CET-223

Web Technologies

Experiment # 17

Experiment Title

Introduction to React JS: Building Your First Component

Assessment of CLO(s): 03

Performed on ____________________

Student Name:			
Roll No.		**Group**	
Semester		**Session**	

Total (Max)	**Performance (03)**	**Viva (03)**	**File (04)**	**Total (10)**
Marks Obtained				
Remarks (if any)				

Experiment evaluated by

Instructor's Name	**Engr. Bilal Iqbal**		
Date		**Signature**	

Objective:

This lab introduces you to the fundamentals of React, a popular JavaScript library for building user interfaces. By the end of this lab, you will understand basic React concepts and create your first React component.

What is React?

- React is a JavaScript library for building fast and interactive user interfaces.
- Developed by Facebook, it helps create reusable UI components.

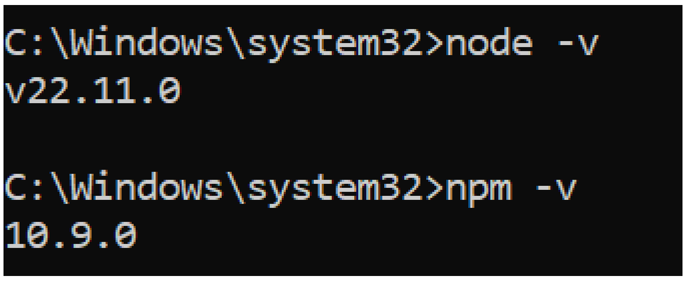
Why use React?

- Efficient and fast with Virtual DOM.
- Component-based architecture allows reusable components.
- Ideal for single-page applications (SPAs).

Setting Up a React Development Environment

1. **Installing Node.js and npm:**

- Download and install Node.js from https://nodejs.org/.
- Open a terminal and check installation by running:



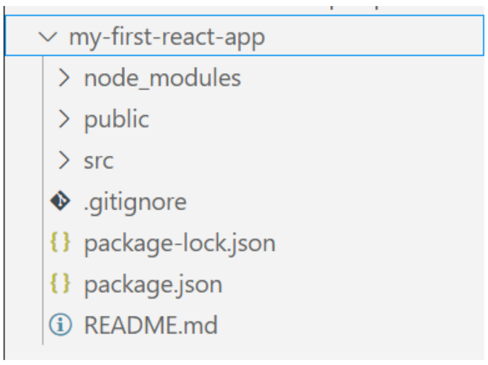
```
C:\Windows\system32>node -v
v22.11.0

C:\Windows\system32>npm -v
10.9.0
```

2. **Creating a New React Application:**

- Open a terminal and use create-react-app to set up a new React project:

```
npx create-react-app my-first-react-app
```



```
my-first-react-app
  node_modules
  public
  src
  .gitignore
  package-lock.json
  package.json
  README.md
```

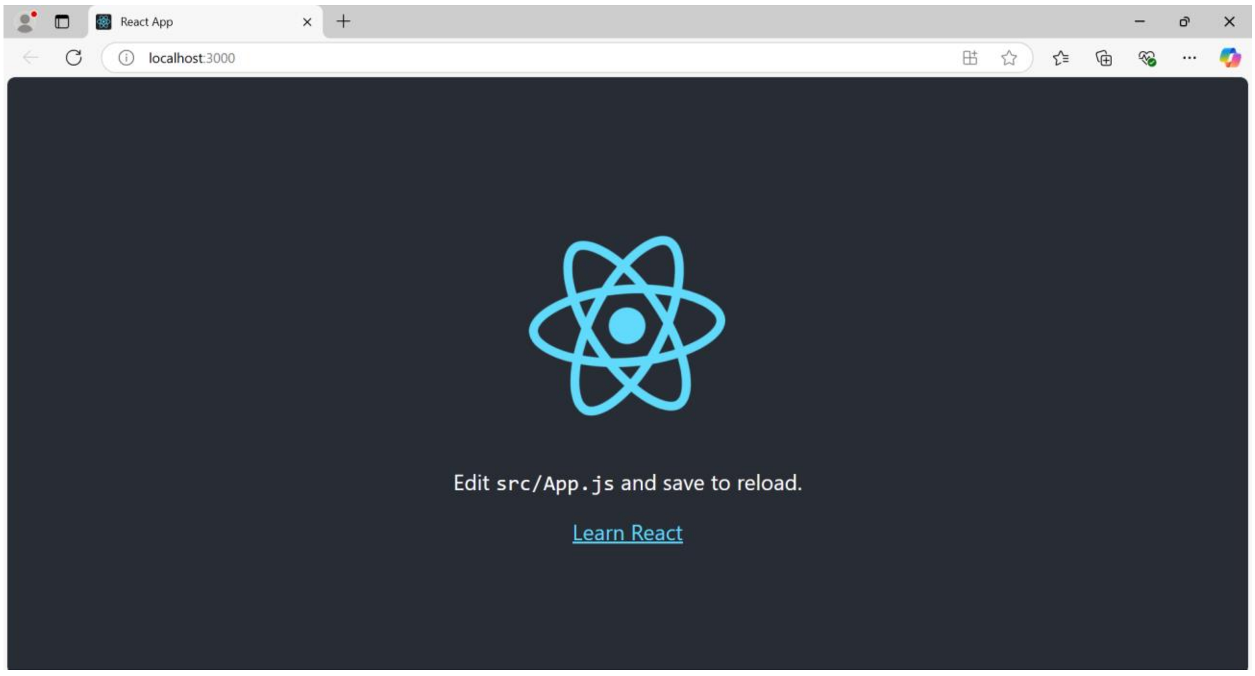
- Navigate into the project folder:

```
cd my-first-react-app
```

- Start the development server:

```
npm start
```

- Open http://localhost:3000 in your browser to view the default React app.



Edit src/App.js and save to reload.

Learn React

3. Understanding React Components

What is a React Component?

- A component is a reusable piece of UI. React applications are built by combining multiple components.

Types of Components in React:

- **Functional Components:** Functions that return JSX to render UI elements.
- **Class Components:** These are JavaScript classes that extend React.Component (we will focus on functional components as they are widely used).

Understanding JSX:

- JSX (JavaScript XML) allows writing HTML-like syntax within JavaScript. JSX is then compiled into JavaScript.

Building Your First React Component

- **Creating a Separate Component File:**

In the src folder, create a new file named Welcome.js for the Welcome component.

```
// src/Welcome.js
import React from 'react';

function Welcome() {
  return (
    <div className="welcome">
    <h1>Welcome to My First React App!</h1>
    <p>This is a simple React Component.</p>
    </div>
  );
}
export default Welcome;
```

- **Importing and Using the Component in App.js:**

```
import './App.css';
import Welcome from './Welcome';

function App() {
  return (
    <>
    <Welcome/>
    </>
  );
}
export default App;
```

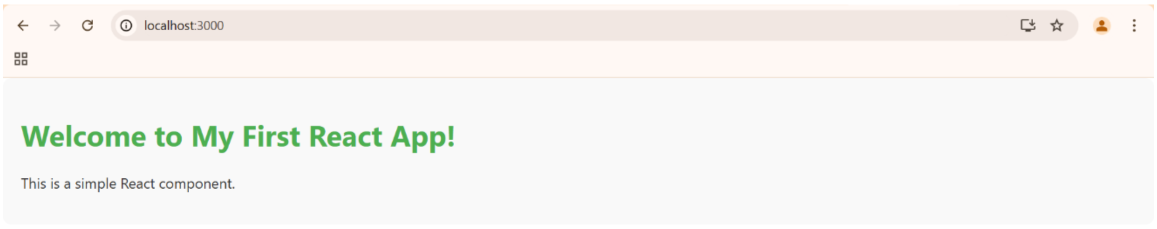
We created a Welcome component that displays a heading and a paragraph.
Inside the App component, we rendered the Welcome component using <Welcome />

- **Styling the Component:**

Open src/App.css to add some basic styles for the Welcome component.

```
/* src/App.css */
.welcome {
  background-color: #f9f9f9;
  padding: 20px;
  border-radius: 10px;
  color: #333;
}

h1 {
  color: #4caf50;
}
```



Welcome to My First React App!

This is a simple React component.

- **Passing Props to a Component:**

Props allow passing data to components in React. Update the Welcome component to accept a name prop.

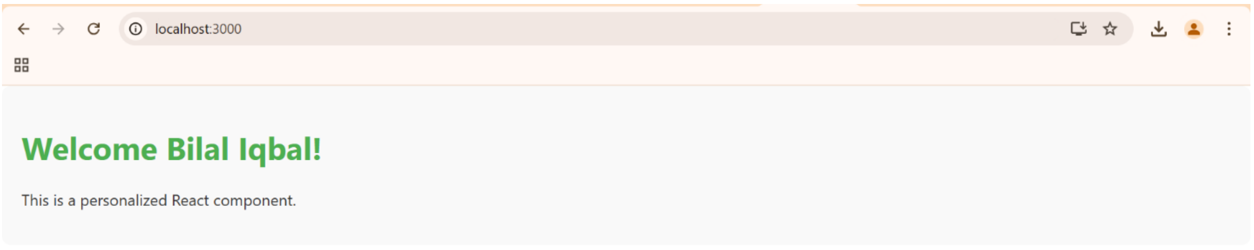
```
// src/Welcome.js
import React from 'react';

function Welcome(props) {
  return (
    <div className="welcome">
    <h1>Welcome {props.name}</h1>
    <p>This is a simple React component.</p>

    </div>
  );
}
export default Welcome;
```

```
// src/Welcome.js
import React from 'react';

function Welcome(props) {
  return (
    <div className="welcome">
    <h1>Welcome {props.name}!</h1>
    <p>This is a personalized React component.</p>
    </div>
  );
}
export default Welcome;
```



Welcome Bilal Iqbal!

This is a personalized React component.

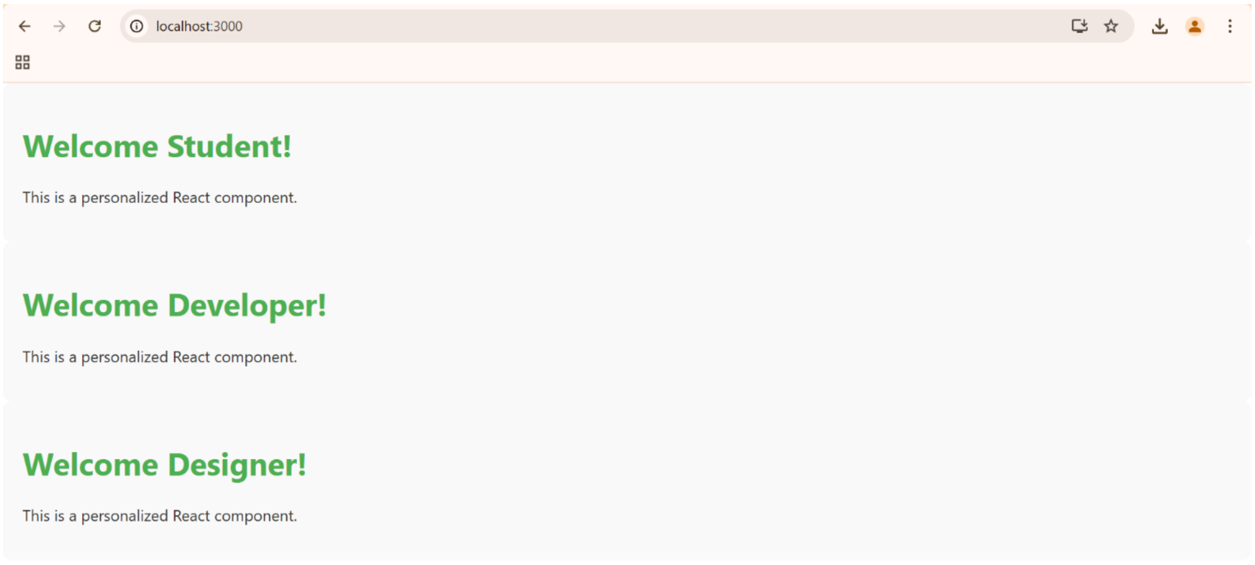
Here, the Welcome component now accepts a name prop, which allows dynamic text rendering.

- **Understanding Component Reusability:**

You can reuse the Welcome component multiple times with different props

```
//src/App.js
import './App.css';
import Welcome from './Welcome';

function App() {
  return (
    <>
    <Welcome name = "Student"/>
    <Welcome name = "Developer"/>
    <Welcome name = "Designer"/>
    </>
  );
}
export default App;
```



localhost:3000

Welcome Student!

This is a personalized React component.

Welcome Developer!

This is a personalized React component.

Welcome Designer!

This is a personalized React component.

This approach highlights React's power with reusable components, each instance rendering with its unique prop value.

Lab Task:

Create a Custom Message Component:

- Create a new file named Message.js.
- Inside Message.js, define a Message component that accepts a text prop and displays it within a <p> tag.
- Import Message into App.js and render it with a custom message.

Experiment with Props:

- Modify the App component to render Message multiple times with different values for the text prop.