## Am planning to built my own portifolio web for my resume:

And am planning to use this tech stack recommendations:

 **React** (via create-react-app or Next.js)

 **Tailwind CSS** (for fast, responsive styling)

 **React Router** (if not using Next.js for routing)

 **Framer Motion** (for smooth animations)

 **Vercel** or **Netlify** (for free deployment)

what is ESLint?

###### ChatGPT said:

**ESLint** is a **static code analysis tool** for JavaScript (and TypeScript) that **automatically checks your code for errors, bad practices, and style issues**.

## 🧠 **What Does ESLint Do?**

When you write JavaScript/React code, ESLint:

* ✅ **Finds syntax errors** (e.g., missing semicolons, undefined variables)
* 🔍 **Catches bugs early** (e.g., unused variables, shadowed variables)
* 🧼 **Enforces consistent coding style** (e.g., indent spacing, quote styles)
* 🔧 **Suggests or auto-fixes code** based on rules

Great question! The term **App Router** can mean slightly different things depending on the context, but in the **React/Next.js** ecosystem, it often refers to the **routing system that controls what pages/components are shown based on the URL.**

## 🗂️ App Router in Next.js (since v13+)

Starting in **Next.js 13**, the **App Router** is a **new routing system** that replaces the old **Pages Router** (pages/ directory).

### 📌 App Router = New way of structuring routes using the app/ directory.

## 🆚 Pages Router vs. App Router

| **Feature** | **Pages Router (pages/)** | **App Router (app/)** |
| --- | --- | --- |
| Folder name | pages/ | app/ |
| Routing | File-based (old way) | File-based (modern + nested) |
| Data fetching | getStaticProps, etc. | fetch() + Server Components |
| File layout | Flat | Supports nested layouts |
| Supported in v13+ | ✅ Still works | ✅ Recommended (default now) |

## 🧭 How App Router Works

In the app/ directory, you define routes like this:

app/

├── layout.tsx ← shared layout

├── page.tsx ← homepage (`/`)

├── about/

│ └── page.tsx ← `/about`

└── projects/

└── page.tsx ← `/projects`

### Special Files:

* page.tsx → The actual page (like route content).
* layout.tsx → A layout wrapper for one or more pages (nav, footer, etc.).
* loading.tsx → A loading state.
* error.tsx → Error boundary.
* not-found.tsx → Custom 404 page.

## ✅ Why App Router is Better (in most cases)

* Supports **nested layouts** (e.g., sidebar per section).
* Enables **streaming UI** (via React Server Components).
* Cleaner way to manage shared components.
* Built for **modern React patterns** like async/await and server-side rendering.

## 🧠 Summary

The **App Router** in Next.js is the **modern, flexible way** to handle routing, layouts, and server/client component architecture — replacing the older pages/ directory system.