Assignment 02 - Igniting the App

1. What is NPM?

- npm is the default **package manage**r for the JavaScript runtime environment **Node.js.**
- Npm is used to install the required dependencies/packages to our project.
- It is an online database registry for open source packages and paid-for private packages. Which consists of over 8,00,000 node packages.
- Open source developers around the world uses the npm to share and borrow packages
- To get started ===> **npm init**

2. What is the difference between “package.json” and “package-lock.json”?

Package.json	Package-lock.json
• It is mandatory for every project.	• It is automatically generated when there is a change in package.json / node_modules
• It contains information such as name, description, author, script, and dependencies.	• It contains the name, dependencies, and **locked the exact version of the project.**
• It records important metadata about the project.	• It will allow future developers to install the same version of the dependency

3. Why should I not modify the ”package-lock.json” file?

- ❖ It is a generated file and is **not designed to be manually edited**.
- ❖ Its purpose is to track the entire tree of dependencies (including dependencies of dependencies → transitive dependencies) and the exact version of each dependency.
- ❖ **You should commit package-lock.json** to your code repository.

4. What is the difference between caret (^) and tilde (~)?

- ❖ When we see the version of any package, it will contain three digits separated by dots. Like **1.2.0**
- ❖ That indicates **Major.Minor.Patch** version changes.

Tilde (~)	Caret (^)
• It will update you to all future **patch versions**, without incrementing the minor version. • ~1.2.3 will use releases from 1.2.3 to <1.3.	• It will update you to all future **minor/patch** versions, without incrementing the major version. • ^2.3.4 will use releases from 2.3.4 to <3.0.0
• It gives you **bug fix** releases.	• It gives you **backwards-compatibility and new functionalities** as well.
• It will update in decimals.	• It will update to its latest version in numbers.

5. What is the difference between dependencies and dev-dependencies?

Dependencies / Prod-Dependencies	Dev-Dependencies
• Production dependencies are **fundamental dependencies** that are required to complete the project. They will be needed during both the development and production phase.	• These are the dependencies needed **only in the development phases**. These are useful for development and testing purposes.
• Ex: react, react-dom	• Ex: babel, jest, bootstrap

6. What is “npx”?

- ❖ Npx is **Node Package eXecute**. Used to execute the npm packages.
- ❖ When we run the command (**npx parcel**) → this will search for the package in the local and global registry, then execute it.
- ❖ If the package is not installed already it will automatically install and execute them. But it will cache them instead of saving.

7. What is “parcel”/”webpack”? Why do we need them?

- ❖ Parcel and webpack are the **bundlers** used mostly for JavaScript or Typescript code that helps you to **minify, clean, and make your code compact.**
- ❖ so that it becomes easier to send a request or receive the response from the server.
- ❖ Both are used to remove the unnecessary comments, new lines, any kind of block delimiters, and white spaces while the functionality of the code remains unchanged.

8. What is “.parcel-cache”?

- This is a cache folder which is **automatically generated** when executing the parcel in our project.
- This folder will contain the **cache data**, so when we re-run/rebuild our app it **will not parse and analyze the data from scratch**. This is the reason parcel is so fast in the development phase.

9. What is”tree shaking”?

- Tree shaking is the process of **removing unused code in production builds**. It is also known as **“Dead code**”.
- As parcel statistically watch the imports and exports of each module and do the tree shaking process.

10. What is Hot Module Replacement (HMR)?

- **HMR** - (**H**ot **M**odule **R**eplacement) improves the development by updating the modules in browsers at runtime **without needing a whole page refresh.**
- The application state will remain the same, when we change small things.
- Parcel’s HMR supports both JavaScript and CSS assets.
- This uses a **file watching algorithm** which is written in **c++**.

11. List the some superpowers/features of Parcel? Describe them.

1. Dev Server
2. Image optimization
3. Minification of files
4. HTTPS

5. Code splitting
6. Differential bundler
7. Error Handling
8. Differential bundling

1. **Dev server:**
    - ❖ Parcel's builtin dev server is automatically started when you run the default parcel command.
    - ❖ which is → **npx parcel index.html**. By default, it starts a server at http://localhost:1234.

2. **Image optimization**
    - ❖ Parcel by default includes **lossless image optimization for JPEG, PNG** images in the production.
    - ❖ This **reduces the size** of the images without affecting their quality.

3. **Minification of files**
    - ❖ In production mode, Parcel automatically **minifies the code to reduce the file sizes** of the bundles. By default, Parcel uses SVGO to perform SVG minification.

4. **HTTPS**
    - ❖ Sometimes we may need to use HTTPS during the development process.
    - ❖ For that we can use → **parcel src/index.html --https**
    - ❖ To learn about more features visit the official website https://parceljs.org/docs/

12. What is the "dist" folder in the parcel?

- ❖ Dist folder will contain the **parcel's minified, bundled, optimized, etc., production ready code.**

13. What is browserslist?

- Browserslist is a tool that allows specifying **which browsers should be supported in your frontend app** by specifying "queries" in a config file. It's used by React, Angular and Vue, but it's not limited to them.

  Why do we want them?

  - During development we are using the **latest javascript features** (e.g ES6) as it makes our jobs easier, leads to cleaner code, possibly better performance.
  - But, not all browsers have built-in support for ES6. By using browserslist, **transpilers/bundlers** know what browsers you want to support, so they can "group" browsers in different categories and generate separate bundles, for example:
    - **Legacy Bundle**: Contains polyfills, larger bundle size, compatible with old browsers without ES6 support.
    - **Modern Bundle**: Smaller bundle size, optimized for modern browsers.

  - Refer https://browserslist.dev/ for more browsers list

  - In package.json,

    “Browserslist”:[“last 2 version”, last 10 chrome version]

14. What are the Script types in HTML?

- <script **type=”module”** src=”./App.js”></script>
- <script **type=”application/javascript”** src=”./App.js”></script>

15. What are node_modules?

- The node_modules are in the root folder of the project which contains **all the React dependency** packages: react, react-dom **and their transitive dependencies** babel, webpack, parcel etc. to build and run a React project.
- This can be regenerated by running the command: **npm install**

16. What is .gitignore? What files do we need to put in the .gitignore?

- gitignore file is a plain text file that contains a list of all the specified files and folders from the project that **Git should ignore and not track.**
- We should keep all the files in .gitignore which **can be regenerated during the development phase.**