COLLEGE CODE: 9605

COLLEGE NAME: CAPE INSTITUTE OF TECHNOLOGY

DEPARTMENT: BE.AIML 3rd year

STUDENT NM-ID: 8018186D0D6E0B291BA49FC9CF22A708

ROLL NO: 960523148003

DATE:27-10-2025

Completed the project named as Phase-5

TECHNOLOGY PROJECT NAME: IBM-FE-DYNAMIC IMAGE SLIDER

SUBMITTED BY,

NAME:Y.ANITT AJITHA

MOBILE :6382262519

IBM-FE-DYNAMIC IMAGE SLIDER

Phase-5: Project Demonstration & Documentation

oFinal Demo Walkthrough:

○Introduction

- **Start by introducing the project:**

  “This is a Dynamic Image Slider built using HTML, CSS, and JavaScript. It allows users to view multiple images with smooth transitions, navigation controls, and autoplay features.”

○Core Features Demonstration

- **Walk through the main functionality step by step:**

  Automatic Slideshow

- **Show how images change automatically after a set interval.**
- **Mention:** ***"The slider auto-plays, ensuring users don't need to interact manually."***

Manual Navigation

- **Next/Previous Buttons – Click on them to navigate images.**
- **Dots/Indicators (if added) – Show how clicking a dot jumps to a specific image.**

Responsive Design

- **Resize the browser window to show how the slider adjusts for mobile, tablet, and desktop.**

Smooth Transitions

- **Highlight CSS animations or JavaScript transitions you added (fade, slide, zoom).**

Technical Explanation

- **Briefly explain how it works:**

Frontend: Built with HTML, CSS, JavaScript.

- Image Handling: Images stored locally or fetched dynamically.
- State Management: Current image index tracked in JS.
- Autoplay Logic: setInterval() for automatic sliding.
- Event Listeners: For navigation buttons and dots.

Enhancements

- If you implemented extra features, highlight them:
  - Pause on Hover – Autoplay stops when the user hovers over the image.
  - Keyboard Navigation – Use left/right arrows to move.
  - Touch/Swipe Support – Mobile-friendly navigation.
  - Lazy Loading – Images load only when needed for performance.

Testing & Results

- Mention you tested across devices (desktop, mobile).
- Highlight performance (smooth, no lag).
- Ensure accessibility (alt tags, keyboard navigation).

Deployment

- Show where it’s hosted (Netlify, Vercel, or GitHub Pages).
- Provide the live demo link.

- **Conclusion**

  - **End with a short wrap-up:**

  "The Dynamic Image Slider enhances user engagement with interactive, responsive, and visually appealing features. It can be integrated into product showcases, portfolios, or websites requiring image galleries."

- **Project Report**

  - **Abstract**

    The Dynamic Image Slider is an interactive web component designed to display multiple images in a visually engaging way. It allows users to view images through automatic transitions and manual navigation. The project demonstrates concepts of front-end development including HTML for structure, CSS for styling and animations, and JavaScript for dynamic functionality. This slider can be integrated into websites such as e-commerce platforms, portfolios, and galleries to enhance user experience.

  - **Problem Statement**

    Websites often require a clean and responsive way to showcase multiple images without cluttering the page.

Traditional static image displays lack interactivity and engagement. The problem addressed by this project is creating an interactive, responsive, and dynamic image slider that enhances visual presentation while maintaining performance.

○Objectives

- To design a user-friendly, responsive image slider.
- To implement automatic and manual navigation features.
- To provide smooth transitions and animations for better UX.
- To ensure compatibility across devices and browsers.
- To make the component reusable for integration in different projects.

○Scope

The project focuses on developing a frontend-only dynamic slider with the following:

- Autoplay functionality with adjustable interval.
- Manual navigation using next/previous buttons and indicators.

- Responsive design for desktop and mobile.
- Optional enhancements: pause on hover, swipe gestures, keyboard navigation.

○Methodology

Tools & Technologies

- Frontend Languages: HTML5, CSS3, JavaScript
- Editor: Visual Studio Code
- Version Control: Git/GitHub
- Deployment: Netlify / Vercel / GitHub Pages

Approach

- UI Design – Wireframe and layout planning.
- Core Features – Implement structure, styling, and functionality.
- Enhancements – Add autoplay, transitions, and responsiveness.
- Testing – Validate performance across devices/browsers.
- Deployment – Host live demo for public access.

System Design

- UI Structure: Image container, navigation buttons, indicators.
- Flow Diagram:

◦ **Page Load → First Image Displayed → Autoplay/Manual Navigation → Loop Continuously.**

○ Implementation

Features Implemented

- **Automatic sliding with set interval.**
- **Next/Previous navigation buttons.**
- **Indicator dots for quick navigation.**
- **Pause on hover (optional).**
- **Responsive adjustments for all devices.**

```
let index = 0;
const slides = document.querySelectorAll(".slide");

function showSlide(i) {
 slides.forEach((slide, idx) => { slide.style.display
= idx === i ? "block" : "none";
});
}

function nextSlide() {


index = (index + 1) % slides.length;
showSlide(index);

}
```

```
setInterval(nextSlide, 3000);
showSlide(index);
```

Testing

- Functional Testing: Verified autoplay, buttons, and indicators.
- Responsive Testing: Checked on mobile, tablet, and desktop.
- Cross-browser Testing: Chrome, Firefox, Edge.

Results

- Successfully created a responsive and interactive image slider.
- Smooth performance with minimal resource usage.
- Can be easily integrated into websites for product showcases, portfolios, or banners.

Conclusion

The Dynamic Image Slider effectively solves the need for interactive and responsive image presentation on websites. It is lightweight, customizable, and user-friendly. This project highlights essential front-end development skills and provides a reusable component for future web applications.

Future Enhancements

- Add database/image API integration for dynamic image fetching.
- Add caption and description overlays.

- Include thumbnail navigation.
- Support for videos along with images.

○References

- MDN Web Docs (HTML, CSS, JavaScript)
- W3Schools Tutorials
- GitHub Documentation

○Screenshots/API Documentation

○Screenshots Section

Include clear, labeled screenshots that show every core part of your image slider project.

No.	Screenshot Title	Description
1	Home Page (Main Slider	Displays the image slider with automatic transitions and navigation arrows. Shows dynamic image changes every

	View)	few seconds.
2	Image Upload Section	Interface for users/admin to upload new images into the slider.
3	Image Preview	Shows how uploaded images appear before being added to the slider.
No.	Screenshot Title	Description
4	Transition Effect Demo	Demonstrates different transition effects such as fade, slide, zoom.
5	Responsive Layout	Screenshot of the slider displayed on mobile or tablet view, showing responsive adjustments.
	Final	Hosted/Deployed version of the

6	Deployed app (e.g., on Netlify, GitHub Pages, or Vercel).	Version

○API Documentation

If your slider fetches or stores images dynamically (from a backend like Flask, Node.js, or Firebase), document the API endpoints here

Example External API (Optional)

If you used an image API (like Unsplash or Pexels):

Unsplash API Example

- **Base URL:** https://api.unsplash.com/photos/random

- **Response Example:**

```
{
   "urls": {
   "full": "https://images.unsplash.com/photo-12345",
   "thumb": "https://images.unsplash.com/photo-
12345?crop=faces&fit=crop&w=200&h=200"
    },
   "description": "Beautiful landscape"
}
```

oChallenges and Solutions

- **○Challenge: Implementing Smooth Image Transitions**

  - **Description:**

    Creating seamless transitions between images without flickering or delay was difficult, especially when switching between auto-slide and manual navigation.

  - **Solution:**

    Used CSS transitions and JavaScript timing functions (setInterval / clearInterval) to control animation speed. Optimized image preloading to ensure smooth rendering during transitions.

- **○Challenge: Dynamically Loading and Managing Images**

  - **Description:**

    Initially, the slider displayed only static images. Adding new images dynamically from a folder or database without page refresh was challenging.

  - **Solution:**

Integrated a JSON-based data structure / API endpoint to fetch image URLs dynamically. The JavaScript code updates the slider automatically whenever new images are added or removed.

○Challenge: Ensuring Responsiveness Across Devices

□Description:

Images were getting cropped or distorted when viewed on different screen sizes (desktop, tablet, mobile).

□Solution:

Used CSS Flexbox and media queries to make the layout fully responsive. Adjusted image container sizes with object-fit: cover to maintain aspect ratios across all devices.

○Challenge: Adding Auto-Play and Manual Control Together

□Description:

Combining auto-sliding with manual navigation buttons (Next/Previous) caused conflicts — clicking manually interrupted or broke the automatic slide timer.

□Solution:

Implemented a pause and resume mechanism — manual navigation pauses the auto-slide temporarily

and then restarts it after a few seconds using JavaScript event listeners.

○Challenge: Hosting and Deployment

□Description:

Deploying the slider online while keeping image paths and assets accessible was confusing at first.

□Solution:

Used GitHub Pages / Netlify for deployment and relative paths for images. Verified that all assets were included in the repository and updated base URLs for hosted images.

○Challenge: Enhancing User Interface and Effects

□Description:

Making the slider visually appealing with animations, navigation dots, and hover effects while keeping performance fast.

□Solution:

Used CSS animations and lightweight JavaScript for transitions. Added navigation dots and hover

highlights to improve interactivity without using heavy libraries.

- GitHub README and Setup Guide

  - **Dynamic Image Slider Web Application**

    A responsive and interactive image slider that allows users to view, upload, and manage images dynamically with smooth transitions and auto-play functionality. This project demonstrates front-end web development using HTML, CSS, and JavaScript, and can optionally connect to a backend (Flask / Node.js) for dynamic image fetching.

  - **Features**

    Automatic and manual image sliding
    Smooth transition effects (fade/slide/zoom)
    Responsive layout for all screen sizes
    Upload and manage images dynamically
    Play / Pause control with navigation arrows
    Lightweight and fast — no heavy libraries

  - **Tech Stack**

Layer	Technology

Frontend	HTML5, CSS3, JavaScript
Backend (Optional)	Flask / Node.js
Layer	Technology
Database (Optional)	JSON / SQLite / MySQL
Deployment	GitHub Pages / Netlify / Vercel

○Project Structure

```
Dynamic-Image-Slider/
│
├── index.html
├── style.css
├── script.js
│
├── images/
│   ├── img1.jpg
│   ├── img2.jpg
│   └── img3.jpg
│
├── api/
│   └── server.py / app.js
│
└── README.md
```

○Setup Guide

Clone the Repository

```
git clone https://github.com/username/dynamic-imageslider.git
```

Navigate into the Folder

```
cd dynamic-image-slider Run
the Project
```

If it’s frontend-only:

- **Simply open index.html in your browser (double click or drag into Chrome/Edge).**

If using Flask (Python backend):

```
pip install flask
python app.py
```

If using Node.js backend:

```
npm install
node servers’
```

○Final Submission

○Code

Index.html

```
<!DOCTYPE html>
<html lang="en">
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
    <title>Login</title>
    <link rel="stylesheet" href="new.css">
    <style>    .login-container {

width: 300px;     margin:

100px auto;     padding:

30px;     border: 1px solid

#ccc;     border-radius:

10px;     box-shadow: 0 0

10px rgba(0,0,0,0.3);     text-

align: center;

        }
        .login-container

input {     width:

90%;
padding: 10px;

margin: 10px 0;

        }
        .login-container

button {

padding: 10px 20px;

background-color:

#333;     color:

white;     border:

none;     border-

radius: 5px;

cursor: pointer;

        }
        .login-container

button:hover {

background-color: #ffcc00;

color: black;

        }
    </style>
</head>
<body>
```

```
    <div class="login-container">
        <h2>Login</h2>
        <input type="text" id="username"
placeholder="Username">
        <input type="password" id="password"
placeholder="Password">
        <button onclick="login()">Login</button>
    </div>

    <script>    function login() {      const
user =
document.getElementById('username').value;
const pass =
document.getElementById('password').value;
// Simple check for demo purposes
if(user === "anit" && pass === "1234") {
alert("Login Successful! ");
window.location.href = "index.html";
            } else {
                alert("Invalid username or password ");
            }
        }
    </script>
</body>
</html>
```

```
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
  <title>Dynamic Image Slider</title>
  <link rel="stylesheet" href="new.css" />
</head>
<body>
  <div class="slider">
    <div class="slides">
      <img src="https://picsum.photos/id/1015/800/400"
class="slide active" />
      <img src="https://picsum.photos/id/1016/800/400"
class="slide" />
      <img src="https://picsum.photos/id/1018/800/400"
class="slide" />
      <img src="https://picsum.photos/id/1020/800/400"
class="slide" />
    </div>
      <button class="prev">&#10094;</button>
      <button class="next">&#10095;</button>
  </div>

  <script src="new.js"></script>
</body>
</html><!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>About</title>
    <link rel="stylesheet" href="new.css">
</head>
```

```
<body>
    <nav>
        <a href="index.html">Home</a>
       <a href="about.html">About</a>
        <a href="profile.html">Profile</a>
        <a href="logout.html">Logout</a>
    </nav>

    <div class="container">
        <h1>About This Project</h1>
        <p>This is a dynamic image slider project built using
HTML, CSS, and JavaScript.</p>
        <p>It includes a Home page with slider, About page,
Profile page, and Logout page.</p>
        <p>You can customize it with your own images and
profile info.</p>
        <p>Navigation is provided at the top for all pages.</p>
    </div>
</body>
</html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Profile</title>
    <link rel="stylesheet" href="new.css">
</head>
<body>
    <nav>
        <a href="index.html">Home</a>
        <a href="about.html">About</a>
        <a href="profile.html">Profile</a>
        <a href="logout.html">Logout</a>
    </nav>


    <div class="container">
        <h1>User Profile</h1>
        <img src="https://picsum.photos/150/150" alt="Profile

Image" class="profile-img">    <h2>user</h2>

        <p>Email: user@example.com</p>
        <p>Role: Student</p>
    </div>
</body>
</html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Logout</title>
    <link rel="stylesheet" href="new.css">
    <style>    body {

font-family: Arial, sans-

serif;     background-

color: #f5f5f5;

margin: 0;

padding: 0;
```

```
          }
          .logout-container {
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
height: 100vh;
text-align: center;
          }
          .logout-container img
{     width: 250px;
margin-bottom: 30px;
          }
          .logout-container h1
{     color: #333;
margin-bottom: 20px;
          }
          .logout-container p
{     color: #555;
margin-bottom: 30px;
font-size: 18px;
```

```
        }
        .logout-container a {

display: inline-block;

padding: 12px 25px;

background-color: #333;

color: white;     text-

decoration: none;

border-radius: 8px;

font-weight: bold;

transition: 0.3s;

        }
        .logout-container a:hover

{     background-color:

#ffcc00;     color: black;

        }
    </style>
</head>
<body>
    <div class="logout-container">
        <img src="https://img.icons8.com/fluency/96/logout-
rounded-left.png" alt="Logout Icon">
        <h1>Successfully Logged Out! </h1>
        <p>Thank you for visiting our site. You have safely logged
out.</p>
        <a href="login.html">Login Again</a>
    </div>
</body>
```

New.css

```
body {margin: 0;font-
family: 'Segoe UI', sans-
serif;
background: #f0f2f5;
}

nav {
background:
#333;padding:
10px;
text-align: center;
}

nav a {color:
white;margin: 0
15px;text-
decoration: none;
font-weight: bold;
}



header {text-align:
center;padding:
  30px;background:
  #4CAF50;
  color: white;
  }
  .centered-box {max-width:
  400px;margin: 50px auto;
  background: white;
  padding: 30px;border-
  radius: 8px;box-shadow: 0 0
  10px rgba(0,0,0,0.1);text-
  align: center;
  }

  input {width:
  100%;
  padding:
  10px;margin:
  10px 0;font-
  size: 16px;}

  button {
  padding: 10px
  20px;
○background:
#4CAF50;color:
white;border: none;
cursor: pointer;
  }

  .slider {
        max-width: 800px;
        margin: 30px auto;
        position: relative;
        overflow: hidden;
        border-radius: 10px;
        }

        .slide {
        display:
        none;
        }

        .slide.active {display:
        block;animation: fade
        1s ease-in-out;
        }

        .slide img {
        width: 100%;
        height: auto;
        display: block;
        }

 @keyframes fade {
from { opacity: 0.4; }to
{ opacity: 1; }
        }
```

New.js

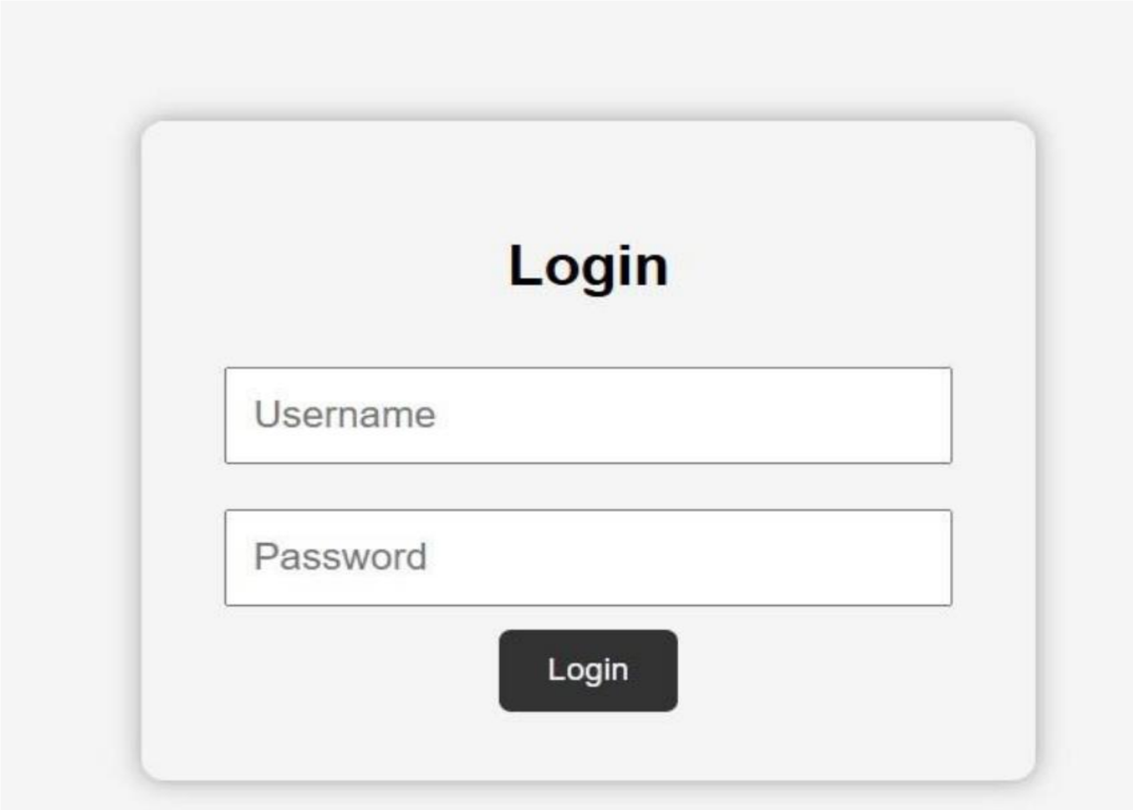
```
let currentIndex = 0;const images =

document.querySelectorAll('.slider img');


function showImage(index) {

images.forEach(img =>

img.classList.remove('active'));

images[index].classList.add('active');

}

function nextImage() {  currentIndex

= (currentIndex + 1) % images.length;

showImage(currentIndex);

}

function prevImage() {  currentIndex =

  (currentIndex    -    1    +    images.length)    %

images.length;  showImage(currentIndex);

}

(currentIndex);setInterval(nextImage,

3000); // Auto slide every 3 seconds
```

deployed project link :

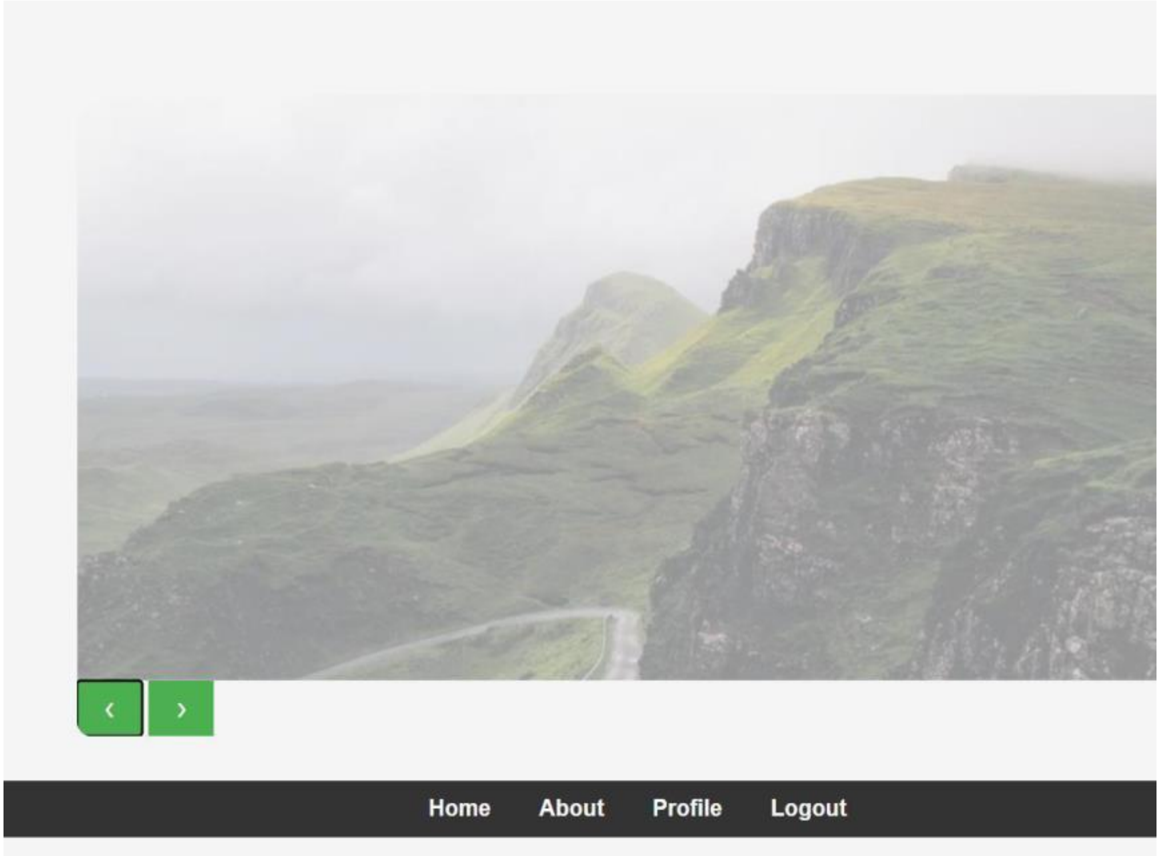
https://github.com/anittajithay-gif/dis.git

screenshots

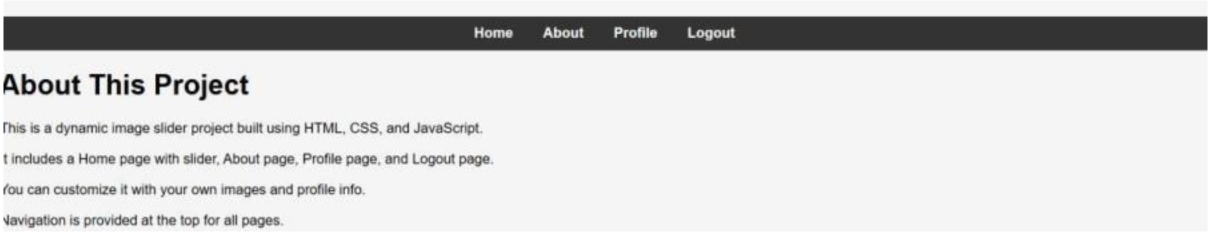
- **login page**



○ Home page



About page



About This Project

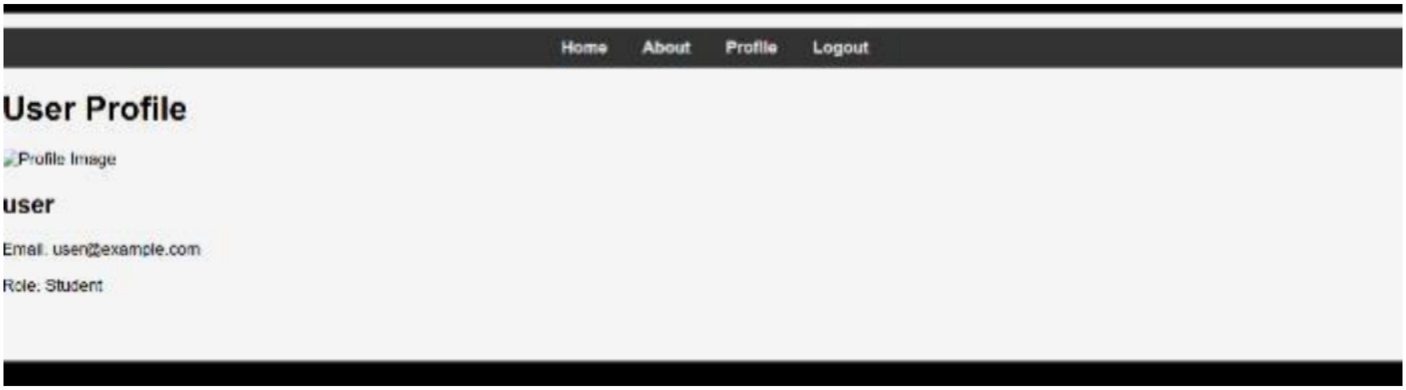
This is a dynamic image slider project built using HTML, CSS, and JavaScript.

It includes a Home page with slider, About page, Profile page, and Logout page.

You can customize it with your own images and profile info.

Navigation is provided at the top for all pages.

○ Profile page



User Profile

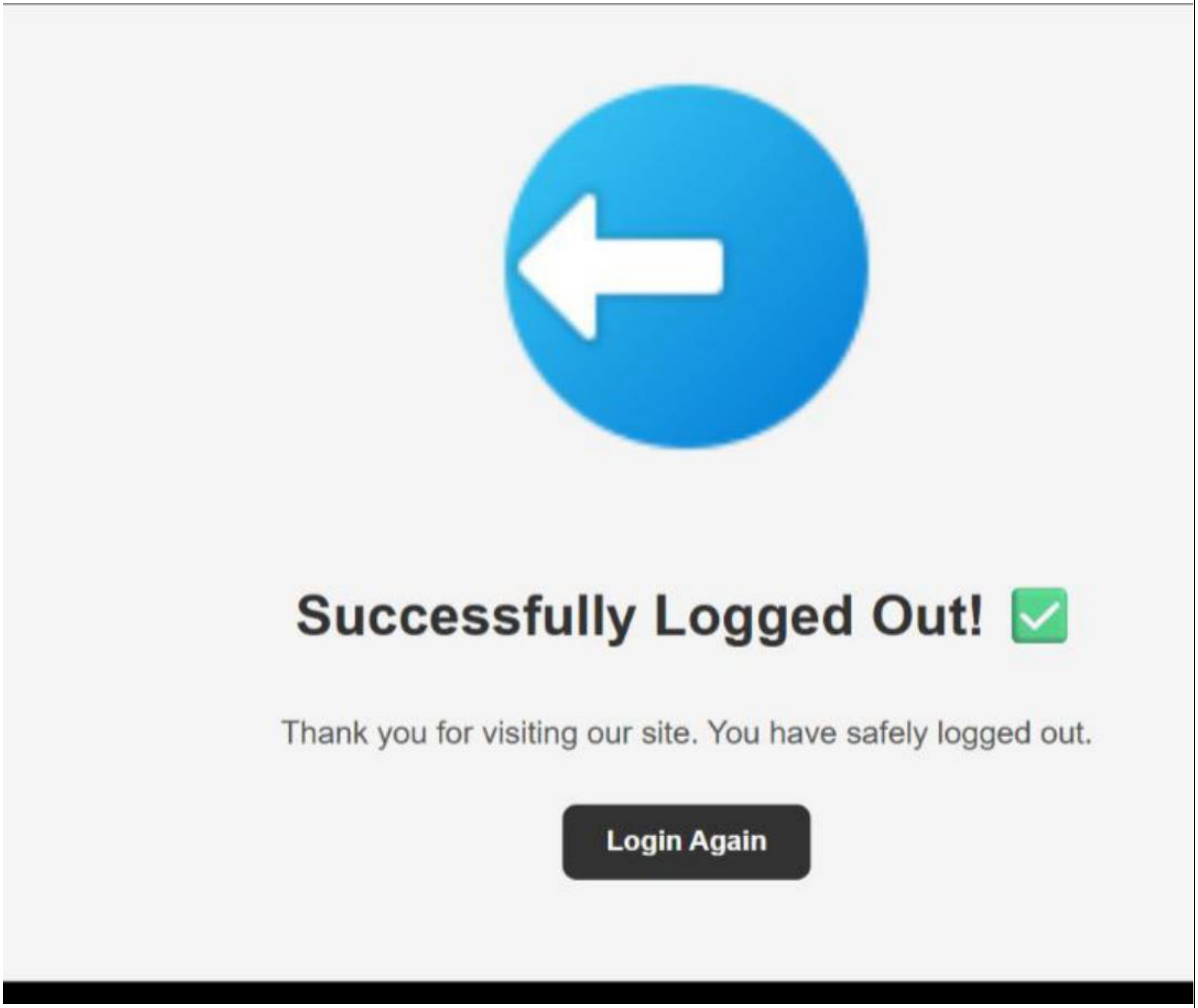
Profile Image

user

Email. user@example.com

Role: Student

Logout page



Successfully Logged Out! ✅

Thank you for visiting our site. You have safely logged out.