Chapter 02 - Assignment - Igniting our App

● - What is `NPM`?
NPM is the biggest, standalone repository for all package managers. All the packages, libraries, utilities that we would need are come from npm.
NPM does not stand for node package manager, It has no full form.

● - What is `Parcel/Webpack`? Why do we need it?
Parcel/Webpack are the bundlers that are used to minify, bundle or compress our code and make our application production ready.

● - What is `.parcel-cache`
It is a folder where parcel stores the build cache so that when we do the next build of our app, It refers to cache data and so build time reduces for subsequent builds.

● - What is `npx` ?
NPX stands for Node Package eXecute. It is used to execute the package that we have installed using npm.

● - What is difference between `dependencies` vs `devDependencies`
The packages/dependencies that we need while development only are installed as dev dependencies. And, the normal dependencies are installed without the -D flag as we need them in production also.

● - What is Tree Shaking?
In production build, Parcel statically analyzes the imports and exports of each module and removes everything that is not used (removes unused code). This is called Tree Shaking or Dead Code Elimination.

● - What is Hot Module Replacement?
HMR is one of the parcel use cases where it refreshes the webpage as we save the file. It uses file matching algorithm (written in c++) to refresh the page

● - List down your favorite 5 superpowers of Parcel and describe any 3 of them in your own words.

● - What is `.gitignore`? What should we add and not add into it?
It is a file that ignores the files/folders mentioned in it when we push our code to github.

The files/folders that we can regenerate, for example node_modules, don't need to be pushed on git or production because we can regenerate it using the “npm i” command. The files package.json, package-lock.json maintains the version of dependencies our app needs so we don't need to add these in .gitignore.

● - What is the difference between `package.json` and `package-lock.json`
Package.json lists all the approx versions of packages that are present in our app. There can be ^ or ~ in front of the dependency version as whenever there will be a new version, It will download the latest version based on that. Package-lock.json maintains the exact version of package/dependency.

● - Why should I not modify `package-lock.json`?
As it maintains the exact versions of dependencies and SHA hash code which needs to be the same when we install the same packages on the production server. So we should not change package-lock.json.

● - What is `node_modules` ? Is it a good idea to push that on git?
Node_modules folder contains all the dependency and sub dependency folders that we have installed. We can regenerate it by using npm i if we have package.json so It’s not good idea to push that on git.

● - What is the `dist` folder?
The dist folder contains the build of our application either production build or development build. It has compressed and minified code of our app in html, css and js files.

● - What is `browserlists`
browerlist is a package present in node_modules which is used to make our app browser compatible. For Ex- If we want our app to be compatible for last 5 versions of all the browsers then we can define that in package.json.we can make it work country specific also.

● - Read about dif bundlers: vite, webpack, parcel
● Read about: ^ - caret and ~ - tilda
● Read about Script types in html (MDN Docs)