

Day 1: 03/01/2022, monday

Topic : Create your salesforce developer org to get started

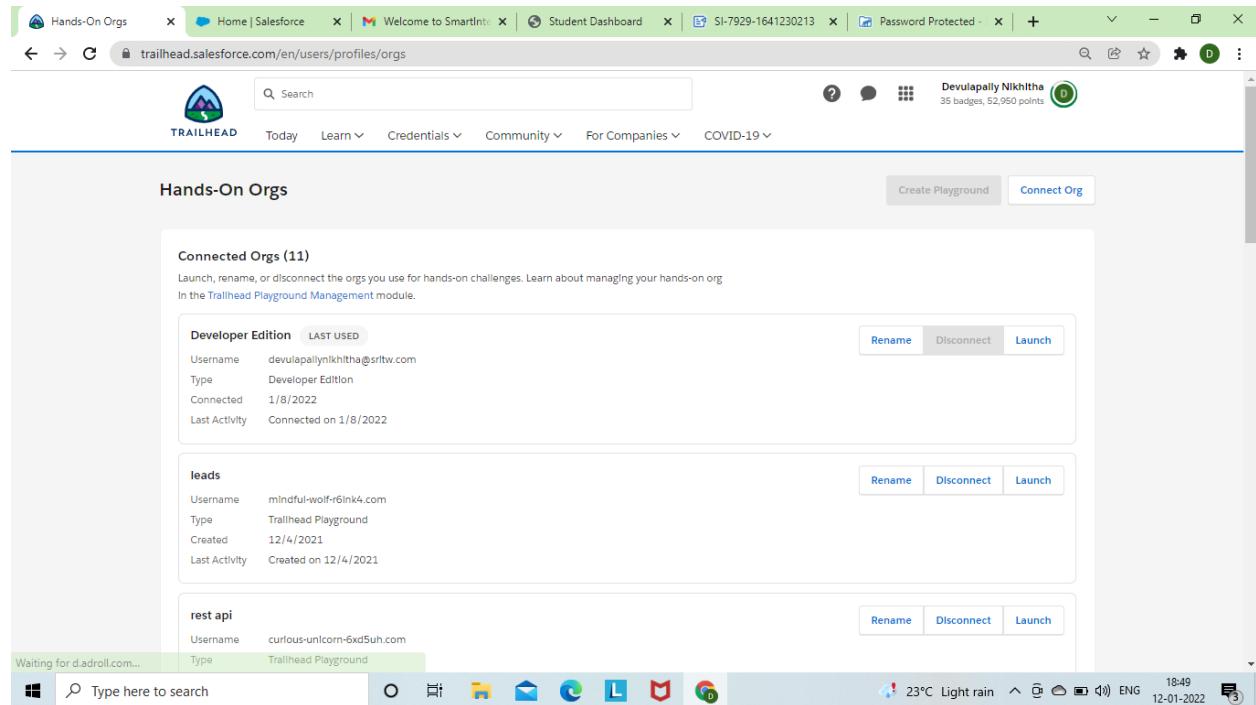
Milestone/Activities : Creating developer account

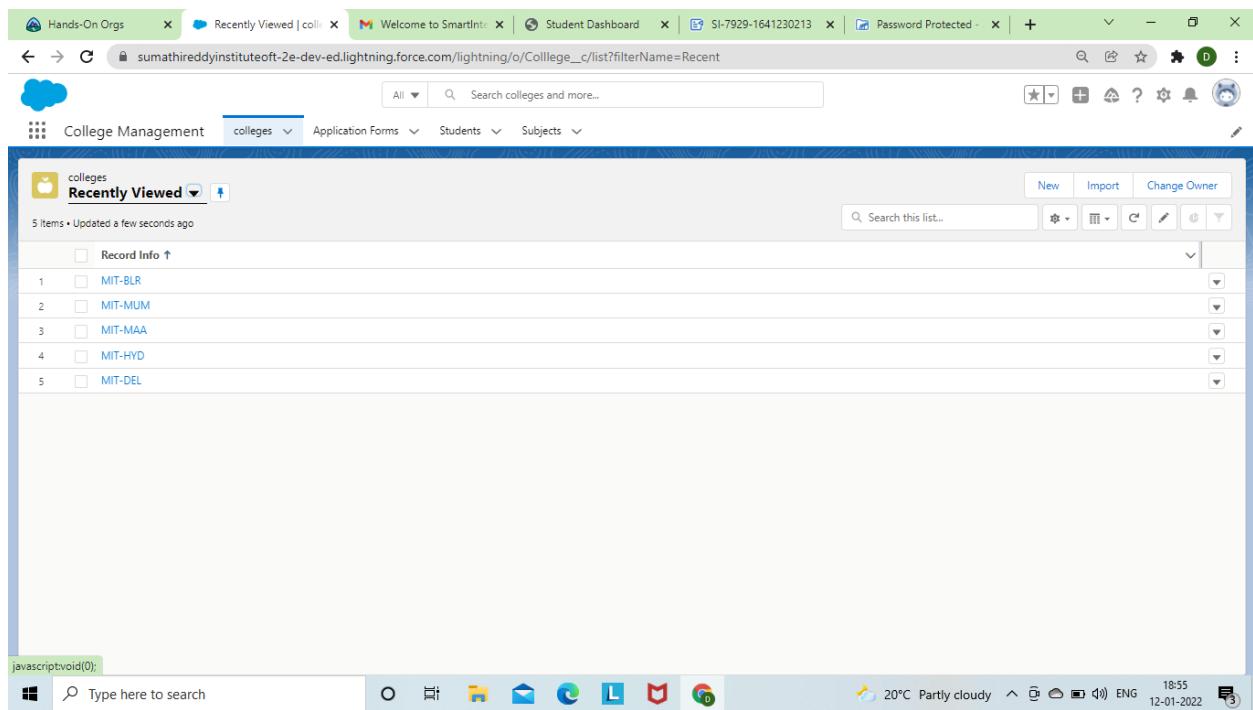
Account activation

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 :





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

The screenshot shows the Salesforce Object Manager page. At the top, there are tabs for Setup, Home, and Object Manager. A search bar at the top right contains the text "College". The main area displays a table with the following data:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
College	College__c	Custom Object		1/10/2022	✓
college	college__c	Custom Object		1/8/2022	✓

The screenshot shows the details page for the "College" object in the Object Manager. The left sidebar lists various configuration options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions (which is selected), Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, Search Layouts for Salesforce Classic, and Restriction Rules. The main panel displays the "Details" section for the "College" object. The "Buttons, Links, and Actions" tab is active. The "Description" field is empty. The "API Name" field is set to "College__c". The "Custom" checkbox is checked. The "Singular Label" field is set to "College". The "Plural Label" field is set to "colleges". On the right side, there are buttons for "Edit" and "Delete". The status bar at the bottom shows the URL: <https://sumathireddyinstituteoft-2e-dev-ed.lightning.force.com/one/one.app#/setup/ObjectManager/011j000001PjV8/ButtonsLinksActions/view>.

The screenshot shows a web browser window with multiple tabs open, including 'Hands-On Orgs', 'Object Manager', 'Welcome to Sm', 'Student Dashboard', 'SI-7929-1641230', 'Password Protect', and the current page, 'sumathireddyinstituteoft-2e-dev-ed.lightning.force.com/lightning/o/College__c/list?filterName=00B5j0000Aax1QEAR'. The main content area displays a 'College Management' interface with a 'colleges' tab selected. A search bar at the top right contains the placeholder 'Search colleges and more...'. Below the search bar is a navigation bar with links for 'colleges', 'Application Forms', 'Students', and 'Subjects'. The main area shows a table with 5 items, sorted by Record Info. The columns are 'Record Info', 'College Fee', 'College Name', and 'Hostel Fee'. The data is as follows:

	Record Info	College Fee	College Name	Hostel Fee
1	MIT-BLR	\$140,000.00	MIT-BLR	\$189,000.00
2	MIT-DEL	\$1,100,000.00	MIT-DEL	\$250,000.00
3	MIT-HYD	\$1,000,000.00	MIT-HYD	\$150,000.00
4	MIT-MAA	\$1,000,000.00	MIT-MAA	\$200,000.00
5	MIT-MUM	\$1,500,000.00	MIT-MUM	\$400,000.00

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

APPLICATION FORM OBJECT:

The screenshot shows the Salesforce Object Manager home page. At the top, there are several tabs: Hands-On Orgs, Recently Viewed, Object Manager (which is active), Welcome to SmartInternz!, Student Dashboard, SI-7929-1641230213, Password Protection, and others. Below the tabs is a search bar labeled "Search Setup". The main content area is titled "Object Manager" and shows a table with one item:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Application Form	application_form_c	Custom Object		1/12/2022	✓

The screenshot shows the details page for the "ApplicationForm" object. The URL in the address bar is <https://sumathireddyinstituteoft-2e-dev-ed.lightning.force.com/lightning/setup/ObjectManager/0115j000001PVdm/Details/view>. The page title is "ApplicationForm". On the left, there is a sidebar with navigation links: 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. The main content area is titled "Details" and contains the following information:

Description	<input type="text"/>
API Name	applicationform_c
Custom	✓
Singular Label	ApplicationForm
Plural Label	ApplicationForms
Enable Reports	<input type="checkbox"/>
Track Activities	<input type="checkbox"/>
Track Field History	<input type="checkbox"/>
Deployment Status	<input type="checkbox"/>
Deployed	<input checked="" type="checkbox"/>
Help Settings	<input type="checkbox"/>
Standard salesforce.com Help Window	<input type="checkbox"/>

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes tabs for Hands-On Orgs, Recently Viewed, Object Manager, Welcome to Sma..., Student Dashboard, SI-7929-1641230, Password Protect..., and several others. The main content area displays a list titled "Recently Viewed" under "Application Forms". The list contains one item: "Application Form ID" with value "F-000001". The interface includes standard Salesforce navigation and search tools.

STUDENT OBJECT:

The screenshot shows the Salesforce Object Manager home page. The top navigation bar includes tabs for Hands-On Orgs, Recently Viewed, Object Manager, Welcome to Sma..., Student Dashboard, SI-7929-1641230, Password Protect..., and several others. The main content area displays a table titled "Object Manager" with one item: "Students". The table columns are labeled: LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. The "Students" entry has an API name of "Students_c", is a "Custom Object", was last modified on 1/8/2022, and is deployed. A search bar at the top right contains the text "STUD". The interface includes standard Salesforce navigation and search tools.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes tabs for 'Hands-On Orgs', 'Students | Salesforce', 'Welcome to SmartInternz!', 'Student Dashboard', and 'SI-7929-1641230213'. The main content area is titled 'SETUP > OBJECT MANAGER' and 'Students'. On the left, a sidebar lists various configuration options: 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. The right panel displays the 'Details' section for the 'Students' object, which includes fields for Description, API Name (Students__c), Singular Label (Students), Plural Label (Students), Enable Reports, Track Activities, Track Field History, Deployment Status (Deployed), Help Settings, and a link to the Standard salesforce.com Help Window. The URL in the browser address bar is <https://sumathireddyinstituteoft-2e-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01i5j000001PVdr/Details/view>.

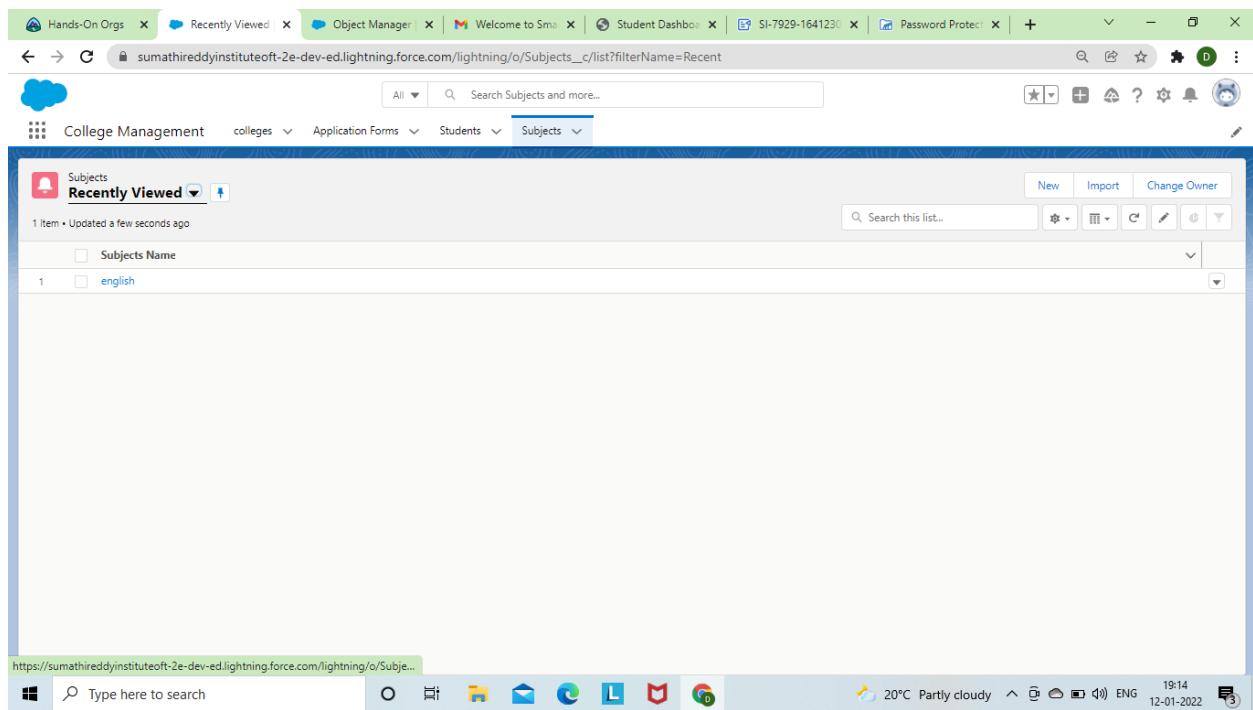
The screenshot shows the 'College Management' application interface. The top navigation bar includes tabs for 'Hands-On Orgs', 'Recently Viewed', 'Object Manager', 'Welcome to SmartInternz!', 'Student Dashboard', and 'SI-7929-1641230'. The main content area is titled 'College Management' and shows a 'Recently Viewed' list under the 'Students' tab. The list contains one item: 'nikhitha'. The URL in the browser address bar is https://sumathireddyinstituteoft-2e-dev-ed.lightning.force.com/lightning/o/Students_c/list?filterName=Recent.

-==
SUBJECT OBJECT:

The screenshot shows the Salesforce Object Manager interface. At the top, there are several tabs: Hands-On Orgs, Recently Viewed, Object Manager (which is active), Welcome to SmartInternz!, Student Dashboard, SI-7929-1641230, and Password Protection. Below the tabs, the URL is sumathireddyinstituteoft-2e-dev-ed.lightning.force.com/lightning/setup/ObjectManager/home. The main area is titled "Object Manager" with a subtitle "1 Items, Sorted by Label". It displays a table with columns: LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. There is one item listed: "Subjects" (API Name: Subjects__c, Type: Custom Object, Last Modified: 1/12/2022, Deployed: ✓). A search bar at the top right contains "SUBJ".

This screenshot shows the details page for the "Subjects" custom object. The URL is sumathireddyinstituteoft-2e-dev-ed.lightning.force.com/lightning/setup/ObjectManager/0115j000001PVe1/Details/view. The page title is "Setup > OBJECT MANAGER Subjects". On the left, a sidebar lists various configuration options: 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 main content area is titled "Details" and shows the following fields:

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



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

Screenshot of the Salesforce Setup interface showing the Picklist Value Sets page.

The left sidebar shows:

- PICKLI
- Data
 - Picklist Settings
 - State and Country/Territory
 - Picklists**
- Objects and Fields
 - Picklist Value Sets**

The main content area displays the "Picklist Value Sets" page with the following details:

Picklist Value Sets

Global picklist value sets let you share the values across objects. Base custom picklist fields on a global value set to inherit its values. The value set is restricted so users can't add unapproved values through the API.

View: All | Create New View

Global Value Sets

Action	Label	Description
Edit Del	College	
Edit Del	Page_1	
Edit Del	Page_2	
Deleted Global Value Sets (0)		

Screenshot of the Salesforce Setup interface showing the Global Value Set detail page for "COLLEGE".

The left sidebar shows:

- PICKLIS
- Data
 - Picklist Settings
 - State and Country/Territory
 - Picklists**
- Objects and Fields
 - Picklist Value Sets**

The main content area displays the "Global Value Set" detail page for "College".

Global Value Set

Information

Label	College
Name	College
Description	

Picklist Values Used

Active and inactive picklist values 6 (1,000 max)

Values

Action	Values	API Name	Default	Chart Colors	Modified By
New					
Reorder					
Replace					
Printable View					
Chart Colors					

Hands-On Orgs | Picklist Value Sets | Salesforce | Welcome to SmartInternz! | Student Dashboard | SI-7929-1641230213

Picklist Value Sets

Picklist Values Used

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Deactivate	MIT-HYD	MIT-HYD	<input type="checkbox"/>	Assigned dynamically	nikitha.devalagally, 19/01/2022, 6:55 PM
Edit Deactivate	MIT-BLR	MIT-BLR	<input type="checkbox"/>	Assigned dynamically	nikitha.devalagally, 19/01/2022, 6:55 PM
Edit Deactivate	MIT-MUM	MIT-MUM	<input type="checkbox"/>	Assigned dynamically	nikitha.devalagally, 19/01/2022, 6:55 PM
Edit Deactivate	MIT-MAA	MIT-MAA	<input type="checkbox"/>	Assigned dynamically	nikitha.devalagally, 19/01/2022, 6:55 PM
Edit Deactivate	MIT-DEL	MIT-DEL	<input type="checkbox"/>	Assigned dynamically	nikitha.devalagally, 19/01/2022, 6:55 PM
Edit Deactivate	MIT-CCU	MIT-CCU	<input type="checkbox"/>	Assigned dynamically	nikitha.devalagally, 19/01/2022, 6:55 PM

Values

Inactive Values

No Inactive Values values defined.

Fields Where Used

Field Label	Object	Data Type	Controlling Field
College Name	College	Picklist	

Always show me more records per related list

PAPER 1:

Hands-On Orgs | Picklist Value Sets | Salesforce | Welcome to SmartInternz! | Student Dashboard | SI-7929-1641230213

Picklist Value Sets

Global Value Set

Global Value Set Detail

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Delete	Paper_1	PAPER_1	<input type="checkbox"/>	Assigned dynamically	nikitha.devalagally, 19/01/2022, 6:55 PM
Edit Delete	Paper_1	PAPER_1	<input type="checkbox"/>	Assigned dynamically	nikitha.devalagally, 19/01/2022, 6:55 PM
Edit Delete	Description	DESCRIPTION	<input type="checkbox"/>	Assigned dynamically	nikitha.devalagally, 19/01/2022, 6:55 PM

Values

Values [4] | Inactive Values [0] | Fields Where Used [1]

Help for this Page

Information

Picklist Values Used

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

Values

Always show me more records per related list

Hands-On Orgs | Picklist Value Sets | Salesforce | Welcome to SmartInternz! | Student Dashboard | SI-7929-1641230213

sumathireddyinstituteoft-2e-dev-ed.lightning.force.com/lightning/setup/Picklists/page?address=%2F0Nt5j000000v5js

Action	Values	API Name	Default	Chart Colors	Modified By	Modified On
Edit Del Deactivate	APEX	APEX	<input type="checkbox"/>	Assigned dynamically	nikhitha.devulapally	1/12/2022, 1:25 AM
Edit Del Deactivate	JAVA	JAVA	<input type="checkbox"/>	Assigned dynamically	nikhitha.devulapally	1/12/2022, 1:25 AM
Edit Del Deactivate	C	C	<input type="checkbox"/>	Assigned dynamically	nikhitha.devulapally	1/12/2022, 1:25 AM
Edit Del Deactivate	C++	C++	<input type="checkbox"/>	Assigned dynamically	nikhitha.devulapally	1/12/2022, 1:25 AM

Inactive Values

No Inactive Values values defined.

Fields Where Used

Field Label	Object	Data Type	Controlling Field
Paper_1	Subjects	Picklist	

Back To Top Always show me more records per related list

PAPER 2:

Hands-On Orgs | Picklist Value Sets | Salesforce | Welcome to SmartInternz! | Student Dashboard | SI-7929-1641230213

sumathireddyinstituteoft-2e-dev-ed.lightning.force.com/lightning/setup/Picklists/page?address=%2F0Nt5j000000v5jx

Global Value Set

Help for this Page

Global Value Set Detail

Information

Label	Paper 2
Name	Paper_2
Description	

Picklist Values Used

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

Values

Action	Values	API Name	Default	Chart Colors	Modified By	Modified On
Edit Del Deactivate	Paper 2	PAPER_2	<input type="checkbox"/>	Assigned dynamically	nikhitha.devulapally	1/12/2022, 1:25 AM

https://sumathireddyinstituteoft-2e-dev-ed.my.salesforce.com/one/one.app#sln...

Picklist Value Sets

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	MATHEMATICS	MATHEMATICS	<input type="checkbox"/>	Assigned dynamically	nikhitha devulapally, 1/12/2022, 1:25 AM
Edit Del Deactivate	ENGLISH	ENGLISH	<input type="checkbox"/>	Assigned dynamically	nikhitha devulapally, 1/12/2022, 1:25 AM
Edit Del Deactivate	STATISTICS	STATISTICS	<input type="checkbox"/>	Assigned dynamically	nikhitha devulapally, 1/12/2022, 1:25 AM

Inactive Values
No Inactive Values values defined.

Fields Where Used

Field Label	Object	Data Type	Controlling Field
Paper_2	Subjects	Picklist	

CREATING FIELD DEPENDENCIES

College Field Dependencies

Action	Controlling Field	Dependent Field	Modified By
Edit Del	College Name	College Email	nikhitha devulapally, 1/9/2022, 7:22 PM
Edit Del	College Name	No of seats	nikhitha devulapally, 1/9/2022, 7:29 PM

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. On the left, a sidebar lists various setup categories like Page Layouts, Lightning Record Pages, and Validation Rules. The Validation Rules section is currently selected. The main content area displays a table titled "Validation Rules" with one item:

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
College_Name	Top of Page	Please use the college name in record info	✓	nikhitha devulapally, 1/11/2022, 12:36 AM

Below this, another screenshot shows the details of the validation rule. The "Validation Rule Detail" section includes fields for Rule Name (College_Name), Error Condition Formula (TEXT(College_Name_c) <> Name), Error Message (Please use the college name in record info), Description, Created By (nikhitha devulapally, 1/11/2022, 12:36 AM), and Modified By (nikhitha devulapally, 1/11/2022, 12:36 AM). The rule is marked as Active.

Validation rule on application form object:

Validation Rules

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Student_Record	Top of Page	please use the college name in record name	✓	nikhitha devulapally, 1/14/2022, 5:14 AM

Validation Rule Detail

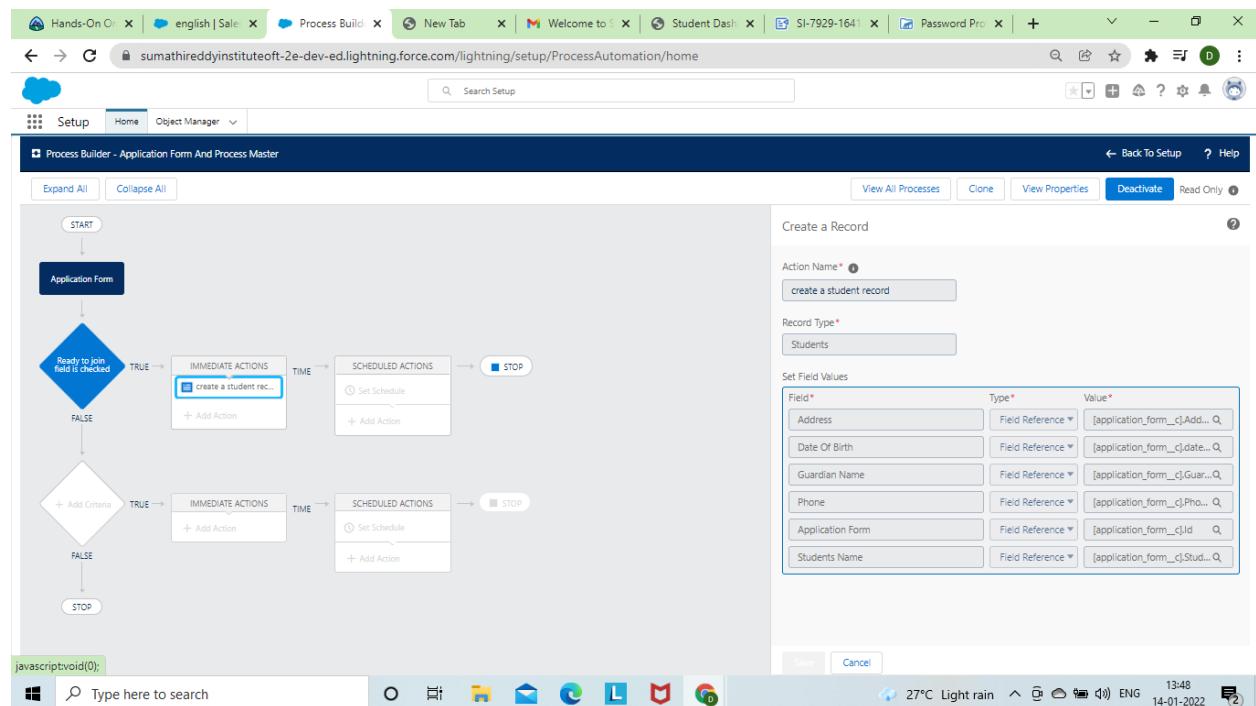
Rule Name	Student_Record	Active	
Error Condition Formula	AND(Ready_To_Join__c == true, OR(ISCHANGED(Address__c), ISCHANGED(college__c), ISCHANGED(date_of_Birth__c), ISCHANGED>Email__c), ISCHANGED(Guardian_Name__c), ISCHANGED(Phone__c)))	✓	
Error Message	please use the college name in record name	Error Location	Top of Page
Description		Modified By	nikhitha devulapally, 1/14/2022, 5:14 AM
Created By	nikhitha devulapally, 1/14/2022, 5:14 AM		

Day 6 :10/01/2022

Topic : Adding business logic to application

Milestone/Activities: process automation

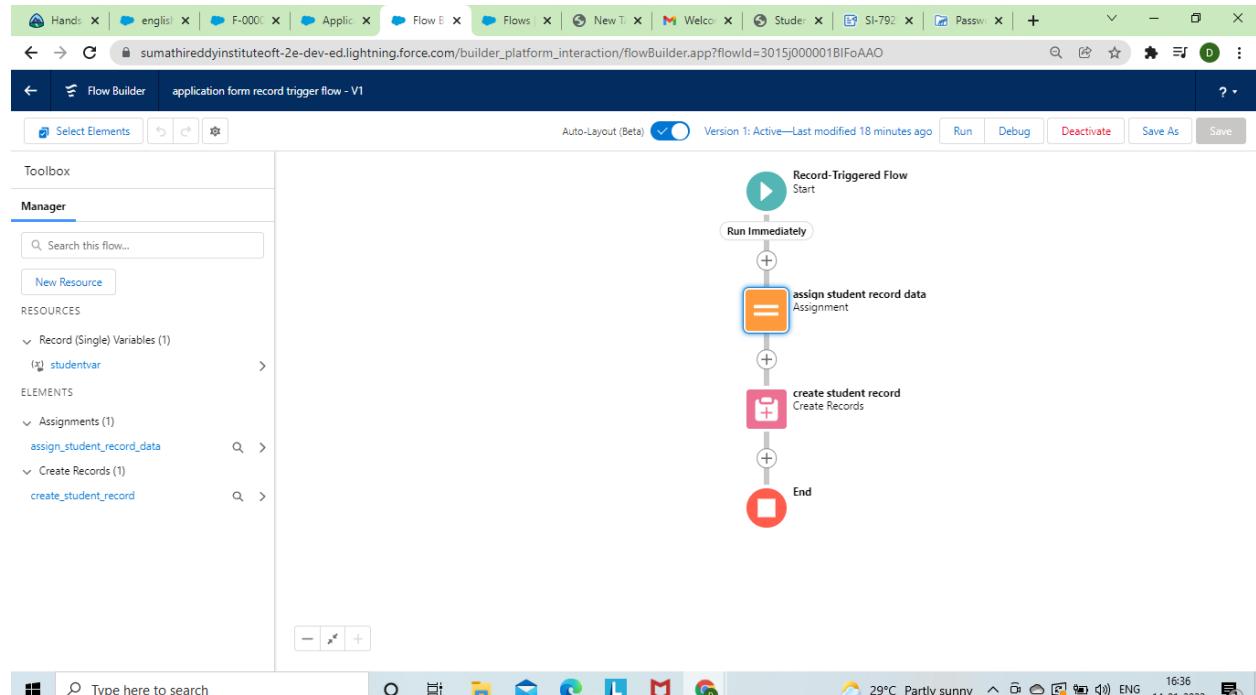
PROCESS AUTOMATION :

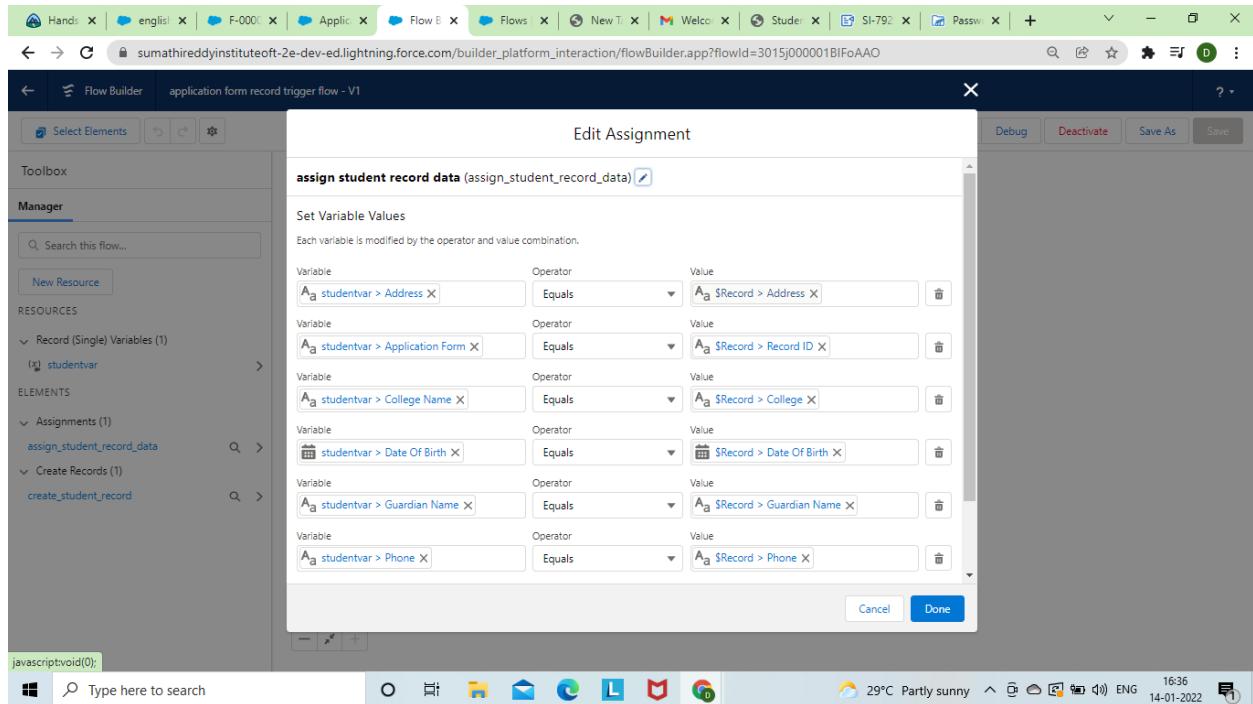


Day 7: 11/01/2022

Topic: Adding Business Logic To Application

Milestone/Activities: CREATE THE STUDENT RECORD USING FLOW





Day 8 : 12/01/2022

Topic : General discussion on student doubts

Day 9: 13/01/2022

Topic: Btch 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

Developer Console - Google Chrome

sumathireddyinstituteoft-2e-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < ApplicationBatchTest.apex

Code Coverage: None API Version: 53

```
1 Public class ApplicationBatchTest implements Database.Batchable<sObject>{
2     //start(), execute(), finish()
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         // gathers the data for you
7         String applicationQuery = 'select id, Name, Ready_To_Join__c from ApplicationForm__c';
8         return Database.getQueryLocator(applicationQuery);
9     }
10    public void execute(Database.BatchableContext bc, List<ApplicationForm__c> formList){
11        // process the data
12        for(ApplicationForm__c af : formList){
13            totalForms++;
14            if(af.Ready_To_Join__c){
15                totalConvertedForms++;
16            }
17        }
18    }
19    public void finish(Database.BatchableContext bc){
20        // emails
21        Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
22        // address, subject, content data to sent to admins
23        mail.setSubject(' Application form and student record data as of today ');
24        mail.setPlainTextBody(' Total no of application form records are : '+totalForms+ ' out of which no of students as per today : '+totalConvertedForms+ );
25        String[] emailAddress = new String[]{your_email_address};
26        mail.setToAddresses(emailAddress);
27    }
28}
```

Logs Tests Checkpoints Query Editor View State Progress Problems

User Application Operation Time Status Read Size

Filter Click here to filter the log list

Type here to search

28°C Mostly cloudy 17:18 14-01-2022

Create A Schedular class

Developer Console - Google Chrome

sumathireddyinstituteoft-2e-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < applicationschedule.apex

Code Coverage: None API Version: 53

```
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
```

Logs Tests Checkpoints Query Editor View State Progress Problems

User Application Operation Time Status Read Size

Filter Click here to filter the log list

Type here to search

28°C Partly sunny 18:28 14-01-2022

The screenshot shows the Salesforce Setup interface with the 'Apex Classes' tab selected. The page title is 'Apex Classes'. A message at the top states 'Percent of Apex Used: 0.02%' and 'You are currently using 1,443 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.' Below this is a button to 'Estimate your organization's code coverage' and another to 'Compile all classes'. A 'View' dropdown is set to 'All'. A table lists two Apex classes:

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del Security	ApplicationBatchTest		53.0	Active	1,225	nikhitha devulapally, 1/14/2022, 3:47 AM	<input type="checkbox"/>
Edit Del Security	applicationschedule		53.0	Active	218	nikhitha devulapally, 1/14/2022, 3:51 AM	<input type="checkbox"/>

The screenshot shows the 'Apex Class Detail' page for the class 'applicationschedule'. The page title is 'Apex Class applicationschedule'. It displays the following details:

Name	applicationschedule	Status	Active
Namespace Prefix		Code Coverage	0% (0/3)
Created By	nikhitha devulapally, 1/14/2022, 3:50 AM	Last Modified By	nikhitha devulapally, 1/14/2022, 3:51 AM

The 'Class Body' tab is selected, showing the following Apex code:

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

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...**

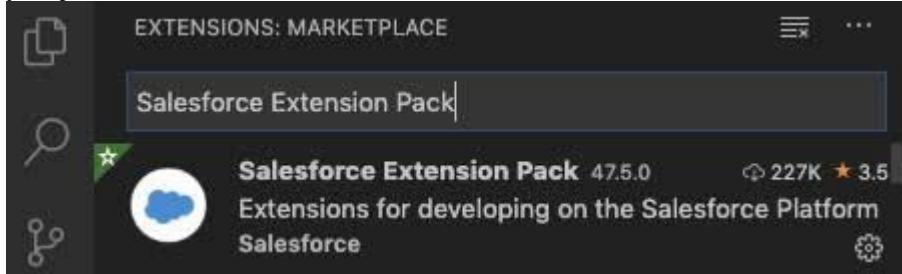
Visual Studio Code

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.

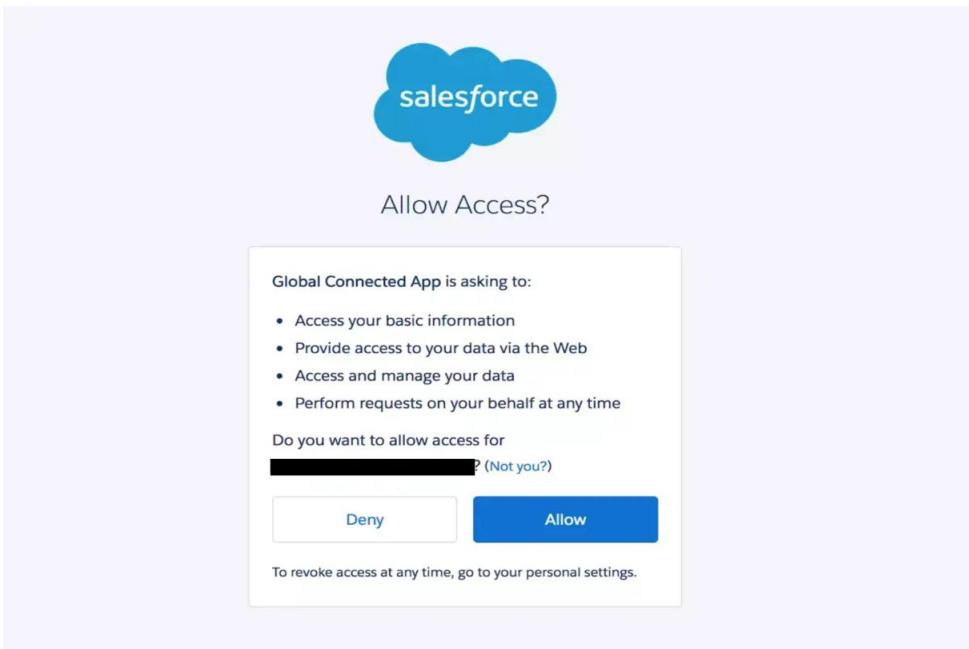
Create a Salesforce DX Project

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:

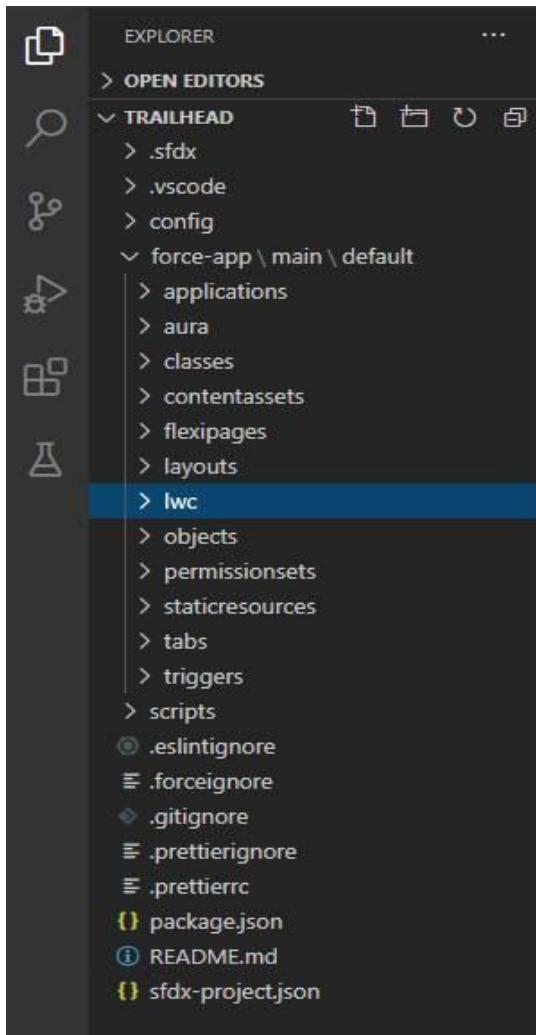
```
17:18:11.779 sfdx force:org:create -f config\project-scratch-def.json --setalias trailhead --durationdays 7 --setDefaultusername --json --loglevel fatal ended with exit code 0
```

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.

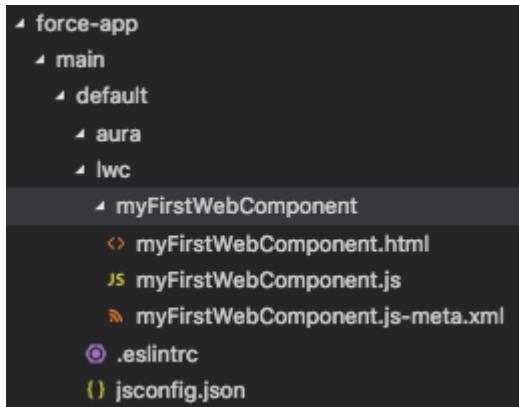
2. Press Command + Shift + P on macOS or Ctrl + Shift + P on Windows or Linux, then type focus terminal. Press Enter.

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.

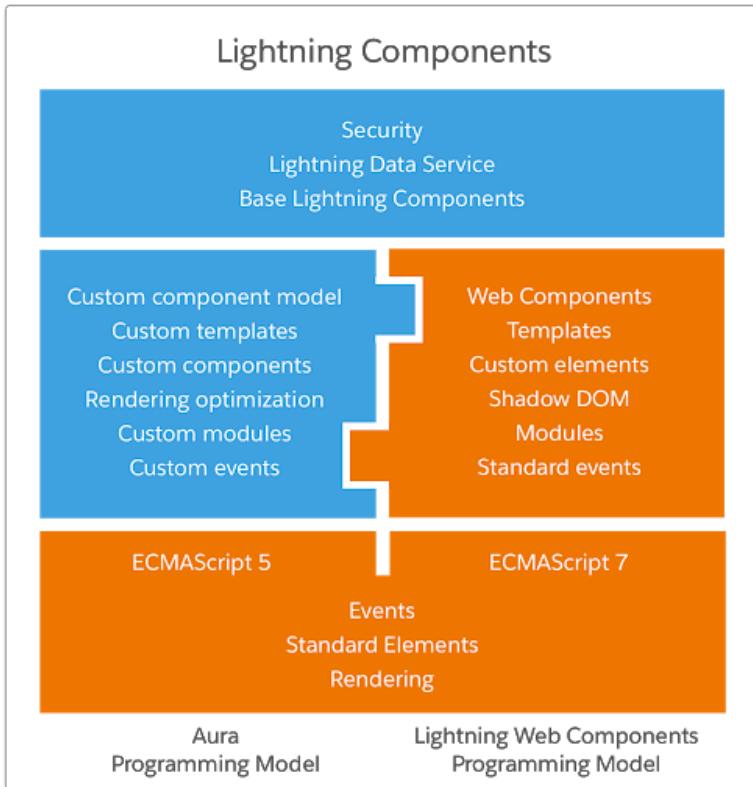
What is Lightning Web Components(LWC)?

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

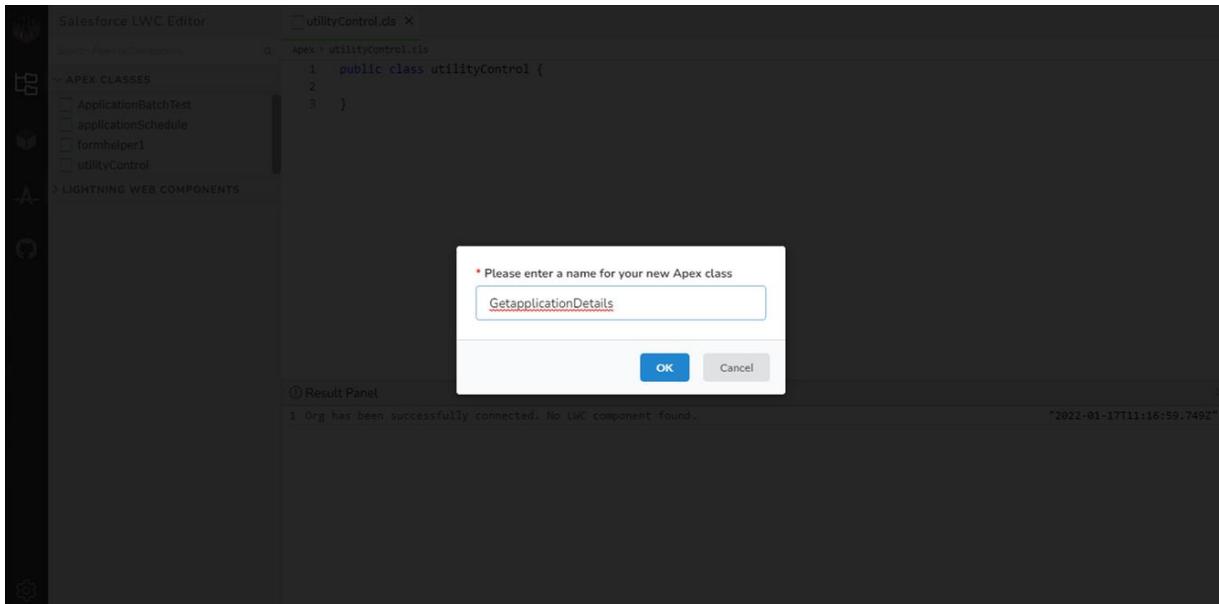
1.Create College DataTable Component(APEX CLASS)

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 Collegeld)
{
    List<Application_Form__c> formlist = [SELECT ID, College_Fee__c, Name,
DateOfBirth__c, Email__c, Hostel_Fee__c, Student_Name__c FROM
Application_Form__c WHERE College__c =:Collegeld];

    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 component's JavaScript code. Below the code editor is a 'Result Panel' showing deployment logs:

```

l 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