Part 1:

Upload code to Git Hub.

Go to the current project

Namaste-react:> git init

> git branch -M main // -M means adding Default branch

> notepad README.md

> git add .

> git commit -m “Episode-01”

> git remote add origin [git@github.com](mailto:git@github.com):username/reponame.git // this we have to do once

> git push origin main

What is NPM?

NPM **DOES NOT** STAND FOR NODE PACKAGE MANAGER.

NPM means nothing

NPM just Manages the packages.

It's a standard repository to manage all the packages.

How to use NPM in the system.

>npm init

> package name:

>version>

> description

> Entry point : App.js

> test command: jest

> git repo: URL

> Keywords:

> Author :

>it is okay?

It created a Package.json file and contains JSON format.

It's a configuration of npm.

Most important package in the project?

Bundler: it packages your page properly so it can send to production.

Which bundler? Parcel

>npm install -D parcel

//What is -D?

Two types of dependencies/packages

1) Dev dependencies: only for dev (-d)

2) normal dependencies. For both dev/prod

When installing any package there is a line like

"parcel": "^2.12.0"

Q. Here what is ^ and also what is ~ (carat and tilt)?

^ to update minor versions 2.1.0 to 2.2.0

~ for major updates 2.1 to 3.0

What is package-lock.json?

Package.json: its configuration for NPM, it keeps track of what version of the package is installed in the system.

^ Also upgrade a minor version

Package-lock.json: keep track of the exact version.

In package lock, JSON file goes to parcel

There is a key called integrity that contains hash values

Its work is to keep track of the exact version we are using in dev and prod.

To avoid misconfiguration on the server.

What are node modules?

It contains all the code that we fetch from NPM.

We installed parcel but we got lots of directories in node modules why?

We have a dependency on the parcel so we installed the parcel but the parcel may have dependencies on other modules and those modules may have dependencies and so on.

So node modules contain all the packages with direct or indirect dependence on the parcel.

**This is known as transitive dependencies**.

How does npm know parcels have so many dependencies?

When we created the npm init. It made the package.json file to list out the dependencies.

When you install any package it will have its package.json file which contains its dependencies.

Should we add these all node module files to Git Hub?

BIG no

Create .gitignore file

In this file we will mention all the files that we cont want to add on git repo.

And simply mention the name of file or directory

>notepad .gitignore

/node\_modules

Would we add package.json and package-lock.json in the git ignore file?

No

We have to add the package.json and package-lock.json files in the repo. Because it contains records of all the packages we are using.

So with the help of the package.json and package-lock.json files, we can regenerate the code.

For example, just delete the node\_modules directory

Go to the terminal and type

> npm install

It will check your package.json file and regenerate the exact packages that we are going to use in the system.

—

Part 2:

Let's ignite the project

> npx parcel index.html

>

It creates the server example localhost:1234

So before that we are opening the file directly from the directory

Now we can open our files on the server

We use npm to install the packages

We will use npx to execute the packages.

CDN is not a good way to use in a project

We have to call to that external link to get the react.development.js file code.

Also, that file version is static which we are passing.

We can install React in our node moduiles

> npm install react

> npm install react-dom

Its added in package.json and package lock file

"dependencies": {

"react": "^18.3.1",

"react-dom": "^18.3.1"

}

After this, we are going to remove cdn react and react-dom links from the index.html file.

Now showing Error

React is not defined.

Now we know we have install the react in node modules

Let's call them command.

import React from 'react';

import ReactDOM from 'react-dom';

Now we are again getting an error

C:\git\Namaste-React\App.js:1:1

> 1 | import React from 'react';

> | ^^^^^^^^^^^^^^^^^^^^^^^^^^

2 | import ReactDOM from 'react-dom';

3 | // const heading = React.createElement('h1', {"id":"heading"}, "Hello worl

C:\git\Namaste-React\index.html:12:5

11 |

> 12 |

> | ^ The environment was originally created here

13 | <script src="./App.js"></script>

14 | </body>

💡 Add the type="module" attribute to the <script> tag.

Here we are importing App.js file in HTML

Its treating that file as a normal js file.

So it can not contain imports

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

So we need to specify that this is not normal js its module

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

Now it will allow to use keyword import.

Now we can say react is not coming from CDN links its coming from our own node module.

Now we are getting one error due to the new version.

App.js:28 Warning: You are importing createRoot from "react-dom" which is not supported. You should instead import it from "react-dom/client".

Simply replace

import ReactDOM from 'react-dom';

With the given suggestion in error

import ReactDOM from 'react-dom/client';

Now whatever changes we make in code, the browser will do automatically.

This is a feature of the parcel.

Parcel (parceljs.org)

-> Create dev build

> Creating a local server for us

> HMR (Hot module replacement) (change in code will reflect the browser)

How HMR work?

Its using File watching Algorithm (written in C++).

> Faster build (with help of caching)

> Image optimization. (most expensive process)

> Minification (minify the code\_

> Bundling (it create the bundles )

> Compress the file.

> consistent hashing

> Code splitting

> differential building (When app is hosted , so it can run on older browsers, different type of apps)

> Diagnostics

> Error handling

> http / https hosting.

> Tree shaking (remove unused code )

> Lazy mode

> different builds for dev and prod.

How to make prod build

> npx parcel build index.html

> initially you may get error because at time of npm init

We have given landing page as app.js while while creating build we are giving index.js

So go to package.json and remove app.js line

"main": "App.js",

Now

> npx parcel build index.html

It will create a directory name dist with some files. And put all files in it.

Created some files

dist\index.html 348 B 720ms

dist\index.8d566482.css 84 B 54ms

dist\index.15bca2c7.js 138.99 KB 580ms

Now we can deploy these files on server.

Also add .parcel-cache on git repo. Its can be generated from npx parcel index.html

Gitignore

/node\_module (on server it will be created automatically)

/dist (on a server it will be created automatically)

.parcel-cache (its cache files)

| Local | Git | Server |
| --- | --- | --- |
| All files codes  With gitignore files | Only files and directories not mentioned in the git ignore files | Code from git  Run comment > npm install  It will create all dependencies based on the file package and package-lock |

We discussed parcel is having differential building

It means it can run on different browsers

So have list of browsers in mobile browserlist

[npmjs.com/package/browserlist](http://npmjs.com/package/browserlist)

Or browserlist.dev

How to configure?

Open package.json

Add

"browserslist":[

"last 2 chrome version",

"last 2 Firefox version"

]

This means it will work in the last 2 versions of chrome and firfox

It may or may not work for others.

How to got what to write (visit: browserlist.dev)

"browserslist":[

"last 2 versions"

]

It will work on all browsers of last 2 versions.