## Ep 2 - Igniting Our App (i.e, Fire your App, Energy to App)

Till now you are writing your code in .html file or .css file or .js file but in Production, these code is no ready for production.
Issue is your code is not optimized

- To make your app ready to go into production, you need to do lot of processing before pushing your code to production.
For this you need to do (bundling, Code spliting, junking, image optimization, remove Comments, minify the Code, progressive etc)

- npm does **not** stand for **node Package Manager**
- npm does not have Full form. npm is everything but not node package manager
- npm is a Package Manager. (library / utilis / packages / dependency)
  - Biggest standard Repository for Package Manager
  - All the packages are hoisted over there.
  - You can use npm package in Your Code.

package.json is a configuration for npm. (package manager)

why do we need package.json?

- npm is a package manager, which helps to use any package in our code.
- npm will take care of • version of package. • name of package • description of package etc } all these Information regarding a particular package is stored in package.json file

- Package.json file also stores information of Multiple Packages.

These Most Important Package in our Code is **Bundler**

- **Bundler** is used to Bundle whole code together

performs → i.e, Minified, Compressed, cleaned before pushing to production

- Dev build
- local server
- HMR (Hot Module Replacement)

→ uses file Watching Algo - written in c++ / parcel-cache

Parcel, Vite, Webpack ⟹ Bundler

↳ Job is Bundle your App, so that it can shift to production

easy to configure

all the bundler's are same

Create-react-app → uses webpack bundler behind the Scenes

Parcel is a beast

npm install parcel → Command to Install Parcel pkg in our Project
<pkg-name>

2 Types of Dependency we can install

→ dev dependencies

- * Required only in development phase of Project.
- * -D indicates you are installing your pkg in only development phase
- * Parcel is dev dependency

↓ normal dependencies

- Used in Production phase also

(^) Caret & tilde (~)

(^) Caret will update every minor version of Package (prefer to use)

(~) tilde will update only major version of package

eg:- "parcel": "^2.8.3" — caret

eg:- "parcel": "~2.8.3" — tilde

If a new version is release, for a pkg even a minor version (^) caret will update to that Version.

If a new version is released for a particular pkg, Only if it's major update, then only (~) tilde will update that Version

<u>package-lock.json</u>: This file package-lock.json locks the Version and keeps the record of it.
- keep record of every exact Version of that pkg/ dependency which is install

<u>node modules</u> :- Contains the actual Code that npm has fetched from npm package and stores all the Code file of package in a node modules
- like a database, which contains actual data of that dependency/pky that are needed for project.
- It is huge, contain's large no. of files

## Transitive Dependency

- Our project contains various packages that are required.
- As our project need Some dependency.
  Similarly that pkg or dependency have their own itself dependency or packages. Hence it is Transitive Dependency

Every package has it's own package.json
↳ and every package.json contains its own dependencies & devdependencies

•<u>gitignore</u>: gitignore is a file which help you when you don't want to push Some files for production.
- Don't want to put some file on github.
- Just write the file name / folder name in gitignore file and those file will be ignored.
  Eg:- /node-module
  .env

It is **Important** to put package.json & package-lock.json to github.
IF you package.json or package-lock.json you can recreate all your node-modules By npm install in your repository
That is why it is not required to push node-modules for production or for git or github

npx parcel index.html → Igniting our App using parcel
- It will host App on localhost Server
- Start our Server

npx ↓ executing a pkg

npm ↓ install a pkg

npm install React
npm install react-dom
} Installing React & React-dom package in our project.

Inorder to use react in your code, you need to first Import it using
import React from "react"
import ReactDom from "react-dom" → It refers to the pkg your installed and Store in nodemodules
↳ It comes from node_modules

## Parcel Features

- Dev Build
- Local Server
- HMR ( Hot Module Replacement)
- File Watching Algo. written in C++
- Caching - Faster Builds
- Image Optimization
- Minification
- Bundling
- Compress
- Consistent Hashing
- Code Splitting
- Differential Bundling - Support older browser
- Diagnostic
- Good Error Handling
- HTTPs
- Tree Shaking - remove unused Code
- Different dev & production bundles

npx parcel **build** index.html

↳ builds production code

parcel-cache folder & dist folder do not push on github. because these are temporary files which can be Regenerate

- Hence include in .gitignore file

**Browserlist** is a npm Package which helps to make our App compatible in older Version of Browsers

- you have to tell your project that what all browser's your app should support
- you have to configure it in package.json
- Array of browser list

```
"browserlist": ["last 2 Chrome Version",
                "last 2 FireFox Version"]
```

↓

means it will definitely work in these Version of browser it may work in other browser also, but in these version of browser it will definitely work

https://browserlist.dev/ ← (visit)

When your app is a government Website, then it should work on 99% of browser. Older browser Versions also.