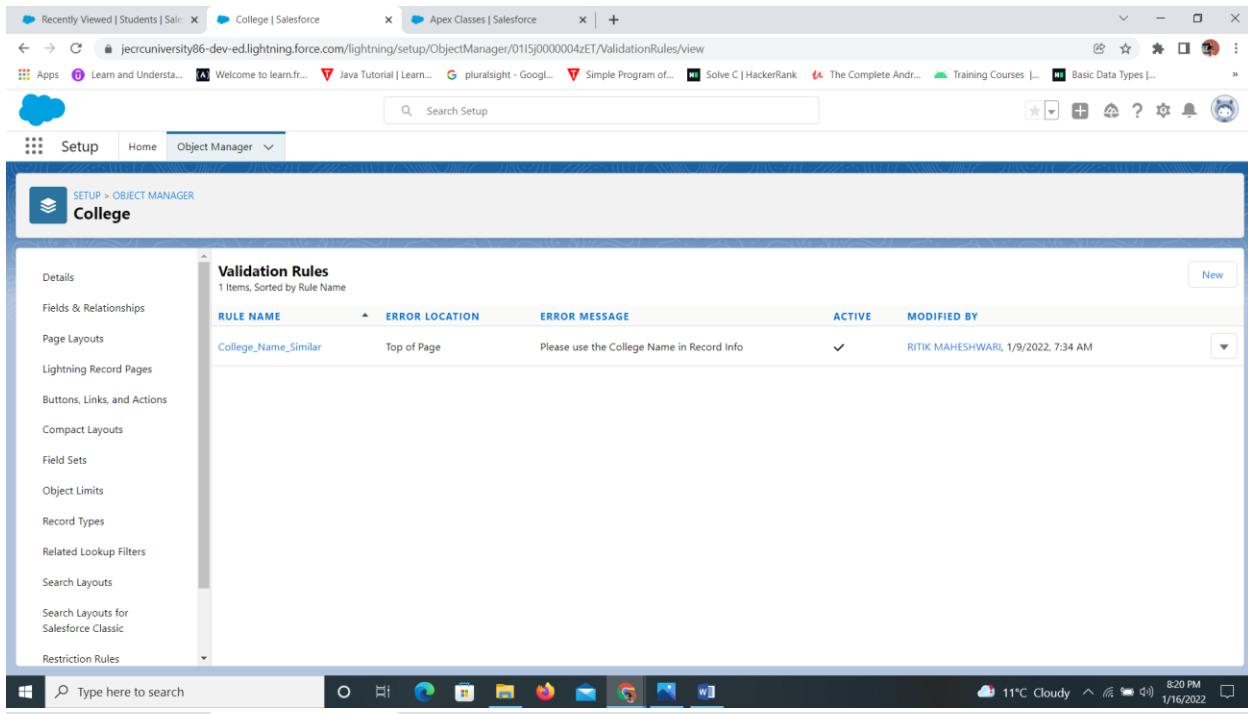
Type here to search

11°C Cloudy 8:19 PM 1/16/2022

Day 5 : 07/01/2022

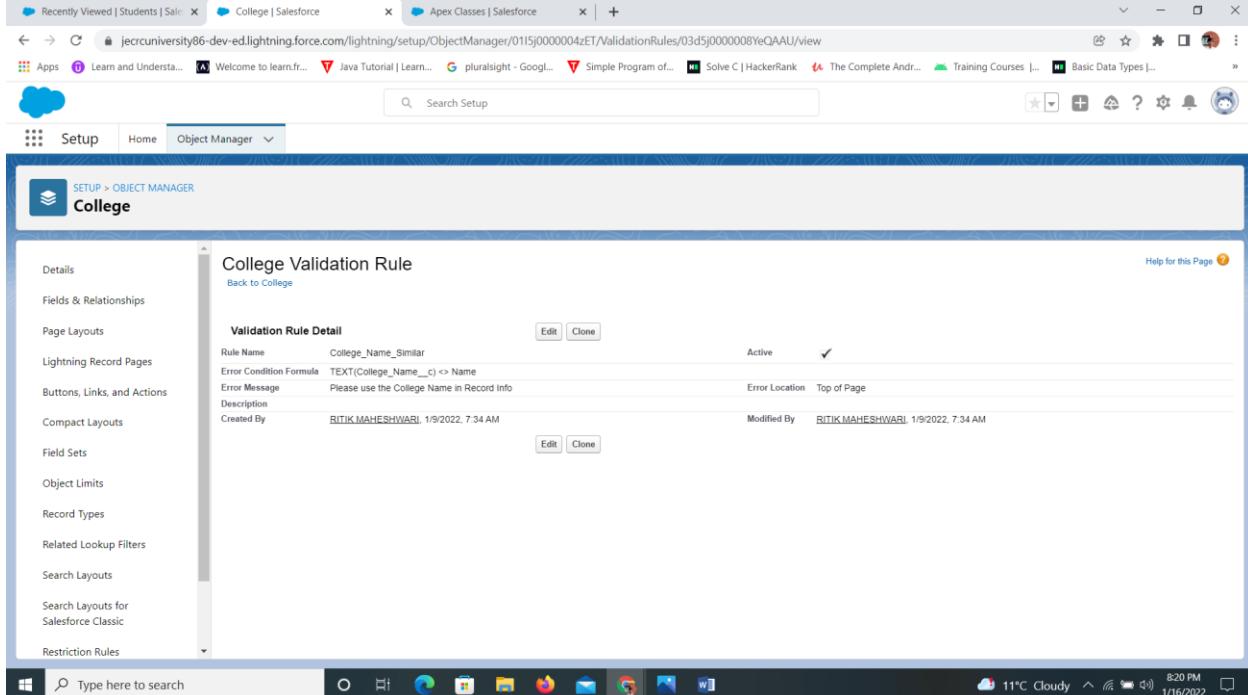
Topic: Adding business logic to application

Milestone/Activities: creating validation rules Validation rules: college object



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 displays a table titled "Validation Rules" with one item listed:

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
College_Name_Similar	Top of Page	Please use the College Name in Record Info	✓	RITIK MAHESHWARI, 1/9/2022, 7:34 AM



The screenshot shows the details of the "College Validation Rule" named "College_Name_Similar". The rule is active and located at the top of the page. The validation formula is `TEXT(College_Name__c) <> Name`. The error message is "Please use the College Name in Record Info". The rule was created by RITIK MAHESHWARI on 1/9/2022, 7:34 AM.

Screenshot of the Salesforce Application Form Validation Rules page.

Validation Rules

Rule Name	Error Location	Error Message	Active	Modified By
Stop_modify_Application_Form	Top of Page	Once the form is submitted, it cannot be edited. Please enter proper information.	✓	RITIK MAHESHWARI, 1/10/2022, 6:56 AM

Application Form Validation Rule

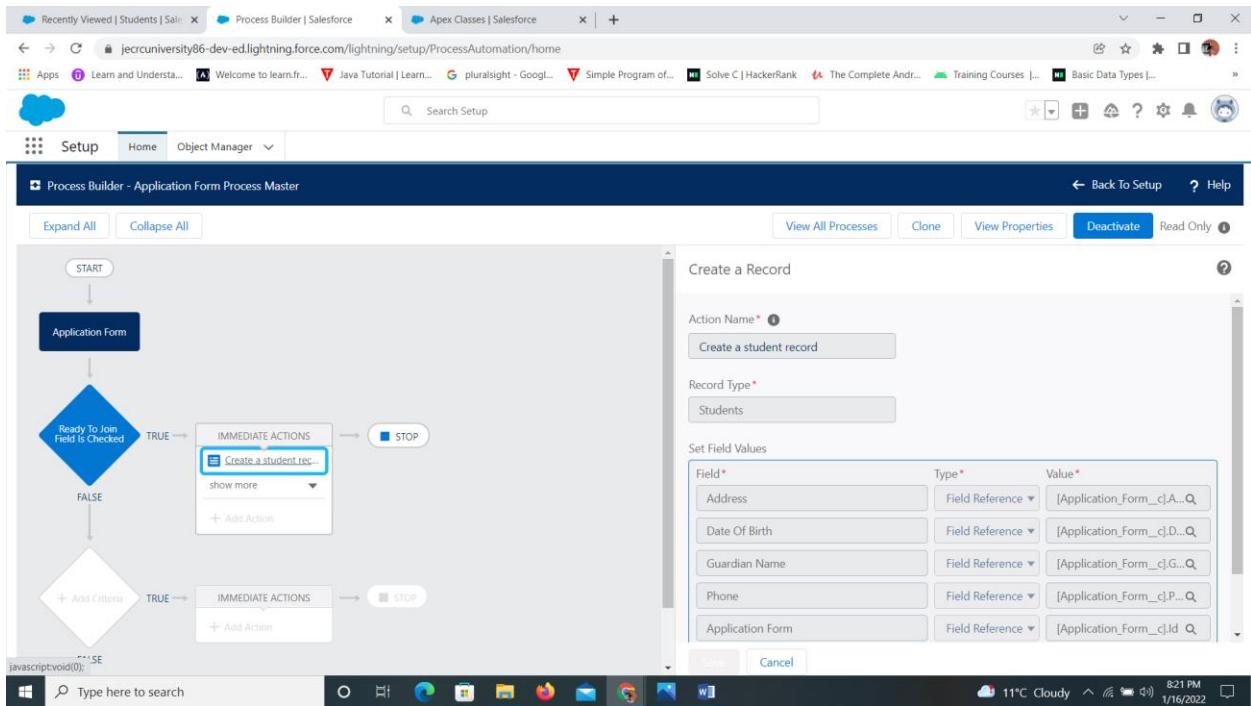
Validation Rule Detail

Rule Name	Stop_modify_Application_Form	Edit Clone
Error Condition Formula	<pre>AND(Ready_To_Join__c == true, ISCHANGED(Address__c), ISCHANGED(College__c), ISCHANGED(dateOfBirth__c), ISCHANGED(DOB__c), ISCHANGED(Guardian.Name__c), ISCHANGED(Phone__c))</pre>	Active ✓
Error Message	Once the form is submitted, it cannot be edited. Please enter proper information.	Error Location Top of Page
Description		
Created By	RITIK MAHESHWARI, 1/10/2022, 6:56 AM	Modified By RITIK MAHESHWARI, 1/10/2022, 6:56 AM

Day 6 :10/01/2022

Topic : Adding business logic to application

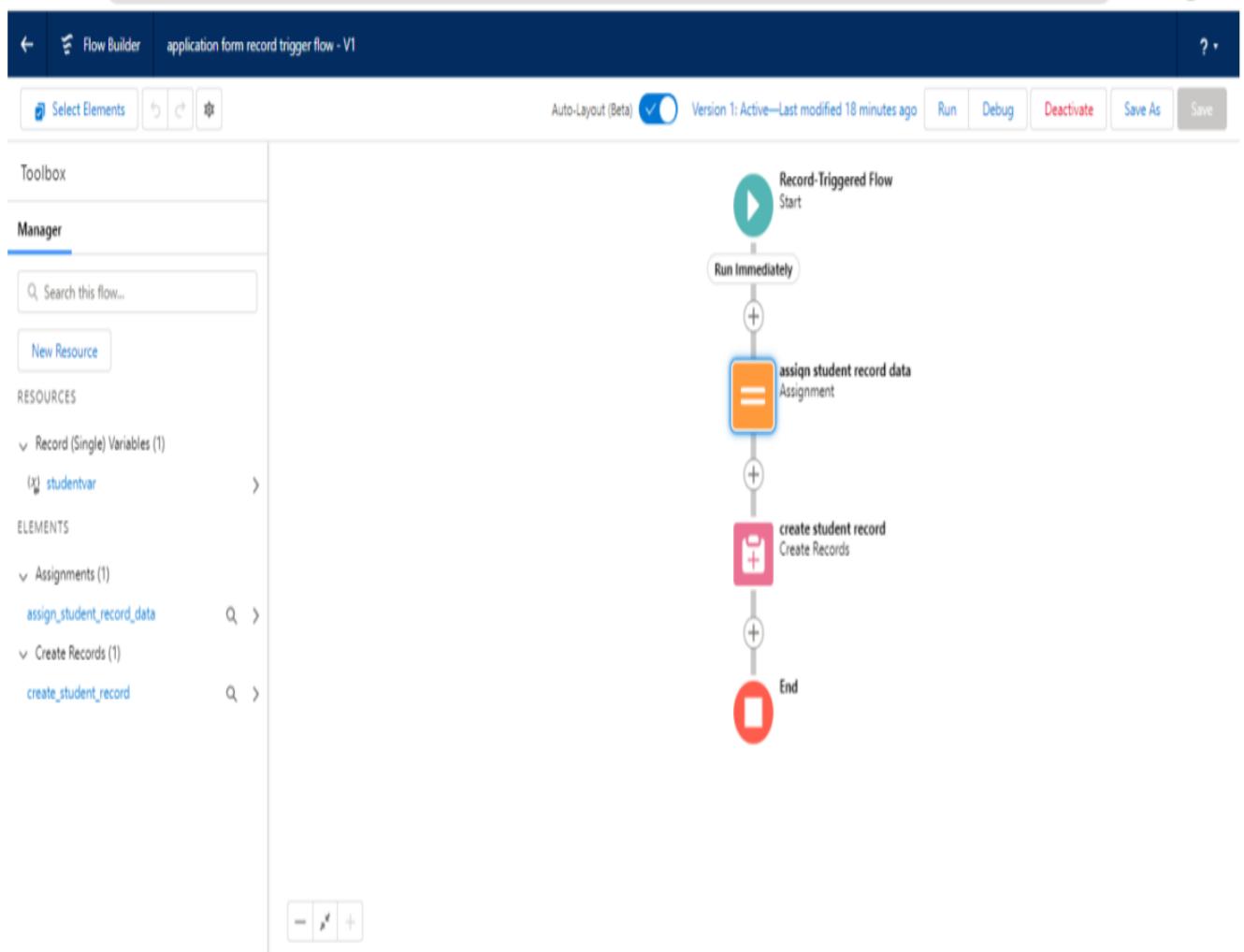
Milestone/Activities: process automation



Day 7: 11/01/2022

Topic: Adding Business Logic To Application

Milestone/Activities: CREATE THE STUDENT RECORD USING FLOW



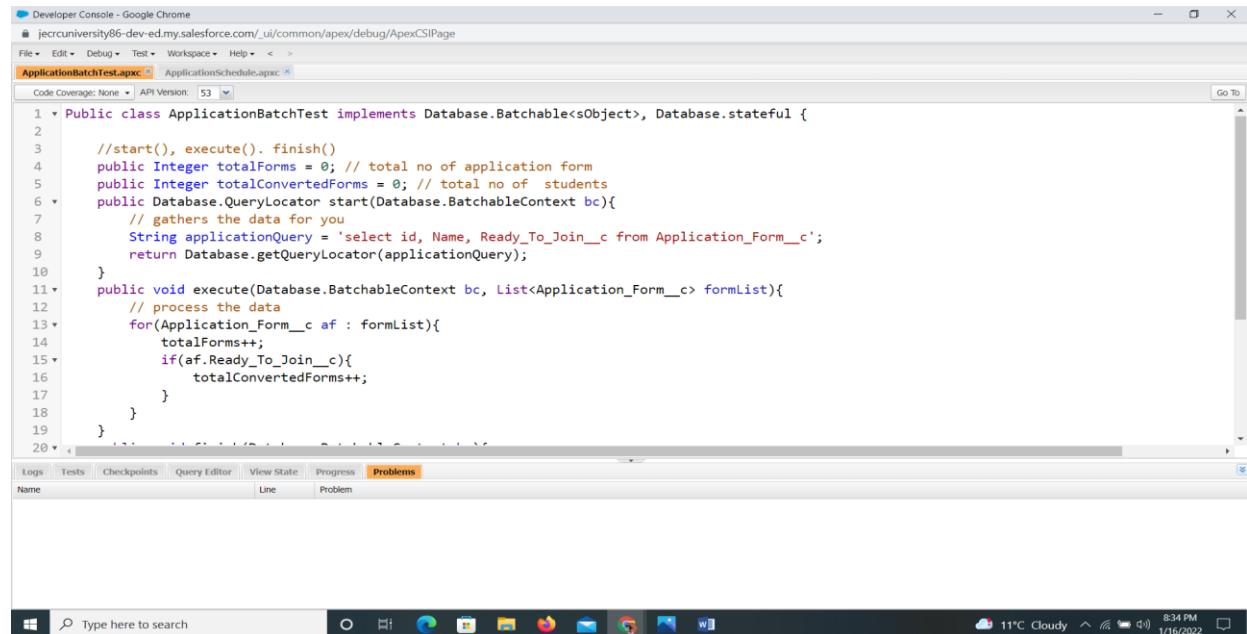
Day 9: 13/01/2022

Topic: Batch Apex

Milestone/Activities: Create Batch Apex For Application Form Create A Schedular Class Detailed

Description : First I Signed up the developers .salesforce.com/with all required information,then I went to developer console after that I created batch apex for application form,after this by same process I created a schedular class. Upload the screenshot the

Milestone/Activities : Create a batch apex for application form



The screenshot shows the Salesforce Developer Console interface. The title bar reads "Developer Console - Google Chrome". The URL is "jccruniversity86-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage". The tabs at the top are "ApplicationBatchTest.apxc" (which is active) and "ApplicationSchedule.apxc". Below the tabs, there are dropdown menus for "File", "Edit", "Debug", "Test", "Workspace", "Help", and "Go To". The main area displays the Apex code for the "ApplicationBatchTest" class:

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

Below the code editor, there are tabs for "Logs", "Tests", "Checkpoints", "Query Editor", "View State", "Progress", and "Problems". The "Problems" tab is selected, showing "Name", "Line", and "Problem" columns. The status bar at the bottom shows "8:34 PM" and "1/16/2022".

Developer Console - Google Chrome
jccruniversity86-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >
ApplicationBatchTest.apxc ApplicationSchedule.apxc

Code Coverage: None API Version: 53 Go To

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

Logs Tests Checkpoints Query Editor View State Progress problems

Name Line Problem

Type here to search 8:34 PM 1/16/2022

Recently Viewed | Students | Sales Flows | Salesforce Flow Builder Apex Classes | Salesforce

Apps Learn and Understand... Welcome to learn fr... Java Tutorial | Learn... pluralsight - Google Simple Program of... Solve C | HackerRank The Complete Andr... Training Courses ... Basic Data Types ...

Cloud Setup Home Object Manager

Quick Find

Setup Home Service Setup Assistant Multi-Factor Authentication Assistant Release Updates Lightning Experience Transition Assistant New Salesforce Mobile App QuickStart Lighting Usage Optimizer ADMINISTRATION Users Data Email PLATFROM TOOLS Apps Feature Settings Einstein Objects and Fields

Apex Classes

Percent of Apex Used: 0.02% You are currently using 1,474 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage | Compile all classes | View: All Create New View

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del Security	ApplicationBatchTest		53.0	Active	1,257	RITIK MAHESHWARI	1/16/2022, 4:57 AM
Edit Del Security	ApplicationSchedule		53.0	Active	217	RITIK MAHESHWARI	1/16/2022, 5:32 AM

Type here to search 8:34 PM 1/16/2022

Day 10:

Topic: LWC

Milestones: Setting up LWC

Salesforce CLI

Like many other programming languages and models, Salesforce includes a command-line interface (CLI). If you've ever used npm, yarn, gradle, or maven, Salesforce CLI will seem familiar to you—just tailor-made for Salesforce development tasks (and if those acronyms look like alphabet soup to you, that's fine, too).

Salesforce CLI allows you to interact with your Salesforce environments in many ways, like retrieving or pushing code or interacting with data. The CLI consists of several plugins. These plugins provide important specific functionality. For example, the salesforcedx plugin provides the ability to interact with Salesforce orgs and their data.

Complete installation instructions.

1. Install the CLI from <https://developer.salesforce.com/tools/sfdxcli>.
2. Confirm the CLI is properly installed and on the latest version by running the following command from the command line.

sfdx update

You should see output like **sfdx-cli: Updating CLI...**

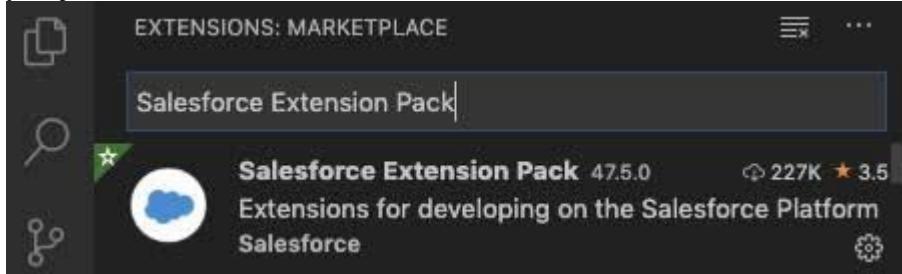
Visual Studio Code

Visual Studio Code is the go-to code editor for Salesforce developers. It is free, open-source, and available for Windows, Linux, and macOS. Visual Studio Code is a well-established IDE among web developers. Now it's also an effective IDE for building Lightning web components, and Salesforce provides free extensions for Visual Studio Code that simplify your development experience even more.

Follow these instructions to install Visual Studio Code.

1. Download and install the latest version of [Visual Studio Code](#) for your operating system. If you already have Visual Studio Code installed, there's no need to reinstall it.
2. Launch Visual Studio Code.
3. Click the  icon for Extensions in the sidebar.

4. Search for Salesforce Extension Pack and click Install. If you already have it installed, then you just need to click on the Reload button.



5. Press Command + Shift + P on macOS or Ctrl + Shift + P on Windows or Linux to reveal the command palette. In the command palette, type sfdx to display an initial list of available commands.

Done! You've installed all the tools you need for developing your first Lightning web component.

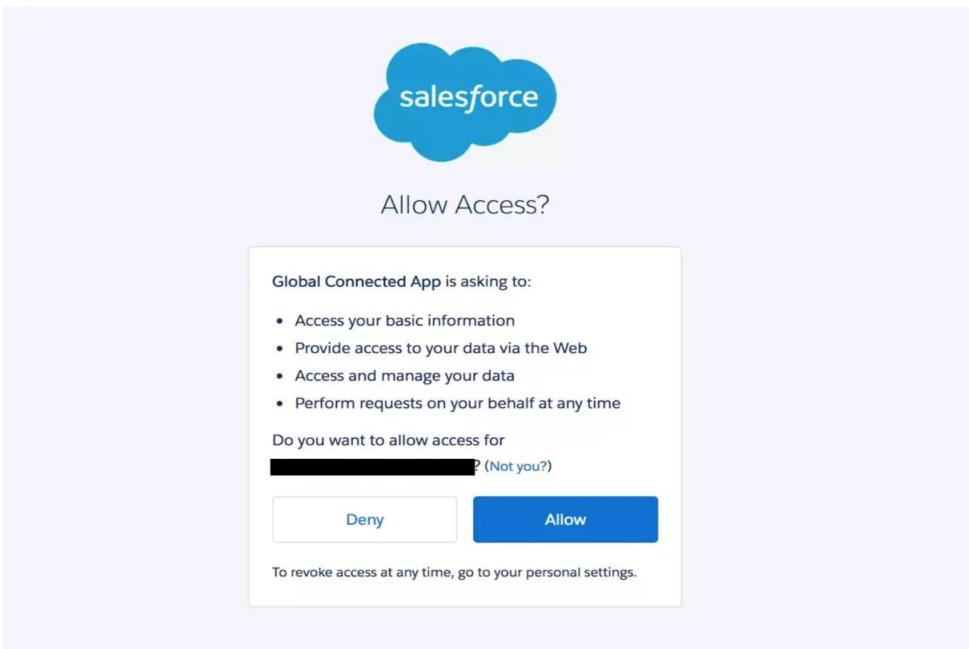
Create a Salesforce DX Project

The basic foundation for interacting with an org using Salesforce CLI is a Salesforce DX project. A project consists of several local configuration files, as well as the code you want to deploy. In Salesforce terms we call this code metadata, which is the foundation of the Salesforce Platform. If you're new to Salesforce, check out Platform Development Basics to learn more about our metadata-driven approach.

1. Open **Visual Studio Code**.
2. Press **Command + Shift + P** on macOS or **Ctrl + Shift + P** on Windows or Linux, then type **create project**. Select **SFDX: Create Project**, and press **Enter**.
3. Leave the default project type selection **Standard** as is, and press **Enter**.
4. Enter **trailhead** as project name, and press **Enter**.
5. Choose a directory on your local machine where the project will be stored. Click **Create Project**.

Authorize Your Dev Hub

1. In **Visual Studio Code**, press **Command + Shift + P** on macOS or **Ctrl + Shift + P** on Windows or Linux.
2. Type **sfdx**.
3. Select **SFDX: Authorize a Dev Hub**.
4. Log in using your Dev Hub org credentials.
5. Click **Allow**.



- After you authenticate in the browser, the CLI remembers your Dev Hub credentials. The success message should look like this:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Starting SFDX: Authorize a Dev Hub

sfdx force:auth:web:login --setdefaultdevhubusername
WARNING: apiVersion configuration overridden at 45.0
Successfully authorized with org ID
You may now close the browser
sfdx force:auth:web:login --setdefaultdevhubusername ended with exit code 0
```

Create a Scratch Org

- In Visual Studio Code, press **Command + Shift + P** on macOS or **Ctrl + Shift + P** on Windows or Linux.
- Type **sfdx**.
- Select **SFDX: Create a Default Scratch Org....**
- Press **Enter** to accept the default `project-scratch-def.json`.
- Press **Enter** to accept the default `trailhead` scratch org alias.
- Press **Enter** to accept the default `7 days` scratch org duration.
- Be patient, creating a scratch org can take a minute. The success message should look like this in the output panel of VS Code:

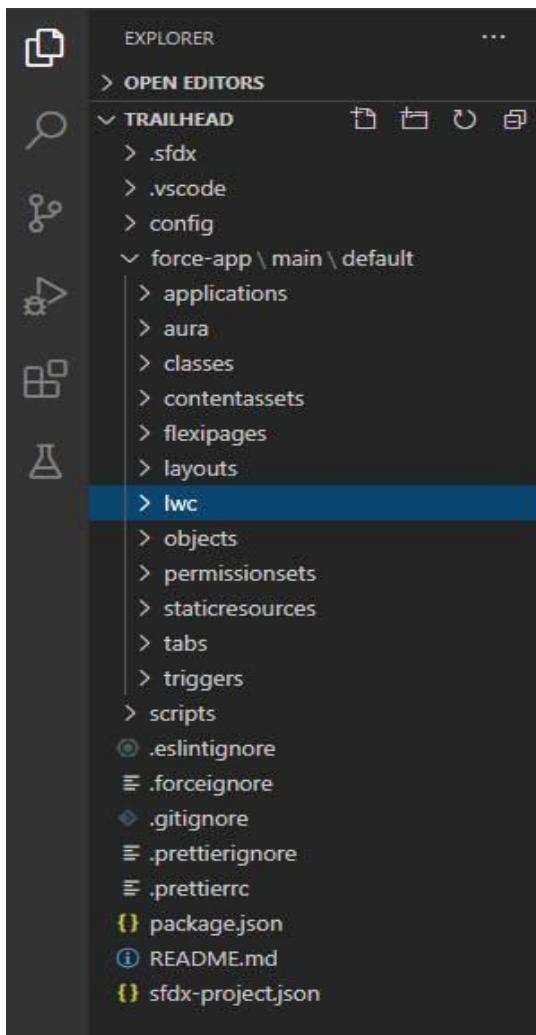
```
17:18:11.779 sfdx force:org:create -f config\project-scratch-def.json --setalias trailhead --durationdays 7 --setDefaultusername --json --loglevel fatal ended with exit code 0
```

Now you're all set to develop your first Lightning web component

Create a Lightning Web Component

Creating a Lightning web component is a straightforward process. And Salesforce CLI already created a project structure that helps make getting started even easier.

The folder structure looks like this:



The project we created has a special folder, **force-app/main/default**. This folder, called a package directory, contains all the metadata of your current Salesforce DX project. Because Lightning web components are also metadata, they are stored in a subfolder named **lwc**. In the next step, we add a Lightning web component to this folder.

We can use Visual Studio Code for creating a Lightning web component, just as we did to create the Salesforce DX project. Or we can use Salesforce CLI directly.

1. Open Visual Studio Code.

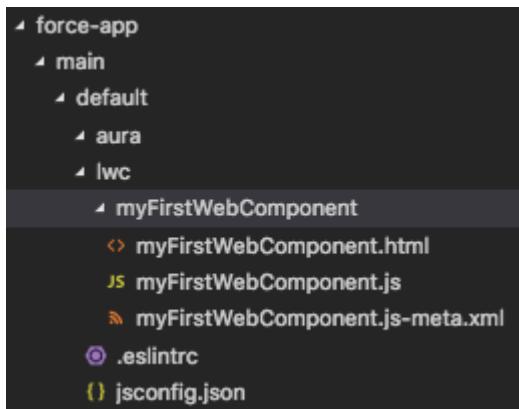
2. Press Command + Shift + P on macOS or Ctrl + Shift + P on Windows or Linux, then type focus terminal. Press Enter.

3. Enter

```
sfdx force:lightning:component:create -n myFirstWebComponent -d force-app/main/default/lwc --type lwc
```

, and confirm with Enter.

This creates the needed files for your first Lightning web component.



Adding Code and Metadata to Your First Lightning Web Component

1. Open the new subfolder for **myFirstWebComponent**, which we just created in the lwc subfolder
2. Click **myFirstWebComponent.js-meta.xml**.
3. Replace the contents of the XML file with this XML markup

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>51.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>lightning__AppPage</target>
    <target>lightning__RecordPage</target>
    <target>lightning__HomePage</target>
  </targets>
</LightningComponentBundle>
```

4. Press **CMD + S** on macOS, or **CTRL + S** on Windows or Linux, to save the file.

Next we're updating the JavaScript file of your Lightning web component.

1. In Visual Studio Code click **myFirstWebComponent.js**.
2. Paste this

```
import { LightningElement } from 'lwc';
export default class MyFirstWebComponent extends LightningElement {
    @track
    contacts = [
        {
            Id: 1,
            Name: 'Amy Taylor',
            Title: 'VP of Engineering',
        },
        {
            Id: 2,
            Name: 'Michael Jones',
            Title: 'VP of Sales',
        },
        {
            Id: 3,
            Name: 'Jennifer Wu',
            Title: 'CEO',
        },
    ];
}
```

3. Press **CMD + S** on macOS, or **CTRL + S** on Windows or Linux, to save the file.

After you save the file you'll immediately notice a few things.

- The annotated word `@track` is underlined with a red squiggly line.
- The JavaScript file color in the explorer changed to red, and has a 2 next to it.

HTML markup

1. In Visual Studio Code click **myFirstWebComponent.html**.
2. Insert this markup within the existing `<template></template>` tags

```
<lightning-card title="ContactInformation" icon-name="custom:custom14">
    <div class="slds-m-around_medium">
        <template for:each={ } for:item="contact">
            <div>
                {contact.Name}, {contact.Title}
            </div>
        </template>
    </div>
</lightning-card>
```

3. Press **CMD + S** on macOS, or **CTRL + S** on Windows or Linux, to save the file.

Deploy and Configure Your New Lightning Web Component

1. Open **Visual Studio Code**.
2. Press **Command + Shift + P** on macOS or **Ctrl + Shift + P** on Windows or Linux, then type **focus terminal**. Press **Enter**.
3. Enter this command to deploy the metadata to your org:
sfdx force:source:push
4. Press **Enter**

Day 11 :

Topic: Lightning Web Components

Why Lightning Web Components(LWC)?

AURA framework which was used for current Lightning Components was based on standards of 2014 but are outdated now and it was time for change because for the following reasons:

- Rendering could be optimized.
- Standard UI elements were scarce.
- Lacked modern constructs.

- Was not fit for modular apps.
- Web standards were updated.
- AURA Framework became skill and had its own learning curve.

Additionally, Lightning Web Components(LWC) can coexist and interoperate with Aura components.

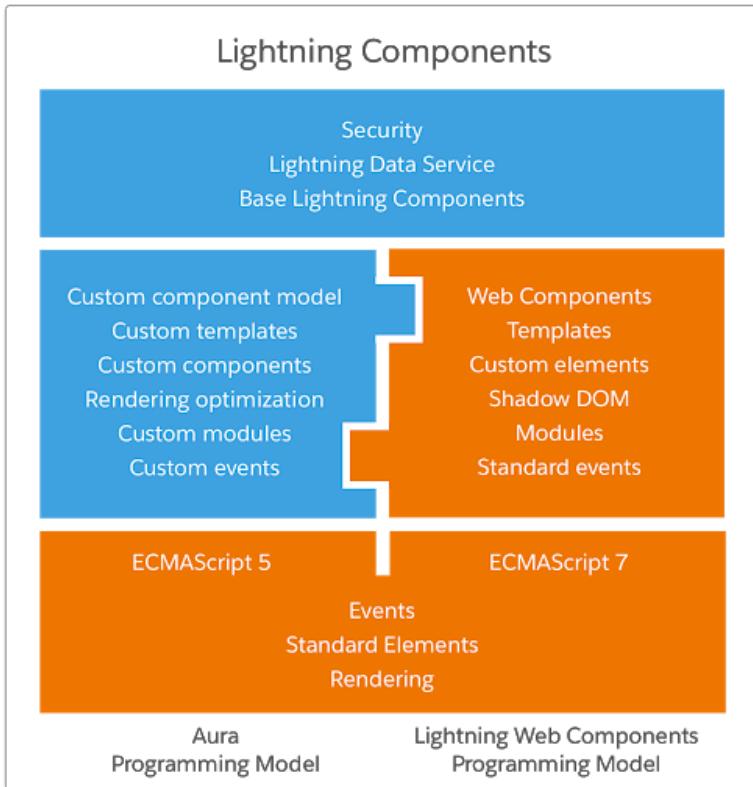
What is Lightning Web Components(LWC)?

LWC is a new programming model levering the recent web standards. Rather than being a totally custom and development wise rigid framework, It's quite flexible. It's mostly the common Web Standards and a Thin Layer of Specialized services to make it a perfect fit for Modern Rich UI Implementations in Salesforce. This thin layer of specialized services contain Base Lightning Components, Lightning Data Service and User Interface API which work behind the curtain for LWC.

A thin layer of specialized services on top of a standard web stack results in:

- Ease of development for large scale modular apps.
- Ease of Leveraging the latest web functionalities and constructs.
- A common model and transferable skills.
(Any web developer working on modern JS frameworks could easily ramp-up LWC).
- Interoperable components.
- Better performance.

So, the new development stack looks like:



How is a Lightning Web Component(LWC) formed?

Similar to an AURA component, the main contents of a LWC are also html, javascript. There are optional content like css. But then in addition to these for LWC, an xml configuration file is also included which defines the metadata values for the component.

```

  ▲ src
    ▲ lwc
      ▲ lightningWebComponent
        # lightningWebComponent.css
        <> lightningWebComponent.html
        JS lightningWebComponent.js
        ⚡ lightningWebComponent.js-meta.xml

```

HTML

- Has a root tag <template> which contains your component's HTML.
- When renders, the <template> tag is replaced with <namespace-component-name>.

```
↳ lightningWebComponent.html •
1  <template>
2    <lightning-card title="Hello Component" icon-name="standard:lightning_component">
3      <div class="slds-m-around_medium">
4        <p>Hello!</p>
5        <lightning-input label="Draft message:" value={message} onchange={handleChange} placeholder="Type
6          your message here..."></lightning-input>
7        <hr>
8        <div>
9          <p class="background-grey">Preview:</p>
10         <p>{message}</p>
11       </div>
12     </div>
13   </lightning-card>
14 </template>
```

Javascript

- Import functionality declared in a module eg-lwc(the core module), use the import statement.
- To allow other code to use functionality in a module, use the export statement.
- LightningElement is custom wrapper of the standard HTML element and we extend it in the component and export.

```
JS lightningWebComponent.js ✘
1  import { LightningElement, track } from 'lwc';
2
3  export default class lightningWebComponent extends LightningElement {
4    @track message;
5
6    handleChange(event) {
7      this.message = event.target.value;
8    }
9  }
```

Configuration

XML file that defines the metadata configuration values for the component eg-

- Components Label
- Availability
- Configuration Attributes
- Builder Attributes

```
# lightningWebComponent.js-meta.xml ✘
1  <?xml version="1.0" encoding="UTF-8"?>
2  <LightningComponentBundle xmlns="urn:metadata.tooling.soap.sforce.com" fqn="LightningWebComponent">
3      <apiVersion>45.0</apiVersion>
4      <isExposed>true</isExposed>
5      <targets>
6          <target>lightning__AppPage</target>
7          <target>lightning__RecordPage</target>
8          <target>lightning__HomePage</target>
9      </targets>
10 
```

CSS

- To style a component.
- The style sheet is applied automatically

```
# lightningWebComponent.css ✘
1  .background-grey {
2      background-color: lightgrey;
3 }
```

Component UI

The screenshot shows a component editor interface. At the top left is a lightning bolt icon followed by the text "Hello Component". Below this, the word "Hello!" is displayed. A section labeled "Draft message:" contains the placeholder text "Type here and as you type you can see the preview below." Below this is a horizontal line. A section labeled "Preview:" contains the same placeholder text "Type here and as you type you can see the preview below.".

Day 12 :

Topic: Component Creation For Application

Milestones: Created Components For Application

1.Create College DataTable Component(APEX CLASS)

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 .

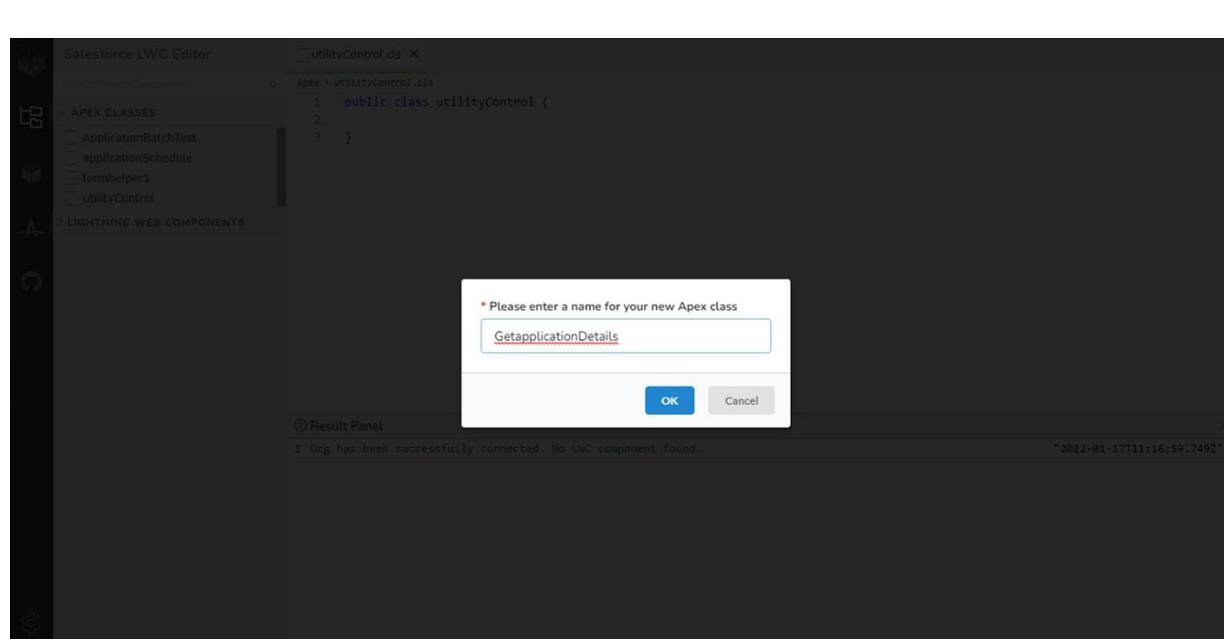
```
public class GetapplicationDetails
{
    @AuraEnabled(cacheable=true)
```

```
public static List<Application_Form__c> getapplicationvalues(id CollegelId)
{
    List<Application_Form__c> formlist = [SELECT ID, College_Fee__c,Name,
Date_Of_Birth__c,Email__c,Hostel_Fee__c, Student_Name__c FROM
Application_Form__c WHERE College__c =:CollegelId];

    return formList;
}

}
```

}



```
□ GetapplicationDetails.cls ×
Apex > GetapplicationDetails
1  public class GetapplicationDetails
2  {
3      @AuraEnabled(cacheable=true)
4  public static List<Application_Form__c> getapplicationvalues(id CollegeId)
5  {
6
7  List<Application_Form__c> formlist = [SELECT ID, College_Fee__c,Name,
8 Date_of_Birth__c,Email__c,Hostel_Fee__c, Student_Name__c FROM
9 Application_Form__c WHERE College__c =:CollegeId];
10
11 return formList;
12 }
13 }
14 }
```

Create College DataTable Component (HTML FILE)

```
<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>
```

```

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

```

Create College DataTable Component(JAVA SCRIPT 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 Fee' , fieldName : 'College_Fee__c', type:'currency' },
    {label : 'Date Of Birth' , fieldName : 'Date_Of_Birth__c', type:'date' },
    {label : 'Email' , fieldName : 'Email__c', type:'email' },
    {label : 'Hostel Fee' , fieldName : 'Hostel_Fee__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;
}
}
}

```

The screenshot shows the Salesforce Dev Console interface. At the top, there are tabs for GetapplicationDetails.cls, lwcApex.html, and lwcApex.js. The lwcApex.js tab is active, displaying the following code:

```

1 import { LightningElement, api, wire } from 'lwc';
2 import getapplicationvalues
3 from '@salesforce/apex/GetapplicationDetails.getapplicationvalues';
4 export default class CollegeDataTable extends LightningElement {
5     columnslist = [
6         {label : 'Application Form' , fieldName : 'Name', type:'text' },
7         {label : 'College Fee' , fieldName : 'College_Fee__c', type:'currency' },
8         {label : 'Date Of Birth' , fieldName : 'Date_Of_Birth__c', type:'date' },
9         {label : 'Email' , fieldName : 'Email__c', type:'email' },
10        {label : 'Hostel Fee' , fieldName : 'Hostel_Fee__c', type:'Currency' },
11        {label : 'StudentName' , fieldName : 'Student_Name__c', type:'text' }
12    ];
13
14    @api recordId;
15    recordList;
16    error;
17
18    @wire(getapplicationvalues, {CollegeId : '$recordId'})

```

Below the code editor is a "Result Panel" showing deployment logs:

Log	Timestamp
lwcApex.html successfully deployed source	1/17/2022 5:13:20 PM
lwcApex.js successfully deployed source	1/17/2022 5:13:20 PM

Create College DataTable Component(META FILE)

```

<?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 screenshot shows the Salesforce LWC Editor interface. On the left is a sidebar with icons for Apex Classes, Lightning Web Components, and a Recent Items section. The main area has tabs for GetapplicationDetails.cls, lwcApex.html, lwcApex.js, and lwcApex.js-meta.xml. The lwcApex.js-meta.xml tab is active, displaying the XML code provided above. Below the code editor is a Result Panel showing deployment logs:

Log	Date
1 lwcApex.html successfully deployed source	1/17/2022 5:18:33 PM
2 lwcApex.js successfully deployed source	1/17/2022 5:18:33 PM
3 lwcApex.js-meta.xml successfully deployed source	1/17/2022 5:18:33 PM