**What is `NPM`?**

**Ans:-** Since react is just a library so any react project requires a lot of packages to run on its own so we need a package manager which provides what all we need to run our react app. So npm comes into the picture. npm is one of the various package manager like yarn. Once we initialize our app using npm init we will get package.json which contains all the configuration which our project needs. NPM is like maven for UI.

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

**Ans:-** These are 0 configuration build tool And once we install parcel/webpack we will get package-lock.json & in package.json we will have one dev dependency if installed it using option -d & we will get a new folder of node modules. Whenever we install any packages it become part of norde-modules. Hence node-modules is the database of our react-app. Parcel & webpack are bundlers.

**What is `.parcel-cache`?**

**Ans:-** Parcel needs some space for performing minification of files, HMR, File watchers and many more to do its functions.So all that is stored in parcel-cache.

**What is `npx` ?**

**Ans:-** This command is used to run a package from local or remote npm package. Major advantage of npx is to execute a package which isn’t installed.

**What is difference between `dependencies` vs `devDependencies`**

**Ans:-** Dev dependencies means which we need only in the development environment. These dependencies are required by our project and these are not global. They are not installed transitively (if A depends on B dev-depends on C, npm install on A will install B only).

Dependencies that your project needs to run, like a library that provides functions that you call from your code. They are installed transitively (if A depends on B depends on C, npm install on A will install B and C). dev dependencies are those that your project really needs to be able to work in production

**What is Tree Shaking?**

**Ans:-** Parcel performs tree-shaking that means removing unwanted code. Suppose app is importing some library which gives access to 10-30 functions and if we import that library then we get access to all those functions. So parcel is very powerful it will remove the unwanted code.

**What is Hot Module Replacement?**

**Ans:-** Basically whenever we save our changes in code it gets automatically reflected so it means parcel keeps a track of whatever changes we are doing**.** There is something known as file-watcher algorithm(written in c++) which is provided by parcel & its very fast and it tells the server to reload.

**List down your favourite 5 superpowers of Parcel and describe any 3 of them in your**

**own words.**

**Ans:-** The bold ones are the favourite

1. **Hot Module Replacement**
2. **File Watcher Algorithm – its written in c++ and it takes seconds to identify which file is changed and immediately reflects it.**
3. Bundling
4. **Minify – it minifies our code optimizes the image and even creates one minified file and change the variable name so that it can be easily bundled and deployed.**
5. Removes console.log/ Cleans the code
6. Manages dev & production build
7. Super fast build algorithm
8. **Image Optimization (minify the image also i.e. the heaviest thing in a website)**
9. Caching while development
10. Compresses the files (renames variable to minify code)
11. **Compatible with older versions of browsers – Adds polyfills**
12. Parcel gives functionality to develop HTTPS on dev (npx parcel index.html --https)
13. Parcel manages port number also
14. Consistent Hashing algorithm
15. Zero Configuration
16. Creates a server

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

**Ans:-**

* Package-lock.json not in gitgnore. Because we might not know th exact version
* We should put parcel-cache in gitignore because it can be auto-generated and it will be automatically generated on server and anyways parcel-cache is just a space which used by parcel.

**Read about: ^ - caret and ~ - tilda**

**Ans:-** tilda symbol will update you to all future patch versions, without incrementing the minor version.

Caret symbol upgrade you to all future minor/patch versions, without incrementing to the major version.

**Browserlists**

**Ans:-** browser list helps us to make our app compatible to older versions of the browser. We can pass the value to make it compatible for eg :- last 2 chrome versions so it will take care and replace the code which can be compatible to the older versions of the browser.

**What is `node\_modules` ? Is it a good idea to push that on git?**

**Ans:-** Node modules is the heaviest. So we should not push it on git and our package.json will have sufficient information to regenerate our node\_modules. So we will generate our node\_modules on our server. Package-lock.json has sufficient information to generate exactly same node\_modules.

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

**Ans:-** package.json contains all the info of the packages used in the react project with its version. Its mandatory for all project. But package-lock.json tells us the specific version with the hash value of each package which is used by our project. Package-lock locks the version, never to be kept in git ignore.

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

**Ans:-** 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.

**What is dist?**

**Ans:-** npx create a development build for us and hosts it on our server. And when we tell parcel to make a production build it creates 3 main file, index.css, index.js, index.html. So parcel bundles all our code and minified it and kept inside dist folder. Dist creates a faster development version of our project and keeps it on the server.

Command to create a prod build – npx parcel build index.html

Command to create a dev build – npx parcel index.html

**Why is react fast?**

**Ans:-** React is fast in itself and also react uses bundlers which works as a catalyst.

**What is browsersList ?**

Ans:- is a thing which will make our code compatible to lot of browsers. Basically its fed with lots of configuration.

Note:- we have our package manager which takes care of all the transitive dependencies.