Q: What is `NPM`?
A: npm is a package manager that helps in installing the packages and managing them so that our application can use them for various purposes like bundling, image optimization, HMR, creating build etc. NPM also manages their transitive dependencies.

Q: What is `Parcel/Webpack`? Why do we need it?
A: Parcel and webpack are development tools that are used for module bundling. They bundle many script files and create a single one so that it is production-ready loadable in browser. Apart from bundling they also create builds for both development and production, does image optimization, cleans our code, text compression, caching of the build, tree shaking, makes our code compatible with older browsers. Thus, making our application fast and performant.

Q: What is `.parcel-cache`?
A: This a development build cache generated by Parcel. So that next time we create a new build it takes less time and creates it for only new changes that are added. This makes its build creation faster.

Q: - What is `npx` ?
A: 'npx' is used to execute the command. It can be considered as being equivalent to 'npm run'

Q: - What is difference between `dependencies` vs `devDependencies`?
A: Dependencies are the packages that our application needs to perform certain expected task. Like React is a dependency that our application needs to create Views. 'devDependencies' are also dependencies but only for development environment, they are not needed for production. Like parcel is a devDependency as in production, the hosted platform uses their own bundlers. EsLint and Prettier are also devDependencies that we need for development purpose only to format code and style it for developers readability and to create a clean source code.

Q: What is Tree Shaking?
A: Tree shaking is dead-code elimination in which any unreachable code is removed from the bundle. This is tracked by checking the export and import statements. Any code, that is not being exported and imported is considered unreachable code and hence is removed.

Q: What is Hot Module Replacement?
A: As we make changes to our code, Parcel automatically re-creates a build of changed files and updates the application on browser. By default, Parcel refreshes the page when updating application on browser, in some cases HMR is used in which the browser does not reloads the whole application but just updates the modules at runtime. This ensures that the state of the app is retained.This also provides smooth development experience.

Q: List down your favourite 5 superpowers of Parcel and describe any 3 of them in your own words.
A: 5 Parcel superpowers:

1. HMR
2. Auto-install
3. Minify files
4. Lazy Mode
5. Developement Target

- **Auto-install**: If there are any dependencies, plugin missing from application that needs installation and are required for a package then parcel automatically installs them.

- **Minify files:** Minification is a process of minimizinjg code and script files. It removes unnecessary characters from the code without affecting its functionality. It reduces the load time and bandwidth usage which helps in increasing speed of the site and improving user experience.
- **Lazy mode**: Instead of creating build for all the files changed, lazy mode only creates build at the runtime depending upon where the user is navigating on application. It creates build for only that part of app where user is traversing.
- **Development target:** For the dev server, only a single target can be build at once. By default, parcel builds for modern browsers only. So, the transpilation of code for older browsers is disabled. To test the code for older browsers we can use –target legacy flag. If there is no explicit target defined then we can use –target default and this would transpile the code just as it would in production mode.

Q: What is `.gitignore`? What should we add and not add into it?
A: This is a file that is used by git to check which files it does not need to keep track of and push on github. Any file name that is added to .gitignore is ignored by git for tracking.

Q: What is the difference between `package.json` and `package-lock.json`?
A: **package.json:** Includes information about our project like project name, version, author name, script commands, entry point of project, git repository and the dependencies and devDependencies that are being used by our project. It stores the approximate or exact version of the dependencies that are needed for our application.
Package-lock.json: Its is lock file that is used for production. It includes the exact (locked) version of each package so that during production that exact version is installed.It does not include other information like scripts, repository, author, license but do include project name and version.

Q: What is `node_modules` ? Is it a good idea to push that on git?
A: Node modules contains all the packages that are installed by package manager for our project. It also includes treansitive dependencies of the installed packages. We should not push it on github as it is only for development environment and during production these packages are automatically installed by checking the package-lock.json file. The node_modules are heavy and as they are not really required during project installation or during production therefore they should not be pushed to github.

Q: What is the `dist` folder?
A: dist folder stores the build files for production. It includes- .map and .js folders which include source map and minified js files.

Q: What is `browserlists?
A: browserslist is used for compatibilty to older broswers. The version that we specify in browserslist, the js code will be transpiled in such a way so that it is compatible with those versions of browsers.