1.What is NPM?

Npm is a package manager through which we can manage various dependencies in our app. It is a repository for all packages.It also supports version controlling of the added packages. NPM is the world’s largest software registry. Open source developers from every continent use npm to share and borrow packages, and many organizations use npm to manage private development as well.  
Npm consists of three distinct components:

* the website.
* The command line interface (CLI)
* The registry.

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

Parcel is a bundler which is used for bundling all the code. Parcel is used to make our application production ready by optimizing our files by doing various operations such as code splitting, bundling, etc. and making our app ready for deployment on production.

Both Parcel and Webpack are Bundlers for our project. This Bundlers will help us to done many

things such as "refresh automatically", "Good Error Handling", "Speed up our project" etc.

The Most/Widely use bundler is Parcel. By the help of these bundlers we can increase our productivity

Lots of work is done by these Bundlers:-

- Caching The code

- Tree Shaking- Remove Unused code

- Image Optimization

- Code Compression

- Bundling The file

- Differential Bundling - Support Older Version also

- Diagonistic

- Code Splitting

These all things will increase our project loading time, our productivity, automatic reload etc.

3. What is '.parcel-cache'.

- As I know Parcel Bundler Cache our data. These all data is store inside the .parcel-cache folder.

By using caching technique, if there is no changes made to a file then it cache the data from

.parcel-cache folder otherwise it build again and store as cache.

4. What is 'npx'?

- we know that 'npm' is responsible for managing the package same as 'npx' is responsible for

executing the package.

Ex:- npx parcel index.html

5. What is difference between 'depandencies' and 'devDependencies'?

- Dependencies:-

When we created a file without any keywords then by default all the file is created inside dependencies.

These type of file is running in both Development and Production Environment.

ex:- npm i react

DevDependencies:-

When we created a file with keyword -D then all the file is created inside devDependencies.

These type of file is only accessable in Development Environment.

ex:- npm i -D parcel

6. What is Tree Shaking?

- Tree Shaking:- It is a process in such that it remove unused code or dead code from our files.

By remove the dead code from our project it will increase loading time and execute

smoothly.

7. What is Hot Module Replacement?

- Hot Module Replacement:- It is a feature provided by bundlers to refresh/reload the project at

runtime without full refreshing the page. It is commonly known as HMR. It automatically reload/

refresh the project.

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

- Parcel is a very powerful bundler. It has provided many feature to an application such that user

experience increase. Parcel provide lots of feature to an application. 5 Features are listed

below:-

a. Code Caching

b. Code Compressing

c. Code Splitting

d. Differential Bundling

e. Excellent Error Handling

a. Code Caching:- In this feature, Parcel cache the data which is not frequently changes mean

that if a file data is not change then it store previous data otherwise its update the file

data inside cache. By the help of code caching, The load time of our application takes less

time. It has store the cached data inside a folder called .parcel-cache.

b. Differential Bundling:- With the help of differential bundling feature, We can run multiple

version of application including previous version. Parcel will create multiple bundling file

for same application.

c. Excellent Error Handling:- Parcel has provide excellent error handling feature to us. it will

give us good error message to us if any error occured to our application. such that we read

this message and then we can easily solve the problem/error. Apart from this it also provide

colourful error message such that user experience became increase.

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

- `.gitignore` is a special type of file which is responsible for restrict to upload a file on git.

If I write any file name inside this file then it restrict all the file (which is inside `gitignore`)

pushing to git. This file is very useful when we do not want to push a particular file to git due to

any reason.

In .gitignore file, We should write all the file name which we can generate any time with the help of

file configuration such as node\_modules.

In .gitignore file, We should not write file name which is responsible for execution of application

such as configuration file , self writing code file etc.

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

- package.json is a configuration file for our application. It has all the details related to current

application such as author, version,licence, dependencies etc. whereas package-lock.json is a file

containing our application exact details such as exact version current dependencies etc.

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

- The `package-lock.json` file containing exact details of our application such as package,dependencies

etc. These details are responsible for running our application. If any changes made to this file

may produce some error for running. Therefore We should not modify 'package-lock.json` file.

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

- `node\_modules` is a special folder which is generated according to our application dependencies.

This folder contains all the dependencies details as well as transitive dependency details.

This folder is automatically generated when ever any package install using npm. This folder is

directly install from production environment.

We should not push `node\_modules` folder to git because it is too heavy and we can generate this

folder according to our application configuration any time using npm package manager. Therefore It

is not a good idea to push 'node\_modules` on git.

13. What is `dist` folder?

- 'dist` is a special folder which is responsible for rendering our app on server. `dist` stands for

distibution. Initially `dist` folder contains all the development build file but when i build the

application in production environment then all production build file created inside `dist` folder.

`dist` folder is responsible for sending data to server means that data present on localhost server

is rendering from this `dist` folder.

14. What is `browserslists`?

- `browserslists` :- As We know that parcel provide browser compatibility mean that multiple version

supports by our application. `browserslists` is responsible for this feature. `browserslists` is an

array which is used to filter based on browser wise, version wise , region wise etc.

Ex:-

“browserslist”: “last 2 versions”

“browserslist”: “last 2 Chrome versions”

“browserslist”: “cover 99.9% in US ”