Theory

What is Emmet?

Its a plugin used to write html,css in very fast way. most boilerplate html and css codes can be written by emmet in very fast way.

Difference between a Library and Framework?

In library, we as developer are in full charge of the code and use the library whenever we want.
But in framework, framework is the boss and controls everything , we developer just provide implementations to the requirements of framework and it calls them as per its need. we use it to speed up the the development process. and it has its onw architecture and rule that we need to abide by always.

What is CDN? Why do we use it

suppose login page of facebook is a static webpage and its hosted on a server in california. people from acrosss the world is hitting that page .ie they are requesting FB server to get them that page now for people of US its quite fast because since server is there latency is low. but for people of India,Australia its quite slow. so to avoid these create a cluster of caches that has the copy of FB login page/or any other such static pages and make them available in all countries/ all availabilty zones around the world.It will decrease the latency and speed up the application.

Why is React known as React?

React is a javascript library and its main purpose is to render content on the browser. so whenever user changes the state/data of the application by clicking button/submititing form or any other user events, then react should "REACT" to that change and display the change to the user.

What is crossorigin in Script tag?

Its for allowing cross-origin resource sharing such that a resource on a server can be shared with other domain outside of its own. It was introduced in to provide safety from cross attacks.

What is diference between React and ReactDOM?

React is for creating views ,that needs to be rendered to UI reactDOm is to render that view to DOM

What is difference between react.development.js and react.production.js files via CDN?

react.production.js is the optimised version of react.development.js and meant for production

What is async & defer?

parsing of a html script is done from top to bottom when during parsing, script tag is encounterd then control first fetches the script ,executes it and then resumes html parsing when we use defer, script is loaded/downloaded in parallel but executing the script will be done after parsing of html is done . when we use async, then both parsing the html and downloading the script is done simultaneously so it doesnot blocks the parsing of html.when script is fetched then it get executed immediately and then html parsing resumes.

What is NPM?

When we develop a production ready app, we need to manage several packages and their versions, npm just does this.It comes with a CLI that intercats with online databases of NPM.The database is NPM registry

How to intiliaze NPM so that we can use NPM--->

``` npm init ```
It will create the package.json

What is parcel?

In local we write code in several modules and directories but when we need to deploy that app on server.We acnnot just deploy the entite folder structure to server.INow lets suppose in every files in every modules,we are using some dpeendencies and every dependency have some furthur dependencies.Maintaining these huge list of dependecies across files/modules is tough.so what bunder does is that it combines all files of applications into a single file maintaining all the dependency order.It also maintains a dependency graph to keep track of depency orders between files

Thats when parcel shines in

Parcel Features

- HMR (Hot Module Replacement) - parcel keeps track of file changes via file watcher algorithm and renders the changes in the files
- File watcher algorithm - made with C++
- Minification
- --> cleaning up the code removing anu unncessary lines,comments and whitespaceDEV and production Build
- Super fast building algorithm
- Image optimization
- Caching while development
- Compresses
- Compatible with older version of browser
- HTTPS in dev
- Port Number
- Consistent hashing algorithm
- Zero Configuration

installation commands:

- Install:

```
npm install -D parcel
```

`-D` is used for development and as a development dependency.

- Parcel Commands :
  - For development build:

```
npx parcel <entry_point>
```

-

- For production build :

```
npx parcel build <entry_point>
```

-

Q: What is `.parcel-cache`

A: `.parcel-cache` is used by parcel(bundler) to reduce the building time. It 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.

Q: What is `npx`?

A: `npx` is a tool that is used to execute the packages. It comes with the npm, when you installed npm above 5.2.0 version then automatically npx will installed. It is an npm package runner that can execute any package that you want from the npm registry without even installing that package.

Q: What is difference between `dependencies` vs `devDependencies`?

A: When we develop an app in local machine,the dependencies that are only needed in local machine to run our app is we generally place in devDependencies and those dependencies that are needed to run the app in production are placed in Dependencies.

```
npm install --save-dev
```

instead of just,

```
npm install --save
```

Q: What is Tree Shaking?

A: Tree shaking is process of removing the unwanted code that we we are not using.Suppose in our app we have using a library, and that library has 10000 functions but we are using only 100 among 10000, so rest functions are not needed so it needs to be removed since it doesn ot adding any functionality.Tress shaking does that for us.

Q: What is Hot Module Replacement?

A: when we do some changes in code while running a server, parcel detects a change and rebuild the code and deploy that devBuild to server. we dont need to rebuild and restart the server to see changes.

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

A: 5 superpowers of Parcel:

- HMR (Hot Module Replacement) - adds, or removes modules while an application is running, without a full reload.
- File watcher algorithm - File Watchers monitor directories on the file system and perform specific actions when desired files appear.
- Minification - Minification is the process of removing unwanted and unused code like console log, comments
- Image optimization
- Caching while development

Q: What is `.gitignore`? What should we add and not add into it?

A: The .gitignore file is a text file that tells Git which files or folders to ignore in a project during commit to the repository. The types of files you should consider adding to a .gitignore file are any files that do not need to get committed. for example, For security, the security key files and API keys should get added to the gitignore. package-lock.json should not add into your .gitignore file.

The entries in this file can also follow a matching pattern.

```
* is used as a wildcard match
/ is used to ignore pathnames relative to the .gitignore file
# is used to add comments to a .gitignore file
```

This is an example of what the .gitignore file could look like:

```
# Ignore Mac system files
.DS_store

# Ignore node_modules folder
node_modules

# Ignore all text files
*.txt

# Ignore files related to API keys
.env

# Ignore SASS config files
.sass-cache
```

Q: What is the difference between `package.json` and `package-lock.json`

A: `package.json`:

It basically consist of the properties to identify the module/project such as the name of the project, current version of the module, license, author of the project, description about the project,module such as the entry/starting point of the module, dependencies in project, scripts being used, repository links of Node project etc.

In simple words,Its is a configuration file that npm uses to manage dependencies.

`package-lock.json`:

- This file is automatically generated for those operations where npm modifies either the node_module tree or package-json.
- It is generated after an npm install
- It allows future devs & automated systems to download the same dependencies as the project.
- it also allows to go back to the past version of the dependencies without actual ‘committing the node_modules folder.
- It records the same version of the installed packages which allows to reinstall them. Future installs wll be capable of building identical description tree.
- It stores an exact, versioned dependency tree rather than using starred versioning like package.json itself (e.g. 1.0.*). This means you can guarantee the dependencies for other developers or prod releases, etc. It also has a

mechanism to lock the tree but generally will regenerate if package.json changes.

~ or ^ in `package.json` file : These are used with the versions of the package installed.

For example in `package.json` file:

```
"dependencies": {
    "react": "^18.2.0",
    "react-dom": "^18.2.0"
  }
```

- ~ : “Approximately equivalent to version”, will update you to all future patch versions, without incrementing the minor version.
- ^ : “Compatible with version”, will update you to all future minor/patch versions, without incrementing the major version.

If none of them is present , that means only the version specified in `package.json` file is used in the development.

Q: Why should I not modify `package-lock.json`?

A: `package-lock.json` file contains the information about the dependencies and their exact versions used in the project. Deleting it would cause dependencies issues in the production environment because versions might upgrade in future and there will be discrepency in version of local and to server,this may break. So don't modify it, It's being handled automatically by NPM.

Q: What is `node_modules` ? Is it a good idea to push that on git?

A: `node_modules` folder like a cache for the external modules that your project depends upon. When you npm install them, they are downloaded from the web and copied into the node_modules folder and Nodejs is trained to look for them there when you import them (without a specific path). Don't push `node_modules`in github because it contains lots of files(more than 100 MB), it will cost you memory space.

Q: What is the `dist` folder?

A: 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. Along with the minified code, the /dist folder also comprises of all the compiled modules that may or may not be used with other systems.

Q: What is `browserslist`?

A: Browserslist is a tool that makes my app compatible with older versions of browsers ,I just need to specify the version upto which i need to make it compatible.