ReactJS - Igniting

16 June 2025 01:25 AM

- Its not just React that creates our app faster.
- There are several packages.
- **We will start with npm.**
- **Npm does not have a full form.**

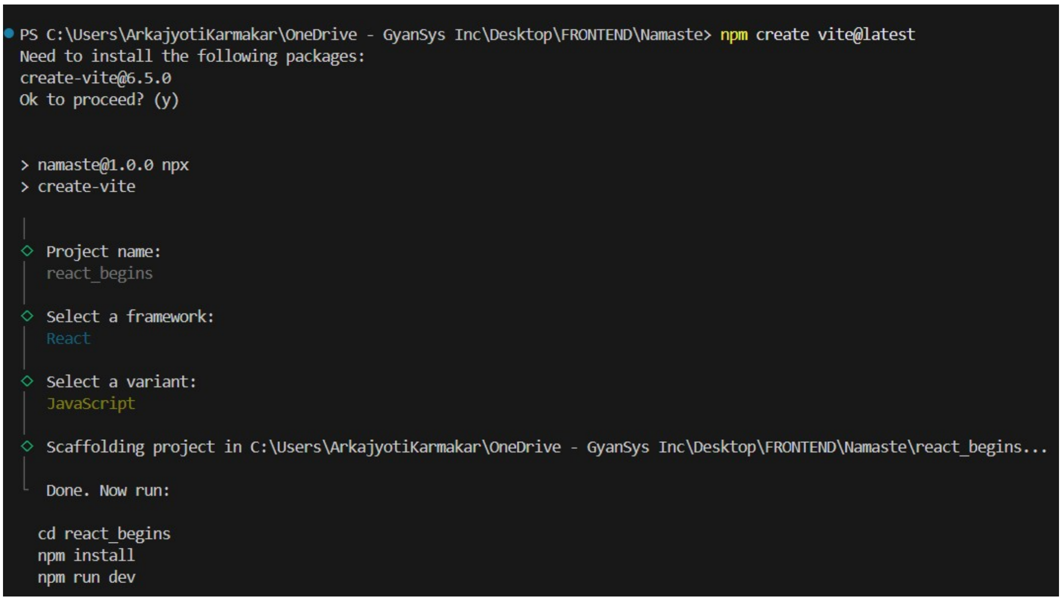
- **Its essentially a package manager.**
- **It's a standard repository for all the packages, utilities or libraries.**

- Package.json is a configuration of our npm.
- Sometimes the packages are known as dependencies.
- **The packages which is project is dependent on are called dependencies and NPM manages that.**

The most important package in our project is a Bundler:

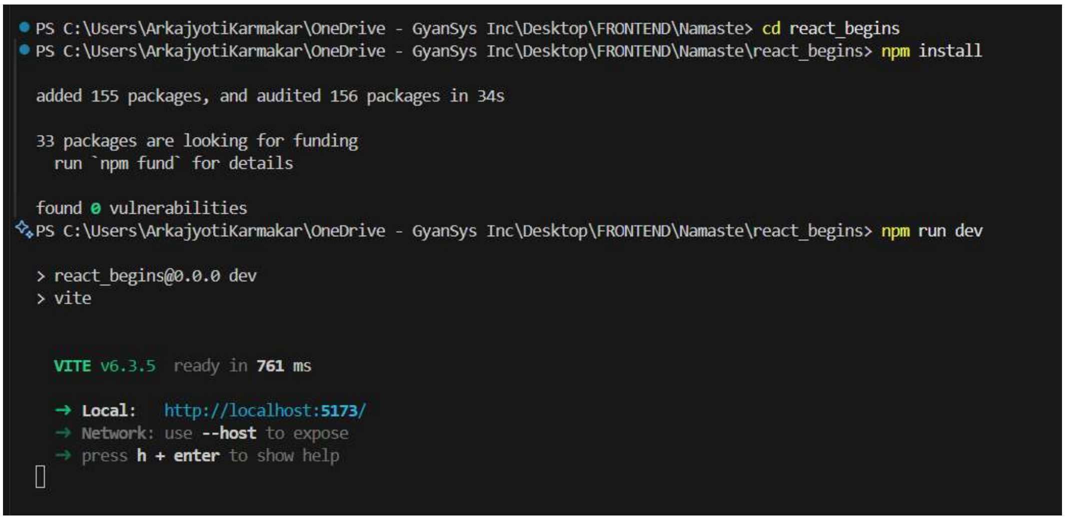
- **Bundles generally bundles our app so that it can be shipped to production.**
  - There are two kinds of dependencies an app can have.
  - **One is the dev dependencies and the other normal dependencies.**
  - A dev dependency is required in the development phase only.

- Here although in the course the Parcel is being used but I will be using Vite here.



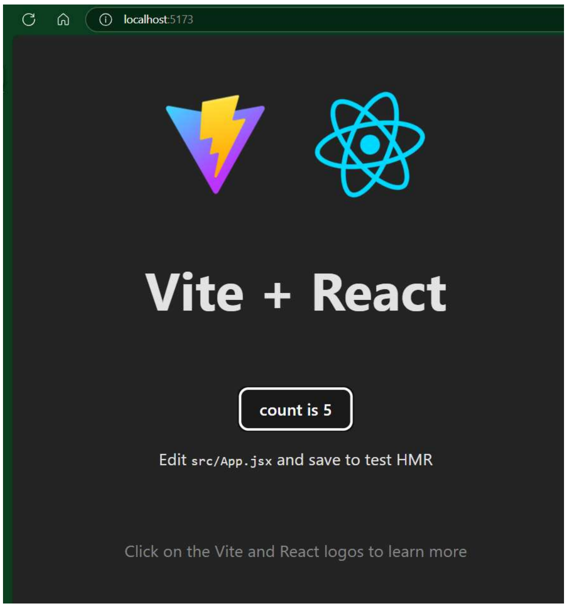
```
PS C:\Users\ArkajyotiKarmakar\OneDrive - GyanSys Inc\Desktop\FRONTEND\Namaste> npm create vite@latest
Need to install the following packages:
create-vite@6.5.0
Ok to proceed? (y)


> namaste@1.0.0 npx
> create-vite

│
◇  Project name:
│  react_begins
│
◇  Select a framework:
│  React
│
◇  Select a variant:
│  JavaScript
│
◇  Scaffolding project in C:\Users\ArkajyotiKarmakar\OneDrive - GyanSys Inc\Desktop\FRONTEND\Namaste\react_begins...
│
└  Done. Now run:

  cd react_begins
  npm install
  npm run dev
```

- Then following the rest of the things will create everything for us.



```
● PS C:\Users\ArkajyotiKarmakar\OneDrive - GyanSys Inc\Desktop\FRONTEND\Namaste> cd react_begins
● PS C:\Users\ArkajyotiKarmakar\OneDrive - GyanSys Inc\Desktop\FRONTEND\Namaste\react_begins> npm install

  added 155 packages, and audited 156 packages in 34s

  33 packages are looking for funding
    run `npm fund` for details

  found 0 vulnerabilities
PS C:\Users\ArkajyotiKarmakar\OneDrive - GyanSys Inc\Desktop\FRONTEND\Namaste\react_begins> npm run dev

  > react_begins@0.0.0 dev
  > vite


    VITE v6.3.5  ready in 761 ms

    ➜  Local:   http://localhost:5173/
    ➜  Network: use --host to expose
    ➜  press h + enter to show help
```

- Now on clicking the link we can see that is ready.



Vite + React

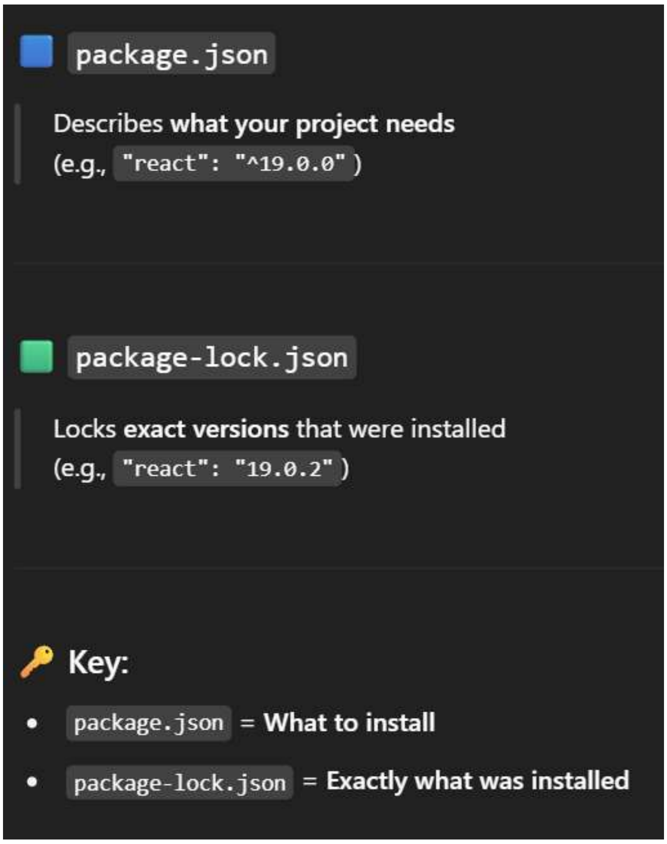
count is 5

Edit `src/App.jsx` and save to test HMR

Click on the Vite and React logos to learn more

- **Now we can check that without mentioning explicitly the vite is already part of our dev dependency.**
- **One thing :**
  - **'^' :** This carrat symbol means that **any smaller version** of that package when released will be automatically installed.
  - **'~' :** This suggests that **any major version** when released will be automatically installed.

- **Generally when we have a major upgrade it can break a lot of things in the code.**
- **Hence we use "^".**



🟦 `package.json`

> Describes **what your project needs** (e.g., `"react": "^19.0.0"`)

🟩 `package-lock.json`

> Locks **exact versions** that were installed (e.g., `"react": "19.0.2"`)

🔑 **Key:**

- `package.json` = **What to install**
- `package-lock.json` = **Exactly what was installed**

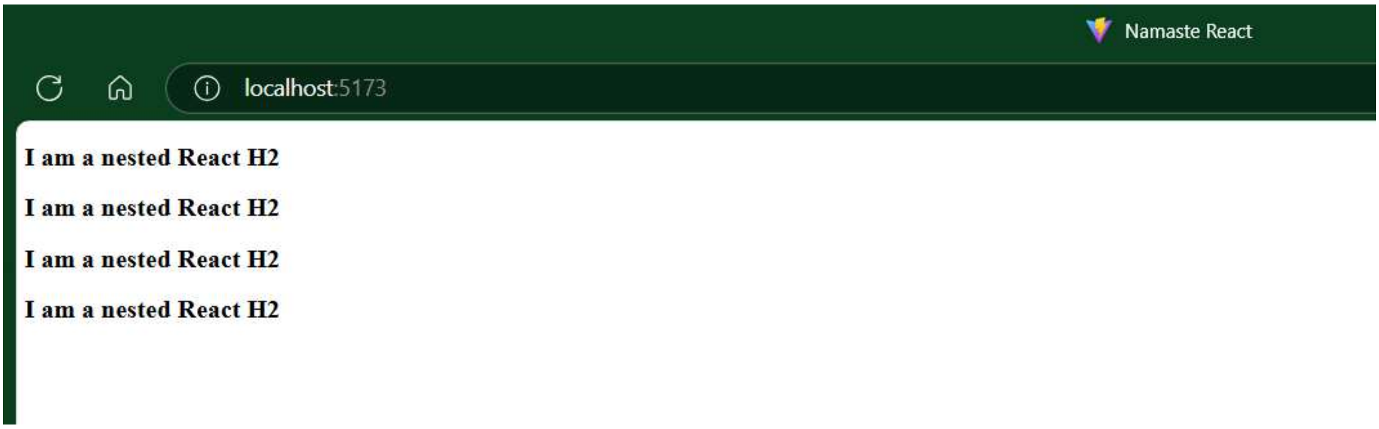
- **The node modules contain all the code we have fetched from NPM.**

****

- **Although we had just installed Vite but a lot other folder can be seen in the node_modules folder.**
  - Now our project needs Vite.
  - Similarly, Vite would be needing certain dependencies and this goes on.
  - This is known as **Transitive Dependencies.**

- **Every dependency has its own package.json.**

- As we run it using **npm run dev** we can see:



I am a nested React H2

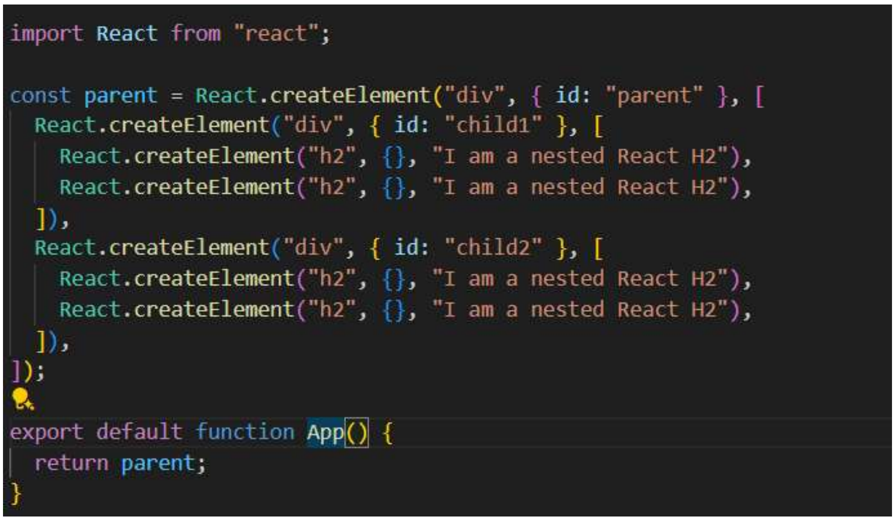
I am a nested React H2

I am a nested React H2

I am a nested React H2

- The window shown here is that of CDN and now we will be using React from package and not CDN links.

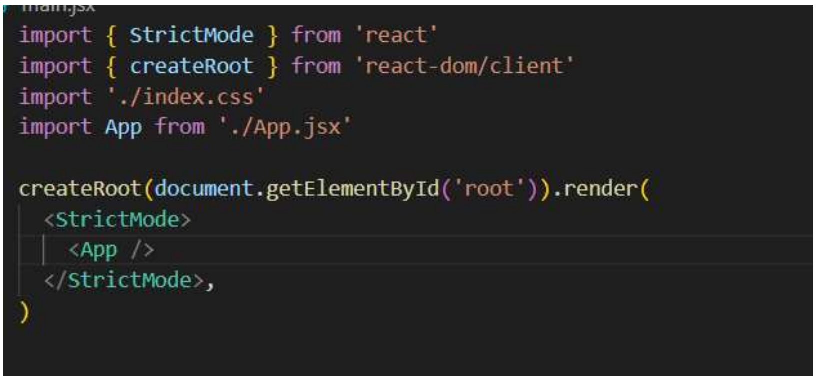
- Normal browser scripts don’t have imports.
- **For Vite the entry point is app.jsx.**

- 

```
import React from "react";

const parent = React.createElement("div", { id: "parent" }, [
  React.createElement("div", { id: "child1" }, [
    React.createElement("h2", {}, "I am a nested React H2"),
    React.createElement("h2", {}, "I am a nested React H2"),
  ]),
  React.createElement("div", { id: "child2" }, [
    React.createElement("h2", {}, "I am a nested React H2"),
    React.createElement("h2", {}, "I am a nested React H2"),
  ]),
]);

export default function App() {
  return parent;
}
```

- It is a default import.

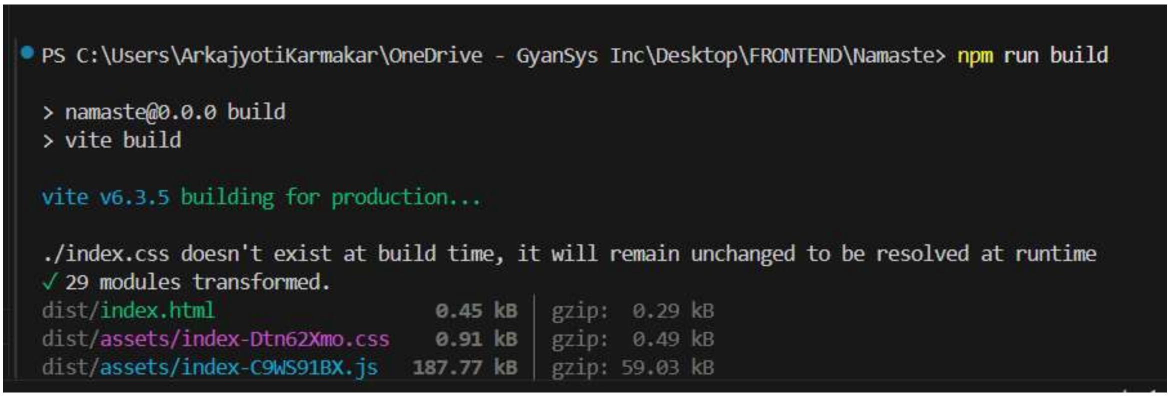
- In our main.jsx file we are receiving it.



```
import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import './index.css'
import App from './App.jsx'

createRoot(document.getElementById('root')).render(
  <StrictMode>
    <App />
  </StrictMode>,
)
```

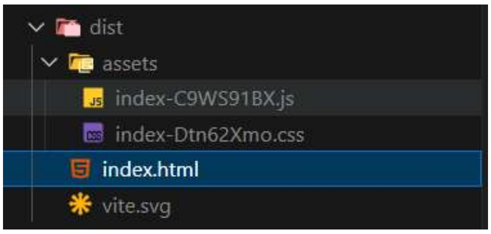
- `<script type="module" src="/src/main.jsx"></script>`
- We need to tell the browser that this file is not a normal javascript file but it’s a module.
- Hence we write **type="module".**

Production build in Vite using npm run build.

- 

```
PS C:\Users\ArkajyotiKarmakar\OneDrive - GyanSys Inc\Desktop\FRONTEND\Namaste> npm run build

> namaste@0.0.0 build
> vite build

vite v6.3.5 building for production...

./index.css doesn't exist at build time, it will remain unchanged to be resolved at runtime
✓ 29 modules transformed.
dist/index.html                  0.45 kB │ gzip:  0.29 kB
dist/assets/index-Dtn62Xmo.css   0.91 kB │ gzip:  0.49 kB
dist/assets/index-C9WS91BX.js  187.77 kB │ gzip: 59.03 kB
```

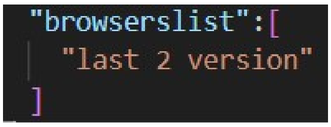
- This will generate a dist folder for us which contains a **js, css and index.html file for us.**

- 

```
dist
  assets
    index-C9WS91BX.js
    index-Dtn62Xmo.css
  index.html
  vite.svg
```

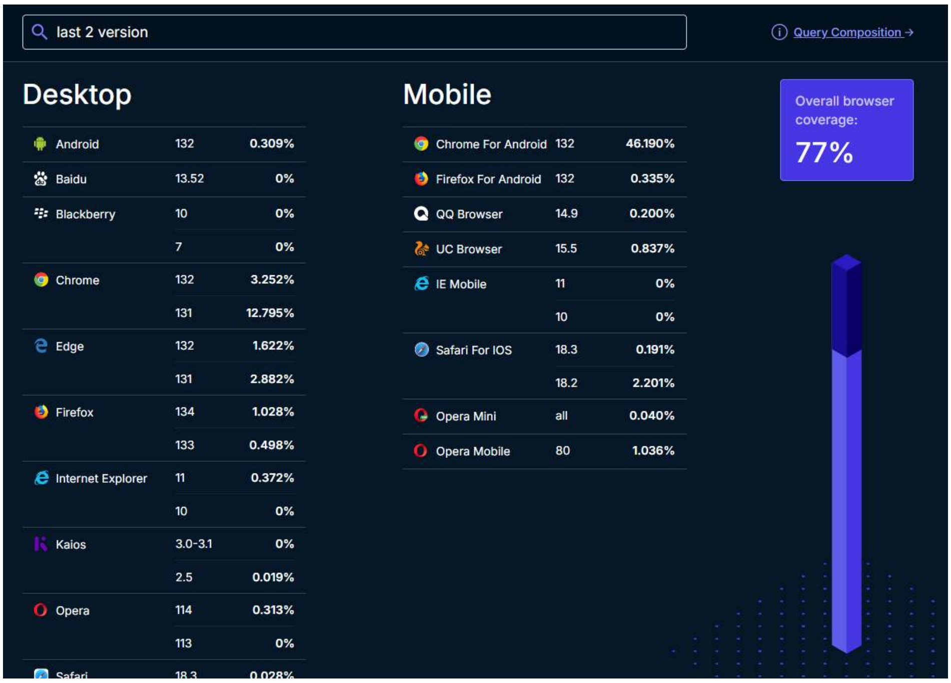
- **The copy of node modules on the server is different than the ones in the local.**

- Now we have a **browserslist npm package** in our node modules. To show the supported browsers.
- But we might also want to allow the older versions of the browsers to be able to use it.



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

- If we just do this then we will be able to use our app on the last 2 versions of multiple web browsers.



last 2 version

Query Composition →

Desktop

Android	132	0.309%
Baidu	13.52	0%
Blackberry	10	0%
	7	0%
Chrome	132	3.252%
	131	12.795%
Edge	132	1.622%
	131	2.882%
Firefox	134	1.028%
	133	0.498%
Internet Explorer	11	0.372%
	10	0%
Kaios	3.0-3.1	0%
	2.5	0.019%
Opera	114	0.313%
	113	0%
Safari	18.3	0.028%

Mobile

Chrome For Android	132	46.190%
Firefox For Android	132	0.335%
QQ Browser	14.9	0.200%
UC Browser	15.5	0.837%
IE Mobile	11	0%
	10	0%
Safari For IOS	18.3	0.191%
	18.2	2.201%
Opera Mini	all	0.040%
Opera Mobile	80	1.036%

Overall browser coverage:
77%

- Here we can see that the command last 2 version has so many accessibility.
- This keeps our app light.