Q what is NPM?

Ans- Node Package Module (which is not the official name of this) is the most conventional name offer by other developer out there. This is because that while working on the react we need different packages for make our app super cool. Hence when any package is installed it goes to NPM, from where we can excess it.

Key points:-

* this npm file can be regenerate it so we don’t need to put it into github instead of doing that we need to put it into .gitignore file .
* another reason is that it’s very heavy file.

Q2: what is parcel/webpack? Why do we need them ?

Ans:- Parcel/webpack are bundler for optimize , minify , compares the code for the better response while requested to the server. Mostly we deploy our code to production then we need the specific feature just to ensure the capability/performance of our app. Such features are:-

* bundling the app
* minify the code
* image optimization
* cleaning the code
* less configuration of parcel

Parcel. js is an open-source bundler. It supports many popular languages like Typescript and SASS, and can also handle file types like images and fonts.

* When there is any update or change is done into the app then parcel will automatically rebuild the app each time the file has saved.

Q3 what is ‘. parcel-cache’?

Ans: - This folder will create when we ignite the app with npx parcel (entery root). This file is used by parcel it’s self , like storing the information about project when build it form. So if further rebuilds is occur due to changes then it doesn’t require to re-parse and re-analyze everything from scratch.

Inshort it’s key reason why parcel is so fast in development mode. Like when the server is restart, parcel will only rebuild files that have changed since the last time it ran.

Q4 what is NPX?

Ans: - it is stand for Node Package eXecute, it is simply an NPM package runner. NPX allows you to run and use packages without needing to locally or globally install them.

Q5 what is dependencies and devDependencies ?

Ans: -

* Dependencies : These are the following libraries which used in project for production environment and for different functionality use in the project. In short project is build with different packages or dependencies , it’s like they provide super power to project to work efficiently. E.g, react library, react-router-dom etc.
* Devdependencies: these are those packages in the package.json file that we need only for project development purposes. Which is present globaly like dependencies. E.g Babel, webpack, parcel etc.

Q6 what is Tree Shaking ?

Ans: -

* Tree shaking , It is the term refer/used to removal of dead code in the current application of JavaScript . It relies on the import and export statement to detect if code modules are exported and imported for use b/w JS file.
* Dead Code Elimination : - is the process of removing code that is not used by the current application. Code is parsed to create an Abstract Syntax Tree which is then traversed to find unused functions and variable, and finally the tree is converted back to JS source code.
* But present JS app , we use bundlers( parcel,webpack) to automatically remove dead code when bundling multiple JS files into single files.

Q7 What is Hot Module Replacement?

Ans:-

* HMR( Hot Module Replacement), is the feature of parcel( or any other bundler) that improves the development experience by automatically updating the changes in the browser at runtime without refreshing the whole page.
* It will retain the state of application as you change small things
* This support both JavaScript and css assets.
* As we safe the file, parcel rebuilds what changed and send an update to any running clients containing the new code. Hence this new code replace the old version and is re-evaluated along with all parents.

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

Ans: - Following super power of parcel are:

* Parcel has an in-built dev-server and hot module replacement to help us to update the changes in the running browser without refresh the full page.
* File watcher algorithm, through which parcel will check the changes occur in different file/folder.
* Minifying
* Cleaning the code
* Development and production build
* Image optimization
* Caching while development
* Compression some file name ( including hashing)
* Compatible with older version of browser.
* Also give the functionality to work on Https on dev environment port number.
* Consistent hashing algorithms
* Zero configuration required.

Some import point about parcel .

Caching – (it’s like the hidden storage) parcel caches everything it builds to disk. If we restart the dev server then parcel will only rebuild files that have changed since the last time it ran.

* Parcel will automatically tracks all of the files, configuration , plugins, dev dependencies that are involved in build which we form using the parcel command.
* E.g if we change configuration file , all of the source files that rely on that configuration will be rebuild.
* This cache is stored in the .parcel-cache folder in our project. And we can put it into .gitignore file cause this can re generate while on the server.

Dev server – it’s the inbuild feature of parcel that automatically started when we use default CLI npx parcel serve.

* it starts a server at <http://localhost:1234> , if port 1234 already used and we create another project then fallback port will used.
* It also support HTTPS , in case use a specific hostname for authentication cookies , any other issues, then we need this HTTPs. And we can achieve this by adding CLI flag –https only.
* Npx parcel (root entry) –https

File watcher-- , This the algorithm used by parcel to support optimal caching and development experience. This algorithm written in c++ .

* With the use of this every file in the project ( including all node\_modules) are watches by parcel.

Minification-- , parcel use minifier for JS, CSS , HMTL, and SVG , by using these minification we reduce the file size of output bundle file by removing whitespace,rename variable to shorts name etc.

* Image optimization , parcel support resizing , converting and so on.

Content Hashing---, Parcel automatically includes content hashes in the names of all output files, which enables long-term browser caching.

Q9 What Is ‘.gitignore’? what should we add and not add into it?

Ans: -

* This is the file in GIT, which specifies untracked files that git should ignore, mean Those file we can re generate are lie into this file and those file which can’t regenerate are push into the project.
* E.g node\_modules are place into it and package.json file is important which can’t be put into the .gitignore file.

Q10 what is the diference b/w package.json and package-lock.json?

Ans: -

* Package.json – file consist the detail of project such as version,licence , description and also having the details of dependencies uses in the project. But those dependencies are display with ^(carot)\_means dependencies version is keep updating automatically as their new update available in server.
* Package-lock.json—is also the same file but main difference is that , is used for production build, because it contain the specific version of any dependencies along with supportive dependencies( also called as transitive dependencies.). It’s purpose to track the entire tree of dependencies and exact version of each dependency.

Q11 why shouldn’t we change in the package-lock.json file?

Ans: -

* It is because that it is a generated file and is not designed to be manually edited. It holds information on the dependencies or packages installed for a node. js project, including their exact version numbers

Q12 what is the node\_modules ? is it good idea to push that on git?

Ans: -

* Node modules is the package manager( which is most developer give it to the name). which consist Tree of all the given dependencies into the project. When ever the dependencies is install then it put into the node\_modules along with it’s supportive dependencies ( also called transitive dependencies).
* No it’s not good idea to put node\_module into git version system. Cause it is really heavy in size and also it can regenerate when pushes into the server.
* All you have to need package-lock.json file into the github so that other members can see it

Q12 what is the dist folder ?

Ans: -

* When we ignite our app with command npx parcel (entry root e.g index.html) then this folder is created along with .parcel-cache. This dist folder hold the production build file.

Q13 what is ^ (caret) and ~ (tilde)?

Ans: -

* ^ (caret) – this represent the upcoming updating version of specific dependencies. E.g if ^react.18.2.0 , in future if this dependency is developed or changed then instead of installing again , it will automatically upated to next version.
* ~(tilde)—this represent the maximum version of dependencies.

Q14 Tell me about the Script types in html?

Ans: -

* This type attribute specifies the type of the script.
* The type attribute identifies the content between the <script>and </script> tags.
* Most of the cases which is default like application/javascript

Q15 what is BrowserList ?

Ans: -

* Browserlist can specify the version along with which browser is used for running the web application.
* It provides a configuration for specifying browser range. It become standard and used by libraries such as Babel parsel, webpack etc.