

INTERNSHIP REPORT

WEEK 6 DAY 1

Submitted to:	Ali Hyder	Submission Date:	28 th July, 2025
Internship Domain:	Front Development	Internship Name:	ProSensia
Student Name:	Yasal Qamar	Roll No.	S25031

React.js Basics

Topics: Introduction to React, create-react-app setup

Objective

To understand the use of **JSX** in React for rendering content, how to create **reusable functional components**, and how to **pass data (props)** between them to build dynamic UIs.

Topics Covered:

- JSX Recap and Rules
- Creating Functional Components
- Reusability with Props
- Rendering multiple components using props

What is React?

React is a **JavaScript library** used to build **user interfaces (UI)**, especially for **web applications**. It helps you create **dynamic, reusable components** that can change based on data without needing to reload the page.

Why Use React?

- **Component-Based:** Break your UI into small, reusable pieces (like buttons, forms, headers).
- **Fast and Efficient:** Uses a virtual DOM to update only the changed parts of the UI.
- **One-Way Data Flow:** Data flows in one direction, making the app easier to debug and manage.
- **Reusable JSX Code:** Write HTML-like code inside JavaScript using JSX.

1. JSX:

- JSX lets you write HTML inside JavaScript functions.

```
function App() {  
  return <h1>Hello, world!</h1>;  
}
```

2. Functional Components

- A component is just a function that returns JSX.
- They can be **reused** and help split UI into smaller, manageable pieces.

```
function Greet(props) {  
  
  return <h3>Hello, {props.name}!</h3>;  
  
}
```

3. Props Example:

- Props (short for "properties") allow data to be passed from one component to another.

```
function Card({ title, description }) {  
  
  return (  
  
    <div>  
  
      <h2>{title}</h2>  
  
      <p>{description}</p>  
  
    </div>  
  
  );  
  
}
```

4. Destructuring Props Example:

```
function Card({ title, description }) {

  return (

    <div className="card">

      <h2>{title}</h2>

      <p>{description}</p>

    </div>

  );

}

function App() {

  return (

    <div>

      <Card title="React" description="JS library for building UI" />

      <Card title="Props" description="Pass data to components" />

    </div>

  );

}

export default App;
```

CODING:

.app.js

```
function KingCard({ name, kingdom, power }) {
  return (
    <div className="king-card">
      <h2>👑 {name}</h2>
      <p><strong>Kingdom:</strong> {kingdom}</p>
      <p><strong>Power:</strong> {power}</p>
    </div>
  );
}
```

```

    );
  }

function App() {
  return (
    <div className="container">
      <h1 className="title">👑 Legendary Kings Collection 👑</h1>
      <p className="subtitle">Explore the greatest rulers of ancient
realms:</p>

      <div className="card-grid">
        <KingCard name="King Arthur" kingdom="Camelot" power="Excalibur &
Honor" />
        <KingCard name="King Leonidas" kingdom="Sparta" power="Courage &
Strategy" />
        <KingCard name="King Ragnar" kingdom="Denmark" power="Warrior's
Spirit" />
        <KingCard name="King Xerxes" kingdom="Persia" power="Immortal Army" />
      </div>
    </div>
  );
}

export default App;

```

.index.css

```

body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto',
'Oxygen',
  'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
  sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
  monospace;
}

body {

```

```
margin: 0;
padding: 0;
background: linear-gradient(135deg, #1e1e2f, #2a2a40);
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
color: #f1f1f1;
}

.container {
  max-width: 960px;
  margin: auto;
  padding: 40px 20px;
}

.title {
  text-align: center;
  font-size: 2.8rem;
  color: gold;
  margin-bottom: 10px;
}

.subtitle {
  text-align: center;
  font-size: 1.2rem;
  color: #ddd;
  margin-bottom: 40px;
}

.card-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 25px;
}

.king-card {
  background: #2e2e4d;
  border: 1px solid #444;
  border-radius: 16px;
  padding: 20px;
  transition: transform 0.2s ease, box-shadow 0.2s ease;
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.4);
}

.king-card:hover {
  transform: translateY(-5px);
  box-shadow: 0 8px 20px rgba(255, 215, 0, 0.4);
}
```

```
}  
  
.king-card h2 {  
  margin-top: 0;  
  color: #ffd700;  
}  
  
.king-card p {  
  margin: 8px 0;  
  color: #ddd;  
}
```



CONCUSLION:

Today, I learned the foundational concepts of building user interfaces in React using JSX, functional components, and props. I understood how JSX allows mixing HTML with JavaScript, how functional components make code modular and reusable, and how props help pass data dynamically to components. By building and styling a reusable "Kings Card" layout, I practiced applying these concepts in a real-world scenario. This knowledge sets a strong base for learning advanced React topics like state management and routing in the upcoming days.
