**How to generate pdf from LWC component**

the implementation of a Lightning Web Component (LWC) that generates a PDF using the jsPDF library, I will provide you with detailed steps for setting up, integrating, and enhancing the solution.

## **Step 1: Add jsPDF as a Static Resource**

### **1. Download jsPDF Library**

* Visit [jsPDF GitHub repository](https://github.com/parallax/jsPDF).
* Download the latest stable release as a .js file (e.g., jspdf.umd.min.js).

### **2. Upload to Salesforce as a Static Resource**

1. Navigate to **Setup** in Salesforce.
2. Search for and open **Static Resources**.
3. Click on **New**.
4. Enter the following:
   * **Name**: jspdf
   * **File**: Upload the jspdf.umd.min.js file you downloaded.
   * **Cache Control**: Set to Public.
5. Save the resource.

## **Step 2: Create the LWC**

In your Salesforce development environment (e.g., Visual Studio Code), create an LWC component folder named pdfGenerator and add the following files.

### **2.1 pdfGenerator.html**

This file defines the UI of your component. It includes an input field to enter text and a button to generate the PDF.

<template>

<lightning-card *title*=*"PDF Generator"* *icon-name*=*"custom:custom63"*>

<div *class*=*"slds-p-around\_medium"*>

<lightning-input *label*=*"Title"* *value*={*pdfTitle*} *onchange*={*handleTitleChange*}></lightning-input>

<lightning-button *label*=*"Generate PDF"* *onclick*={*generatePDF*}></lightning-button>

</div>

</lightning-card>

</template>

### **2.2 pdfGenerator.js**

This file contains the JavaScript logic for your component. It imports the jsPDF library and provides methods to handle input and generate the PDF.

import { LightningElement, track } from 'lwc';

import { loadScript } from 'lightning/platformResourceLoader';

import jsPDF from '@salesforce/resourceUrl/jsPDF';

export default class PdfGenerator extends LightningElement {

@track pdfTitle = '';

@track pdfContent = '';

jsPDFLib;

connectedCallback() {

loadScript(this, jsPDF)

.then(() => {

this.jsPDFLib = window.jspdf.jsPDF;

})

.catch(error => {

console.error("Error loading jsPDF library", error);

});

}

handleTitleChange(event) {

this.pdfTitle = event.target.value;

}

generatePDF() {

if (this.jsPDFLib) {

const doc = new this.jsPDFLib();

doc.text(this.pdfTitle, 10, 10);

doc.text(this.pdfContent, 10, 20);

doc.save('generated.pdf');

} else {

console.error("jsPDF library is not loaded.");

}

}

}

### **2.3 pdfGenerator.js-meta.xml**

This configuration file defines where the component can be used within Salesforce.

<?xml *version*=*"1.0"*?>

<LightningComponentBundle *xmlns*=*"http://soap.sforce.com/2006/04/metadata"*>

<apiVersion>62.0</apiVersion>

<isExposed>true</isExposed>

<targets>

<target>lightning\_\_AppPage</target>

<target>lightning\_\_HomePage</target>

<target>lightning\_\_RecordPage</target>

<target>lightning\_\_FlowScreen</target>

</targets>

</LightningComponentBundle>

## **Step 3: Deploy the LWC to Salesforce**

1. Save all the files in the pdfGenerator folder.
2. Deploy the LWC to your Salesforce org using the Salesforce CLI or your development environment.

## **Step 4: Add the Component to a Salesforce Page**

1. Open **App Builder** for a Lightning page:
   * Go to **Setup** → **Lightning App Builder**.
   * Choose a page (Home Page, Record Page, or App Page).
2. Drag and drop the pdfGenerator component onto the page layout.
3. Save and activate the page.

## **Step 5: Test the Component**

1. Open the page where the pdfGenerator component is added.
2. Enter some text in the input field.
3. Click the **Generate PDF** button.
4. A PDF file named GeneratedDocument.pdf will be downloaded with the entered text.