

COLLEGE CODE : 9111

COLLEGE NAME : SRM Madurai College For Engineering And  
Technology

DEPARTMENT : B.E Computer Science and Engineering

STUDENT NM-ID : 373871F4FB98BB599D44EACC4F5D2DA8

REGISTER NO : 911123104046

DATE : 06/10/2025

Completed the project named as : IBM-FE-Dynamic Image Slider

SUBMITTED BY,

NAME :Sanjay Kanna K J

MOBILE NO : 8668089144

## 1. Final Demo Walkthrough

Script:

"Hello everyone, this is my final project demonstration: the Dynamic Image Sliding application.

The primary goal of this project was to build a modern, responsive, and easily manageable image slider.

1. Core Slider Functionality: As you can see on the screen, the slider displays a series of images.

- I can navigate manually using the 'Next' and 'Previous' arrow buttons.
- We can also jump to a specific image using these navigation dots at the bottom.
- The slider is set to autoplay, smoothly transitioning to the next slide every 5 seconds. Hovering over the slider pauses this autoplay feature.

2. Dynamic Content: The most important feature is that it's dynamic. The images aren't hardcoded in the

HTML. Instead, they are fetched from an external data source, like a JSON file or an API.

- (Show the JSON file or code snippet) Here is the data source. If I want to add a new image or change the order, I simply update this file. The slider will automatically reflect the changes on reload, requiring no changes to the core code. This makes it incredibly easy to maintain.

3. Responsiveness: The entire component is fully responsive.

- (Resize the browser window from desktop to mobile size) Notice how the slider and the images automatically adjust to fit the screen size perfectly, ensuring a great user experience on any device, from a large monitor to a mobile phone.

In conclusion, the Dynamic Image Sliding project is a flexible and efficient solution for showcasing images on a website. Thank you.

## 2. Project Report

Abstract

The Dynamic Image Sliding project is a front-end web application designed to display a sequence of images in an automated or manually controlled slideshow. Unlike traditional static sliders, this project dynamically fetches image data from an external source, allowing for easy updates and management without altering the source code. The application is built with modern web technologies, focusing on responsiveness, performance, and user experience.

## Features

- Dynamic Image Loading: Images are loaded from a JSON configuration file or an API endpoint.
- Auto-Play: Slides transition automatically at a set interval.
- Manual Navigation: Users can navigate using next/previous arrow buttons and pagination dots.
- Pause on Hover: The automatic slideshow pauses when the user's mouse is over the slider.
- Fully Responsive: The layout seamlessly adapts to all screen sizes, including desktop, tablet, and mobile.
- Smooth Animations: Utilizes CSS transitions for fluid and performant sliding animations.

## Technology Stack

- Languages: HTML5, CSS3, JavaScript (ES6+)
- Framework/Library: (e.g., React, Vue.js, or Vanilla JavaScript)
- Styling: CSS Flexbox/Grid for layout, CSS Transitions for animations.
- Data Handling: Fetch API for retrieving image data from a JSON file.

## 3. Screenshots / API Documentation and Code

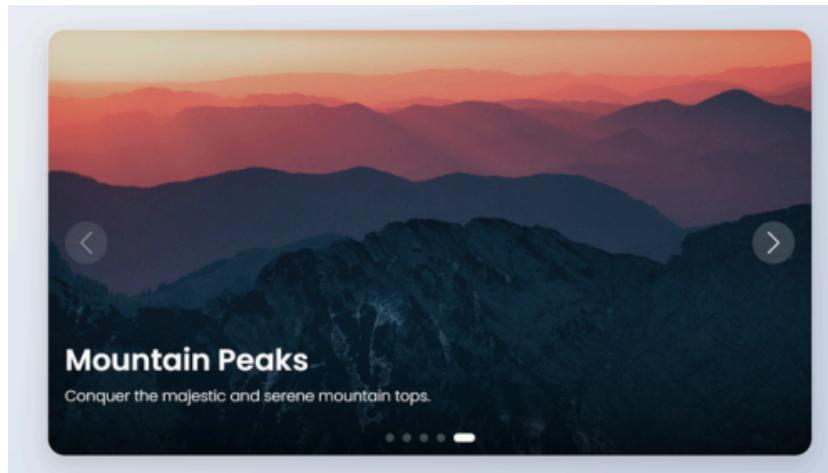
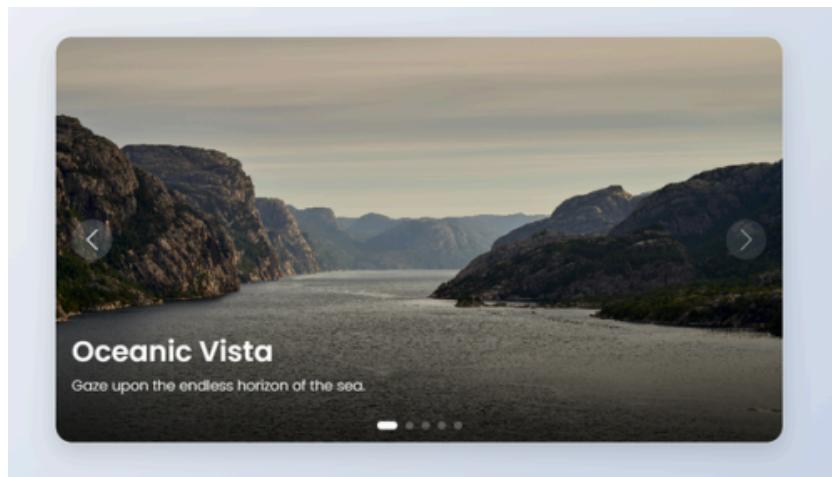
```
<<button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleAutoplaying"
data-bs-slide="prev">
<span class="carousel-control-prev-icon" aria-hidden="true"></span>
<span class="visually-hidden">Previous</span>
</button>
<button class="carousel-control-next" type="button" data-bs-target="#carouselExampleAutoplaying"
data-
bs-slide="next">
<span class="carousel-control-next-icon" aria-hidden="true"></span>
<span class="visually-hidden">Next</span>
</button>
</div>
```

```
<div id="carouselExampleAutoplaying" class="carousel slide" data-bs-ride="carousel">
<div class="carousel-inner">
<div class="carousel-item active">

</div>
<div class="carousel-item">

</div>
<div class="carousel-item">

</div>
</div>
```



## 4. Challenges & Solutions

During the development of the Dynamic Image Sliding project, several challenges were encountered.

- Challenge 1: Asynchronous Image Loading
  - Problem: The slider script would sometimes execute before the images were fully fetched and loaded from the external source, causing errors or a blank display.
  - Solution: This was solved by using JavaScript Promises with the `async/await` syntax. The main slider initialization logic was placed inside an `async` function and made to await the `fetch` call's completion. This ensures that the slider only tries to render the images after the data has been successfully retrieved.
- Challenge 2: Smooth and Performant Animations
  - Problem: Early attempts at animation using JavaScript to manipulate CSS properties (e.g., `element.style.left`) resulted in choppy or stuttering animations, especially on less powerful devices.
  - Solution: The animation logic was refactored to use CSS Transitions on the `transform: translateX()` property. Offloading the animation to the browser's CSS engine leverages hardware acceleration, resulting in significantly smoother and more performant transitions.
- Challenge 3: Maintaining the Correct Slide State
  - Problem: It was difficult to keep track of the currently active slide, especially when handling both automatic and manual navigation simultaneously.
  - Solution: A state variable, `currentIndex`, was introduced to always hold the index of the active slide. All functions that change the slide (e.g., `goToNextSlide`, `goToPrevSlide`, `jumpToSlide`) were designed to only update this variable. A single `updateSlider()` function reads `currentIndex` and applies the necessary CSS transform, creating a single source of truth and preventing state conflicts.

## 5. GitHub README & Setup Guide

### Dynamic Image Slider

A modern, responsive, and easy-to-use image slider that dynamically loads images from an external data source.

#### ⭐ Features

**Dynamic Content:** Loads image data from a JSON file

**Responsive Design:** Works seamlessly on desktop, tablets, and mobile devices.

**Auto-Play:** Slides transition automatically with a configurable interval.

**Manual Controls:** Includes Next/Previous buttons and pagination dots for easy navigation.

**Pause on Hover:** Autoplay pauses when the user hovers over the slider.

#### 🛠 Technologies Used

HTML5

CSS3

JavaScript (ES6+)

## 6. Final Submission

### Demo Output Link:

[https://drive.google.com/file/d/1vtEdeo2Dwm0Gqx\\_h9Tk2rgs1Iniw3Ppd/view](https://drive.google.com/file/d/1vtEdeo2Dwm0Gqx_h9Tk2rgs1Iniw3Ppd/view)

### Repository Link:

| NAME                   | GITHUB ID   |
|------------------------|---|
| Vigneshwaran J S       | <a href="https://github.com/vigneshwaran2830s/Dynamic-Image-Slider-NM-">https://github.com/vigneshwaran2830s/Dynamic-Image-Slider-NM-</a> |
| Prasanth Narayanan V S | <a href="https://github.com/prasanthjoueur/NaanMudhalvan">https://github.com/prasanthjoueur/NaanMudhalvan</a>                             |
| Jagadish J             | <a href="https://github.com/Sachin19-jaga/naanmudhalvan.git">https://github.com/Sachin19-jaga/naanmudhalvan.git</a>                       |
| Sanjay Kanna K J       | <a href="https://github.com/Sanjaykanna1995/NM-Dynamic-Image-Slider">https://github.com/Sanjaykanna1995/NM-Dynamic-Image-Slider</a>       |
| Kirubananthan M        | <a href="https://github.com/kirubananthanM/NaanMudhalvan">https://github.com/kirubananthanM/NaanMudhalvan</a>                             |