**index.js:**

javascript

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

import './index.css';

ReactDOM.render(<App />, document.getElementById('root'));

**index.css:**

css

/\* Add any global styles here \*/

body {

font-family: sans-serif;

margin: 0;

padding: 0;

}

**App.js:**

javascript

import React from 'react';

import RadioButton from './RadioButton';

import './App.css';

function App() {

return (

<div className="container">

<h1>Clickable Radio Button Example</h1>

<RadioButton id="radio1" name="radios">

Option 1

</RadioButton>

</div>

);

}

export default App;

**App.css:**

css

/\* CSS styling for the App component \*/

.container {

margin: 0 auto;

padding: 20px;

max-width: 600px;

}

h1 {

text-align: center;

}

**RadioButton.js:**

javascript

import React, { useState } from 'react';

import './RadioButton.css';

function RadioButton({ id, name, children }) {

const [checked, setChecked] = useState(false);

const handleClick = () => {

setChecked(!checked);

};

return (

<label className="radio-button" onClick={handleClick}>

<input type="radio" id={id} name={name} checked={checked} readOnly />

<div className="radio-circle">

<div className={`radio-dot ${checked ? 'checked' : ''}`}></div>

</div>

{children}

</label>

);

}

export default RadioButton;

**RadioButton.css:**

css

/\* CSS styling for the RadioButton component \*/

.radio-button {

display: inline-block;

position: relative;

padding-left: 25px;

margin-right: 10px;

cursor: pointer;

user-select: none;

}

.radio-button input {

position: absolute;

opacity: 0;

cursor: pointer;

}

.radio-button .radio-circle {

position: absolute;

top: 0;

left: 0;

height: 13px;

width: 13px;

border-radius: 50%;

background-color: #e0f2ff;

border: 2px solid #b3d8ff;

}

.radio-button .radio-dot {

position: absolute;

top: 4.5px;

left: 4px;

height: 5px;

width: 5px;

border-radius: 50%;

background-color: #0066ff;

display: none;

}

.radio-button label {

color: #000;

}

.radio-button input:checked ~ .radio-circle .radio-dot {

display: block;

}

.radio-dot.checked {

display: block;

}

**app.test.js:**

javascript

import React from 'react';

import { render, fireEvent } from '@testing-library/react';

import App from './App';

test('renders radio button option', () => {

const { getByText } = render(<App />);

const option = getByText('Option 1');

expect(option).toBeInTheDocument();

});

test('changes radio button style on click', () => {

const { getByText } = render(<App />);

const option = getByText('Option 1');

fireEvent.click(option);

const radioCircle = option.previousElementSibling;

const radioDot = radioCircle.querySelector('.radio-dot');

expect(radioCircle).toHaveStyle('background-color: #e0f2ff');

expect(radioCircle).toHaveStyle('border-color: #b3d8ff');

expect(radioDot).toHaveStyle('display: block');

fireEvent.click(option);

expect(radioCircle).toHaveStyle('background-color: transparent');

expect(radioCircle).toHaveStyle('border-color: #b3d8ff');

expect(radioDot).toHaveStyle('display: none');

});

Please note that the CSS styles from the original HTML code have been divided into the corresponding component-specific CSS files (RadioButton.css). The radio button's label has been omitted, but you can easily add it back based on your requirements.

why does app test need code

The app.test.js file is used to write tests for the components and functionality implemented in the App.js file. It helps ensure that the application behaves as expected and detects any potential bugs or issues.

In this case, the provided radio button code was wrapped within the RadioButton component, which is used in the App component. To test the functionality of the radio button, the app.test.js file includes two test cases: