NORAIZ RANA

Task #8

Topic

Introduction to React, JSX and Components

Date:

12 July 2025

SUBMITTED TO:

APPVERSE TECHNOLOGIES

▧ Learning Report: Introduction to React, JSX, and Components

1. Introduction to React

React is a **JavaScript library** developed by **Facebook** for building fast, interactive user interfaces, especially for single-page applications (SPAs). It helps developers create large web applications that can update and render efficiently in response to data changes.

Key Features:

- **Component-Based Architecture**: Everything in React is a component.
- **Virtual DOM**: Optimizes rendering by updating only what has changed.
- **Declarative Code**: We describe what UI should look like and React takes care of the rest.
- **Reusable Components**: Promotes code reusability and maintainability.
- **Unidirectional Data Flow**: Data flows in one direction, making it easier to debug and manage.

Why Use React?

- Easy to learn for developers familiar with JavaScript
- Strong community and ecosystem
- Widely used in the industry (used by Facebook, Instagram, Netflix, etc.)

---

2. JSX (JavaScript XML)

JSX stands for **JavaScript XML**. It is a **syntax extension** for JavaScript, used with React to describe what the UI should look like.

JSX allows us to write HTML elements inside JavaScript code, making it easier to visualize UI components.

Example:

```
const element = <h1>Hello, React!</h1>;
```

Why JSX?

- Makes code readable and expressive
- Helps write HTML-like structures in JavaScript
- Can embed JavaScript expressions using `{}`

JSX Rules:

- Must return **one parent element**
- Use `className` instead of `class`
- Close all tags (even self-closing like `<img />`)
- Use `{}` to insert dynamic expressions

---

3. React Components

In React, **components** are the building blocks of UI. Components let us **split the UI into independent, reusable pieces**.

Types of Components:

1. **Functional Components** (Modern & Preferred)
2. **Class Components** (Older, still in use)

Example of a Functional Component:

```
function Welcome() {
  return <h1>Welcome to React!</h1>;
}
```

Example of a Class Component:

```
class Welcome extends React.Component {
  render() {
    return <h1>Welcome to React!</h1>;
  }
}
```

Component Structure:

- **Props** (short for properties): Used to pass data into components
- **State**: Represents mutable data within a component (usually with hooks like `useState`)
- Components can **render other components**, allowing for nesting and composition.