Theory ---2

1. What is npm? :

It is a tool used for package managemnet.

Note :

npm does not stand for node package manager but everything else.
npm alternative -> yarn

npm setup :

npm init
npm init -y can be used to skip the setup step, npm takes care of it and creates the package.json json file automatically , but without configurations.

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

Parcel/Webpack are type of bundlers that we use to power our application with different type functionalities and features.

3. What is .parcel-cache?

.parcel-cache is used by parsel to reduce the building time. .parcel-cache contains the imformation about the project so that it can reduce the building time if we rebuild the application again.

4. What is npx ?

npx is a tool that is used to execute the packages registered on the npm registry without installin them.

5. What is difference between dependencies vs devDependencies ?

A dependency is a library that a project needs to function effectively. DevDependencies are the packages a developer needs during development.

6. What is Tree Shaking?

Tree shaking is process of removing the unwanted code that we do not use while developing the application

7.What is Hot Module Replacement?

Hot Module Replacement or HMR is a feature that bundler provides. HMR uses File Watcher algorithms to keep track of file changes while development and renders those changes on the Web UI.

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

HMR - Hot module replacement : Parcel provides HMR properties to keep track of files changes by using file watcher algorithms.
Compression : Parcel provides compression properties to compress our files e.g images to optimize the performance of our app.
Minification : Parcel minifies the code.
Zero config : Unlike Webpack, Parcel requires zero configurations to setup.
Old browser competibility : Parcel allows support for older broser also.

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

This file contains the list of files that we should not commit to the repository.

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

this file is mandatory for every project
It contains basicinformation about the project
Application name/version/scripts (ng scripts)
package-lock.json :
This file is automatically generated for those operations where npm modifies either the node_module tree or package-json.
It is generated after an npm install
It allows future devs & automated systems to download the same dependencies as the project.
it also allows to go back to the past version of the dependencies without actual ‘committing the node_modules folder.
It records the same version of the installed packages which allows to reinstall them. Futuee installs wll be capable of building identical description tree.
~ or ^ in package.json file : These are used with the versions of the package installed.
For example in package.json file:

```
"dependencies": {
   "react": "^18.2.0",
   "react-dom": "^18.2.0"
  }
```

~ : we can use it for minor version updates for a package.
^ : we can use it for major version updates for a package.
If none of them is present , that means only the version specified in package.json file is used in the development.

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

package-lock.json file contains the information about the dependencies and their versions used in the project. Deleting it would cause depencies issues in the production environment.

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

node_modules is a folder that contains all the packages and dependecies used the the development of the applicaion. \n It is big file that we should not push to github.

13. What is the dist folder?

dist folder is created when parcel creates the build for the application . It contains the HTML and CSS and JS with other files for the build.

14.What is browserlists?

browserlist is a list brosers that our applicaion should support . It also covers the browser coverage over a geographical area.

Build a production version of your code using `parcel build` ?

Parcel’s production mode automatically bundles and optimizes your application for production. It can be run using the **`parcel build`** command:

```
parcel build src/index.html
```

Parcel includes minifiers for JavaScript, CSS, HTML, and SVG out of the box. Minification reduces the file size of your output bundles by removing whitespace, renaming variables to shorter names, and many other optimizations.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8"/>
    <title>My First Parcel App</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```