Owner - Mohit R
Assignment - 2

1. What is NPM?

- NPM manages packages or is a package manager. In other word, NPM is a standard repository for ALL packages.
- All packages, dependencies, utilities are hosted in NPM.
- It keeps track of all our packages which we can use in our project.

2. What is parcel/webpack? Why do we need it?

- Parcel/Webpack are bundlers which are used to bundle files in our project before it is shipped to production.
- It is useful to optimize the experience for the user and developer.

3. What is parcel-cache?

- Parcel-cache is a file where parcel does caching while building the project for production or development.
- It mainly caches the files and details to check if any file has been changed or not.
- If any file has been changed, it build the page again but the build time is very less compared to the first time because of the cache. It is generated automatically.

4. What is 'npx'?

- NPX (node package eXecute) is similar to NPM but is used to execute our dependencies that have been installed with npm.
- It allows developers to execute any JS package available on the npm registry without even installing it.

5. What is the difference between 'dependencies' and 'devDependencies'?

- Dependencies are normal dependencies which are also used in production. On the other hand, devDependencies are mainly used during development phase.

6. What is Tree shaking?

- Tree shaking is an algorithm used by parcel to remove unused code from our project.

7. What is Hot Module Replacement?

- HMR or Hot Module Replacement is used to replace modules while the application is running without a full reload.
- It automatically refreshes the web when you only save your code. There is need to refresh the page.

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

- Local server: Basically it creates a port and deploys our application in the server.
- Dev build: when parcel is executed, it generates the development build and stores inside the 'dist' folder, which we see in our browsers.
- File watching algorithm:Monitors each and every folder for arrival of files.upon arrival it implements a certain process.
- Image optimization.
- Gives a way to host the application in HTTPS when you want to test something that only uses SSL.

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

- .gitignore is used to store those files, dependencies and folders which we don’t want to push to Github/ production.
- We should add those files which can be regenerated and not the ones which cannot.
- For example: node_modules can be skipped as it can be generated through package.json and package-lock.json file.

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

- package.json keeps an approx record of our versions dependencies. We can make use of caret (^) or tilda(~) to keep the minor or major changes.
- package-lock.json keeps the exact record of the dependency version.

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

- We should not modify our package-lock.json because it keeps exact record of the dependency version along other dependencies.
- It maintains the transitive dependencies.

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

- node_modules is basically a database of all our packages/ dependencies.
- node_modules -> collection of dependencies.
- It stores all the dependencies that is required by our project along with their dependencies (**transitive dependencies**).

12. What is dist folder?

- 'dist' folder is the development build that is built when parcel is executed.

13. What is browserslist?

- Browserlists are used in our application to make it compatible with the older versions of browsers.
- We need to specify the older versions of browser which need to be used within our project.
- Ex:
  "browserlist: [
  "last 2 versions"
  ]

14. What is caret - (^) and tilda - (~)?

- **Caret(^)** - It is usually used to specify a range that allows for both minor and patch updates.

- We can say that our project can use any version of the package which has the same major version but greater or equal to the minor version.
- Example: parcel: “^1.2.3” —> We can have any major version 1 and equal to the minor versions i.e 1.3.0, 1.4.0 but not 2.0.0.
- **Tilda (~)** - It is usually used to specify the range that allows patch updates. Patch updates are usually for patch updates and not for any breaking changes.
- Example: parcel: “~1.2.3” —> 1.2.4, 1.2.5, 1.2.6 etc but cannot have 1.3.0.
- It is generally better to use caret(^) over tilda(~).