

Chapter 02 - Igniting our App - Theory

1.What is NPM

- It is a package manager
- Command line utility which aids in package installation, version management and dependency management

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

- Parcel is a beast
- React alone is not sufficient to create whole application we need a lot of super powers for that which are provided by parcel
- Parcel is dev dependency and we need it to aid few processes while development like live server , file watching , build etc.
- Parcel helps in a lot of things including -
 - i. Hot module replacement
 - ii. File watcher algo
 - iii. Bundler(bundles all or code)
 - iv. minification
 - v. Cleaning our code (removing console log)

- vi. Dev and prod build
- vii. Fast algo
- viii. Image optimisation (as media takes most time to load in website)
- ix. Caching while development (older build took 12-15s from onwards just took 3-4ms)
- x. Compressing > everything in index.html , variables are also renamed
- xi. Compatible with older versions
- xii. HTTPS on dev > just add flag in npx —https
- xiii. Created a server
- xiv. Port number (if you have multiple local website served it will automaicaly manage new port for you)
- xv. Consistent hashing algo (to cash things up)
- xvi. Zero config

3.What is ``.parcel-cache`

- It is a folder which is generated automatically once we run `npx parcel index.html` .
- This folder is created by parcel to speed up build speed and also it needs some space to perform execution for the features it provides , for example file watcher algo in c++ needs some space to be executed etc.

4. What is `npx` ?

- Npx is a command we use in CLI which tells to execute with npm
- Ex: `npx parcel index.html`

5. What is difference between `dependencies` vs `devDependencies`

- Dependencies are packages that will be required by our application in both dev and prod
- Dev dep are packages that are only required while developing locally/testing and we don't need in prod

6. What is Tree Shaking?

- Js context to describe Removal of dead code
- It relies on the import and export statements to detect if code modules are exported and imported for use between JavaScript files.
- Bundlers like webpack/parcel automatically removes dead code

7. What is Hot Module Replacement?

- Exchanges , add and removes the modules while an application is running without full reload
- It speeds up development and saves time by only updating what's changed

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

- File watcher > interesting algo in c++ which determines which files needs to be rebuilt

- Bundler > this is also interesting as everything is optimized and kept under one single file, also how it determines the code flow and accordingly adds up in the file
- Hot module replacement > add,removes,changes module dynamically while our application is running without full load
- Minification
- Compressing
- Caching while development

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

- It is file in git repo in which we can add the name of the files which git should ignore
- We should add all the files in gitignore which can be generated with help of any other file and hence can be generated in server as well. Ex- package.json have all the info of packages installed hence we dont need to push our node-modules on remote or server

- All files which cant be regenerated should not be added to gitingore

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

- Package-lock.json locks all the versions of dep, as well as sub dep making sure all the version remains intact so that it can be used further on any other machine/server which wont cause our app not to work.
- As we hear a lot time people saying its working on my machine but not there . Version can be reason for this. Hence this problem is solved with lock.json
- Package.json keeps all the packages installed but here version can be upgraded to minor updates like 10.0.1 to 10.0.9
- Package.json also define how to communicate and execute your api. It also contains cli commands.

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

- Don't modify sub dep versions

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

- Node-module is a folder in our project which contains minified code for all dep packages we require in our application
- No it should not be pushed to git as it can be regenerated using package.json

13. What is the `dist` folder?

- When we build our application the minified and optimized files are stored inside dist folder

14. What is `browserlists`

- It is a package which make our application supported in older browsers

15. ^ - caret and ~ - tilda

- Using tilde ~ gives you new bug fix releases for the package in same version installed
- Using ^ caret gives you backward compatible new functionality

16.