REACT.JS:

Firstly we need to check if node js and npm packages are already installed or not. And if not we would firstly write:

- **apt update** .
- And then **apt upgrade** .
- And then **apt install nodejs and apt install npm**.

```
root1@LAPTOP-R268MI6J:~$ nodejs --version
v18.19.1
root1@LAPTOP-R268MI6J:~$ npm --version
9.2.0
root1@LAPTOP-R268MI6J:~$ sudo apt update
[sudo] password for root1:
Hit:2 https://download.docker.com/linux/ubuntu noble InRelease
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble InRelease
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:6 https://storage.googleapis.com/bazel-apt stable InRelease
Hit:5 https://pkg.jenkins.io/debian-stable binary/ Release
```

Then we would run the curl command to get data of Debian package of node source:

```
root1@LAPTOP-R268MI6J:~$ curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
2025-03-11 09:35:36 - Installing pre-requisites
Hit:1 https://download.docker.com/linux/ubuntu noble InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:4 https://storage.googleapis.com/bazel-apt stable InRelease
Hit:5 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:2 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease
```

Then to create a react app using terminal:

```
root1@LAPTOP-R268MI6J:~$ npx create-react-app hello-world
Need to install the following packages:
  create-react-app@5.1.0
Ok to proceed? (y) y
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache i
f you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerfu
l.
npm WARN deprecated uid-number@0.0.6: This package is no longer supported.
npm WARN deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated fstream-ignore@1.0.5: This package is no longer supported.
npm WARN deprecated fstream@1.0.12: This package is no longer supported.
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please
 upgrade asap.
create-react-app is deprecated.

You can find a list of up-to-date React frameworks on react.dev
For more info see:https://react.dev/link/cra

This error message will only be shown once per install.

Creating a new React app in /home/root1/hello-world.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

(                    ) ⸬ idealTree:workbox-webpack-plugin: timing idealTree:node_modules/workbox-webpack-plugin Completed
```

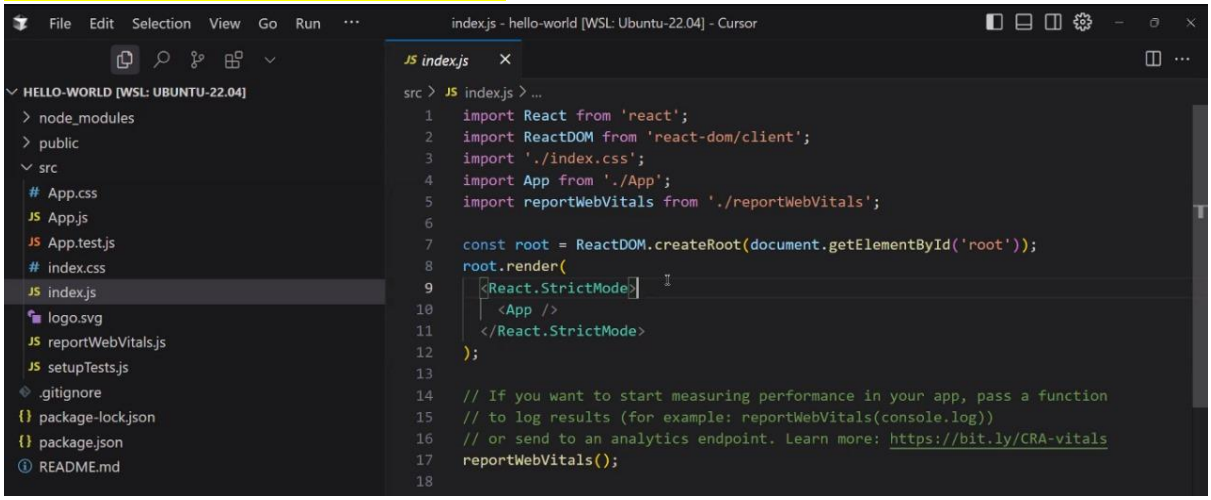
Then go into the hello world file and open cursor for working on app:

```
Success! Created hello-world at /home/root1/hello-world
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd hello-world
  npm start

Happy hacking!
root1@LAPTOP-R268MI6J:~$ cd hello-world/
root1@LAPTOP-R268MI6J:~/hello-world$ cursor .
```

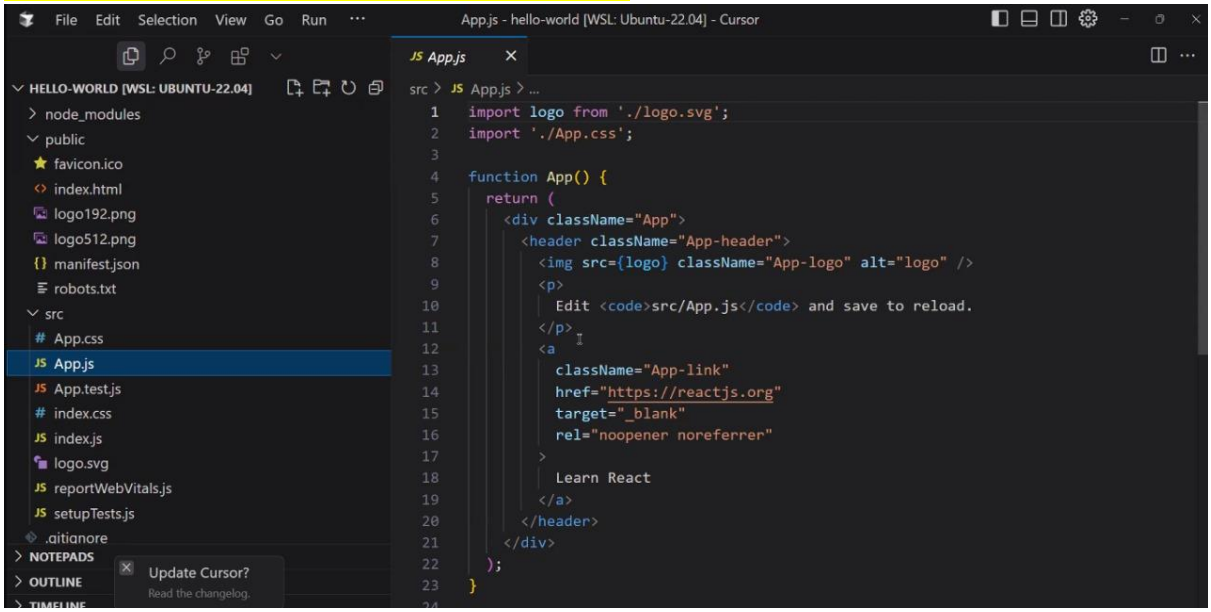
In cursor it would look like:

- Node module contains system files .
- Public folder contains all static files (like index.html , css etc).
- Then src folder contains all the js and react files.

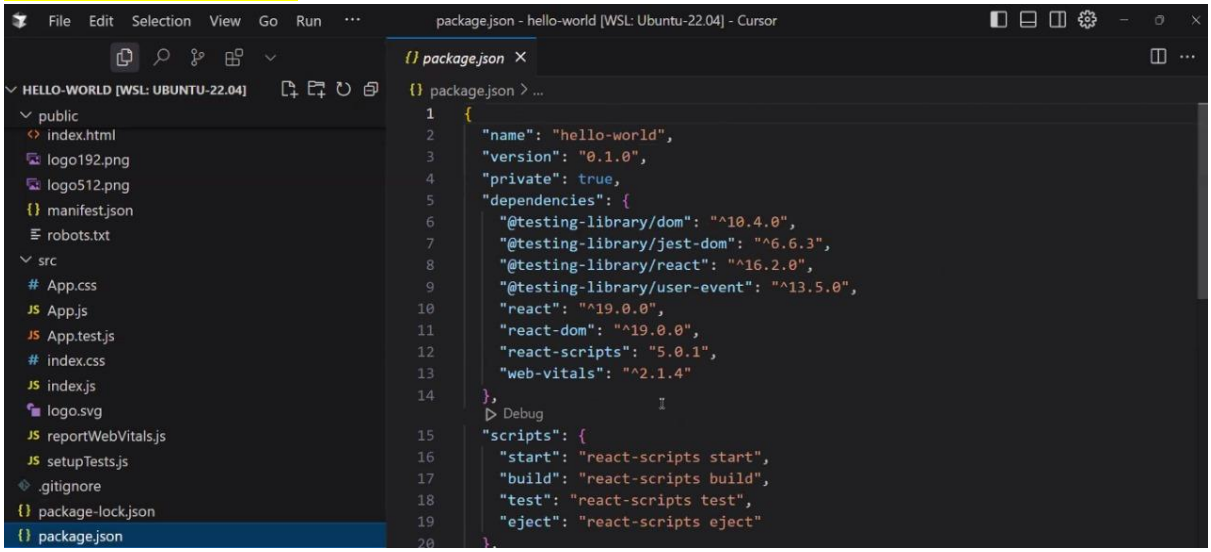
Now this is the first page from where the execution of our app begins "index.js" within src :



And app.js is the root controller , which interprets all the routings for the app for smooth functioning:

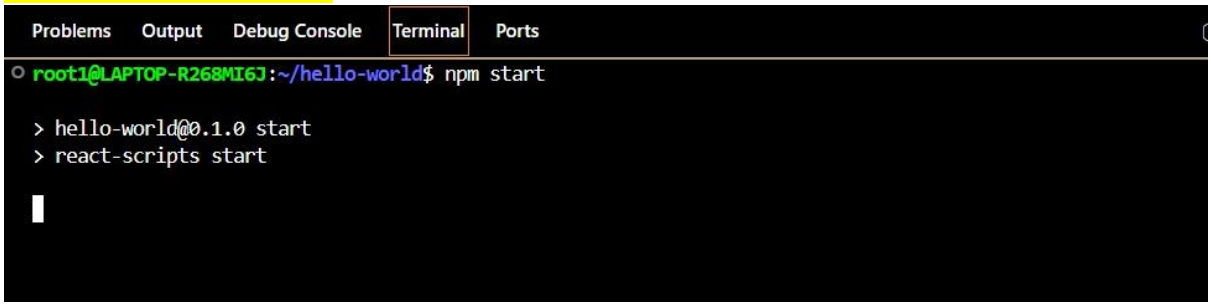


Package.json file define all the dependencies just like the requirement.txt:



And when we run our application it gets locked.

To start React app:



Index.js refers to all index.html and css and all other files. it kind of hacks and add code of lines in html and other files and app.js is the point where we write the actual code .

Example: in index.html

```
  -->
  <title>React App</title>
</head>
<body>
  <noscript>You need to enable JavaScript to run this app.</noscript>
  <div id="root"></div>
  <!--
    This HTML file is a template.
    If you open it directly in the browser, you will see an empty page.

    You can add webfonts, meta tags, or analytics to this file.
```

Chat Ctrl+I Edit Ctrl+K

Here we would create root using app.js

JS App.js M × <> index.html

src > JS App.js > App

```
1  import logo from './logo.svg';
2  import './App.css';
3
4  function App() {
5    return (
6      <div className="App">
7        <header className="App-header">
8          <h1>Hello World</h1>
9          <p>This is a test</p>
10       </header>
11     </div>
12   );
13 }
14
15 export default App;
```

Now we can create a component say counter.js:

```
JS App.js M    JS Counter.js U ×    index.html
src > JS Counter.js > ...
1  import React , { useState } from 'react';
2
3  function Counter() {
4      const [count, setCount] = useState(0);
5
6      return (
7          <div>
8              <p>Count: {count}</p>
9              <button onClick={() => setCount(count + 1)}>Increment</button>
10             <button onClick={() => setCount(count - 1)}>Decrement</button>
11             <button onClick={() => setCount(0)}>Reset</button>
12         </div>
13     )
14 }
15
16 export default Counter;
17
```

And then in app.js we simply write:

```
JS App.js M ×    JS Counter.js U    index.html
src > JS App.js > App
1  import logo from './logo.svg';
2  import './App.css';
3  import Counter from './Counter';
4  function App() {
5    return (
6      <div className="App">
7        <header className="App-header">
8          <h1>Hello World</h1>
9          <p>This is a test</p>
10         <Counter />
11       </header>
12     </div>
13   );
14 }
15
16 export default App;
17
```

That is firstly we are simply importing it and then we are using it within our code.

And within the component we are exporting the code.

Similarly we can create various components like:

Todo list:

```
import React , { useState } from 'react';
function Todo() {
    const [todos, setTodos] = useState([]);
    const [input, setInput] = useState('');
    const addTodo = () => {
        if (input.trim() !== '') {
            setTodos([...todos, input]);
            setInput('');
        }
    };
    const removeTodo = (index) => {
        setTodos(todos.filter((_, i) => i !== index));
    };
    return (
        <div>
            <h1>Todo List</h1>
            <input
                type="text"
                value={input}
                onChange={(e) => setInput(e.target.value)}
                placeholder="Add a todo"
            />
            <button onClick={addTodo}>Add</button>
            <ul>
                {todos.map((todo, index) => (
                    <li key={index}>{todo}
                    </li>
                ))}
            </ul>
        </div>
    )

}
export default Todo;
```

here we can create an html for this to return the value in list format when displaying result of todo.

Theme-switcher:

```
import React, { useState, useEffect } from 'react';

function ThemeSwitcher() {
  const [theme, setTheme] = useState('light');

  useEffect(() => {
    // Access the document body to change the background color
    document.body.style.backgroundColor = theme === 'light' ? 'white' :
'black';
    document.body.style.color = theme === 'light' ? 'black' : 'white';
  }, [theme]); // Re-run effect when theme changes

  const toggleTheme = () => {
    setTheme(theme === 'light' ? 'dark' : 'light');
  };

  return (
    <button onClick={toggleTheme}>
      {theme === 'light' ? 'Dark Mode' : 'Light Mode'}
      <style jsx>{`
        button {
          background-color: ${theme === 'light' ? 'white' : 'black'};
          color: ${theme === 'light' ? 'black' : 'white'};
        }
      `}</style>
    </button>
  );
}

export default ThemeSwitcher;
```

Now an component to take user info and display within the table:

```
import React, { useState } from 'react';
function UserInfo() {
  const [users, setUsers] = useState([
    { name: '', age: '', email: '' },
  ]);

  const [tableStyle, setTableStyle] = useState({
    border: '1px solid black',
    borderColor: 'black',
  });

  const handleChange = (index, e) => {
    const { name, value } = e.target;
    const updatedUsers = [...users];
    updatedUsers[index][name] = value;
    setUsers(updatedUsers);
  };

  const handleStyleChange = (e) => {
    const { name, value } = e.target;
    setTableStyle({ ...tableStyle, [name]: value });
  };

  const addUser = () => {
    setUsers([...users, { name: '', age: '', email: '' }]);
  };

  return (
    <div>
      <div>
        <label>Border:</label>
        <input type="text" name="border" value={tableStyle.border}
onChange={handleStyleChange} />
        <label>Border Color:</label>
        <input type="color" name="borderColor" value={tableStyle.borderColor}
onChange={handleStyleChange} />
      </div>

      {users.map((user, index) => (
        <div key={index}>
          <label>Name:</label>
          <input
            type="text"
            name="name"
            value={user.name}
            onChange={(e) => handleChange(index, e)}
          />
```

```
          <label>Age:</label>
          <input
            type="number"
            name="age"
            value={user.age}
            onChange={(e) => handleChange(index, e)}
          />
          <label>Email:</label>
          <input
            type="email"
            name="email"
            value={user.email}
            onChange={(e) => handleChange(index, e)}
          />
        </div>
      ))}

      <button onClick={addUser}>Add User</button>

      <table
        style={{
          border: tableStyle.border,
          borderColor: tableStyle.borderColor,
          borderCollapse: 'collapse',
        }}
      >
        <thead>
          <tr>
            <th style={{border: tableStyle.border}}>Name</th>
            <th style={{border: tableStyle.border}}>Age</th>
            <th style={{border: tableStyle.border}}>Email</th>
          </tr>
        </thead>
        <tbody>
          {users.map((user, index) => (
            <tr key={index}>
              <td style={{border: tableStyle.border}}>{user.name}</td>
              <td style={{border: tableStyle.border}}>{user.age}</td>
              <td style={{border: tableStyle.border}}>{user.email}</td>
            </tr>
          ))}
        </tbody>
      </table>
    </div>
  );
}

export default UserInfo;
```

and lastly for the routing thing:

Home.js-

```
import {useNavigate} from 'react-router-dom';
function Home() {
    const navigate = useNavigate();
    return (
        <div>
            <h1>Home</h1>
            <button onClick={() => navigate('/about')}>About</button>
        </div>
    );
}
export default Home;
```

About.js-

```
function About() {
    return (
        <div>
            <h1>About</h1>
        </div>
    );
}
export default About;
```

and then we need to install router for react using:

```
root1@LAPTOP-R268MI6J:~/hello-world$ npm install react-router-dom
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'react-router-dom@7.3.0',
npm WARN EBADENGINE   required: { node: '>=20.0.0' },
npm WARN EBADENGINE   current: { node: 'v18.19.1', npm: '9.2.0' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'react-router@7.3.0',
```

and finally in app.js we import all the components and use them:

```
import './App.css';
import Counter from './Counter';
import Todo from './Todo';
import ThemeSwitcher from './ThemeSwitcher';
import UserInfo from './UserInfo';
import { Routes, Route } from 'react-router-dom';
import Home from './Home';
import About from './About';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <h1>Hello World</h1>
        <p>This is a test</p>
        <Counter />
        <Todo />
        <ThemeSwitcher />
        <UserInfo />
        <Routes>
             <Route path="/home" element={<Home />} />
             <Route path="/about" element={<About />} />
        </Routes>
      </header>
    </div>
  );
}
export default App;
```

Output window: