Q. What is `NPM`?
A: NPM is a package manager that allows us to install, update and manage dependencies in a project

Q. What is `Parcel/Webpack`? Why do we need it?
A: Parcel/Webpack are bundlers that come with multiple capabilities which can be used while creating and setting up our application. These include code-splitting, hot module replacement, image optimization, caching, https configuration, among others.

Q. What is `.parcel-cache`
A: parcel supports caching. It creates a parcel-cache folder when we build our app, which greatly improves performance and load time. It can be regenerated each time we build our app, so it does not need to be pushed to git.

Q.What is `npx`?
A: npx is a package runner that is used to run packages either installed locally or fetched remotely. It simplifies the execution of the packages

Q.What is the difference between `dependencies` vs `devDependencies`?
A: There are two types of dependencies. Normal dependencies and dev-dependencies. Dev dependencies are used for development and testing and are not required for production. They are not included in the final production build. Normal dependencies are used for production.

Q. What is Tree Shaking?
A: TreeShaking is an optimization technique used to remove the unused or dead code. It significantly reduces the bundle size, improving application load time and performance.

Q. What is Hot Module Replacement?
A: When we make any code changes, the changes are immediately applied to the running application without having to reload the page. This is known as hot module replacement.

Q. List down your favorite 5 superpowers of Parcel and describe any 3 of them in your own words?
A: Caching, Hot Module replacement, image optimization, bundling, and tree shaking are the 5 superpowers of Parcel.

Caching- Parcel supports caching. It creates a parcel-cache folder when we build our app, which greatly improves performance and load time. It can be regenerated each time we build our app, so it does not need to be pushed to git.

Hot Module Replacement- When we make any code changes, the changes are immediately applied to the running application without having to reload the page. This is known as hot module replacement.

ImageOptimization: Parcel has the power to optimize images and thereby reduce expensive network operations.

Bundling- It creates a bundle of files that are minified and optimized for production.

TreeShaking- It is an optimization technique used to remove the unused or dead code. It significantly reduces the bundle size, improving application load time and performance.

Q. What is `.gitignore`? What should we add and not add into it?
A .gitignore is a file where we include the list of all files and folders that we should not push to git. Anything that can be regenerated should not be pushed to git and therefore included in .gitignore.

Q. What is the difference between `package.json` and `package-lock.json`?
A: package.json is a configuration for npm. It lists the packages being used in the application along with their versions, which can come with a caret for minor version and a tilde for major version. Package-lock.json contains the exact version of the package being used in the application.

Q. Why should I not modify `package-lock.json`?
A: This is to ensure all developers have the same version of the package. Otherwise, it would lead to the application working in sme machines and breaking in others.

Q. What is `node_modules`? Is it a good idea to push that on git?
A: Node modules are like a database for all the dependencies of an application. It is very large in size and can be regenerated whenever we run the npm install command if package.json is present. So we should not push it to git.

Q. What is the `dist` folder?
A: The dist folder is created every time we run npm parcel filename or npm parcel build filename(for production). It contains the files that are o[optimized and minified for production.

Q. What is `browserlists`
A: Browserslist is a package that allows our code to run in older browsers as well. It can be configured in package.json.