Angular Project Structure - Quick Notes

1. Root-Level Folders

- .angular: Internal Angular CLI cache/config. (Do not edit manually)
- .git: Git version control history.
- .vscode: VS Code editor settings.
- node_modules: Installed dependencies. (Auto-generated)
- public: Static files (favicon, etc.).
- src: Main source code (components, modules, etc.).

2. Inside src/

- app/: All components, modules, services.
- assets/: Images and static files.
- environments/: Config for dev/prod modes.
- index.html: Root HTML file.
- main.ts: Entry point (bootstraps AppModule).
- styles.css: Global styles.
- polyfills.ts: Browser compatibility.

3. Root-Level Files

- .editorconfig: Formatting rules.
- .gitignore: Files ignored by Git.
- angular.json: Angular build & project config.
- package.json: Project dependencies and scripts.
- package-lock.json: Locks dependency versions.
- README.md: Documentation file.
- tsconfig.json: Base TypeScript config.
- tsconfig.app.json: App-specific TypeScript config.
- tsconfig.spec.json: Testing TypeScript config.

4. What You Actually Modify

You commonly modify: src/app/, styles.css, angular.json, package.json

Do not modify: node_modules, .angular, .git

Must-Know Questions

Basic:

1. Purpose of angular.json?
2. Use of package.json?
3. Role of main.ts?
4. Difference between dependencies and devDependencies?
5. Why do we need node_modules?

Intermediate:

6. How does Angular find the entry point?
7. What happens when you run ng serve?
8. What is AppModule?
9. Purpose of src/assets?
10. How to configure environments?

Advanced:

11. tsconfig.app.json vs tsconfig.json?
12. Add global styles/scripts via angular.json?
13. How does CLI use angular.json?
14. What if you delete package-lock.json?
15. Difference between npm install and npm ci?

Mnemonic:

"People Always See More Stylish Angular Projects."

(Package.json, Angular.json, Src, Main.ts, Styles.css, App/, Polyfills.ts)