DAY-2:

1.What is NPM?

NPM is default package manager for JavaScript’s Node.js. It consists of CLI tool for publishing and downloading packages and an online repository that hosts JavaScript packages.

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

Parcel/Webpack are bundler packages that combines development experience with scalable architecture that can take our project to a massive production application.

It helps to pack the assets of web applications(code,images,packages) into bundles so that applications can be served easily.

3.What is parcel-cache?

The .cache folder (or .parcel-cache in parcel v2) stores information about your project when parcel builds it, so that when it rebuilds, it doesn't have to re-parse and re-analyse everything from scratch. It's a key reason why parcel can be so fast in development mode.

4.What is npx?

It is an npm package runner that can execute any package from npm registry without even installing the package.

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

| **Dependencies** | **devDependencies** | **peerDependencies** |
| --- | --- | --- |
| A dependency is a library that a project needs to function effectively. | DevDependencies are the packages a developer needs during development. | A peer dependency specifies that our package is compatible with a particular version of an npm package. |
| If a package doesn’t already exist in the node\_modules directory, then it is automatically added. | As you install a package, npm will automatically install the dev dependencies. | peerDependencies are not automatically installed. You need to manually modify your package.json file in order to add a Peer Dependency. |
| These are the libraries you need when you run your code. | These dependencies may be needed at some point during the development process, but not during execution. | Peer dependencies are only encountered when you publish your own package, that is, when you develop code that will be used by other programs. |
| Included in the final code bundle. | Included in the final code bundle . | Can be included only when you are publishing your own package. |
| Dependencies can be added to your project by running :  npm i <package\_name> | Dev dependencies can be added to your project by running :  **npm i <package\_name>**  **--save-dev** | Change the package.json file manually. |

6. What is Tree Shaking?

In modern JavaScript applications, we use module bundlers (e.g., webpack or Rollup) to automatically remove dead code when bundling multiple JavaScript files into single files. This is important for preparing code that is production ready, for example with clean structures and minimal file size.

7. - What is Hot Module Replacement?

Hot Module Replacement (or HMR) is**one of the most useful features offered by webpack.** It**allows all kinds of modules to be updated at runtime without the need for a full refresh.**

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

1.Hot Module Replacement

2.Tree shaking

3.Differential bundling

4.consistent hashing

5.code splitting

1.Hot Module Replacement:

Hot Module Replacement (or HMR) is**one of the most useful features offered by webpack.** It**allows all kinds of modules to be updated at runtime without the need for a full refresh.**

2.Tree shaking:

In modern JavaScript applications, we use module bundlers (e.g., webpack or Rollup) to automatically remove dead code when bundling multiple JavaScript files into single files. This is important for preparing code that is production ready, for example with clean structures and minimal file size.

3.Differential Bundling:

Differential bundling is the concept of sending various copies of your code to different targets and letting the browser decide which one to download.

Instead of transpiling modern syntax into older JavaScript versions, shipping it can reduce bundle sizes and improve load times.

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

.gitignore file is used in a git repository to ignore the files and directories which are unnecessary to project this will be ignored by the git once the changes as been committed to the Remote repository. Modules or files which can be recreated in production are usually ignored. eg:node modules,dist folder,parcel-cache.

10. 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. |

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

It describes a single representation of a dependency tree such that teammates, deployments, and continuous integration are guaranteed to install exactly the same dependencies.Hence modifying it will cause mismatch between local and production environment.

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

The **node\_modules** folder is a directory downloaded automatically while setting up our node project. This folder plays a crucial role in Node.js, it acts as a directory that will store all the dependencies and modules your project is dependent on. This also implements version control which manages the development environment for every specific project.

It can be recreated in production or other environment using npm install and hence not good idea to push on git.

13. 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).

14. What is `browserlists

Browserslist is a tool that allows specifying **which browsers** should be supported in your frontend app by specifying "queries" in a config file.

15.Difference between webpack and parcel

Parcel:

1.Suitable for small-medium sized projects

2.zero-configuration bundler -Does not require external configuration file.

Webpack:

1.battle-tested bundler. Highly flexible.

2.If more customization needed, we go for webpack.

14.Difference between tilde and carat

The tilde (~) notation is employed to match the latest patch version while freezing the major and minor versions.

* **Example:**The ~1.2.0 will update all the future patch updates. We have to write just ~1.2.0 and all the next patch update dependencies. For example, 1.2.**1,**1.2.**2,**1.2.**5**……………1.2.x.

It automatically updates both minor and patch updates.

* **Example:**The^1.2.4 will update all the future **Minor**and **patch**updates, for example, ^1.2.4 will automatically change the dependency to **1.x.x** if any update occurs.