# PERSONAL BLOG ON IBM CLOUD STATIC WEB APPS

**Development part – 2:**

To make an interactive personal travel blog, we can add various features using

HTML, CSS, and some JavaScript. Here are some features along with code

snippets,

**1.Image Sliders:**

Creating an image slider to showcase the travel photos.

**#HTML**

<div class="slider">

<img src="paris1.jpg" alt="Paris">

<img src="paris2.jpg" alt="Paris">

<img src="paris3.jpg" alt="Paris">

</div>

**#CSS**

.slider{

display: flex;

overflow: hidden;

}

.slider img {

width: 100%;

transition: transform 0.5s;

}

**#JavaScript**

<script>

const slider = document.querySelector('.slider');

let counter = 1;

setInterval(() => {

slider.style.transition = 'transform 0.5s ease-in-out';

slider.style.transform = `translateX(${-counter \* 100}%)`;

counter++;

if (counter === 3) {

setTimeout(() => {

slider.style.transition = 'none';

slider.style.transform = 'translateX(-100%)';

counter = 1;

}, 500);

}

}, 3000);

</script>

**Explanation:**

The provided code is for creating a simple image slider or carousel. The main goal of the program is to display a sequence of images one after the other in a sliding fashion, creating an automatic slideshow effect. the program's main goal is to create a simple image slider that automatically cycles through a set of images, providing a continuous slideshow with smooth transitions.

**2. Interactive Maps:**

Adding Embed interactive maps using services like Google Maps or Mapbox.

**#HTML**

<div id="map"></div>

**#CSS**

#map {

height: 300px;

width: 100%;

}

**#JavaScript**

<script>

function initMap() {

const location = { lat: 48.8588443, lng: 2.2943506 };

const map = new google.maps.Map(document.getElementById('map'), {

center: location,

zoom: 14

});

const marker = new google.maps.Marker({ position: location, map: map });

}

</script>

<script async defer

src="https://maps.googleapis.com/maps/api/js?key=YOUR\_API\_KEY&callback=ini

tMap"></script>

**Explanation:**

The main goal of the provided program is to embed an interactive map into a web page using the Google Maps JavaScript API. the program's main goal is to integrate and display an interactive Google Map on a web page, allowing users to view a specific location and interact with the map features, such as zooming in and out and viewing a marker at the specified location.

**3. Image Lightbox:**

It allows users to view images in a larger, interactive lightbox when they click on

them.

**#HTML**

<img src="paris.jpg" alt="Paris" class="lightbox-trigger">

**#CSS**

.lightbox {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(0, 0, 0, 0.8);

text-align: center;

}

.lightbox img {

max-width: 80%;

max-height: 80%;

margin: 10% auto;

}

**#JavaScript**

<script>

const lightboxTrigger = document.querySelector('.lightbox-trigger');

const lightbox = document.createElement('div');

lightbox.className = 'lightbox';

lightboxTrigger.addEventListener('click', () => {

lightbox.innerHTML = `<img src="${lightboxTrigger.src}"

alt="${lightboxTrigger.alt}">`;

document.body.appendChild(lightbox);

lightbox.style.display = 'block';

});

lightbox.addEventListener('click', () => {

lightbox.style.display = 'none';

});

</script>

**Explanation:**

The main goal of the provided code is to create a simple lightbox effect for viewing images in a larger and interactive manner when a user clicks on them. A lightbox is a common user interface element used to display images or other media in a popup or modal window, often with an overlay to darken the background. this code creates a basic lightbox functionality that displays an enlarged version of an image when the user clicks on it, and it allows the user to close the lightbox by clicking anywhere on it. It's a common pattern for presenting images in a user-friendly and interactive way.

4. Social Media Sharing Buttons:

Adding social media sharing buttons to our blog posts.

**#HTML**

<div class="share-buttons">

<a

href="https://www.facebook.com/share?url=YOUR\_BLOG\_POST\_URL">Facebook

</a>

<a href="https://twitter.com/share?url=YOUR\_BLOG\_POST\_URL&text=Check

out this awesome travel blog!">Twitter</a>

</div>

**#CSS**

.share-buttons {

margin-top: 10px;

}

.share-buttons a {

margin-right: 10px;

text-decoration: none;}

**Explanation:**

The provided code demonstrates how to add social media sharing buttons to blog posts, allowing users to easily share the content on platforms like Facebook and Twitter. The result is a set of social media sharing buttons (Facebook and Twitter in this case) that users can click to easily share the specified blog post on their social media profiles. The URLs in the `href` attributes direct users to the sharing mechanisms on Facebook and Twitter, with pre-filled information about the blog post. The styling in the CSS section adds some spacing and removes underlines from the links. You can further customize the appearance and add more sharing options as needed.

5. Comments Section:

It allows users to leave comments on our blog posts.

**#HTML**

<div class="comments">

<h3>Comments</h3>

<div class="comment">

<p>Great post! I love Paris.</p>

</div>

<!-- Add more comments here -->

</div>

**#CSS**

.comments {

margin-top: 20px;

}

.comment {

border: 1px solid #ccc;

padding: 10px;

margin-bottom: 10px;

**Explanation:**

The provided code is a basic HTML and CSS structure for allowing users to leave comments on blog posts. provides a basic structure for adding comments to a blog post. It includes a comments section where users can read and add their comments. You can customize the styling and structure further, and typically, for a live website, you would incorporate back-end logic to handle user input and store the comments in a database.

**Conclusion**:

Static web apps in the cloud offer a number of advantages over traditional dynamic web apps. They are typically faster, more scalable, more secure, and more cost-effective. Static web apps in the cloud are a good choice for a variety of websites and web applications.