Footer:

<!DOCTYPE html>

<html>

<head>

<title>Your Page Title</title>

<link rel="stylesheet" type="text/css" href="//fonts.googleapis.com/css?family=Open+Sans">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css"> <!-- Include Font Awesome CSS -->

<style>

/\* CSS styling for the blue footer \*/

.blue-footer {

background-color: #004A94;

color: white;

padding: 10px;

display: flex;

justify-content: space-between;

align-items: center;

position: fixed;

bottom: 0;

left: 0;

width: 100%;

}

/\* Styling for the footer logo \*/

.footer-logo img {

height: 30px; /\* Adjust the height as needed \*/

}

/\* Styling for the footer arrow icons \*/

.footer-buttons i {

font-size: 20px;

margin: 5px;

cursor: pointer;

}

/\* Additional styles for the icons if needed \*/

.footer-buttons i:hover {

color: #007BFF;

}

.footer-buttons button {

background-color: #007BFF;

color: white;

border: none;

padding: 10px 20px;

border-radius: 5px;

margin: 5px;

cursor: pointer;

}

/\* Additional styles for the buttons if needed \*/

.footer-buttons button:hover {

background-color: #0056b3;

}

.footer-buttons button:active {

background-color: #003d80;

}

/\* CSS styling for the page content to push it above the footer \*/

.page-content {

padding-bottom: 70px; /\* Adjust the padding value to match the height of the footer \*/

}

</style>

</head>

<body>

<!-- Page content here -->

<div class="page-content">

<!-- Your page content goes here -->

</div>

<footer class="blue-footer">

<div class="footer-logo">

<img src="https://retireone.com/wp-content/uploads/2019/01/logo\_allianz-tr\_gray.png" alt="Logo">

</div>

<div class="footer-buttons">

<i class="fas fa-arrow-left"></i> <!-- Font Awesome arrow left icon -->

<i class="fas fa-arrow-right"></i> <!-- Font Awesome arrow right icon -->

<button class="save-button">Save</button>

</div>

</footer>

<!-- Additional scripts or other content -->

</body>

</html>

**REACT:**

To create the equivalent React code for the provided HTML, you can follow these steps:

1. Set up a new React project or use an existing one.
2. Create a functional component for the page content.
3. Create a functional component for the footer.
4. Import the necessary CSS and Font Awesome library.
5. Use the components within the main App component.

Here's the React code:

1. Install Font Awesome: First, install the Font Awesome library using npm or yarn.

bash

npm install @fortawesome/fontawesome-free

1. Create the React components:

jsx

// Import necessary libraries

import React from 'react';

import './App.css'; // Import your CSS file

// Create the PageContent component

const PageContent = () => {

return (

<div className="page-content">

{/\* Your page content goes here \*/}

</div>

);

};

// Create the Footer component

const Footer = () => {

return (

<footer className="blue-footer">

<div className="footer-logo">

<img src="https://retireone.com/wp-content/uploads/2019/01/logo\_allianz-tr\_gray.png" alt="Logo" />

</div>

<div className="footer-buttons">

<i className="fas fa-arrow-left"></i> {/\* Font Awesome arrow left icon \*/}

<i className="fas fa-arrow-right"></i> {/\* Font Awesome arrow right icon \*/}

<button className="save-button">Save</button>

</div>

</footer>

);

};

// Create the App component

function App() {

return (

<div className="App">

<PageContent />

<Footer />

</div>

);

}

export default App;

1. Add the CSS:

Make sure you have the CSS styles from the provided HTML in your App.css file.

1. Include Font Awesome Icons:

To use Font Awesome icons, you need to import the Font Awesome library and add it to your project's main index.js file.

jsx

// index.js

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

import { library } from '@fortawesome/fontawesome-svg-core';

import { fas } from '@fortawesome/free-solid-svg-icons';

import './index.css';

library.add(fas);

ReactDOM.render(

<React.StrictMode>

<App />

</React.StrictMode>,

document.getElementById('root')

);

With this setup, you should have a React app that mimics the behavior of the provided HTML page, including the blue footer with the Allianz logo and navigation buttons. Don't forget to replace the sample logo URL with your actual logo image source.

**---------------------**

BUFFERING: To combine the provided HTML code with the buffering page code and JavaScript for network status checking, you can follow these steps:

1. Create the buffering page HTML and CSS as mentioned in the previous answer.
2. Add the provided HTML code inside the main-content div in the main.html file.
3. Modify the JavaScript code in main.html to include network status checking and show/hide the buffering page accordingly.

Here's the modified main.html file:

html

<!DOCTYPE html>

<html>

<head>

<title>Your Page Title</title>

<link rel="stylesheet" type="text/css" href="featurestab.css">

<link rel="stylesheet" type="text/css" href="//fonts.googleapis.com/css?family=Open+Sans">

</head>

<body>

<!-- JavaScript code to check the network status and show/hide the buffering page -->

<script>

// Function to check the network status

function checkNetworkStatus() {

return navigator.onLine; // Returns true if online, false if offline

}

// Function to show the buffering page

function showBufferingPage() {

document.getElementById("buffering-page").style.display = "block";

}

// Function to hide the buffering page and show the main content

function showMainContent() {

document.getElementById("buffering-page").style.display = "none";

document.getElementById("main-content").style.display = "block";

}

// Function to check the network status and display the buffering page if offline

function checkAndLoad() {

if (!checkNetworkStatus()) {

// If offline, show the buffering page

showBufferingPage();

} else {

// If online, show the main content

showMainContent();

}

}

// Event listener to detect when the main content is loaded

window.addEventListener("load", checkAndLoad);

</script>

<!-- Buffering page (loading page) -->

<iframe id="buffering-page" src="buffering.html" style="display: none; width: 100%; height: 100vh; border: none;"></iframe>

<!-- Main content -->

<div id="main-content" style="display: none;">

<!-- Place your provided HTML code here -->

<!-- ... your provided HTML code ... -->

</div>

</body>

</html>

Note that in this modified main.html file, the provided HTML code is placed inside the main-content div, which will be hidden initially. The JavaScript code checks the network status using the navigator.onLine property. If the user is offline, the buffering page will be shown, and if the user is online, the main content will be displayed.

Now, when the main.html page is loaded, it will automatically check the network status and display the buffering page if necessary. Once the network status is confirmed to be online, the main content will be displayed, and users can access the features without issues.