Tutorial 4: JavaScript

CS 104

Spring, 2024-25

TA: Abhi Jain

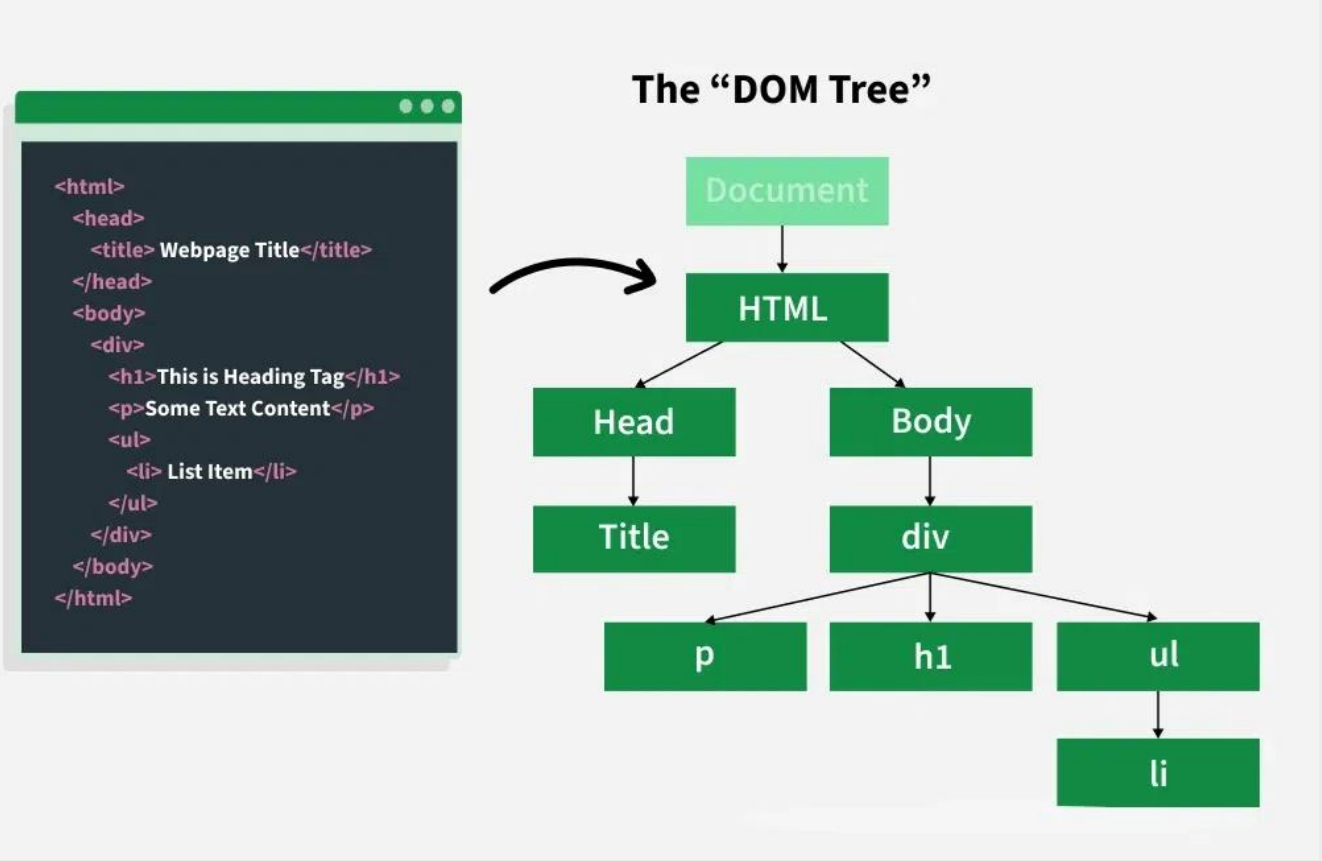
Credits: TA-2023-2: Guramrit Singh

Topics

- JavaScript (Basics)
- Fun Activities

JavaScript (Basics)

- Functions
- Iterables
- Events


The “DOM Tree”
<html>
<head>
<title> Webpage Title</title>
</head>
<body>

<h1>This is Heading Tag</h1>
Some Text Content

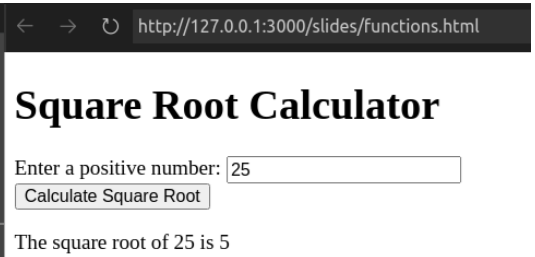
 List Item

</body>
</html>
Document
HTML
Head
Body
Title
div
p
h1
ul
li

Functions

slides > functions.html > html > body > br

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Function</title>
</head>
<body>
    <h1>Square Root Calculator</h1>

    <label for="number">Enter a positive number: </label>
    <input type="number" name="number" id="number">
    <br>
    <button onclick="calculateSquareRoot()">Calculate Square Root</button>
    <p id="result"></p>

    <script>
        function calculateSquareRoot() {
            var number = document.getElementById("number").value;
            var squareRoot = Math.sqrt(number);
            document.getElementById("result").innerHTML =
            "The square root of " + number + " is " + squareRoot;
        }
    </script>
</body>
</html>
```



- The getElementById() method is the DOM method that returns the element with the specified ID.
- The innerHTML property is used to change the content of an HTML element.
- The Math.sqrt() method is used to calculate the square root of a number.
- The value property is used to get the value of an input element.
- The document object represents the HTML document that is displayed in the browser.

- The syntax of function is very similar to C++, except that we don't need to specify the return type of the function. The function keyword is used to define a function.
- the onclick attribute is used to call the function when the button is clicked..

External Javascript

slides > functions.html > html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-s
    <title>Function</title>

</head>
<body>
    <h1>Square Root Calculator</h1>

    <label for="number">Enter a positive number: </label>
    <input type="number" name="number" id="number">
    <br>
    <button onclick="calculateSquareRoot()">Calculate Square Root</button>
    <p id="result"></p>

    <script src="functions.js"></script>
</body>
</html>
```

slides > functions.js > calculateSquareRoot

```js
function calculateSquareRoot() {
    var number = document.getElementById("number").value;
    var squareRoot = Math.sqrt(number);
    document.getElementById("result").innerHTML =
    "The square root of " + number + " is " + squareRoot;
}
```

- We do not use the link tag to link to external javascript files. We use the self-closing script tag instead.
- It may be placed in the head or the body tag. The src attribute is used to specify the location of the external javascript file.

Arrays and Loops

slides > objects.html > html > body > script

```
12      <table>
13          <tr>
14              <th>Name</th>
15              <th>Grade</th>
16          </tr>
17          <tr>
18              <th class="name">John</th>
19              <th class="grade">A</th>
20          </tr>
21          <tr>
22              <th class="name">Jane</th>
23              <th class="grade">B</th>
24          </tr>
25          <tr>
26              <th class="name">Joe</th>
27              <th class="grade">C</th>
28          </tr>
29          <tr>
30              <th class="name">Jack</th>
31              <th class="grade">D</th>
32          </tr>
33      </table>
34
35      <p id="demo"></p>
36
37      <script>
38          let names = document.getElementsByClassName("name");
39          let students = "";
40          for(let i=0; i<names.length-1; i++){
41              students += names[i].innerHTML + ", ";
42          }
43          students += names[names.length-1].innerHTML;
44          students = "The students are: " + students + ".";
45          document.getElementById("demo").innerHTML = students;
46      </script>
47  </body>
48  </html>
```

http://127.0.0.1:3000/slides/objects.html

Arrays and For Loops

Table of Students and Grade

Name	Grade
John	A
Jane	B
Joe	C
Jack	D

The students are: John, Jane, Joe, Jack.

- getElementsByClassName() returns an array of objects with the same class name.
- The for loop is used to iterate on an array of objects.

UI Events

```
    <style>
        #box1, #box2, #box3{
            width: 50px;
            height: 50px;
            background-color: blue;
            position: relative;
            left: 0;
        }
    </style>
</head>
<body>
    <h1>UI Events</h1>
    <div id="box1"></div>
    <br> <br>
    <div id="box2"></div>
    <br> <br>
    <div id="box3"></div>
    <br> <br>
    <p id="demo"></p>
```

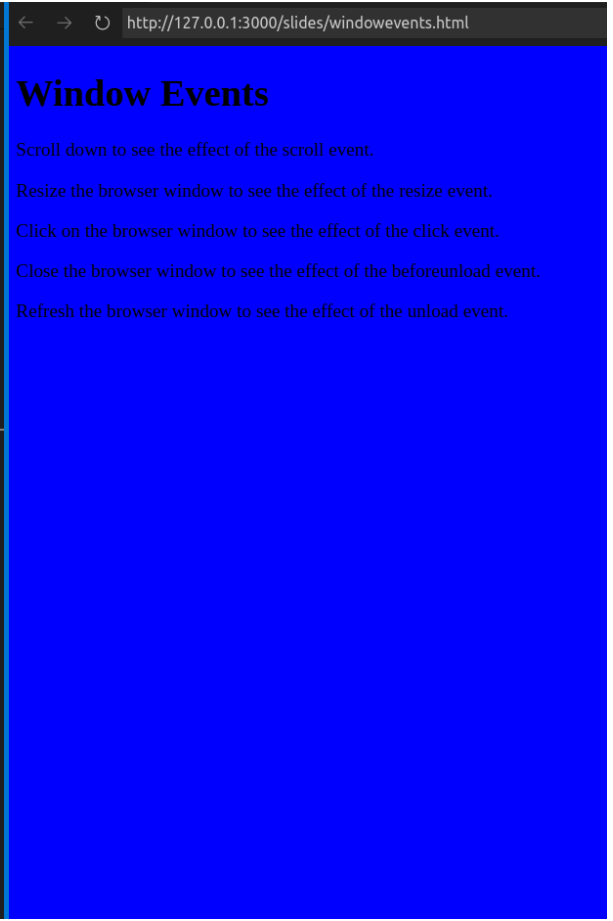
- Demo shown in tutorial session.

```
<script>
    let box1 = document.getElementById("box1");
    let box2 = document.getElementById("box2");
    let box3 = document.getElementById("box3");
    box1.addEventListener("mouseover", function(){
        box1.style.backgroundColor = "red";
    });

    box1.addEventListener("mouseout", function(){
        box1.style.backgroundColor = "blue";
    });

    box2.addEventListener("mousedown", function(){
        box2.style.backgroundColor = "green";
    });

    box2.addEventListener("mouseup", function(){
        box2.style.backgroundColor = "blue";
    });

    box3.addEventListener("dblclick", function(){
        box3.style.backgroundColor = "yellow";
    });

    box3.addEventListener("click", function(){
        box3.style.backgroundColor = "orange";
    });

    document.addEventListener("keydown", function(event){
        document.getElementById("demo").innerHTML = "You pressed the " + event.key + " key.";
    });

    document.addEventListener("keyup", function(event){
        document.getElementById("demo").innerHTML = "You released the " + event.key + " key.";
    })
</script>
```

Window Events

slides > windowevents.html > html > body > script

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial
    <title>Document</title>
    <!-- We will discuss event handling for window/document ev

</head>
<body>

    <h1>Window Events</h1>
    <p>Scroll down to see the effect of the scroll event.</p>
    <p>Resize the browser window to see the effect of the resi
    <p>Click on the browser window to see the effect of the cl
    <p>Close the browser window to see the effect of the befor
    <p>Refresh the browser window to see the effect of the unl

    <script>
        window.addEventListener("load", function(){
            alert("The page has loaded.");
        });

        window.addEventListener("scroll", function(){
            document.body.style.backgroundColor = "red";
        });

        window.addEventListener("resize", function(){
            document.body.style.backgroundColor = "blue";
        });

        window.addEventListener("click", function(){
            document.body.style.backgroundColor = "green";
        });
    </script>

</body>
</html>
```



Window events occur on the global window object and help track user interactions with the browser window.

Common Window Events:

- **- Load –** Fires when the entire page has fully loaded.
- **- Scroll –** Fires when the user scrolls the page.
- **- Resize –** Fires when the browser window is resized.
- **- Click –** Fires when the user clicks anywhere on the window.

Form Events:

- Form events will be discussed in the Fun Activities section.

DOM Manipulation

```html
<body>
    <h1 id="title">DOM Manipulation Example</h1>
    <button id="add">Add Paragraph</button>
    <button id="remove">Remove Paragraph</button>
    <button id="change">Change Title</button>
    <button id="highlight">Highlight Title</button>

    <div id="container"></div>

    <script>
        document.getElementById("add").addEventListener("click", function () {
            let newPara = document.createElement("p");
            newPara.textContent = "This is a dynamically added paragraph.";
            document.getElementById("container").appendChild(newPara);
        });

        document.getElementById("remove").addEventListener("click", function () {
            let container = document.getElementById("container");
            if (container.lastChild) {
                container.removeChild(container.lastChild);
            }
        });

        document.getElementById("change").addEventListener("click", function () {
            document.getElementById("title").textContent = "Title Changed!";
        });

        document.getElementById("highlight").addEventListener("click", function () {
            document.getElementById("title").classList.toggle("highlight");
        });
    </script>
</body>
```

`document.createElement(tagName)`

- Creates a new HTML element.

`appendChild(element)`

- Appends a child element to a parent.

`removeChild(element)`

- Removes a child element from its parent.

`element.remove()` **(Modern approach)**

- Removes the selected element itself.

`setAttribute(name, value)`

- Sets an attribute to an element.

`classList.add()` **and** `classList.remove()`

- Adds or removes CSS classes dynamically.

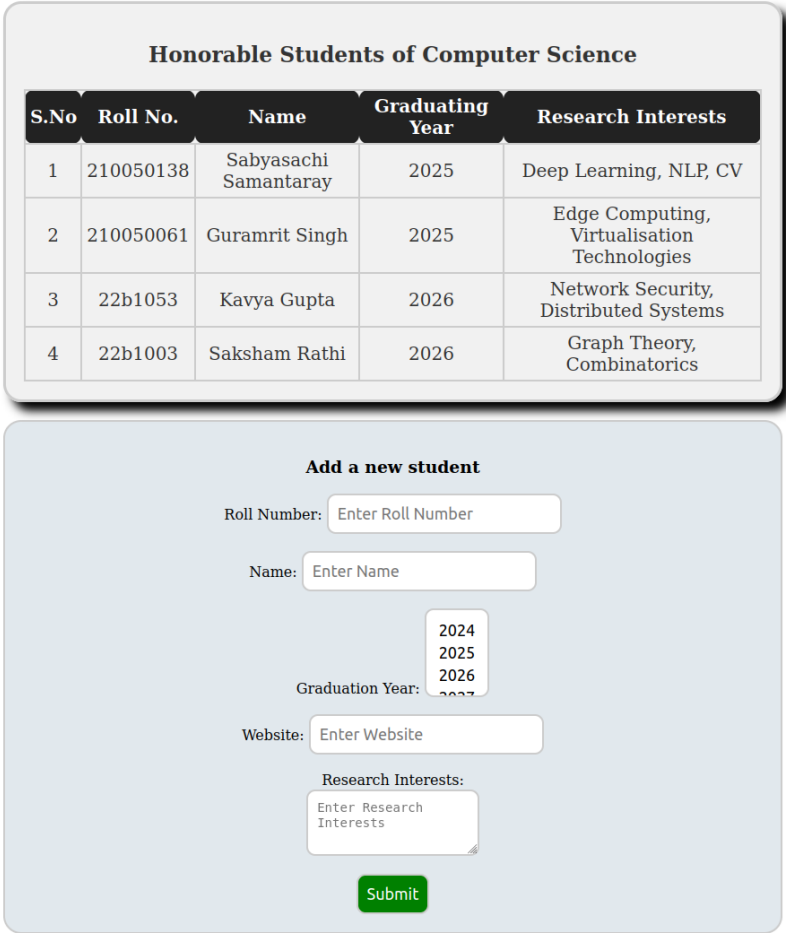
Fun Activities

- Form Validation
- Submission Handling
- Your Own Carousel

Activity 1 - Form Validation

- In this activity, we will validate the entries in the form to be in a specific format(or in other words, obey a pattern or a regular expression)
- When the form is submitted, the verifier is invoked which checks for correctness of the entries.
- Remove all the required attributes we added to make the section compulsory. We instead do this verification in the submit_verifier itself.
- The following functions may be of help:
  - .reset() for form resetting
  - alert() for popping up an alert window.

Solution: Provided as index_validate.html



Honorable Students of Computer Science

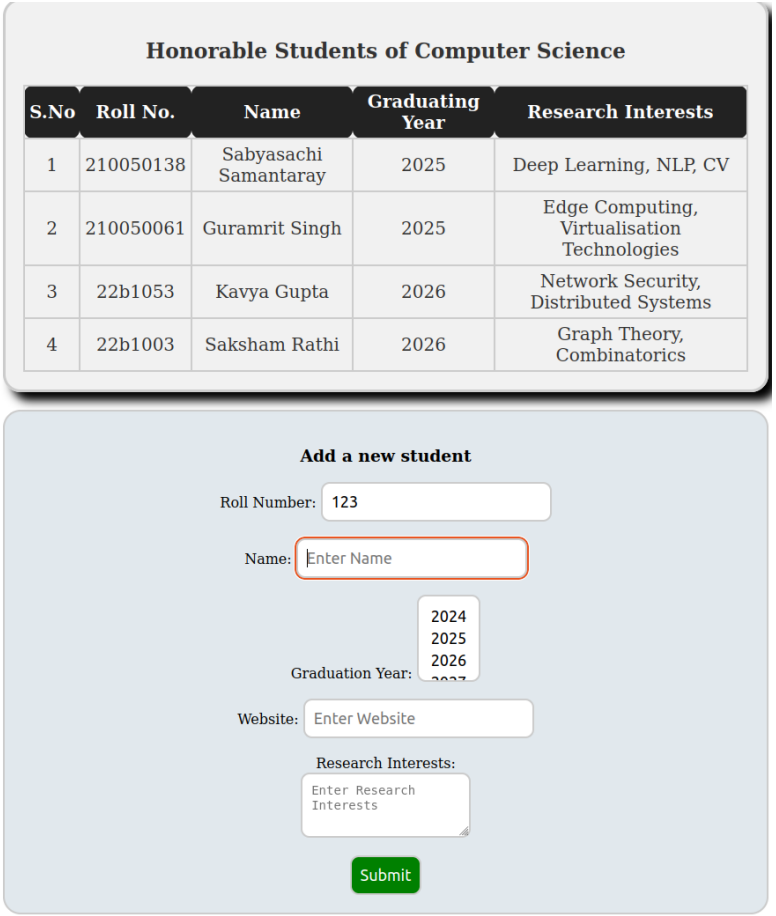
S.No	Roll No.	Name	Graduating Year	Research Interests
1	210050138	Sabyasachi Samantaray	2025	Deep Learning, NLP, CV
2	210050061	Guramrit Singh	2025	Edge Computing, Virtualisation Technologies
3	22b1053	Kavya Gupta	2026	Network Security, Distributed Systems
4	22b1003	Saksham Rathi	2026	Graph Theory, Combinatorics

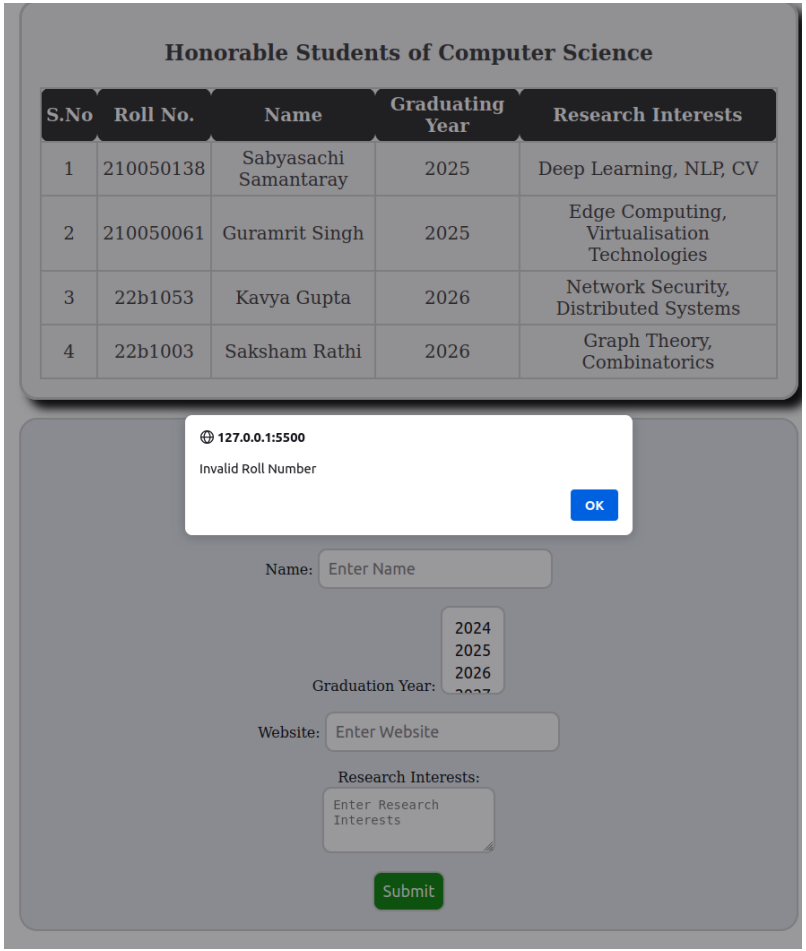
Activity 1 - Hints

- We shall follow the following simple checking rules:
  - Roll Number should be of Form YYBXXXX - where Y and X are digits
  - Name should contain letters and spaces. Ensure no two consecutive spaces, and no space at the end.
  - Research Interests and GradYear fields should not be empty.

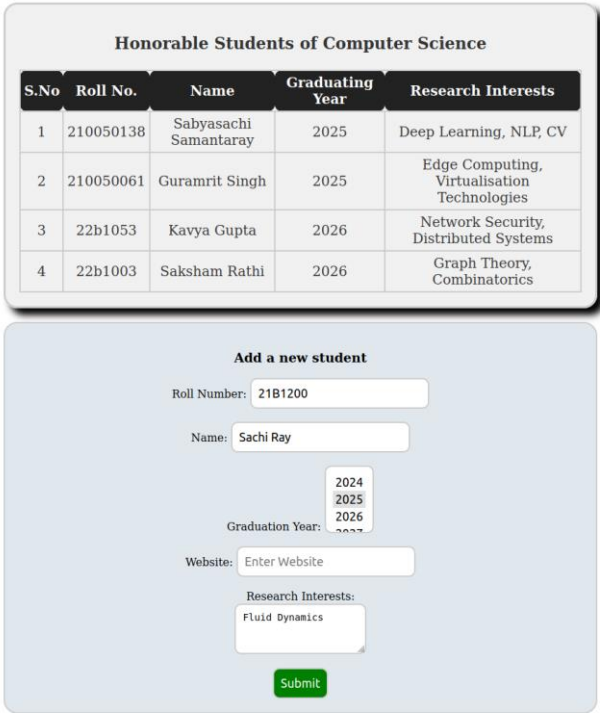
- You would have to write your own regular expression to verify/test the input. Refer here for ECMAScript(JavaScript) RegExp References.
- To build and test Regular Expressions, check out this.

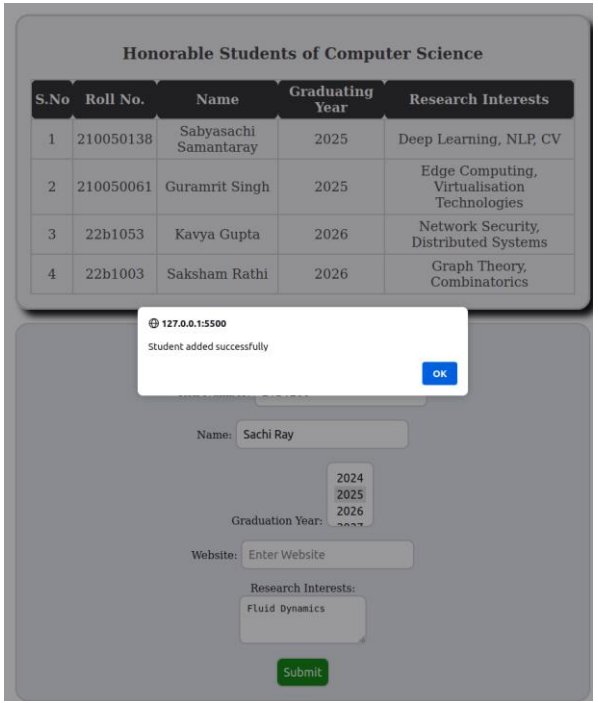
Activity 1 - Demo

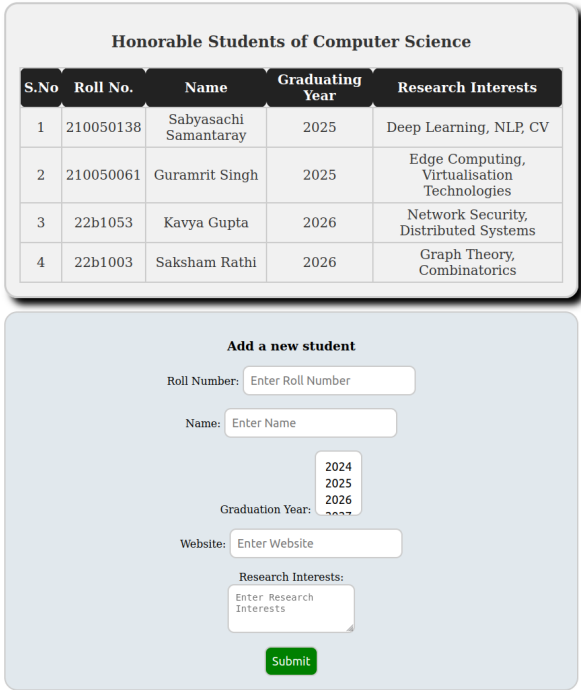




Activity 1 - Demo





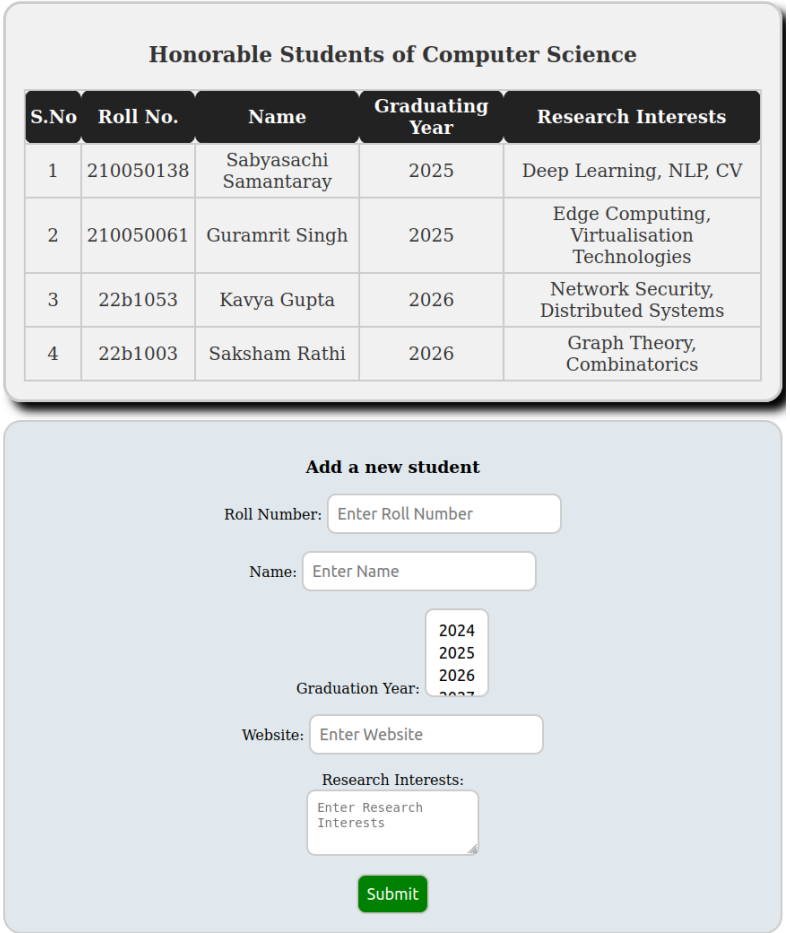


1. Alert box shows that all information was correct
2. Resets the form

Activity 2 - Submission Handling

- In this activity, we will be handling the submission of the add-student form created in the activity last week.
- When the form is submitted, append the student information to the end of the table and reset the form after submitting.
- Hints:
  - Disable the default form submission behavior.
  - The website input field is not mandatory.
  - Backtick character ` creates template literals which allow for multi-line strings and string interpolation.

Solution: Provided as index_submit.html



Honorable Students of Computer Science

S.No	Roll No.	Name	Graduating Year	Research Interests
1	210050138	Sabyasachi Samantaray	2025	Deep Learning, NLP, CV
2	210050061	Guramrit Singh	2025	Edge Computing, Virtualisation Technologies
3	22b1053	Kavya Gupta	2026	Network Security, Distributed Systems
4	22b1003	Saksham Rathi	2026	Graph Theory, Combinatorics

Add a new student

Roll Number: Enter Roll Number

Name: Enter Name

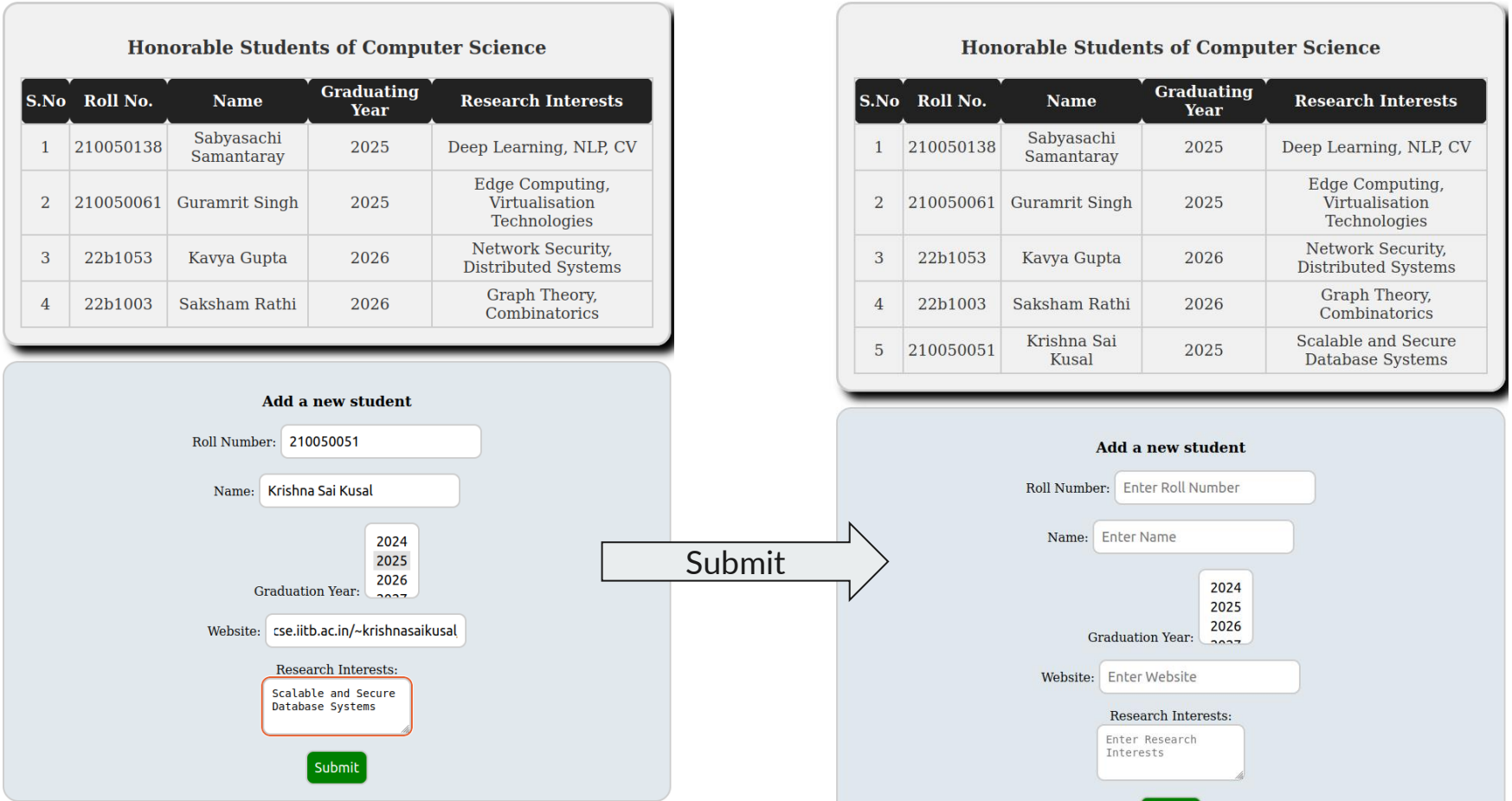
Graduation Year: 2024 2025 2026

Website: Enter Website

Research Interests: Enter Research Interests

Submit

Activity 2 - Demo

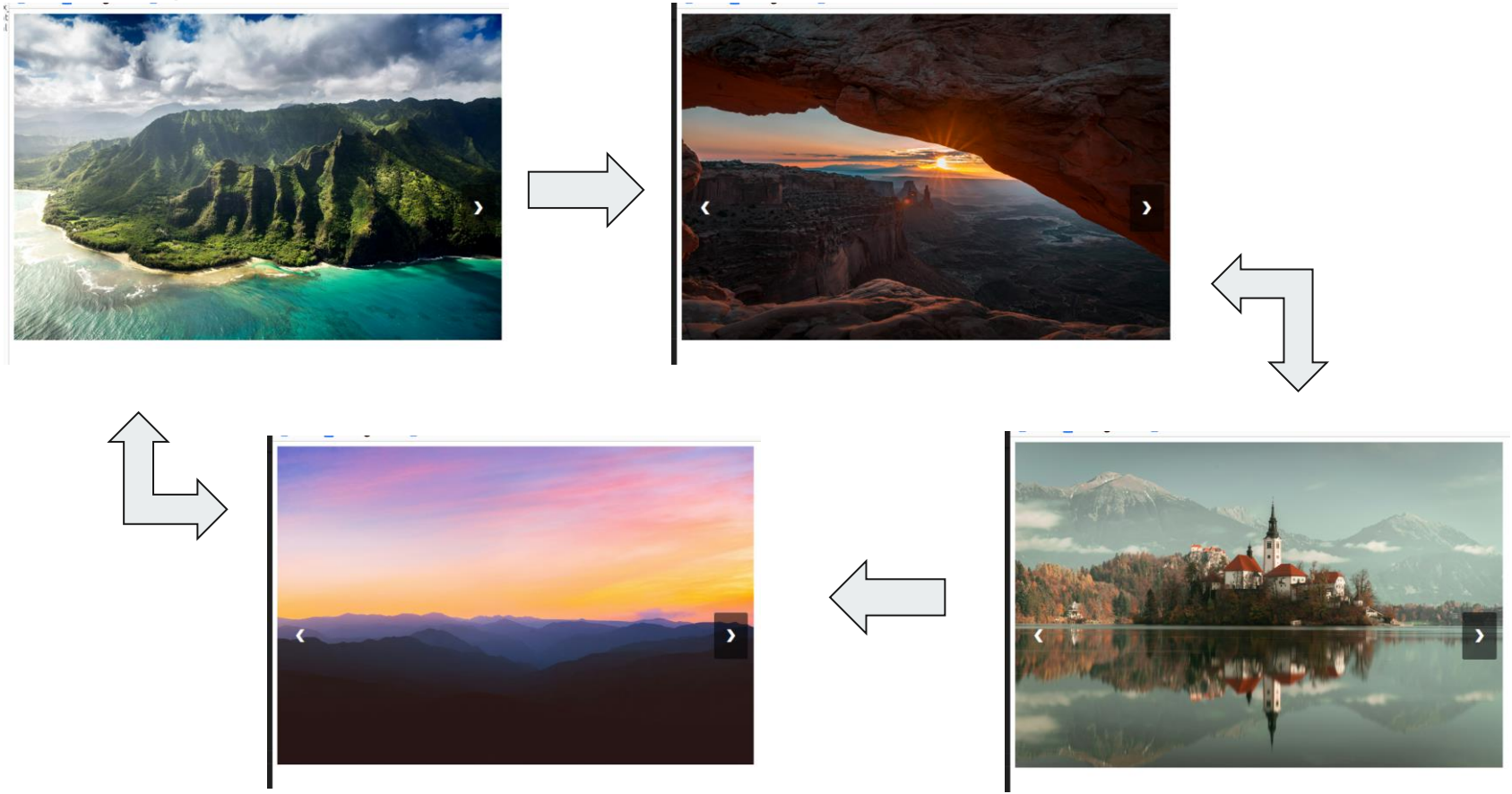


Activity 3 - A Simple Carousel

- Description:
  - The carousel will have 4 images and 2 buttons to navigate through the images.
  - The images will be displayed one at a time.
  - The buttons will be used to navigate to the next and previous images.
  - You shall ensure that there is a cyclic navigation i.e.
    - Next on the last image displays the first image
    - Previous on the first image displays the last image

- In this activity, we will build a simple carousel/slideshow.
- You can use the images provided in the resources section or use your own images.
- You can use the Unicode Decimal Code for arrows to display the next and previous buttons.
- Check out CSS Transitions.
- HTML Code for Left Angle is ❮ and for Right Angle is ❯
- Solution: Provided as carousel.html

Activity 3 - Demo



Thank You !!!