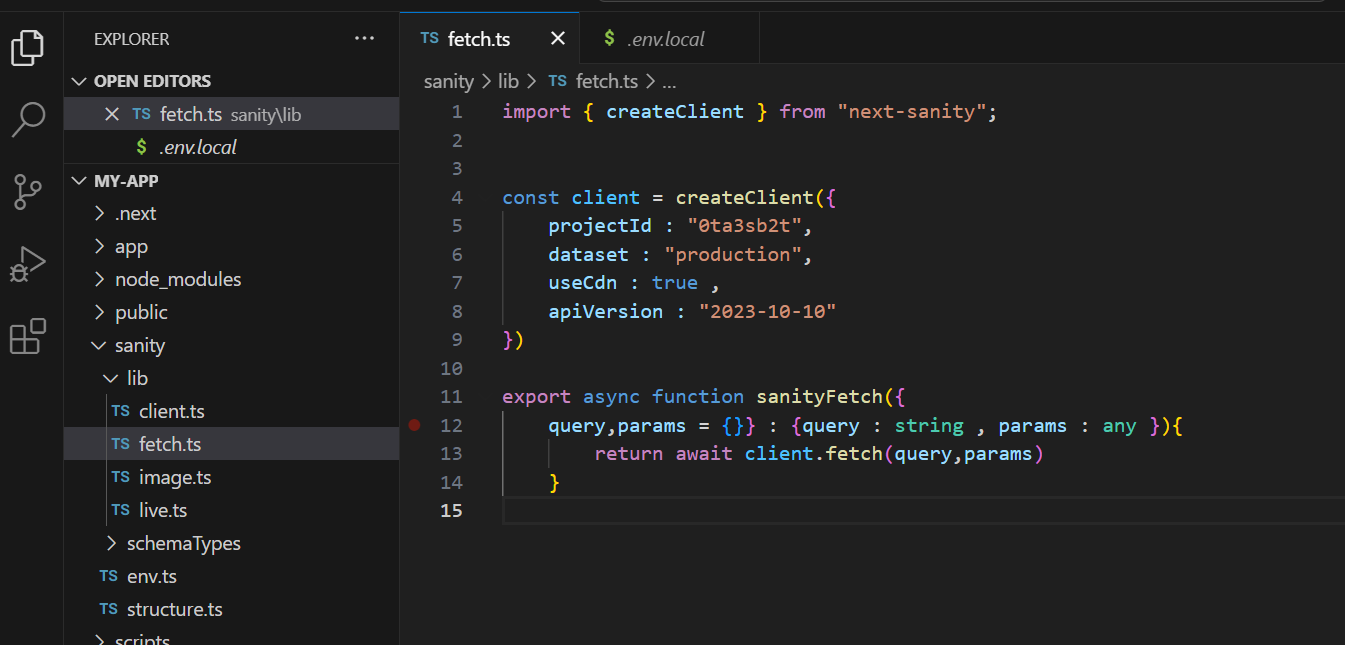
**Day 4 - Dynamic Frontend Components - [The Luxurious Marketplace]**

1. **STEPS TO FETCH DATA FROM SANITY TO THE**

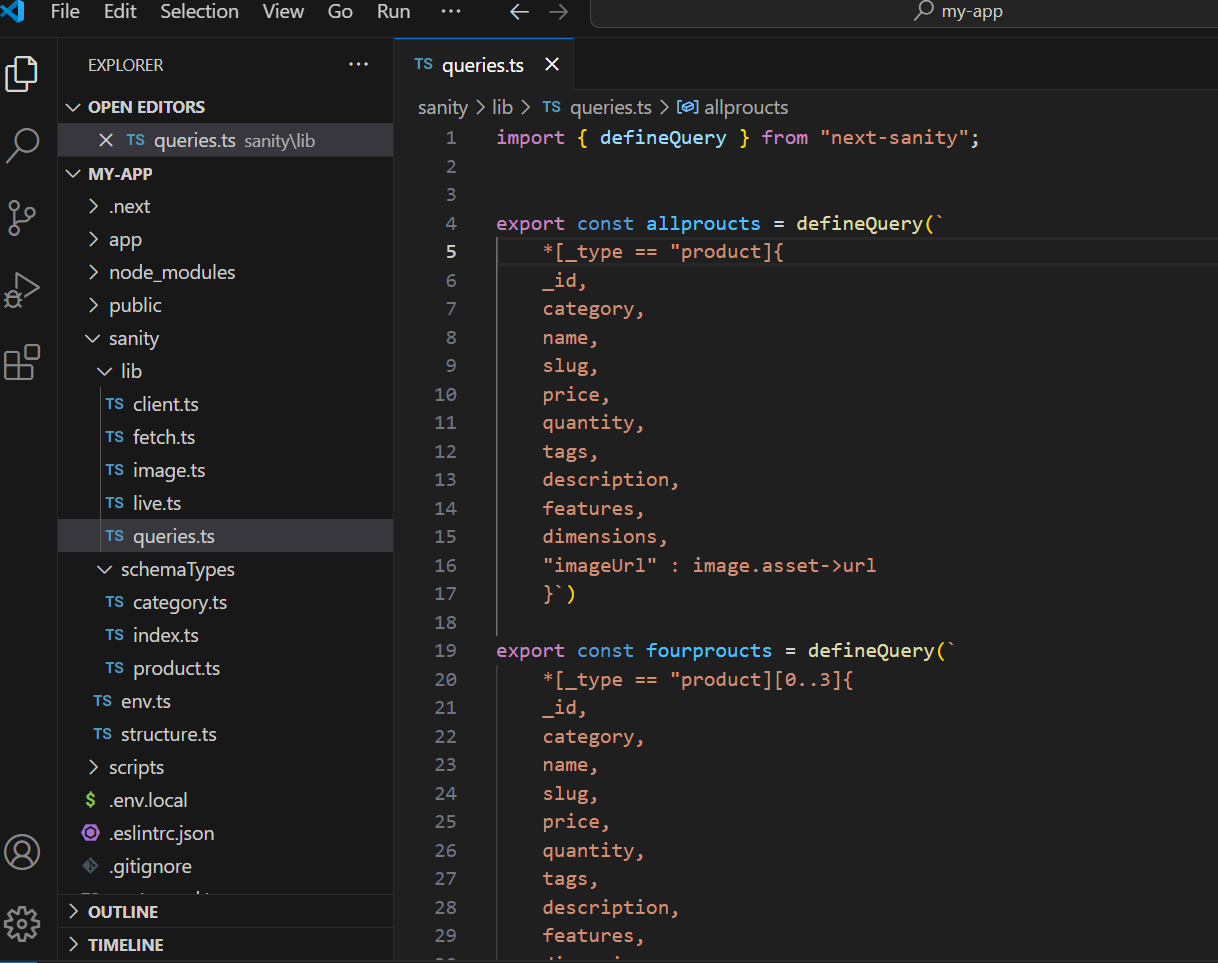
**HOME PAGE :**

* Make file fetch.ts in lib inside sanity and fetch the data from sanity.

(/sanity/lib/fetch.ts)

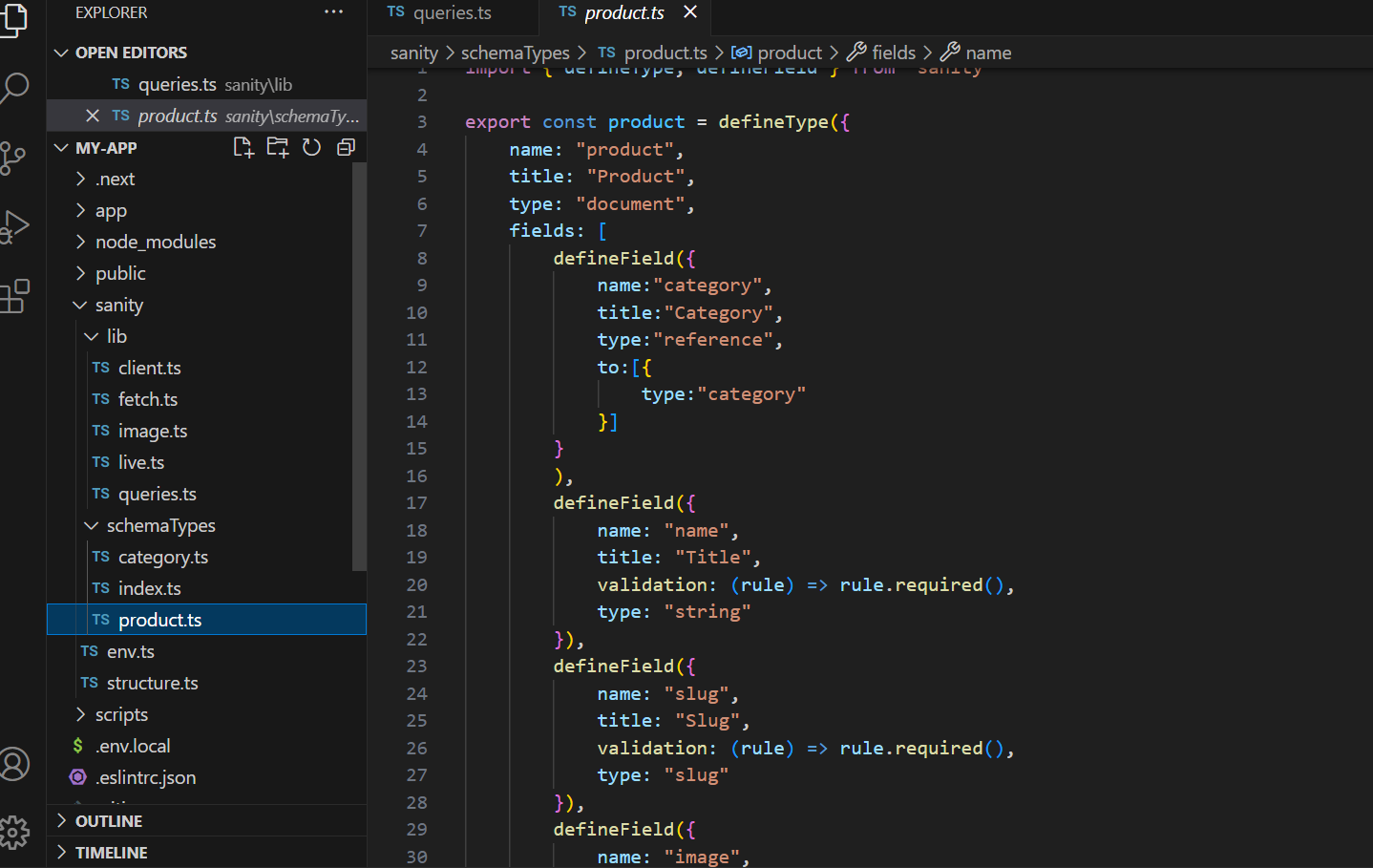
****

* Make another file in sanity, inside lib named queries.ts (/sanity/lib/queries.ts) .
* Define queries in it, the thing we want to fetch from the sanity, that we had defined in the /schemaTypes/product.ts.

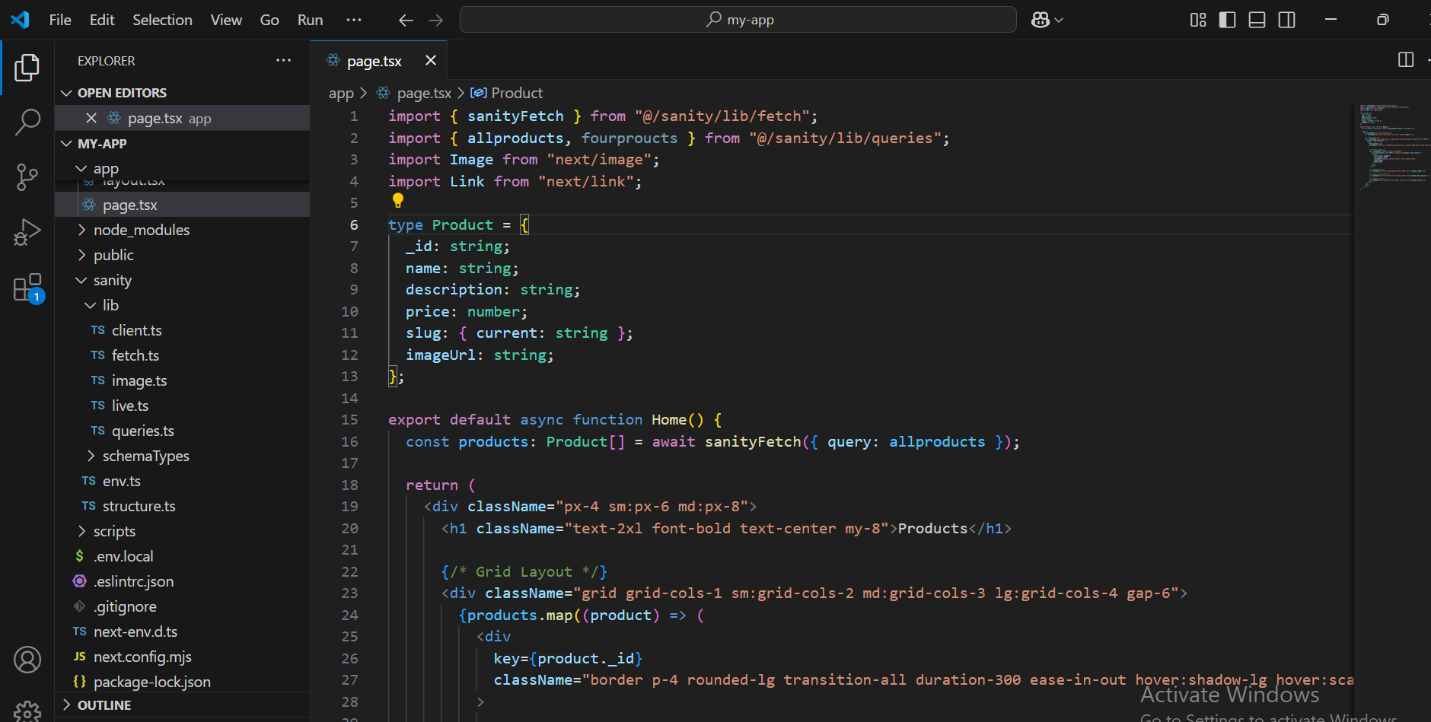


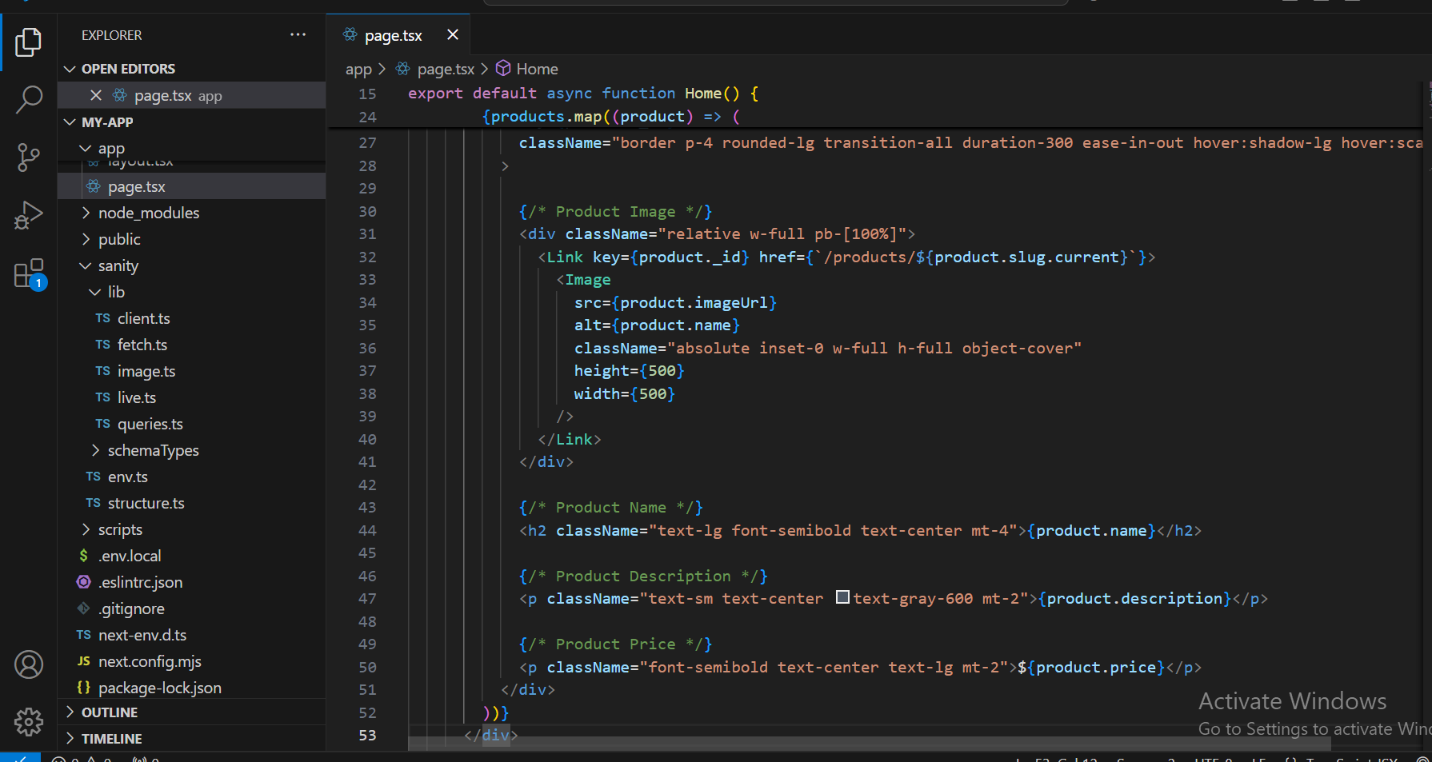
* What have we done??

Ans: We have fetched all the data, things that I had defined in product.ts, I have made two different queries because on main page I need all products and on some places I need 1 row only containing 4 products.



We have defined type of all of these in queries.ts.  
Goal: I choose 4 random from all products later, currently the products are fixed 0 to 3.

* Now I want to display these products on the homepage (currently I am doing these things in a new file so using home page directly for understanding , this could also be done in a specific allproduct page, later I’ll do it. )  
  



* **Things Done In This Code** :

**sanityFetch**: A utility function used to fetch data from the Sanity CMS.

**allproducts**: A query string for retrieving all product information.  
**Dynamic Routing:** This page also has dynamic routing to the specific product detail page using..

**Image**: A Next.js component for optimized image rendering.

**Link**: A Next.js component for client-side navigation between pages. The link is placed on Image to route to specific page product details page.  
**Styling:** Used tailwind CSS classed for styling.

**2.Detailed Dynamic Routing Also Detail Of**

**Things That I Have Done On Homepage For**

**Dynamic Routing.**

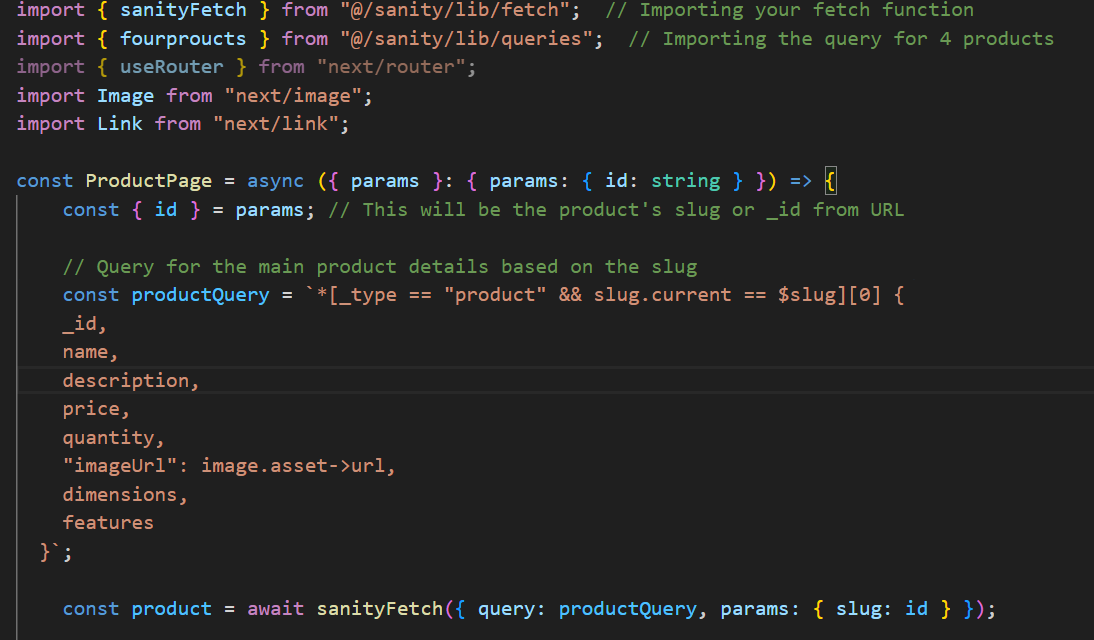
* Make a folder product, Dynamic folder of id in it [id] and a page in It page.tsx.

(/products/[id]/page.tsx)

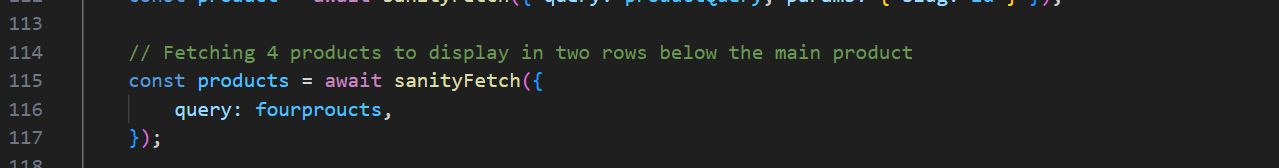
* **Fetching Main Product Details By slug**:  
  Utilizes a productQuery to fetch specific product information from Sanity CMS using the product’s slug.

The fetched details include:

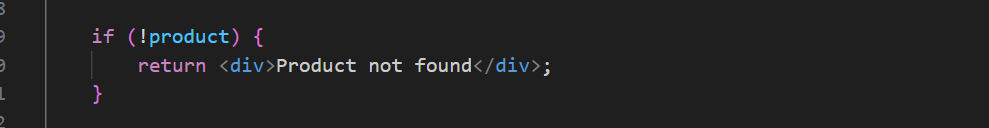
1. \_id, name,
2. description,
3. price,
4. quantity
5. Image URL (derived from the image asset in Sanity)
6. Product dimensions (height, width, depth)
7. Features (list of attributes or benefits)



* **Fetching Suggested Products**:  
  Executes the **fourproucts** query to fetch four additional products to display as recommendations below the main product.

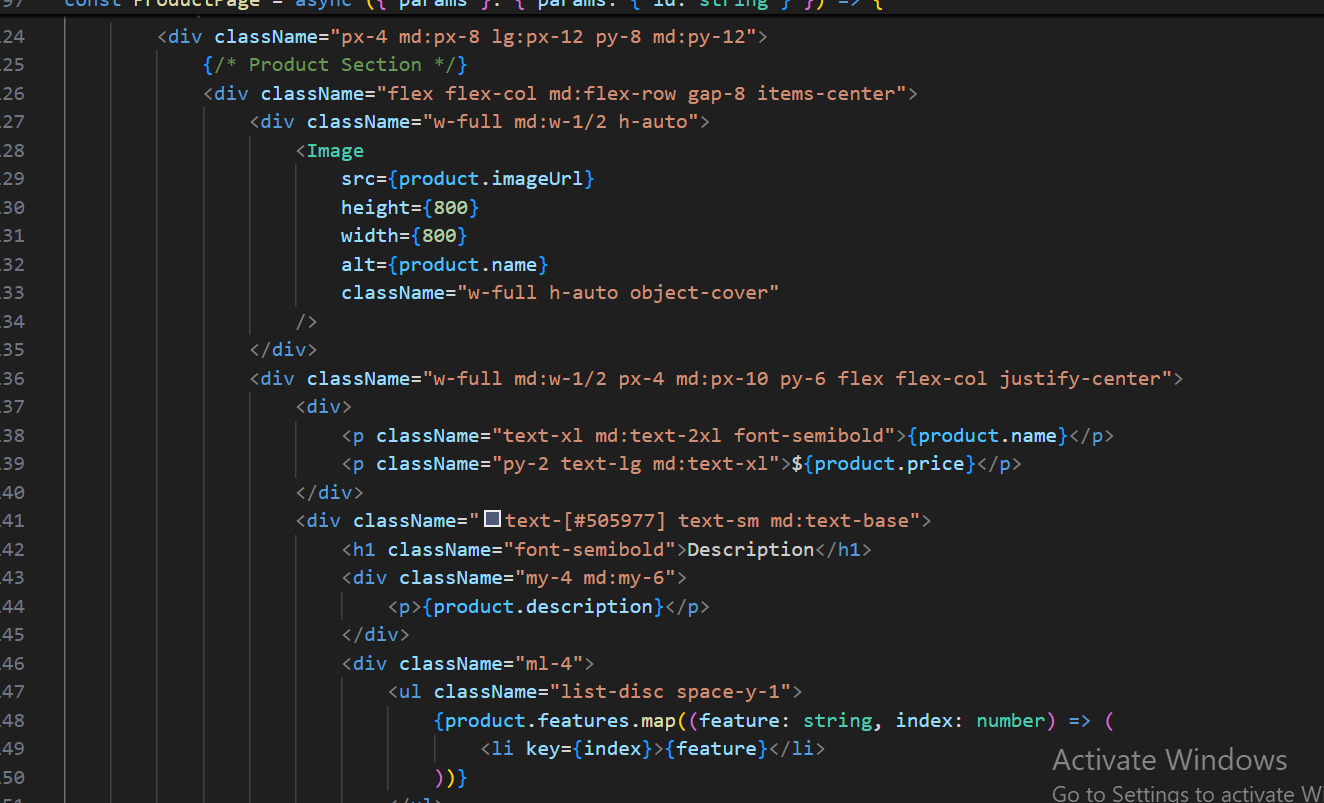


* **Error Handling**:  
  Displays a "Product not found" message if the product object is null, ensuring the user is informed when no product matches the given id.



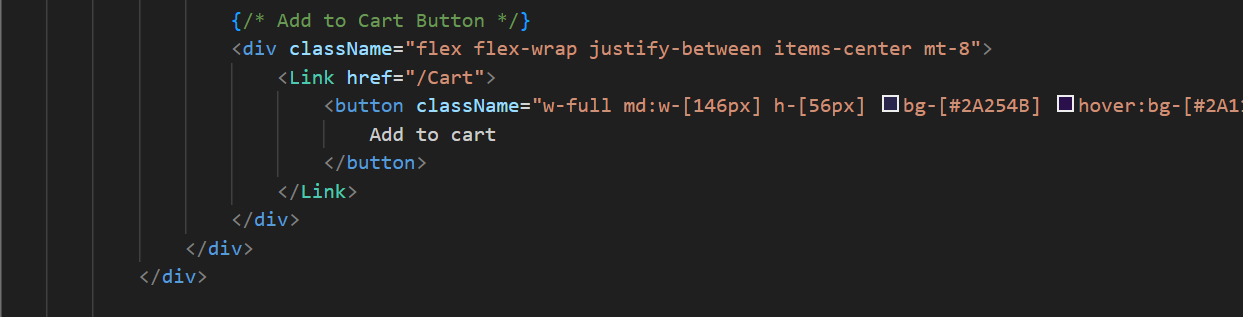
* **Dynamic Rendering of Product Details**:

1. **Image Rendering**: The product's main image is displayed using the optimized Image component.
2. **Textual Information**: Dynamically renders the product name, price, description, dimensions, and a list of features.



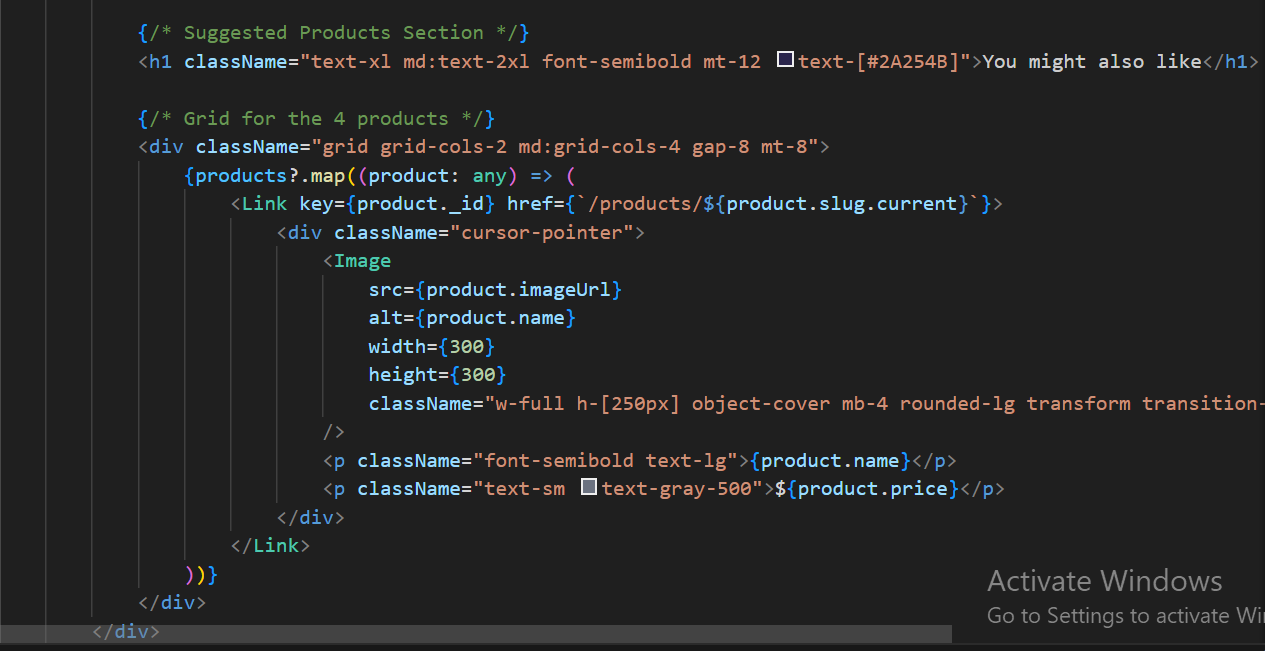


* **Add to Cart Button**:  
  Includes a button that redirects to the /Cart page, simulating the "Add to Cart" functionality.(this page is not added as yet , but will add soon inshaAllah.)



* **Rendering Suggested Products**:

1. Displays a grid layout for four suggested products.
2. Each product is linked to its respective product detail page using the product's slug.



* **Styling and UI Enhancements**:

1. Tailwind CSS is used for styling the layout, ensuring responsiveness for both mobile and desktop views.

* **Connecting Both Pages**

This uses both the **id** and the **slug**.

**Id:**

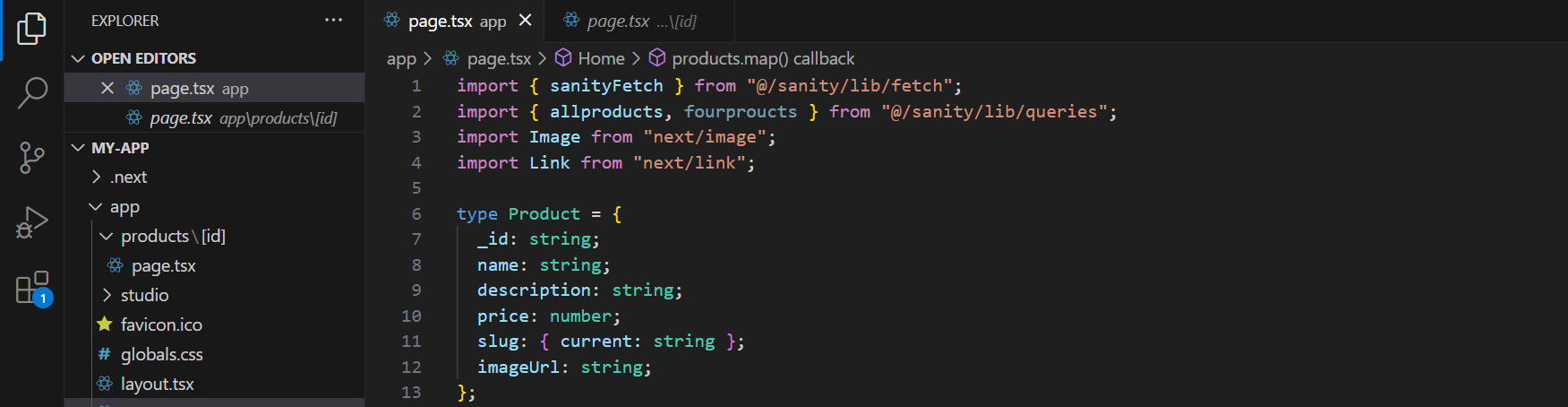
* **Role:** The id is the unique identifier for a product, extracted from the URL when a user navigates to a specific product page.
* **Usage:** It is used to dynamically query the product's details from the database (Sanity CMS).
* **Example:**
  + URL: /products/chair123
  + Extracted id: "chair123"

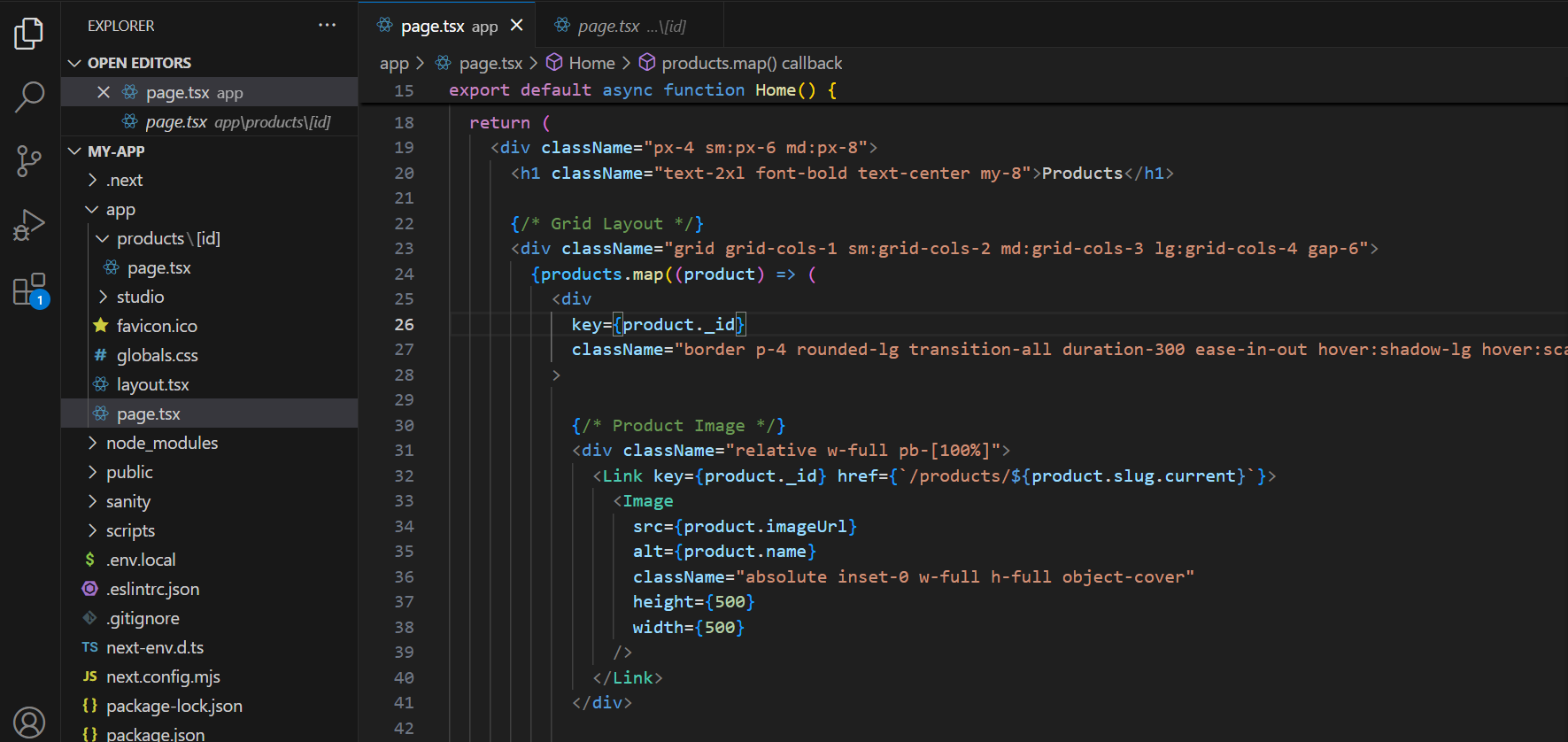
**Slug:**

* **Reason for Use:** Using slug because it is easier to understand.
* **How it Works:** Slug uses id as a parameter to link to the product details page.

Both pages are connected by placing a link on each image on the allproduct page which is currently homepage.

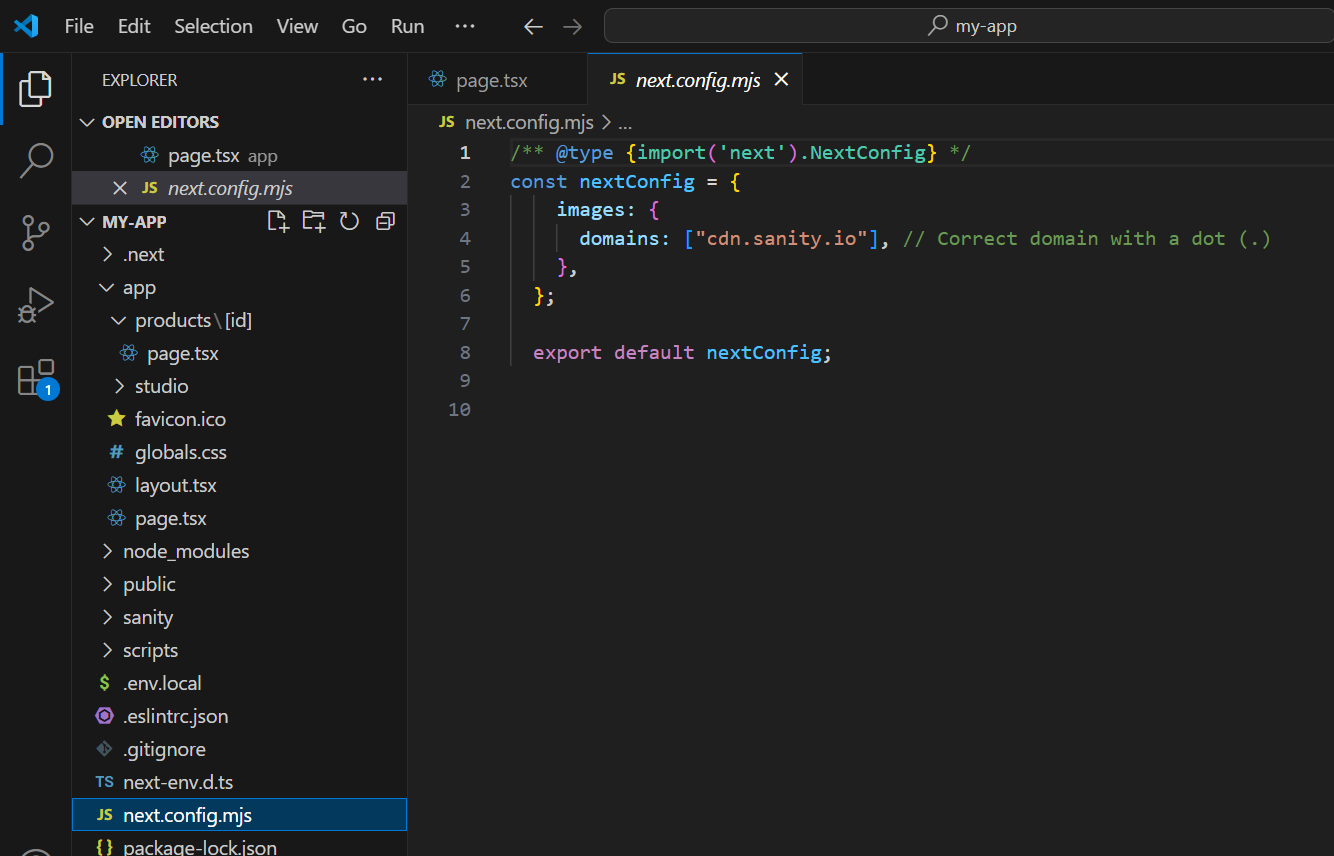
The link uses slug



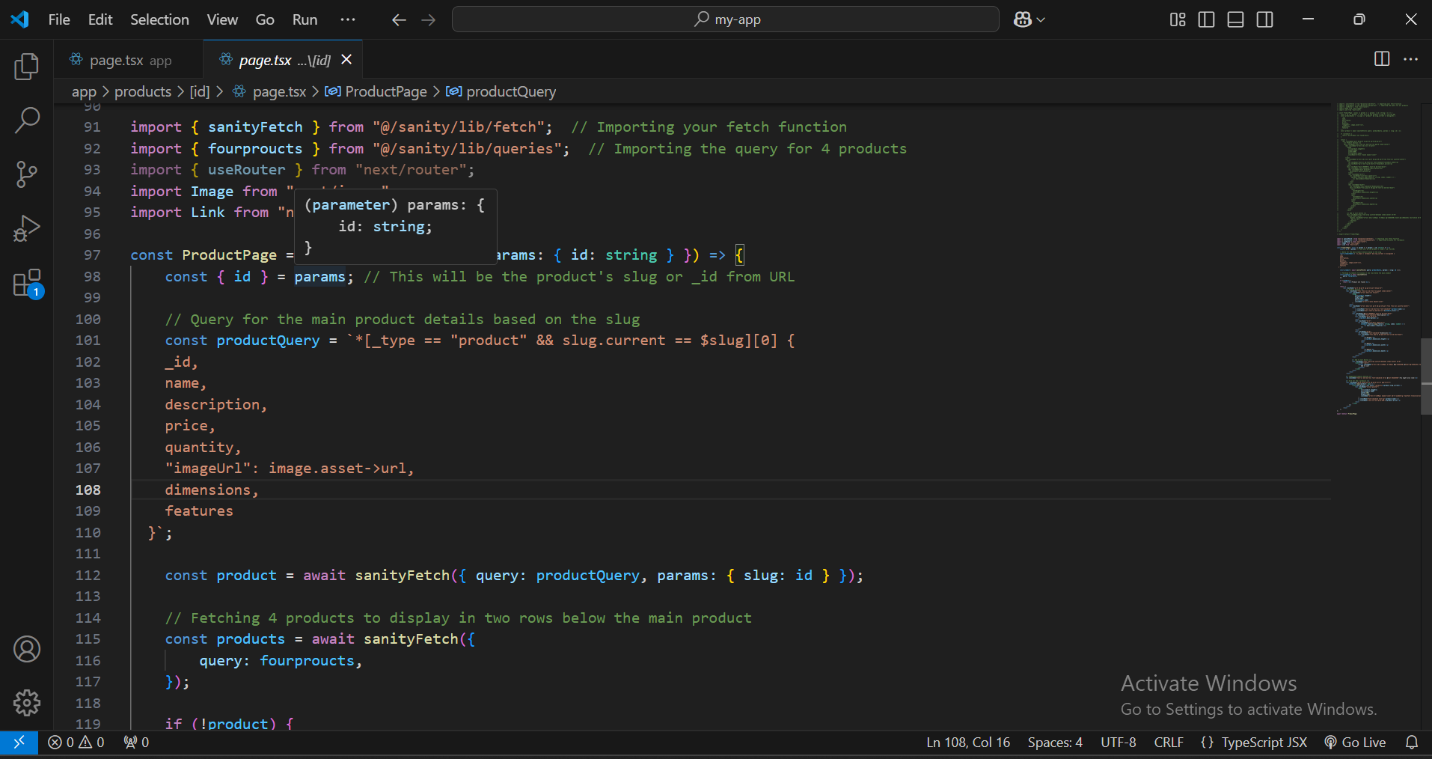


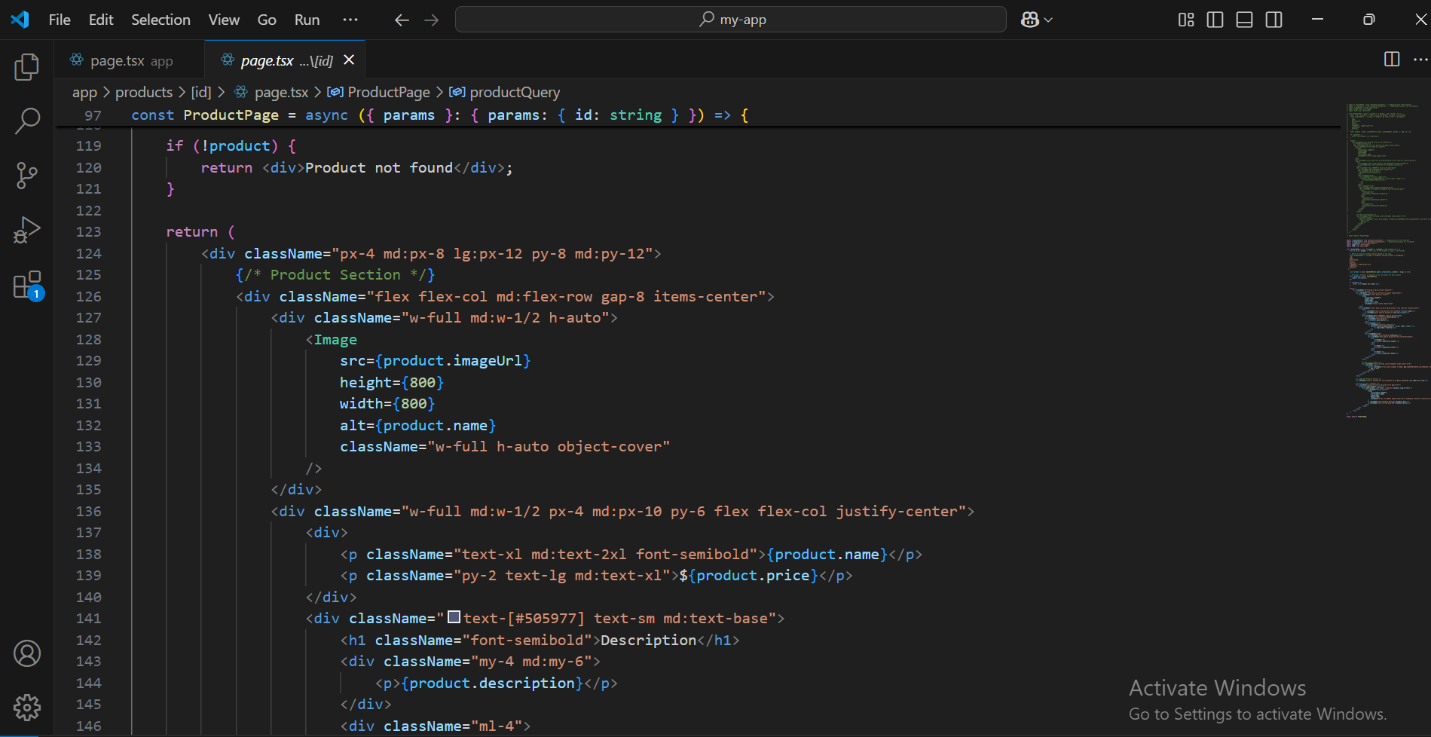
**Errors I Faced:**

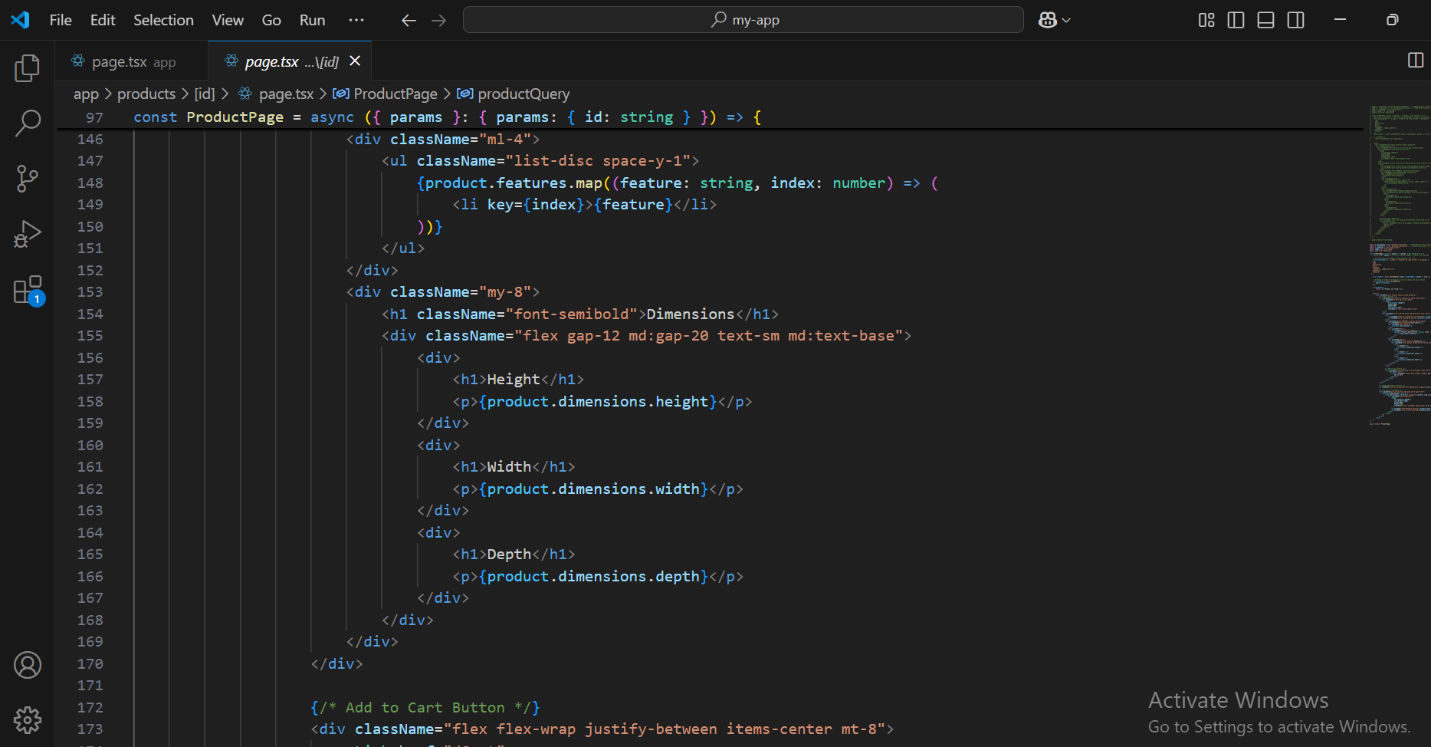
Wasnot modifying next.config.mjs

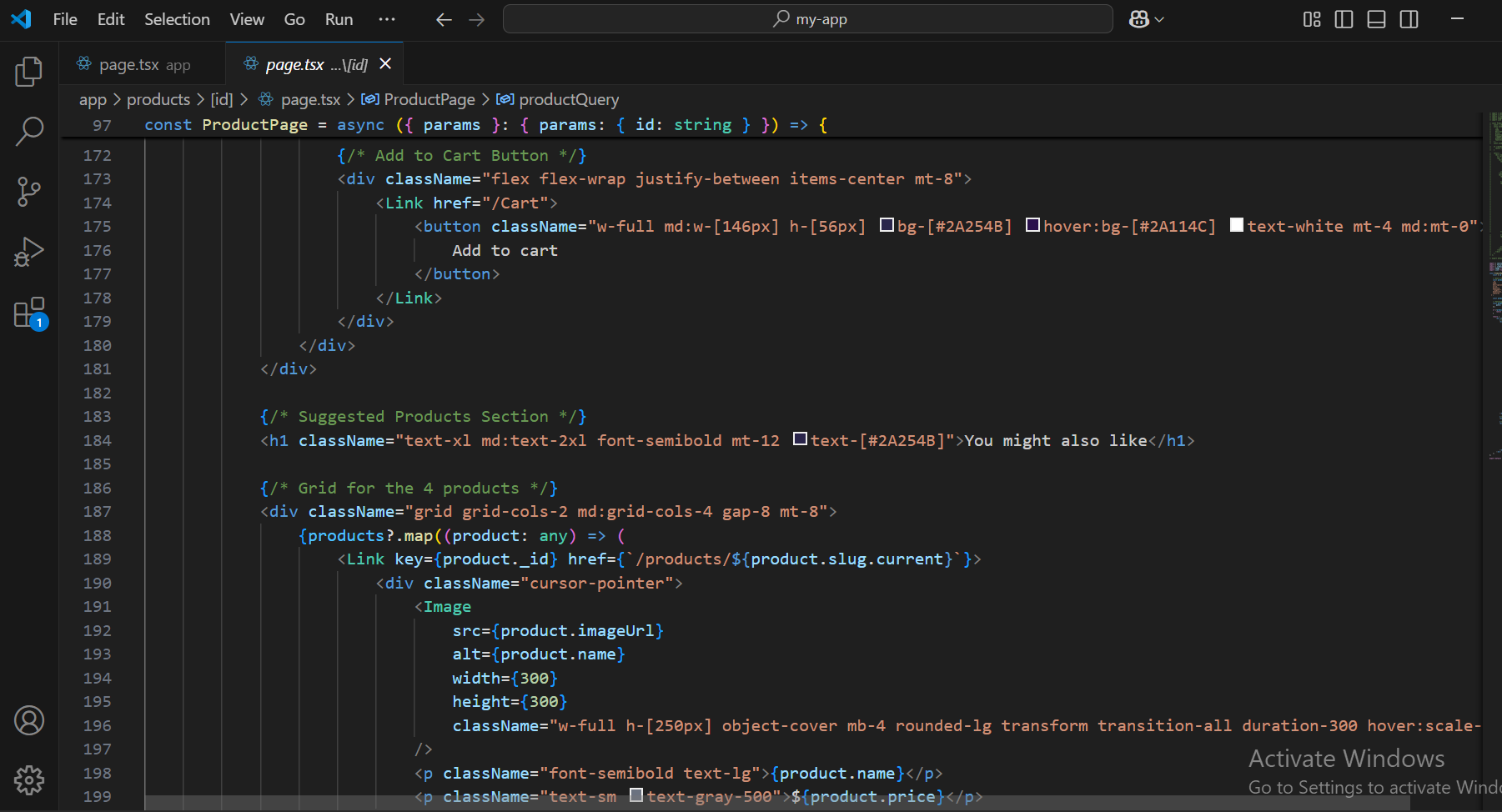


* **Full code of products/[id]/page.tsx**

****

****

****

****