STUDENT NAME	**R.ROJA**	
STUDENT REGISTRATION NUMBER	**251U1R2064**	**CLASS: CSE(AIML)**
PROGRAM	**UG**	**YEAR and TERM: 1st year & 1st term**
SUBJECT NAME	**HTML**	
NAME OF THE ASSESSMENT	**Reflective lab journal-6**	
DATE OF SUBMISSION	**2.11.25**	

WEEK-VI

Write a program for sizing

```
<html>
    <head>
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/nootstrap.min.css"
        rel="stylesheet"
        </head>
    <body>
        <div class="bg-primary w-25 p-3 text-white">width 25%</div>
        <div class="bg-success w-25 p-3 text-white">width 50%</div>
        <div class="bg-danger w-75 p-3 text-white">width 75%</div>
        <div class="bg-warning w-100 p-3 text-white">width 100%</div>
        <div class="bg-secondary h-50 p-3 text-white">Height 100%</div>
    </body>
</html>
```

Output:

Explanation:

width → controls how wide an element is.

height → controls how tall an element is.

Absolute units (fixed size)

- px → pixels (most common)
- cm, mm, in → print units (rarely used)

- `%` → relative to parent element
- `em` → relative to current element's font size
- `rem` → relative to root (`html`) font size
- `vw` → 1% of viewport width
- `vh` → 1% of viewport height

Controls how total size is calculated (content + padding + border).

Two main modes:

- content-box → default, size = content only.
- border-box → includes padding + border inside width/height.

2. Write a program for icons:

```
<html>
    <head>
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/nootstrap.min.css"
        rel="stylesheet"
```

```
        integrity="sha.384-
EVSTQN3/azprGIAnm3QDgpJLIm9Nao0Yz1ztcQTwFspdy3yD65Vohhpuu
        COmIASjC"crossorigin="annonymous">
        <link rel="stylesheet"
        href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.10.5/font/bootstrap-icons.css">
        <style type="text/css">
            .bi{
                font size:60px;
                color:blue;
            }
            </style>
        </head>
        <body>
            <i class="bi bi-alarm"></i>
            <i class="bi bi-heart-fill"></i>
            <i class="bi bi-cloud-download"> </i>
            <i class="bi bi-cart-fill"></i>
        </body>
        </html>
```

Output:

Explanation:

Icons are small graphical symbols that represent actions, files, or functions (like a trash bin for delete, or a magnifier for search).

There are **three main methods** to use icons on a webpage:

A. Icon fonts (e.g., Font Awesome, Bootstrap Icons)

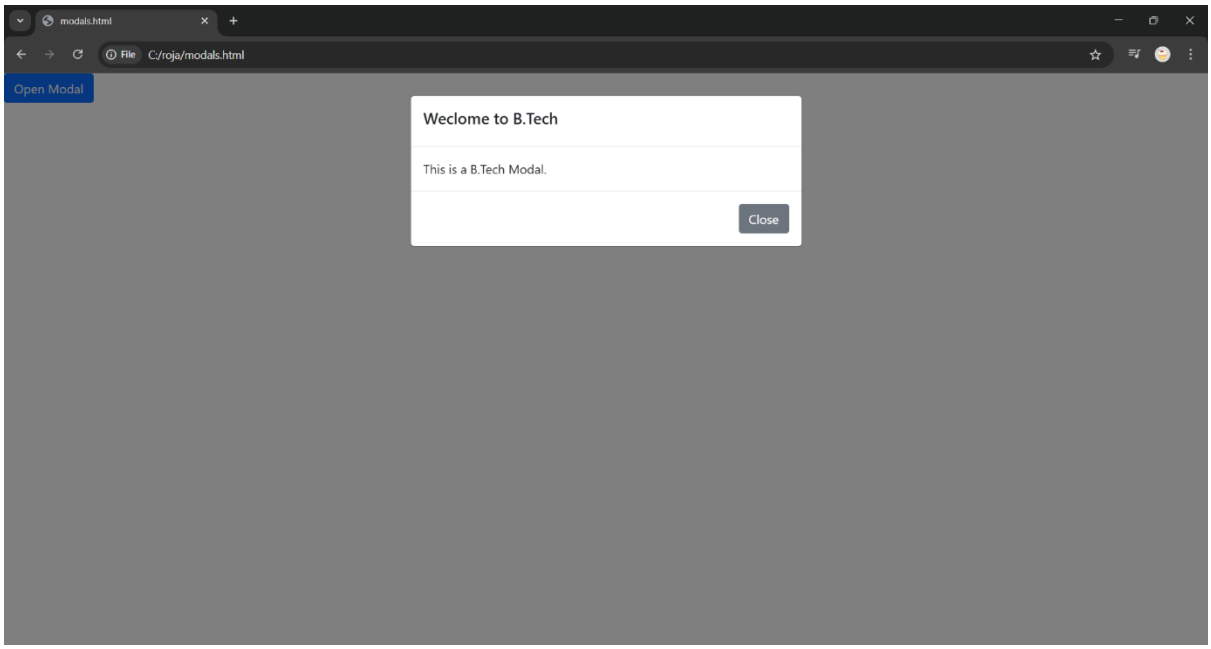
- Icons are included as fonts — scalable, easy to style with CSS.
-
- They help make websites more user-friendly and visually appealing. Icons can be added using **icon libraries** (like Font Awesome or Bootstrap Icons), **SVG files**, or **image files (PNG, JPG, etc.)**.

Using icons properly improves navigation, saves space, and enhances accessibility when combined with aria-label attributes for screen readers.

3. write a program on modals:

```
html>
    <head>
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
        <script
         src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.b
undle.min.js"></script>
    </head>
<body>
    <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-
bs-target="#myModal">
    Open Modal
    </button>
      <div class="modal fade" id="myModal">
      <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
            <h5>Weclome to B.Tech</h5>
        </div>
        <div class="modal-body">
            This is a B.Tech Modal.
        </div>
        <div class="modal-footer">
            <button class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>
        </div>
      </div>
      </div>
</body>
</html>
```

Output:



Explanation:

A **modal** is a pop-up window that appears on top of a webpage to display important information, forms, or alerts without leaving the current page. It's commonly used for login forms, confirmations, or messages. A modal is usually built using **HTML**, **CSS**, and **JavaScript** — HTML defines the structure (the box and its content), CSS styles it and positions it in the center of the screen, and JavaScript controls when it opens or closes.

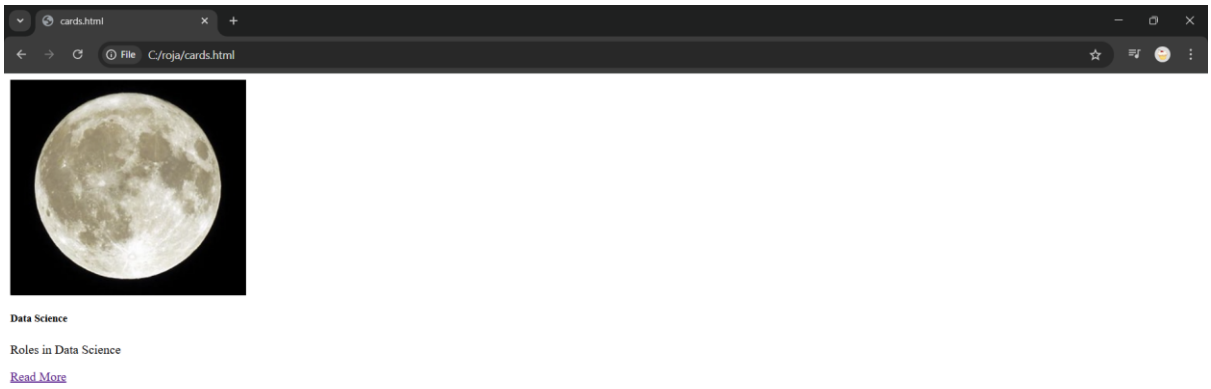
For example, clicking a button might trigger a script that changes the modal's display property from none to block, making it visible. Many frameworks like **Bootstrap** or **Tailwind CSS** provide built-in modal components that are easy to customize. Modals improve user experience by keeping users on the same page while interacting with important content.

4. Write a program on cards:

```
<html>
    <head>
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.3/dist/css/bootstrap.min.css"
        rel="stylesheet">
        <script
```

```
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle
.min.js"></script>
    </head>
    <body>
        <div class="card"style="width:300px;">
        <img src="C:\Users\roja\OneDrive\Desktop\web devolopement\moon.jpg">
        <div class="card-body">
            <h5>Data Science</h5>
            <p>Roles in Data Science</p>
            <a href="#" class="btn btn-primary">Read More</a>
            </div>
        </div>
    </body>
</html>
```

Output:



Explanation:

A **card** in web development is a flexible, content container used to display related information in a clean and organized way — such as product details, user profiles, or blog previews. A card typically includes elements like an image, title, description, and buttons, all grouped together inside a styled box. It is created using **HTML** for structure and **CSS** for styling such as background color, shadows, borders, and spacing.

Cards help create modern, visually appealing layouts and can be easily made responsive so they adjust nicely to different screen sizes. Many frameworks like **Bootstrap** and **Material UI** include pre-designed card components that save development time. Developers can also

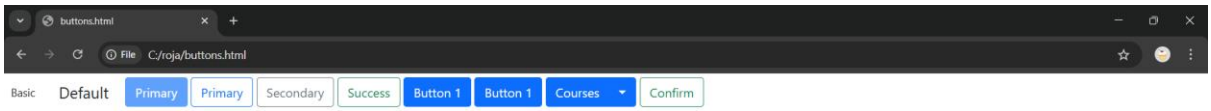
use **Flexbox** or **CSS Grid** to align multiple cards in rows or columns for a neat, professional layout.

5. write a program on buttons:

```
<html>
    <head>
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
        rel="stylesheet"
        <link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.1/font/bootstrap-
icons.css">
        <script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bindle
.min.js"></script>
    </body>
    </head>
    <body>
        <button type="button" class="btn btn-sm">Basic</button>
        <button type="button" class="btn btn-default btn-lg">Default</button>
        <button type="button" class="btn btn-primary btn-xs
disabled">Primary</button>
        <button type="button" class="btn btn-outline-primary">Primary</button>


        <button type="button" class="btn btn-outline-
secondary">Secondary</button>
        <button type="button" class="btn btn-outline-success">Success</button>
        <button type="button" class="btn btn-primary btn-block ">Button
1</button>

        <button type="button" class="btn btn-primary btn-block ">Button
1</button>

        <div class="btn-group">
<button type="button" class="btn btn-primary">Courses</button>
<button type="button" class="btn btn-primary dropdown-toggle"

data-bs-toggle="dropdown"></button>
       <ul class="dropdown-menu">
       <li><a class="dropdown-item" href="#">Python</a></li>
       <li><a class="dropdown-item" href="#">Java</a></li>
       </ul>
       </div>
        <button type="button" class="btn btn-outline-success">
```

```
        <i class="bi bi-check-circle"></i> Confirm
         </button>
</body>
</html>
```

Output:



Explanation:

A **button** in web development is an interactive element that users click to perform an action, such as submitting a form, opening a modal, or navigating to another page. Buttons are created using the <button> HTML tag or sometimes <a> tags styled to look like buttons. They can be styled using **CSS** to control their color, size, shape, borders, and hover effects, making them visually appealing and easy to identify.

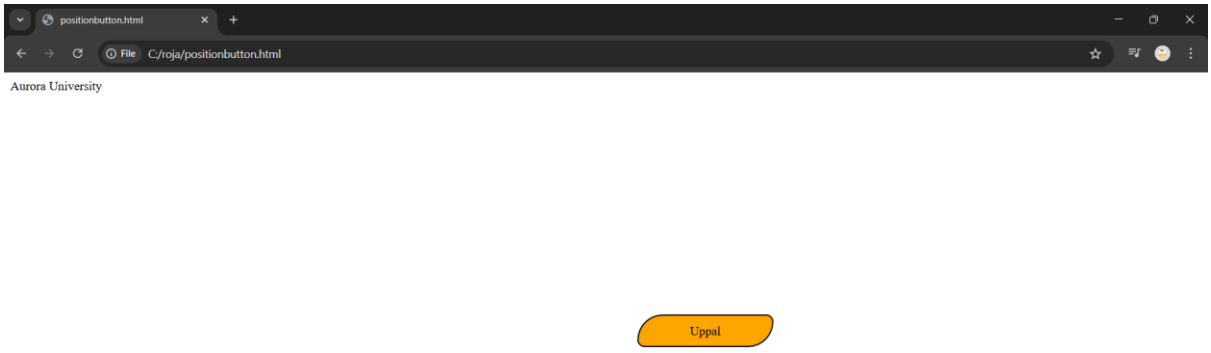
Buttons can also include **icons**, **text**, or both to clearly indicate their purpose — for example, a trash icon for delete or a paper plane for send. Developers often use **JavaScript** to make buttons functional, like showing alerts or triggering animations when clicked. Many frameworks such as **Bootstrap**, **Tailwind CSS**, or **Material Design** provide pre-styled button classes for consistency and faster development.

6. Write a program on position and buttons:

```
<html>
<head>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"
```

```
integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOm
LASjC" crossorigin="anonymous">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.1/font/bootstrap-
icons.css">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min
.js"></script>
</head>
<body>
<div class="position-relative" style="height: 300px;">
<div class="rounded-pill position-absolute top-50 start-50 translate-middle
bg-primary text-white p-3">
Aurora University
</div>
</div>
<div style="text-align:center;margin-left:800px;width:150px; border: 2px solid
black;
background-color:orange; border-radius: 35px 10px; padding: 10px;">
Uppal
</div>
</body>
</html>
```

Output:



Explanation:

In web development, **positioning** determines where an element appears on a webpage. CSS provides different position types like **static** (default), **relative** (moves relative to its normal spot), **absolute** (placed relative to the nearest positioned ancestor), **fixed** (stays in place when scrolling), and **sticky** (switches between relative and fixed). **Buttons** are interactive elements used for actions like submitting forms or triggering events. They are created using the <button> tag in HTML and styled with CSS for color, size, borders, padding, and hover effects. Buttons can also be **positioned** anywhere on the page using CSS positioning, making them flexible for different layouts and user interactions. Combining button styling with positioning allows developers to create **visually appealing and functional UI elements** that improve user experience.