Intercepting Routes in Next.js

Intercepting Routes:

In Next.js, intercepting routes refer to the capability to intercept or stop the default routing behavior of the application to present an alternate view or component when navigating through the user interface while still preserving the intended route.

The purpose of intercepting routes in Next.js is to provide more control over the rendering of UI elements based on navigation actions. This can be particularly useful in scenarios where you want to display content without fully navigating away from the current page, such as showing a modal instead of navigating to a separate page when clicking on a link.

By intercepting routes, you can modify the behavior of route navigation dynamically, allowing for a smoother and more seamless user experience. It enables developers to manage routing logic more flexibly and tailor the UI to specific use cases, ultimately enhancing the usability and functionality of the application.

For example, in a photo feed application, users can browse through a collection of images. Typically, clicking on a photo would direct users to a separate page dedicated solely to that image. However, with intercepting routes, this behavior is altered. Instead of redirecting users to a new page, clicking on a photo triggers a modal window to appear within the feed. This modal showcases the selected photo in an enlarged view along with additional details. Despite this change in display, the URL is updated to reflect the chosen photo, ensuring its shareability. Even if users access the page directly via URL or reload the page, they are still presented with a comprehensive view of the photo. Implementing intercepting routes in a Next.js application follows specific conventions.

In Next.js, there are conventions for defining intercepting routes:

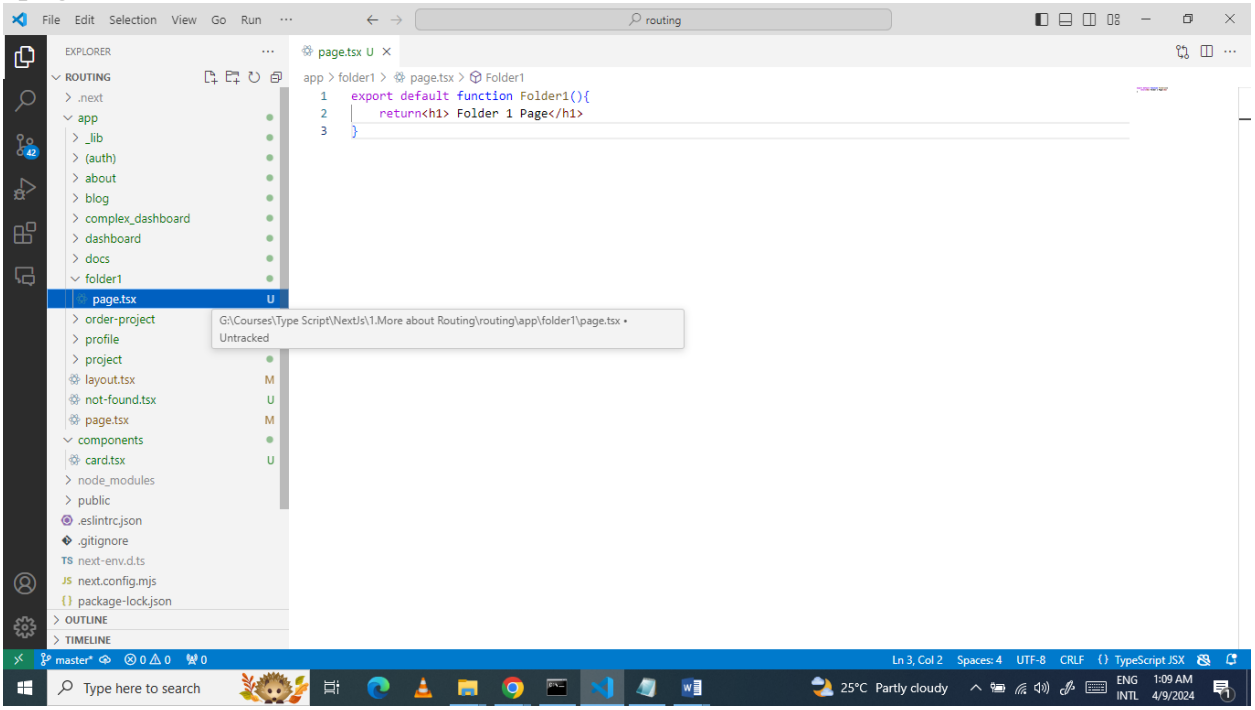
(.) : This convention is used to match segments on the same level.

(..) : This convention is used to match segments one level above.

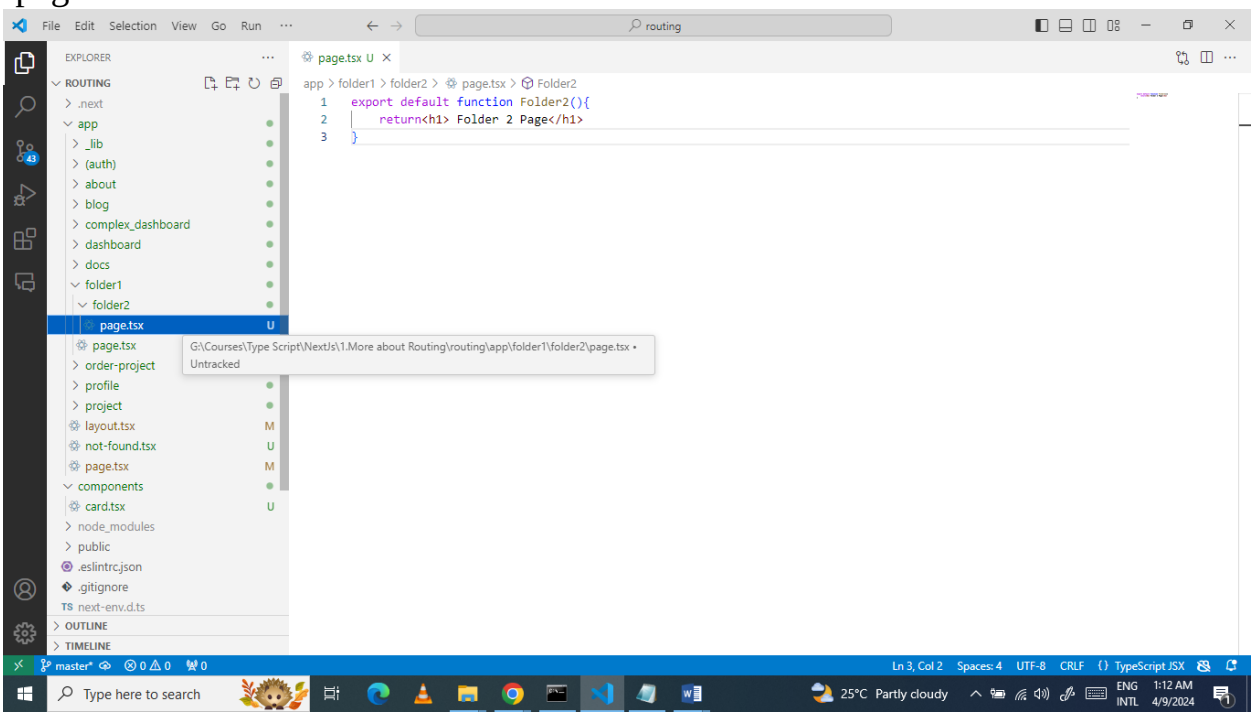
(…) : This convention is used to match segments from the root app directory.

The implementation of the convention using a single dot within parentheses (.) for intercepting routes in Next.js

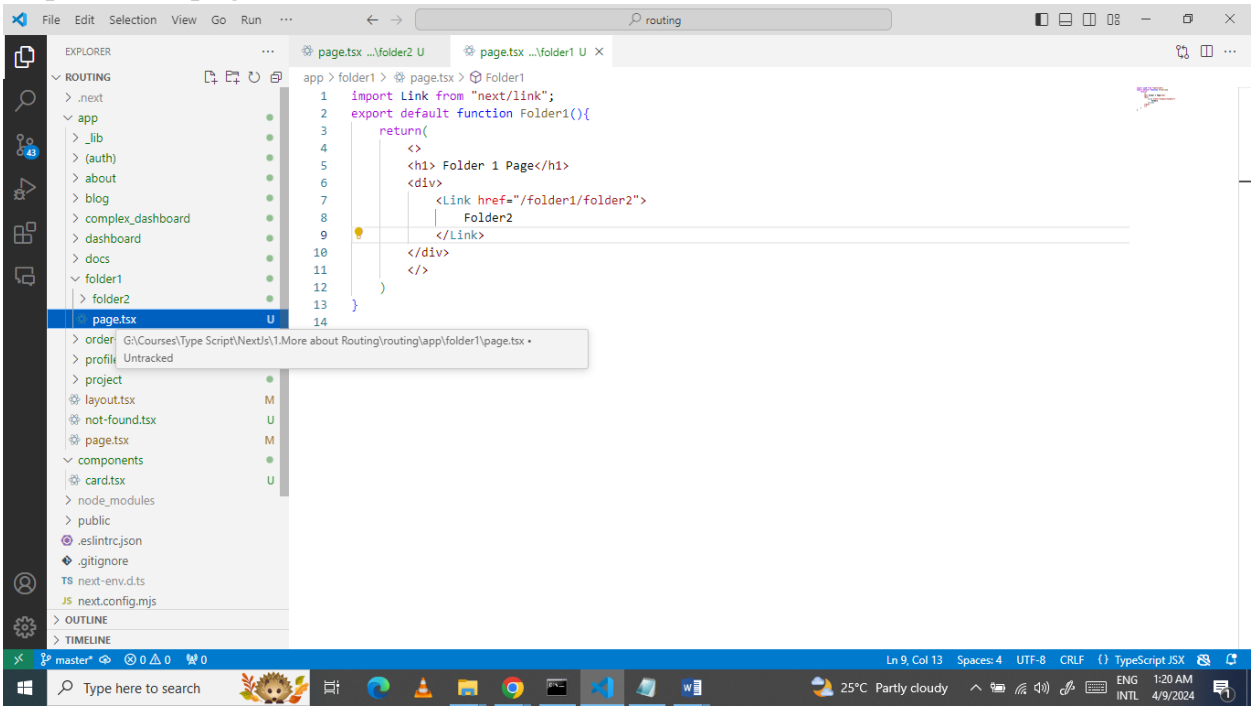
Create a folder named "folder1" in the "app" folder. Inside "folder1", create a "page.tsx" file.



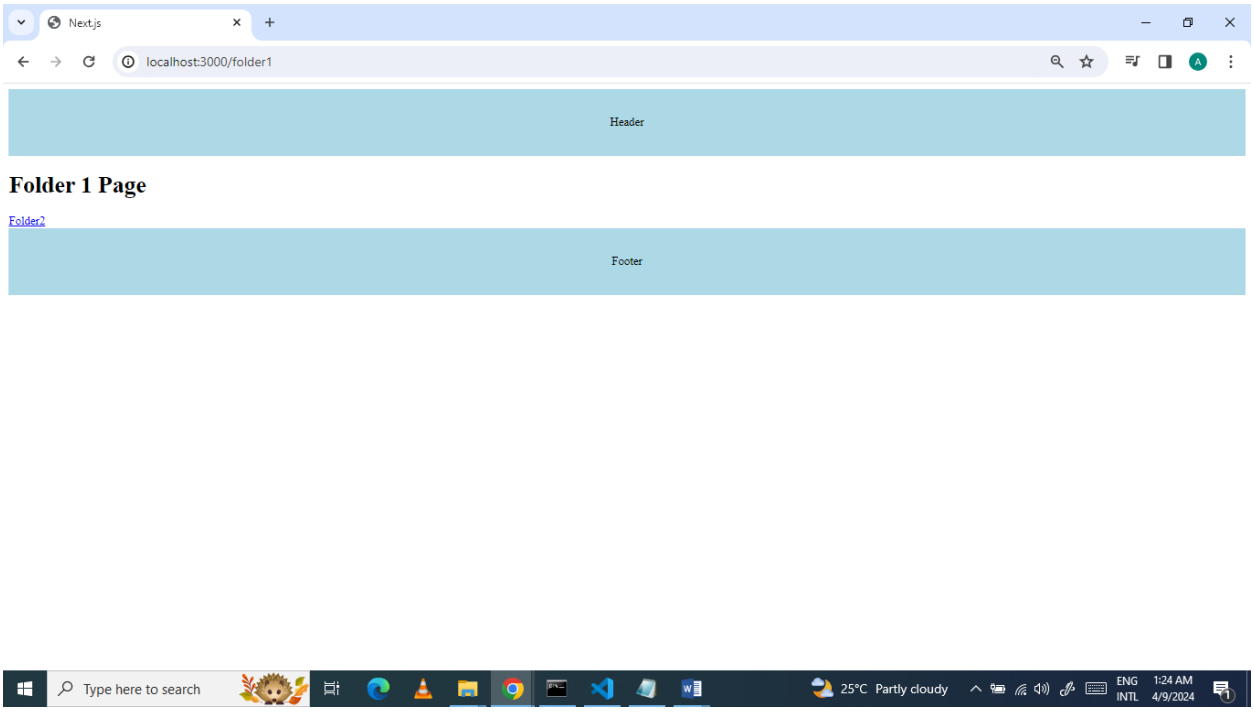
Inside `folder1`, create another folder named `folder2`. Then, create a `page.tsx` file inside `folder2`.



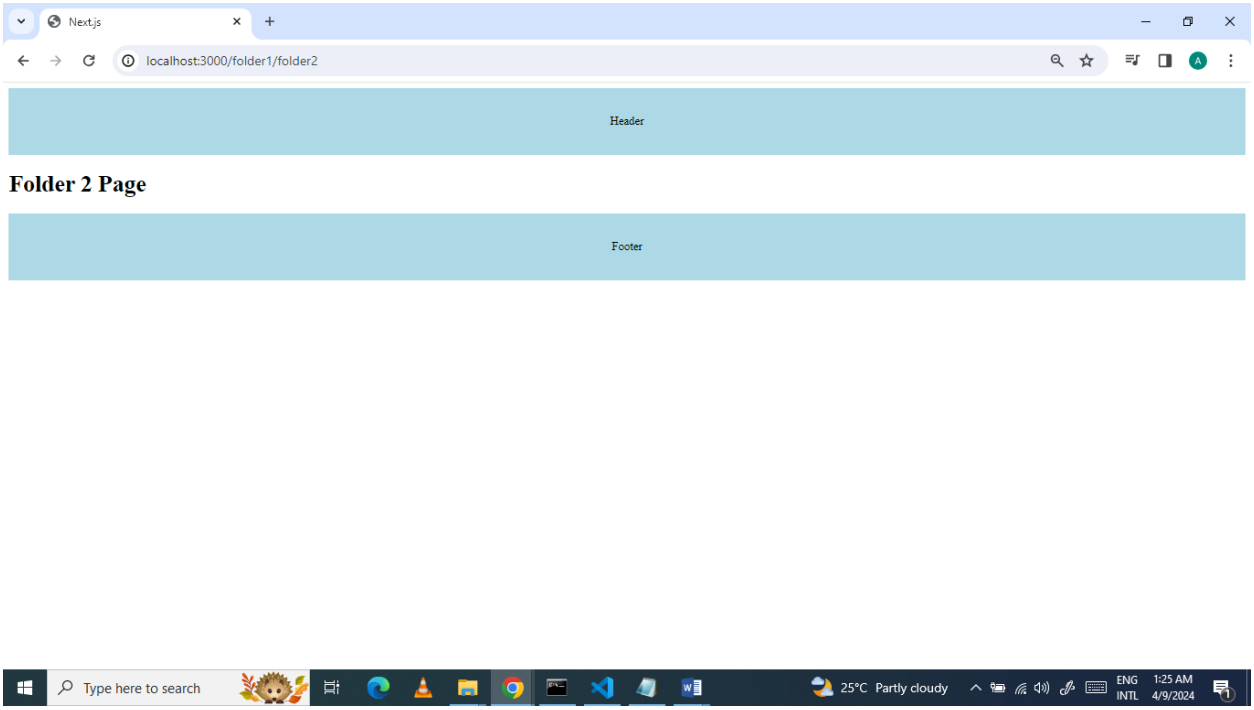
"Update the page.tsx file inside folder1 as shown in the screenshots below."



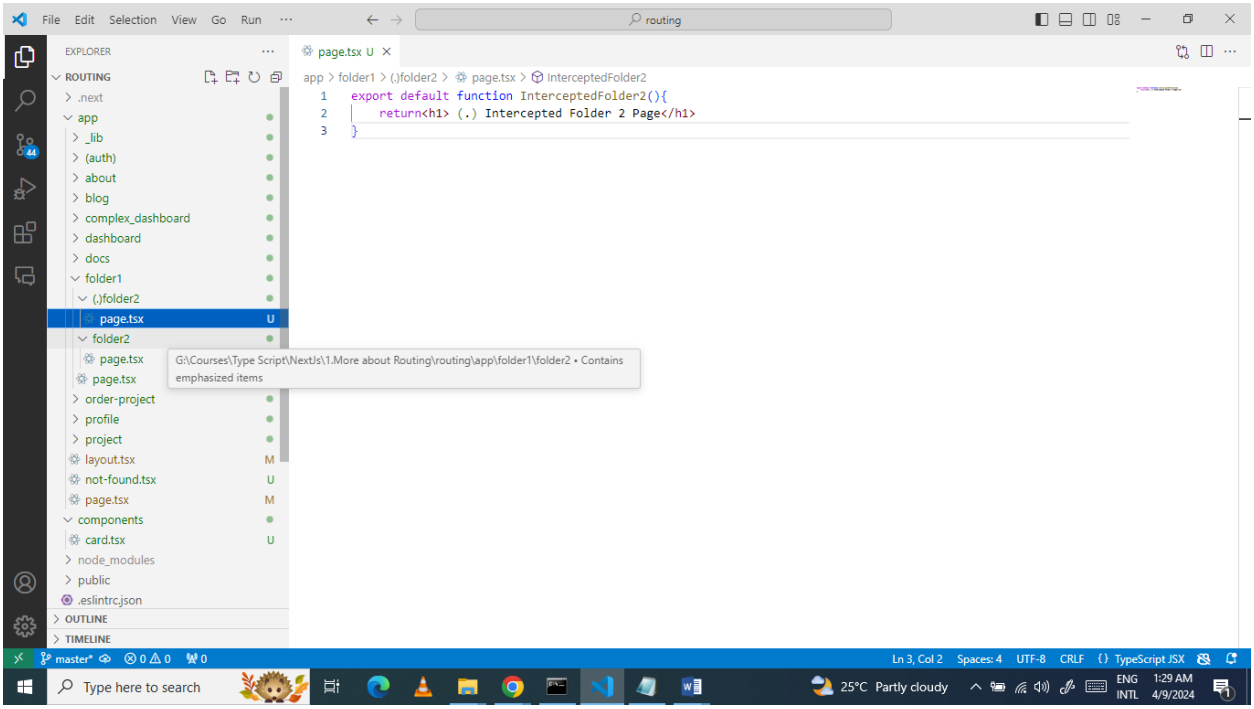
"Go to the browser, navigate to http://localhost:3000/folder1, and you will see the link to Folder2 below the Folder1 page."



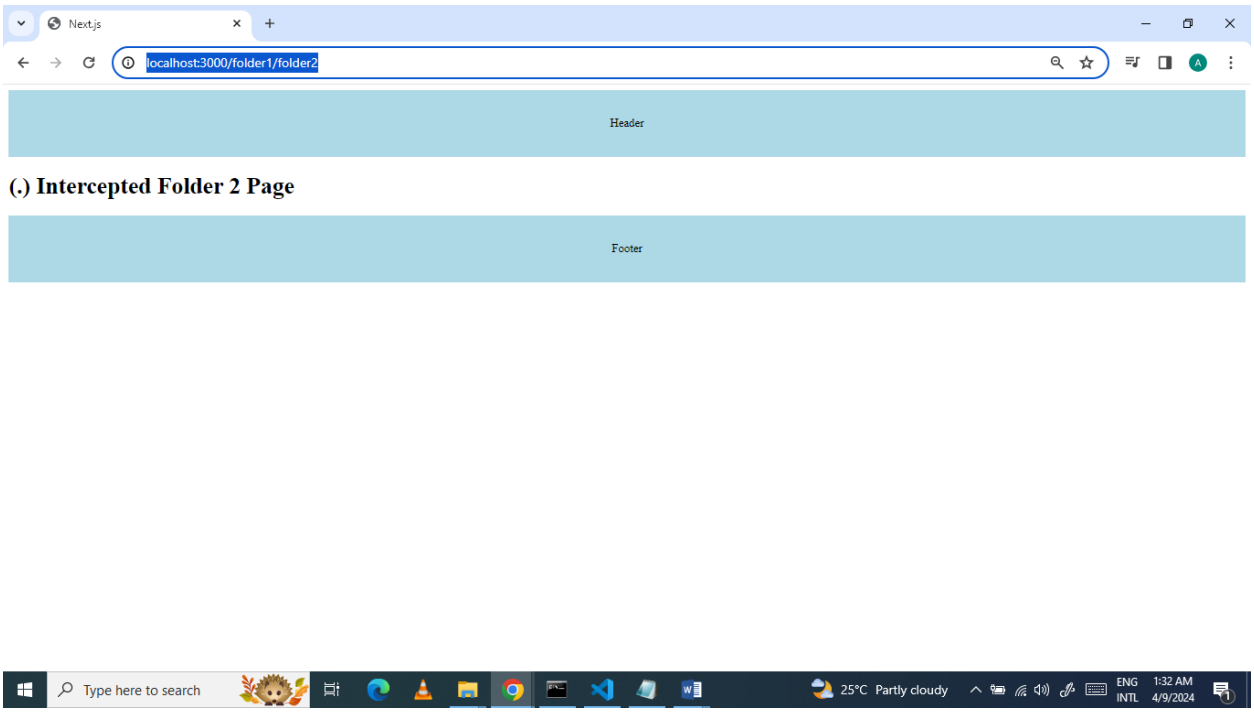
"Now, click on the 'Folder2' link. It will open the 'Folder2' page, and after refreshing the page multiple times, you will see the same page."



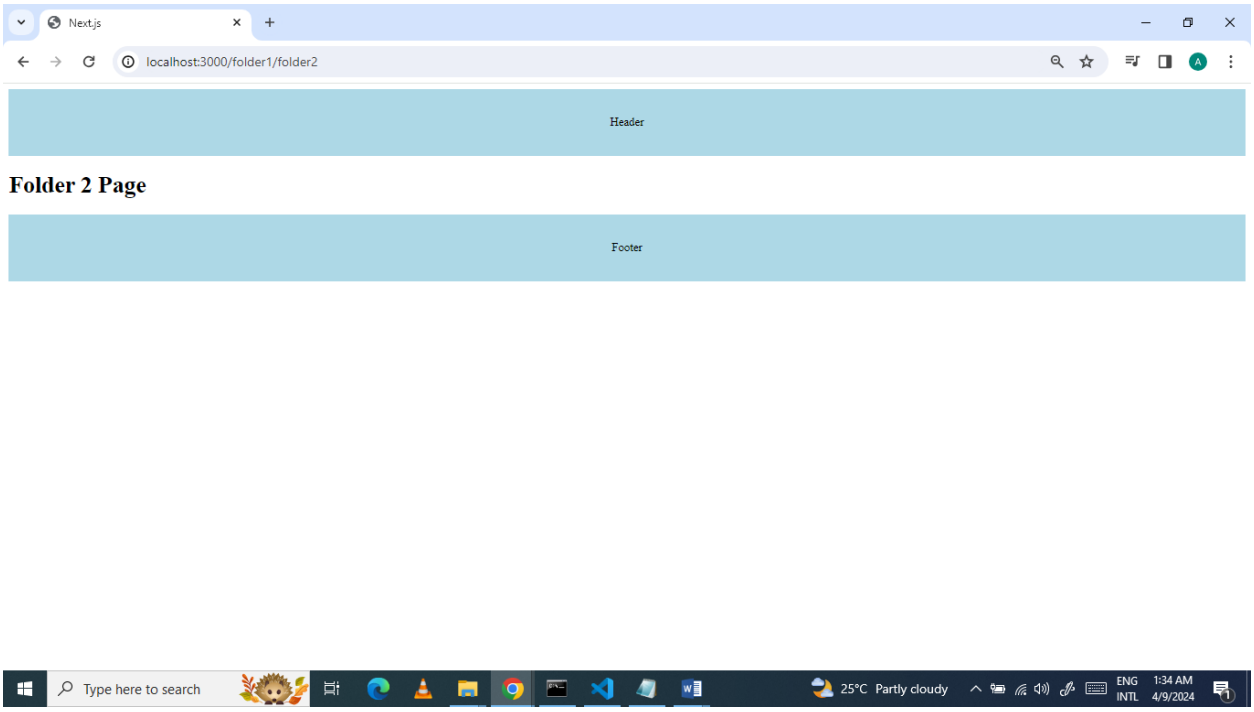
"Create a folder inside folder1 named (.)folder2. Inside (.)folder2, create a page.tsx file and update it as per the screenshots below."



Go to the browser, navigate to http://localhost:3000/folder1, and you will see the "Folder2" link below the "Folder1" page. Click on the "Folder2" link, and it will open the Intercepted Folder2 page as shown in the screenshots below.

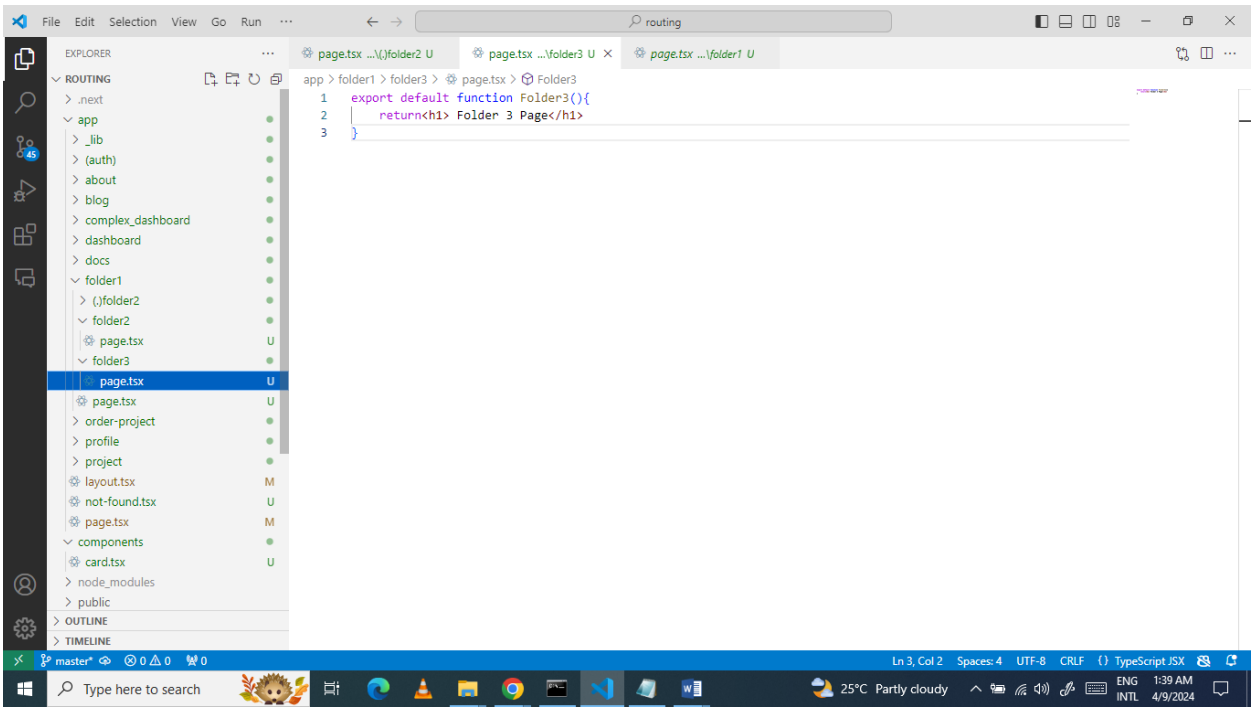


When you refresh the page or reload it, the Folder 2 page will open, as shown in the screenshots below.

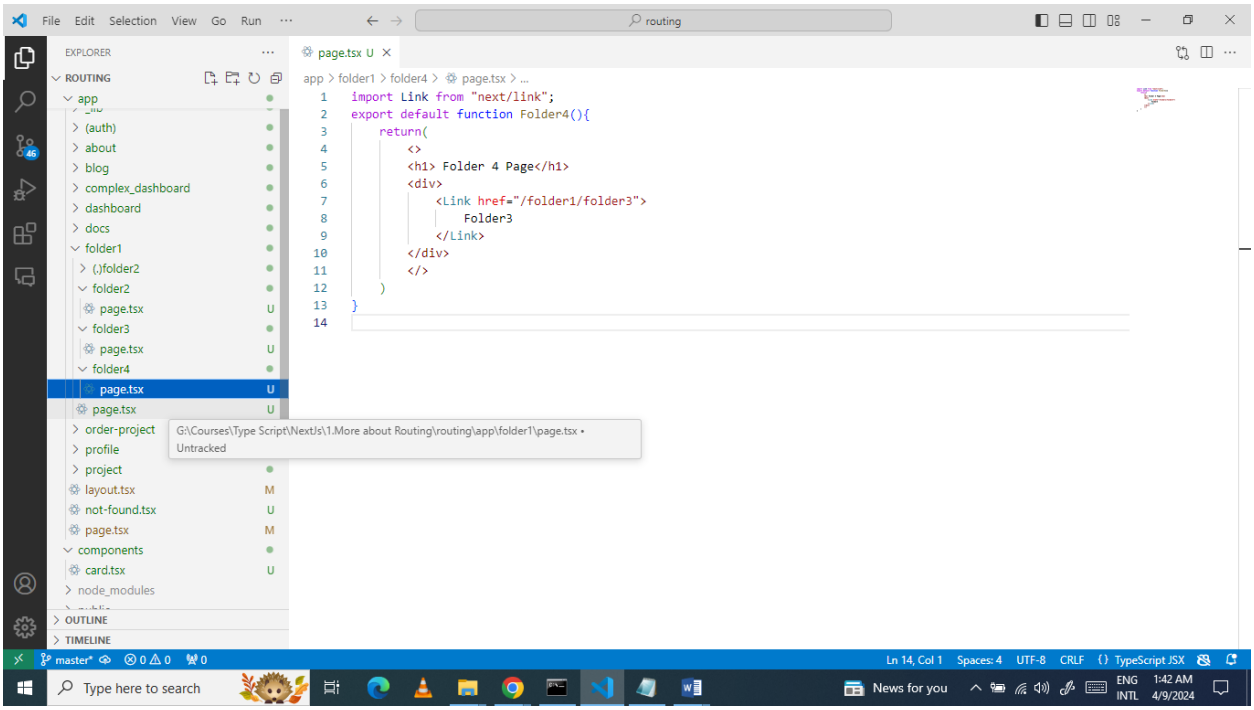


The implementation of the convention using two dots within parentheses (..) for intercepting routes in Next.js

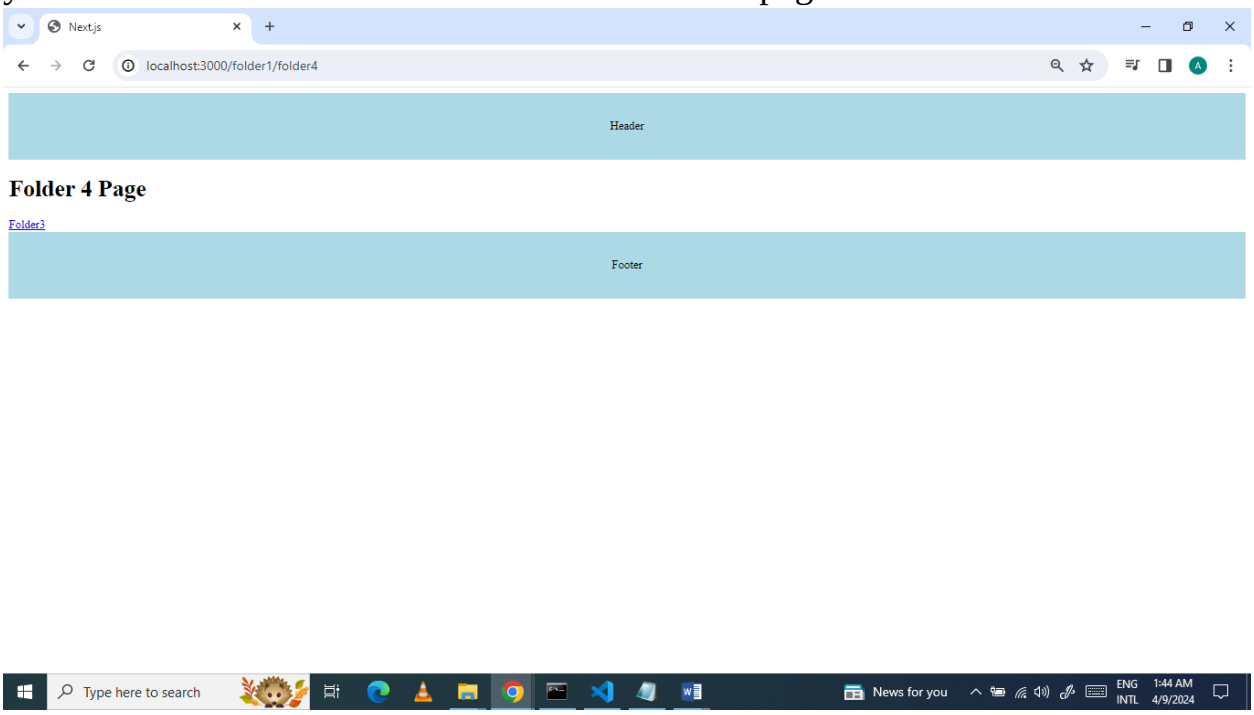
Inside `folder1`, create another folder named `folder3`. Create a `page.tsx` file inside `folder3`.



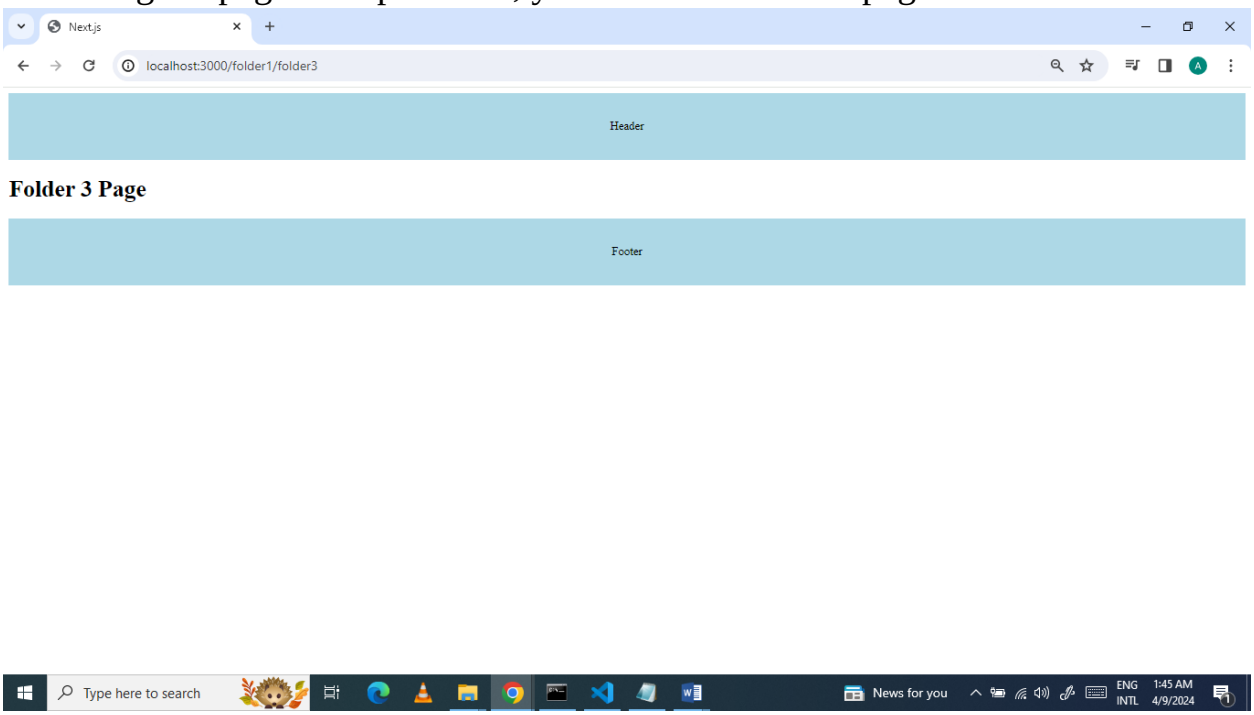
Inside folder1, create another folder named folder4. Create a page.tsx file inside folder4.



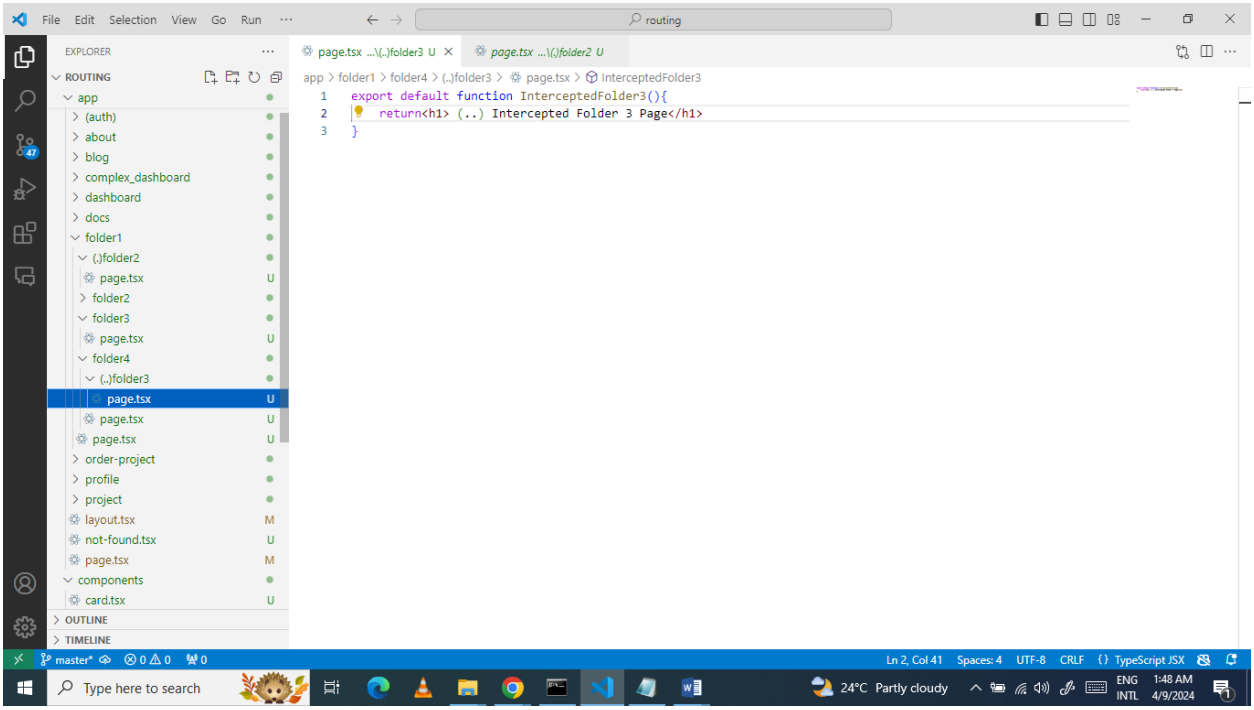
"Go to the browser, navigate to http://localhost:3000/folder1/folder4, and you will see the link to Folder3 below the Folder4 page."



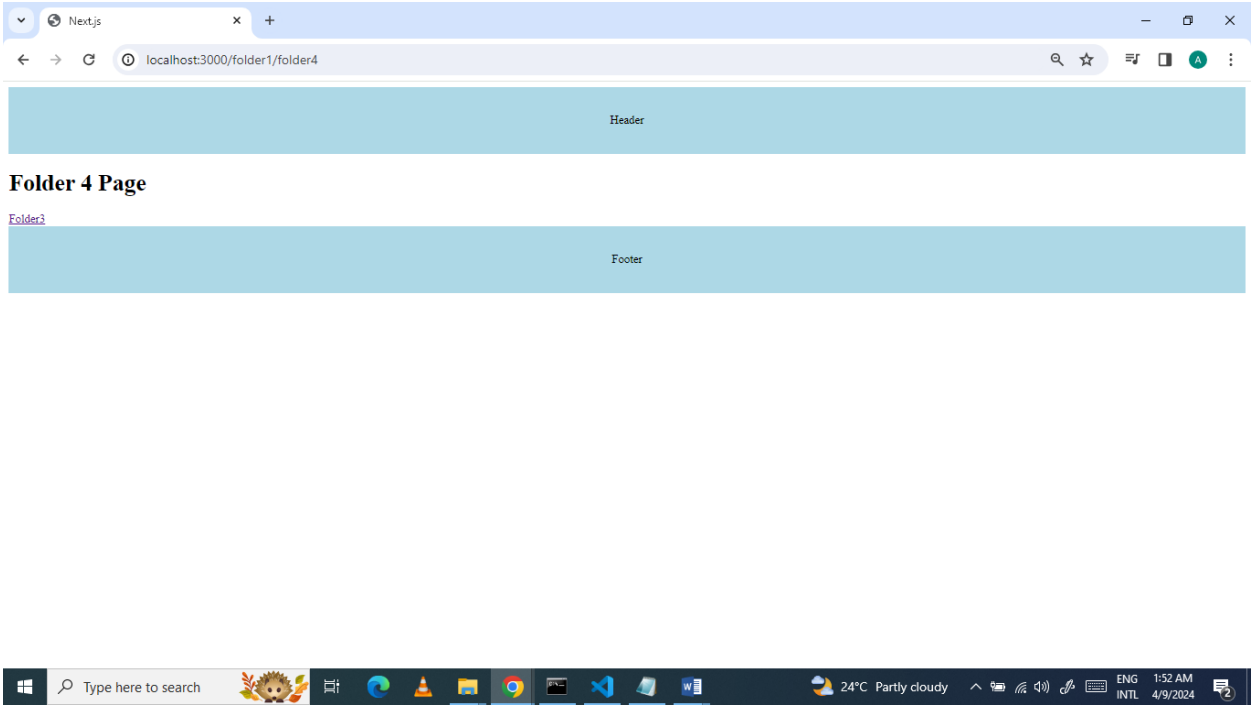
"Now, click on the Folder3 link. It will open the Folder3 page, and after refreshing the page multiple times, you will see the same page."

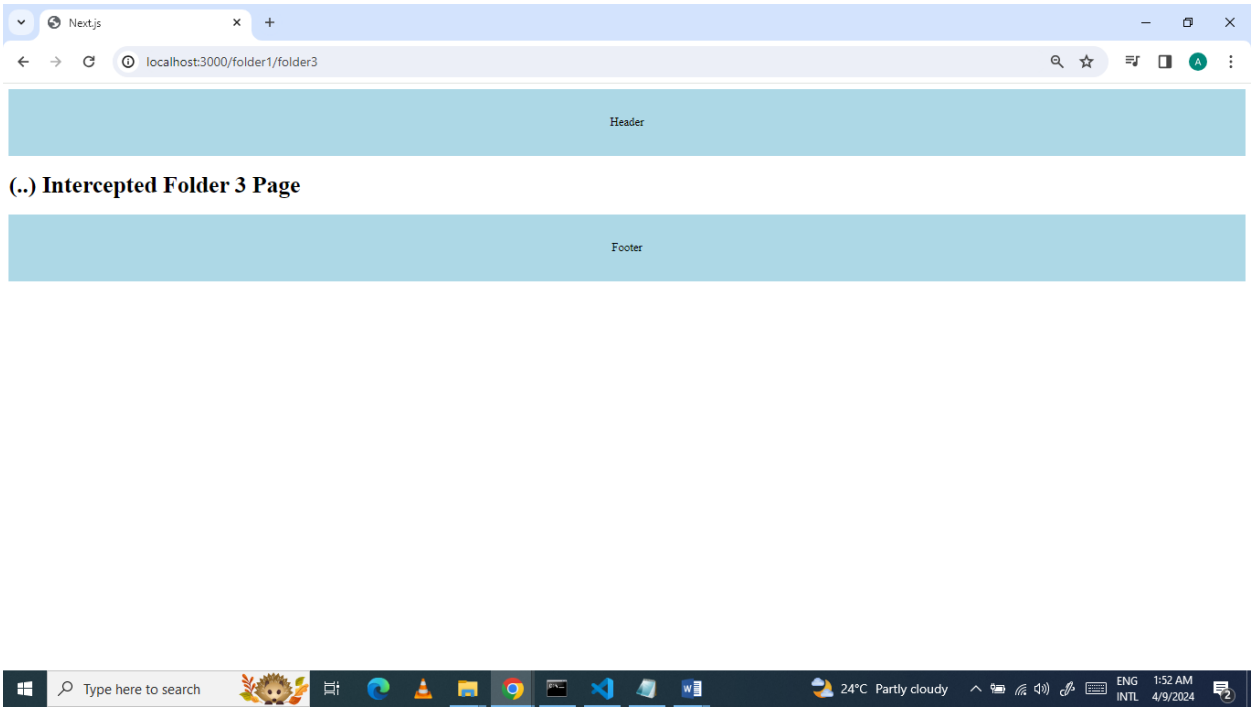


"Create a folder inside folder4 named (..)folder3. Inside (..)folder3, create a page.tsx file and update it as per the screenshots below."

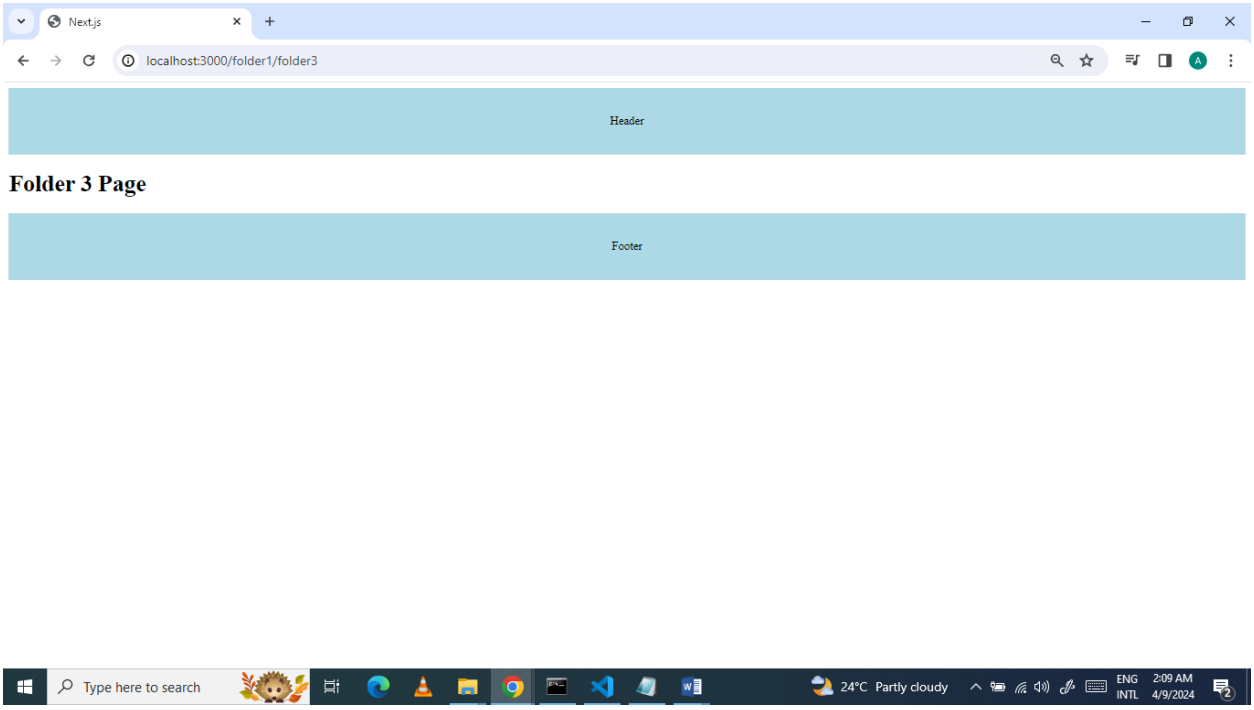


Go to the browser, navigate to http://localhost:3000/folder1/folder4, and you will see the "Folder3" link below the "Folder4 Page". Click on the "Folder3" link, and it will open the intercepted "Folder3" page as shown in the screenshots below.



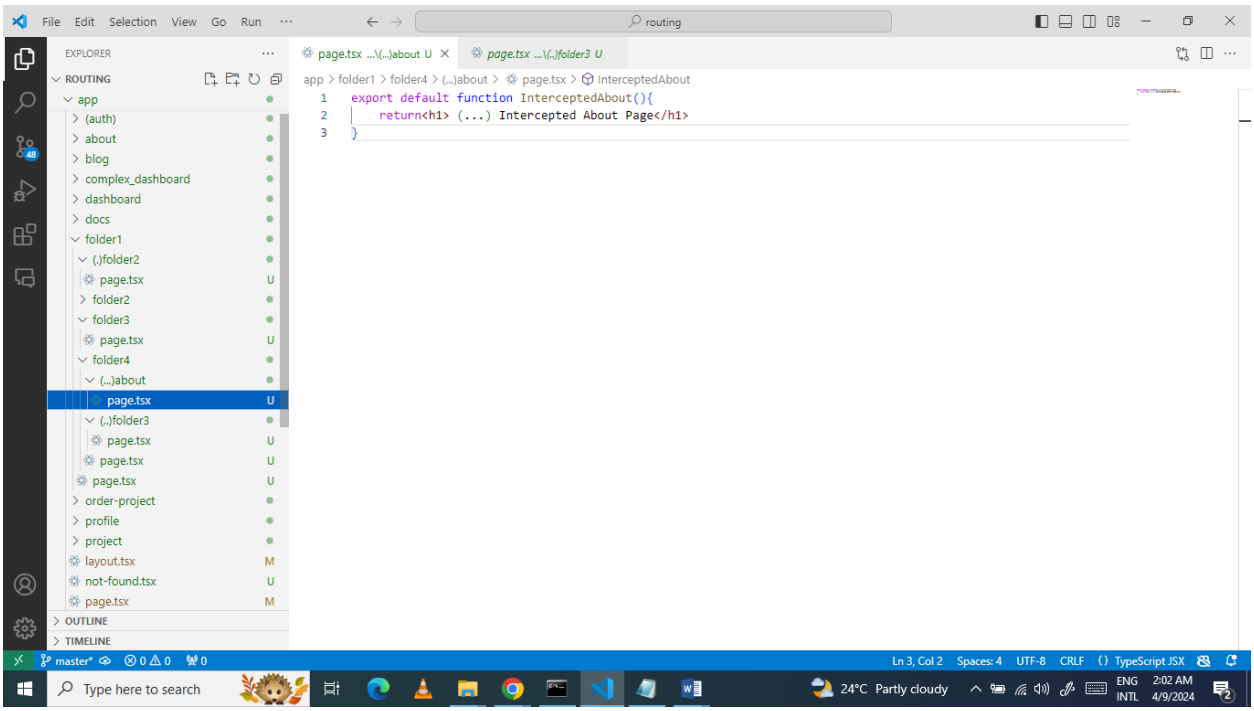


"When you refresh or reload the page, the Folder 3 page will open as shown in the screenshots below."

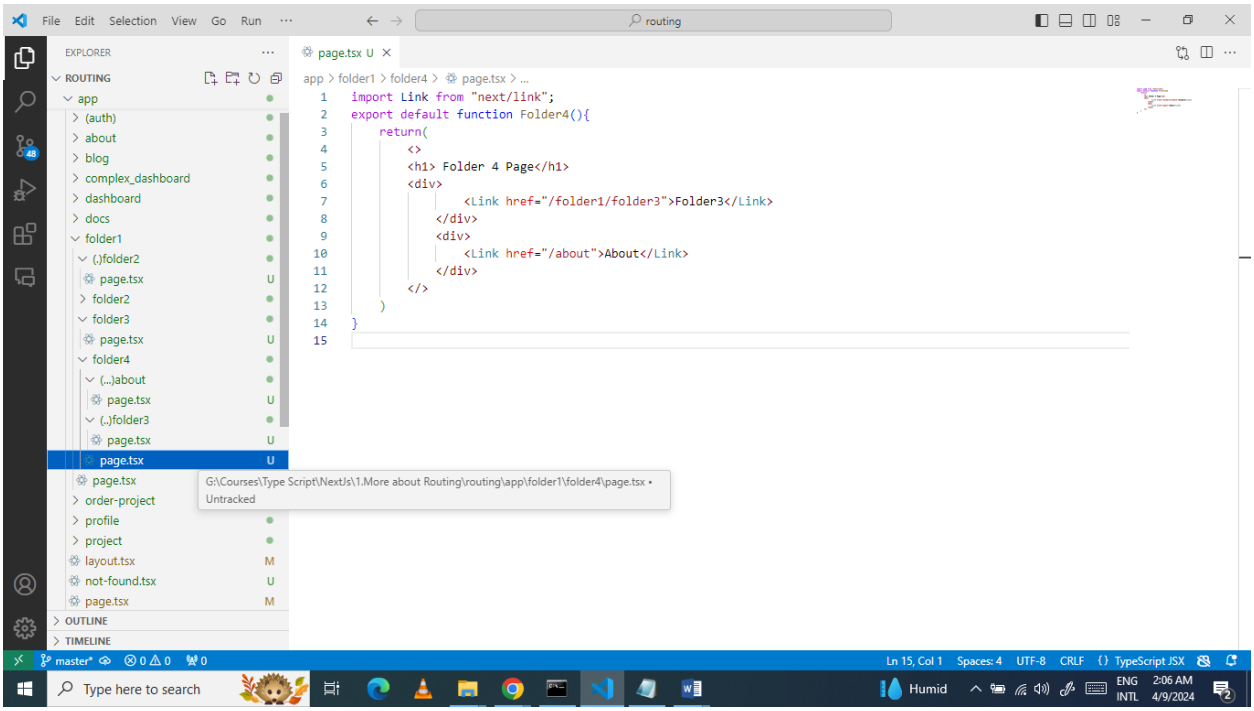


The implementation of the convention using three dots within parentheses (…) for intercepting routes in Next.js

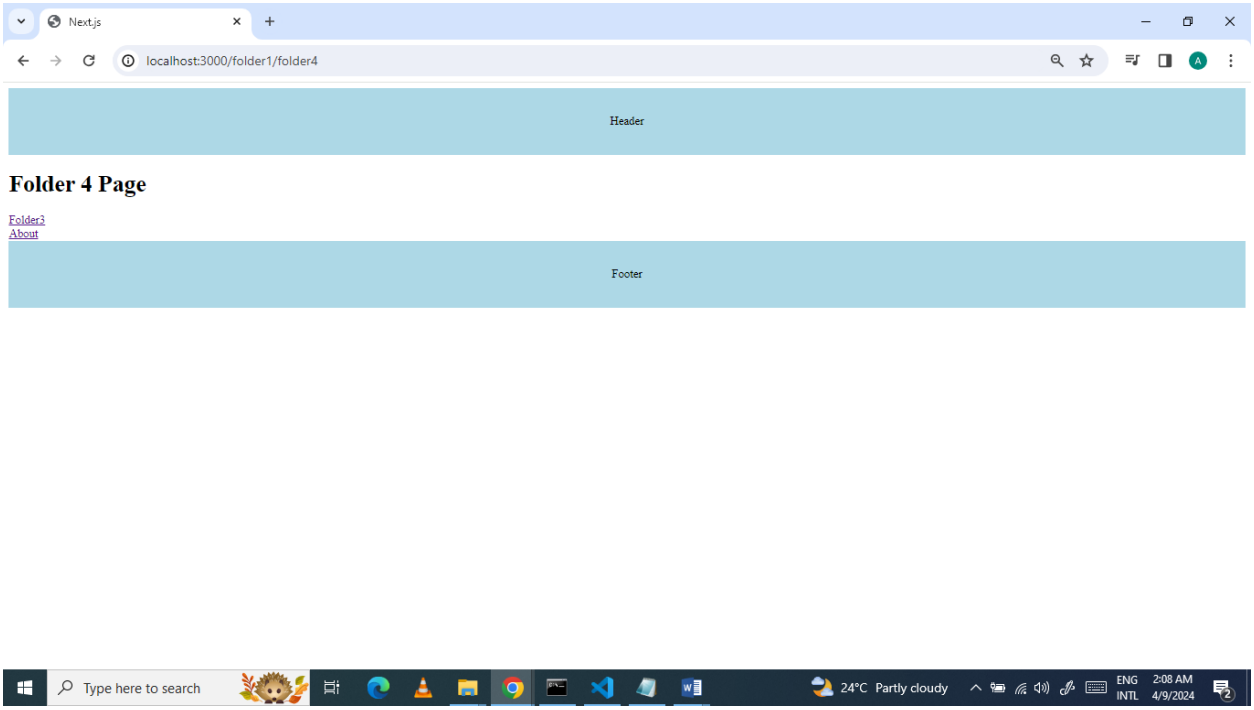
Create a folder inside "folder4" named "(...)about". Inside "(...)about", create a "page.tsx" file and update it as per the screenshots below.

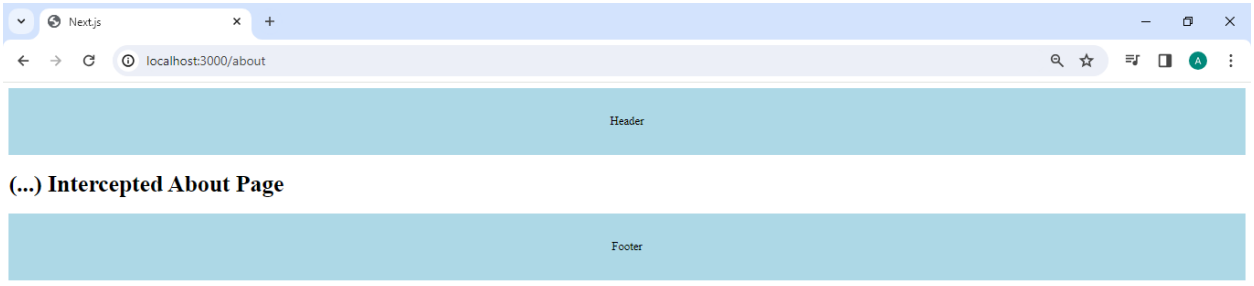


"Update the page.tsx file inside Folder4."

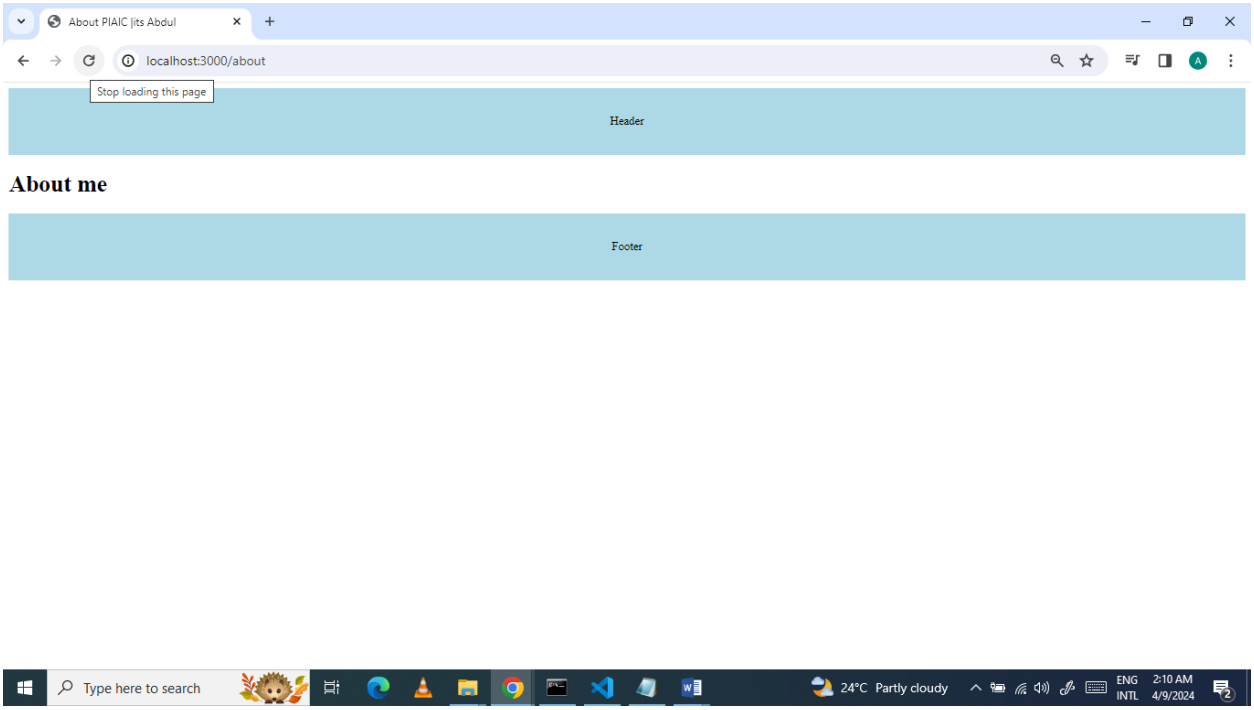


Go to the browser, navigate to http://localhost:3000/folder1/folder4, and you will see the "About" link below the "Folder3" link. Click on the "About" link. It will open the Intercepted About page, as shown in the screenshots below.





"When you refresh or reload the page, the About page will open as shown in the screenshots below."



In summary, intercepting routes enable you to incorporate a route from another section of your application into the current layout. This routing approach proves beneficial when you aim to present route content without requiring users to switch to a separate context.