Q1. What is NPM?
A1. 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.

https://docs.npmjs.com/about-npm

Q2. What is 'Parcel/Webpack'. Why we do need it?
A2. 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 when it usually takes you to transfer multiple files without using any bundler for loading the page of your application.

Difference b/w Parcel & Webpack?

https://www.educba.com/parcel-vs-webpack/

Parcel optimizes your whole app for production automatically. This includes tree-shaking and minifying your JavaScript, CSS, and HTML, resizing and optimizing images, content hashing, automatic code splitting, and much more.

Q3. What is. parcel-cache?
A3. The. cache folder (or. parcel-cache in parcel) stores information about your project when parcel builds it, so that when it rebuilds, it doesn't have to re-parse and re-analyze everything from scratch. It's a key reason why parcel can be so fast in development mode. I think committing it to git would be a bad idea - it would add a large number of (unnecessary) changes to your commit history, and it could easily get out-of-sync with the code that generated it.

Q4. What is npx?
A4. The npx stands for **Node Package Execute** and it comes with the npm, when you installed npm above 5.2.0 version then automatically npx will installed. It is a npm package runner that can execute any package that you want from the npm registry without even installing that package. The npx is useful during a single time use package.

https://www.geeksforgeeks.org/what-are-the-differences-between-npm-and-npx/

Q5. What is difference b/w 'dependencies' and 'devDependencies'?
A5. Dependency is an object that contains the library, which your project requires for production environments and functioning effectively. devDependencies are those packages in the package.json file that you need only for project development purposes. Example- Babel, Webpack, etc.

https://www.geeksforgeeks.org/difference-between-dependencies-devdependencies-and-peerdependencies/

Q6. What is tree shaking?
A6. Tree shaking is a term commonly used with a JavaScript context to describe the removal of deal code.
It relies on the import and export statements to detect if code modules are exported and imported for use between JavaScript files.

Q7. What is Hot Module Replacement (HMR)?
A7. Hot Module Replacement (HMR) exchanges, adds, or removes modules while an application is running, without a full reload. This can significantly speed up development in a few ways:

- Retain application state which is lost during a full reload.
- Save valuable development time by only updating what's changed.
- Instantly update the browser when modifications are made to CSS/JS in the source code, which is almost comparable to changing styles directly in the browser's dev tools

Q8. List down 5 superpowers of parcel and describe any 3 of them in your own words?
A8.

- Tree Shaking
- Minification
- Image Optimization
- Compression
- HMR

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

A9. The. gitignore file tells Git which files to ignore when committing your project to the GitHub repository. gitignore is located in the root directory of your repo.

- The files contain sensitive data.
- The files are system specific and do not need to exist on every machine's copy.
- Excluding the files maintains system security rules and privileges. (Remember, Git repos only contain the files necessary to get tech support—not to share the entire software.)

Q10. What is difference b/w package.json and package-lock.json?
A10. **package. json** is a metadata file in a Node.js project that describes the project's dependencies, scripts, configuration, and other details.
It typically contains information about the project such as its name, version, author, and license. It also lists the project's dependencies on other Node.js packages, along with their version numbers, so that these dependencies can be automatically installed when the project is set up or updated.

package-lock.json file is like a one-stop solution of your entire problem. package-lock.json is a file that is automatically generated by npm when a package is installed. It records the exact version of every installed dependency, including its sub-dependencies and their versions.

https://www.atatus.com/blog/package-json-vs-package-lock-json/#:~:text=json%20file%20is%20like%20a,sub%2Ddependencies%20and%20their%20versions.

Q11. Why should I not modify 'package-lock.json'?
A11. It stores an exact versioned dependency tree rather than using starred version (*) in package.json file itself, which means the same dependencies will be installed on prod or any other machine. It also has a mechanism to lock the tree but generally will regenerate if package.json changes.

To install the same dependency on another machine, we should not modify the package-lock.json.

Q12. What are node modules? Is it good to push that on git?
A12. The node_modules directory is one of the crucial parts of any node or React project, but it shouldn't be tracked by the version control system (git) due to its large size. The right approach is to track the package. json file, and use the npm tool to regenerate node_modules.

Q13. What is 'dist.' Folder?
A13. the dist folder is the build folder which contains all the files and folders which can be hosted in server. The dist folder contains the transpiled code of your angular application in the format of JavaScript and also the required html and CSS files.

Q14. What is browserlists?

A14. Since different browsers support ECMAScript and CSS differently, developers need to set the correct browser range for web applications.

Browserslist can specify which browsers your web application can run in, it provides a configuration for specifying browsers range. Browserslist has become a standard in the industry, it is used by libraries such as Autoprefixer, Babel, ESLint, PostCSS, SWC and Webpack.

When you specify a browser range through Browserslist, Builder will compile JavaScript and CSS code to the specified syntax, and inject the corresponding polyfill code. **When you only need to be compatible with modern browsers, the compilation process will introduce less compatible code and polyfills, and the performance of the page will be better.**