**app/ – App Router**

* Kjo është dosja kryesore kur përdor **App Router**, e prezantuar në Next.js 13+.
* Çdo dosje dhe skedar brenda app/ përfaqëson një segment të rrugës.
  + Shembull:

**pages/ – Pages Router**

* Është mënyra e vjetër për të menaxhuar rrugët, e njohur si **Pages Router**.
* Çdo skedar .js/.ts/.tsx brenda pages/ përfaqëson një rrugë:
  + Shembull:
  + pages/
  + ├── index.tsx → për faqen kryesore `/`
  + └── about.tsx → për faqen `/about`
* Nuk ka layout.tsx si App Router; layout-i duhet të bëhet përmes komponentëve të zakonshëm ose /\_app.tsx.

### public/ ****– Static assets to be served****

* Kjo dosje përdoret për **skedarë statikë** që serveri i jep direkt klientit.
* Shembuj:
  + Imazhe: /public/logo.png → mund të përdoret si /logo.png
  + Fonta, skedarë JSON, videos, ose çdo gjë që nuk ka nevojë për transformim nga Next.js.
* Nuk nevojitet import; mund të referohet direkt nga URL.

### src/ ****– Optional application source folder****

* Kjo është **opsionale**, por shumë devs e përdorin për të ndarë kodin e aplikacionit nga skedarët e konfigurimit.
* Brenda src/ mund të vendosen:
  + app/ ose pages/
  + components/
  + lib/ (funksione helper)
  + hooks/, context/, etj.
* Nuk ndryshon funksionalitetin; thjesht e mban projektin më të organizuar.

Sigurisht! Ja një **përshkrim i shkurtër për secilin top-level file** në një projekt Next.js:

**1️⃣ next.config.js**

* Skedari kryesor i konfigurimit për Next.js.
* Mund të vendosësh rregullime për build, routing, environment variables, redirects, rewrites, etj.

**2️⃣ package.json**

* Përmban **varësitë e projektit** (dependencies) dhe skriptet për run, build, dev, test, etj.

**3️⃣ instrumentation.ts**

* Përdoret për **OpenTelemetry dhe instrumentation**, pra monitorim i performancës dhe gjurmim i aplikacionit.

**4️⃣ middleware.ts**

* Lejon **ndërhyrje në request/response** në server (p.sh. autentikim, logging, redirects).

**5️⃣ .env / .env.\***

* .env → variabla globale të mjedisit.
* .env.local → variabla për development lokal (jo e publikuar në git).
* .env.production → variabla për prodhim.
* .env.development → variabla për development.

**6️⃣ .eslintrc.json**

* Konfigurimi për **ESLint**, për të kontrolluar stilin dhe cilësinë e kodit.

**7️⃣ .gitignore**

* Skedarët dhe folderët që **Git duhet t’i injorojë**, p.sh. node\_modules/.

**8️⃣ next-env.d.ts**

* Deklarata **TypeScript** për Next.js, për të njohur tipat e Next.js automatikisht.

**9️⃣ tsconfig.json**

* Konfigurimi kryesor për **TypeScript**, si target, paths, strict mode, etj.

**🔟 jsconfig.json**

* Konfigurimi për **JavaScript**, p.sh. për alias dhe paths.

### layout.js/.jsx/.tsx

* Përfaqëson **layout-in e përbashkët** për një rrugë ose grup rrugësh.
* Mund të përfshijë elementë që përsëriten si **header, nav, footer**.

### ****2️⃣**** page.js/.jsx/.tsx

* Skedari që përfaqëson **faqen e vetme për një rrugë**.
* Çdo page.tsx bëhet një route automatikisht.

### ****3️⃣**** loading.js/.jsx/.tsx

* Për **UI të ngarkimit (skeletons)** ndërsa faqja ose komponentët po load-ohen.
* Shfaqet automatikisht kur përdoren **dynamic routes ose fetch**.

### ****4️⃣**** not-found.js/.jsx/.tsx

* Përfaqëson faqen e **“Not Found”** (404) për një rrugë të caktuar.
* Shfaqet kur një resource nuk gjendet.

**5️⃣ error.js/.jsx/.tsx**

* Për **error boundaries** për një rrugë të caktuar.
* Kap gabime të komponentëve brenda një layout ose page.

**6️⃣ global-error.js/.jsx/.tsx**

* Kap të gjitha gabimet **në nivel global** për të gjithë aplikacionin.
* Shërben si back-up për gabimet që nuk kapen nga error.js.

**7️⃣ route.js/.ts**

* Për **API endpoints**.
* Mund të përdoret për **GET, POST, PUT, DELETE**, dhe trajton kërkesat HTTP.

**8️⃣ template.js/.jsx/.tsx**

* **Layout që re-renderohet** sa herë ndryshon rruga ose props.
* Ndryshe nga layout i cili ruan gjendjen, template e rifreskon çdo herë.

**9️⃣ default.js/.jsx/.tsx**

* **Fallback page për parallel routes**.
* Përdoret kur përdor **parallel routes** (p.sh. multiple layouts në të njëjtën rrugë) dhe nuk ka përmbajtje specifike.

NESTED ROUTES

### ****Koncepti kryesor****

* **Folders = URL segments**
  + Çdo dosje brenda app/ krijon një segment të rrugës.
  + **Nesting folders = nested URL segments**
    - Për shembull: app/blog/authors/page.tsx → /blog/authors
* **Layouts wrap children**
  + Çdo layout.tsx në një folder rrethon (wrap) të gjitha faqet ose folderët e tij brenda.
  + Kjo lejon që të krijosh **UI të përbashkët** për një grup rrugësh (p.sh. header, nav, footer).
* **Public routes**
  + Një rrugë bëhet publike kur ekziston një **page.tsx** ose **route.ts** në atë folder.

DYNAMIC ROUTES

**Koncepti kryesor**

* **Dynamic routes** përdoren kur një segment i URL ndryshon bazuar në parametra.
* Sintaksa me **kllapa katrore [ ]** përcakton parametrin:
  1. [segment] → një parametër i vetëm
  2. [...segment] → **catch-all** (kap shumë segmente)
  3. [[...segment]] → **optional catch-all** (mund të mos ketë segment fare)
* Parametrat mund të aksesohen në komponentin page përmes **params prop**.

**📌 Shembuj**

| **Path** | **URL pattern** | **Shpjegim** |
| --- | --- | --- |
| app/blog/[slug]/page.tsx | /blog/my-first-post | [slug] zëvendësohet me çdo post të ndryshëm |
| app/shop/[...slug]/page.tsx | /shop/clothing/shoes, /shop/clothing/shirts | [...slug] kap çdo rrugë të gjatë nën /shop |
| app/docs/[[...slug]]/page.tsx | /docs, /docs/layouts-and-pages, /docs/api-reference/use-router | [[...slug]] është optional, mund të mos ketë segment |

### ****Nested routes / groups****

* Bazohen në **strukturën e folderëve**.
* Çdo folder = segment i URL.
* Layout-i i folderit **rrethon të gjitha faqet ose folderët brenda**.
* Nuk ndryshon nga URL dinamike; URL-të janë statike bazuar në emrin e folderit.

### ****2️⃣ Dynamic routes****

* Përdoren kur **segmenti i URL duhet të jetë fleksibël** (parametrik).
* Përdorin **kllapa katrore [ ]** për të kapur parametrat.

### ****Route Groups****

* **Çfarë janë:** Një mënyrë për të organizuar kodin **pa ndryshuar URL-në**.
* Përdoret sintaksa me **kllapa normale (group)** në emrin e folderit.
* **URL nuk ndryshon**, por folderi ndihmon për strukturim dhe layout modular.

### ****1️⃣ Parallel Routes****

* **Çfarë janë:** Lejojnë të kesh **multiple layouts ose segmente të ndryshme që shfaqen paralelisht** në të njëjtën URL.
* Përdor **@slot** për të identifikuar “slot”-et ku një layout fëmijë do të renderohet.
* Shpesh përdoret për **UI slot-based**, ku një pjesë e ekranit ndryshon pa ndryshuar të gjithë layout-in.

### ****2️⃣ Intercepted Routes****

* **Çfarë janë:** Lejojnë të **renderosh një route tjetër brenda layout-it aktual pa ndryshuar URL-në**.
* Shpesh përdoret për **modals, dialogs, ose detaje të shfaqura mbi listë**.

**(..)(..)folder – Intercept two levels**

* Përdoret për të **kapur dhe renderuar dy nivele të thella të URL-së brenda një layout ose modal**.
* Kjo është një mënyrë për **overlay shumë të thellë** (deeply nested overlay) pa ndryshuar URL-në kryesore.
* Sintaksa (…) tregon intercept të një segmenti, dhe (…)(…) tregon intercept të dy segmenteve njëkohësisht.

### ****Metadata Files****

* Këto skedarë përdoren për të përcaktuar **ikonat e aplikacionit dhe favicon**.

### ****Open Graph dhe Twitter Images****

* Këto janë **imazhe metadata** që përdoren kur ndan një link të aplikacionit në rrjete sociale.
* Sigurojnë që postimet të kenë një **preview të bukur me imazh, titull dhe përshkrim**.

### ****SEO (Search Engine Optimization)****

* SEO ndihmon që faqja të **rangohet më mirë në motorët e kërkimit** si Google.
* Në Next.js, një nga elementët kryesorë për SEO është **sitemap.xml**.

### sitemap.xml

* Është një **skedar XML** që përmban **të gjitha URL-të e faqes**.
* Motorët e kërkimit (Google, Bing, etj.) e përdorin për të kuptuar strukturën e faqes dhe për të indexuar faqet më shpejt.

### ****Colocation****

* **Colocation** do të thotë të **vendosësh skedarë të lidhur me një route brenda të njëjtit folder**.
* Në app/, **folderët e nested përcaktojnë strukturën e URL-së**.
  + Çdo folder = një segment URL.

### ****Private Folders****

* Këto janë **folderë që nuk gjenerojnë URL** dhe nuk janë të aksesueshëm si rrugë.
* Për t’i krijuar, përdor **\_ në fillim të emrit të folderit**:

### ****Route Groups****

* **Çfarë janë:** Folderë që përdoren vetëm për **organizim të kodit** dhe **nuk ndryshojnë URL-në**.
* Për t’i krijuar, **rrethoni emrin e folderit me kllapa ()**:

### src ****Folder****

* **Çfarë është:** Një folder opsional ku mund të vendosësh **të gjithë kodin e aplikacionit**, përfshirë folderin app.
* **Qëllimi:** Ndani **kodin e aplikacionit** nga **skedarët e konfigurimit** që jetojnë kryesisht në root (si package.json, next.config.js, .env).