**Dynamic Web Component Assignment: Invoice Builder**

**Goal:**  
The goal of this assignment is to design a dynamic web component for an **Invoice Builder** using **JavaScript, data, and config**. The entire structure and logic should be handled in **JavaScript**, ensuring that the **index.html** file only displays the dynamically generated output. Bootstrap classes should be used wherever possible, and CSS should only be applied if Bootstrap does not provide the required styling.

**GitHub Link for Sample Code and Design:**

Use this link: <https://github.com/apnasite>. Fork & Clone the repository, utilize the sample code provided, follow the specified design number, and complete the assignment accordingly.

**Assignment Instructions:**

**1. JavaScript Implementation (Data & Config):**

* **All code should be written in JavaScript.**
* The Invoice Builder should be dynamically generated using a **data object and a config object**.
* The **index.html** file should only contain the **web component output** without static content.
* The JavaScript file should:
  + Fetch and process the **data and config**.
  + Generate and inject the Invoice Builder dynamically.
  + Ensure flexibility by allowing different designs through configuration.

**2. HTML Structure:**

* The **HTML file should not contain static content**; it should only load the JavaScript file.
* The Invoice Builder layout should be dynamically built inside the .invoice-background div.

**3. Bootstrap and CSS Styling:**

* **Use Bootstrap classes wherever possible.**
* **Only use CSS when Bootstrap does not provide the required styling.**
* Apply a background gradient to the .invoice-background div to enhance the visual appeal of the invoice builder.
* Use custom web fonts for the text elements. You can explore Google Fonts or other font services.
* Apply appropriate positioning and transforms to achieve the desired layout.
* Utilize border-radius for rounded corners where necessary.

**4. Invoice Content:**

* Ensure that the content (invoice details, itemized list, customer info, etc.) is fetched dynamically from the **data object**.
* The invoice should include sections for:
  + Customer details (name, address, contact info, etc.)
  + Itemized list (product name, quantity, unit price, total price)
  + Tax, shipping, and total
  + Invoice date and due date
  + Invoice number
* Make sure the content is displayed clearly with proper margins, paddings, and text alignment.

**5. Content Styling:**

* Style the invoice title, item names, quantities, unit prices, and total values.
* Add appropriate margins, paddings, borders, and text styling to make the invoice readable and professional.

**6. Customization Options:**

* Use the **config object** to allow users to customize various aspects of the invoice, such as font colors, background colors, section visibility, tax rates, currency format, etc.

**7. Measurement Unit (only if CSS used):**

* For width, use the mm unit.
* For font size and border, use the pt unit.
* Don’t use px or em or any unit other than mm and pt.

**Submission Guidelines:**

1. Create a folder for your assignment.
2. Include the **JavaScript file** handling all logic.
3. The **index.html** file should only display the web component output.
4. Include any additional images used in the design (if applicable).
5. Commit the assignment code and push it into the GitHub repository named mmeac-invoice-builder-assignment.

**Evaluation Criteria:**

✅ Proper use of **JavaScript for dynamic rendering**.  
✅ Clear and effective application of **Bootstrap and minimal CSS**.  
✅ Visual appeal and professionalism of the invoice design.  
✅ Responsiveness of the design for different screen sizes.  
✅ Creativity and attention to detail in the design.  
✅ **Bonus points** for incorporating additional features such as:

* Print functionality for invoices
* Download options for the invoice in PDF or other formats
* Ability to edit and save invoice content

**Note:**

Your design number will be available in the comments section of the Jira account. The code you will clone from GitHub will already include the sample code and design.