

Internship Report

Week: 7

Day: 3

Topic: React Project Folder Structure

Intern: Hamza Rafique

Organization: ProSensia

What I Learned Today

Today, I explored and understood how to structure a React project for better **scalability**, **readability**, and **maintenance**. A well-organized folder structure is essential, especially when working on larger applications or in teams.

Recommended React Folder Structure

perl

CopyEdit

my-app/

|

|— public/

| └─ index.html

|

|— src/

| └─ assets/ # Images, fonts, videos

| └─ components/ # Reusable UI components (Button, Card, etc.)

| └─ pages/ # Route-specific page components (Home, About, etc.)

| └─ layouts/ # Common layouts (Navbar, Footer, etc.)

| └─ hooks/ # Custom React hooks

| └─ context/ # React Context API files for global state

```
| └─ services/    # API calls and logic (e.g., axios)
| └─ utils/       # Utility functions or helpers
| └─ App.jsx      # Root component
| └─ main.jsx     # Entry point
|
└─ .gitignore
└─ package.json
└─ README.md
```

✅ Key Takeaways

- Separating concerns improves **collaboration** and **debugging**.
 - Grouping reusable components avoids duplication and confusion.
 - Folder structure can be **customized** depending on project size and needs.
 - Following **conventions** (like placing all components in a components/ folder) keeps the codebase **clean and consistent**.
-

🚀 Practical Experience

I also analyzed the folder structure of my current React projects and reorganized them using this clean structure. This boosted my understanding of **modularity** in React apps.