

Resume adding Button:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>DOM</title>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.2/css/all.min.css" />
</head>
<style>
  #download-button{
    text-decoration: none;
    background-color: black;
    padding: 10px 10px;
    text-align: center;
    color: aliceblue;
    cursor: pointer;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 2);
    font-family: sans-serif;
    font-weight: 900;
    border-radius: 2px;
  }
  #download-button:hover {
    background-color: white;
    border: 1px solid black;
    color: black;
  }
  i {
    margin-left: 6px;
  }
</style>
```

```

<body>
  <!-- Link url using javascript -->
  <button id="download-button">Download Resume<i class="fa-solid fa-
file"></i></button>

  <script src="resumadd.js"></script>
</body>

</html>

```

HTML Code:

1. Document Type Declaration (<!DOCTYPE html>):

- This declaration defines the document type and version of HTML being used.

2. HTML Structure:

- `<html lang="en">`: Specifies that the document is written in English.
- `<head>`: Contains metadata about the HTML document, such as character set, viewport settings, and the title of the page.
- `<meta charset="UTF-8">`: Sets the character set to UTF-8.
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Configures the viewport for responsive design.
- `<title>DOM</title>`: Sets the title of the HTML document.
- `<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css" />`: Links an external stylesheet (Font Awesome) for styling.

3. Inline CSS Styles:

- Styling for the download button and icon using the `<style>` tag.

4. Body Section:

- `<button id="download-button">Download Resume<i class="fa-solid fa-file"></i></button>`: Creates a button with the id "download-button" and an icon inside it.
- `<script src="resumadd.js"></script>`: Links an external JavaScript file named "resumadd.js."

```
const downloadButton = document.getElementById("download-button");
downloadButton.addEventListener("click", downloadFile);

function downloadFile() {
  const link = document.createElement("a");
  link.href = "https://app.luminpdf.com/viewer/658f9bde1a19cbd49b7e2650";
  link.download = "resume.pdf";
  link.click();
}
```

JavaScript Code (resumadd.js):

1. DOM Manipulation:

- `const downloadButton = document.getElementById("download-button");`: Retrieves the button element with the id "download-button" and stores it in the `downloadButton` variable.
- `downloadButton.addEventListener("click", downloadFile);`: Adds a click event listener to the button, which will execute the `downloadFile` function when the button is clicked.

2. downloadFile Function:

- `function downloadFile() {}`: Defines the `downloadFile` function.
- `const link = document.createElement("a");`: Creates an `<a>` (anchor) element dynamically.
- `link.href = "https://app.luminpdf.com/viewer/658f9bde1a19cbd49b7e2650";`: Sets the "href" attribute of the anchor element to the URL of the file to be downloaded.
- `link.download = "resume.pdf";`: Sets the "download" attribute of the anchor element to specify the default file name for the downloaded file.
- `link.click();`: Programmatically triggers a click on the anchor element, initiating the file download.

CSS Styles:

- Styling for the download button (`#download-button`) and its hover state.
- Styling for the icon (`i`) inside the button, adding some margin.

In summary, this code creates an HTML page with a button labeled "Download Resume" and an associated JavaScript function to initiate the download of a PDF file when the button is clicked. The styling is provided using both inline and external CSS.