Before LWC

The things we need to know before learning LWC.

A Beautiful & Boring story!!

In olden days if we need develop a desktop app, we need to develop 3 apps for 3 platforms means Windows, MAC, Linux.

But, the entry of JavaScript the entire things were changed a lot.

- ✓ We can develop native app in native HTML, CSS & JS and we can use this app in Windows, MAC and Linux.

Similarly, in Android and IOS ecosystem we can develop one app with React native rather than developing 2 apps for each eco system.

😳Hmm 😌, Boring stuff!

Okok... Salesforce wanted to give that power 🔥 to Salesforce Developers.

And Salesforce Come up with a new framework Aura Components! Like write Aura Code and that code compiled to native JS in runtime.

Then Why LWC...?

Simply, the problem with Aura Components programming model supports ES5 and ES6 Syntax but if we want to use later version syntax. Then LWC comes in highlight for modern JS development.

Not only this there is lot of differences, google it to explore more.....

Should I Start with Aura or LWC...?

Always try to use LWC and in case if there is any limitations in framework that stopping you from using LWC then go with Aura Components...

What is this Component based Framework....?

Yes, write your code snippet in one component and use in multiple application.

E.g.: A login form we can use in different application. Like that...

Pre requisite require to learn LWC...

- ✓ Basics of HTML
- ✓ Basics of CSS
- ✓ Basics of Java Script and

What is Lightning application?

Lightning Application Contains a Set of "Lightning Components". Because Lightning Components can't be executable by their own.

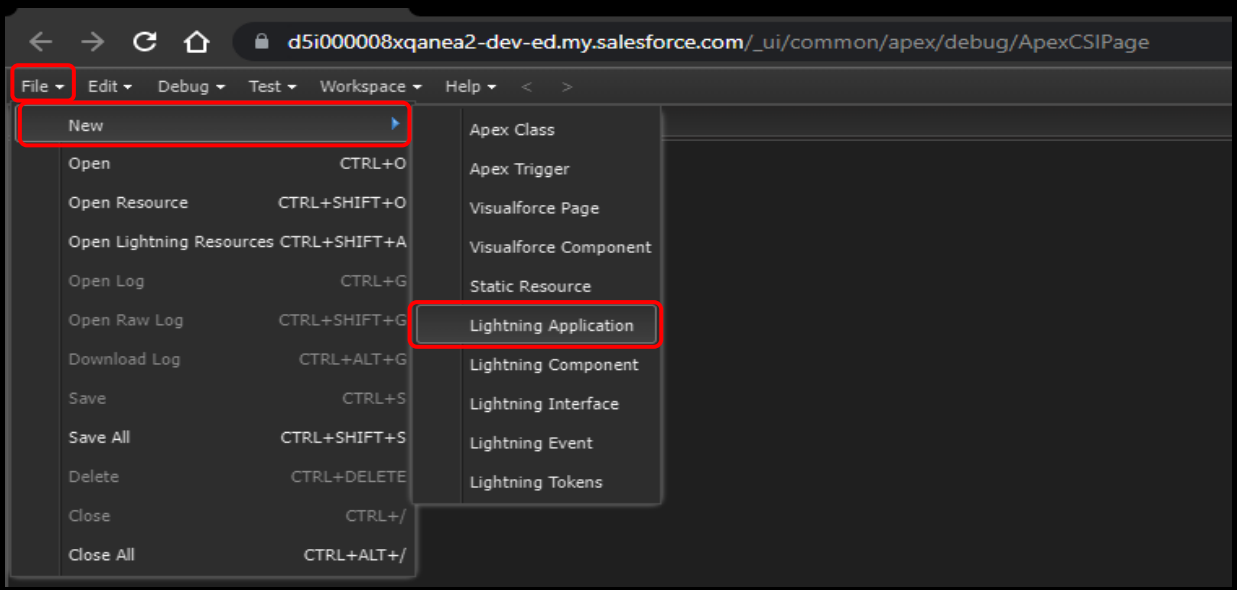
Each Lightning Application should start and ends with "<aura:Applications" tag. Inside a Lightning Application, we can place multiple Lightning Components, which includes AURA Components and LWC Components.

Syntax:

```
<aura:Application>
        <Component 1>
        <Component 2>
        <Component 3>
</aura:Application>
```

How to create a lightning application?

Open Developer Console >>> File menu >>> New >>> Lightning Application



Prerequisite for LWC

JavaScript

Here I'm not discuss more about JS because every programming has similar stuff the only difference is Syntax for cool basic stuff you can refer W3Schools.com

What is JS...?

JavaScript is a Lightweight, Interpreter based programming, which can be used to implement the Client-Side Business Logic, which can be used in the Web Application Development.

JavaScript is a Scripting Language, which supports the Object-Oriented Programming Principles.

What if ...? We need to execute JS Code snippet ...

Hohooo! JS doesn't require any Environmental Setup to write and execute the code and view the result Only Browser is Enough.

Note***

But for LWC we need Environmental Setup, which we can discuss later.

- ✓ Remember this: JavaScript is a Case-Sensitive programming. Hence while writing the code, we have to take care of "Case-Sensitivity"

Some things I'm gone clear,

Variables:

Variables are like containers, which can store values inside it.

In apex we can write and print like this:

```
String myString = 'Hello';
Integer barrelNumbers = 1000;
Boolean shipmentDispatched = true;
```

In Java Script we can write like this

```
var myString = 'Hello'
var isActive = true
var num = 190.56
```

- ✓ If you don't assign a value. Then JS assign value as "Undefined".

While defining variables we must follow this thing!

- ✓ Variable name can start with "either a Character / Underscore / Dollar Sign"
- ✓ But it should not start with a "Digit / Number".
- ✓ Don't use the reserved words for the variable names.
  (Ex: If, switch, for, while, Else,. Etc.)

😏 No, I don't follow this?

Then you'll get this error.

- ❖ Uncaught Syntax Error: Invalid or unexpected token.

Meanwhile you can explore these things in Java Script:

- ✓ How can we write comments in JavaScript
- ✓ Explore All Data Types
  (Num, String, Boolean, BigInt, Undefined, Null)
- ✓ Explore these operators
  1. Arithmetic / Mathematical Operators.
  2. Relational / Comparison Operators,
  3. Logical Operators,
  4. Assignment Operators and
  5. Conditional Operators:
     a) IF Condition
     b) Switch Statement
     c) Ternary Operator.
- ✓ And All Iterative Statements:
  a) While
  b) Do-While.
  c) FOR.
- ✓ Some String Functions & Arrays
- ✓ Objects and Functions

 Ho Huff! Why these many topics....???

Hay! Don't panic its just for reference's sake its too easy if you already learn Apex Programming

Environment Setup for LWC & Creating first LWC

Firstly, install the VS Code on your desktop/pc

Why VS Code...? Is it possible to create a Lightning Web Component in Developer console?

NO, it's no possible to create a LWC on Developer console.

Follow This step that's it:

Step 1: Install Salesforce CLI Click here

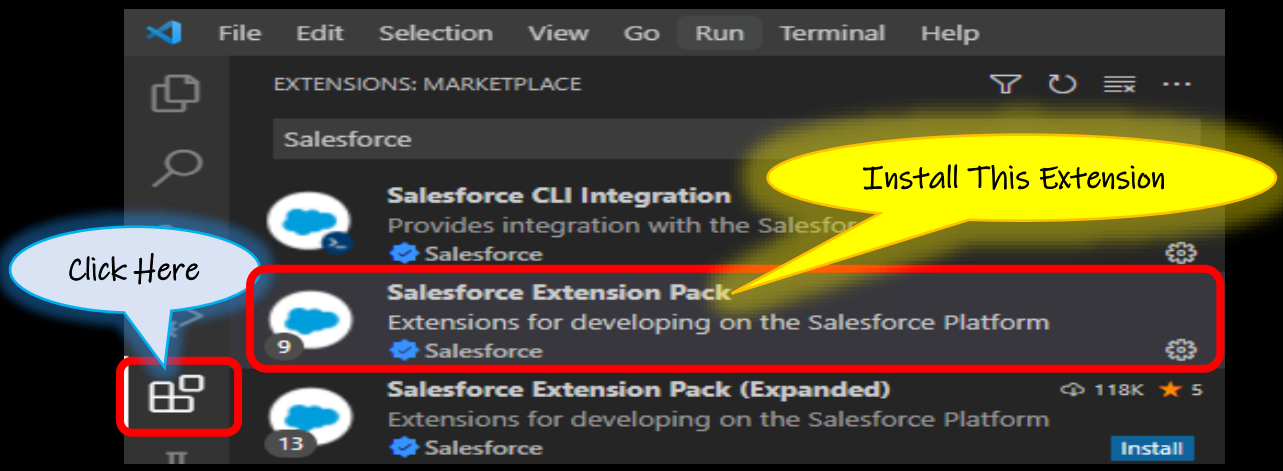
- ✓ Install like how you install any software in your desk 😂
- ✓ Check its installed or not Open you CMD and Type "SFDX "hit Enter if it shows Version info you're Done!

```
C:\Users\oleti>SFDX
Salesforce CLI

VERSION
  sfdx-cli/7.161.0 win32-x64 node-v16.16.0
```

Step2: Install VS Code **Click here ←**

- ✓ Install like how you install any software in your desk 😂

Step 3: Launch VS Code & Salesforce Install Extension Pack see this Image



Step 4: Enable dev Hub in Salesforce org

>>> open org >>> setup >>> Search for Dev Hub in the Quick Find box >>> Select Dev Hub >>> Enable that Dev Hub toggle

How to Create a Project?

Ok ...

1.Open Visual Studio Code.
2.Press Command + Shift + P on macOS or Ctrl + Shift + P on Windows or Linux, then type create project.
3.Select SFDX: Create Project, and press Enter.
4.Enter 'Name' of your 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. For the pop up.

Create a Scratch Org

simply we can say it's like 1 - 30 days sandbox

1.In Visual Studio Code,
2.press Command + Shift + P on macOS or Ctrl + Shift + P on Windows or Linux. Type sfdx. Select SFDX: Create a Default Scratch Org....

Creating First LWC Component:

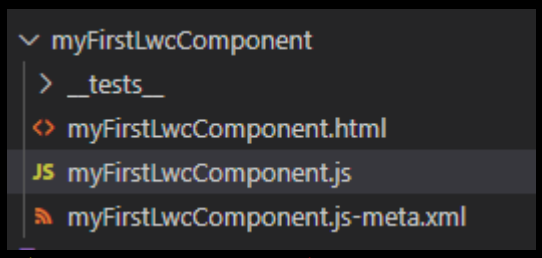
Step 1: Open VS Code → Press Ctrl +Shift + P on keyboard
Step 2: Select this SFDX: Create Lightning Web Component
Step 3: Enter You LWC Component Name starting Letter should be Small Letters >>> Enter >>>Enter

Things Keep in Mind

a) Name Must start with a Letter
b) Starting letter Must be small letter

😏 Hmm, I Don't use small letter in staring letter hoo Np 😊 VS code automatically changes to small caps for the staring letter.

As I created this Component



Here 1st file is HTML where we can write a UI code
And 2nd file is JS file mostly we can write Client-side business Logic
Finally, 3rd file is XML file it defines that where we have to place these components in salesforce like Home-Page, Record-Page like....

- Even we can add CSS file also....

Before going further Learn these things even basics all the basic stuff is available in W3Schools.com

- ✓ Basics of HTML
- ✓ CSS
- ✓ JavaScript

Discuss about Our First LWC Component

There are few things we need follow and focus while creating a Lightning Web Component.

- ✓ It Must begin with Lowercase letter.
- ✓ We can only use only Alpha, Numeric and Underscore characters.
- ✓ It cannot accept Hyphen (Dash) & Blank Space
- ✓ Cannot end with Underscore

Did you notice that there are only 3 files in that component.

We already discuss about those files Right.

Do you Know this we can also create 2 extra files also like CSS & SVG files:

- ✓ While creating these files we need remember one point that is "after creating those files the content of the file cannot be blank".
- ✓ If they are Blank, it throws error while deploy our source code to salesforce org

Let's see the JS file in our component.

This is like Base Class or we can call it as default class

myFirstLwcComponent.js

force-app > main > default > lwc > myFirstLwcComponent > JS myFirstLwcComp

```
1  import { LightningElement } from 'lwc';
2
3  export default class MyFirstLwcComponent extends LightningElement
4  {
5
6  }
```

😳 What is import & export...?

Here Line 1 says that importing the "Lightning Element" from the "LWC" module.

Line3 says that export the present LWC component JS file and extend the default LightningElement Class with the current component JS file.

😏😐😏 OK let's stop this ... Boring Part

Now were directly diving into LWC sea.

Scenario 1: In Our First LWC Component, Display the "Hello Welcome to Salesforce Simplified" text 3 times.

Note: All html tags do not work in LWC

Our Html file: ⬇

```
<template>
    <div>
        Hello Welcome to Salesforce Simplified <br/>
        Hello Welcome to Salesforce Simplified <br/>
        Hello Welcome to Salesforce Simplified
    </div>
</template>
```

Ok ....! we write code 😢 write how can we see ... is that code is rendered properly or not.

✓ Simple! Use Aura App Dude 😎😎

We can create Aura app in 2 ways in VSCode & Developer console

In VS Code Open VS Code → click Ctrl +Shift+P → Select SFDX: Create Aura App →give Name → Enter → Enter → It will create some files shown below

In that file we place our LWC component inside that app
How...? Let's see

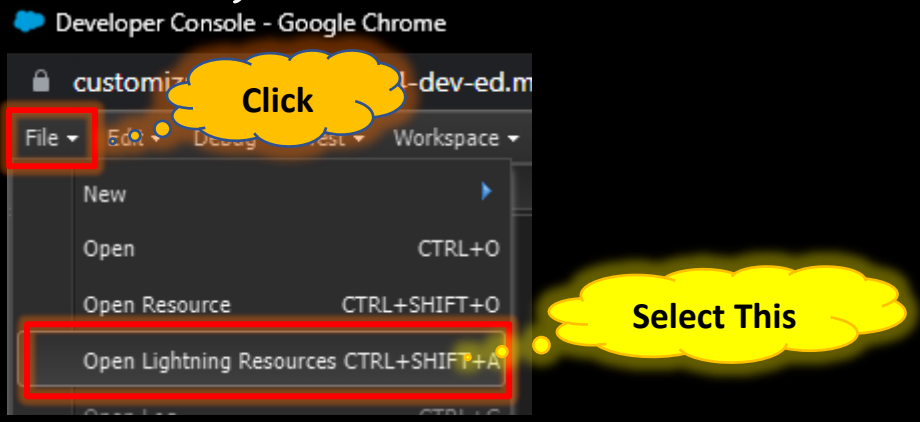
Test.app

force-app > main > default > aura > Test > Test.app >

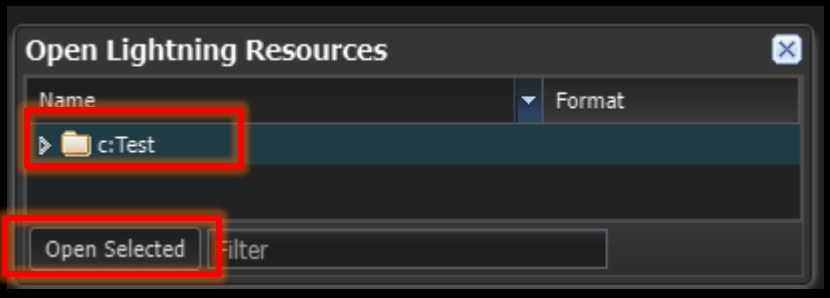
```
1  <aura:application>
2      <c:myFirstLwcComponent/>
3  </aura:application>
```

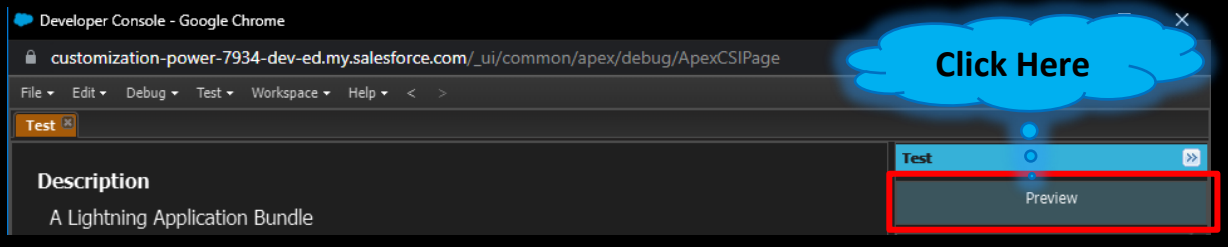
Here we place our component C: Defined that it's a component or we can say its Custom Component

At last, open the Developer console >>> And open Our Aura App
See below images



Then Search for our app >>> and click Down Open selected app





Output Like This: ➲

Hello Welcome to Salesforce Simplified
Hello Welcome to Salesforce Simplified
Hello Welcome to Salesforce Simplified

More Examples on LWC

Scenario 1:

Create one Aura App, and keep inside 2 LWC components in that app.

Use these similar while practicing for better understanding.

Aura app name: *Test*

LWC Component 1: *myFirstLwcComponent*

LWC Component 2: *mySecondLwcComponent*

→This is 1st component Html file code:

```
<template>
    <b><u>This is 1st First Component</u></b><br/>
    <b>If this text is visible 1st Component is working fine</b>
</template>
```

→This is 2nd component Html file code:

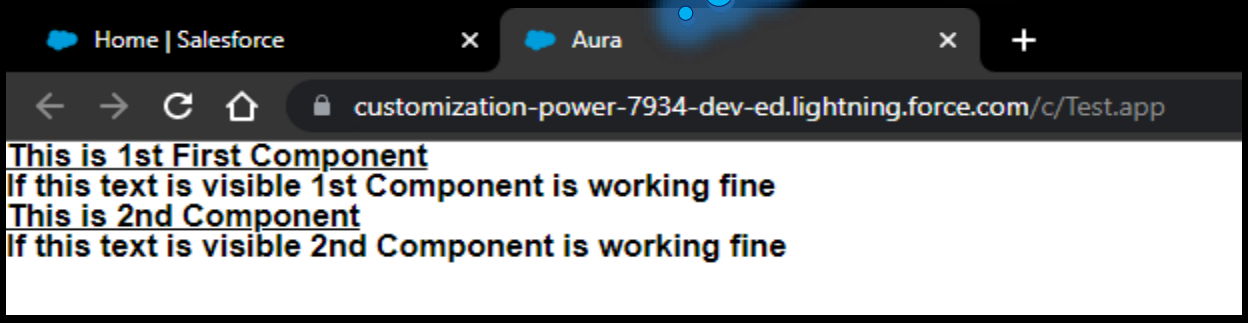
```
<template>
    <b><u>This is 2nd Component</u></b><br/>
    <b>If this text is visible 2nd Component is working fine</b>
</template>
```

→ This our Aura – Test application

```
<aura:application>
    <c:myFirstLwcComponent/> <br/>
    <c:mySecondLwcComponent/>
</aura:application>
```

Can we implement 2 components in one app?
Answer is **Yes** we can , How ? explain this

Output:



😍😍 Ara baba! Don't stop here do practice some inline CSS in components bring up you hidden talents dude.

🤔🤔🤔What if...?

We need to implement one Lwc component in another Lwc Component

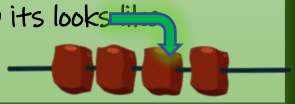
Here I'm using above 2 components!

For this example:

App Code:

```
<aura:application>
    <c:myFirstLwcComponent/>
</aura:application>
```

🥹🥹Hhuu? Wait what is this we discuss about camel casing right this is not looks like that!
Hay nice! your right this is different from that the name of this casing is kabab casing see how its looks like



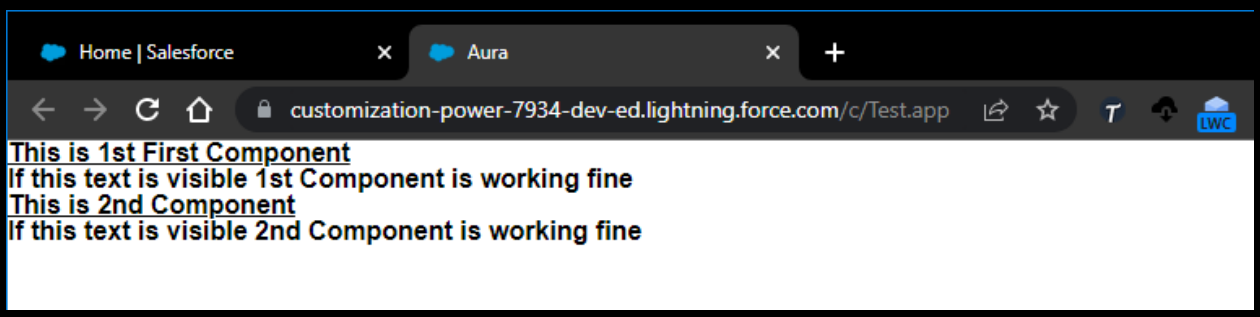
Component 1 – Code:

```
<template>
    <b><u>This is 1st First Component</u></b><br/>
    <b>If this text is visible 1st Component is working fine</b> <br/>

    <c-my-second-lwc-component></c-my-second-lwc-component>
</template>
```

Component 2 – Code:

```
<template>
    <b><u>This is 2nd Component</u></b><br/>
    <b>If this text is visible 2nd Component is working fine</b>
</template>
```

Output:



Here don't panic the output remains same because we use previous same components the main objective of this example is can we implement one component in side another component?

Answer is Yes, we can! Explain this scenario in detail.

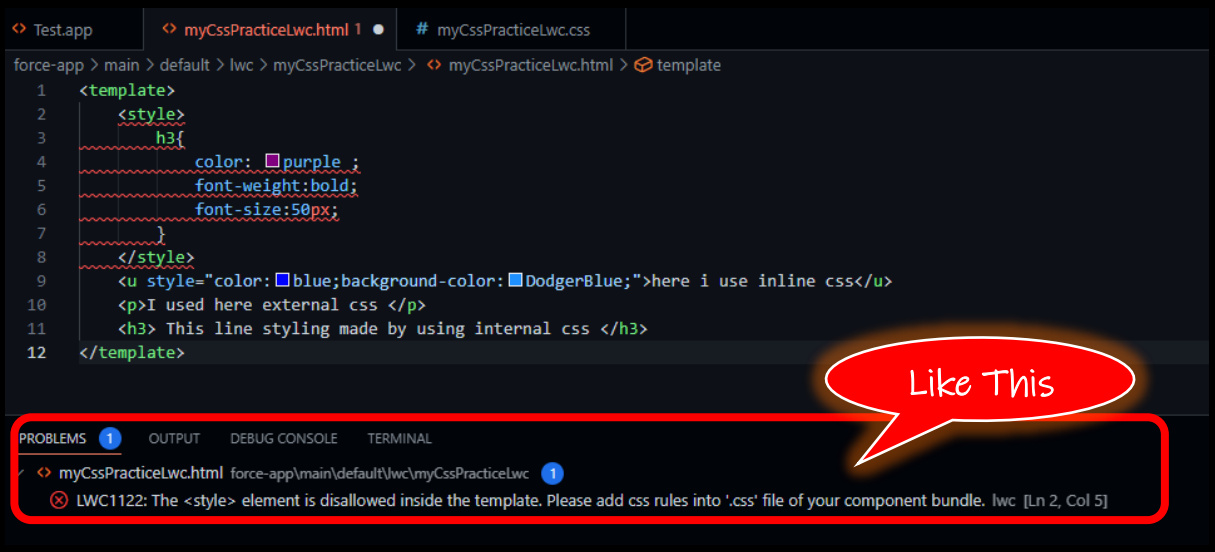
Can we use CSS in LWC?

Yes!

we can use but the problem is we already know there is 3 types of CSS

1. Inline CSS
2. Internal CSS
3. External CSS

The Problem is LWC not allowed you to use Internal CSS/Embedded CSS. If you use it shows error message like this.



✓ Component - HTML Code:

```
<template>
    <u style="color:White;background-color:Black;">here i use inline css</u>
    <p>I used here external css </p>
    <h3> This line styling made by using internal css 😁 We cannot use in LWC this type css </h3>
</template>
```

✓ Component - CSS Code:

```
p {
    color: red;
    text-align: center;
    font-family: verdana;
}
```

✓ Output:

here i use inline css

I used here external css

This line styling made by using internal css 😁 We cannot use in LWC this type css

Style Components with Lightning Design System

Salesforce Lightning Design System (SLDS) is a CSS framework that provides same look and feel that we already see in Lightning Experience UI in our Org.

Ok Why we have to use this because if we use this life become easier.

See this scenario If we need one button like exactly looks like save button and that button looks like this 

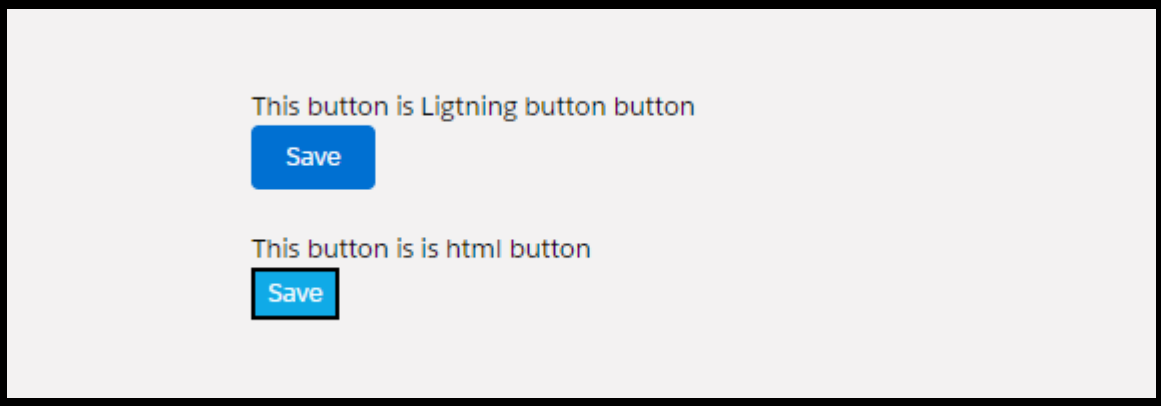
by using SLDS we can build that button easily but by using CSS its complicate to build same button.

Ok let's see this example:

- ✓ Component – HTML Code:

```
<template>
    <div>
        This button is Ligtning button button  <br/>
        <lightning-button label="Save" variant="brand"></lightning-button>
<br/> <br/>
        This button is is html button<br/>
        <button style="background-color: rgb(16, 170, 231); color:white; ">
Save </button>
        </div>
</template>
```

- ✓ Output:



See even simple styling is also not enough need extra styling need for that html button

At last, we say use LWC Component reference Page, Hmm! link? HERE ←

There are lot of examples see that page once in detail

Decorators in Lightning Web Component

In LWC we have 3 Unique Decorators that adds functionality to property or function.

Decorators alter the functionality of function/property dynamically.
Before we discuss we discuss about a small topic,
How to assign a value from JS file to Html...?
By using Data Binding:
We can define the values in JS file, and we can access them from LWC based on need we can we use those.
See this example:

Html -Code:

```
<template>
    <lightning-card  title="Here im going to display Static data ">
        My Name : Sravan <br/>
        Im from : India
    </lightning-card>
    <lightning-card  title="Here im going to display data in Dynamically ">
        My Name : {name}<br/>
        Im from : {country}
    </lightning-card>
</template>
```

JS – File -Code:

```
import { LightningElement } from 'lwc';

export default class FirstCompLwc extends LightningElement
{
     name = 'sravan from js';
     country = 'India from Js ';
}
```

Here the name and country values dynamically pass from JS file to Html file

App – Code:

```
<aura:application extends="force:slds">
    <c:firstCompLwc></c:firstCompLwc>
</aura:application>
```

Output:

Ok Let's come to decorators in LWC ....!

Simply, decorators define that whether this function/property accessible within the component or outside of the component.

3 types of decorators

- ✓ @track
- ✓ @api
- ✓ @wire

@api: This decorator will make a JS Properties & Function as "Public" means we can access these from another component not only access we can pass values and also, we can invoke child component from Parent component.

@track: Track decorator make a function/property as a "Private" means we can access that function within the component

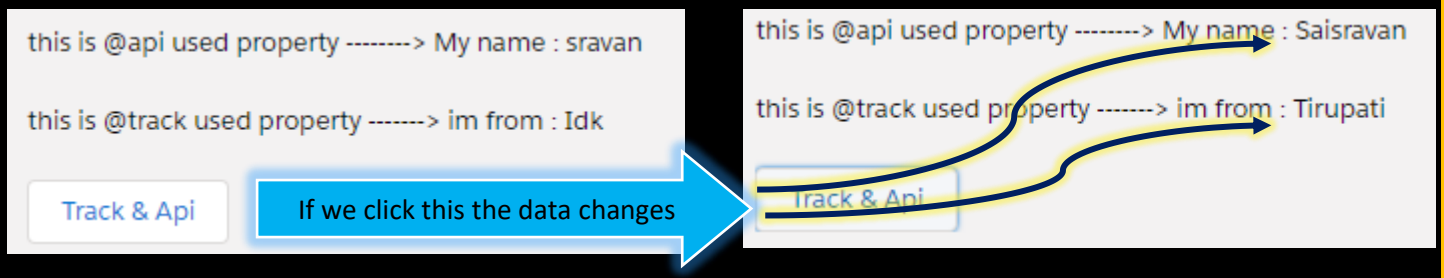
Ok I take an example

I have 2 components

1 → firstCompLwc (Parent)

2 → secondCompLwc (Child)

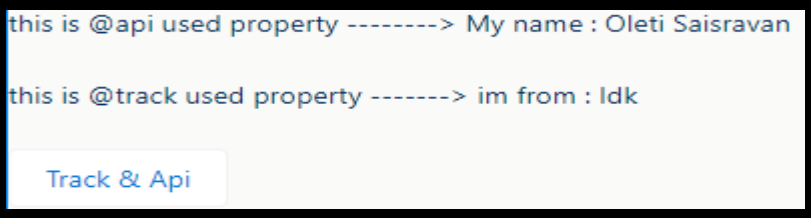
First see this I write a code in child component looks like this once we open



Ok I need t change my name and city in Parent component is this possible. yes, it is possible only @api decorated variables are changeable not other one.

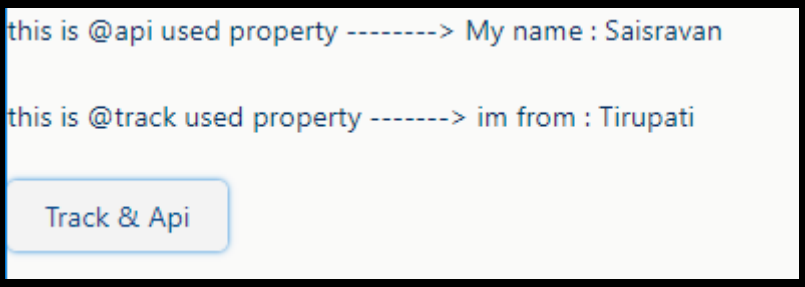
```
<template>
    <c-second-comp-lwc myname = 'Oleti Saisravan' city = 'Delhi'></c-second-comp-lwc>
</template>
```

Output :



See the one **@api** decorated property gets changes but not **@track** decorated property

Once if we click the page gets re render and gives the value like this



We will discuss about wire later as of now just know some definition of wire

@Wire:

Wire is used to call the inbuilt data services like to read the salesforce data

Like apex classes and method ....

For example:

We have a Apex class: StudentClass → inside - One method name: dataStu

We need to use those functions in our component.

We need to call like this: in JS file

Syntex:

```
import apexMethodName
from'@salesforce/apex/Namespace.Classname.apexMethodReference';
```

Eg:

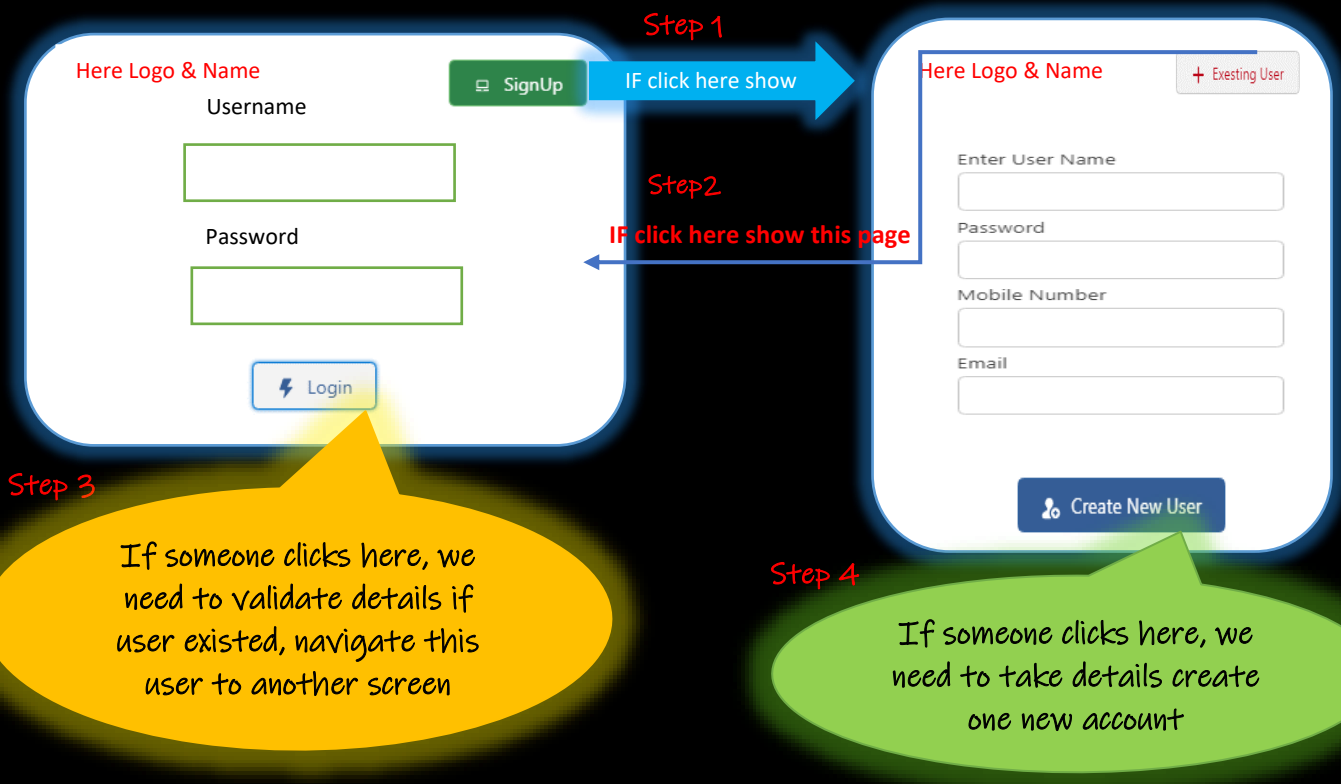
```
import dataStu
 from '@salesforce/apex/studentclass.datastu;
```

and this is also Mandatory keep this annotation in front of the method in your apex class.

```
@AuraEnabled(cacheable=true)
```

Mini Project LWC

Our scenario is we need this output developed by using LWC!



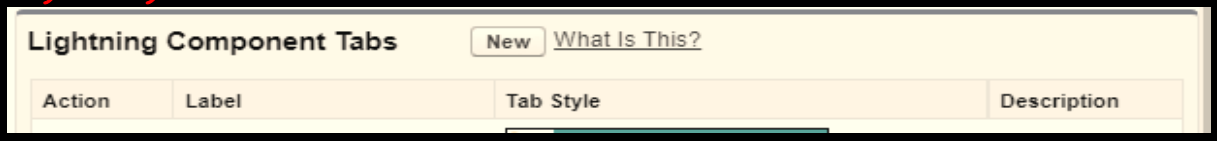
Here Step 1 & 2 both covered in one topic condition-based rendering in JS.
Here step 3&4 done by using some wire properties and Apex Code.
I already made a code for step 1 & 2. I mean UI part.

XML file Code-

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>55.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__Tab</target>
    </targets>
</LightningComponentBundle>
```

I need to make this component exposed & make available for Custom tabs
After deploying this component do this.

In Quick Find search (Tabs)→ Select Tab → Scroll Down →Click new on Lightning Component Tabs.



HTML file Code-

```
<template>
    <lightning-card title="Login form By Using LWC" icon-
name='standard:agent_home' variant='warning' size="large">
        <div class="slds-align_absolute-center">
            <lightning-input type="text" label="Enter User Name"></lightning-
input>
        </div>
        <div class="slds-align_absolute-center">
            <lightning-input type="text" label="Password"></lightning-input>
        </div>

        <!-- If Our condition is true these things render -->
        <template if:true={fullForm}>
            <div class="slds-align_absolute-center">
                <lightning-input type="text" label="Mobile
Number"></lightning-input>
            </div>
            <div class="slds-align_absolute-center">
                <lightning-input type="text" label="Email"></lightning-input>
            </div><br/>
            <lightning-button label="Exesting User"  variant="destructive-
text" icon-name="utility:add" slot="actions"
onclick={loginControl}></lightning-button>
            <lightning-button label="Create New User" variant="brand" icon-
name="utility:adduser" slot="footer"></lightning-button>
        </template>

        <!-- If Our condition is false these things render -->
        <template if:false={fullForm}>
            <lightning-button label="SignUp" variant="success" icon-
name="custom:custom27" slot="actions" onclick={signupControl}></lightning-
button>
            <lightning-button label="Login" variant="brand-outline" icon-
name="utility:connected_apps" slot="footer"></lightning-button>
        </template>
    </lightning-card>
</template>
```

JS file Code-

```
import { LightningElement } from 'lwc';

export default class MyLoginPage extends LightningElement
{
    fullForm = false;
    signupControl(event)
    {
        this.fullForm = true;
    }
    loginControl(event)
    {
        this.fullForm = false;
    }
}
```

As of now our front-end part is completed what next...?
We Capture the data from the input fields and store it in JS
From JS we can connect Apex.

Summary of what are the changes we made!!

The Below code is final code now below code is a bit enhanced version of the above code then these changes is mainly focused on how server-side controller talks with Clint side controller pass data and get data.

Things before seeing below code changes.

Please read below Concepts:

- ✓ Show Toast Event in LWC
- ✓ NavigationMixin in LWC
- ✓ How to Call Apex methods from JS in LWC
- ✓ Arrow Functions in JS

I'm assuming you people are already familiar with the topics and Apex programming, but if not, look at the Salesforce documentation.

We use Toast Events when a user clicks on Create record and saves data in Salesforce. We also use it when a user logs in using their credentials and when an issue occurs.

Toast Events Documentation Link **CLICK Here** ←

When a user attempts to log in, our client-side JS verifies their credentials by connecting to the server. If the client-side controller receives a response, we then use NavigationMixin to navigate the user's current screen to one of the tabs in the same navigation.

NavigationMixin Documentation Link **Click Here** ←

My Apex Controller code: Name of the Class is → LoginControllerLwc

```
public class LoginControllerLwc {
    @AuraEnabled
    public Static void createNewUser(String uName, String passWd, String phNo,
String mailId) {
        Login__c newLogin = new Login__c();
        newLogin.Name = uName;
        newLogin.Password__c = passWd;
        newLogin.Phone__c = phNo;
        newLogin.Email__c = mailId;
        insert newLogin;
    }
    @AuraEnabled
    public static List<Login__c> validateLogin(String Uname, String passWd) {
        List<Login__c> listOfLogis = [SELECT Id,Name,Password__c from Login__c
WHERE Name=:Uname AND Password__c=:passWd];
        return listOfLogis;
    }
}
```

Html Code :

```
<template>
    <lightning-card title="Login form By Using LWC" icon-
name='standard:agent_home' variant='warning' size="large">
        <div class="slds-align_absolute-center">
            <lightning-input type="text" label="Enter User Name"
onchange={Uname}></lightning-input>
        </div>
        <div class="slds-align_absolute-center">
            <lightning-input type="text" label="Password"
onchange={passCode}></lightning-input>
        </div>

        <!-- If Our condition is true these things render -->
        <template if:true={fullForm}>
            <div class="slds-align_absolute-center">
                <lightning-input type="text" label="Mobile Number"
onchange={pno} ></lightning-input>
            </div>
            <div class="slds-align_absolute-center">
                <lightning-input type="text" label="Email"
onchange={mail}></lightning-input>
            </div><br/>
            <lightning-button label="Exesting User"  variant="destructive-
text" icon-name="utility:add" slot="actions"
onclick={existingControl}></lightning-button>
            <lightning-button label="Create New User" variant="brand" icon-
name="utility:adduser" slot="footer" onclick={newUserCreate}></lightning-
button>
        </template>

        <!-- If Our condition is false these things render -->
        <template if:false={fullForm}>
            <lightning-button label="SignUp" variant="success" icon-
name="custom:custom27" slot="actions" onclick={signupControl}></lightning-
button>
            <lightning-button label="Login" variant="brand-outline" icon-
name="utility:connected_apps" slot="footer"
onclick={loginHandler}></lightning-button>
        </template>

    </lightning-card>
</template>
```

Js File:Code

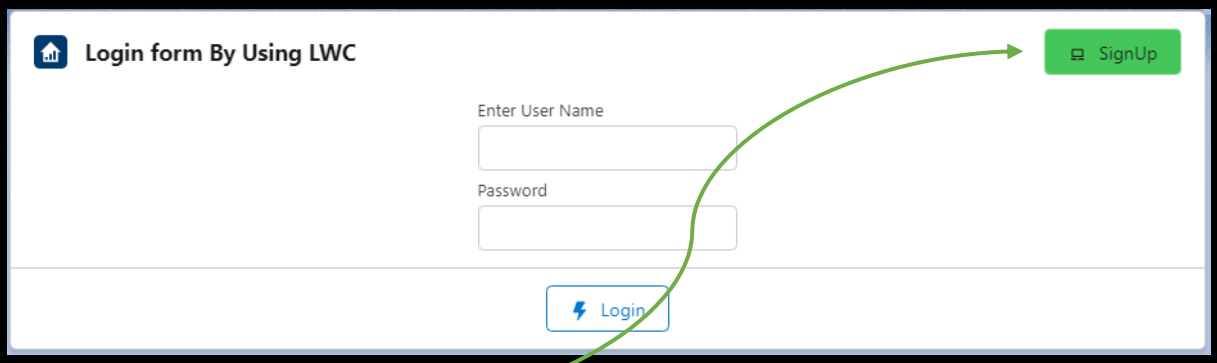
```
import { LightningElement } from 'lwc';
import createNewUser from '@salesforce/apex/LoginControllerLwc.createNewUser';
import loginAccess from '@salesforce/apex/LoginControllerLwc.validateLogin';
import { ShowToastEvent } from 'lightning/platformShowToastEvent';
import {NavigationMixin} from 'lightning/navigation'

export default class MyLoginPage extends NavigationMixin(LightningElement )
{
    //Some functions used to condition based rendering
    fullForm = false;
    signupControl(event)
    {
        this.fullForm = true;
    }
    existingControl(event)
    {
        this.fullForm = false;
    }

    // Capture data for User name Password,Email,Phone
    UserName;
    Uname(event)
    {
       this.UserName = event.target.value;
    }
    Pwd;
    passCode(event)
    {
       this.Pwd = event.target.value;
    }
    PhoenNo;
    pno(event)
    {
       this.PhoenNo = event.target.value;
    }
    eMail;
    mail(event)
    {
       this.eMail = event.target.value;
    }


    //Using these variables in Toast Messages
    title = 'Sucess';
    message = 'HoolyMolly! New User Created Sucessfully';
    variant = 'success';
```

```
    //To create a new user
    newUserCreate()
    {
        // alert(`User Name : ${this.UserName}  \n Password Entered :
${this.Pwd} \n Phone no is: ${this.PhoenNo} \n Email is : ${this.eMail}`);
        createNewUser({
            uName:this.UserName , passWd:this.Pwd, phNo:this.PhoenNo,
mailId:this.eMail
        })
        .then(result =>{
            const evt = new ShowToastEvent({
                title: this._title,
                message: this.message,
                variant: this.variant,
            });
            this.dispatchEvent(evt);
        })
        .error(error =>{});
    }
    //below code handle if user click on login
    loginHandler(event)
    {
        //alert(`User Name : ${this.UserName} and Password Entered :
${this.Pwd}`);
        loginAccess({Uname:this.UserName, passWd:this.Pwd})
        .then(result => {
        if(result && result.length>0)
        {
            console.log(result[0])
            console.log(typeof(result))
            const evt = new ShowToastEvent({
                title: 'Login Sucess',
                message: 'Hohoo 💯✅! Your are Authorized User 😎 . We are
moving to Search Books Application',
                variant: 'success',
            });
            this.dispatchEvent(evt);
            this[NavigationMixin.Navigate]({
                type:'standard__navItemPage',
                attributes:{
                    apiName:'Search_Books'
                }
            })
        }
        else
        {
            const evt = new ShowToastEvent({
                title: 'We 😓 Hit a Snag! User Not Found',
```

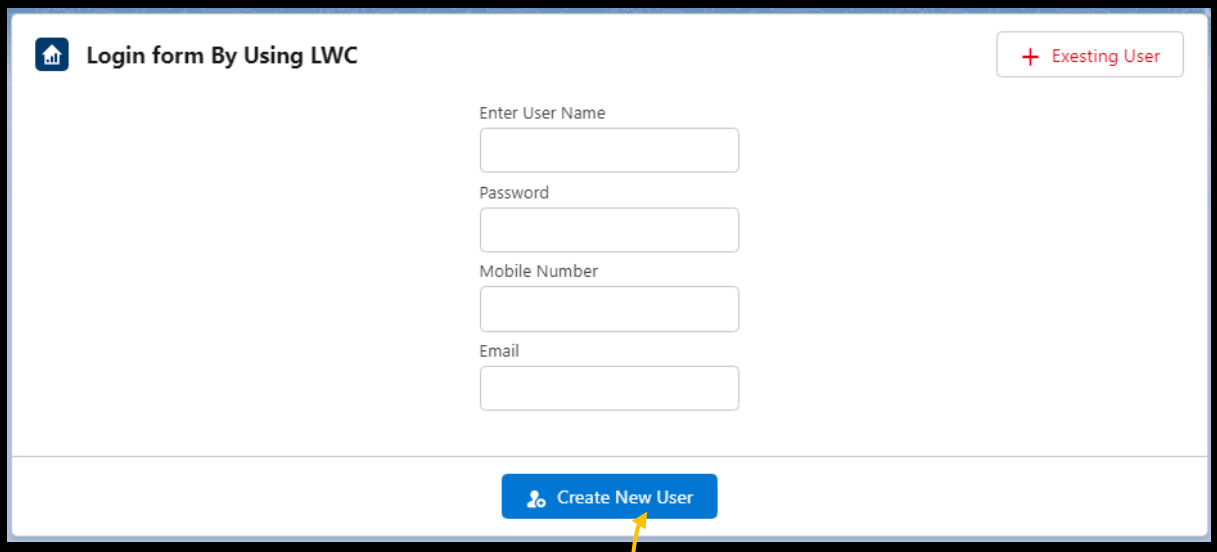
```
                message: 'Omg! 🙏🙏Create New User instead of Login',
                variant: 'warning',
            });
            this.dispatchEvent(evt);
        }
        }).catch(error => {});
    }
}
```

And There is no Changes in XML File

Here Some Screen Shots!



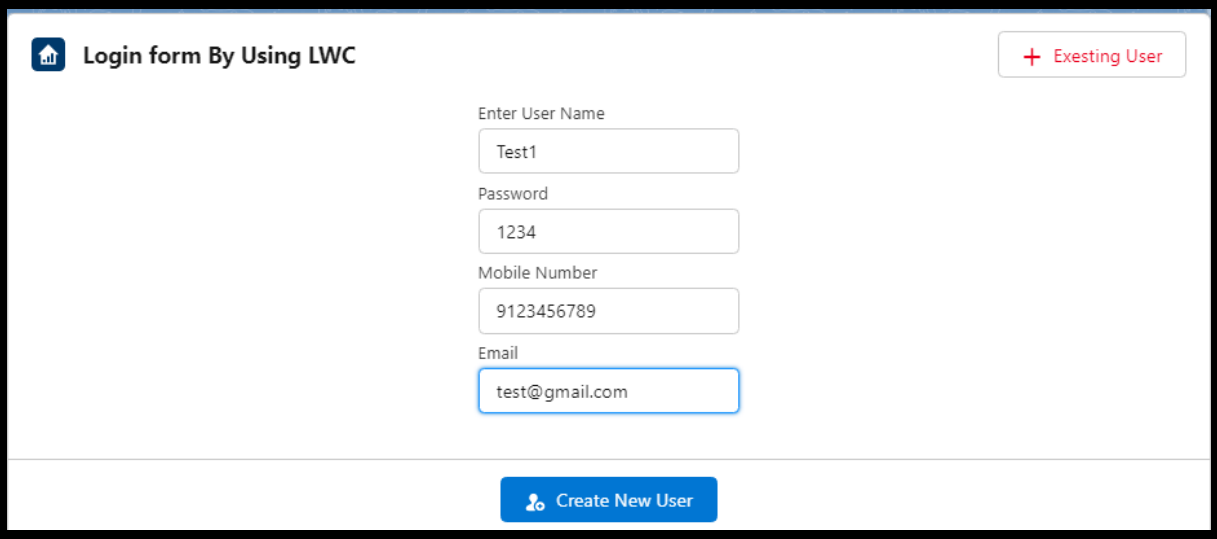
If user Clicks Signup



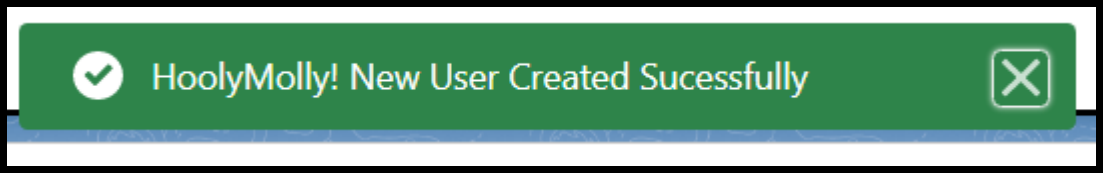
If the user fills the data and Click the Create new user automatically creates a record in Credentials object

Create an object: I named like this **Login__C** this is my API Name and these fields required

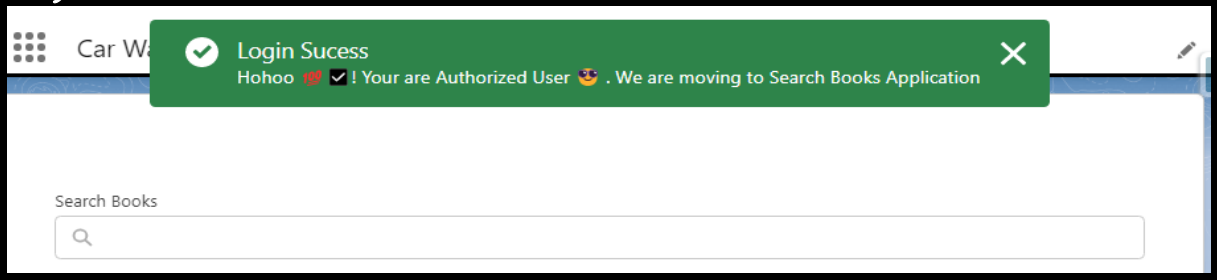
FIELD LABEL	FIELD NAME	DATA TYPE
User Name	Name	Text(80)
Phone	Phone__c	Phone
Password	Password__c	Text(20)
Email	Email__c	Email



See I have given data to create a record if I click record one toast message come



If I login with before details its show one toast message and redirect page to another tab



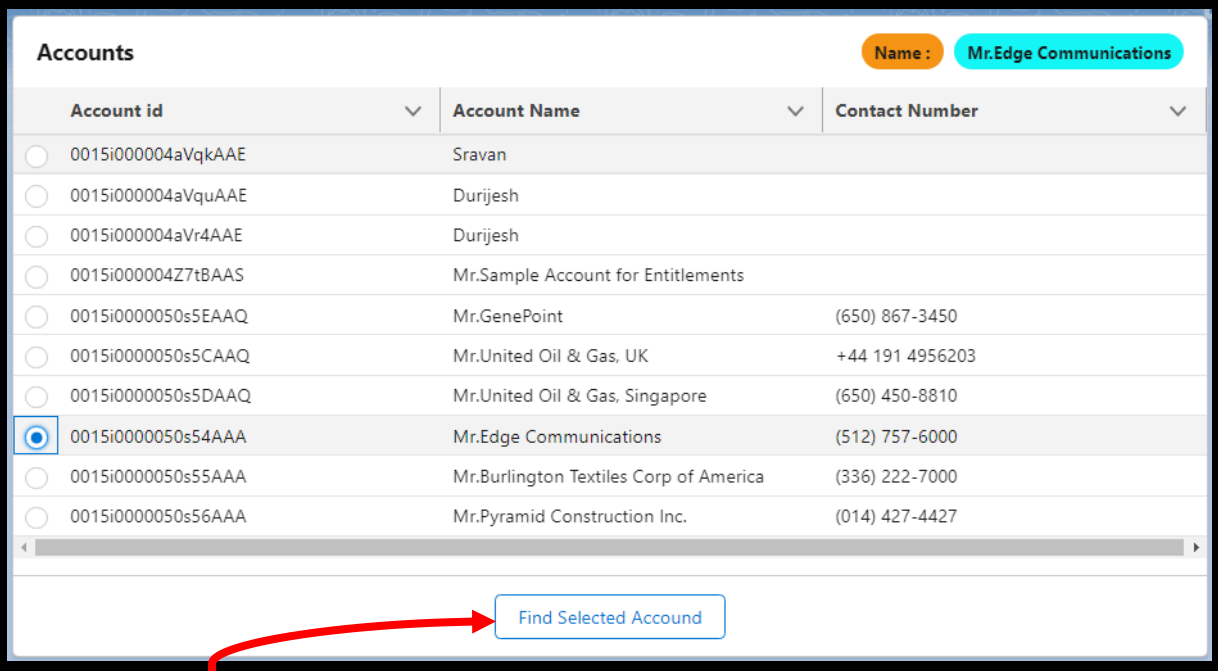
Now everything is going well.

☝ Don't Stop here I can give some ideas if you like follow and do that in your org and for above code also don't copy code first of all analyse the requirement and implement in your own way take time to think.

1. Create some validation rules
2. Write a trigger to check the duplicate name and email
3. If user create a new user get all the data and create one account along with contact.
4. Try to navigate to community pages if the user clicks. explore on this topic
5. Finally Create a data table on account object to show the 10 records in Lwc component if the user selects record, we need to display user name and if the user clicks the edit button, we need to give exact edit page of the user selected record.

[I know the last 5th one is not related to current project but in my case to fulfill my requirement I learn a lot in Lwc .]

See this exact 5th one looks like.



If I click here it moved to edit page of the current selected record.

- *Keep Learning Don't stop if you feel tough to understand.* 🙃