



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!

JS header.js JS about.js JS App.js X JS profile.js JS hello.js

```
portfolio-react > src > JS 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 |
```

JS header.js JS about.js JS App.js JS profile.js JS hello.js X

```
portfolio-react > src > JS hello.js > ...
1 function Hello() {
2   return <h1>Hello, welcome to React!</h1>;
3 }
4
5 export default Hello;
6 |
```

❖❖ Exercise 2: Passing Props (Name Card)

Goal: Learn Props

Task:

1. Create a component Student.

2. Parent component sends:

- o name
- o 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

```
portfolio-react > src > JS student.js > ...
1  function Student(props) {
2    return (
3      <div>
4        <h2>Name: {props.name}</h2>
5        <h3>Class: {props.className}</h3>
6      </div>
7    );
8  }
9
10 export default Student;
11 |
```

```
JS header.js JS about.js JS App.js X JS student.js JS profile.js JS hello.js
portfolio-react > src > JS 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:
 - o Login
 - o Register
 - o Logout

✓ Concept learned: Reusable Components



JS header.js JS about.js JS App.js X JS button.js JS student.j

```
portfolio-react > src > JS 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
```

JS header.js JS about.js JS App.js JS button.js X JS s

```
portfolio-react > src > JS button.js > ...
1 function Button(props) {
2   return <button>{props.text}</button>;
3 }
4
5 export default Button;
6
```

❖❖ 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.

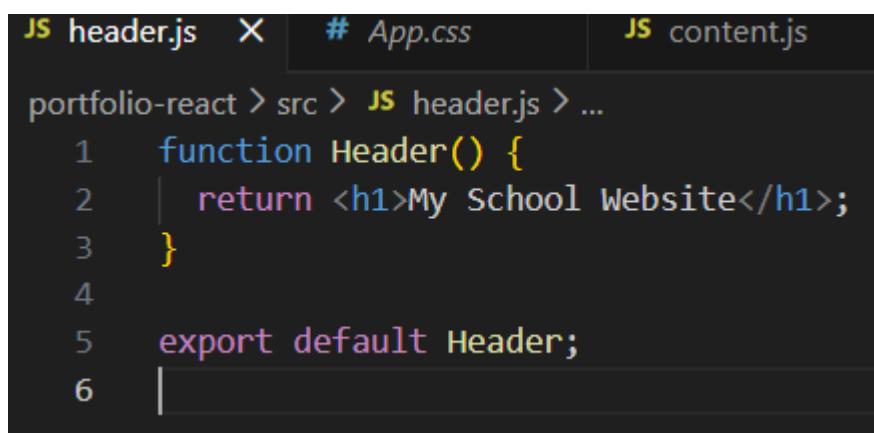
❖❖ Like building a LEGO house ❖❖

✓ Concept learned: Component structure

My School Website

Welcome Students!

© 2026 School



```
JS header.js X # App.css JS content.js
portfolio-react > src > JS header.js > ...
1  function Header() {
2    return <h1>My School Website</h1>;
3  }
4
5  export default Header;
6
```

JS header.js # App.css X JS content.js X

portfolio-react > src > JS content.js > ...

```
1 function Content() {
2   return <p>Welcome Students!</p>;
3 }
4
5 export default Content;
6
```

JS header.js # App.css JS content.js JS footer.js X

portfolio-react > src > JS footer.js > ...

```
1 function Footer() {
2   return <h4>© 2026 School</h4>;
3 }
4
5 export default Footer;
6
```

JS header.js # App.css JS content.js JS footer.js X

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

```
1 import Header from "./header";
2 import Content from "./content";
3 import Footer from "./footer";
4
5 function App() {
6   return (
7     <div>
8       <Header />
9       <Content />
10      <Footer />
11    </div>
12  );
13}
14
15 export default App;
16
```

❖❖ Exercise 5: SPA Thinking (No Reload)

Goal: Understand SPA idea

Task:

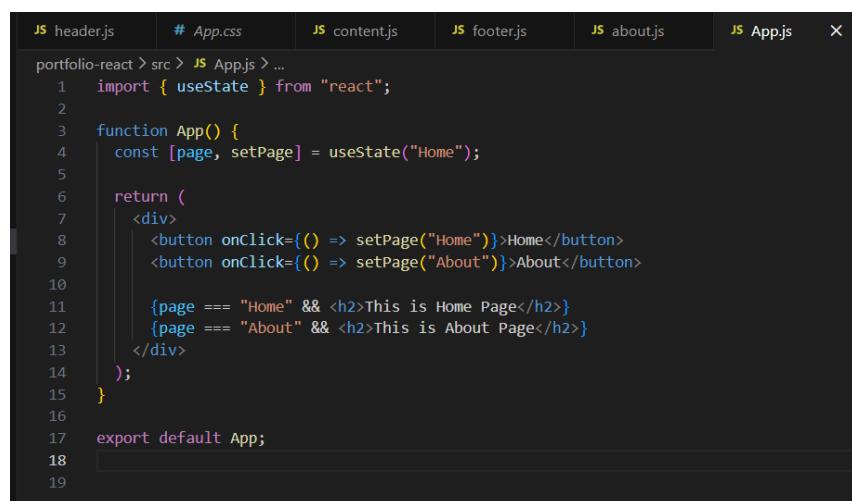
1. Create two buttons:
 - o Home
 - o About
2. Clicking each button should change text on the page:
 - o Home → “This is Home Page”
 - o About → “This is About Page”

❖❖ No page reload

✓ Concept learned: SPA behavior (basic)

Home About

This is Home Page



```
JS header.js # App.css JS content.js JS footer.js JS about.js JS App.js X
portfolio-react > src > JS App.js > ...
1 import { useState } from "react";
2
3 function App() {
4   const [page, setPage] = useState("Home");
5
6   return (
7     <div>
8       <button onClick={() => setPage("Home")}>Home</button>
9       <button onClick={() => setPage("About")}>About</button>
10
11      {page === "Home" && <h2>This is Home Page</h2>}
12      {page === "About" && <h2>This is About Page</h2>}
13    </div>
14  );
15}
16
17 export default App;
18
19
```

Exercise 6: Parent → Child Message

Goal: One-way data flow

Task:

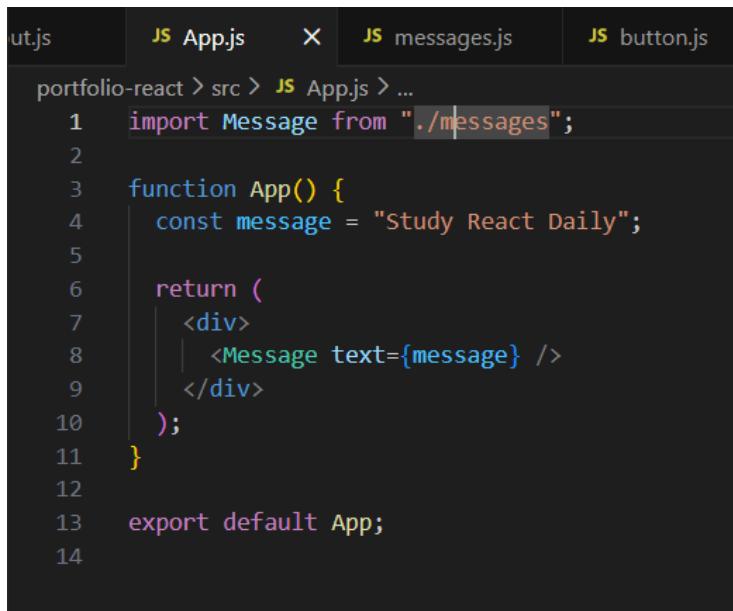
1. Parent component stores a message:
2. "Study React Daily"
3. Pass it to Message child component.
4. Child displays it.

❖❖ Child should NOT change the

message ✓ Concept learned: One-way

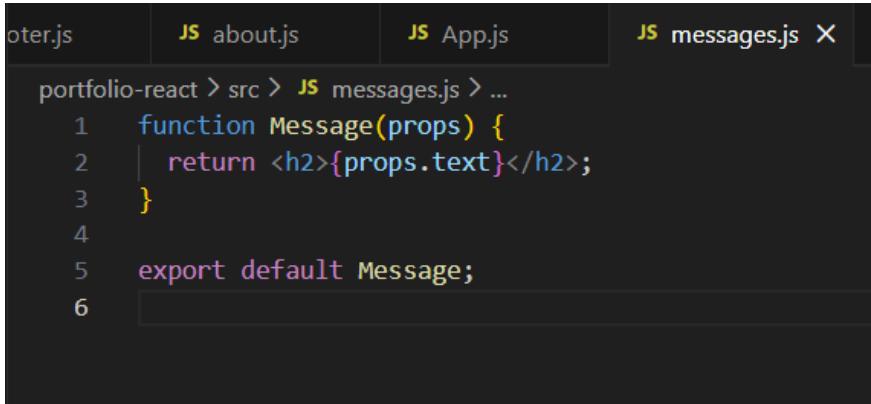
data flow

Study React Daily



A screenshot of a code editor showing the `App.js` file. The file contains the following code:

```
ut.js      JS App.js      X  JS messages.js      JS button.js
portfolio-react > src > JS 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
```



```
portfolio-react > src > JS messages.js > ...
1  function Message(props) {
2    |  return <h2>{props.text}</h2>;
3  }
4
5  export default Message;
6
```

❖❖ 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

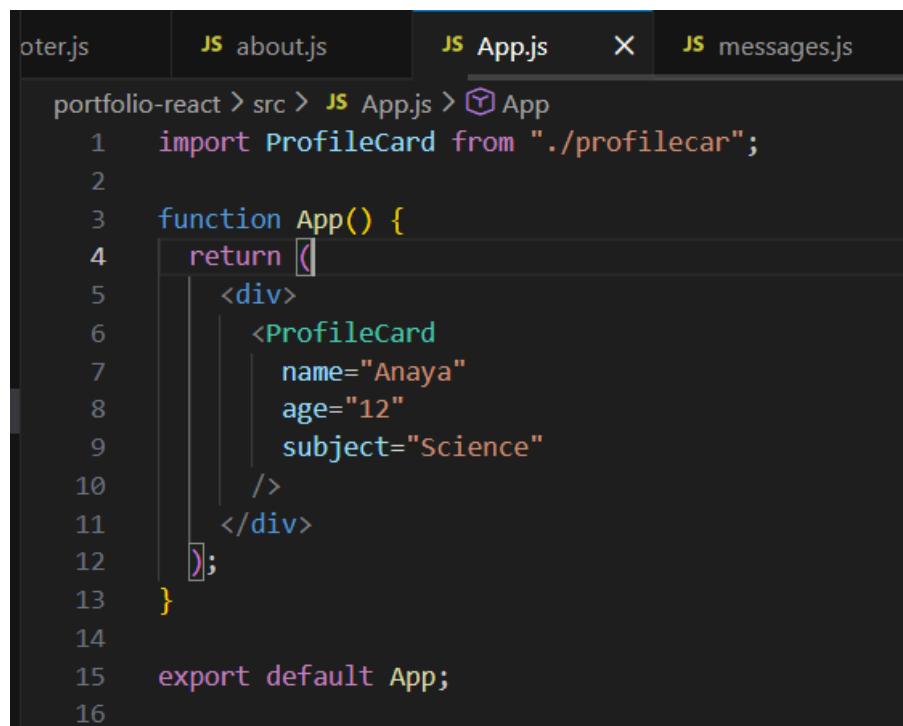
Favorite Subject: Science

 Concept learned: Props + Reusability

Name: Anaya

Age: 12

Favorite Subject: Science



The screenshot shows a code editor with the tab bar at the top containing four tabs: 'oter.js', 'JS about.js', 'JS App.js' (which is the active tab), and 'JS messages.js'. The code in the editor is as follows:

```
portfolio-react > src > JS App.js > App
1 import ProfileCard from "./profilecar";
2
3 function App() {
4     return (
5         <div>
6             <ProfileCard
7                 name="Anaya"
8                 age="12"
9                 subject="Science"
10            />
11        </div>
12    );
13 }
14
15 export default App;
16
```

oter.js

JS about.js

JS App.js

JS messages.js

JS profilecar.j

```
portfolio-react > src > JS profilecar.js > ...
1  function ProfileCard(props) {
2    return (
3      <div>
4        <h2>Name: {props.name}</h2>
5        <h3>Age: {props.age}</h3>
6        <h3>Favorite Subject: {props.subject}</h3>
7      </div>
8    );
9  }
10
11 export default ProfileCard;
12
```