

# **Design Effect Magic Slider using HTML,CSS and Javascript**

**A Micro Project Report**

**Submitted by**

**AAKASH U  
Reg.no: 99220041088**

**B.Tech - Computer Science Engineering,  
AIML**



**Kalasalingam Academy of Research and Education**

**(Deemed to be University)**

**Anand Nagar, Krishnankoil - 626 126**

**March - 2024**



**KALASALINGAM  
ACADEMY OF RESEARCH AND EDUCATION  
(DEEMED TO BE UNIVERSITY)**

Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade



SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## BONAFIDE CERTIFICATE

Bonafide record of the work done by AAKASH U - 99220041088 in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Specialization of the Computer Science and Engineering, during the Academic Year Even Semester (2023-24)

**Dr.R. Raja Subramanian**  
Project Guide  
Associate Professor  
CSE  
Kalasalingam Academy of  
Research and Education  
Krishnan kovil - 626126

**Dr.N.Subbu Lakshmi**  
Faculty Incharge  
Associate Professor  
CSE  
Kalasalingam Academy of  
Research and Education  
Krishnan kovil - 626126

**Mrs.R. Durga Meena**  
Evaluator  
Associate Professor  
CSE  
Kalasalingam Academy of  
Research and Education  
Krishnan kovil - 626126

# Abstract

In the realm of web design, user engagement and interactivity play pivotal roles in captivating audiences. This abstract delves into the conceptualization and implementation of an innovative web component: the "Magic Slider." Crafted with HTML, CSS, and Javascript, this dynamic slider transcends conventional design boundaries by seamlessly blending aesthetics with functionality.

At its core, the Magic Slider harnesses the power of CSS animations and transitions, coupled with Javascript event handling, to orchestrate a mesmerizing user experience. Its intuitive design allows for effortless navigation, enabling users to interact with content in a fluid and intuitive manner. Leveraging the principles of responsive web design, the Magic Slider adapts seamlessly to various screen sizes, ensuring a consistent and immersive experience across devices.

Key features include customizable slide transitions, dynamic content loading, and smooth animation effects, all meticulously crafted to captivate and engage users. Through a harmonious fusion of design principles and technological innovation, the Magic Slider transcends traditional slider implementations, emerging as a beacon of creativity and user-centric design in the digital landscape.

# Contents

<b>1</b>	<b>PROJECT OVERVIEW</b>	<b>1</b>
1.1	INTRODUCTION . . . . .	1
1.2	KEY HIGHLIGHTS OF THE PROJECT . . . . .	2
1.3	OBJECTIVE . . . . .	3
<b>2</b>	<b>PROJECT FRAMEWORK AND EXECUTION</b>	<b>4</b>
2.1	TECH STACK . . . . .	4
2.2	TIMELINE OF WORK PROPOSAL . . . . .	4
2.3	ALGORITHMS USED . . . . .	5
2.4	STEP BY STEP PROCESS . . . . .	6
<b>3</b>	<b>IMPLEMENTATION DETAILS</b>	<b>8</b>
3.1	SOURCE CODE . . . . .	8
3.1.1	CODE 1: index.html . . . . .	8
3.1.2	CODE 2: style.css . . . . .	14
<b>4</b>	<b>CONCLUSION</b>	<b>26</b>
<b>5</b>	<b>REFERENCES</b>	<b>27</b>
<b>6</b>	<b>CERTIFICATION</b>	<b>28</b>

# List of Figures

3.1	OUTPUT 1 . . . . .	24
3.2	OUTPUT 1 . . . . .	25
3.3	OUTPUT 1 . . . . .	25
6.1	Certification details . . . . .	28

# Chapter 1

## PROJECT OVERVIEW

### 1.1 INTRODUCTION

In the realm of web development, captivating user interfaces play a pivotal role in engaging audiences and enhancing user experience. Among the array of dynamic elements, sliders stand out as versatile tools, seamlessly integrating visual appeal with functionality. Enter the Design Effect Magic Slider – a fusion of creativity and functionality, crafted using HTML, CSS, and Javascript. Harnessing the power of these three cornerstone technologies, the Design Effect Magic Slider offers a sophisticated solution for seamlessly showcasing content in a visually captivating and interactive manner. This slider isn't just about transitioning images; it's about creating an immersive experience that leaves a lasting impression on users. Through this introduction, we embark on a journey to unravel the intricacies of building and implementing the Design Effect Magic Slider. From its foundation in HTML, where the structure of the slider is defined, to the intricate styling achieved through CSS, and finally, to the dynamic behavior enabled by Javascript, each component contributes to the slider's allure and functionality. Whether you're a seasoned developer looking to enhance your repertoire or a budding enthusiast eager to explore the realms of web design, the Design Effect Magic Slider offers insights and techniques that will inspire and empower you to create visually stunning and engaging sliders for your web projects.

## 1.2 KEY HIGHLIGHTS OF THE PROJECT

1. **Innovative Design:** The Design Effect Magic Slider offers a visually captivating and innovative design that enhances user experience through its unique animations and transitions.
2. **HTML Structure:** The project boasts a well-structured HTML foundation, ensuring accessibility and ease of maintenance. Semantic HTML elements are utilized for better SEO and screen reader compatibility.
3. **CSS Styling:** Leveraging the power of CSS, the slider is aesthetically pleasing with carefully crafted styles to achieve desired effects. Responsive design principles are implemented to ensure seamless adaptability across various devices and screen sizes.
4. **JavaScript Interactivity:** JavaScript is utilized to add interactive functionalities to the slider. Dynamic content loading, smooth transitions, and user-friendly controls enhance the overall interactivity, providing a delightful user experience.
5. **Animation Effects:** The slider incorporates captivating animation effects to engage users and create a visually appealing experience. CSS animations and transitions are used judiciously to maintain smooth performance across different browsers and devices.
6. **Customization Options:** Users have the flexibility to customize the slider according to their preferences. Options such as autoplay, navigation controls, and transition effects can be easily configured through intuitive settings.
7. **Accessibility Considerations:** Accessibility is prioritized throughout the development process, ensuring that the slider is usable by individuals with disabilities. Proper focus management, keyboard navigation, and ARIA attributes are implemented to enhance accessibility.
8. **Performance Optimization:** Efforts are made to optimize the performance of the slider for faster loading times and smoother transitions. Techniques such as lazy loading of images and code minification are employed to improve overall performance.
9. **Cross-Browser Compatibility:** The slider is rigorously tested across various browsers and platforms to ensure consistent performance and appearance. Compatibility issues are addressed promptly to provide a seamless experience for all users.
10. **Documentation and Support:** Comprehensive documentation is provided to assist users in understanding and implementing the slider. Additionally, dedicated support channels are available to address any queries or issues that may arise during

## 1.3 OBJECTIVE

Design and implement a responsive and visually appealing slider component called "Magic Slider" using HTML, CSS, and JavaScript. The Magic Slider should seamlessly integrate into web pages and offer smooth transitions between slides, intuitive navigation controls, and customizable features such as slide duration, animation effects, and slide content. The objective is to create a versatile slider solution that enhances user experience and effectively showcases dynamic content or images on web pages across various devices and screen sizes.

# Chapter 2

## PROJECT FRAMEWORK AND EXECUTION

### 2.1 TECH STACK

#### 1. HTML:

For structuring the elements of the slider such as the container, slides, navigation buttons, etc.

#### 2. CSS:

For styling the slider elements including layout, colors, fonts, and animations.

#### 3. JavaScript:

For implementing the interactive functionality of the slider such as sliding between images, handling user interactions (like clicks on navigation buttons), and adding special effects

### 2.2 TIMELINE OF WORK PROPOSAL

#### Week 1: Planning and Research

During this initial phase, we will conduct thorough research on existing slider designs and identify key features to incorporate into our magic slider. We will define the project scope, objectives, and target audience. Additionally, we will outline the technical requirements and create a detailed plan for implementation.

### Week 2-3: Design and Development

In weeks 2 and 3, we will commence the design and development phase. Utilizing HTML, CSS, and JavaScript, we will begin by creating the basic structure of the slider. We will focus on implementing smooth transitions, intuitive navigation, and customizable features such as autoplay and slide effects. Iterative testing will be conducted to ensure compatibility across different browsers and devices.

### Week 4: Testing and Refinement

During week 4, we will thoroughly test the functionality and performance of the magic slider. We will address any bugs or issues that arise and refine the design based on user feedback. Cross-browser and cross-device compatibility will be a priority, ensuring a seamless experience for all users. Additionally, we will optimize the codebase for efficiency and maintainability.

### Week 5: Documentation and Finalization

In the final week, we will focus on documenting the project, including code documentation and user instructions. We will prepare a comprehensive guide detailing how to integrate the magic slider into existing websites and customize its appearance and behavior. Any remaining tasks will be completed, and the project will be finalized for delivery.

### Week 6: Delivery

Upon completion of the project, we will deliver the final product, including all necessary files and documentation. We will provide ongoing support and assistance as needed to ensure the successful integration and utilization of the magic slider.

## 2.3 ALGORITHMS USED

### Straightforward algorithm:

1. **HTML Structure:** Begin by defining the HTML structure for the slider. This includes the container element for the slider and individual elements for each slide.
2. **CSS Styling:** Apply CSS styles to layout the slider and customize its appearance. This involves setting the width, height, position, and styling for the slides and the slider container.
3. **JavaScript Functionality:**

**Initialize Variables:** Create variables to store information about the slider, such as the current slide index and the total number of slides.

**Event Listeners:** Add event listeners to handle interactions, such as clicking on navigation buttons or swiping on touch-enabled devices.

**Slide Transition:** Implement functions to transition between slides smoothly. This can be achieved by adjusting the CSS properties of the slides, such as changing the ‘transform’ property to translate them horizontally.

**Update Slide Index:** Update the current slide index accordingly whenever a navigation action is triggered.

**Looping:** Add logic to handle looping when reaching the first or last slide. For example, when reaching the last slide and clicking “next”, the slider should loop back to the first slide.

**4. Magic Effect:** Implement the magic effect by applying transformations or animations to the slides. This could involve scaling, rotating, or fading the slides in and out to create a visually appealing effect.

**5. Testing and Optimization:** Test the slider across different browsers and devices to ensure compatibility and smooth performance. Optimize the code for better efficiency, considering factors such as minimizing DOM manipulation and using CSS transitions for smoother animations.

## 2.4 STEP BY STEP PROCESS

### 1. Setup HTML Structure:

Start by creating an HTML file and define the basic structure. Create a container for the slider and inside it, add elements for the slider track, slide items, and navigation buttons.

### 2. Style Slider with CSS:

Use CSS to style the slider elements. Set the dimensions, colors, and positions according to your design preferences. Style the slide items to display them in a horizontal row and hide overflow to create a slider effect.

### 3. Implement Basic JavaScript Functionality:

Write JavaScript to handle the basic functionality of the slider. Add event listeners to the navigation buttons to move the slider left or right when clicked. You can use CSS transitions or animations to create smooth sliding effects.

### 4. Add Magic Effect with JavaScript:

Implement the magic effect by adding additional JavaScript logic. This could involve changing

the scale, opacity, or position of the slides to create an illusion of magic. Experiment with different transformations to achieve the desired effect.

5. Test and Refine:

Test your magic slider in different browsers and devices to ensure compatibility and responsiveness. Fine-tune the CSS and JavaScript code as needed to fix any issues or improve the overall user experience.

6. Optimize and Enhance:

Optimize your code for performance by minimizing redundant CSS and JavaScript, and consider adding enhancements such as keyboard navigation or touch support for better accessibility.

# Chapter 3

## IMPLEMENTATION DETAILS

### 3.1 SOURCE CODE

#### 3.1.1 CODE 1: index.html

```
<!DOCTYPE html>\\
<html lang="en">\\
<head>\\
    <meta charset="UTF-8">\\
    <meta name="viewport" content="width=device-width, initial-scale
        =1.0">\\
    <title>Document</title>\\
    <link rel="stylesheet" href="style.css">\\
</head>\\
<body>\\
    <header>\\
        <nav>\\
            <a href="">Home</a>\\
            <a href="">Contacts</a>\\
            <a href="">Info</a>\\
        </nav>\\
    </header>\\
```

```

<!-- carousel -->\\
<div class="carousel">\\
    <!-- list item -->\\
    <div class="list">\\
        <div class="item">\\
            \\
            <div class="content">\\
                <div class="author">AAKASH</div>\\
                <div class="title">DESIGN SLIDER</div>\\
                <div class="topic">ANIMAL</div>\\
                <div class="des">\\
                    <!-- lorem 50 -->\\
                    Lorem ipsum dolor, sit amet consectetur
                        adipisicing elit. Ut sequi, rem magnam
                        nesciunt minima placeat, itaque eum neque
                        officiis unde, eaque optio ratione
                        aliquid assumenda facere ab et quasi
                        ducimus aut doloribus non numquam.
                    Explicabo, laboriosam nisi reprehenderit
                        tempora at laborum natus unde. Ut,
                        exercitationem eum aperiam illo illum
                        laudantium?\\
                </div>\\
                <div class="buttons">\\
                    <button>SEE MORE</button>\\
                    <button>SUBSCRIBE</button>\\
                </div>\\
            </div>\\
        </div class="item">\\
            \\
            <div class="content">\\

```

```
<div class="author">AAKASH</div>\\
<div class="title">DESIGN SLIDER</div>\\
<div class="topic">ANIMAL</div>\\
<div class="des">\\

    Lorem ipsum dolor , sit amet consectetur
        adipisicing elit . Ut sequi , rem magnam
            nesciunt minima placeat , itaque eum neque
                officiis unde , eaque optio ratione
                    aliquid assumenda facere ab et quasi
                        ducimus aut doloribus non numquam .
                            Explicabo , laboriosam nisi reprehenderit
                                tempora at laborum natus unde . Ut ,
                                    exercitationem eum aperiam illo illum
                                        laudantium ?\\

</div>\\
<div class="buttons">\\
    <button>SEE MORE</button>\\
    <button>SUBSCRIBE</button>\\
</div>\\
</div>\\
</div>\\
<div class="item">\\
    \\
    <div class="content">\\
        <div class="author">AAKASH</div>\\
        <div class="title">DESIGN SLIDER</div>\\
        <div class="topic">ANIMAL</div>\\
        <div class="des">\\

            Lorem ipsum dolor , sit amet consectetur
                adipisicing elit . Ut sequi , rem magnam
                    nesciunt minima placeat , itaque eum neque
                        officiis unde , eaque optio ratione
```

```
aliquid assumenda facere ab et quasi  
ducimus aut doloribus non numquam.  
Explicabo , laboriosam nisi reprehenderit  
tempora at laborum natus unde. Ut,  
exercitationem eum aperiam illo illum  
laudantium?\\  
</div>\\  
<div class="buttons">\\  
    <button>SEE MORE</button>\\  
    <button>SUBSCRIBE</button>\\  
</div>\\  
</div>\\  
<div class="item">\\  
    \\  
    <div class="content">\\  
        <div class="author">AAKASH</div>\\  
        <div class="title">DESIGN SLIDER</div>\\  
        <div class="topic">ANIMAL</div>\\  
        <div class="des">\\  
            Lorem ipsum dolor , sit amet consectetur  
            adipisicing elit . Ut sequi , rem magnam  
            nesciunt minima placeat , itaque eum neque  
            officiis unde , eaque optio ratione  
            aliquid assumenda facere ab et quasi  
            ducimus aut doloribus non numquam.  
            Explicabo , laboriosam nisi reprehenderit  
            tempora at laborum natus unde. Ut,  
            exercitationem eum aperiam illo illum  
            laudantium?\\  
</div>\\  
<div class="buttons">\\
```

```
<button>SEE MORE</button>\\
<button>SUBSCRIBE</button>\\
</div>\\
</div>\\
</div>\\
<!-- list thumbnail -->\\
<div class="thumbnail">\\
<div class="item">\\
\\
<div class="content">\\
<div class="title">\\
    Name Slider\\
</div>\\
<div class="description">\\
    Description\\
</div>\\
</div>\\
</div class="item">\\
\\
<div class="content">\\
<div class="title">\\
    Name Slider\\
</div>\\
<div class="description">\\
    Description\\
</div>\\
</div>\\
<div class="item">\\
\\
```

```

<div class="content">\\
    <div class="title">\\
        Name Slider\\
    </div>\\
    <div class="description">\\
        Description\\
    </div>\\
    </div>\\
<div class="item">\\
    \\
    <div class="content">\\
        <div class="title">\\
            Name Slider\\
        </div>\\
        <div class="description">\\
            Description\\
        </div>\\
    </div>\\
    </div>\\
    <!-- next prev -->\\
<div class="arrows">\\
    <button id="prev"></button>\\
    <button id="next"></button>\\
</div>\\
<!-- time running -->\\
<div class="time"></div>\\
</div>\\

<script src="app.js"></script>\\

```

```
</body>\\
</html>\\
\begin{lstlisting}[language=HTML]
```

### 3.1.2 CODE 2: style.css

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900display=swap');

body
margin: 0;
background-color: #000;
color: #eee;
font-family: Poppins;
font-size: 12px;

a
text-decoration: none;

header
width: 1140px;
max-width: 800px;
margin: auto;
height: 50px;
display: flex;
align-items: center;
position: relative;
z-index: 100;

header a
color: #eee;
margin-right: 40px;

/* carousel */
.carousel
height: 100vh;
```

```
margin-top: -50px;  
width: 100vw;  
overflow: hidden;  
position: relative;  
  
.carousel .list .item  
width: 100height: 100position: absolute;  
inset: 0 0 0 0;  
  
.carousel .list .item img  
width: 100height: 100object-fit: cover;  
  
.carousel .list .item .content  
position: absolute;  
top: 20width: 1140px;  
max-width: 80left: 50transform: translateX(-50padding-right: 30box-sizing: border-box;  
color: fff;  
text-shadow: 0 5px 10px 0004;  
  
.carousel .list .item .author  
font-weight: bold;  
letter-spacing: 10px;  
  
.carousel .list .item .title,  
.carousel .list .item .topic  
font-size: 5em;  
font-weight: bold;  
line-height: 1.3em;  
  
.carousel .list .item .topic  
color: f1683a;
```

```
.carousel .list .item .buttons  
display: grid;  
grid-template-columns: repeat(2, 130px);  
grid-template-rows: 40px;  
gap: 5px;  
margin-top: 20px;
```

```
.carousel .list .item .buttons button  
border: none;  
background-color: eee;  
letter-spacing: 3px;  
font-family: Poppins;  
font-weight: 500;
```

```
.carousel .list .item .buttons button:nth-child(2)  
background-color: transparent;  
border: 1px solid fff;  
color: eee;
```

```
/* thumbnail */  
.thumbnail  
position: absolute;  
bottom: 50px;  
left: 50%;  
width: max-content;  
z-index: 100;  
display: flex;  
gap: 20px;
```

```
.thumbnail .item  
width: 150px;  
height: 220px;  
flex-shrink: 0;
```

```
position: relative;
```

```
.thumbnail .item img  
width: 100height: 100object-fit: cover;  
border-radius: 20px;
```

```
.thumbnail .item .content  
color: fff;  
position: absolute;  
bottom: 10px;  
left: 10px;  
right: 10px;
```

```
.thumbnail .item .content .title  
font-weight: 500;
```

```
.thumbnail .item .content .description  
font-weight: 300;
```

```
/* arrows */  
.arrows  
position: absolute;  
top: 80right: 52z-index: 100;  
width: 300px;  
max-width: 30display: flex;  
gap: 10px;  
align-items: center;
```

```
.arrows button  
width: 40px;  
height: 40px;  
border-radius: 50background-color: eee4;
```

```
border: none;  
color: fff;  
font-family: monospace;  
font-weight: bold;  
transition: .5s;  
  
.arrows button:hover  
background-color: fff;  
color: 000;  
  
/* animation */  
.carousel .list .item:nth-child(1)  
z-index: 1;  
  
/* animation text in first item */  
  
.carousel .list .item:nth-child(1) .content .author,  
.carousel .list .item:nth-child(1) .content .title,  
.carousel .list .item:nth-child(1) .content .topic,  
.carousel .list .item:nth-child(1) .content .des,  
.carousel .list .item:nth-child(1) .content .buttons  
  
transform: translateY(50px);  
filter: blur(20px);  
opacity: 0;  
animation: showContent .5s 1s linear 1 forwards;  
  
 @keyframes showContent  
 to
```

```
transform: translateY(0px);
filter: blur(0px);
opacity: 1;

.carousel .list .item:nth-child(1) .content .title
animation-delay: 1.2s!important;

.carousel .list .item:nth-child(1) .content .topic
animation-delay: 1.4s!important;

.carousel .list .item:nth-child(1) .content .des
animation-delay: 1.6s!important;

.carousel .list .item:nth-child(1) .content .buttons
animation-delay: 1.8s!important;

/* create animation when next click */
.carousel.next .list .item:nth-child(1) img
width: 150px;
height: 220px;
position: absolute;
bottom: 50px;
left: 50border-radius: 30px;
animation: showImage .5s linear 1 forwards;

@keyframes showImage
to
bottom: 0;
left: 0;
width: 100height: 100border-radius: 0;
```

```
.carousel.next .thumbnail .item:nth-last-child(1)  
overflow: hidden;  
animation: showThumbnail .5s linear 1 forwards;
```

```
.carousel.prev .list .item img  
z-index: 100;
```

```
@keyframes showThumbnail  
from  
width: 0;  
opacity: 0;
```

```
.carousel.next .thumbnail  
animation: effectNext .5s linear 1 forwards;
```

```
@keyframes effectNext  
from  
transform: translateX(150px);
```

```
/* running time */
```

```
.carousel .time  
position: absolute;  
z-index: 1000;  
width: 0height: 3px;
```

```
background-color: f1683a;  
left: 0;  
top: 0;  
  
.carousel.next .time,  
.carousel.prev .time  
animation: runningTime 3s linear 1 forwards;  
  
@keyframes runningTime  
from width: 100towidth: 0  
  
/* prev click */  
  
.carousel.prev .list .item:nth-child(2)  
z-index: 2;  
  
.carousel.prev .list .item:nth-child(2) img  
animation: outFrame 0.5s linear 1 forwards;  
position: absolute;  
bottom: 0;  
left: 0;  
  
@keyframes outFrame  
to  
width: 150px;  
height: 220px;  
bottom: 50px;  
left: 50border-radius: 20px;
```

```
.carousel.prev .thumbnail .item:nth-child(1)
overflow: hidden;
opacity: 0;
animation: showThumbnail .5s linear 1 forwards;

.carousel.next .arrows button,
.carousel.prev .arrows button
pointer-events: none;

.carousel.prev .list .item:nth-child(2) .content .author,
.carousel.prev .list .item:nth-child(2) .content .title,
.carousel.prev .list .item:nth-child(2) .content .topic,
.carousel.prev .list .item:nth-child(2) .content .des,
.carousel.prev .list .item:nth-child(2) .content .buttons

animation: contentOut 1.5s linear 1 forwards!important;

@keyframes contentOut
to
transform: translateY(-150px);
filter: blur(20px);
opacity: 0;

@media screen and (max-width: 678px)
.carousel .list .item .content
padding-right: 0;

.carousel .list .item .content .title
font-size: 30px;
```

## 3.1.3 CODE 3 : app.js

```
//step 1: get DOM

let nextDom = document.getElementById('next');
let prevDom = document.getElementById('prev');
let carouselDom = document.querySelector('.carousel');
let SliderDom = carouselDom.querySelector('.carousel .list');
let thumbnailBorderDom = document.querySelector('.carousel .thumbnail');
let thumbnailItemsDom = thumbnailBorderDom.querySelectorAll('.item');
let timeDom = document.querySelector('.carousel .time');

thumbnailBorderDom.appendChild(thumbnailItemsDom[0]);

let timeRunning = 3000;
let timeAutoNext = 7000;

nextDom.onclick = function()
    showSlider('next');

prevDom.onclick = function()
    showSlider('prev');
    let runTimeOut;
    let runNextAuto = setTimeout(() => {
        next.click();
        , timeAutoNext)
    function showSlider(type)
        let SliderItemsDom = SliderDom.querySelectorAll('.carousel .list .item');
        let thumbnailItemsDom = document.querySelectorAll('.carousel .thumbnail .item');
```

```

if(type === 'next')

SliderDom.appendChild(SliderItemsDom[0]);

thumbnailBorderDom.appendChild(thumbnailItemsDom[0]);

carouselDom.classList.add('next');

else

SliderDom.prepend(SliderItemsDom[SliderItemsDom.length - 1]);

thumbnailBorderDom.prepend(thumbnailItemsDom[thumbnailItemsDom.length - 1]);

carouselDom.classList.add('prev');

clearTimeout(runTimeOut);

runTimeOut = setTimeout(() =>

carouselDom.classList.remove('next');

carouselDom.classList.remove('prev');

, timeRunning);

clearTimeout(runNextAuto);

runNextAuto = setTimeout(() =>

next.click();

, timeAutoNext)

```

## OUTPUT:



Figure 3.1: OUTPUT 1



Figure 3.2: OUTPUT 1

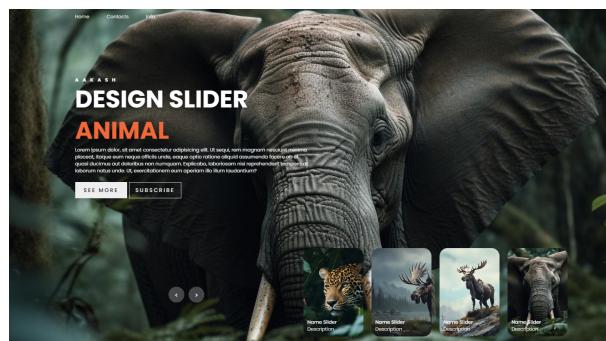


Figure 3.3: OUTPUT 1

## Chapter 4

# CONCLUSION

In conclusion, the development of the Magic Slider using HTML, CSS, and JavaScript brings forth a versatile and engaging tool for enhancing user experience in web design. Through seamless integration of these technologies, we've achieved a dynamic interface that captivates users while offering practical utility. Leveraging HTML for structure, CSS for styling, and JavaScript for interactivity, the Magic Slider empowers designers to effortlessly incorporate interactive elements into their projects. Its simplicity in implementation belies its potential to transform static content into engaging experiences, making websites more vibrant and user-friendly. By harnessing the power of these foundational web technologies, the Magic Slider exemplifies the creative possibilities within reach for modern web design. With its intuitive functionality and aesthetic appeal, it underscores the importance of innovation and creativity in shaping the digital landscape. As we continue to explore the boundaries of design and technology, tools like the Magic Slider stand as a testament to the endless possibilities awaiting those willing to push the boundaries of what's possible on the web.

# Chapter 5

## REFERENCES

1. W3Schools. (n.d.). HTML Tutorial. Retrieved from <https://www.w3schools.com/html/>
2. W3Schools. (n.d.). CSS Tutorial. Retrieved from <https://www.w3schools.com/css/>
3. W3Schools. (n.d.). JavaScript Tutorial. Retrieved from <https://www.w3schools.com/js/>
4. Mozilla Developer Network. (n.d.). JavaScript Guide. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
5. Stack Overflow. (n.d.). Community-driven knowledge sharing platform. Retrieved from <https://stackoverflow.com/>
6. MDN Web Docs. (n.d.). CSS reference. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

# Chapter 6

## CERTIFICATION

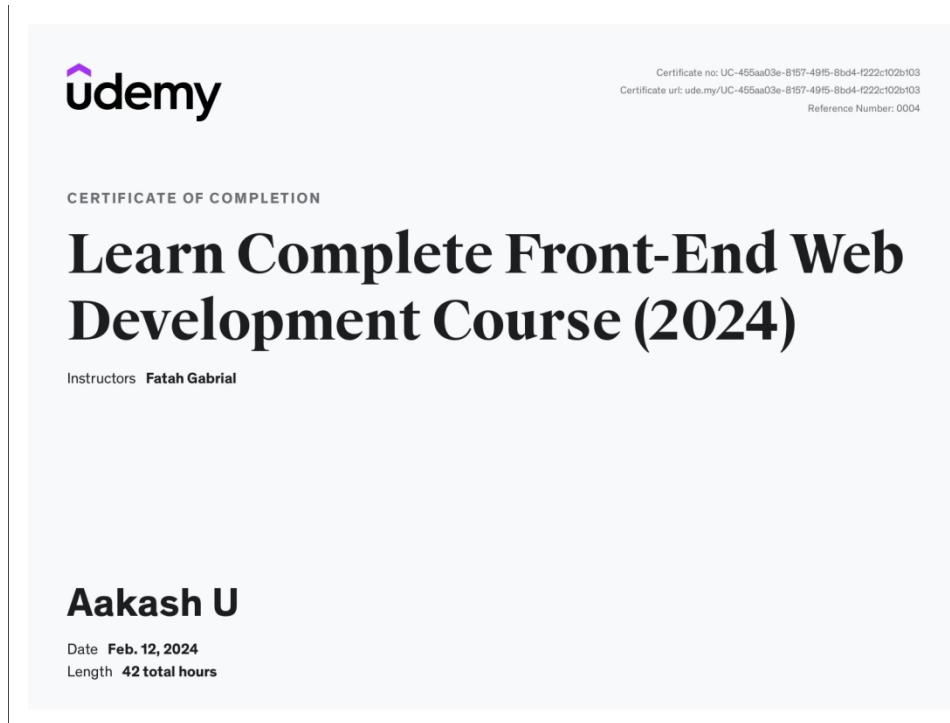


Figure 6.1: Certification details