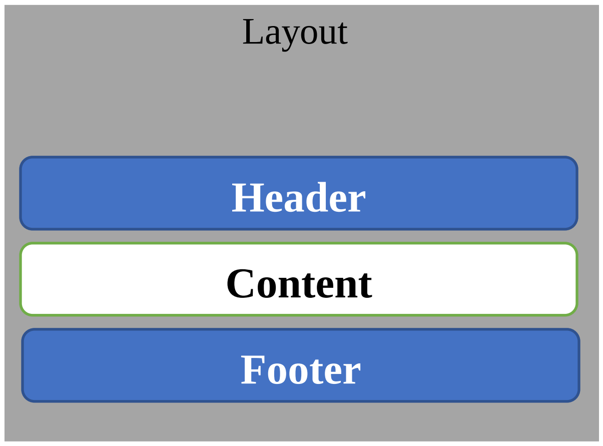
Layout in Next.js

What is page

In Next.js, a "page" refers to the specific content or UI (user interface) that is associated with a particular route in your application. Each page corresponds to a specific URL path, and when a user navigates to that URL, they see the content defined by the corresponding page component

What's a Layout?

A layout is like a template for the pages on your website or app. It includes common elements that you want to appear on multiple pages, such as a header, footer, or sidebar.



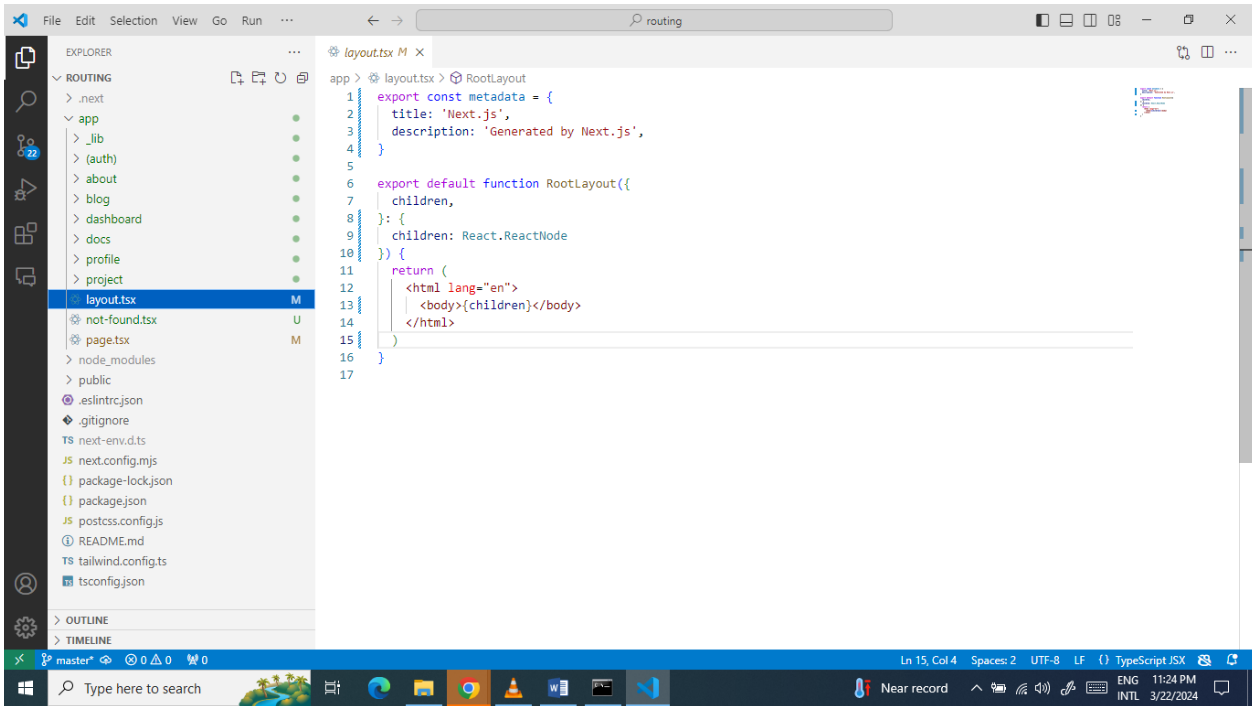
How to Create Layouts

You can define a layout by default exporting a React component from a layout.tsx file.

Inside layout.tsx, create a React component that accepts a children prop. This prop will be populated with the child components that are wrapped by the layout component.

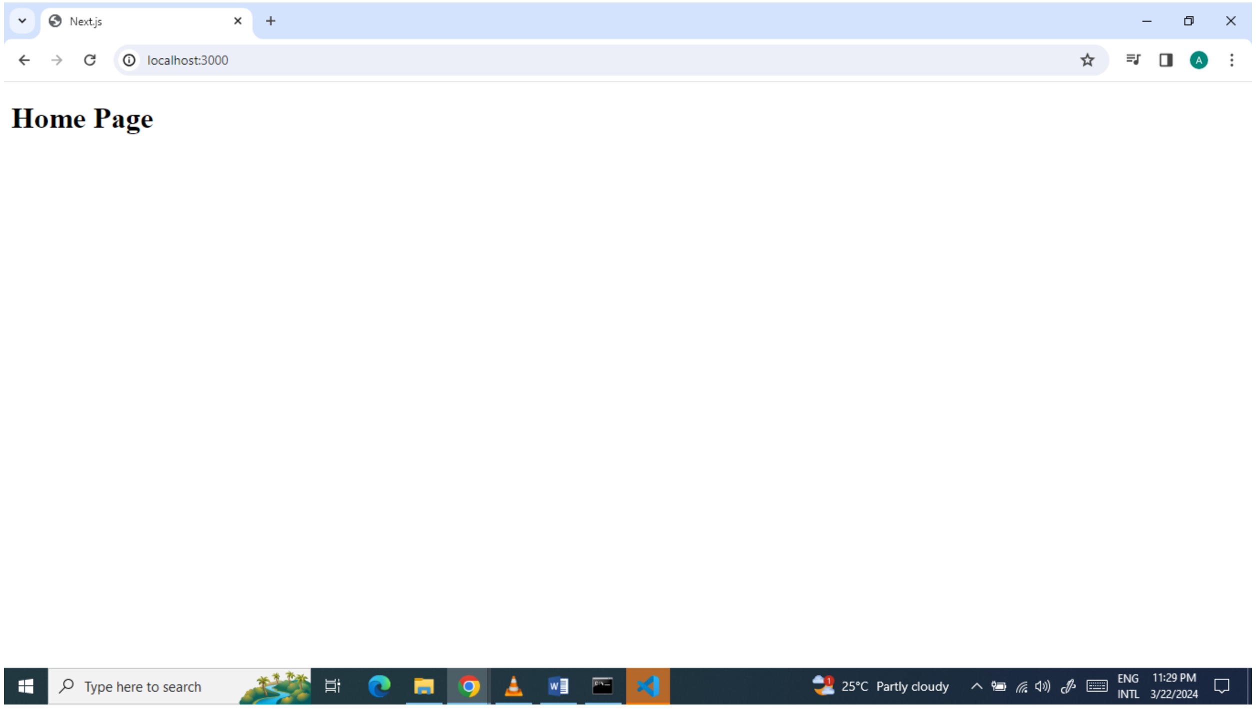
Next.js automatically serves the top-most layout as the route layout. This means that the layout component you import in your pages will be applied to those pages by default.

Ensure that every layout component you create accepts a children prop. This prop will be populated with the child components passed to the layout component.



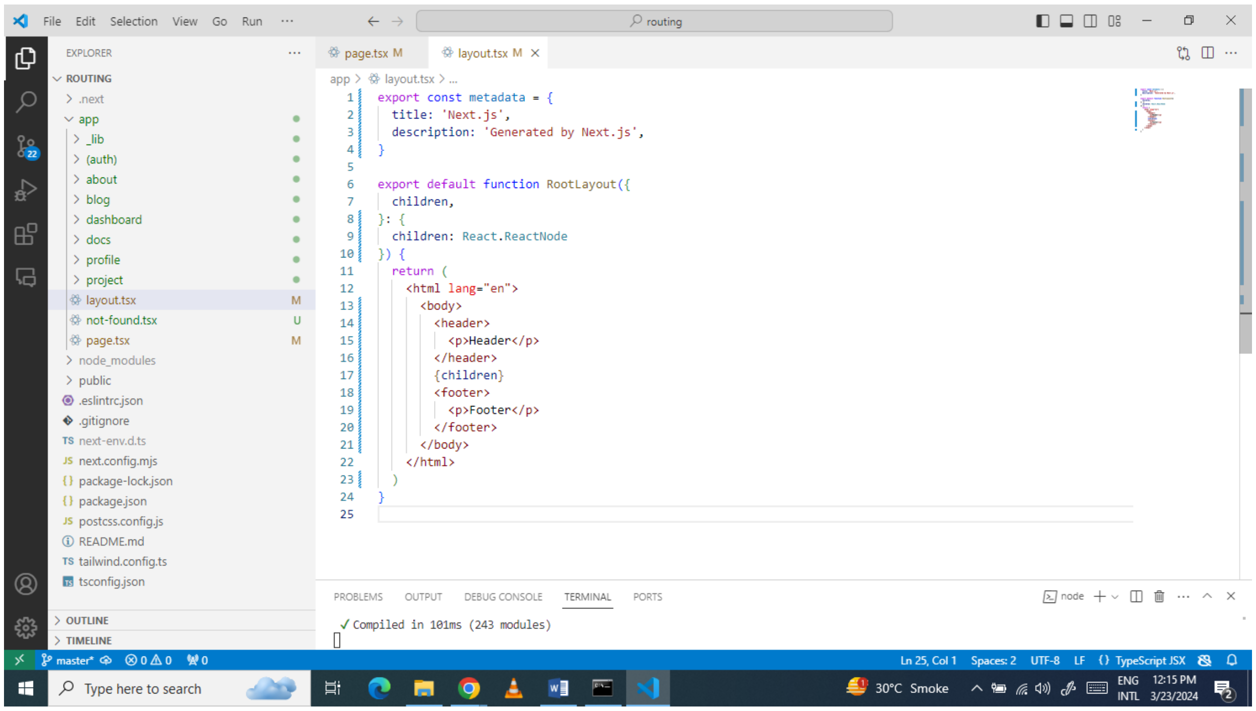
```tsx
export const metadata = {
  title: 'Next.js',
  description: 'Generated by Next.js',
}

export default function RootLayout({
  children,
}: {
  children: React.ReactNode
}) {
  return (
    <html lang="en">
      <body>{children}</body>
    </html>
  )
}
```

If you haven't made any changes to your layout.tsx file and you run the **npm run dev** command in your Next.js project, the output you'll see will depend on what's written in your page.tsx file.



Home Page

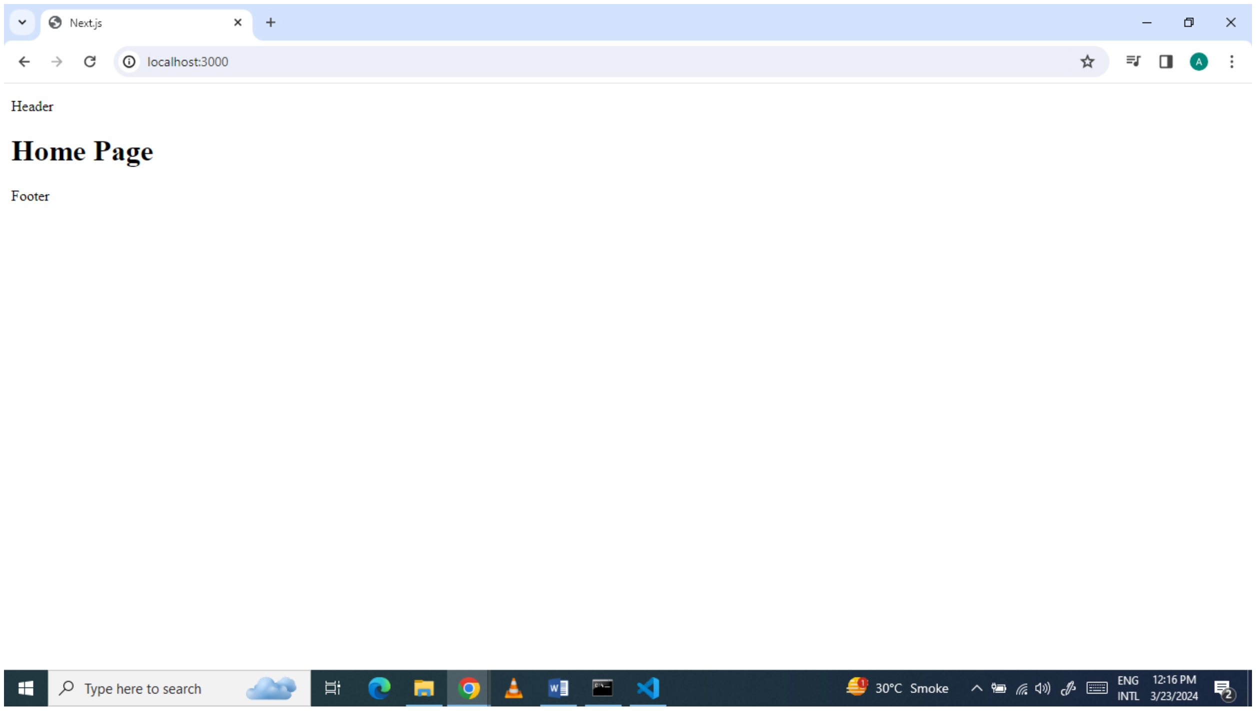
Now make some changes to the layout.tsx file to add a header and footer.



```tsx
export const metadata = {
  title: 'Next.js',
  description: 'Generated by Next.js',
}

export default function RootLayout({
  children,
}: {
  children: React.ReactNode
}) {
  return (
    <html lang="en">
      <body>
        <header>
          <p>Header</p>
        </header>
        {children}
        <footer>
          <p>Footer</p>
        </footer>
      </body>
    </html>
  )
}
```

```
✓ Compiled in 101ms (243 modules)
```

Output page will be:



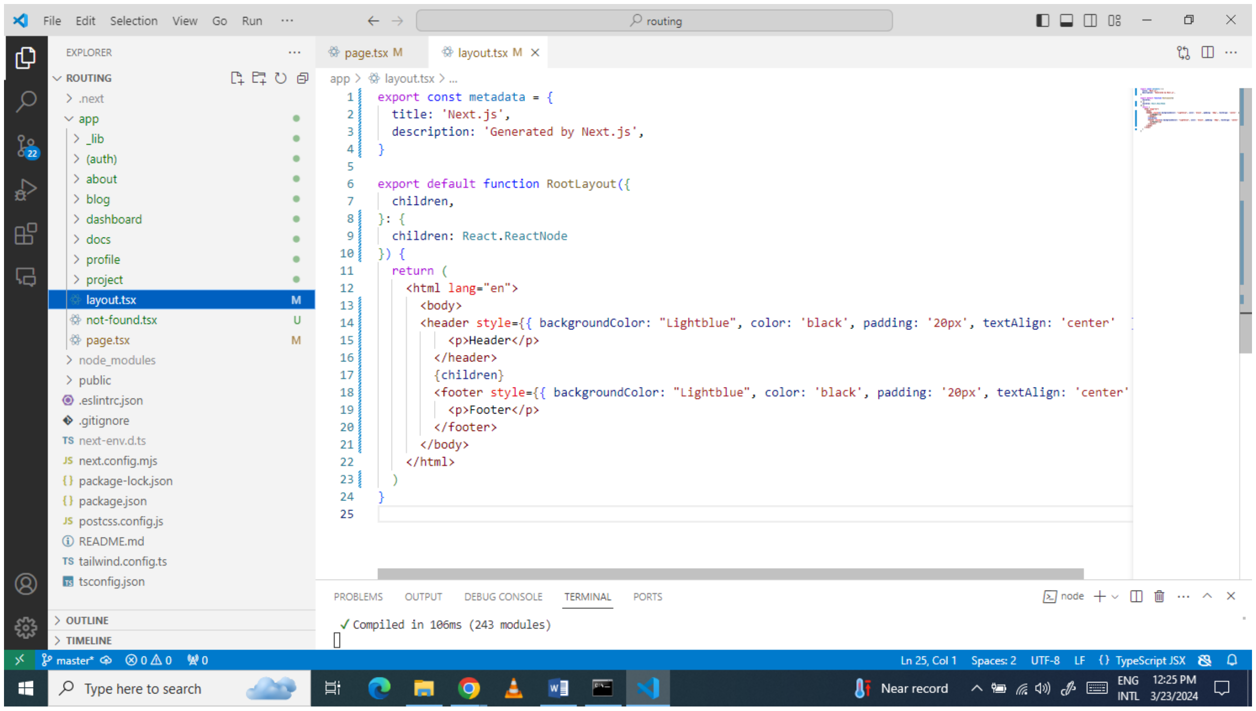
localhost:3000

Header

Home Page

Footer

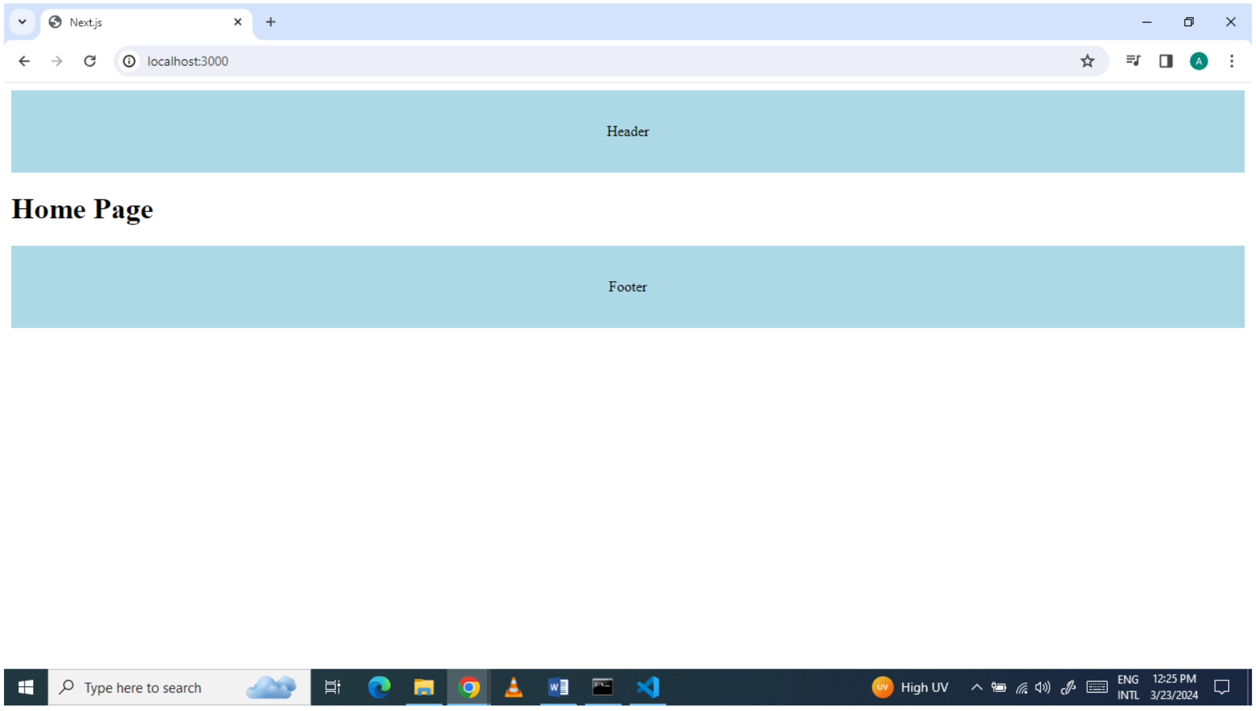
Now make some changes to the layout.tsx file to add a header and footer along with some basic styling:



```
export const metadata = {
  title: 'Next.js',
  description: 'Generated by Next.js',
}

export default function RootLayout({
  children,
}: {
  children: React.ReactNode
}) {
  return (
    <html lang="en">
      <body>
        <header style={{ backgroundColor: "Lightblue", color: 'black', padding: '20px', textAlign: 'center'
          <p>Header</p>
        </header>
        {children}
        <footer style={{ backgroundColor: "Lightblue", color: 'black', padding: '20px', textAlign: 'center'
          <p>Footer</p>
        </footer>
      </body>
    </html>
  )
}
```

```
✓ Compiled in 106ms (243 modules)
```

With these changes, your Layout component will have a header and footer with some basic styling. Now, when you run **npm run dev** and navigate to your application in the browser, any page that uses this layout (by wrapping its content with the Layout component) will display the header and footer according to the styles defined.



Header

Home Page

Footer