



Episode 2 - Ignite Our App

1) Create a Repo on GitHub

2) In project add these commands

- git init
- git add README.md
- git commit -m main
- git remote add origin
- git push -u origin main

Can only React make our fast?

React can only make an app fast to an extent

NPM → It does not have a full form. It stands for anything but no node package manager.

It manages packages. All the packages and utilities that we need in a project come from npm.

How to include npm in project?

In terminal

- npm init
- package name: (namaste-react)
- version: (1.0.0)
- description: This is Namaste React
- entry point: (App.js)
- test command: jest
- git repository:
- keywords: react, namaste react, js
- author: Khushi Johri
- license: (ISC)

package.json has been added to the project

package.json file is a configuration for our npm

packages are also known as dependencies

npm will take care of the version of the package

What important package/dependencies do we need to install?

Bundler → Our whole code needs to be bundled together, it needs to be minified, cached, compressed, cleaned before we send it to production

Webpack, parcel, vite are some examples of bundler

It packages your app so that it can be shifted to production

How to install Parcel?

Run the following command:

```
npm install -D parcel
```

dev dependencies → It is generally required during development

normal dependencies → It is used in production also.

What is package-lock.json?

It keeps the track of exact version of the dependency

Package.json keeps approx version & package-lock.js keeps exact version.

What is node-modules?

All the code we fetch from npm. It's like a database for our dependency.

Dependencies required for a dependency is called Transitive dependencies. Every dependency will have package.json with its own dev & normal dependencies

Should I put all these node modules to git, production?

No, if I have package.json & package-lock.json
It can regenerate our node modules

What is .gitignore?

When we don't want anything to not go on github or production, we use .gitignore

Go to .gitignore
and type /node-modules

Should I put package-lock.json & package.json on git?

Yes, maintains the note of what all dependencies our project needs

Ignite the app

npx parcel index.html

parcel has created a server

npm → command of npm

npx → executing a package

Another way of getting react in app

① npm

② CDN links → not preferred way of using react in a project

→ We can already have React in our node module we don't have to make a network called externally through CDN

→ Using CDN will not update it to the newest version automatically

In terminal,

npm install react / npm i react

npm install react-dom / npm i react-dom

How to use react in file after modules installation?

import React from "react"

import ReactDOM from "react-dom"

Error:

Browser Scripts can't have imports and Exports

```
<script src = ". /App.js"> </script>
```

↓
Browser treats it like normal Js file

Normal Js files can't have imports & exports

We have to tell the browser that its a module

```
<script type = "module" src = ". /App.js">  
</script>
```

Now react is coming from module not from CDN links

import ReactDOM from "react-dom/client"
To remove warning

What does Parcel do?

- Dev Build
- Local Server
- HMR → Hot Module Replacement
- File Watching Algo → written in C++
To keep a track of every change
- Caching → Faster Builds present in .parcel-cache
- Image Optimization
- Minification
- Bundling
- Compress files
- Consistent Hashing
- Code Splitting
- Differential Bundling → To support older & newer browsers along with devices
- Diagnostic
- Error Handling
- HTTPs
- Tree shaking Algo → Remove unused code
- Different dev and prod. bundlers

How to create production build?

In terminal,
npx parcel build index.html

// Error

Because here we are giving entry point as index.html but in package.json entry point is ./App.js

This conflict will lead to error

→ So "main": "App.js" needs to be removed from package.js

After creating production ready code, it will put it in dist folder in form of development build

Don't need to put dist & .parcel-cache on github as they can be created again



How to make a project compatible for older versions of browsers?

By using Browsers list in node-modules

Add this in package

Eg

```
"browserlist": {  
  "last 2 Chrome version",  
  "last 2 Firefox version"  
}
```

You can search it on browserlist website

```
"browserlist": [  
  "last 2 versions"  
]
```

