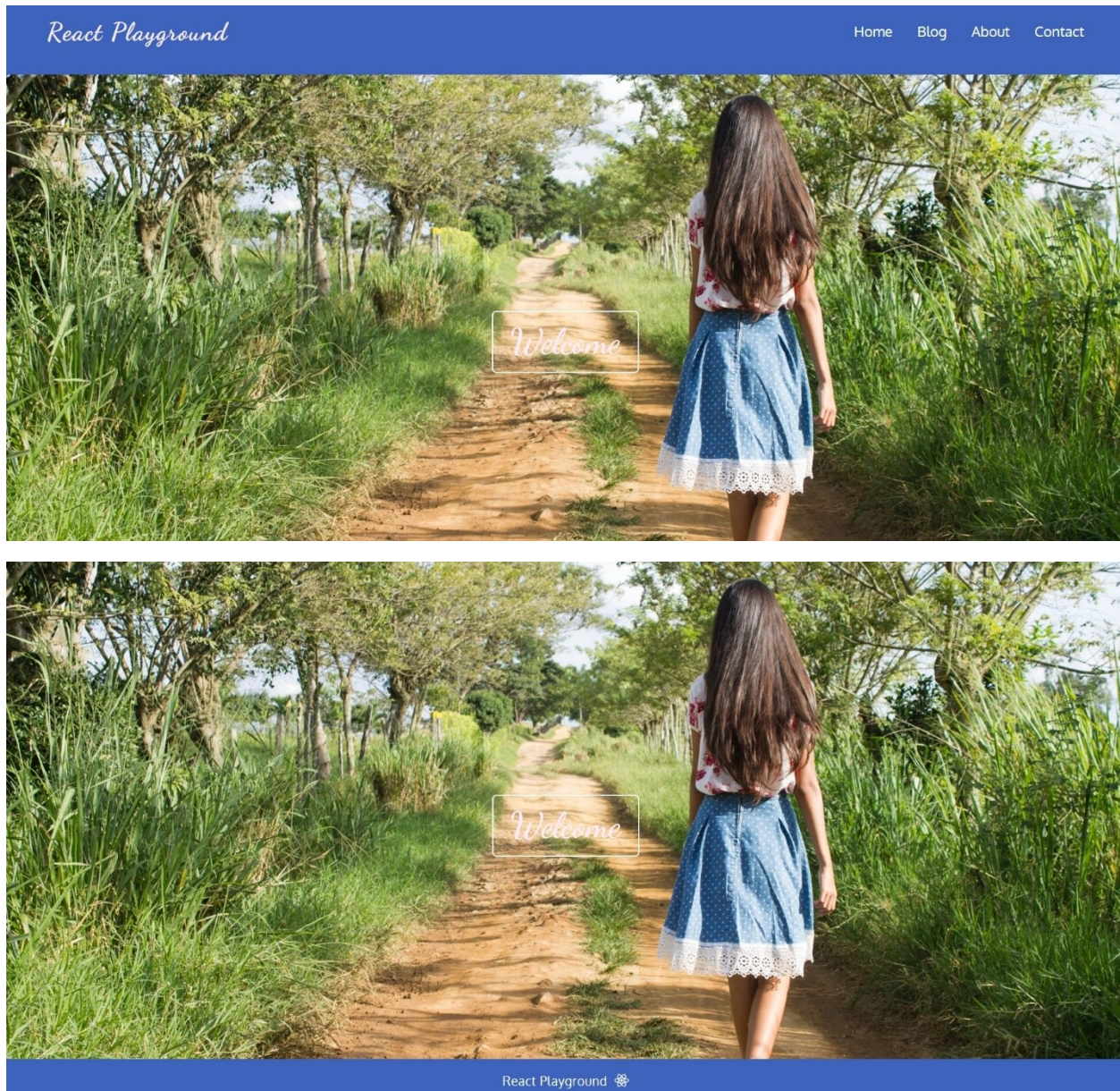


HOW TO NOT RENDER A COMPONENT ON A PARTICULAR PAGE IN REACT

The problem: Layout components rendering on pages you don't want them to.

I was recently working on a project in React, and I had two different component folders: Pages and Layout. Inside the Layout folder, I had my Navbar component and my Footer component. The Navbar and Footer both rendered nicely on all my pages, which is what I wanted. However, I wanted my landing page to have a different layout and did not want to have the Navbar and Footer on my landing page. But after creating my landing page, sure enough, the Navbar and Footer were there.



I know that there are a number of ways to approach this. You can see a variety of approaches on Stack Overflow, as I did. I also tried different approaches, but did not succeed. I was able to find a solution to this with the help of React Helmet, a reusable React component that is used to manage changes to the document head. I like React Helmet, and I was already using it on this project to manage the titles that appear in the browser tabs.

So, as I was already using React Helmet, I decided to try to use it for this too. The code below is a simplified version of the code for my landing page component (I removed the code for the component, since that is not the focus here), which I called Home.js.

After installing React Helmet, I then brought it into my landing page, Home.js (line 3). Wrapping the title element inside the Helmet element (lines 9-11) allows us to give a title that will appear on the tab in the browser. This is what I was already using React Helmet for.

```
1 import React from 'react';
2 import { Container, Row, Col } from 'react-bootstrap';
3 import { Helmet } from 'react-helmet';
4 import background from '../img/background.webp';
5
6 const Home = () => {
7   return (
8     <div>
9       <Helmet>
10        <title>React Playground</title>
11      </Helmet>
12
13      <Container className='container' fluid>
14        <Row>
15          <Col md={12}>
16
17            CODE GOES HERE
18
19          </Col>
20        </Row>
21      </Container>
22    </div>
23  );
24 };
25
26 export default Home;
```

In order to deal with my component issue, I was able to use React Helmet to add CSS to only this component (lines 7-12). Specifically, I used the "display" property and set that to "none" for both the Navbar and Footer components (lines 8-10). And that's it. That fixed it. If I had done this in App.css (which is what I used for my project's global styles), then the styles would have gotten applied to all of the components, which is not what I wanted. Then the Navbar and Footer would not show on any of the pages. React Helmet allowed me to apply this CSS to only my landing page, which is what I wanted.

```
1 const Home = () => {
2   return (
3     <div>
4       <Helmet>
5         <title>React Playground</title>
6
7         <style type='text/css'>{`
8           .navbar,
9           .footer {
10             display:none;
11           }
12         `}</style>
13       </Helmet>
14
15
16         CODE GOES HERE
17
18     </div>
19   );
20 };
21
22 export default Home;
```


And as you can see, I no longer have the Navbar and Footer showing in my landing page, which is what I wanted.

