**Notes**

1. The CDN link is used to get font and image to optimize the overall code.
2. I have used image tag for setting the image on full page.
3. The page is responsive on all possible devices.
4. I have used Java Script click event to show the alert pop-up

Below sections depicts the code functionalities.

**HTML Full Page Component**

* It consists of a div with the class .full-page-component, which acts as the container for the entire component.
* Inside this container, you have:
  + A header element with the class .page-header, containing the site's header content, including a navigation menu.
  + A section element that holds the background image, overlay, and content of the main section.
  + An img element with the class .background-image that displays the background image.
  + A footer element with the class .page-footer, containing the site's footer content.

**CSS (style/style.css):**

* The CSS file defines the styling of full-page component.

**General Styles:**

* Resets default margin, padding, and sets a consistent font family (Nunito Sans).
* box-sizing is set to content-box to ensure consistent box sizing.
* line-height and font-size are set for the entire document.
* The overflow: hidden property ensures that no scrollbars appear on the page.

**.full-page-component:**

* Defines the main container that covers the entire viewport (100vw and 100vh).
* Uses Flexbox to center its children vertically and horizontally.

.**page-header, .page-footer:**

* Styling for the header and footer sections.
* They have a background color with some transparency (rgba(0, 0, 0, 0.5)).
* Fixed height of 100px.
* Positioned using position: absolute; to stay at the top and bottom of the viewport.
* z-index is set to 152 to ensure they appear above other content.
* Font size and color are adjusted for better visibility.
* Content is centered within each section using Flexbox.

.**background-image:**

* Styles the background image within the section.
* Uses object-fit: cover; to make the image cover the entire section.
* Positioned absolutely to cover the entire section.

**.overlay:**

* Styles the translucent overlay above the background image.
* Positioned absolutely to cover the entire section.
* Uses rgba(0, 0, 0, 0.5) to create the semi-transparent effect.

**.content:**

* Styles the content within the section.
* Text is centered horizontally and has white color.
* Positioned relatively within the section.

**Button Styles:**

* Styles the "Click Here" button with a background color, padding, and cursor pointer.
* Changes background color on hover for interactivity.

**Media Query:**

* Contains styles for smaller screens (max-width: 575px).
* Changes the layout of .header-wrapper and .footer-wrapper to a vertical layout.
* Adjusts font size and spacing for better mobile readability.
* Adds some padding to the content within .content.

**JavaScript:**

* + The JavaScript code in HTML file selects the button with the id "clickButton" and adds a click event listener to it.
  + When the button is clicked, it triggers an alert with the message "Hello, world!"
  + The JavaScript also sets the content of an element with the id "year" to the current year using new Date().getFullYear().

Overall, this code creates a responsive full-page component with a header, footer, background image, and content section. It's designed to look good on both desktop and mobile devices. The provided CSS classes help organize and style the different parts of the component.