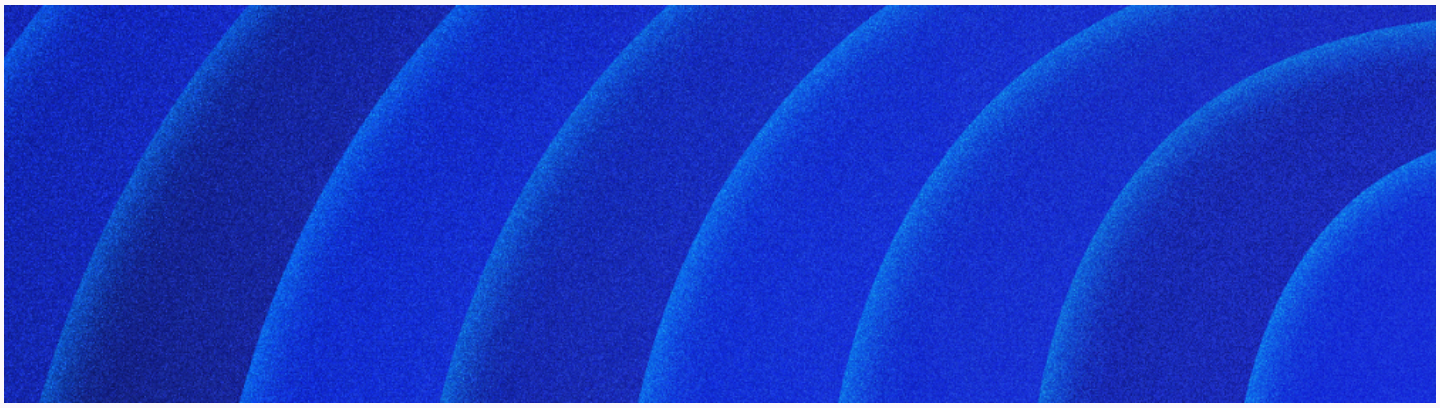


Dynamic Components

Date

Features of Dynamic Components

1. **Lazy Loading (Code Splitting):**
   Dynamic components are implemented with lazy loading, where the component is loaded only when the user navigates to the relevant page or feature.
2. **Client-Side Rendering:**
   Dynamic components are often rendered on the client side, meaning they load JavaScript code instead of HTML from the server and execute it on the client.
3. **Conditional Rendering:**
   You can render different components dynamically based on conditions.

Dynamic Components in Next.js

In Next.js, the `dynamic` function is used to make dynamic imports and enable lazy loading easily.

Example:

```
import dynamic from 'next/dynamic';


// Dynamically imported component

const DynamicComponent = dynamic(() =>
import('../components/MyComponent'), {

  loading: () => <p>Loading...</p>, // Optional loading UI

  ssr: false, // Disable server-side rendering

});


export default function Home() {
```

```
  return (
    <div>
      <h1>Dynamic Component
Example</h1>
      <DynamicComponent />
    </div>
  );
}
```

Now, I will explain how these components work in real-world applications and how they will function within a web application. I will implement everything in a real web application and demonstrate it with practical work.

I will create separate components and make them reusable so that they can be displayed consistently across different parts of the website. Additionally, I will build a dynamic component to display the product data dynamically to the users. This dynamic component will show all the product details.

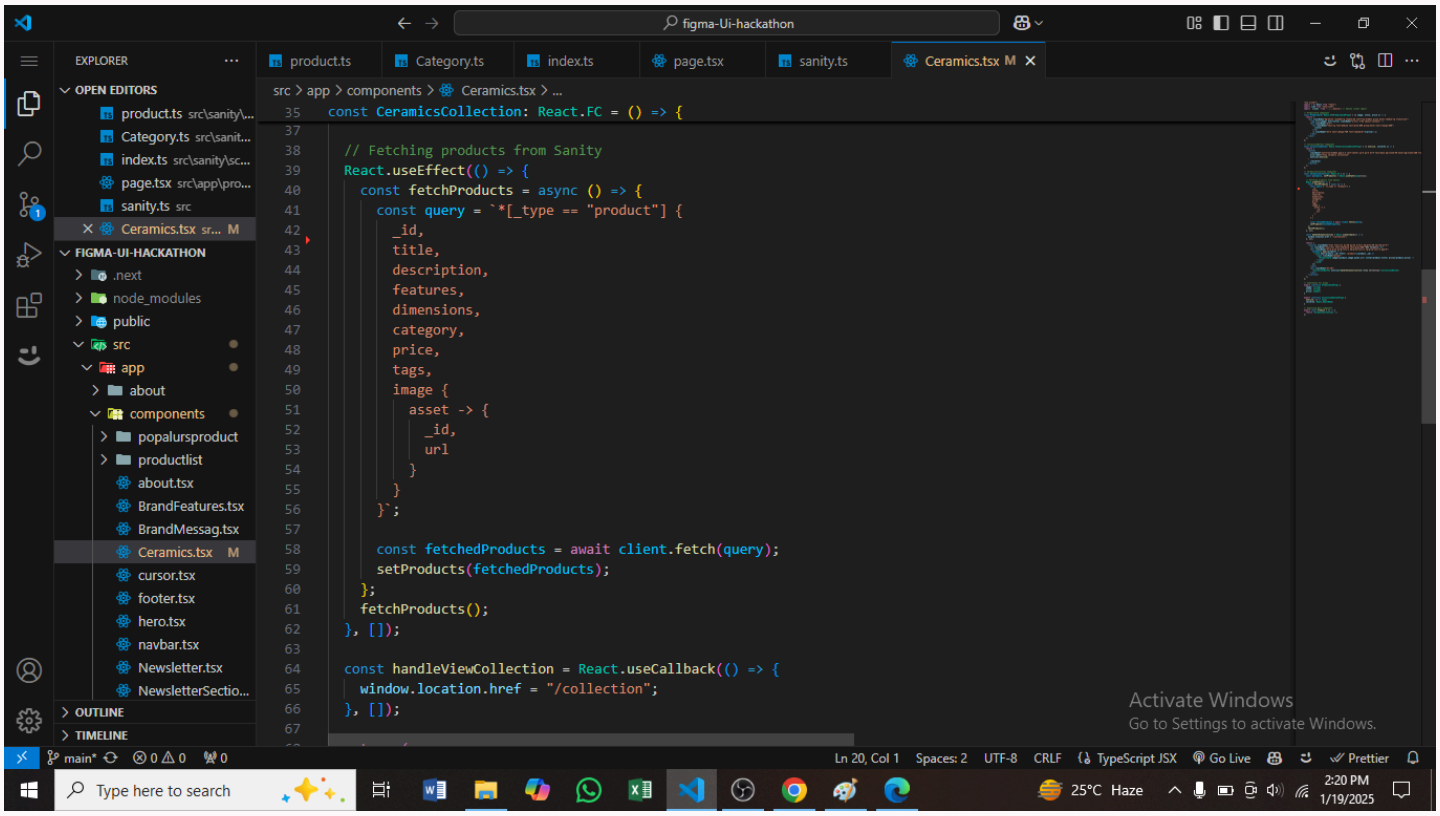
There will also be some reusable components in the pages that can be used in multiple areas of the application, ensuring a clean and efficient structure.

DynamicComponent:

In this component, the JavaScript code will load first, and the API fetching the data will be called dynamically. Once the data is fetched, it will be displayed on the website.

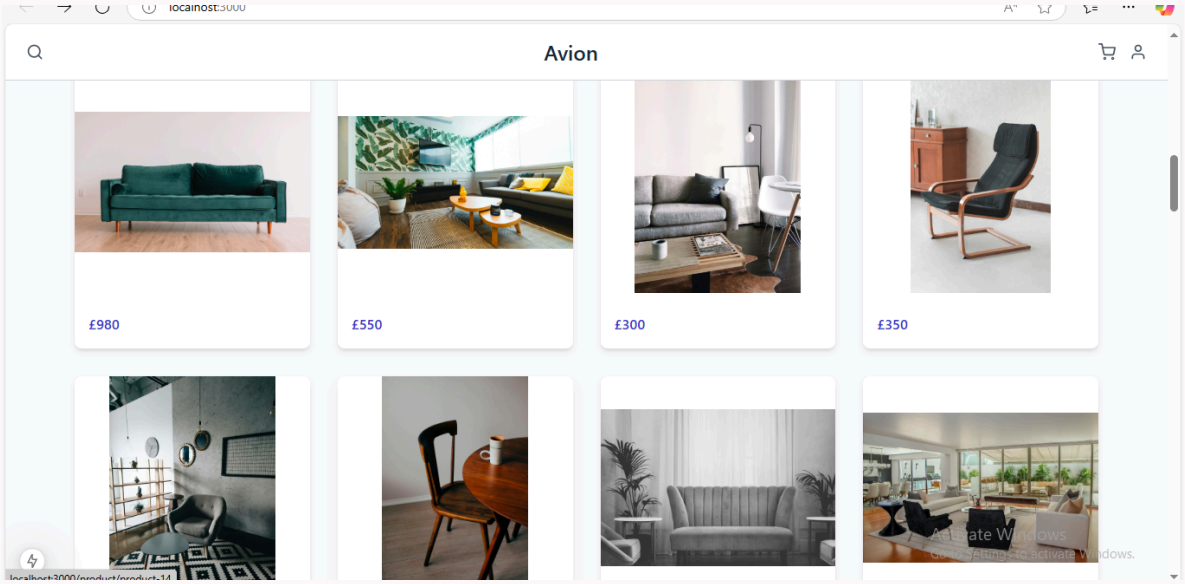
At the end of the code, we will include functionality to ensure that the data updates automatically every 20 seconds. This way, the data will refresh dynamically every 20 seconds, keeping the component updated in real-time.

By doing this, our components will remain dynamic and ensure fast reloads, providing data to the user quickly without delays, as it will be running dynamically in the background.

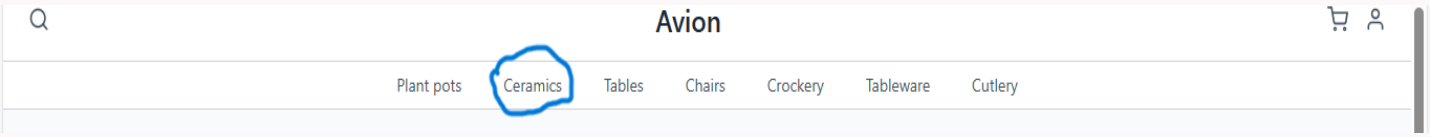


The ceramic components will work dynamically and display the results on the website. The data will be fetched and displayed dynamically, ensuring real-time updates and providing users with the latest information instantly.

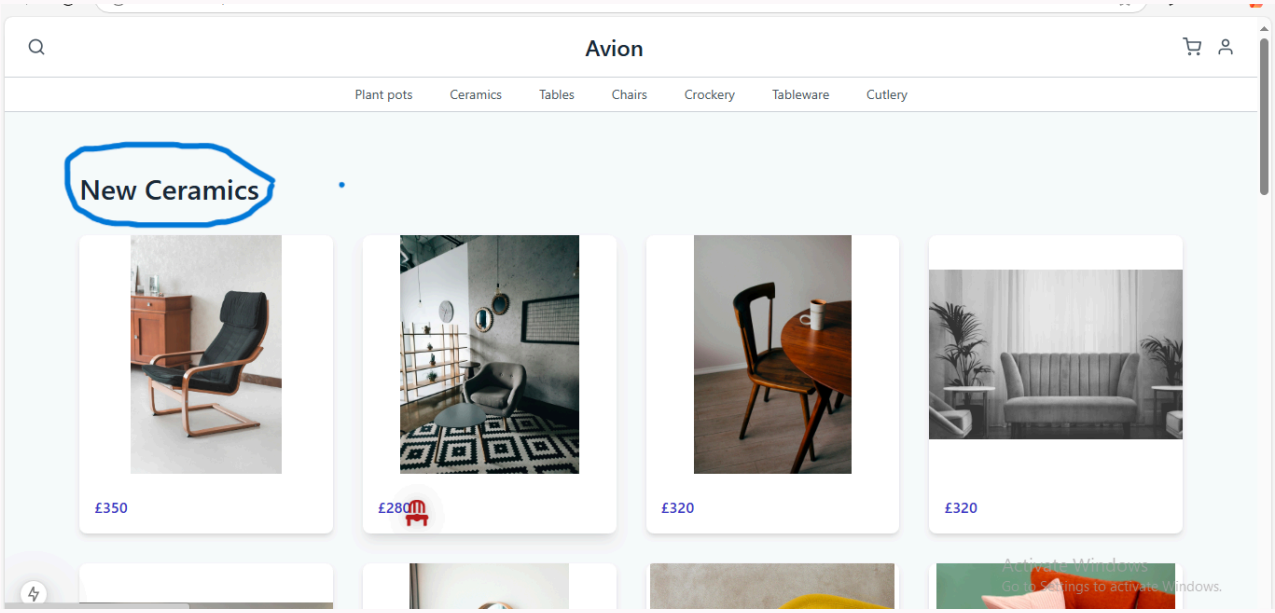
This approach makes the components highly responsive and allows the website to show the most up-to-date data without any delay, keeping the user experience smooth and interactive.



Components create

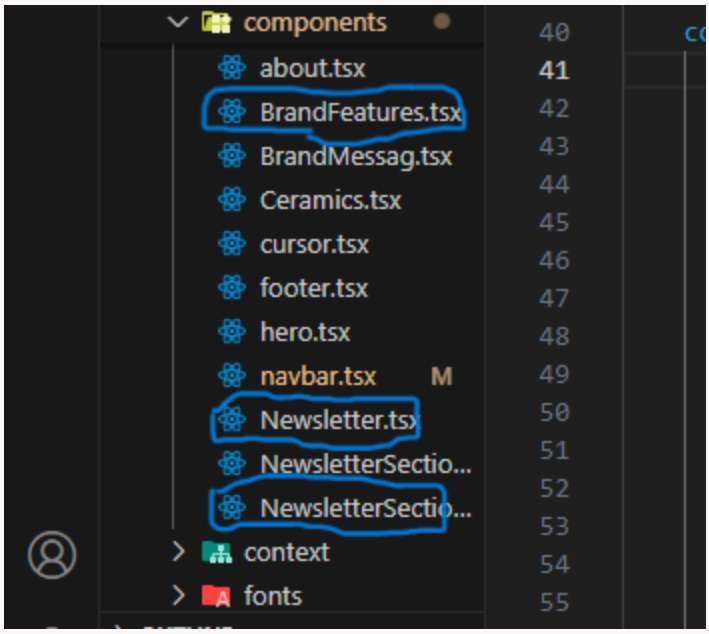


Now I will create a Ceramic component that will display our products. When users click on the Ceramic component, its data will appear in front of them. Similarly, the products inside the Ceramic category will also be displayed.



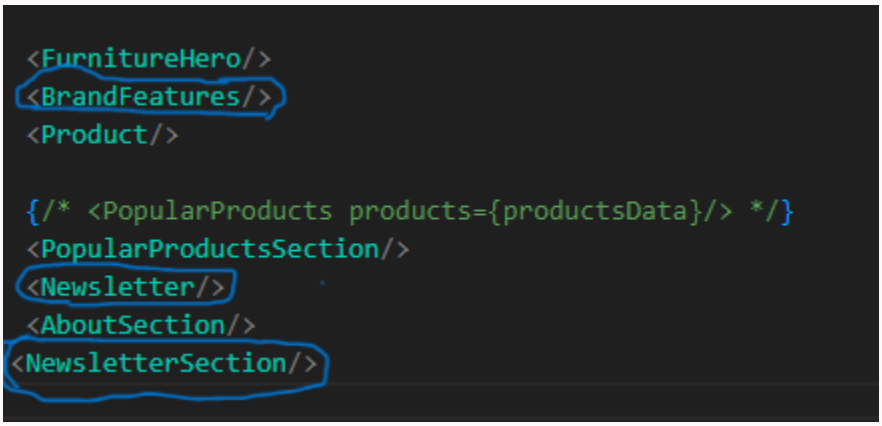
Reusable components

There are some components that, once created, can be reused everywhere. I will now create some reusable components that I can use across different parts of the application. These components will make the development process more efficient and allow me to maintain consistency throughout the project.

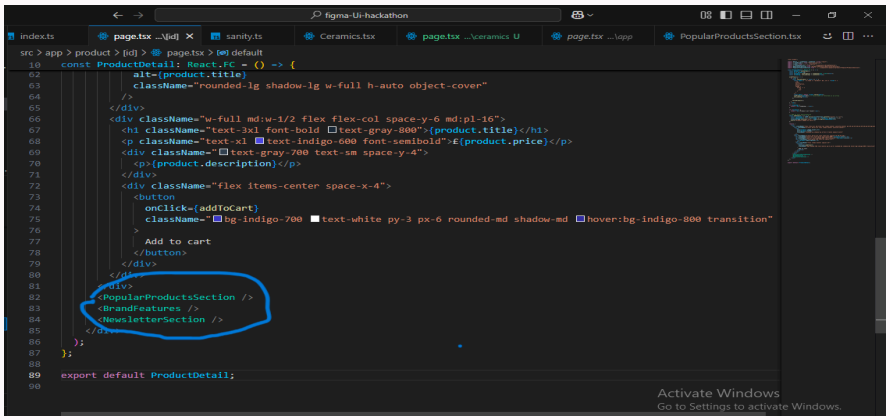


Now I have used these components in two places. Below are the pictures showing how they have been implemented.

1: First use

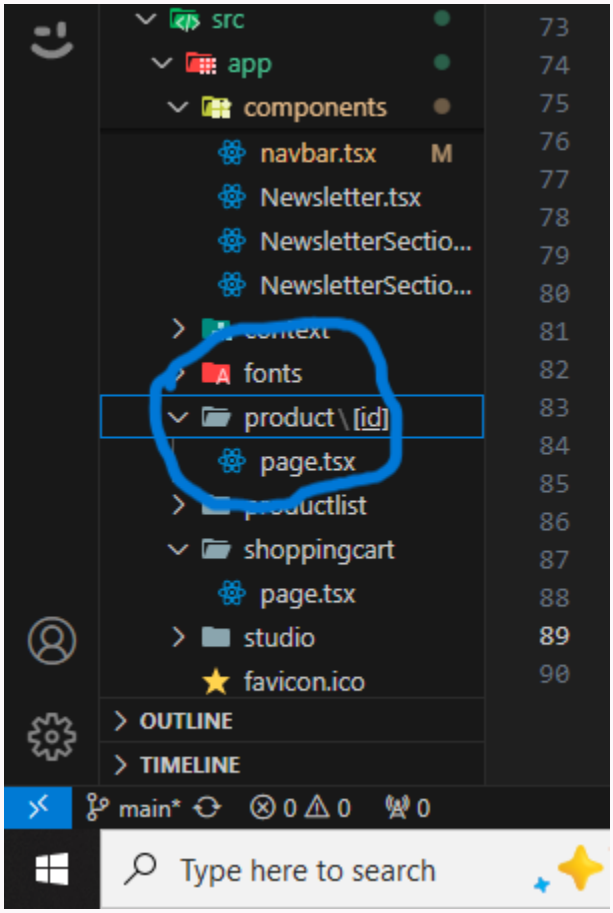


2: second use



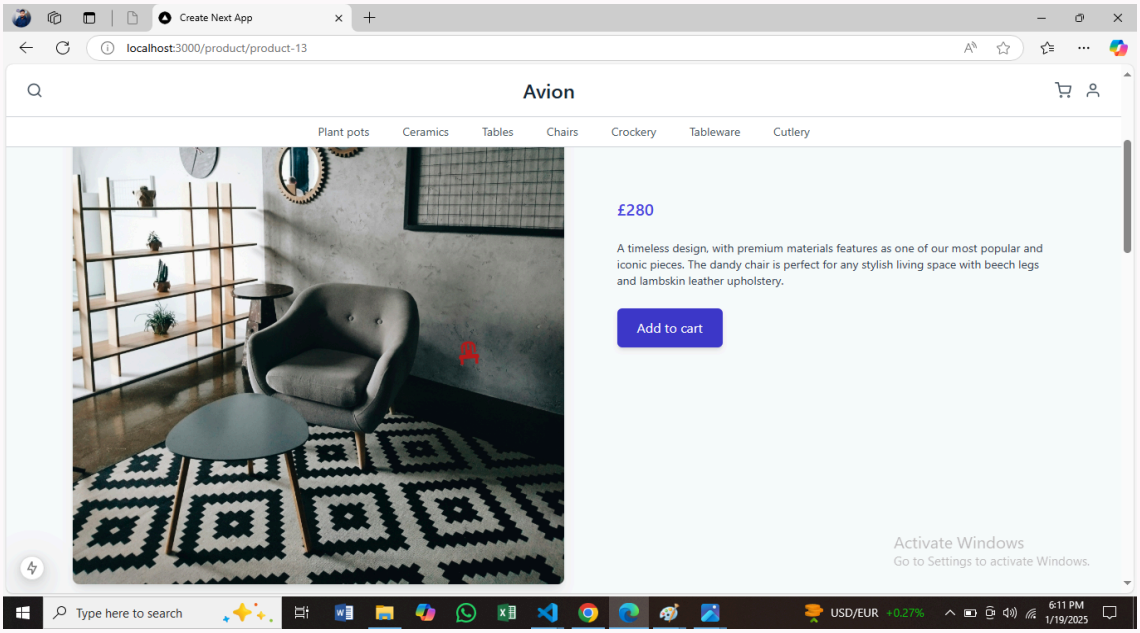
Create Dynamic

Now, these components will work in such a way that when we click on a product, its data will be dynamically fetched and displayed on the next page. The data for each product will be fetched dynamically and shown accordingly.



Now, let's check how it works dynamically. In the first image, we have our product, and when we click on it, in the second image, the product's data will be shown dynamically according to the product's details.

Second click work is dynamically



Thank you for your message, and it's noted: *Preferably Shahmeer Ali.*

Feel free to ask if you need anything else!