Complete Next.js Portfolio Structure Explanation

📁 Root Level Files & Folders

Configuration Files

- `package.json` - Project metadata, dependencies, and scripts
- `package-lock.json` - Exact dependency versions (auto-generated)
- `next.config.ts` - Next.js configuration settings
- `tsconfig.json` - TypeScript configuration
- `tailwind.config.js` - Tailwind CSS configuration
- `postcss.config.js` - PostCSS configuration (for Tailwind)
- `next-env.d.ts` - TypeScript declarations for Next.js (auto-generated)
- `eslint.config.mjs` - Code linting rules
- `README.md` - Project documentation
- `.gitignore` - Files to ignore in version control

Build & Dependencies

- `.next/` - Build output folder (auto-generated, don't touch!)
- `node_modules/` - Installed packages (auto-generated)

📁 public/ - Static Assets

```
public/
├── favicon.ico          # Website icon in browser tab
└── [your images]        # Profile pics, project screenshots, resume PDF
```

Purpose: Files here are served directly at the root URL

- `public/resume.pdf` → accessible at `/resume.pdf`
- `public/profile.jpg` → accessible at `/profile.jpg`

📁 src/app/ - The Heart of Your App (App Router)

Core App Files

- **`layout.tsx`** - Root layout that wraps ALL pages
  - Contains `<html>`, `<body>` tags
  - Navigation bar, footer that appear everywhere
  - Global providers, fonts, metadata
- **`page.tsx`** - Your homepage (when someone visits `/`)
  - The main landing page
  - Usually contains hero section, about me, featured projects
- **`globals.css`** - Global CSS styles
  - Tailwind imports
  - Custom CSS that applies everywhere
  - Reset styles, typography

Page Routes (Folders with page.tsx)

Each folder creates a new URL route:

- **`contact/page.tsx`** → `/contact` URL
  - Contact form, email, social links
  - Your ways to get in touch
- **`projects/page.tsx`** → `/projects` URL
  - Showcase all your projects
  - Grid of project cards, filters
- **`playground/page.tsx`** → `/playground` URL
  - Future AI demos, experiments
  - Interactive components, cool stuff

Components Folder

- **`components/`** - Reusable UI pieces
  - **`Navbar.tsx`** - Navigation menu (Home, Projects, Contact, etc.)
  - **`Footer.tsx`** - Footer with links, copyright
  - **`HeroSection.tsx`** - Main banner on homepage
  - **`ProjectCard.tsx`** - Individual project display card

🗂️ How It All Works Together

1. App Router Magic

```
URL: /projects
↓
Next.js looks for: src/app/projects/page.tsx
↓
Wraps it with: src/app/layout.tsx
↓
Shows: Complete page with navbar + projects content + footer
```

2. Component Reusability

```
layout.tsx imports Navbar.tsx → Shows on every page
page.tsx imports HeroSection.tsx → Only shows on homepage
projects/page.tsx imports ProjectCard.tsx → Shows project grid
```

3. File-Based Routing

- No need to configure routes manually
- Folder name = URL path
- `page.tsx` = the actual page content
- `layout.tsx` = shared wrapper around pages

🎯 Quick Mental Model

Think of it like a house:

- **`layout.tsx`** = The house structure (walls, roof) - same for all rooms
- **`page.tsx`** files = Different rooms (bedroom, kitchen, etc.)
- **`components/`** = Furniture you can move between rooms
- **`public/`** = Your front yard (visible to everyone)
- **`globals.css`** = House-wide utilities (electricity, plumbing)
- **Config files** = Building permits and blueprints

🚀 Development Flow

1. **Start dev server:** `npm run dev`
2. **Edit pages:** Modify `page.tsx` files
3. **Create components:** Build in `components/` folder
4. **Add assets:** Drop images in `public/`
5. **Style:** Use Tailwind classes or add to `globals.css`
6. **Test:** Navigate between `/`, `/projects`, `/contact`

📝 Key Next.js App Router Concepts

- **Server Components:** Pages render on server by default (faster)
- **File Conventions:** `page.tsx`, `layout.tsx`, `loading.tsx`, `error.tsx`
- **Nested Layouts:** Each folder can have its own `layout.tsx`
- **Automatic Code Splitting:** Each page loads only what it needs