COLLEGE CODE : 9111

COLLEGE NAME : **SRM Madurai College For Engineering**

**And Technology**

DEPARTMENT : **B.E Computer Science and Engineering**

STUDENT NM-ID : **5AD5D2FE0760609B6F048D417B3D393E**

REGISTER NO : **911123104026**

DATE : **15/09/2025**

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

**SUBMITTED BY,**

**NAME : Kirubananthan M**

**MOBILE NO : 97875 89869**

## Problem Statement :

In the current era of digital platforms, engaging users visually has become essential. Static images on websites are no longer sufficient to capture attention. The IBM-FE Dynamic Image Slider project aims to solve this by building an interactive, responsive, and dynamic slider that integrates seamlessly with a Node.js REST API.

The problem arises from the need to handle large image sets dynamically, manage performance, and provide smooth transitions without compromising load speed. By addressing these challenges, the slider ensures an engaging user experience while allowing easy backend integration.

This also helps businesses showcase products, portfolios, or advertisements effectively. The project emphasizes **scalability, modularity, and usability** across multiple devices and platforms.

## Users & Stakeholders

**Users:**

1. End Users – Individuals browsing a website who will interact with the slider for viewing content.

2. Business Clients – Organizations that will use the slider to display promotional content, banners, or product images.

3. Developers – Engineers responsible for integrating and customizing the slider into applications.

**Stakeholders:**

- UI/UX Designers – Ensure the slider is visually appealing and user-friendly.

- Product Managers – Define requirements and prioritize features.

- Marketing Teams – Use the slider for promotions and campaigns.

- Backend Developers – Maintain the Node.js REST API and ensure smooth data flow.

- QA Engineers – Test functionality, performance, and responsiveness.

## User Stories

1. As a user, I want to navigate through images easily using arrows, swipe gestures, or auto-play, so that I can view content seamlessly without manual effort.

2. As a user, I want the slider to be responsive so that I can view images properly across desktop, tablet, and mobile.

3. As a developer, I want the slider to fetch images dynamically from a Node.js API so that updates can be managed without redeployment.

4. As a business owner, I want to highlight promotional content using the slider so that users are attracted to the latest offerings.

5. As a designer, I want to customize the slider's transitions, themes, and animations so that it matches the website's branding.

6. As an accessibility advocate, I want keyboard navigation and alt text enabled so that differently-abled users can also use the slider.

## MVP Features

**Functional Features:**

- Responsive slider design that adapts to different devices.

- Dynamic loading of images via Node.js REST API.

- Auto-play feature with configurable time intervals.

- Manual navigation using arrows, dots, and swipe gestures.

- Smooth CSS/JS-based transitions.

- Accessibility compliance (keyboard navigation, alt attributes).

**Non-Functional Features:**

- High performance with minimal load times.

- Scalability to support large sets of images.

- Cross-browser compatibility.

- Secure API integration to handle image data safely.

## Wireframes / API Endpoint List

**Wireframe Concept:**

- A rectangular image display area with left and right arrows for navigation.

- Small dots at the bottom indicating the position of the current slide.

- Option for autoplay with pause on hover.

**API Endpoints:**

1. **GET /api/images** → Returns a JSON list of image URLs and metadata.
   Example Response: { 'id':1, 'url':'/images/img1.jpg', 'title':'Image 1' }

2. **POST /api/images** → Allows uploading a new image with metadata.
   Example Input: { 'url':'/images/img2.jpg', 'title':'Image 2' }

3. **PUT /api/images/:id** → Updates image details such as title or description.

4. **DELETE /api/images/:id** → Deletes an image by ID.

The endpoints ensure **CRUD functionality** for managing images dynamically.


## Acceptance Criteria

1. The slider must fetch and display images dynamically from the Node.js API without manual refresh.

2. It should work on all major devices (desktop, tablet, mobile) with proper responsiveness.

3. Auto-play should transition slides smoothly within the configured interval.

4. Users must be able to pause, navigate forward, and backward using buttons or gestures.

5. API must handle errors gracefully (e.g., missing or broken links should not crash the slider).

6. Accessibility standards must be met (alt text, ARIA roles, keyboard navigation).

7. Performance benchmarks: The slider must load within 2 seconds and handle at least 50 images efficiently.

8. Cross-browser testing: Functionality must be consistent across Chrome, Firefox, Safari, and Edge.