**PRACTICAL NO : 10A A65\_PRASANNA ANJANKAR**

**Aim:** A] To Create a Hello World Aura Component.

**Theory:**

**Salesforce Lightning Component Framework** is a User Interface (UI) framework developed to create dynamic Salesforce web applications for mobile and desktop devices. A [Salesforce Developer](https://www.tutorialkart.com/salesforce-tutorials/salesforce-developer-tutorials/), builds lightning components and [Salesforce administrator](https://www.tutorialkart.com/salesforce-tutorials/salesforce-administrator/), assembles lightning components and create **lightning page.**  Salesforce lightning page will be a record’s detail page, home page or a page used in SFDC.

## **What is Salesforce Lightning Component Framework?**

**Salesforce Lightning Component Framework** is a User Interface (UI) framework developed to create dynamic Salesforce web applications for mobile and desktop devices.



### **Why is the lightning component Framework is used?**

1. lightning component Framework provides the components to build applications faster. No need to start from the scratch.
2. Lightning component Framework tacks browser vents when a user click a button, scrolling and many more.
3. Our own components can be published and shared with other Salesforce users.
4. It used JSON to exchange data between the server and the client.
5. Salesforce Lightning Component Framework supports latest browser technologies like HTML5, CSS3 and touch events.

## **What is Salesforce Lightning Component?**

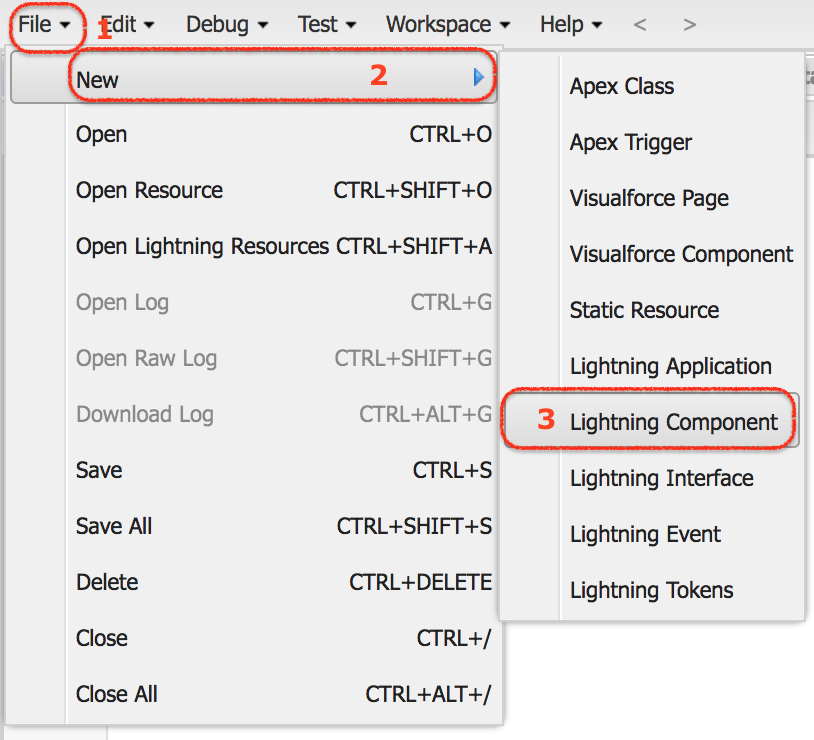
Components are self-contained and reusable units of an application. Any lighting application is built using these small Salesforce Lightning Components and these components are reusable.

* Lightning components have .cmp extensions.
* Lightning components are based on the concept of web components.
* It uses Apex, JavaScript, HTML and CSS to build Salesforce lightning app.

### **How to create Salesforce Lightning Component?**

To create Salesforce Lightning Components, log in to your Salesforce developer account and navigate to Developer Console.

* Now go to File | New | Lightning Component.
* Name the lightning component with your desired name and click on **Save button.**



**Code :**

1. **HelloWorld1.cmp**

<aura:component >

<h1>Hello World!!</h1>

</aura:component>

1. **HelloWorld2.cmp**

<aura:component >

<h1>Welcome to Salesforce Lightning Components</h1>

<h1>Prasanna Anjankar</h1>

1. **HelloWorld3.app**

<aura:application >

<c:HelloWorld1 />

<c:HelloWorld2 />

</aura:application>

1. CSS

.THIS {

background-color:White;

padding-top:10px;

margin: auto;

width: 50%;

border: 1px solid green;

padding: 10px;

font-family:Arial, Helvetica, sans-serif;

}

h1.THIS {

font-size:30px;

margin: auto;

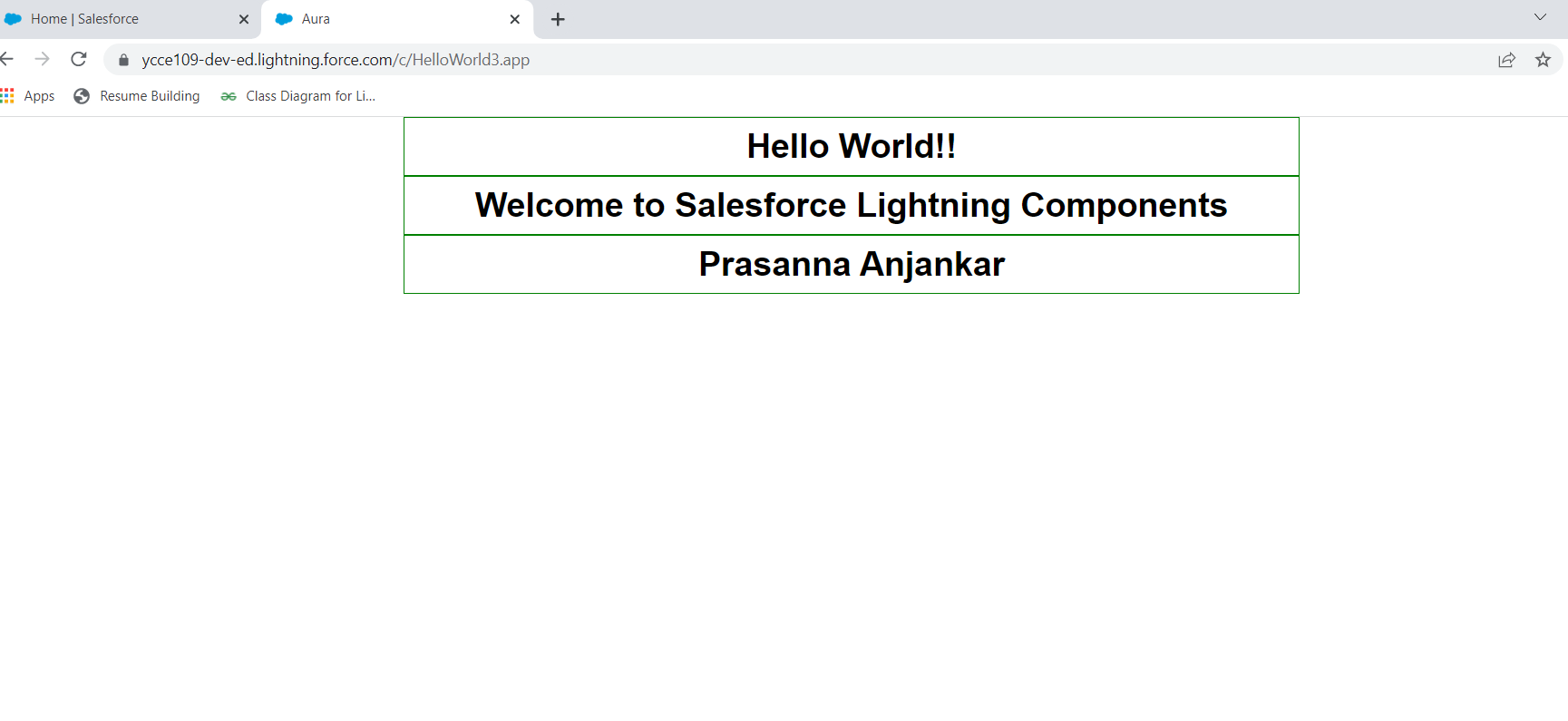
font-weight:bold;

padding-top:10px;

text-align: center;

}

**Screnshot Of Output :**



**Conclusion :** Hence I have Successfully created a Hello World Aura Component.

**Name : Prasanna Rajendra Anjankar Registration No : 19010886**

**Roll No : 65 Class : 6CTA**

**Course Name : CRM Course Code : CT2370**

**PRACTICAL NO : 10A A65\_PRASANNA ANJANKAR**

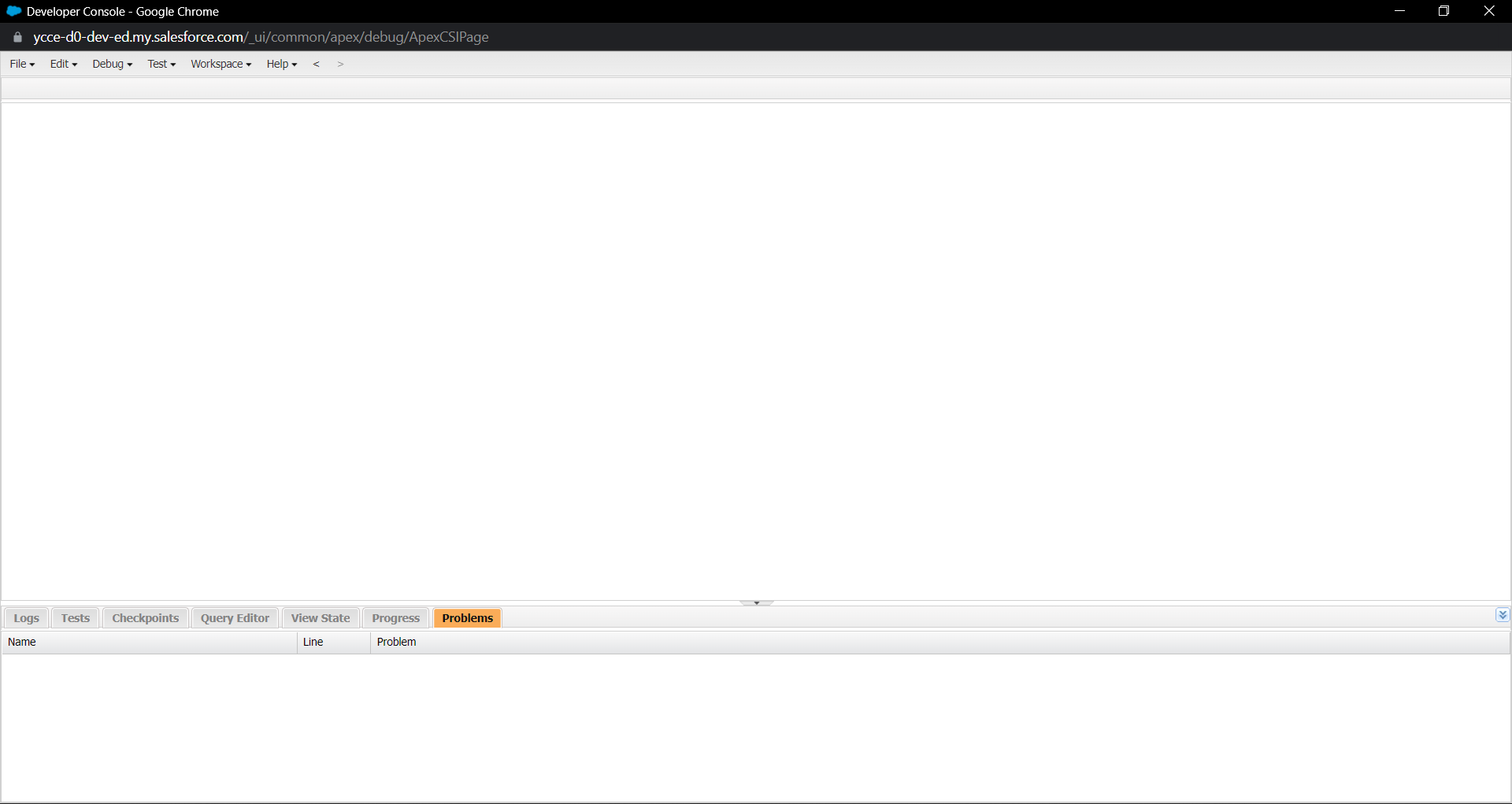
**Aim:** B] To Create and Edit Aura Component.

**Theory:**

**Creating and Editing Aura Components:**

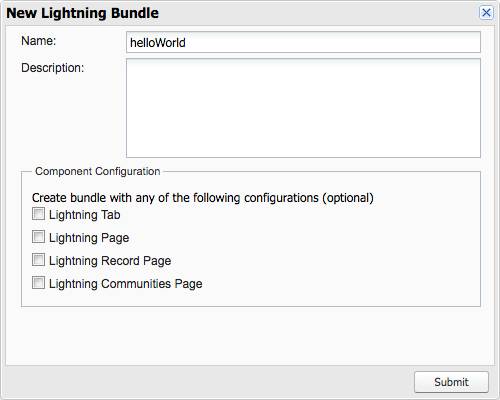
The first step to writing Aura component code. Fortunately, this is really easy. In your org, open the Developer Console under Your Name or the quick access menu ⚙️

| **Lightning Experience** | **Salesforce Classic** |
| --- | --- |
| Opening the developer console | Classic mode: Opening the developer console |



**Create Aura Components in the Developer Console**

So, let’s write something. **Select File | New | Lightning Component** to create an Aura component. In the New Lightning Bundle panel, enter **helloWorld** for the component name, and click Submit.



This creates a new **helloWorld** component bundle, with two open tabs. Close the **helloWorld** tab, and keep the **helloWorld.cmp** tab open.

**helloWorld.cmp** contains the opening and closing tags for an Aura component, **<aura:component>**. Between them, add the following markup, and save:

Now,

**Select File | New | Lightning Application** to create a new **Lightning app**. In the New Lightning Bundle panel, enter “ harnessApp” for the app name, and click Submit.

This creates a new **firstLightning** bundle, with two open tabs. Close the **firstLightning** tab, and keep the **firstLightning.app** tab open.

**firstLightning.app** contains the opening and closing tags for a Lightning app, **<aura:application>**. Between them, add the following markup, and save:

<c:helloWorld/>

This adds the **helloWorld** component we created earlier to the **firstLightning** app.

**What Is an App?**

An app is just a special kind of component! For the purposes of this module, you can think of an app as being different from a component in only two meaningful ways:

An app uses **<aura:application>** tags instead of **<aura:component>** tags. Only an app has a Preview button in the Developer Console.

**What Are Apps For?**

The main items are the following.

* When writing markup, you can add a component to an app, but you can’t add an app to another app, or an app to a component.
* An app has a standalone URL that you can access while testing, and which you can publish to your users. We often refer to these standalone apps as “my.app.”
* You can’t add apps to Lightning Experience or the Salesforce app—you can only add components. <aura:component> An Aura components app—that is, something defined in a <aura:application> —can’t be used to create Salesforce apps.

**Code :**

1. **HelloWorld1.cmp**

<aura:component >

<h1>Hello World!!</h1>

</aura:component>

1. **HelloWorld2.cmp**

<aura:component >

<h1>Welcome to Salesforce Lightning Components</h1>

<h3>Prasanna Anjankar</h3>

<p>Styling to this component!</p>

<div>

<div class = "red">User 1 - Red</div>

<div class = "yellow">User 2 - Yellow</div>

<div class = "green">User 3 - Green</div>

</div>

</aura:component>

1. **HelloWorld3.app**

<aura:application >

<c:HelloWorld1 />

<c:HelloWorld2 />

</aura:application>

1. CSS

.THIS {

background-color:White;

padding-top:10px;

margin: auto;

width: 50%;

border: 1px solid green;

padding: 10px;

font-family:Arial, Helvetica, sans-serif;

}

h1.THIS {

font-size:30px;

margin: auto;

font-weight:bold;

padding-top:10px;

text-align: center;

}

.THIS {

background-color:White;

padding-top:10px;

margin: auto;

width: 50%;

border: 1px solid green;

padding: 10px;

font-family:Arial, Helvetica, sans-serif;

}

h1.THIS {

font-size:30px;

margin: auto;

font-weight:bold;

padding-top:10px;

text-align: center;

}

h3.THIS {

font-size:30px;

margin: auto;

font-weight:bold;

padding-top:10px;

text-align: center;

}

.THIS .red {

background-color: red;

}

.THIS .green {

background-color: green;

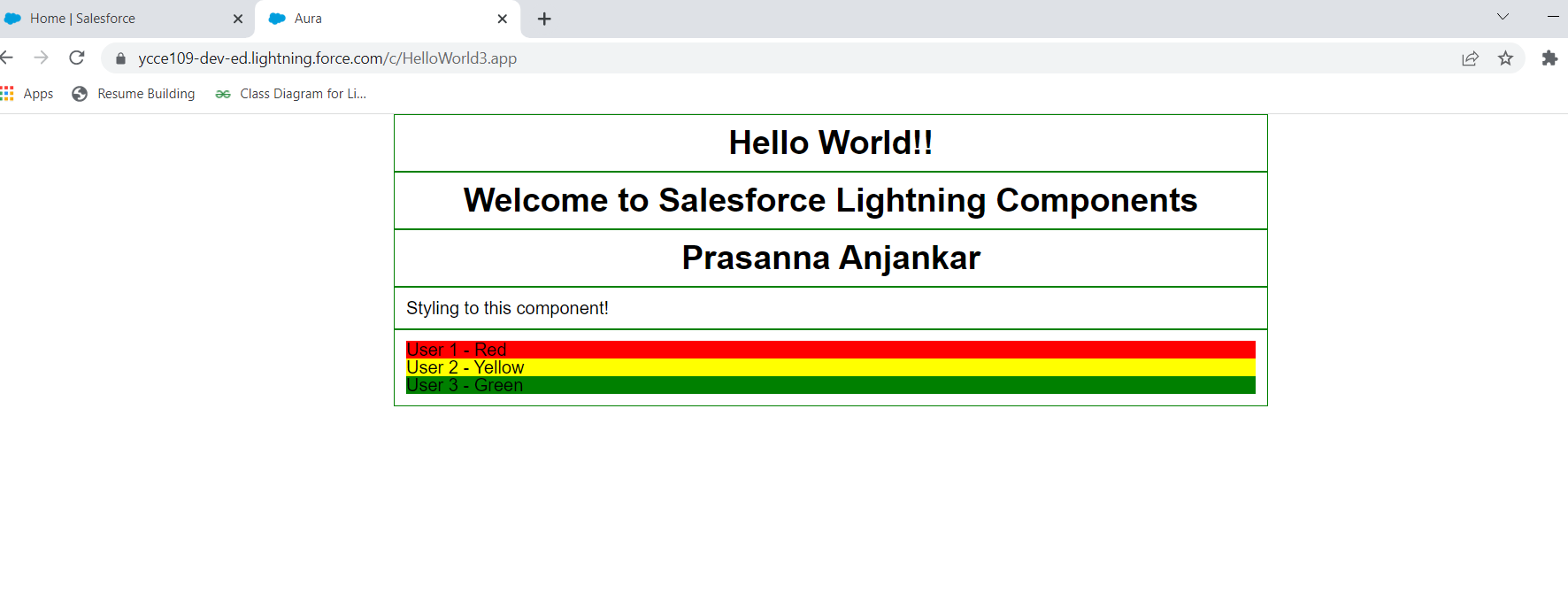
}

.THIS .yellow {

background-color: yellow;

}

**Screenshot of Output :**



**Colclusion :** Hence I have successfully created and edited the Aura Compunent.

**Name : Prasanna Rajendra Anjankar Registration No : 19010886**

**Roll No : 65 Class : 6CTA**

**Course Name : CRM Course Code : CT2370**