

Namaste React

Homework Day - 2

Q1. What is `NPM`?

Ans - NPM is the package manager for the Node JavaScript platform. It puts modules in place so that node can find them, and manages dependency conflicts intelligently.

Q2. What is `Parcel/Webpack`? Why do we need it?

Ans - Parcel/Webpack combines a great out-of-the-box development experience with a scalable architecture that can take your project from just getting started to massive production application. Need of it is, Bundling, caching in development, Image optimization, local server create, file change watcher, dev build and production build etc.

Q3. What is `parcel-cache`?

Ans - Parcel cache create cache file which development build new file change build and show in local environment quickly file change build development.

Q4. What is `npx`?

Ans - A tool for executing Node packages.

By default, npx will check whether `<command>` exists in \$PATH, or in the local project binaries, and execute that.

If `<command>` is not found, it will be installed prior to execution.

Another major advantage of npx is the ability to execute a package which wasn't previously installed.

Q5. What is difference between `dependencies` vs `dev-Dependencies`?

Ans - A dependency is a library that a project needs to function effectively. DevDependencies are the packages a developer needs during development. DevDependencies: Babel, Webpack, Parcel.

Q6. What is Tree Shaking?

Ans - Tree shaking or dead code elimination means that unused modules will not be included in the bundle during the build process.

Q7. What is Hot Module Replacement?

Ans - Hot Module Replacement (HMR) exchanges, adds, or removes modules while an application is running, without a full reload.

This can significantly speed up development in a few ways:

Retain application state which is lost during a full reload.

Save valuable development time by only updating what's changed.

Instantly update the browser when modifications are made to CSS/JS in the source code, which is almost comparable to changing styles directly in the browser's dev tools.

Q8. List down your favourite 5 superpowers of Parcel and describe any 3 of them in your own words.

Ans - Dev server: start local development server which define default port if port busy then automatic add another port number

Hot Reloading: development files change watch and build show to browser

Development Target

Lazy Mode

caching: Caching development build to build only new changes file after first time run local file

HTTPS

Bundling

Q9. What is `.gitignore`? What should we add and not add into it?

Ans - gitignore file where any file or folder define that's will not push on github will be ignore file and folder

Gitignore file we should add auto generate files and folder.

Q10. What is the difference between `package.json` and `package-lock.json`

Ans - Although `package.json` is automatically generated and updated by npm, it can also be edited manually.

The problem with `package.json` is that you and your vulnerability scanner can end up seeing different versions of dependencies.

Yet worse, you could be running different versions compared to what is running in production.

package-lock.json avoid differences in installed dependencies on different environments and to generate the same results on every

environment we should use the package-lock.json file to install dependencies.

Q11. Why should I not modify `package-lock.json`?

Ans - The `package-lock.json` file was introduced in npm version 5 to solve this problem.

It is a generated file and is not designed to be manually edited. Its purpose is to track the

entire tree of dependencies (including dependencies of dependencies) and the exact version of each dependency.

Q12. What is `node_modules`? Is it a good idea to push that on git?

Ans - node_modules inside all package downloaded which use in our project and also dependencies of dependencies package downloaded

it's not required push on git because it's all module version and link define in package and package-lock.json files

When setup the project that's will be downloaded from npm.

Q13. What is the `dist` folder?

Ans - Dist folder contains development and production build files.

Q14. What is `browserlists`?

Ans - Browser list is define it's define in package.json file which browser and it's version supported js file will be generate.

Read about: ^ - caret and ~ - tilda

^ - caret : ^version "Compatible with version", will update you to all future minor/patch versions, without incrementing the major version. ^2.3.4 will use releases from 2.3.4 to <3.0.0.

~ - tilda ~version "Approximately equivalent to version", will update you to all future patch versions, without incrementing the minor version. ~1.2.3 will use releases from 1.2.3 to <1.3.0.
