A close up of a logo

Description automatically generated

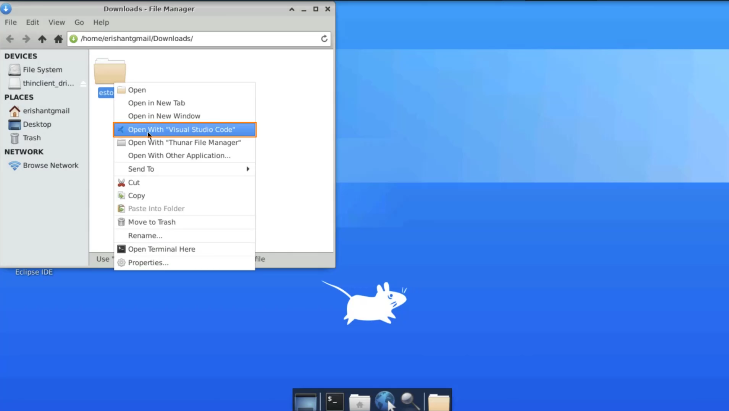
**Lesson 2 Demo 1**

**Component Interaction**

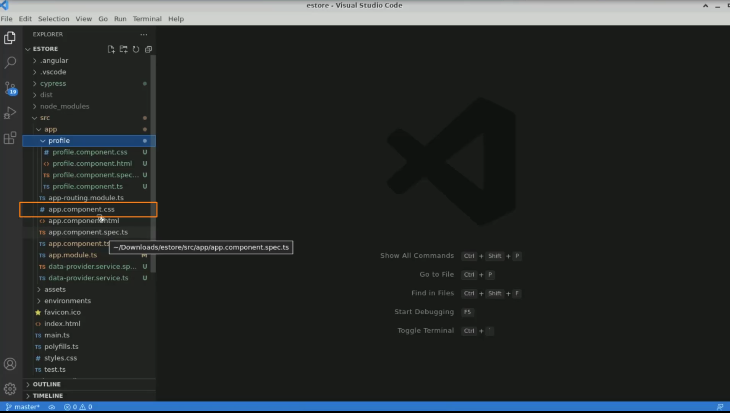


**Steps to perform:**

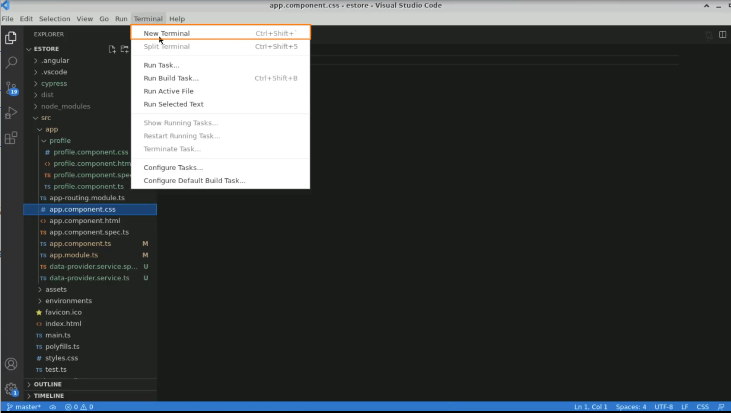
1. Open the project with the Visual Studio



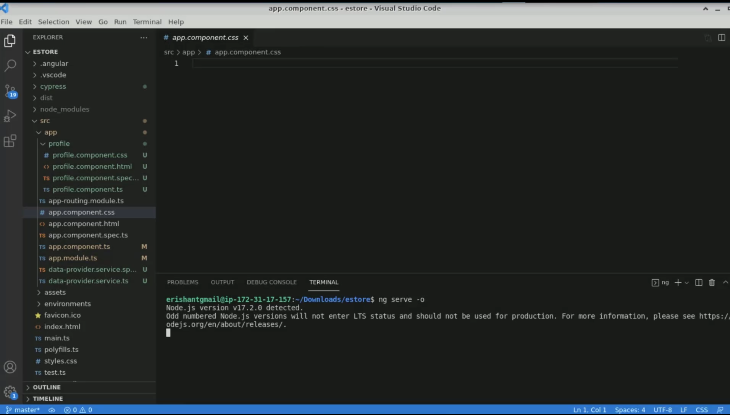
1. Open the **app.component.css**



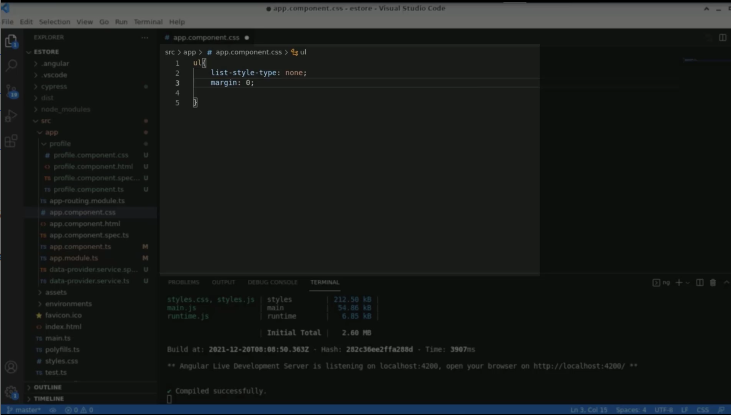
1. Open the terminal window



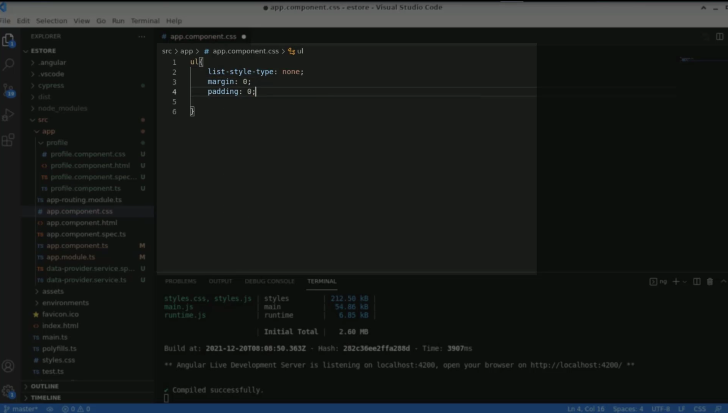
1. Execute the command: **ng serve -o**



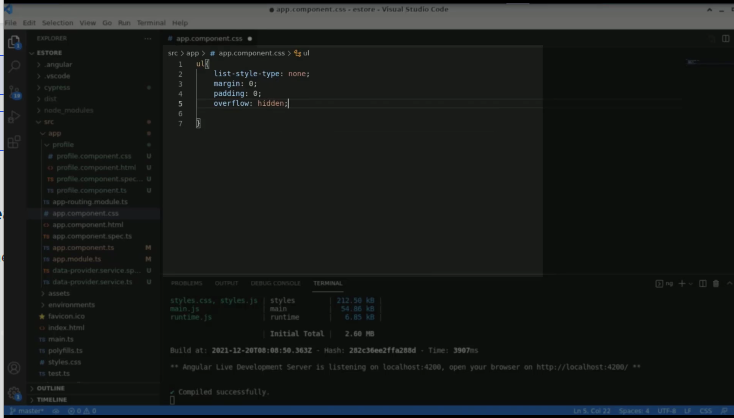
1. Define a CSS, create an unordered list with a margin of zero



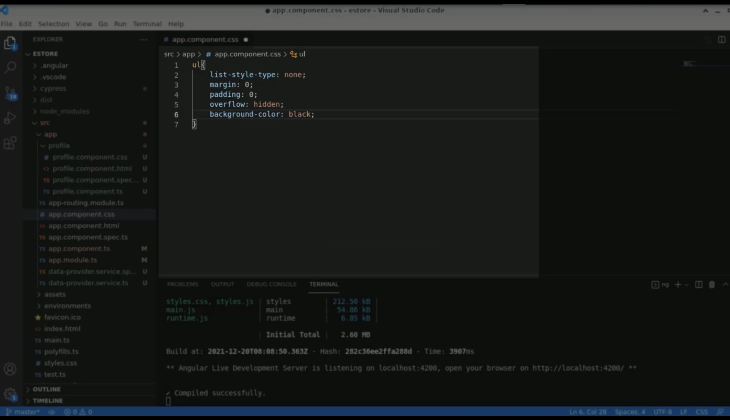
1. Add padding to be zero



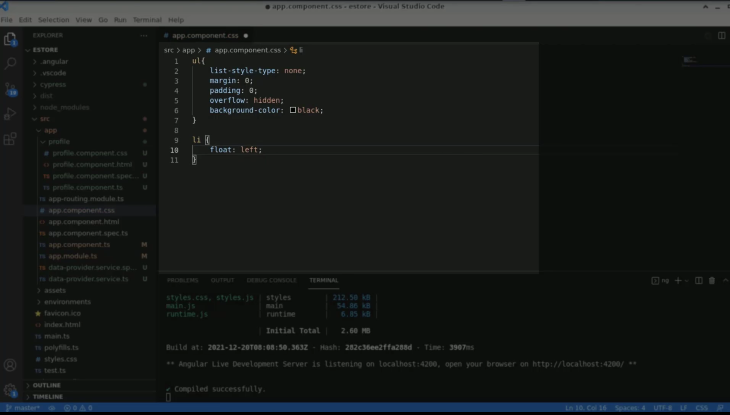
1. Add overflow as hidden



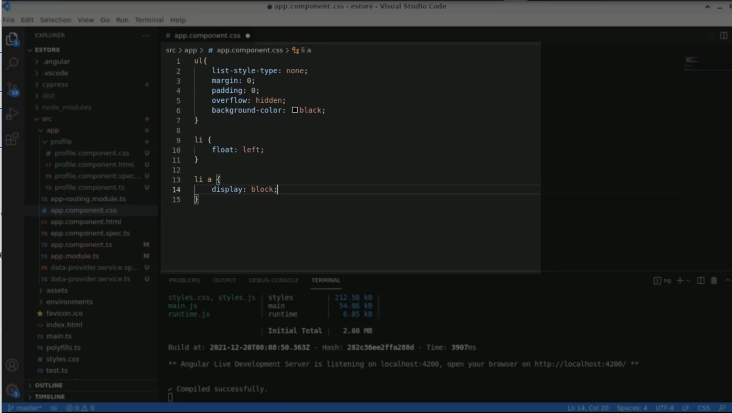
1. Add a background color



1. Add CSS to the list with float to the left



1. For hyperlinks, put display as a block



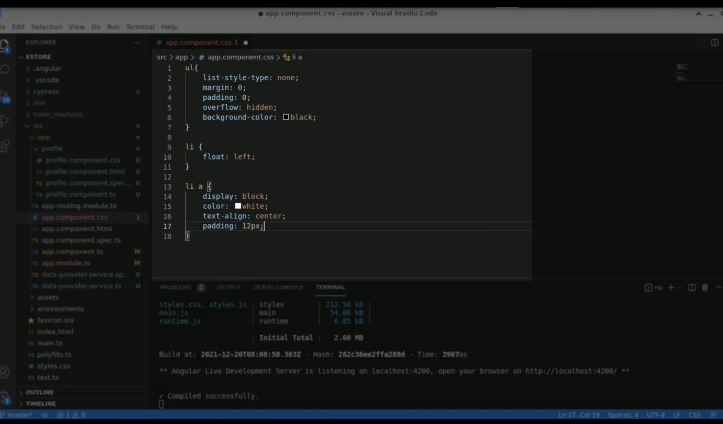
1. Add a color



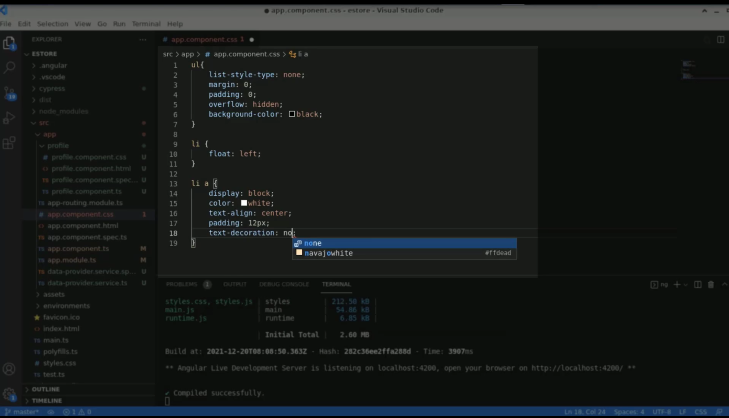
1. Align text at the center



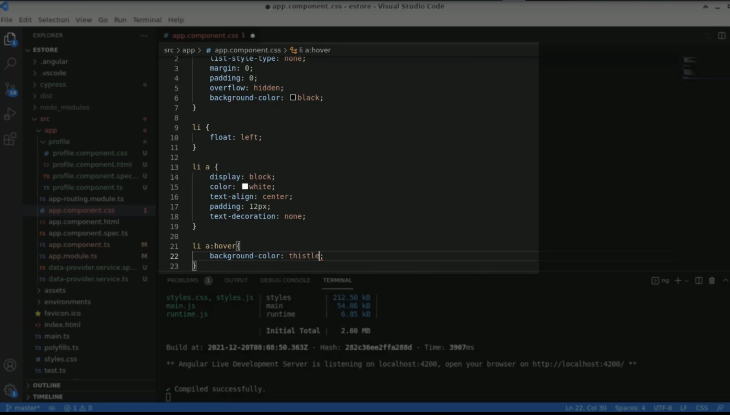
1. Add padding of 12 pixels



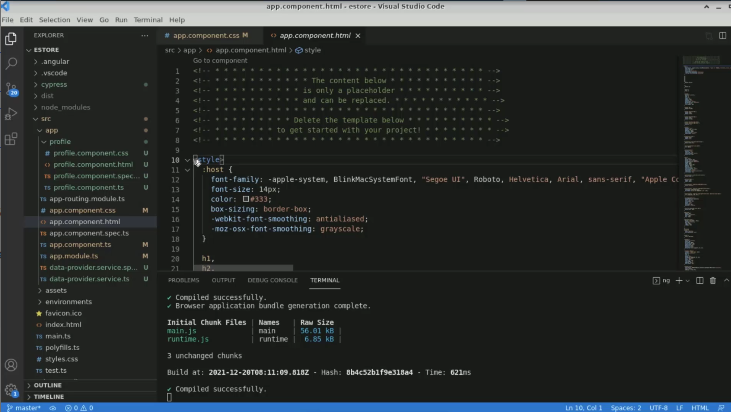
1. Add text decoration as none



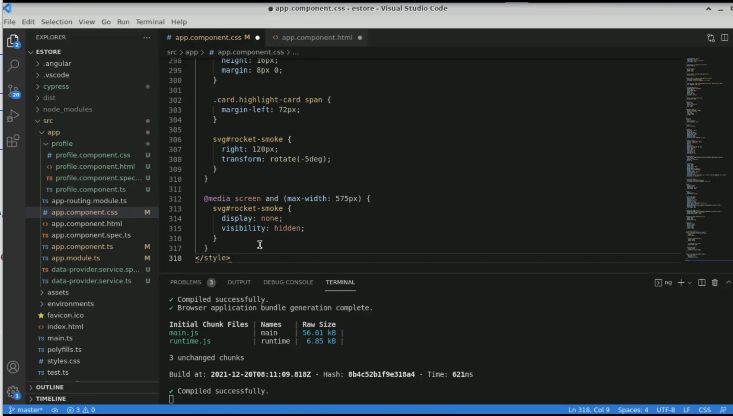
1. Type **li a as a hover** and add a background color



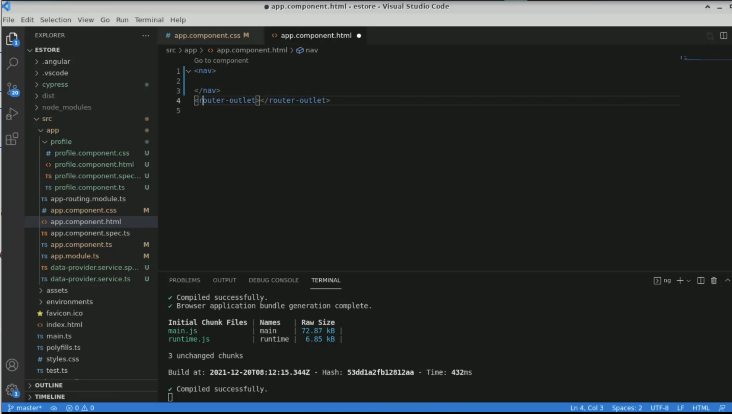
1. Cut the entire style from the HTML file



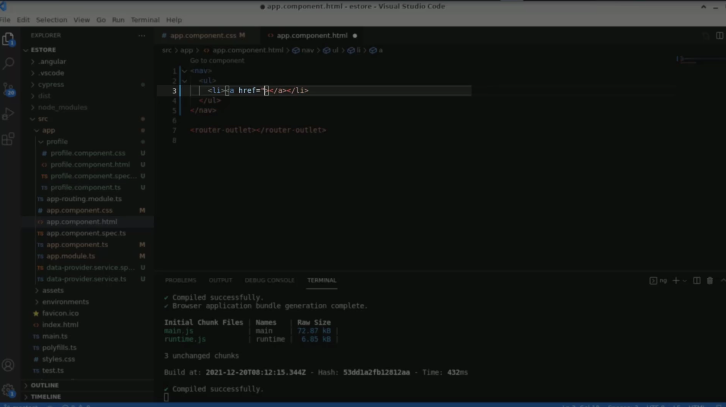
1. Bring the style here and remove the style tag



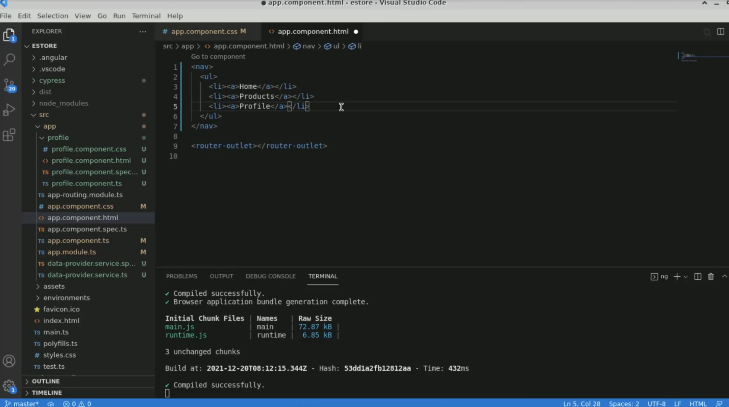
1. Create a navigation component in **app.component.css**



1. Create an unordered list with the anchor tag



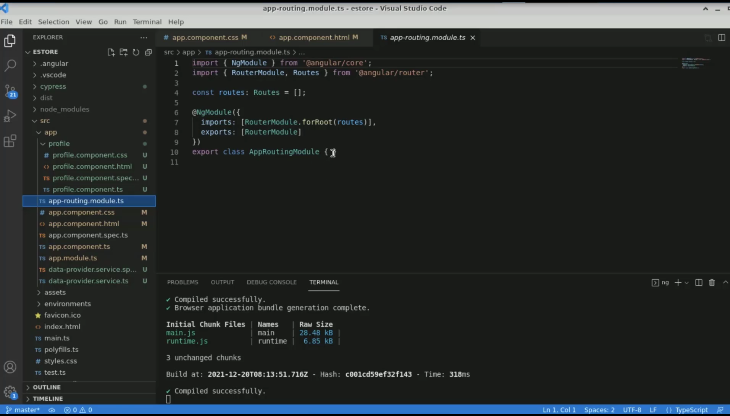
1. Add fields such as profile, products, and home



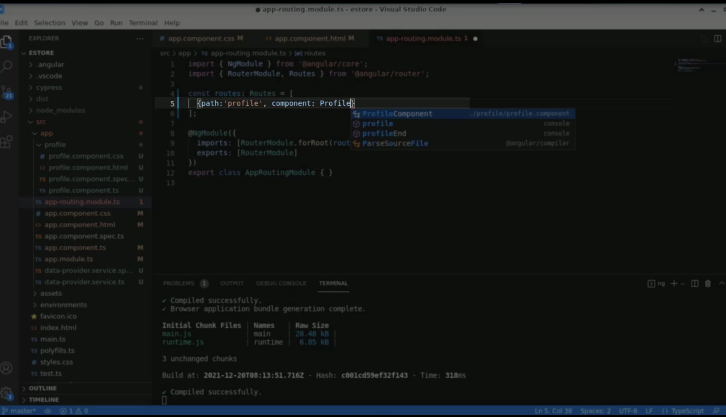
1. Check the browser for the result



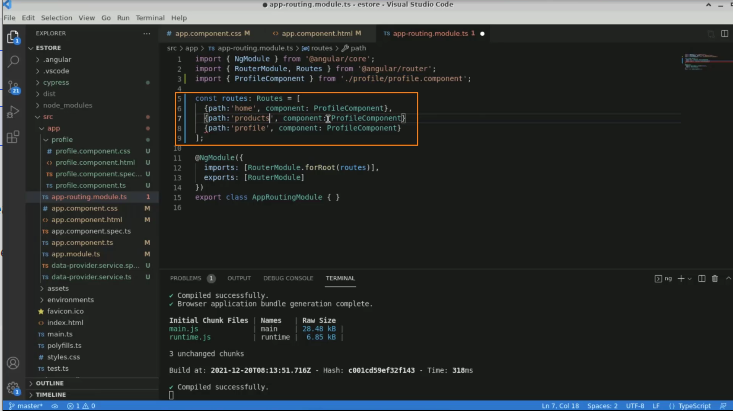
1. Configure routing in the **app.routing.module.ts** file



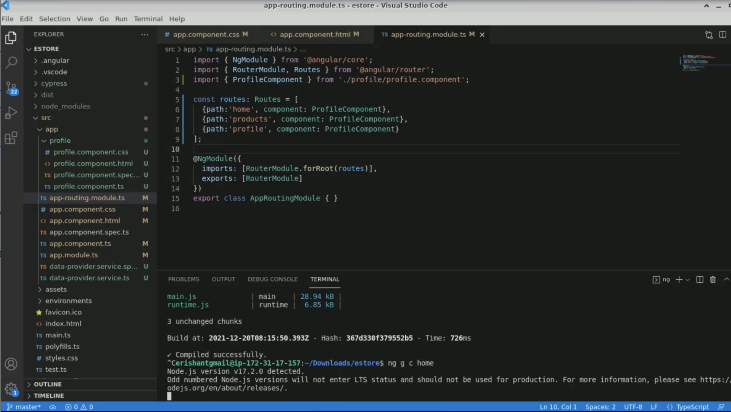
1. Create a JSON object with a path and component



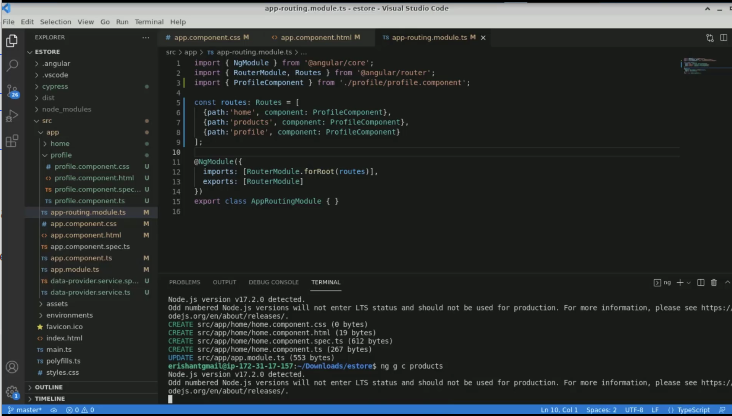
1. Create roots with different components



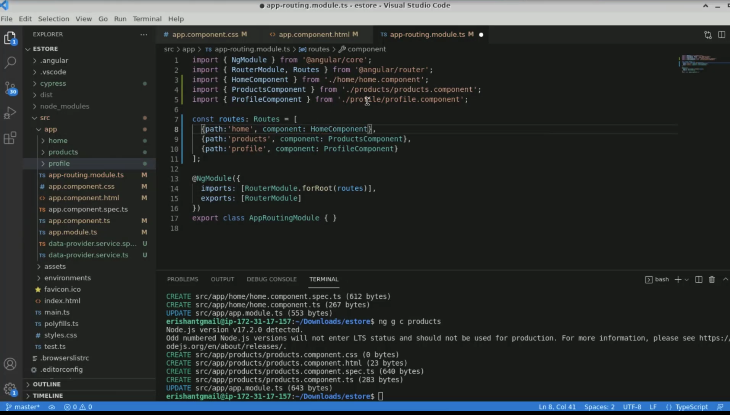
1. Generate the components for the home, using the terminal



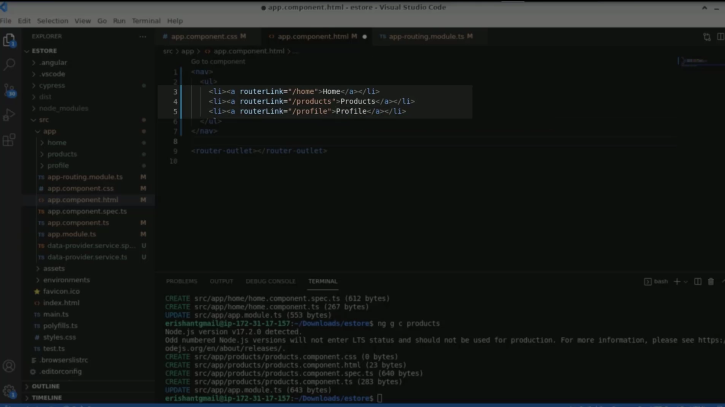
1. Generate components for products



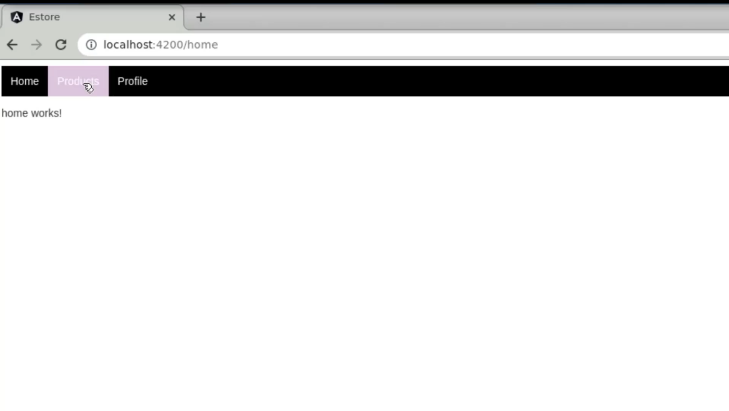
1. Write **homecomponent** for **home** and **productscomponent** for **products**



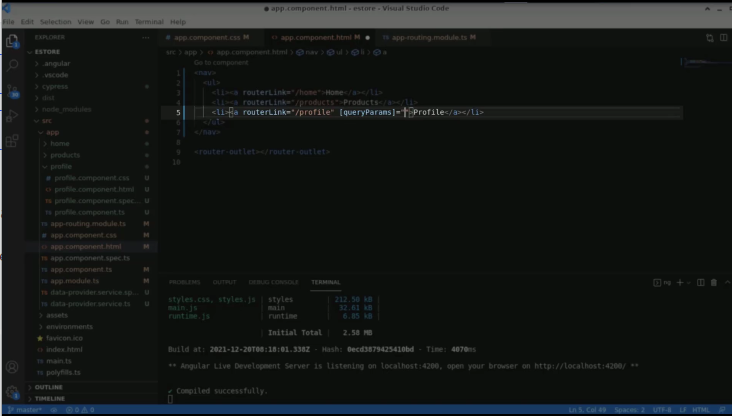
1. Add a path in the anchor tag for each component



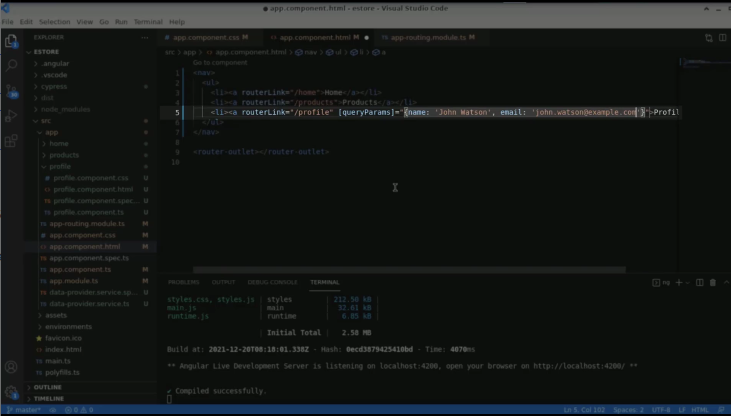
1. Execute the code in the terminal again



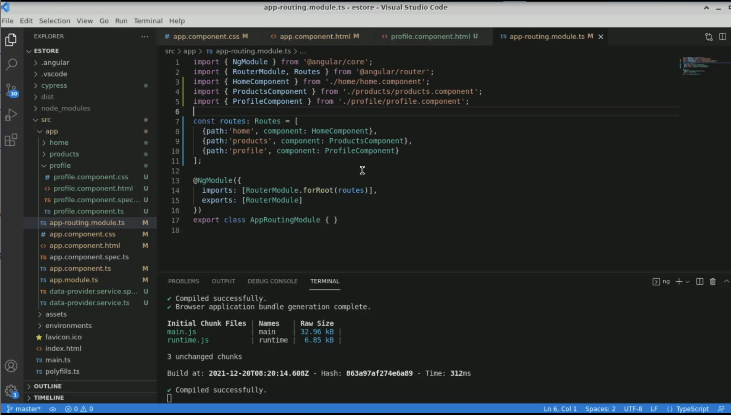
1. Share query parameters for the profile component



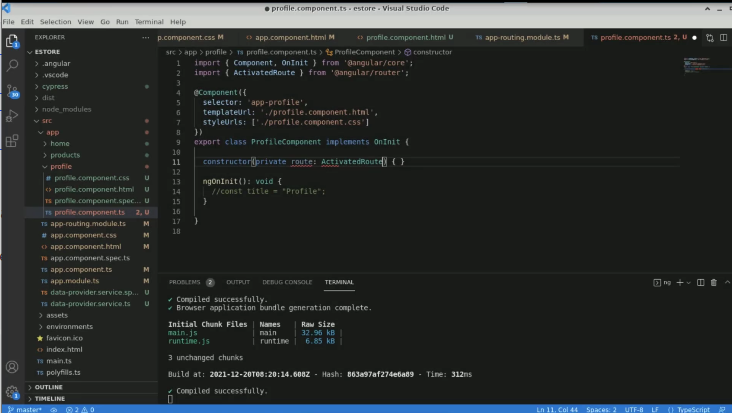
1. Create objects for query parameters with their values



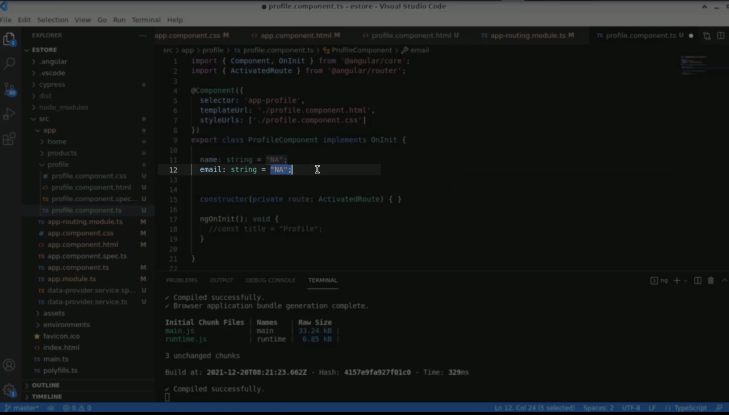
1. Open a **profile.component.html** file



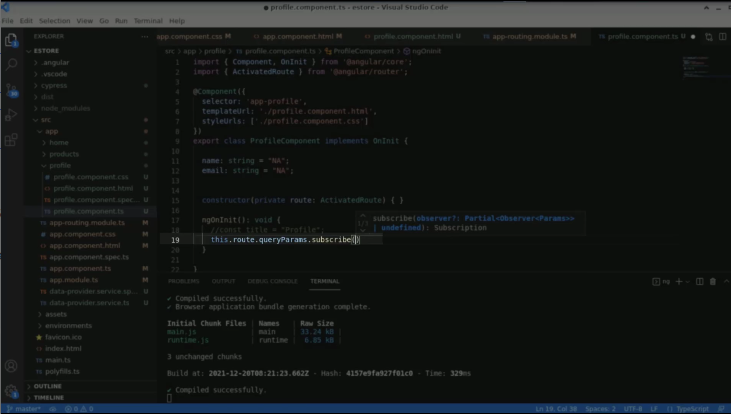
1. Type code to extract the data



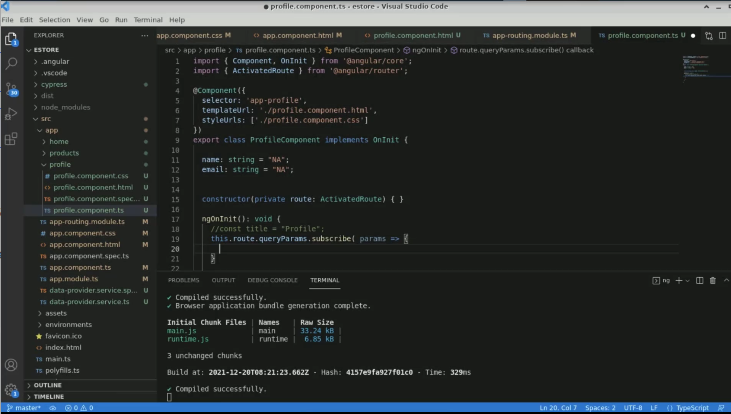
1. Add a dependency injection to extract the data



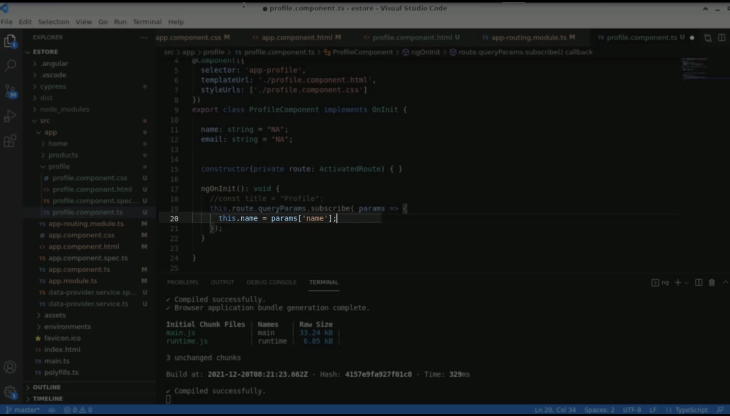
1. Add the **.route** function in the **ngOnInit** to access the variable



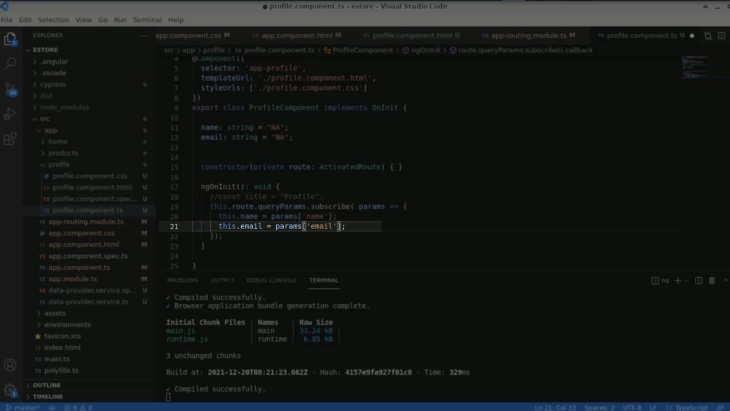
1. Pass one input in the param query



1. Extract the name from the params parameter



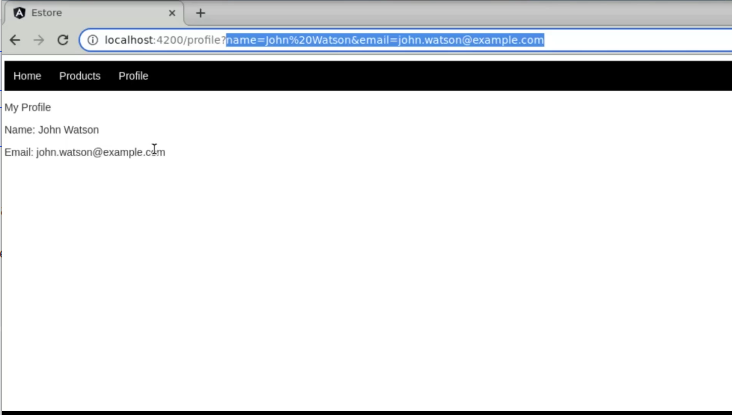
1. Extract email from the params parameter



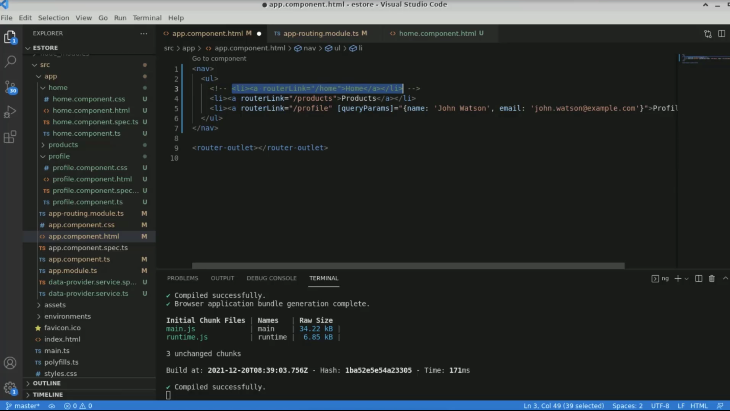
1. Display the name and the email in **profile.component.html** using the paragraph tag



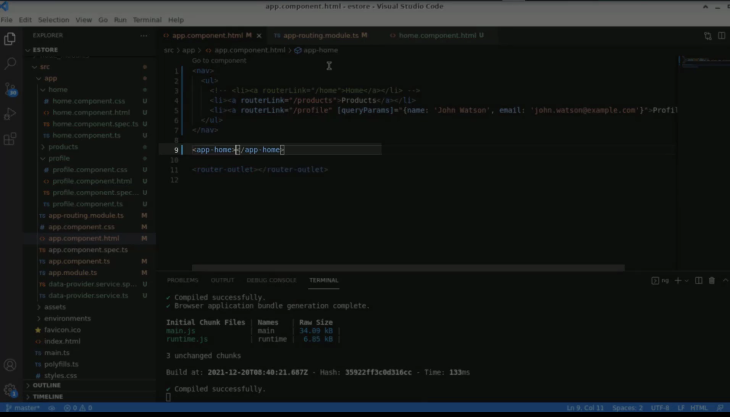
1. Check the output



1. Comment on the **home** statement



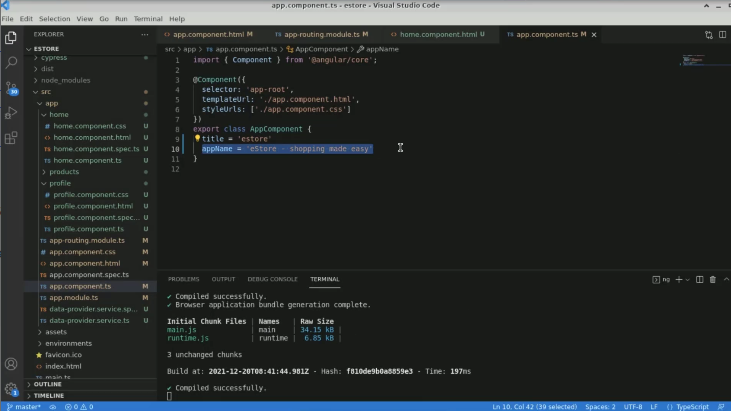
1. Bring the home onto the page using the **app home** selector



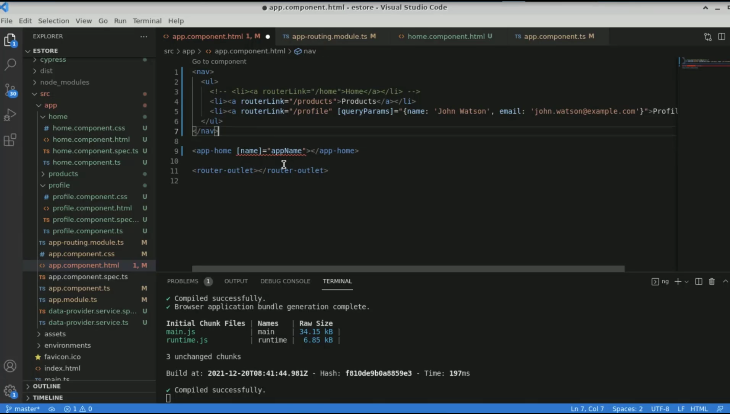
1. Refresh the page



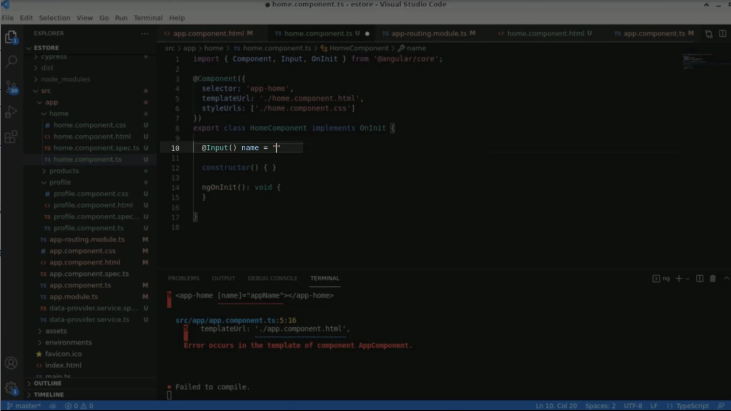
1. Open the **app.component.ts** file, add **appName**



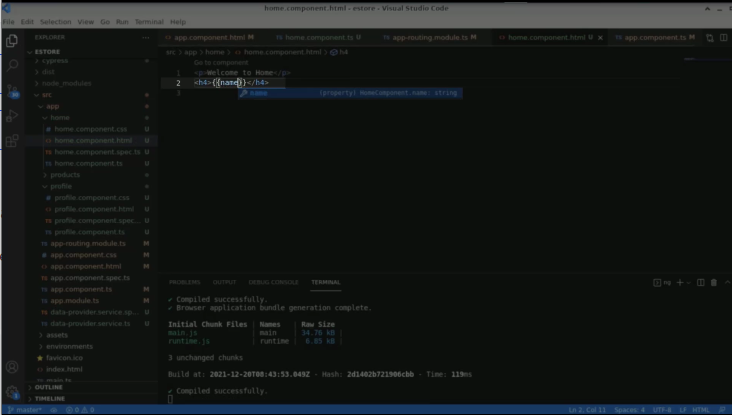
1. Put a variable with its value in a selector



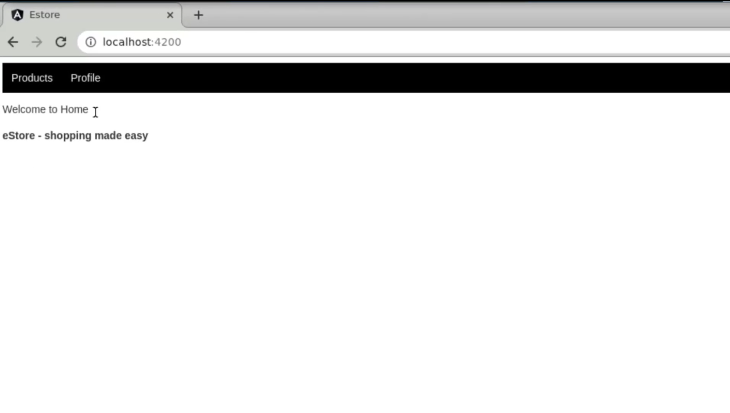
1. Come to the **home.component.ts** file and declare a variable



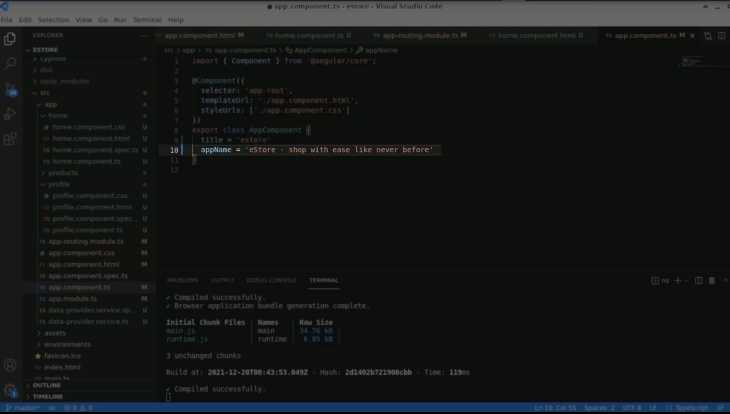
1. Create one more variable with an h4 tag and save it



1. Refresh the page



1. Change the variable value



1. Output is changed again

