

EPISODE 2 IGNITING OUR APP

PART 01:

Added episode 1 code to git

Link : [Git Repo Link](#)

NPM:

- NPM manages package.
- NPM is a standard repository of all packages.

Steps to configure npm:

- init npm (package.json) file is created

```
shanmukha@shanmukha-Latitude-3420:~/Music/the_react$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help init' for definitive documentation on these fields
and exactly what they do.

Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (the_react)
version: (1.0.0)
description: This is Namasthe React
entry point: (App.js) index.html
test command: jest
git repository: (https://github.com/Shanmukha136/the_react.git)
keywords: vani react
author: vani
license: (ISC)
About to write to /home/shanmukha/Music/the_react/package.json:

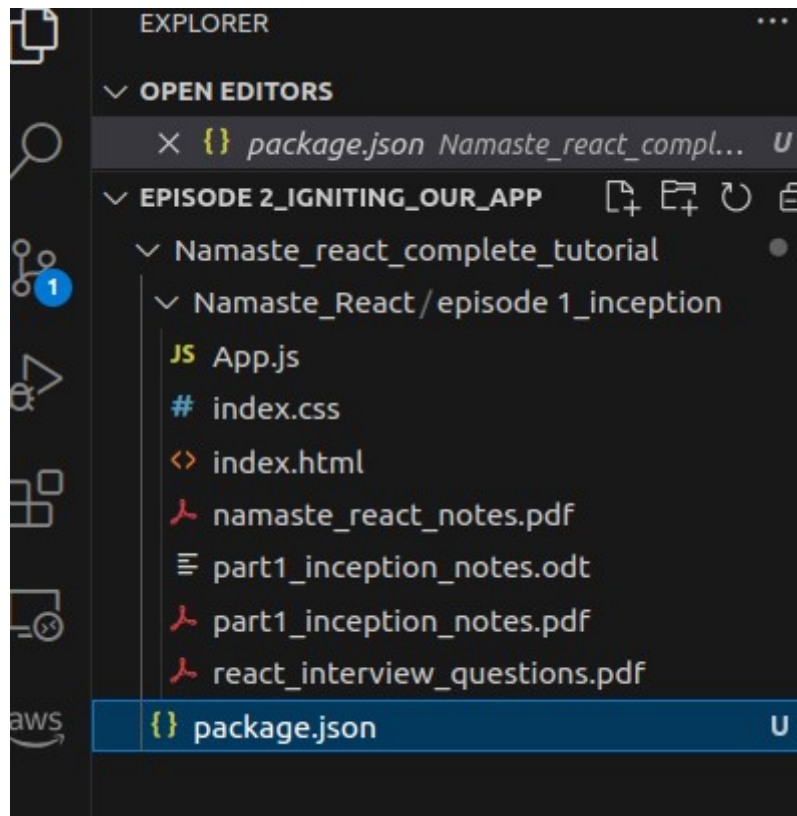
{
  "name": "the_react",
  "version": "1.0.0",
  "description": "This is Namasthe React",
  "main": "index.html",
  "scripts": {
    "test": "jest"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/Shanmukha136/the_react.git"
  },
  "keywords": [
    "vani",
    "react"
  ],
  "author": "vani",
  "license": "ISC",
}
```

```
Press ^C at any time to quit.
package name: (the_react)
version: (1.0.0)
description: This is Namasthe React
entry point: (App.js) index.html
test command: jest
git repository: (https://github.com/Shanmukha136/the_react.git)
keywords: vani react
author: vani
license: (ISC)
About to write to /home/shanmukha/Music/the_react/package.json:

{
  "name": "the_react",
  "version": "1.0.0",
  "description": "This is Namasthe React",
  "main": "index.html",
  "scripts": {
    "test": "jest"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/Shanmukha136/the_react.git"
  },
  "keywords": [
    "vani",
    "react"
  ],
  "author": "vani",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/Shanmukha136/the_react/issues"
  },
  "homepage": "https://github.com/Shanmukha136/the_react#readme"
}

Is this OK? (yes) yes
shamukha@shanmukha-Latitude-3420:~/Music/the_react$
```

Now you can see a new file `package.json` is created.



- `Package.json` file is configuration for our npm.
- Next we will install bundler ,the job of bundler is to bundle or packages our app.
Ex: Parcel , vite etc .,
- we are using parcel here in our code.
- We use same command to install all dependencies.

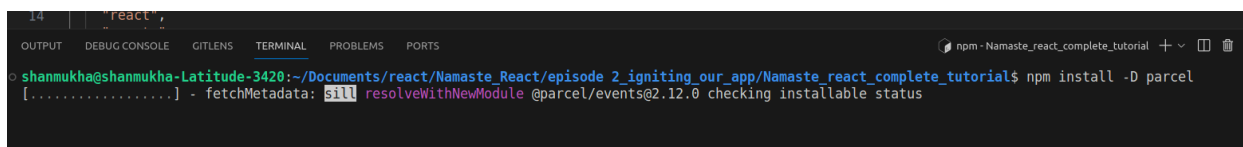
INSTALLING PARCEL



- `[npm install -D parcel]` -> package-lock.json is created

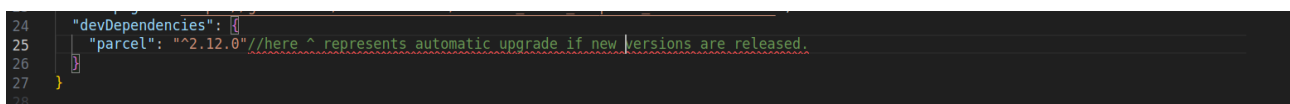
Why we are giving the parameter -D here?

- Basically there are two types of dependencies they are Dev dependencies and normal dependencies.
- Dev dependencies are required for ***Development Phase*** purpose and normal dependencies can also be used for
- ***production phase*** .
- So we are asking parcel here like we want parcel for dev dependency.



```
14 | react |
    |-----|
    OUTPUT  DEBUG CONSOLE  GITLENS  TERMINAL  PROBLEMS  PORTS
shanmukha@shanmukha-Latitude-3420:~/Documents/react/Namaste_React/episode 2_igniting_our_app/Namaste_react_complete_tutorial$ npm install -D parcel
[.....] - fetchMetadata: sill resolveWithNewModule @parcel/events@2.12.0 checking installable status
```

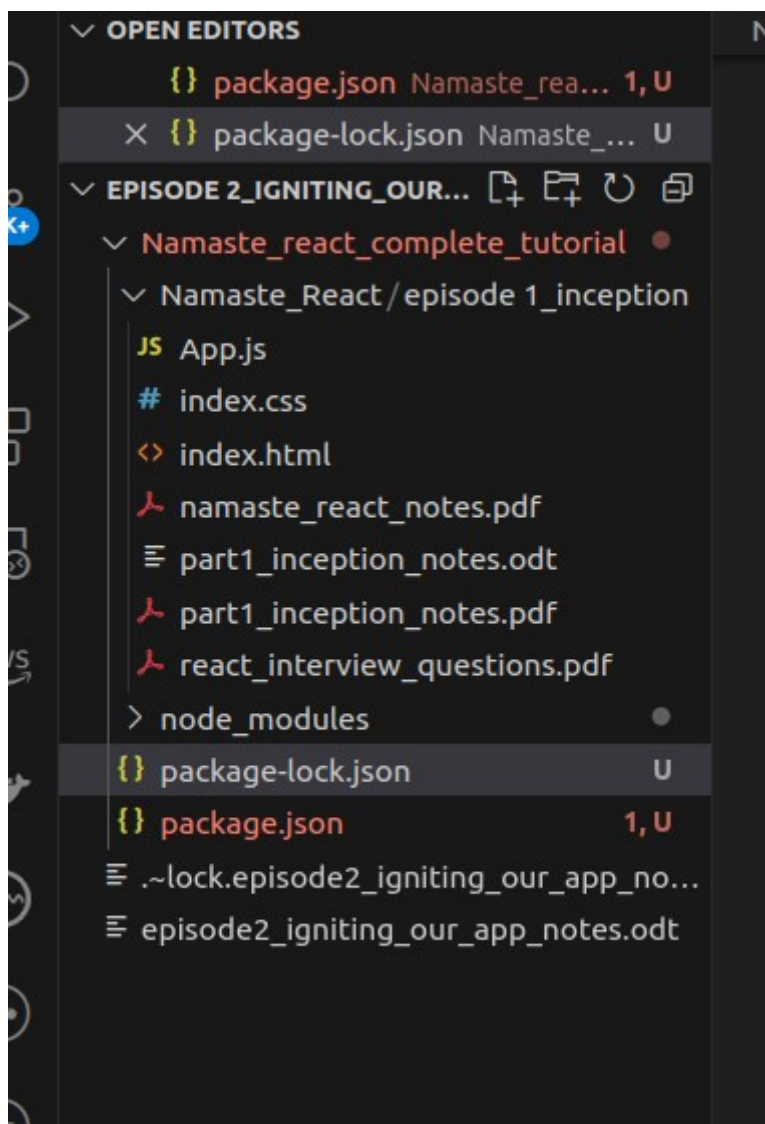
if u see package.json u will see something like this



```
24 | "devDependencies": {
25 |   "parcel": "^2.12.0" //here ^ represents automatic upgrade if new versions are released.
26 | }
27 |
28 |
```

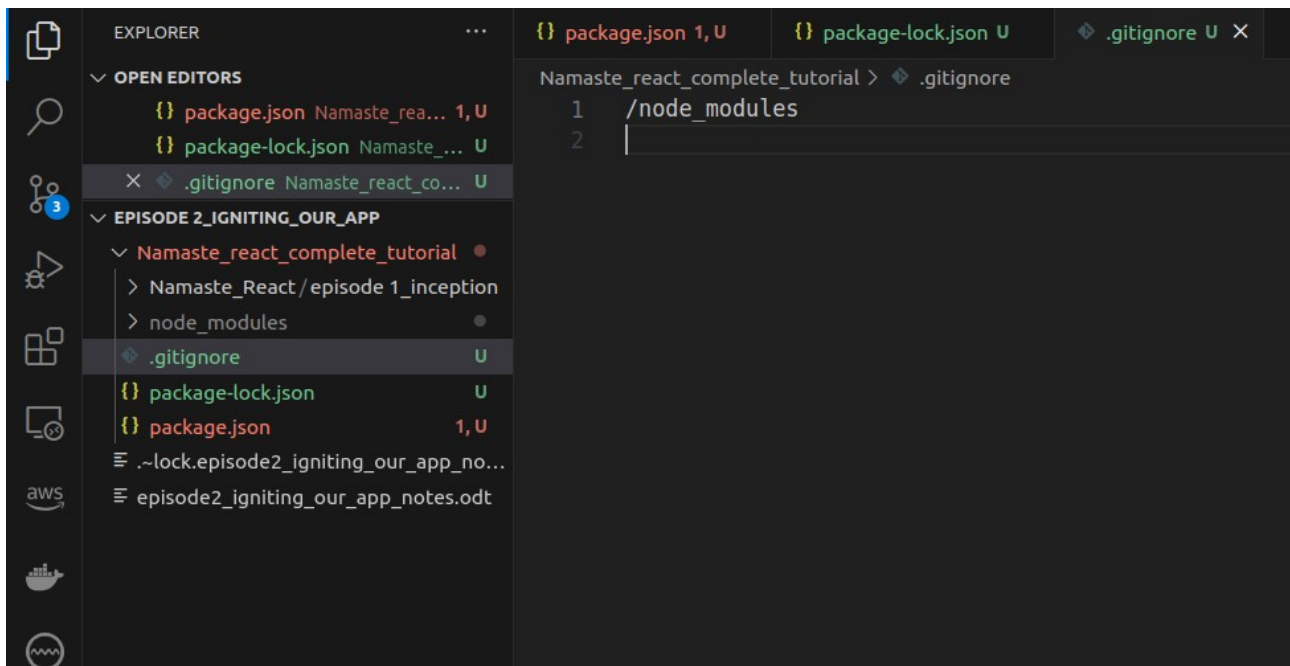
- Also a new file named package-lock.json is created.(it keeps the track of exact version)
- If you search parcel you will find the exact version of parcel inside the package -lock.json file.
- There will be no ^symbol

- Package.json keeps the aprox version in it where as package-lock.json holds exact version
- You can also see a new folder **node modules** is also created.
- It has multiple folders , its because we have installed parcel so those folders are dependencies
- and dependencies dependencies so we have that many folders.



- You can also see a new folder node modules is also created.
- It has multiple folders , its because we have installed parcel so those folders are dependencies
- and dependencies dependencies so we have that many folders
- Dependencies of dependencies is called as transitive dependencies
- In a project there is not only a single package.json and package-lock.json file .In npm folder
- every dependency has its own package.json and package-lock.json files.
- Node modules are collection of dependencies
- There is no need to keep all these files in git , by keeping these files in .gitignore file we can
- achieve this.
- Now if you check git symbol in vs code you will see lot many changes(5k+),so do the following

- First create a file called .gitignore and now in that file write the following
- /node_modules (then save the file and check at the git symbol you will see only 4-5 changes it is clean and simple.
- It is ok not to keep node modules in git but you should keep package.json and package-lock.json
- files in git it is very important.
- It is because you can recreate your node modules with package.json and package-lock.json if node modules are deleted
- (TRY ONCE IF YOU DELETE NODE MODULES , JUST GIVE COMMAND **npm install**)
- You can recreate the node modules.so there is no need to required to push node modules into git
- Gitignore file contains the modules that are not required to push into the git.



EPISODE 2

PART 02:

HOSTING APP INTO THE SERVER

- Type the following command
- (**npx parcel index.html**)-->index.html is source file
- Then you see it gives you a local server where you can run your app
- Npm is used to install a package
- Npx is used to execute a package

Note :
if you get below error:

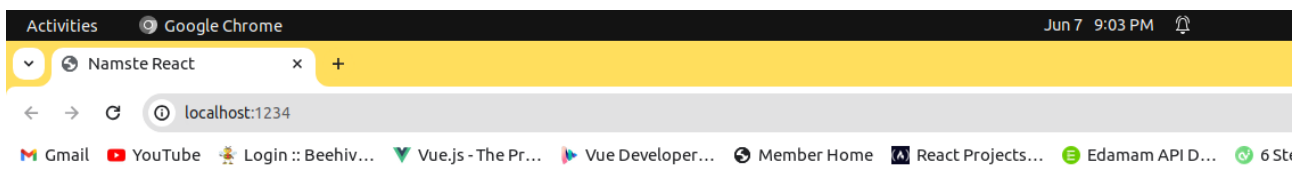
```
shanmukha@shanmukha-Latitude-3420:~/Music/the_react$ npx parcel index.html Server running  
at http://localhost:1234    Build failed. @parcel/core: Unexpected output file type .html in target
```

"main" /home/shanmukha/Music/the_react/package.json:5:11 4 | "description": "This is Namasthe React", > 5 | "main": "index.html", > | ^^^^^^^^^^^^^ File extension must be .js, .mjs, or .cjs 6 | "scripts": { 7 | "test": "jest" The "main" field is meant for libraries. If you meant to output a .html file, either remove the "main" field or choose a different target name. Learn more: <https://parceljs.org/features/targets/#library-targets>

make sure to remove main from the package.json

```
index.html 2  {} package.json U X  .gitignore U  JS App.js
package.json > ...
1  {
2    "name": "the_react",
3    "version": "1.0.0",
4    "description": "This is Namasthe React",
5    "scripts": {
6      "test": "jest"
7    },
8    "repository": {
9      "type": "git",
10     "url": "git+https://github.com/Shanmukha136/the_react.git"
11   },
12   "keywords": [
13     "vani",
14     "react"
15   ],
16   "author": "vani",
17   "license": "ISC",
18   "bugs": {
19     "url": "https://github.com/Shanmukha136/the_react/issues"
20   },
21   "homepage": "https://github.com/Shanmukha136/the_react#readme",
22   "devDependencies": {
23     "parcel": "^2.12.0"
24   }
25 }
```

OUTPUT:



Hello world from React App.js

BRINGING REACT USING NPM RATHER THAN USING CDN LINKS

- This cdn links have only particular version if version changes we need to keep on changing the number

```
<title>Namste React</title>
</head>
<body>
  <div id="root"></div>
  <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
  <script src="./App.js"></script>
</body>
```

- We have version 18 in this case.so we can also bring react using npm.
- **npm install react** -->It brings react into our node module.
- Then install react-dom **npm install react-dom**

```

• shanmukha@shanmukha-Latitude-3420:~/Music/the_react$ npm install react
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @lmdb/lmdb-darwin-x64@2.8.5 (node_modules/@lmdb/lmdb-
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for @lmdb/lmdb-darwin-x64@2.8.5: w
os:"linux", "arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @lmdb/lmdb-darwin-arm64@2.8.5 (node_modules/@lmdb/lmd
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for @lmdb/lmdb-darwin-arm64@2.8.5:
: {"os":"linux", "arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @lmdb/lmdb-win32-x64@2.8.5 (node_modules/@lmdb/lmdb-w

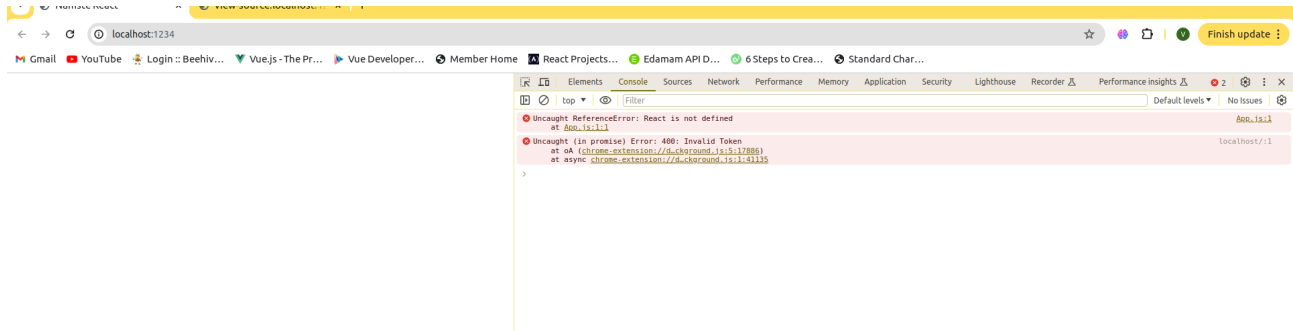
```

```

• shanmukha@shanmukha-Latitude-3420:~/Music/the_react$ npm install react-dom
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @lmdb/lmdb-darwin-arm64@2.8.5 (node_modules/@lmdb/lmd
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for @lmdb/lmdb-darwin-arm64@2.8.5:
: {"os":"linux", "arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @lmdb/lmdb-darwin-x64@2.8.5 (node_modules/@lmdb/lmdb-
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for @lmdb/lmdb-darwin-x64@2.8.5: w
os:"linux", "arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @lmdb/lmdb-linux-arm@2.8.5 (node_modules/@lmdb/lmdb-l

```

- Now you can see react and react-dom folders are created in the node modules.
- Here I am not using the -D param while installing the react and react-dom because here I want normal dependencies only not dev dependencies.
- Remove cdn links
- Give command npx parcel index.html
- Browse the app with in localhost server
- Now everything executes fine , the elements in html file appears in browser but the elements created using React.createElement in app.js can't displayed because in console you see error react is not defined because we have removed cdn links and we are using React.createElement in app.js file which browser could not understand so we import react and react-dom inside app.js.



- Import React from “react”
- Import ReactDOM from “react-dom” in app.js
- You will see the error in browser that browser scripts does not contain a import and export statements Because browser assumes app.js as a normal js file
- So while importing app.js to index.html give type=”module”
- Ex: `<script type="module" src="./App.js"></script>`
- Everything works fine in browser now but you will see a warning like createRoot is not a method
- Now change import ReactDOM from “react-dom” to import ReactDOM from “react-dom/client”
- In react if you made any changes in the file and saved they will automatically reflected in browser there is no need to refresh browser its all because of the parcel.

JS App.js > ...

You, 1 second ago | 2 authors (Shanmukhavani and others)

```
1 | import React from "react";
2 | import ReactDOM from "react-dom";      You, 1 second ago • Uncommitted changes
3 | heading = React.createElement('h1',{}, "Hello world from React App.js");
4 | const root = ReactDOM.createRoot(document.getElementById('root'));
5 | root.render(heading)
6 |
```

OUTPUT

DEBUG CONSOLE

GIT LENS

TERMINAL

PROBLEMS 2

PORTS

○ shanmukha@shanmukha-Latitude-3420:~/Music/the_react\$ npx parcel index.html

Server running at http://localhost:1234

🔥 Build failed.

@parcel/transformer-js: Browser scripts cannot have imports or exports.

/home/shanmukha/Music/the_react/App.js:1:1

```
> 1 | import React from "react";
> | ~~~~~
2 | import ReactDOM from "react-dom";
3 | heading = React.createElement('h1',{}, "Hello world from React App.js");
```

/home/shanmukha/Music/the_react/index.html:47:5

```
46 | <div id="root"></div>
> 47 | <script src="./App.js"></script>
> | ~~~~~ The environment was originally created here
48 | </body>
49 | </html>
```

```
42 | <link rel="stylesheet" href="./index.css">
43 | <title>Namste React</title>
44 | </head>
45 | <body>
46 | <div id="root"></div>
47 | <script type="module" src="./App.js"></script>
48 | </body>
49 | </html>
```

You, 1 second ago • Uncommitted change

OUTPUT

DEBUG CONSOLE

GIT LENS

TERMINAL

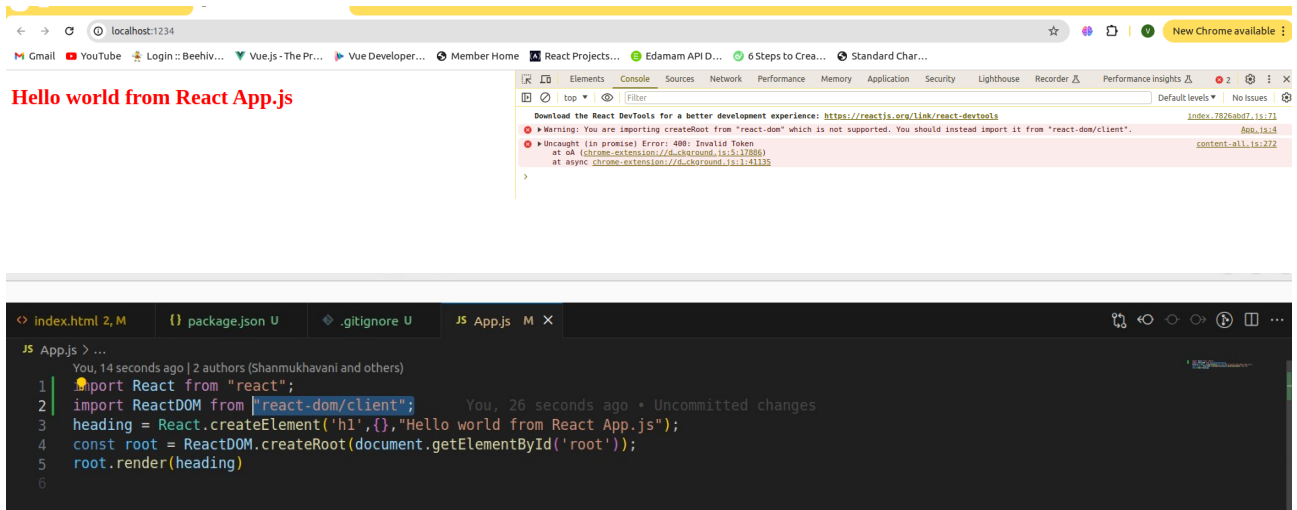
PROBLEMS 2

PORTS

Server running at http://localhost:1234

🌟 Built in 368ms

□



What parcel do?

- Dev builder
- Local server
- HMR –hot module replacement(auto change of browser when we save code changes)
- HMR is done because of File Watching Algorithm in parcel it is written in c++.
- If you do any changes in code and save , it builds app again but the time will be reduced everytime because of Caching
- Faster Builds By –caching(.parcel-cache file holds the cache)
- Image optimization
- Minification
- Bundling
- Compressing
- Consistent hashing

- Code splitting
- Differential bundling-for supporting older browsers
- Diagnostic-beautiful error showing by exactly showing the location where error occurs
- Error handling
- Tree shaking algorithm – removes unused code in the file
- Can load as https
- Builds different dev and production bundles(because production build will take a little more time than the dev bundle)

CREATING A PRODUCTION BUILD

- run the below command.
- **npx parcel build index.html**

```

✦ Built in 399ms
• shanmukha@shanmukha-Latitude-3420:~/Music/the_react$ npx parcel build index.html
✦ Built in 766ms

dist/index.html          347 B    237ms
dist/index.459ed095.css   81 B     29ms
dist/index.12e56685.js  138.77 KB 261ms
o shanmukha@shanmukha-Latitude-3420:~/Music/the_react$

```

You may get this error

```
"description": "",
```

```
> 5 | "main": "App.js",
```

```
> |
```

```
^^^^^^^^ Did you mean "App.html"?
```

```
6 | "scripts": {
```

```
7 | "test": "jest"
```

- The error is because we set main path as index.html but in the package.json main has App.js
- So now just delete that line in package.json
- When you built npx parcel build index.html it bundles all the files into the dist folder(these files are used in the production)
- The files in dist folder that is created by npx parcel index.html are used in development
- What you are seeing in browser just comes from the dist files itself

```

✦ Built in 399ms
• shanmukha@shanmukha-Latitude-3420:~/Music/the_react$ npx parcel build index.html
✦ Built in 766ms

dist/index.html          347 B    237ms
dist/index.459ed095.css   81 B     29ms
dist/index.12e56685.js  138.77 KB 261ms
o shanmukha@shanmukha-Latitude-3420:~/Music/the_react$

```

- You can observe index.html and dist/index.459ed095.css 81 B 29ms
- dist/index.12e56685.js it means the entire code you wrote is compressed into these files
- Whenever u run the npx parcel index.html or npx parcel build index.html , the folders parcel-cache and dist are automatically generated even if you delete them. So keep them in gitignore

```

.
├── .gitignore
├── 1 /node_modules
├── 2 /dist
├── 3 /.parcel-cache
└──

```

CONFIGURATING OUR APP IN ALL BROWSERS

Go to package.json and code something like this

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

All these things makes our react apps super faster and strengthen our app

```
25 },  
26 "dependencies": {  
27   "react": "^18.3.1",  
28   "react-dom": "^18.3.1"  
29 },  
30 "browserslist": [  
31   "last 2 versions"  
32 ]  
33 }  
34 |
```