

## Internship Report – Week [Insert Week], Day [Insert Day]

**Topic:** React Context API – Basics

**Name:** Hamza Rafique

---

### Work Done Today:

Today I learned about the **React Context API**, which allows us to share data between components **without passing props manually at every level** (prop drilling). It is useful for global data like themes, user info, or app settings.

---

### Key Concepts Learned:

#### 1. Purpose of Context API:

- Solves the problem of **prop drilling**.
- Allows global state sharing across components.

#### 2. Creating Context:

javascript

CopyEdit

```
import { createContext } from "react";
```

```
export const MyContext = createContext();
```

#### 3. Providing Context:

- Wrap the component tree with a **Provider**.

javascript

CopyEdit

```
import React, { createContext, useState } from "react";
```

```
export const MyContext = createContext();
```

```
function App() {
```

```
const [user, setUser] = useState("Hamza");

return (
  <MyContext.Provider value={{ user, setUser }}>
    <ChildComponent />
  </MyContext.Provider>
);
}

export default App;
```

#### 4. Consuming Context:

- Using **useContext** Hook (recommended).

javascript

CopyEdit

```
import React, { useContext } from "react";
import { MyContext } from "../App";
```

```
function ChildComponent() {
  const { user } = useContext(MyContext);
  return <h1>Hello, {user}!</h1>;
}
```

#### 5. Flow:

- **createContext → Provider → useContext**

---

#### Key Learnings:

- Context API is a built-in React feature (no extra library needed).
- Best for small to medium projects or limited global state.

- For large, complex state management, tools like Redux may be better.
- 

### **Challenges Faced:**

- Initially, I confused Context with Redux, but Context is simpler and only for data passing, not for complex state logic.
- 

### **Next Steps:**

- Practice Context API with theme switching and authentication examples.
- Combine Context API with useReducer for better state control.