

Class - 2 Assignment

Theory Assignment

Q What is NPM?

Ans → NPM is no standard of full form at it is a package manager, which manage our package, install the package and delete the package so on.

→ npm is the largest Software Registry over more than 10 million, and become the center for JS code sharing.

Q what is parcel / web pack ? why do we need it

Ans → Basically there are The Bundlers we often use as a package, This Package helps us to bundle our vast application different code to a single, optimised, minified piece of code which need for the production:-
why do we need it?

Basically production level code is different a lot of factors takes place when we move from local host to server, ex

Manifest, Optimization, clean code, unused code, etc

And we know Browser under HTML, CSS, JS, { when we replace 3 think in JS, React, SASS, we need some Bundlers

to do some optimisation, minification in our application in under hood.

In Term Of (Parcel in a Boat)

- HMR - Hot Module Replacement
- file watching algo CTF
- Compatibility with older version
- Minification
- Consistency Algo for Bundles
- Compression
- HTTPS Order
- Caching
- Pack Number & many more.

Q what is npx?

Ans x-executor (Node Package executor, it comes with npm, when you install npm above 5.2.0 version then automatically npx will installed. it is an npm package runner that help us to execute the package which are registered on npm with installing or without installing that package.

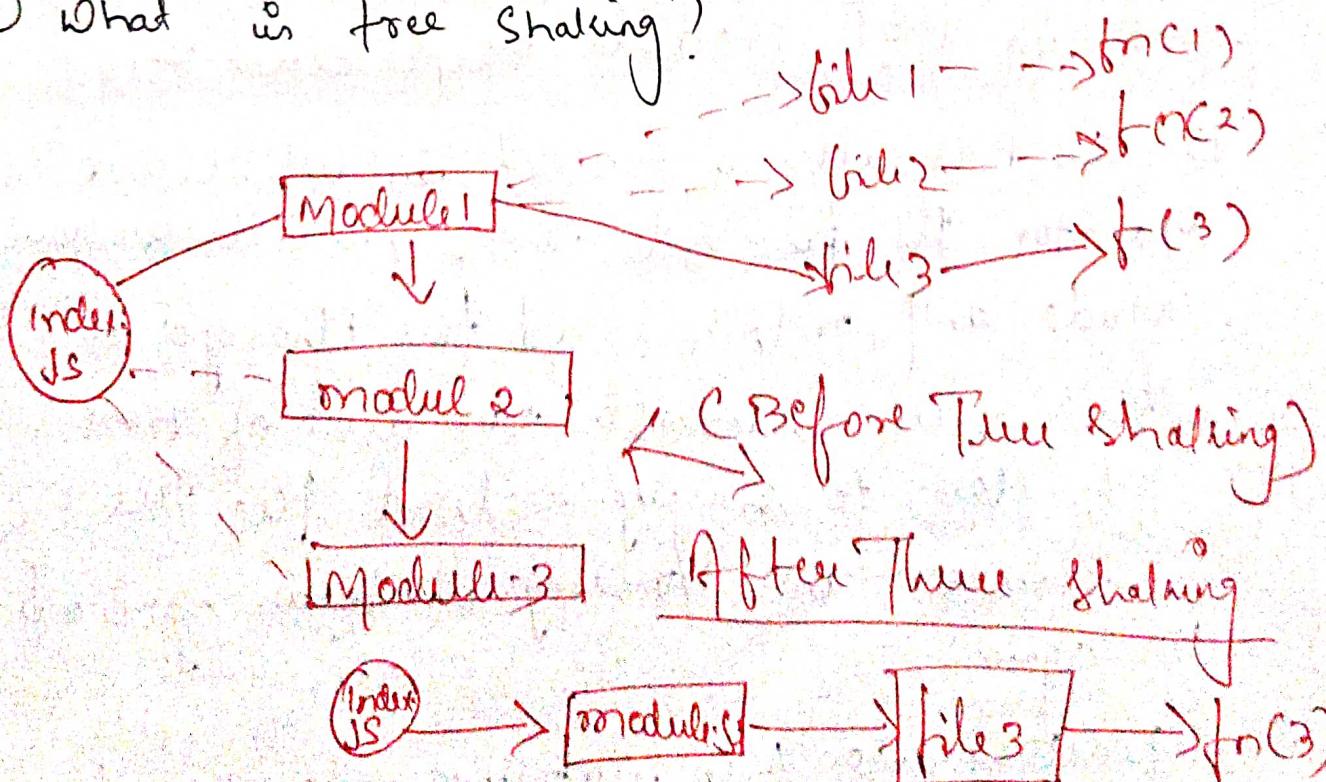
- ⑩ what is difference b/w dependency and dev dependency
- Ans → dev dependency are the dependency which is available for the help of developer, which is available with developer environment
- Ex - node mon (HMR) Type
→ parcel, web pack

Syntax → npm i → nodemon

or
npm install --save-dev nodemon

Dependency are something is Global in nature in which the application dependent → need to available in client site. Ex - jsonwebtoken, bcrypt.js, dotenv, Many more ..

- What is tree shaking?



Tree Shaking is a term commonly used in JavaScript Context describes the removal of dead code.

- It is relies b/w import and export which statement to detect if code module are exported and imported for us between JS file.
- In Modern JS applications, we use module bundler like Webpack, parcel which automatically handle the Tree Shaking or remove dead code, which is ready for the production level code.

① What is HMR Module Replacement:-

- It is basically a concept in which we monitor all the file change, which we have made and auto-reload the changes in the browser, which save a lot of time of dev to simply reload the browser or restarting the browser again and again.
- Parcel use file watching algorithm to achieve HMR.

Q) List down your fav 5 super powers of parcel and describe any 3 of them in your own word.)

Ans → HMR - Hot Module Replacement

Cache -

Image Optimisation -

Compatibility of older version

Consistency algo for the bundlers.

Description

HMR → Hot Module Replacement, As you made any changes in your file parcel automatically changes the and rebuild the changed file, and upload your app in the browser;

→ In some code we refresh the hole page and also uses HMR - which improve the developer experience by only updating module in the browser without reloading the page.

Some api - module hotC :)

module.dispose()

Any changes has made enter here at dispose and the module unload.

Caching :- Generally the caching term, gives us the idea of temporary memory, the same phenomena apply here whenever we build or start the server if create a panel cache file with containing the cache of all about the app, whenever we do change in any module it changes or see build files only.

→ Panel tracks all of the file, config, plugin and dev dependency that involved in your build any time you changes in files module or config. It change the file which are relies on them.

HTTPS Sometimes, You may need or use HTTPS during development. For ex., You may need to use certain host name for authentication worries, or authentication cookies or debug related context issues.

→ Panel dev's support HTTPS out of the box. You either use automated generated certificate or provide your own.

Q What is .gitignore? what we should ~~not~~ not be add into it.

Ans → it file in which we add those folder or file which we do not want to commit or push, which means any file which can be regenerated, change, ~~can not~~ should be mentioned in the .gitignore.

Q What is the different between 'package.json' and 'package-lock.json'.

Ans → when ever we are init the npm - i nit in our directory, we get a package.json file which ^{contains} a lot of info like package name, author name, entry point, script and testing and a lot more.

→ And we add any dependency or the dependency will get another file and node modules, package-lock.json

→ Ex if we add xyz package as dependency and at mention as the package.json with version " ^1.2.3 " which means it load or update update automatically when the new version is available without us.

→ But the lock file contains the exact version of the package which we install and along with that dependency which we have with that package.

Q Why should not I modify the package-lock.json

Ans → As it contains the sensitive information about the package version and info about the dependency (franchise) and when you try to create node_modules after running the command npm install node → take a look in the package-lock.json.

Q What is node-modules? Is it good idea to push that on git?

Ans → Node-modules is just like the database for the npm. Whenever we install any package at get install and the code file will available inside the node-modules. As this is the file which can be run again, so it is foolish to add this in git.

Q What is 'dist folder'?

Ans → whenever we create or execute npx parcel index.html or build

npx parcel build index.html, we get a .dist folder, which contains 3 files mainly, i.e. bundle.js, index.html and css.

These are mini files which → ready for the production.

④ Different Script?

Only we are writing module type at module.

`<script type="module" src="js.js">`

→ It tell the browser that whatever js are we encounter it is a part of module that a module we can import and import and export they would.