

Front End Engineering-II

Project Report

Semester-IV (Batch-2022)

DOUBLE VERTICAL SLIDER



Supervised By:

Dr. Raveesh Samkaria

Submitted By:

**Saksham Vashisht
2210990768(G-12)**

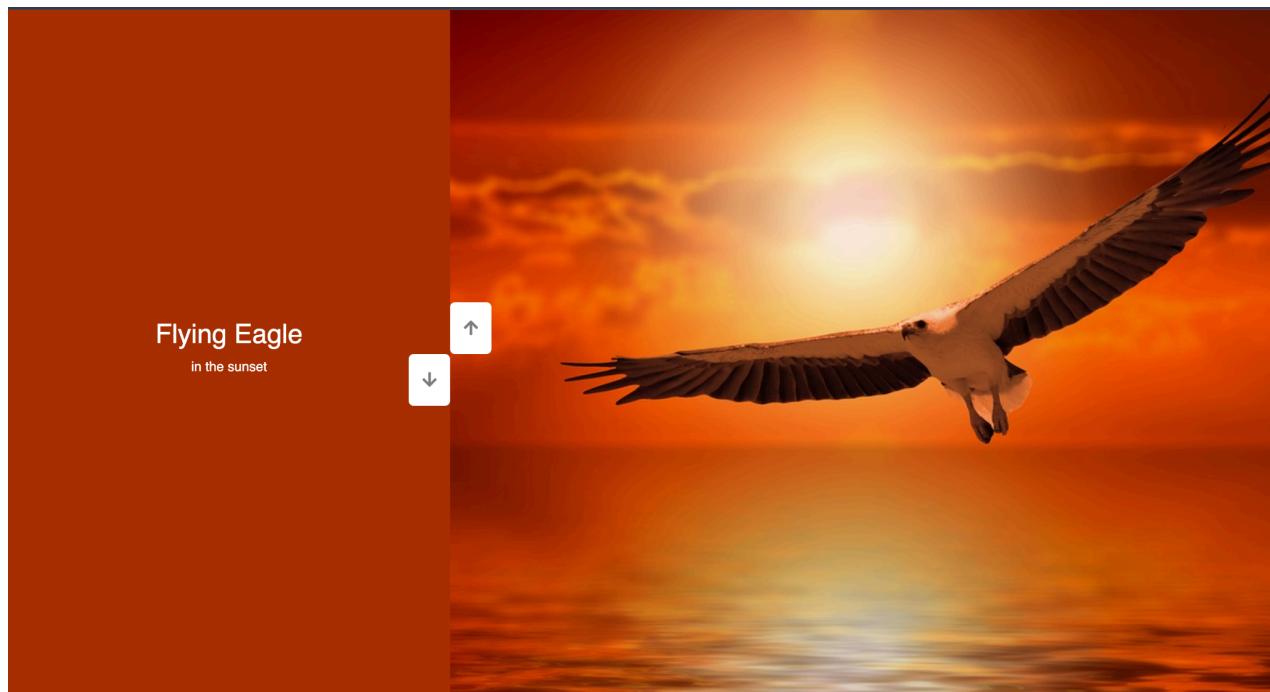
**Department of Computer Science and Engineering Chitkara
University Institute of Engineering & Technology
Chitkara University, Punjab**

Abstract

In contemporary user interface design, sliders serve as fundamental components for interactive controls. This abstract introduces the concept of a Double Vertical Slider (DVS), a novel UI element that offers enhanced versatility and functionality compared to traditional single-axis sliders. The DVS comprises two vertically oriented sliders arranged in parallel, each independently adjustable along its respective axis.

The DVS facilitates a wide range of applications across various domains including user experience design, data visualization, gaming interfaces, and more. Its unique design enables intuitive control over two distinct parameters simultaneously, thereby expanding the scope of user interaction possibilities.

This abstract explores the key features and potential uses of the DVS, highlighting its advantages over conventional single-axis sliders. Additionally, it discusses implementation considerations, usability considerations, and potential future directions for research and development. **Keywords:** Double Vertical Slider, User Interface Design, Interactive Controls, User Experience, Data Visualization, Gaming Interfaces.



INDEX

S.No.	Title	Page Number(s)
1	Introduction	4
2	Problem Statement	5
3	Software Requirements	5
4	Proposed Design	6-9
5	Results	10-13
6	References	14

1. Introduction:

Enter the Double Vertical Slider (DVS). This innovative user interface element redefines the way users interact with controls by introducing a dual-axis manipulation system. With the DVS, users can effortlessly adjust two parameters independently, unlocking a new realm of possibilities for customization and control.

In this project, we aim to explore the potential of the Double Vertical Slider across a myriad of applications, ranging from user experience design to data visualization and gaming interfaces. Through thorough research, design iterations, and usability testing, we endeavor to showcase the versatility, usability, and practicality of the DVS in enhancing user interaction.

1.1 Background:

The inspiration for the Double Vertical Slider arises from the need to overcome these limitations and provide users with greater flexibility and control. Drawing from principles of human-computer interaction and ergonomic design, the DVS introduces a dual-axis manipulation system, enabling users to simultaneously adjust two parameters along independent axes. This innovative approach not only expands the scope of user interaction but also opens doors to new possibilities in customization and user engagement.

1.2 Objectives:

An objective for a double vertical slider could be to provide users with an intuitive and efficient way to adjust two separate parameters simultaneously, enhancing user experience and allowing for precise control over settings or values. This could involve ensuring smooth and responsive slider movement, clear visual feedback, and customization options to meet diverse user needs. Additionally, optimizing accessibility and compatibility across different devices and screen sizes would be important considerations.

1.3 Significance:

A double vertical slider typically refers to a user interface element where there are two sliders that can be moved vertically along a track. These sliders are often used to represent a range or interval, allowing users to select a value within that range. Overall, the significance of a double vertical slider lies in its ability to provide users with a flexible and intuitive means of selecting a range of values, enhancing the usability and effectiveness of interfaces that utilize them.

2. Problem Statement

Many user interfaces require the selection of a range of values, such as prices, time intervals, or other numeric ranges. Current UI elements often lack precision and user-friendliness in facilitating this selection process. Implementing a double vertical slider would address these shortcomings by providing users with a more intuitive and interactive means of selecting a range of values. However, designing and integrating such a slider into the interface requires careful consideration of usability, functionality, and visual design to ensure an optimal user experience. The problem statement aims to explore the challenges and opportunities in implementing a double vertical slider to enhance range selection in user interfaces.

3. Software Requirements

a) Integrated Development Environment (IDE):

- Visual Studio Code (VS Code) for code editing and project management.

b) Frontend Technologies:

- HTML: Markup language for structuring the web application.
- CSS: Styling language for enhancing the presentation and layout.
- JavaScript (JS): Programming language for implementing interactive features and logic.

c) User Interface (UI) Framework:

- Bootstrap 5: Frontend framework for building responsive and visually appealing user interfaces.

d) Version Control:

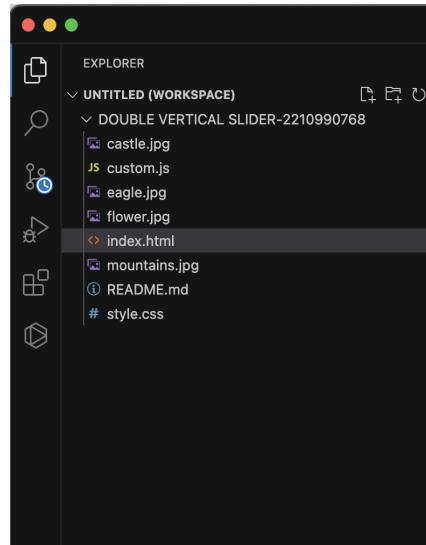
- Git: Distributed version control system for tracking changes in the project codebase.

4. Proposed Design

- **User Interface Design:** Utilize Bootstrap 5 for a responsive, visually appealing lay-out. Employ card-based design for intuitive organization.
- **Frontend Development:** Develop using HTML, CSS, and JavaScript. Utilize HTML5 semantics, CSS for styling, and JavaScript for dynamic UI updates.
- **Double Vertical Slider Logic:** The logic behind a double vertical slider revolves around enabling users to intuitively select a range of values within a defined range while ensuring a smooth and responsive interaction experience.
- **User Experience Optimization:** Focus on real-time feedback, interactive elements, and cross-browser compatibility. Ensure responsiveness for varied devices.
- **Testing and Quality Assurance:** Conduct comprehensive testing, including manual and unit tests. Ensure functionality and UI consistency.
- **Documentation and Deployment:** Provide detailed documentation. Deploy on web server with domain. Maintain and update documentation regularly
- **Integration of Libraries:** Utilize libraries like SweetAlert2 for user-friendly alerts and confetti for celebratory effects, enhancing user experience and engagement

4.1 File Structure

Ensuring proper file and folder structure to maintain consistent file paths and clean structure.



4.2 HTML Code Structure

These screenshots present the HTML code for our Double Vertical Slider project, revealing the layout and content of our web pages in a code format.

```
index.html # style.css custom.js
DOUBLE VERTICAL SLIDER-2210990768 > index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Double Vertical Slider</title>
9      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css" />
10     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
11          integrity="sha384-QWTKZyjpPEIISvJPFqjlY6HcNpPQKkxHmOjM0dLwYmJ8lXoYQ" crossorigin="anonymous">
12     <link rel="stylesheet" href="style.css">
13 </head>
14
15 <body>
16     <div class="slider-container position-relative overflow-hidden vw-100 vh-100">
17         <div class="left-slide h-100 position-absolute">
18             <div class="h-100 w-100 d-flex flex-column align-items-center justify-content-center text-white"
19                 style="background-color: #9a73a9;">
20                 <h1 class="fs-2">Nature Flower</h1>
21                 <p>all in pink</p>
22             </div>
23             <div class="h-100 w-100 d-flex flex-column align-items-center justify-content-center text-white"
24                 style="background-color: #2A86BA;">
25                 <h1 class="fs-2">Blue Sky</h1>
26                 <p>with it's mountains</p>
27             </div>
28             <div class="h-100 w-100 d-flex flex-column align-items-center justify-content-center text-white"
29                 style="background-color: #252E33;">
30                 <h1 class="fs-2">Lonely Castle</h1>
31                 <p>in the wilderness</p>
32             </div>
33             <div class="h-100 w-100 d-flex flex-column align-items-center justify-content-center text-white"
34                 style="background-color: #a82f02;">
35                 <h1 class="fs-2">Flying Eagle</h1>
36                 <p>in the sunset</p>
37             </div>
38         </div>
39     </div>
```

4.3 CSS Code Structure

This screenshot exhibits the CSS code for our Double Vertical Slider, illustrating the styling and design elements implemented across our web pages.

```
< index.html # style.css JS custom.js >
DOUBLE VERTICAL SLIDER-2210990768 > # style.css > ✎ *  
1  * {  
2      box-sizing: border-box;  
3      margin: 0;  
4      padding: 0;  
5  }  
6  
7  body {  
8      font-family: "Open Sans", sans-serif;  
9      height: 100vh;  
10 }  
11  
12 .left-slide {  
13     width: 35%;  
14     top: 0;  
15     left: 0;  
16     transition: transform 0.5s ease-in-out;  
17 }  
18  
19 .right-slide > div {  
20     background-repeat: no-repeat;  
21     background-size: cover;  
22     background-position: center center;  
23     height: 100%;  
24     width: 100%;  
25 }  
26  
27 .right-slide {  
28     left: 35%;  
29     width: 65%;  
30     transition: transform 0.5s ease-in-out;  
31 }  
32
```

```
32  
33 button:hover {  
34     color: #222;  
35 }  
36  
37 button:focus {  
38     outline: none;  
39 }  
40  
41 .slider-container .action-buttons button {  
42     position: absolute;  
43     left: 35%;  
44     top: 50%;  
45     z-index: 100;  
46 }  
47  
48 .slider-container .action-buttons .down-button {  
49     transform: translateX(-100%);  
50 }  
51  
52 .slider-container .action-buttons .up-button {  
53     transform: translateY(-100%);  
54 }  
55
```

4.4 Javascript Code Structure

This screenshot exhibits the JS code for our Double vertical Slider project, illustrating the different functions and events we trigger according to different scenarios.

```
DOUBLE VERTICAL SLIDER-2210990768 > JS custom.js > ...
1  const sliderContainer = document.querySelector('.slider-container')
2  const slideRight = document.querySelector('.right-slide')
3  const slideLeft = document.querySelector('.left-slide')
4  const upButton = document.querySelector('.up-button')
5  const downButton = document.querySelector('.down-button')
6  const slidesLength = slideRight.querySelectorAll('div').length
7
8
9  let activeSlideIndex = 0
10
11 slideLeft.style.top = `-${(slidesLength - 1) * 100}vh`
12
13 upButton.addEventListener('click', () => changeSlide('up'))
14 downButton.addEventListener('click', () => changeSlide('down'))
15
16
17
18 const changeSlide = (direction) => {
19   const sliderHeight = sliderContainer.clientHeight
20   if(direction === 'up'){
21     activeSlideIndex++
22     if(activeSlideIndex > slidesLength -1){
23       activeSlideIndex = 0
24     }
25   }else if(direction === 'down'){
26     activeSlideIndex--
27     if(activeSlideIndex < 0){
28       activeSlideIndex = slidesLength - 1
29     }
30   }
31   slideRight.style.transform = `translateY(-${activeSlideIndex * sliderHeight}px)`
32   slideLeft.style.transform = `translateY(${activeSlideIndex * sliderHeight}px)`
33 }
34
```

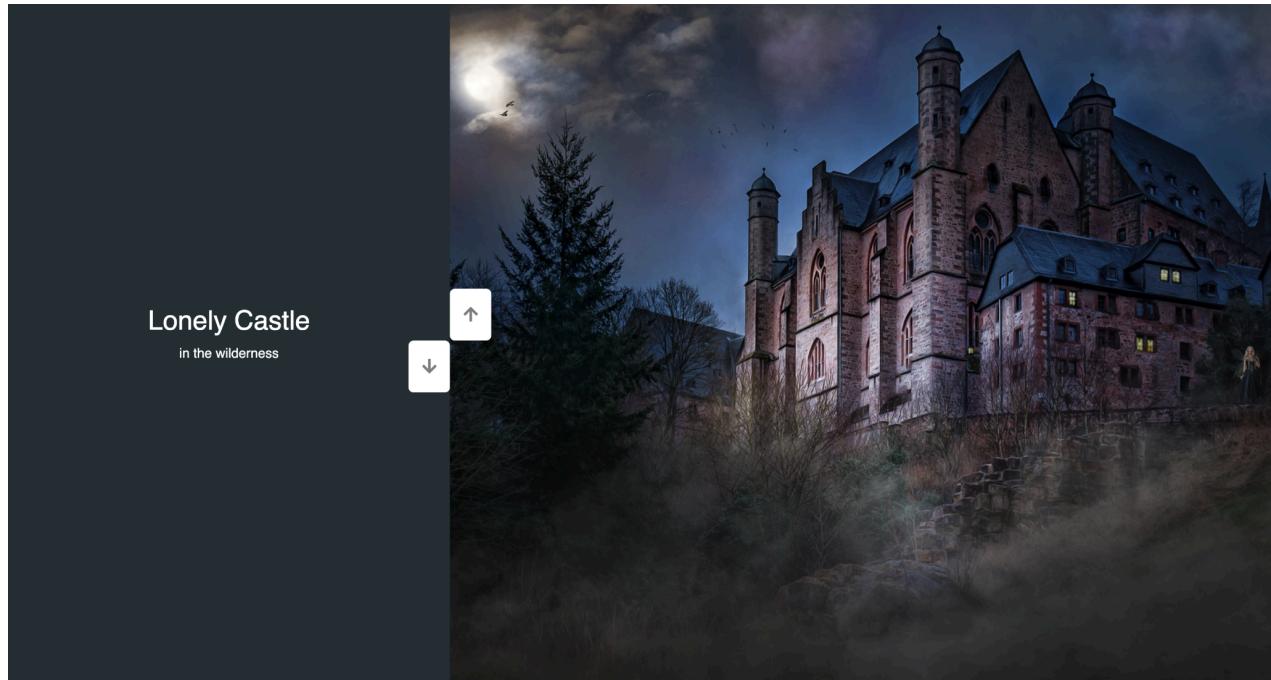
5. Results

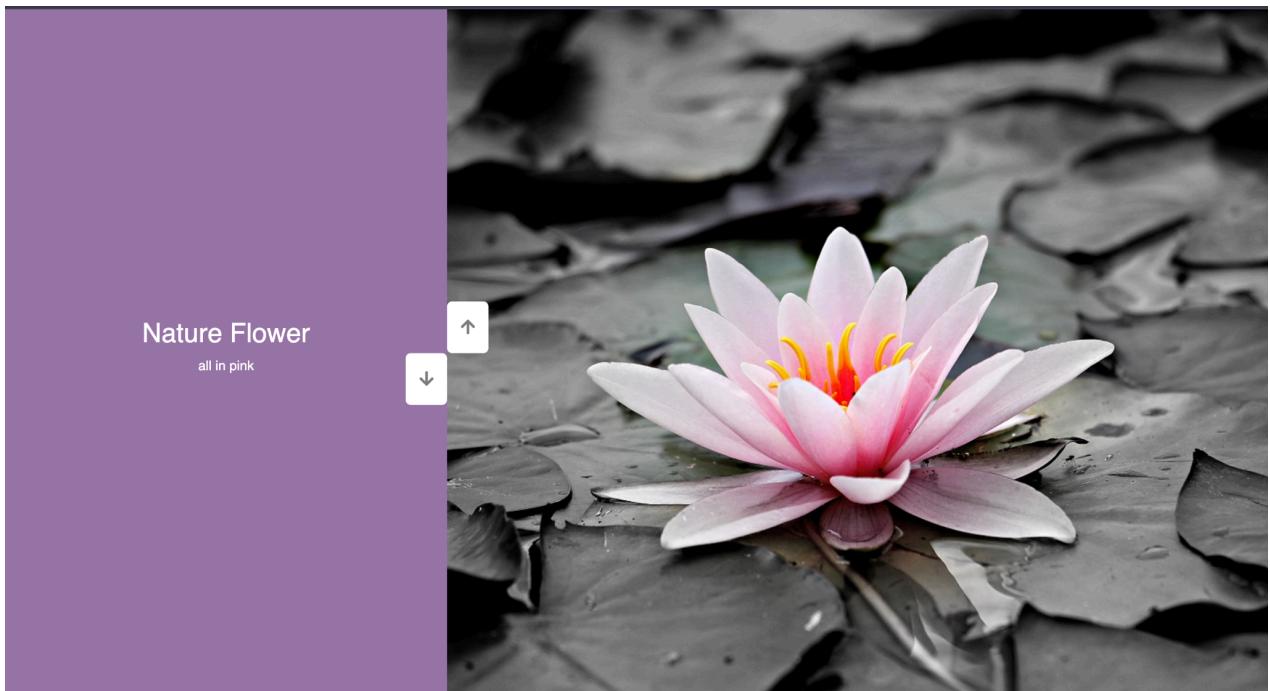
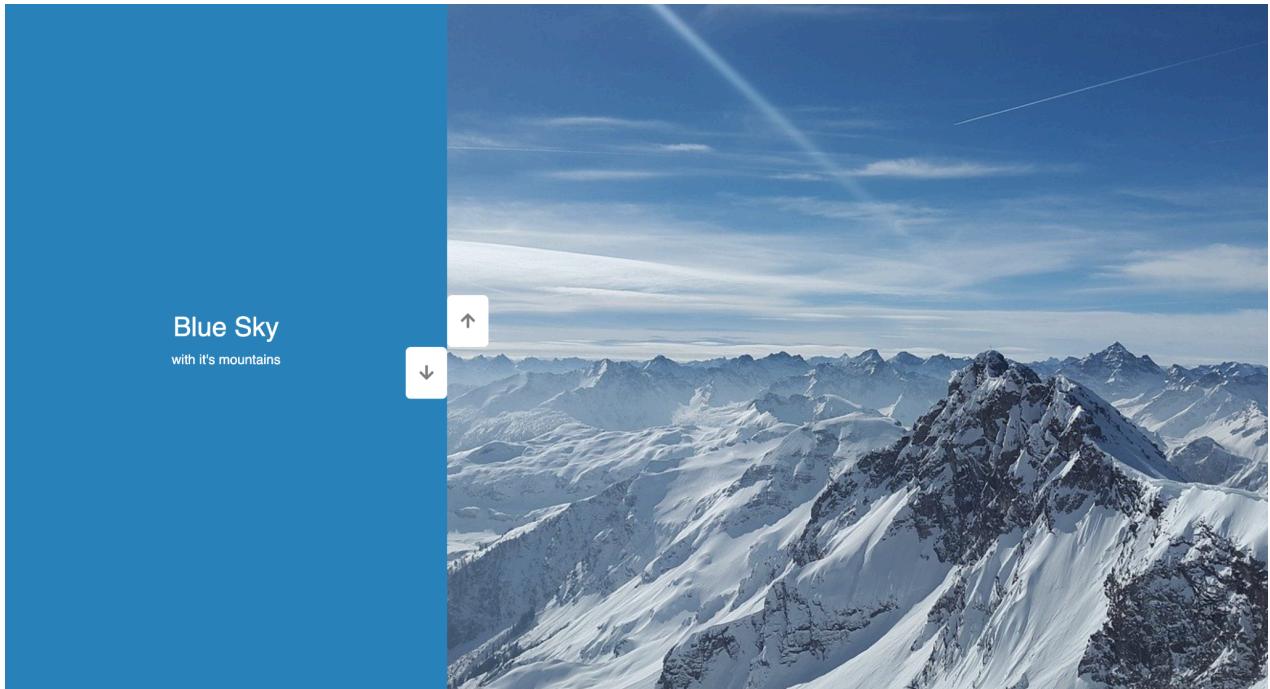
The application successfully fulfills its intended purpose implementing a Double Vertical Slider will result in a versatile and user-friendly interface component for selecting ranges of values, offering enhanced precision, flexibility, and ease of use.

GitHub Repository Link: <https://github.com/SakshamVashisht1/Double-Vertical-Slider.git>

GitHub Pages Link: <https://sakshamvashisht1.github.io/Double-Vertical-Slider/>

Project Screenshots for all scenarios:





Flying Eagle

in the sunset



Accuracy: The Double Vertical Slider logic, implemented in JavaScript, By incorporating these elements, a DVS can achieve high levels of accuracy, allowing users to precisely define the range of values they need for their tasks or selections.

- **User Experience:** The user interface design, leveraging Bootstrap 5, provides a seamless and visually appealing experience across different devices and screen sizes. Interactive elements and real-time feedback enhance usability and engagement.
- **Functionality:** A DVS can provide users with a powerful and intuitive tool for selecting and manipulating ranges of values, enhancing usability and effectiveness in various applications.
- **Integration of Libraries:** The integration of libraries such as SweetAlert2 for user-friendly alerts and confetti for celebratory effects adds an element of delight to the user experience, enhancing engagement and satisfaction.
- **Documentation and Deployment:** Comprehensive documentation guides users through setup, usage, and troubleshooting. Deployment on a web server with domain access ensures public availability, while regular updates maintain relevance and accuracy.

6. References

- **HTML, CSS, and JavaScript Documentation:**
 - **Mozilla Developer Network (MDN) - HTML:** <https://developer.mozilla.org/en-US/docs/Web/HTML>
 - **Mozilla Developer Network (MDN) - CSS:** <https://developer.mozilla.org/en-US/docs/Web/CSS>
 - **Mozilla Developer Network (MDN) - JavaScript:** <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- **Bootstrap Documentation:**
 - **Bootstrap Official Documentation:** <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
 - **W3Schools Bootstrap Tutorial:** https://www.w3schools.com/bootstrap/boot-strap_get_started.asp
- **Confetti.js Documentation and Examples:**
 - **Confetti.js GitHub Repository:** <https://github.com/mathusummut/confetti.js>
 - **Confetti.js Examples and Demos:** <https://mathusummut.github.io/confetti.js/>
- **SweetAlert2 Documentation and Examples:**
 - **SweetAlert2 Official Documentation:** <https://sweetalert2.github.io/>
 - **SweetAlert2 GitHub Repository:** <https://github.com/sweetalert2/sweetalert2>
- **Frontend Development Tutorials and Articles:**
 - **CSS-Tricks:** <https://css-tricks.com/>
 - **Smashing Magazine - HTML/CSS:** <https://www.smashingmagazine.com/category/css/>
 - **JavaScript.info:** <https://javascript.info/>