**Responsive Design**

To ensure the application works seamlessly on kiosks, tablets, and smartphones, we can use responsive design techniques. We'll use CSS Modules to write modular, reusable CSS code.

First, let's create a new CSS file called **styles.css** in the root of our project:

/\* styles.css \*/

In this example, we've defined some basic styles for our application, including a primary color, secondary color, background color, and text color. We've also defined a container class that will hold our application's content.

We've used CSS media queries to apply different styles based on the screen size. For example, on screens with a maximum width of 768px (e.g., tablets), we've reduced the margin and padding of the container. On screens with a maximum width of 480px (e.g., smartphones), we've further reduced the margin and padding.

**CSS Modules**

To use CSS Modules, we need to rename our **styles.css** file to **styles.module.css**. This tells Webpack to treat the file as a CSS module.

We can then import the CSS module in our JavaScript file like this:

import styles from './styles.module.css';

By using CSS Modules, we can write modular, reusable CSS code that's scoped to our component.

**TailwindCSS**

Alternatively, we can use TailwindCSS to write utility-first CSS code. TailwindCSS provides a set of pre-defined classes that we can use to style our application.

First, let's install TailwindCSS:

npm install tailwindcss

Next, let's create a new file called **tailwind.config.js** in the root of our project:

This configuration file tells TailwindCSS to use the "jit" mode, which allows us to use the **@apply** directive to apply classes to our components.

We can then import TailwindCSS in our JavaScript file like this:

By using TailwindCSS, we can write utility-first CSS code that's highly customizable and maintainable.

**Kiosk Mode**

To ensure our application works seamlessly on kiosks, we can use the **kiosk** mode in our CSS. We can add the following code to our **styles.module.css** file:

This code applies a full-screen layout to our application when the screen size is at least 1024x768 pixels.

**Tablet and Smartphone Support**

To ensure our application works seamlessly on tablets and smartphones, we can use the **responsive** design techniques we discussed earlier. We can add the following code to our **styles.module.css** file:

This code applies different layouts to our application based on the screen size.

By using these techniques, we can ensure our application works seamlessly on kiosks, tablets, and smartphones.