Assignment 2

▼ What is `NPM`?

- No official full form of NPM though it is called as node package manager
- We use npm to share and borrow packages. React cannot be built only using react, it also needs other packages and to use these packages we use npm

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

Parcel is a bundler as is webpack.

We need it for many reasons -

- Hot Module Replacement
- Minifying
- Cleaning our code
- Super Fast build Algo
- Caching while development
- Compatible with older version of browser
- zZero Config
- Tree Shaking - Removing unwanted code

▼ What is `.parcel-cache` ?

The `.cache` folder stores information about your project when parcel builds it, so that when it rebuilds, it doesn't have to re-parse and re-analyze everything from scratch. It's a key reason why parcel can be so fast in development mode.

▼ What is `npx` ?

It comes with npm, It is basically for Xecution → npx parcel index.html

▼ What is the difference between `dependencies` vs `devDependencies` ?

DevDependencies are used in our DEV environment which we don’t want globally but to only stay in our machine while dependency is something which we want globally.

▼ What is `Tree Shaking`?

Tree shaking is a term commonly used in the JavaScript context for dead-code elimination.

▼ What is `Hot Module Replacement`?

HMR enables you to see your changes in the browser almost immediately as you make them, usually without the need to refresh the page or lose your application state.

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

.gitignore is a file that you can create in a Git repository to tell Git which files or directories to ignore when you make a commit. This can be useful if you have files that you don't want to track, such as temporary files or build artifacts.

Anything that can be autogenerated should be put in .gitignore. For ex: With the help of package.json we can generate node_modules folder, thus it should be put in .gitignore

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

As we know the parcel is a beast, it has many features,

- **Dev Server**: By default when we open our server it opens at port: 1234. If port 1234
  is already in use, then a fallback port will be used. After Parcel starts, the location where the dev server is listening will be printed to the terminal. We can even open in our favourite web browser.

  npx parcel index.html --port 3456
  npx parcel index.html --open safari.
  Igniting our App 6
- **Hot Reloading:** Whenever we make changes to our code, parcel automatically detects it and does the changes in the browser. Parcel has HMR (Hot module Replacement) which improves the development experience in the browser by updating modules in the browser at runtime without needing a whole page refresh.
- **HTTPS:** Sometimes we need to test code which are only possible in HTTPS like authentication cookies etc. Parcel provides us HTTPS support
  parcel src/index.html --https

Other features includes
Bundling
Caching While Development

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

package.json	package.lock.json
It contains basic information about the project.	It describes the exact tree that was generated to allow subsequent installs to have the identical tree.
It is mandatory for every project.	It is automatically generated for those operations where npm modifies either node_modules tree or package.json.
It records important metadata about the project.	It allows future devs to install the same dependencies in the project.
It contains information such as name, description, author, script, and dependencies.	It contains the name, dependencies, and locked version of the project.

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

We have no need to push node_modules on git, since it can be installed with the help of package.json

It is a directory that is created by the Node package manager (npm) when you install packages in a Node.js project. It contains all of the packages and their dependencies that are required by your project, as well as any scripts that are associated with the packages.

When we are building a production-ready app we need a lot of things like compression, minification, optimization, caching, bundling, hot reload etc . we can’t do this alone and we need some dependencies and those dependencies

are also dependent on other dependencies it creates a `dependencies tree` This is called ~transitive dependency~. Thats why node_modules is heavy.

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

because it contains the exact version of the files downloaded so that they can run on different environments.

▼ What is the `dist` folder?

1. The **/dist** stands for **distributable**.
2. The /dist folder contains the minimized version of the source code.
3. The code present in the /dist folder is actually the code which is used on production web applications.
4. Along with the minified code, the /dist folder also comprises of all the compiled modules that may or may not be used with other systems.
5. It is easier to add files to the /dist folder as it is an automatic process. All the files are automatically copied to the dist folder on save.
6. The /dist folder also contains all those files which are required to run/build a module for use with other platforms- either directly in the browser, or in an AMD system (eg. require.js).
7. Ideally, it is considered a good practice to clean the /dist folder before each build.

▼ What is `browserlists` ?

It let;'s us specify how many browser versions we need our application to run on, or if we need some specific browsers so only on those our application should run.

Read about dif bundlers: vite, webpack, parcel

- Read about: ^ - caret and ~ - tilda
- Read about Script types in html (MDN Docs)