

Q1. What is NPM?

'NPM' is a package manager where all the libraries are hosted, and we can use it to download those libraries which we need in our app in the form of dependencies.

Q2. What is 'Parcel/Webpack' ? Why do we need it ?

'Parcel/Webpack' are what we call module bundlers for our Javascript code. A bundler is a tool used to merge your code into a single file.

They are used for optimizing our react application by performing various optimization tasks for us which includes,

- Creating Dev Build
- Creating Local Server
- Hot Module Replacement
- File Watching Algorithm
- Provides Caching - Faster Builds
- Performs Image Optimization
- Minification
- Bundling
- Compressing
- Consistent Hashing
- Code Splitting
- Differential Bundling - to support older browsers
- Diagnostics
- Error Handling
- Ability to host app on HTTPS
- Tree Shaking - remove unused code
- Different dev and production bundles

Q3. What is '.parcel-cache' ?

'.parcel-cache' is a folder which parcel creates while building our app and it stores/caches the information about our app in it. The benefit of it is that when parcel rebuilds our application, it uses these cached data and it doesn't need to re-parse everything. It makes our build creating process fast.

Q4. What is 'npx' ?

While 'npm' is used to install the packages, 'npx' is the command we use to run those packages. We can use npx command to even execute packages which are not even installed locally.

Q5. What is the difference between 'dependencies' and 'devDependencies' ?

'dependencies' are the packages which we need for our app to run everywhere including dev and production environment. 'devDependencies' on the other hand are the dependencies which we need only while developing our app and will not be needed on the production environment.

Q6. What is Tree Shaking ?

Tree Shaking is a type of optimization where the unused code is removed from our build. For react app, module bundlers like parcel do this kind of optimization for us.

Q7. What is Hot Module Replacement ?

Hot Module Replacement (or HMR) is one of the most useful features offered by module bundlers like webpack/parcel. It allows all kinds of modules to be updated at runtime without the need for a full refresh. Parcel does so by using a File Watching Algorithm which is written in C++.

Q8. List down your favorite 5 superpowers of Parcel and describe any 3 of them in your own words.

Parcel provides us with a lot of superpowers which make our application optimized and fast. Below are some of the superpowers of parcel,

1. Minification - Parcel gives us a superpower of Code Minification, which means optimizing code to save space, reduce page load times, and lower website bandwidth usage.

2. Bundling - Parcel gives us the ability to bundle our code files. Bundling is a process of following imported files and merging them into a single file: a "bundle". This bundle can then be included on a webpage to load an entire app at once.

3. Hot Module Replacement - Hot Module Replacement (or HMR) is one of the most useful features offered by module bundlers like webpack/parcel. It allows all kinds of modules to be updated at runtime

without the need for a full refresh. Parcel does so by using a File Watching Algorithm which is written in C++.

4. Caching - Parcel caches our app data at the time of build. It creates a folder named '.parcel-cache' where it stores/caches the information about our app in it. The benefit of it is that when parcel rebuilds our application, it uses these cached data and it doesn't need to re-parse everything. It makes our build creating process fast.

5. Tree Shaking - Parcel also provides us with Tree Shaking feature which means all the unused code in our application is removed from our code automatically.

Q9. What is '.gitignore' ? What should we add and not add in it ?

A .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. Inside .gitignore , you can tell Git to ignore only a single file or a single folder by mentioning the name or pattern of that specific file or folder.

We should add files/folders to gitignore which can be auto-generated by running commands like npm install. For eg. node_modules, dist and .parcel-cache folder.

We should not add files/folders which are necessary for our app and it's configuration and which are required to generate our dependencies and builds. For eg. package.json and package-lock.json.

Q10. Why should I not modify 'package-lock.json' file ?

The reason being that 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) and the exact version of each dependency. We should commit package-lock. json to our code repository.

Q11. What is 'node_modules' ? Is it a good idea to push that on git ?

The node_modules folder contains all the saved JavaScript dependencies of a project. It contains our app's own dependencies as well as all the Transitive dependencies too.

Since this folder often has a massive size, we should not commit it inside the Git repository.

Q12. What is the 'dist' folder ?

The /dist stands for distributable folder. The /dist folder contains the minimized version of the source code. The code present in the /dist folder is actually the code which is used on production web applications.

Q13. 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. The browserslist configuration controls the outputted JavaScript so that the emitted code will be compatible with the browsers specified