St. Francis Institute of Technology

(Engineering College)

An Autonomous Institute, Affiliated to University of Mumbai NAAC

A+ Accredited | CMPN, EXTC, INFT NBA Accredited | ISO 9001:2015 Certified

Department of Artificial Intelligence and Machine Learning

Academic Year: 2025-2026 Term: Even (Jan. 2026 – Jun. 2026) Class / Branch: SE – AIML
Semester: IV Course: Web Programming Lab. (AI4VS_LR4)
Date of Assignment: / /2026 Date of Submission: / /2026

Recitation EXP 7

�� Exercise 1: My First Component

Goal: Understand what a component is

Task:

1. Create a functional component called Hello.
2. It should display:
3. Hello, Welcome to React!

Hint:

```
function Hello() {
 return <h1>...</h1>;
}
```

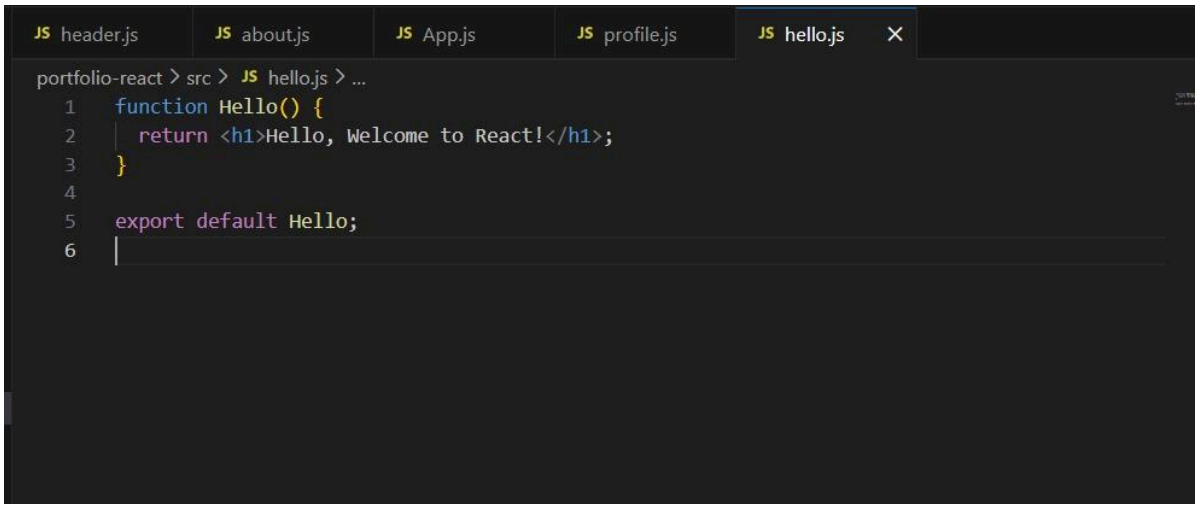
✅ Concept learned: Functional Component

Hello, Welcome to React!

header.js | about.js | App.js | profile.js | hello.js

portfolio-react > src > App.js > ...

```
1  import Hello from "./hello";
2
3  function App() {
4    return (
5      <div>
6        <Hello />
7      </div>
8    );
9  }
10
11 export default App;
12
```



�� Exercise 2: Passing Props (Name Card)

Goal: Learn Props

Task:

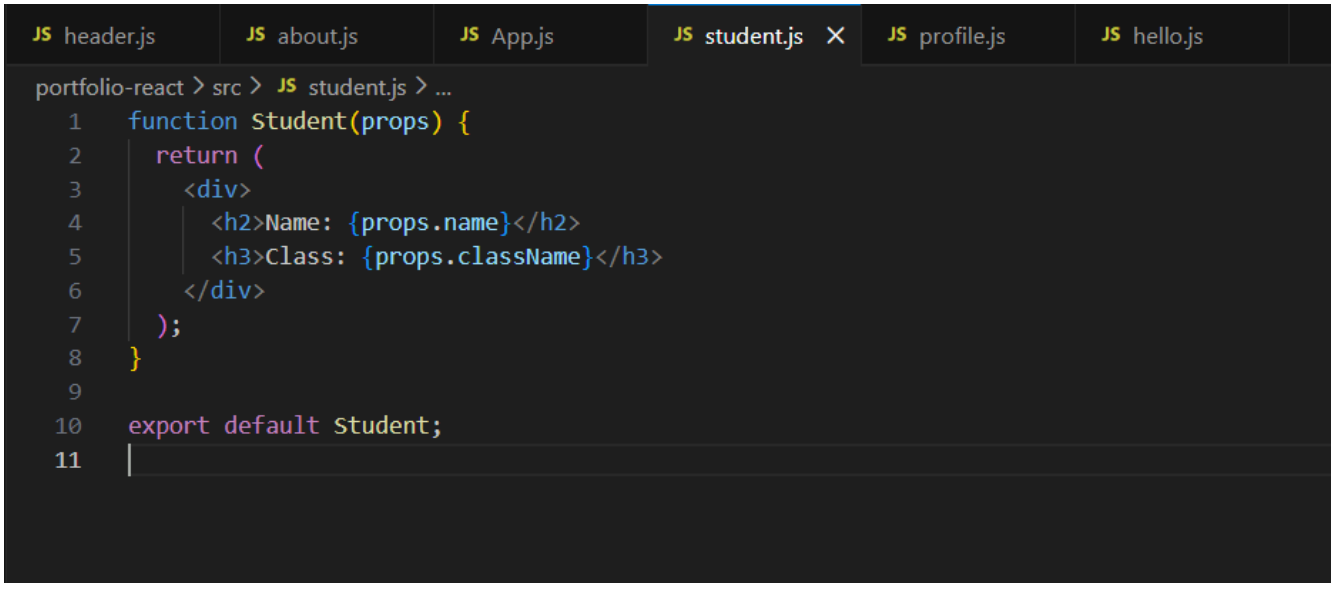
1. Create a component Student.
2. Parent component sends:
    - name
    - class
3. Child displays:
4. Name: Rahul
5. Class: 7

Hint:
<Student name="Rahul" class="7" />

✅ Concept learned: Props, Parent → Child

Name: Rahul

Class: 7



```
header.js | about.js | App.js | student.js | profile.js | hello.js
portfolio-react > src > App.js > App
1  import Student from "./student";
2
3  function App() {
4    return (
5      <div>
6        <Student name="Rahul" className="7" />
7      </div>
8    );
9  }
10
11 export default App;
12
13
```

✅ Exercise 3: Reusable Button

Goal: Learn Reusability

Learning Task:

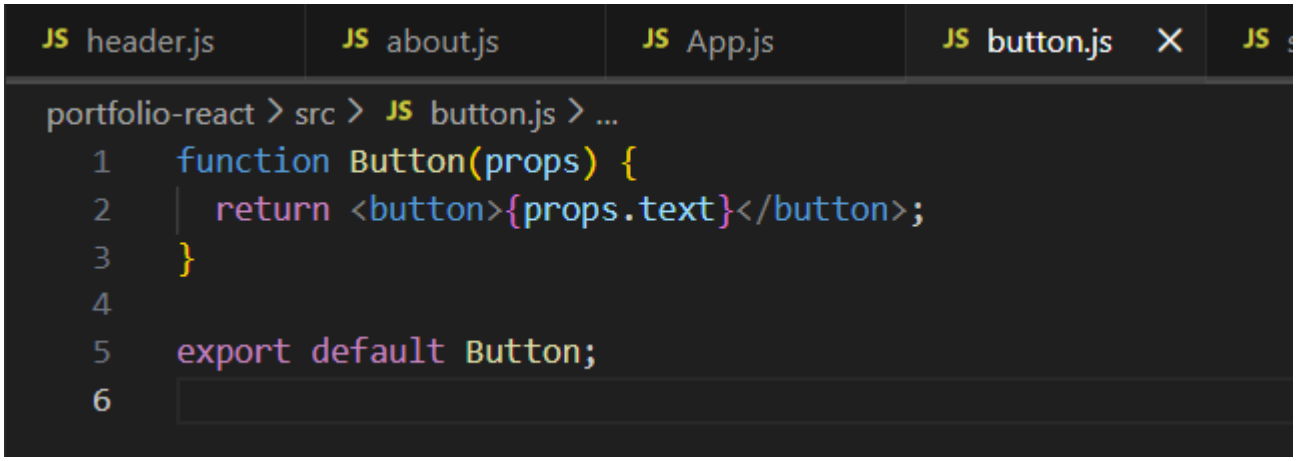
1. Create a Button component.
2. It should accept text as props.
3. Use it three times with different text:
    - Login
    - Register
    - Logout

✅ Concept learned: Reusable Components

header.js | about.js | App.js | button.js | student.j

portfolio-react > src > App.js > ...

```
1  import Button from "./button";
2
3  function App() {
4    return (
5      <div>
6        <Button text="Login" />
7        <Button text="Register" />
8        <Button text="Logout" />
9      </div>
10   );
11 }
12
13 export default App;
14
```



🧩 Exercise 4: Website Parts

Goal: Understand components as website

parts Task:

Create these components:

- Header → “My School Website”
- Content → “Welcome Students!”
- Footer → “© 2026 School”

Use them inside App.js.

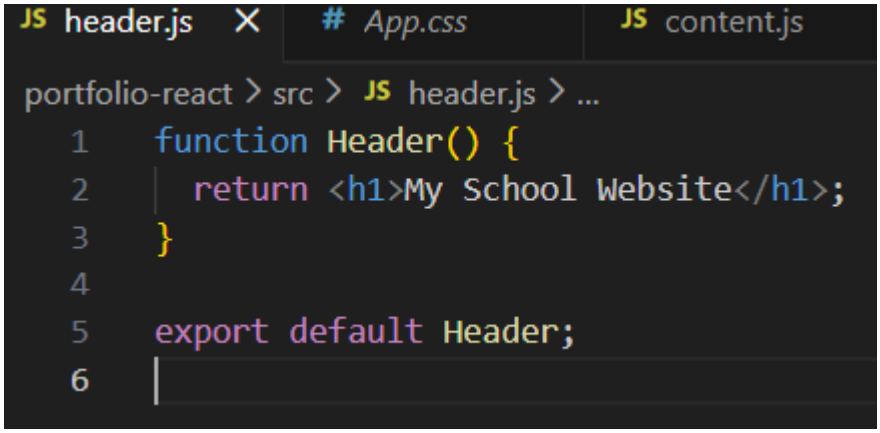
🧱 Like building a LEGO house 🏠

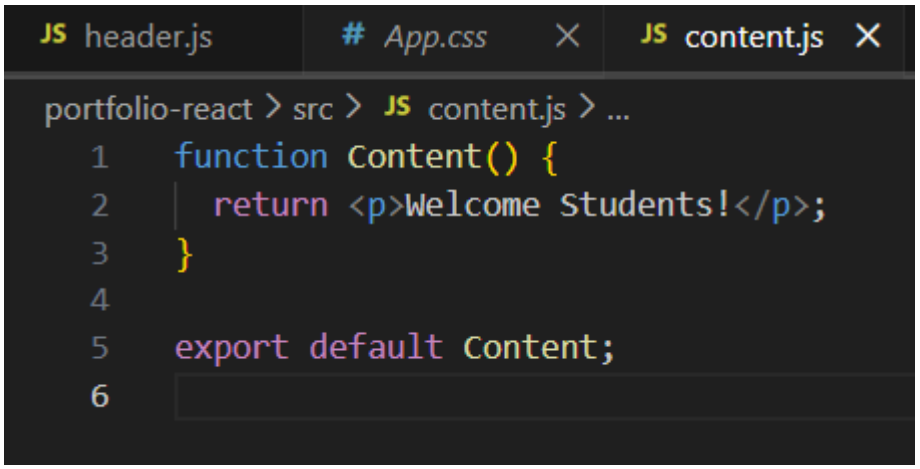
✅ Concept learned: Component structure

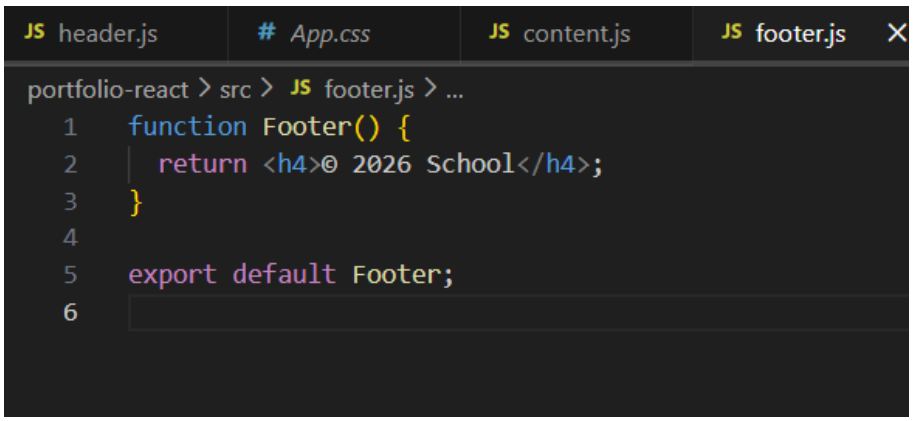
My School Website

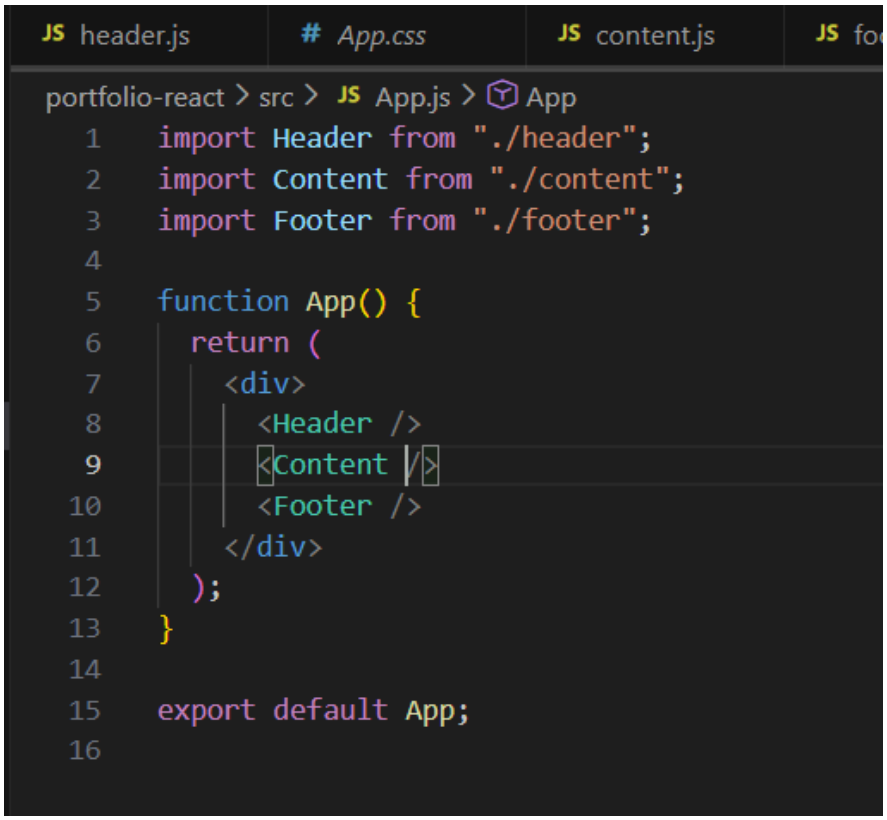
Welcome Students!

© 2026 School




header.js
App.css
content.js
portfolio-react > src > content.js > ...
function Content() {
return Welcome Students! ;
}
export default Content;


header.js
App.css
content.js
footer.js
portfolio-react > src > footer.js > ...
function Footer() {
return <h4>© 2026 School</h4>;
}
export default Footer;


header.js
App.css
content.js
portfolio-react > src > App.js > App
import Header from "./header";
import Content from "./content";
import Footer from "./footer";
function App() {
return (

<Header />
<Content />
<Footer />

);
}
export default App;

✅ Exercise 5: SPA Thinking (No Reload)

Goal: Understand SPA idea

Task:

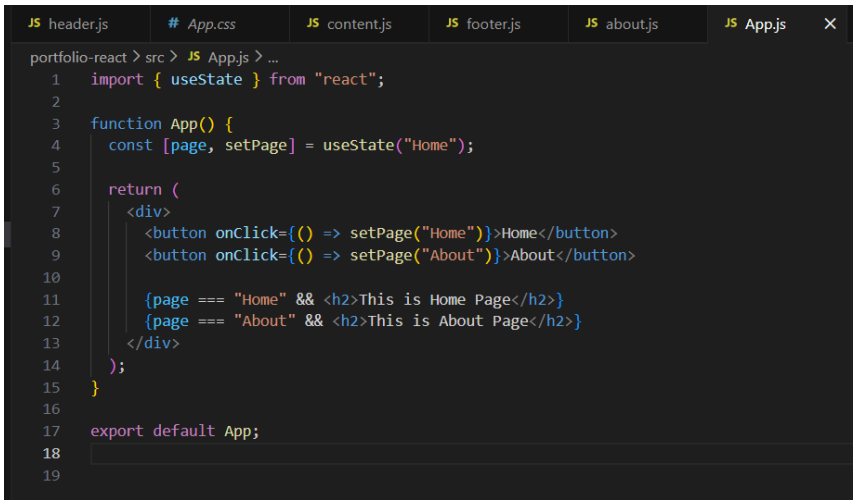
1. Create two buttons:
   - Home
   - About
2. Clicking each button should change text on the page:
   - Home → “This is Home Page”
   - About → “This is About Page”

✅ No page reload

✅ Concept learned: SPA behavior (basic)

Home About

This is Home Page



Exercise 6: Parent → Child Message

Goal: One-way data flow

Task:

1. Parent component stores a message:
2. "Study React Daily"
3. Pass it to Message child component.
4. Child displays it.

🤔 Child should NOT change the

message ✅ Concept learned: One-way

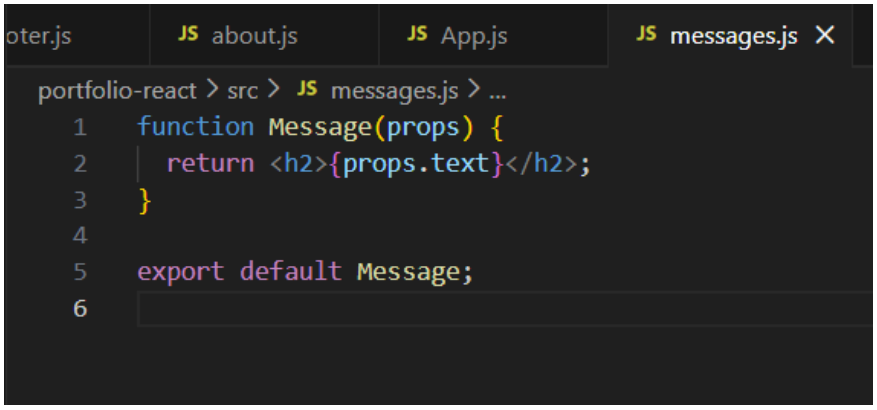
data flow

Study React Daily

ut.js | App.js | messages.js | button.js

portfolio-react > src > App.js > ...

```
1  import Message from "./messages";
2
3  function App() {
4    const message = "Study React Daily";
5
6    return (
7      <div>
8        <Message text={message} />
9      </div>
10   );
11 }
12
13 export default App;
14
```



�? Exercise 7: My Profile Card (Fun

One!) Goal: Combine everything

Task:

Create a ProfileCard component that shows:

- Name
- Age
- Favorite Subject

Use props to send data from parent.

Example Output:

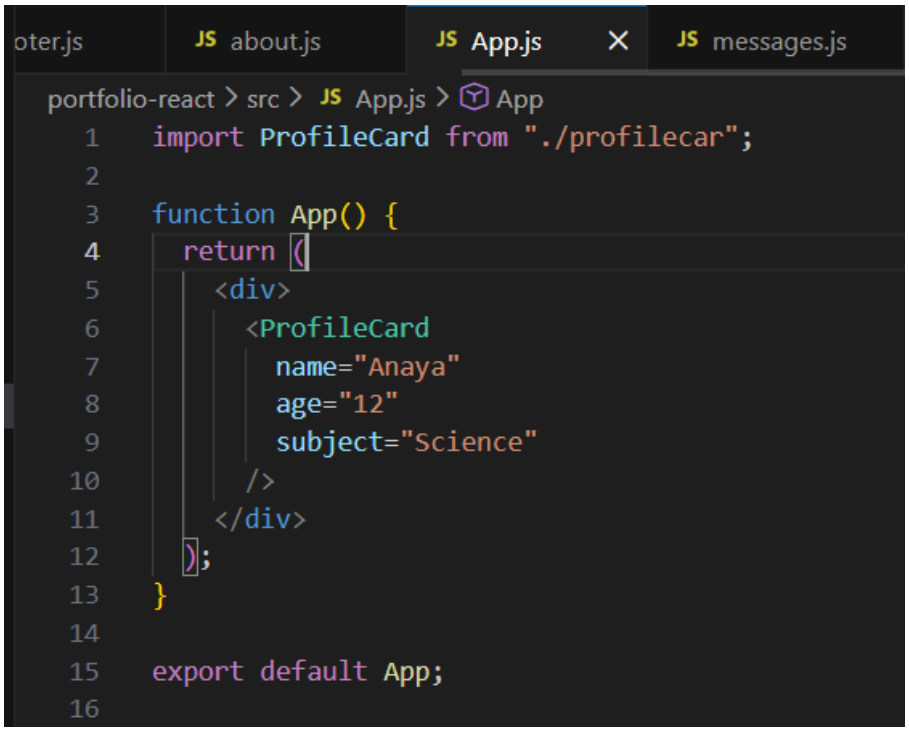
Name: Anaya
Age: 12
Favorite Subject: Science

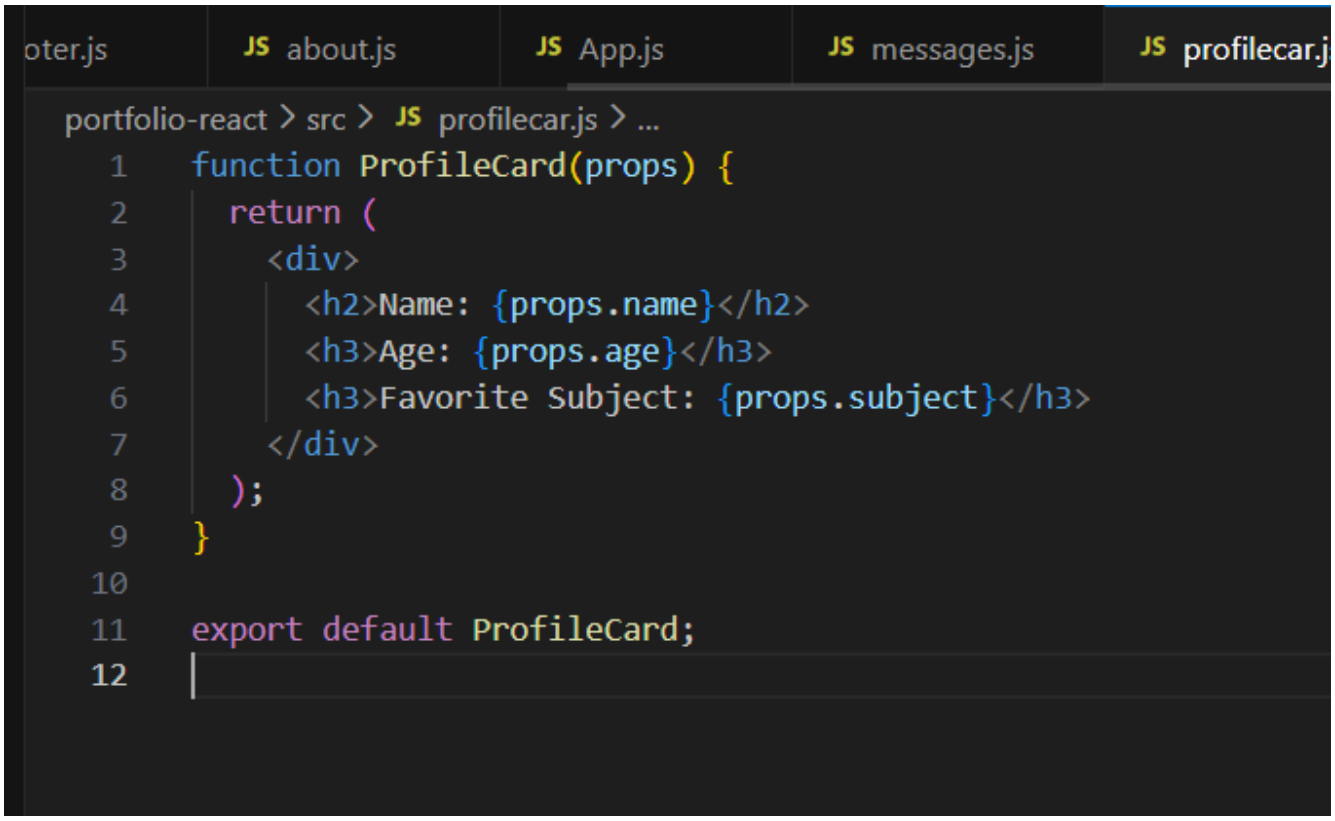
✅ Concept learned: Props + Reusability

Name: Anaya

Age: 12

Favorite Subject: Science




oter.js
about.js
App.js
messages.js
profilecar.j
portfolio-react > src > profilecar.js > ...
function ProfileCard(props) {
return (

<h2>Name: {props.name}</h2>
<h3>Age: {props.age}</h3>
<h3>Favorite Subject: {props.subject}</h3>

);
}
export default ProfileCard;