by Kunal Sir

Introduction To The ReactJS

React is open-source JavaScript library for building user interfaces.

React is JavaScript library, not a framework.

React is focused on building user-interface, instead of navigations and API calls.

React has it's on virtual DOM.

React provides some inbuild classes and functions by using them we can create react elements, those react elements are used to create UI.

React is widely used for building single page applications (SAP).

Features Of React:

- React has component (re-usable blocks) based architecture.
- React will efficiently handle rendering and updating of the UI components with its virtual DOM.
- DOM updates are handled gracefully in react.

CJC Complete Java Classes

Q) What is single page application?

A single-page-application is a application which has only one HTML page. And the perfect example of single-page-applications are Netflix and Gmail.

In the single-page-application we can navigate around URL, and it seems like we are visiting different pages. But in the reality page never get changes only content of page changes. (It will re-write the page).

❖ **Advantages of SAP: -**

1. High performance and stability.
2. Code separation.
3. Fast development.

Prerequisites: -

1. HTML
2. CSS
3. JavaScript fundamentals (ES6).

React Elements: -

The elements, which are created using React library are react elements.

To create react elements, react library provides us one pre-define function called **createElement( ).** Using this function we can create react elements.

Ex.

let element =React.createElement("h1",{id:"heading" }, "Welcome to ReactJS");

createElement( ) function have three parameter arguments as below: -

1st argument: - Tag name for element which we want to create.

2nd argument: - Object which contain attributes and their values for the element.

3rd argument: - children/ innerHTML of an element.

ReactDOM: -

ReactDOM is also known as virtual dom.

ReactDOM is an important library and it is use to add react element in to the Actual DOM.

ReactDOM has per-define function **render( ).**

EX.

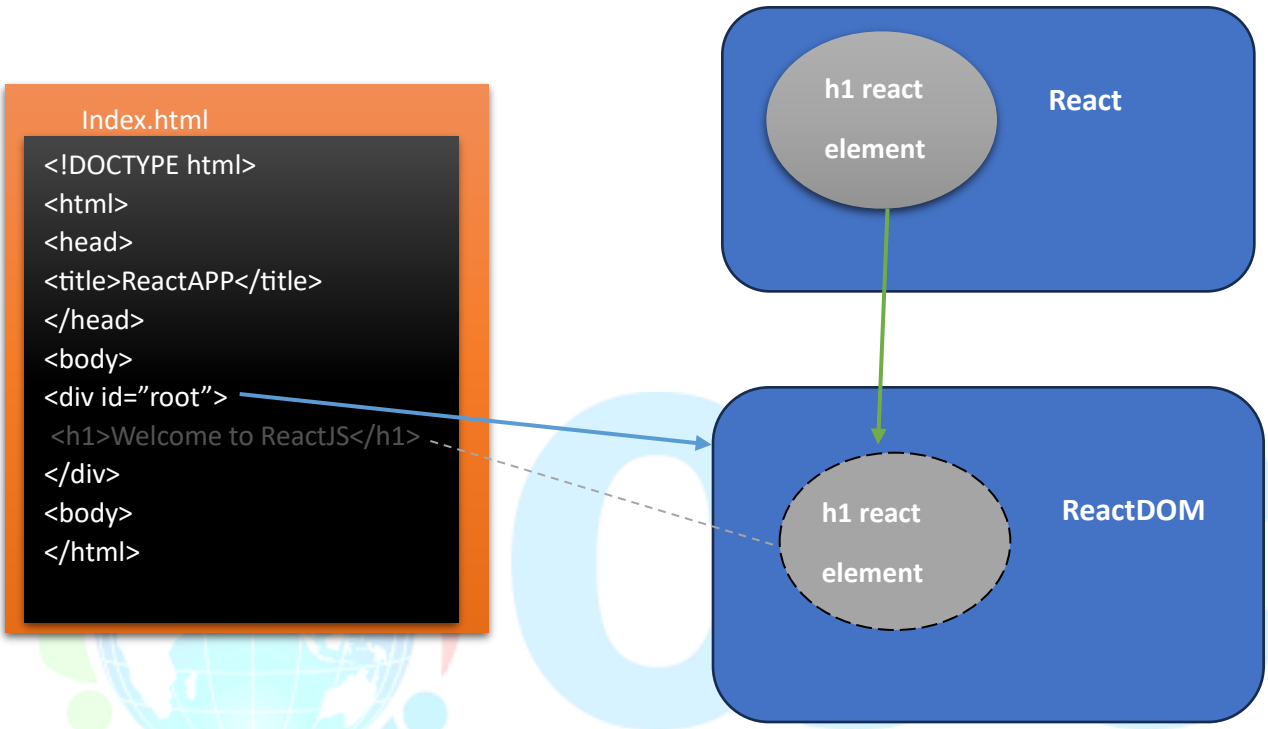
```
let element =React.createElement("h1",{id:"heading"  ,width:'100px'}, "Welcome to ReactJS");

ReactDOM.render(element , document.getElementById("root"));
```

render( ) function have two parameter arguments.

1st argument: - React element which need to render.

2nd argument: - reference of dom element where we want to add our react element.

Q) Create one simple h1 element using React and ReactDOM library.



Note :- To use React and ReactDOM libraries add their CDN links in html page.

```
<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>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <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>
    <title>Document</title>
</head>
<body>
    <div id="root">

    </div>
    <script>
        // Creating h1 React Element
        let h1Element=React.createElement("h1",{id:"heading"},"Welcome To ReactJs.");

        // Adding h1 react element to the DOM Using ReactDOM library.
        ReactDOM.render(h1Element,document.getElementById("root"));

    </script>
</body>
</html>
```

Task: - Create One div element using react library and add some inline CSS to it. Render this react div element in “root” dom element using ReactDOM.

Problem statement: -

Creating User-interfaces with React is as complex as creating user-interfaces with JavaScript.

Q) What is JSX and why was it introduced?

- JSX stands for JavaScript XML Extension.
- JSX allows us to write HTML in React.
- JSX makes it easy to write and add HTML in react.
- JSX is very similar to the HTML.
- Before JSX, Create User-interfaces with React is as complex as creating user-interfaces with JavaScript.

Q) What is babel compiler?

Babel compiler is use in react to compile the JSX code, and convert it into react elements.

```
<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
```

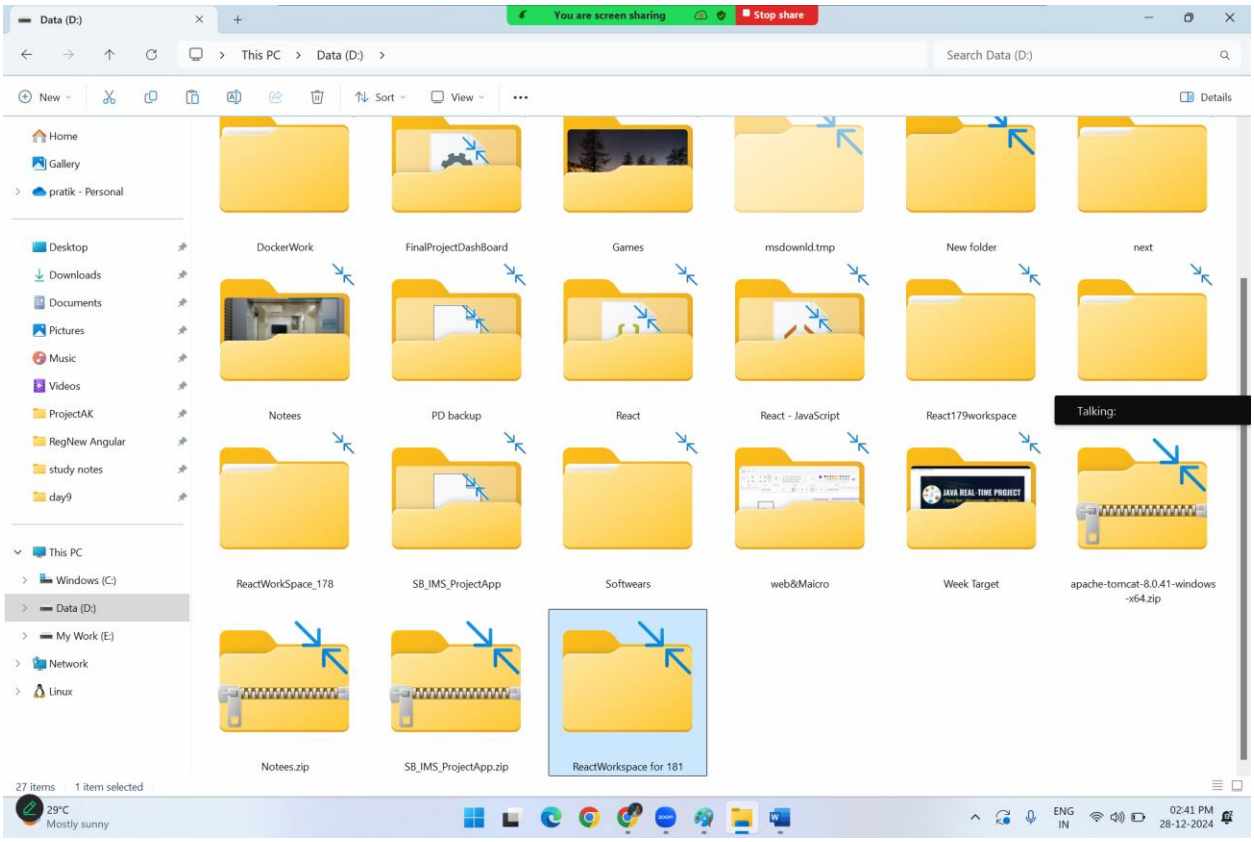
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
    <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
    <title>Document</title>
</head>
<body>
    <div id="root">
    </div>
    <script type="text/babel">
        var btn=<h1>Welcome to Easy ReactJS </h1>;
        ReactDOM.render(btn,document.getElementById("root"));
    </script>
</body>
</html>
```

CJC
Complete Java Classes

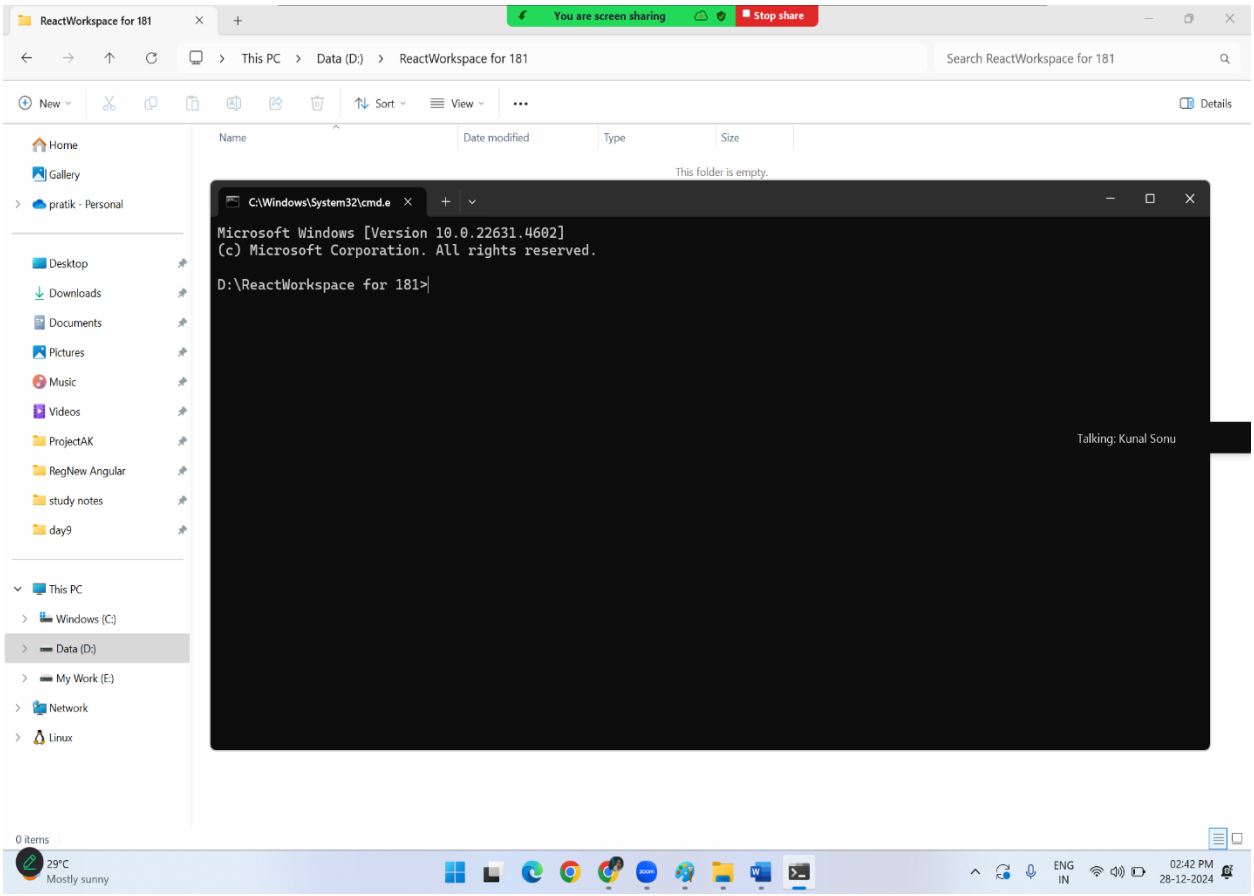
by Kunal Sir

Creating React Application using Vite:

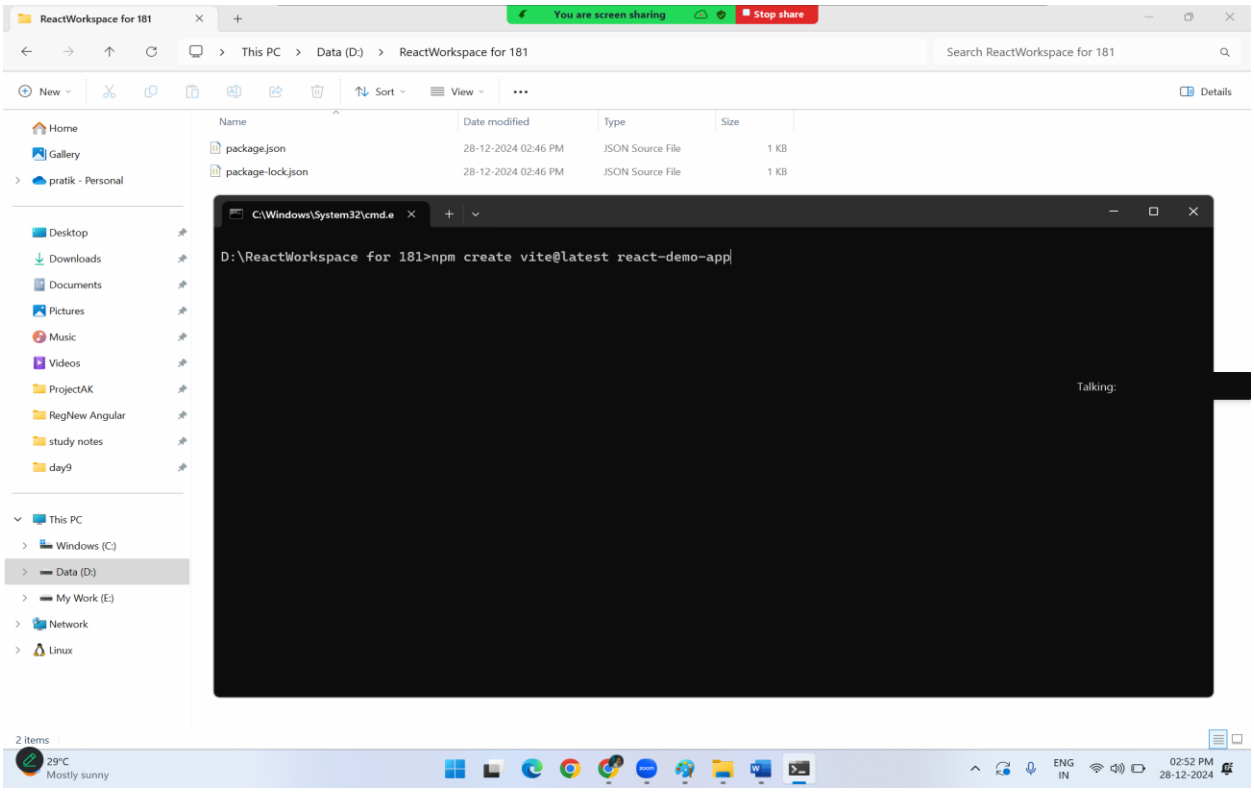
Step 1: - Create A workspace for react projects.



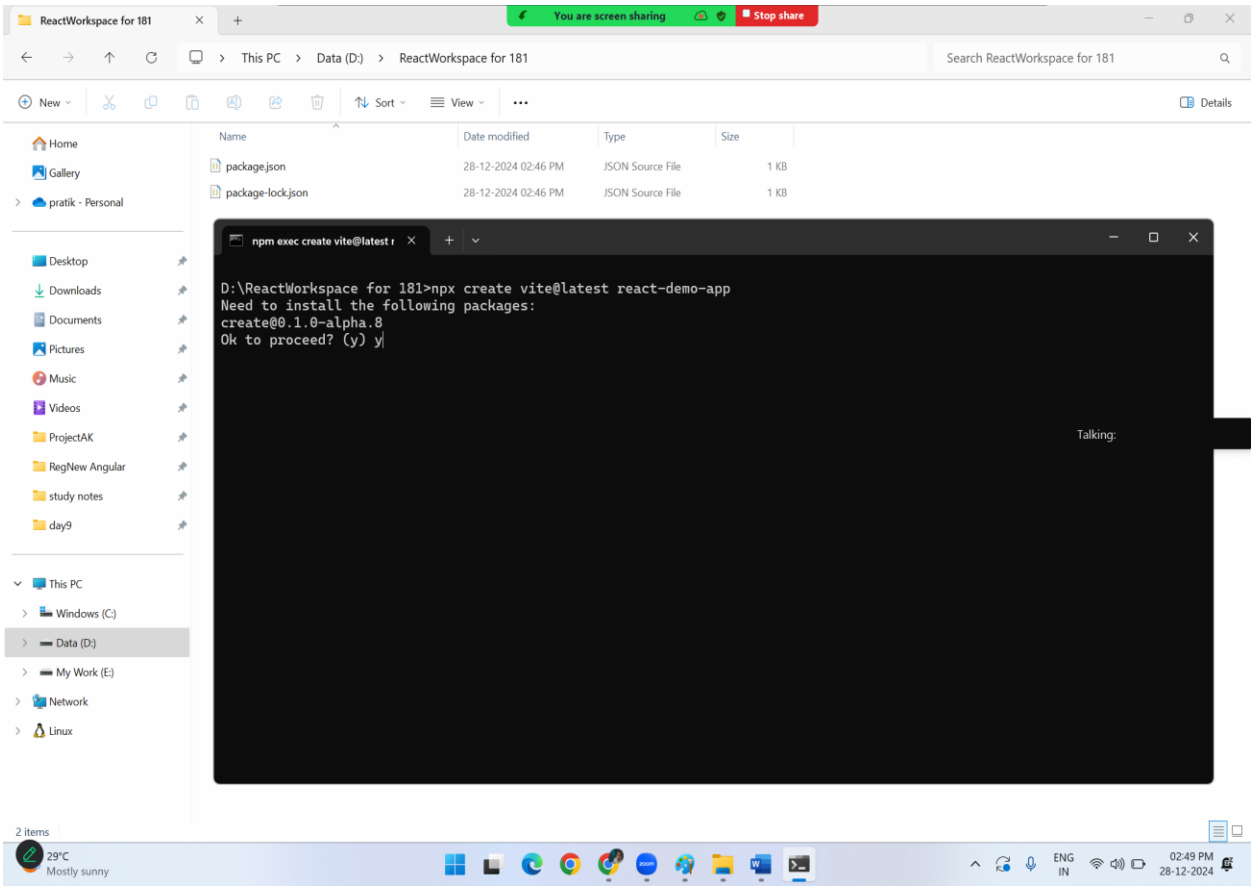
Step 2: - Open workspace directory and launch CMD.



Step 3: - Create React application using vite tool.

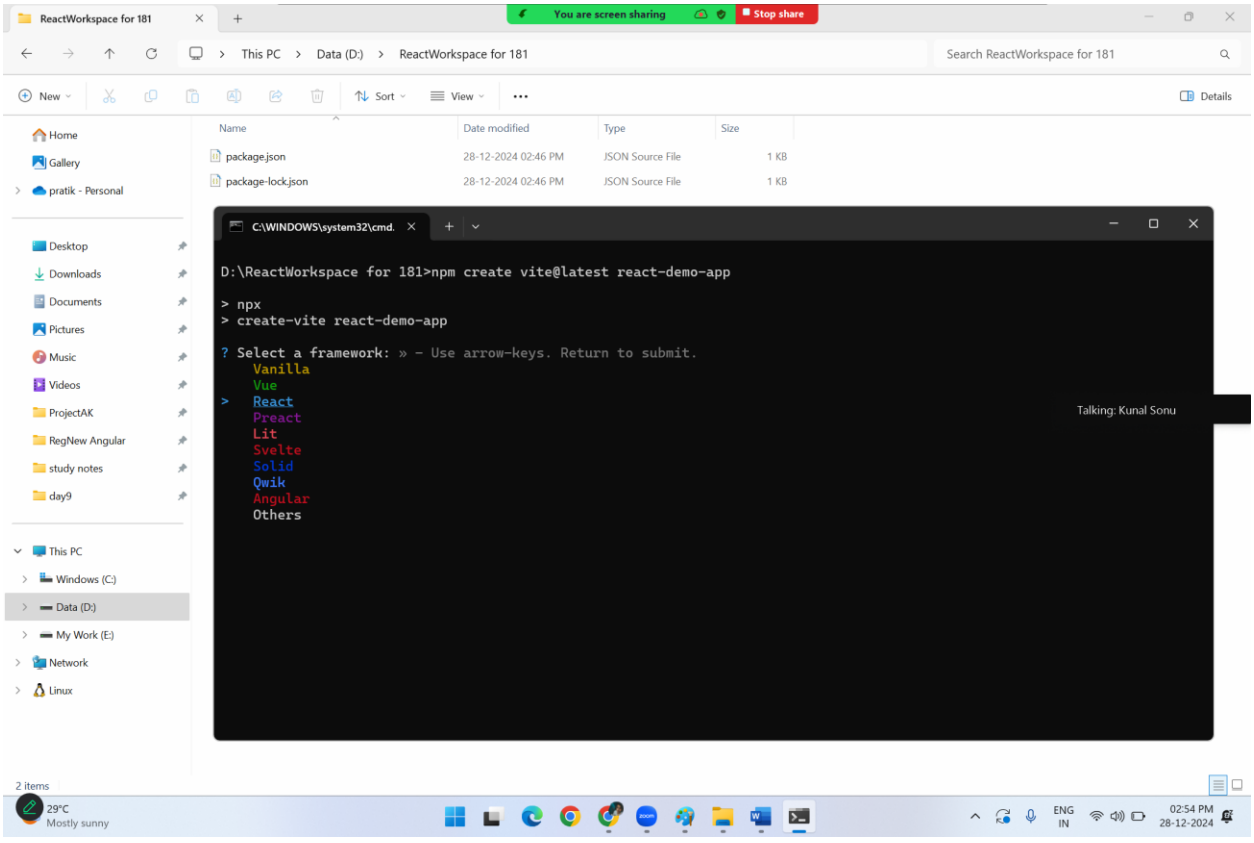


Step 4: - for the first time only, it will ask to install vite tool. Enter ` y` to proceed.

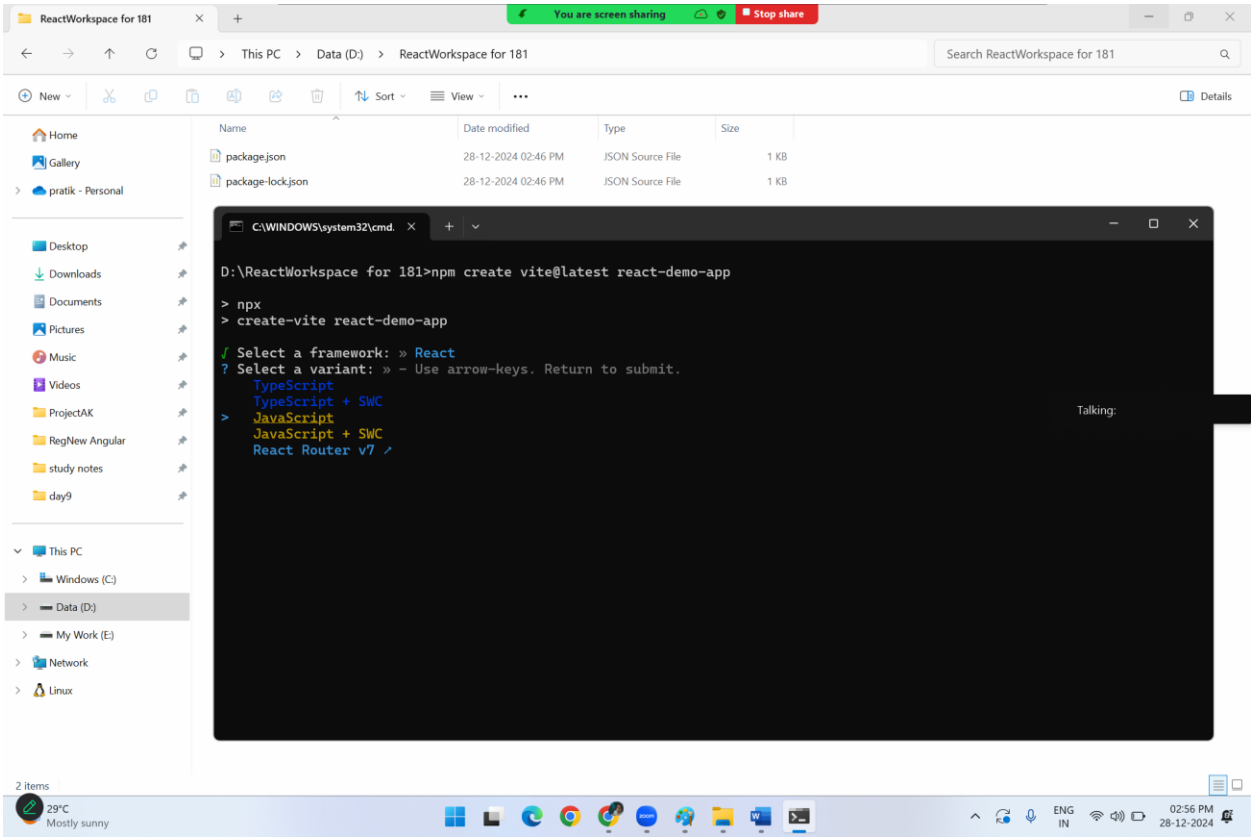


Step 5: - After Enter, it will install require packages and further proceed to create application, we need to select out project framework/library which we want to use.

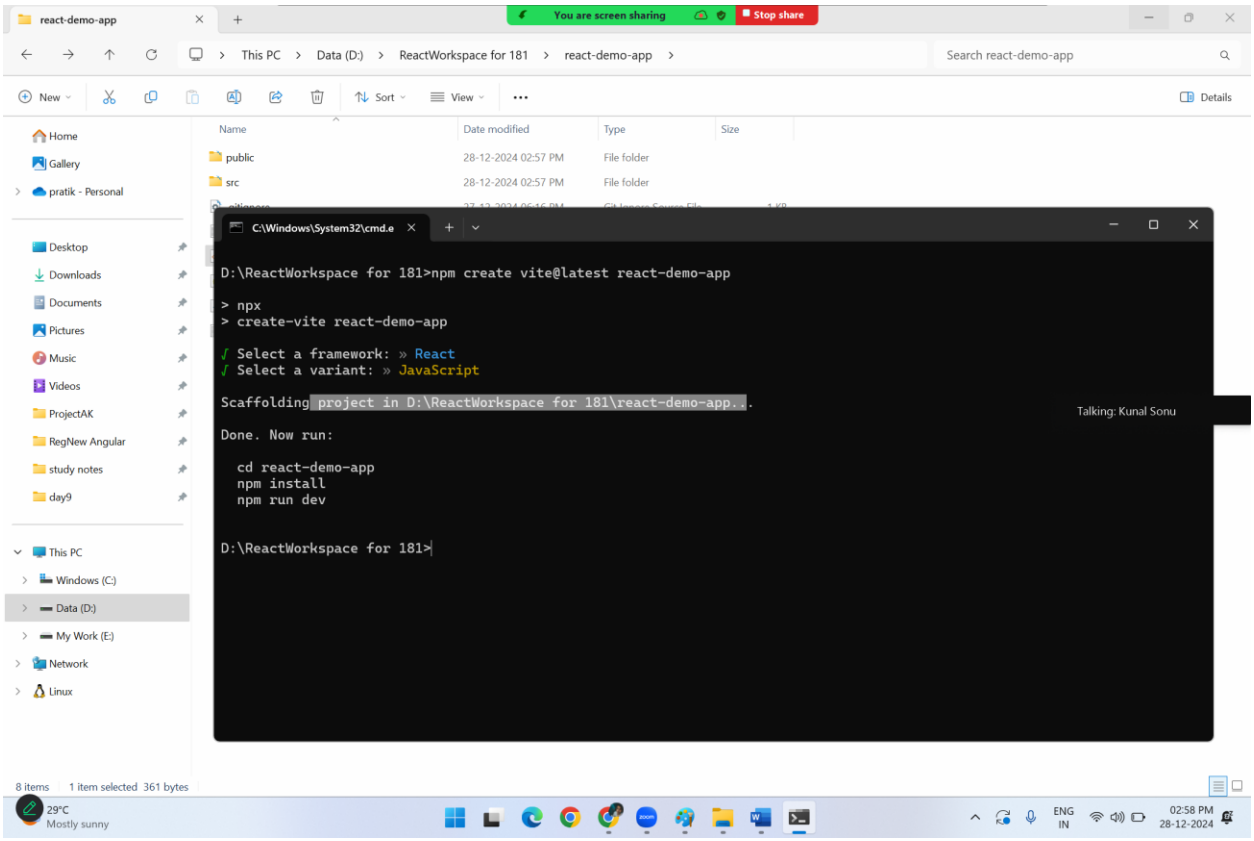
We can switch between given option using up-key or down key



Step 6: Now we have select variant/programming language. Select JavaScript as a programming language.

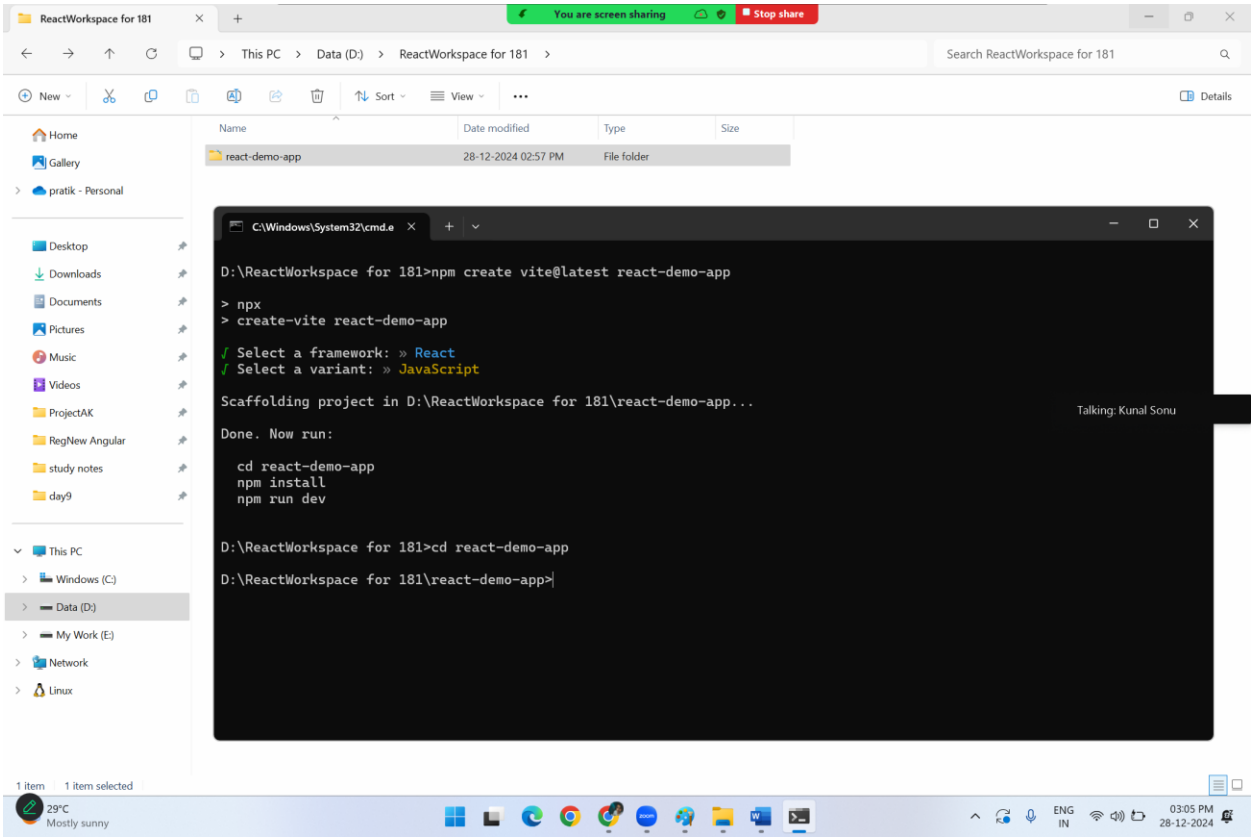


Step 7: - Once we specify programming language, vite will generate the project structure in our workspace directory.

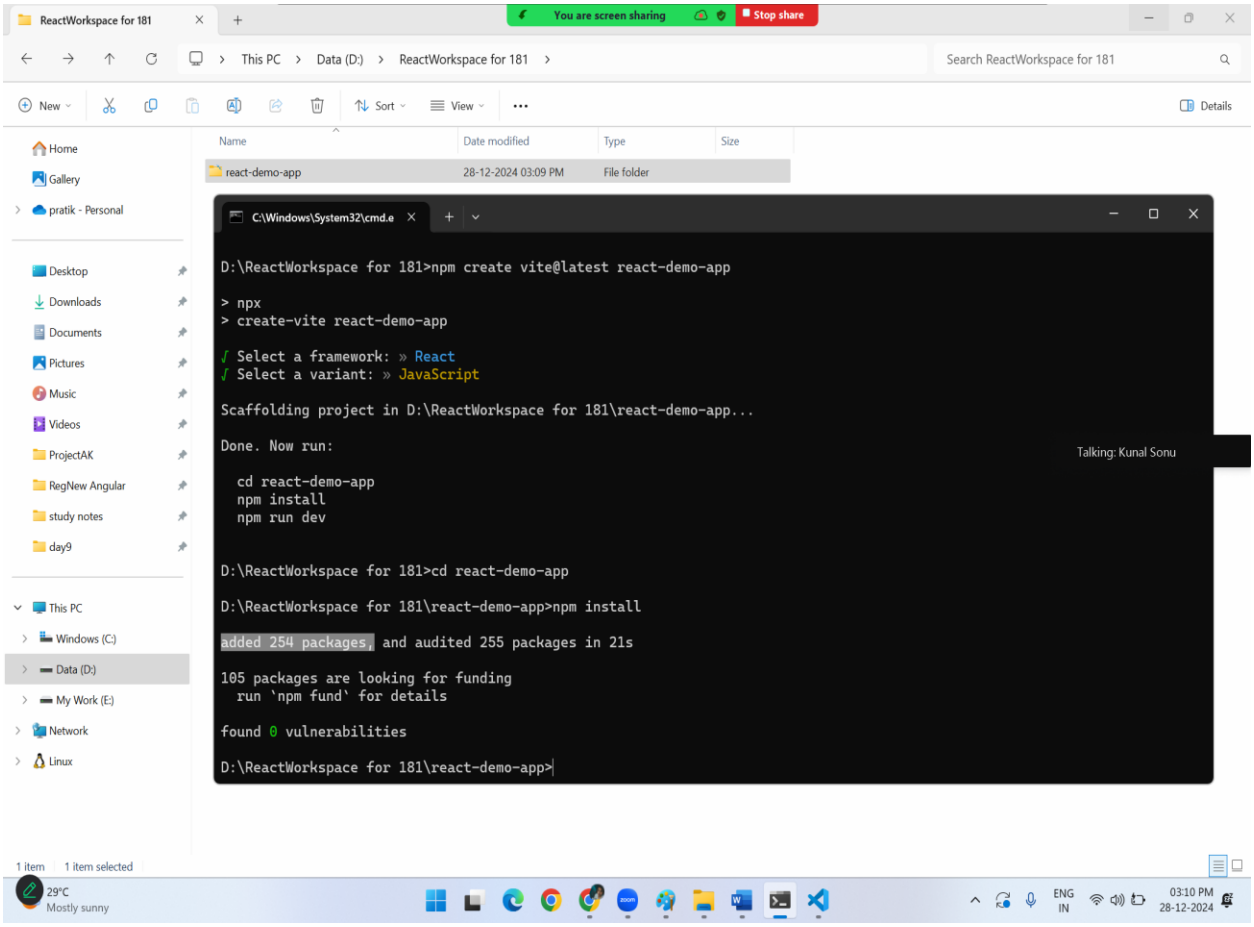


Note: - Now, this project will not hand any required libraries yet. To get/download the require libraries we need to perform below steps

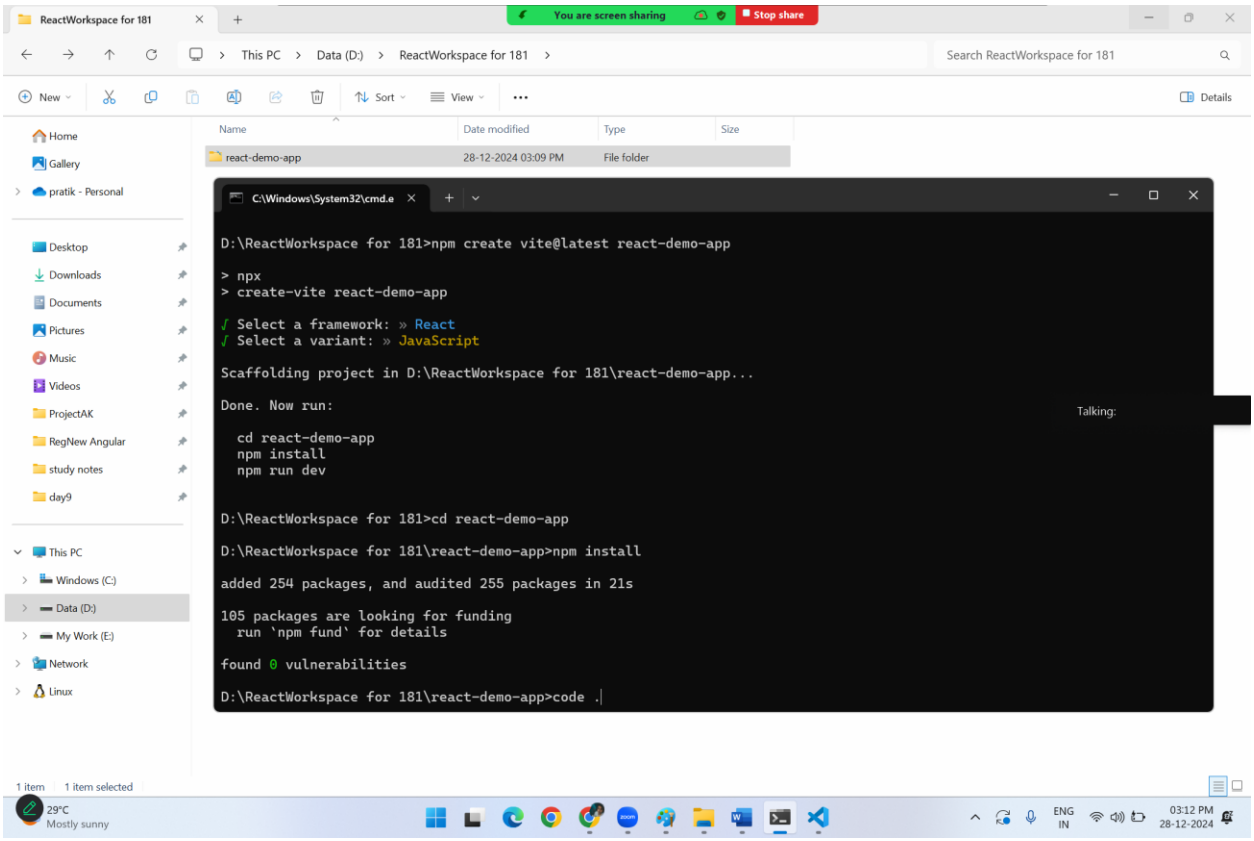
Step 8: - move the command prompt directory to the project directory using ` cd ` command.



Step 9: - Now using `npm install` command we can download all the require dependencies which is specified in package.json file in project directory.



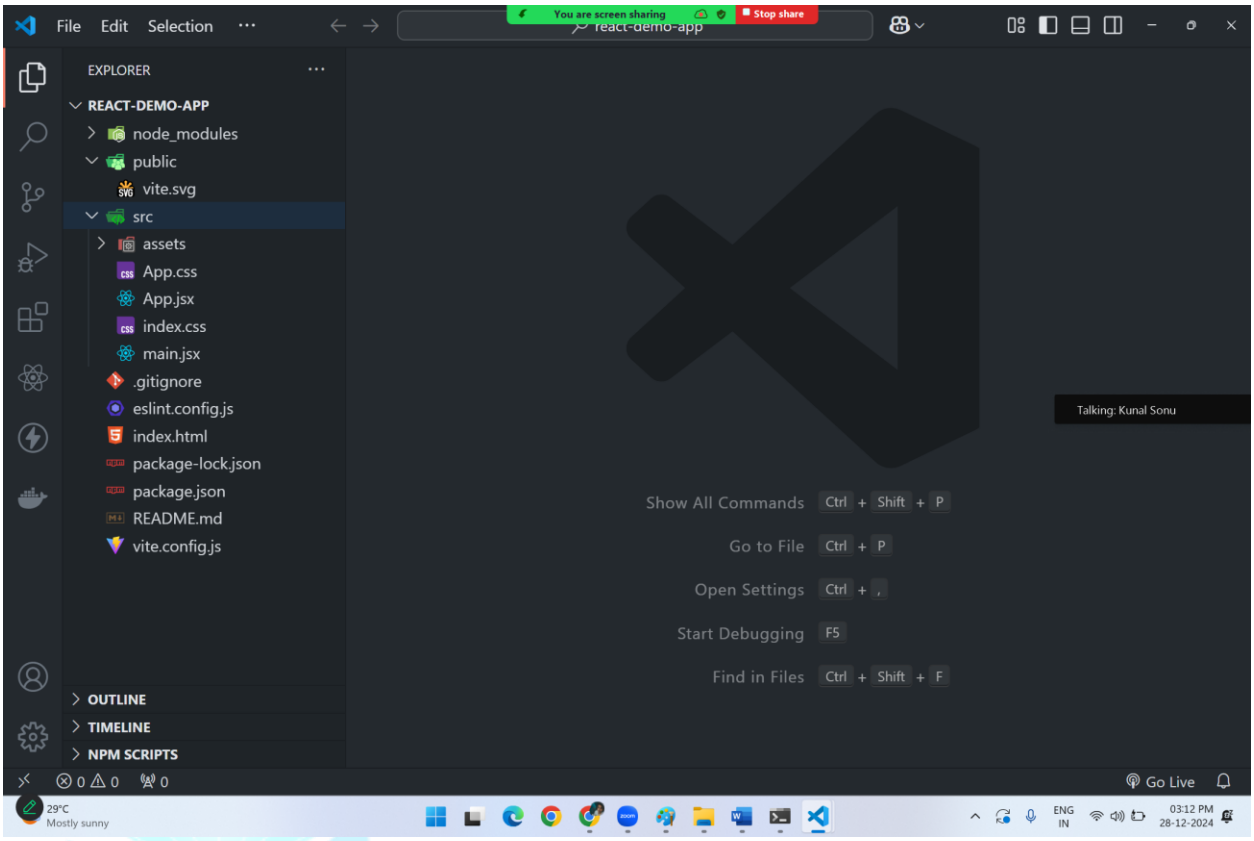
Step 10: - type `code .` to launch the VS-Code ide into project root directory.



CJC

Complete Java Classes

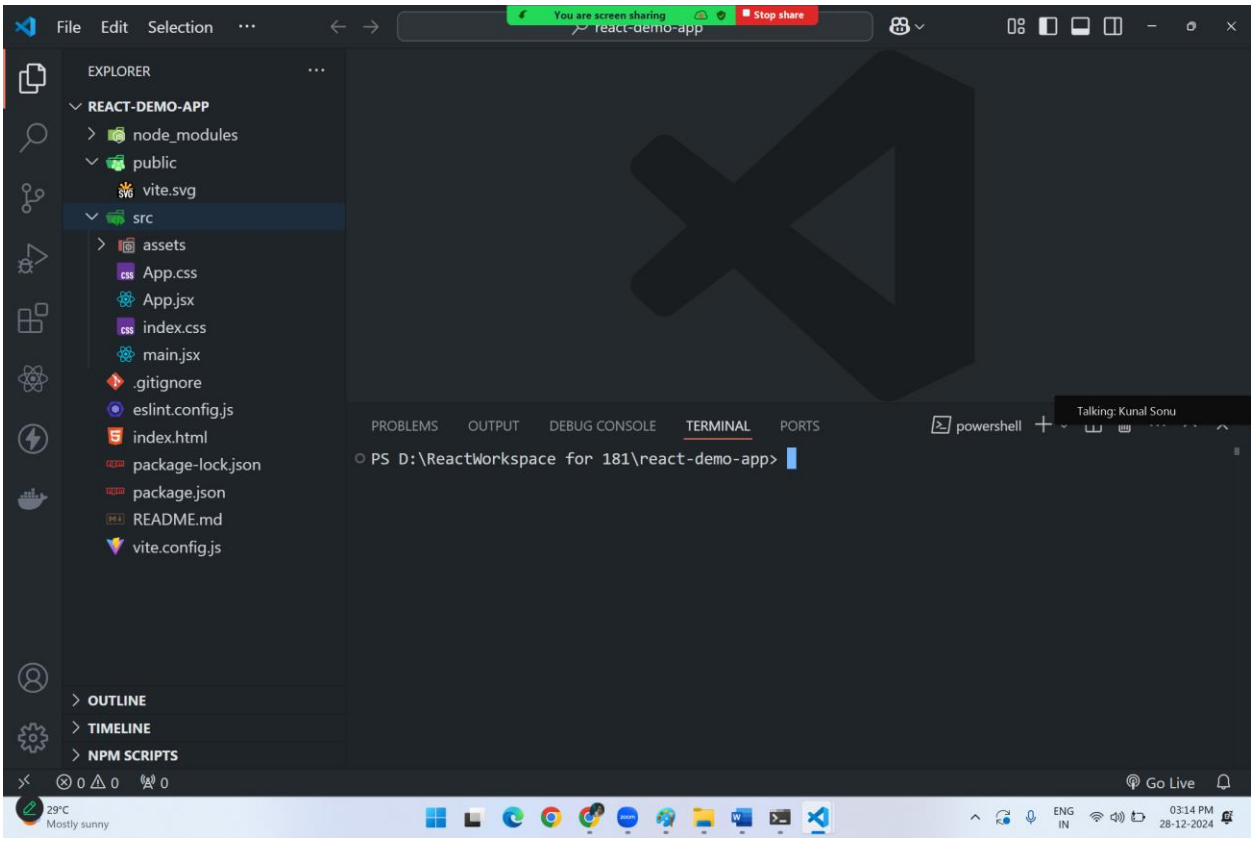
by Kunal Sir



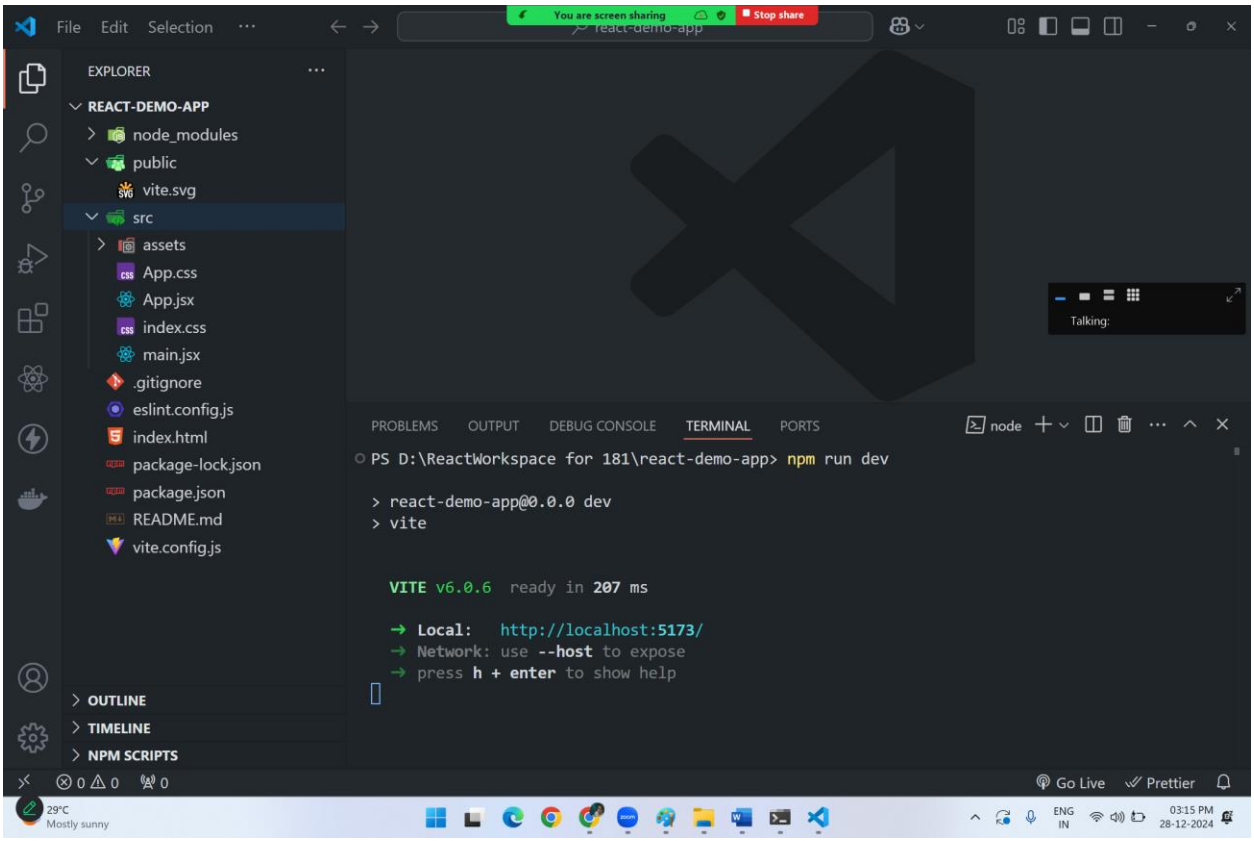
Complete Java Classes

Stop, Near, 1st Floor, Above Rupam Sweets/ Priyanka Collections Building Vikas Mitra Mandal Chowk Road, Karve Nagar, Pune, Maharashtra 411052 , Mobile No.- 8888022204

Step 11: - open the VS-Code's terminal to run the project.

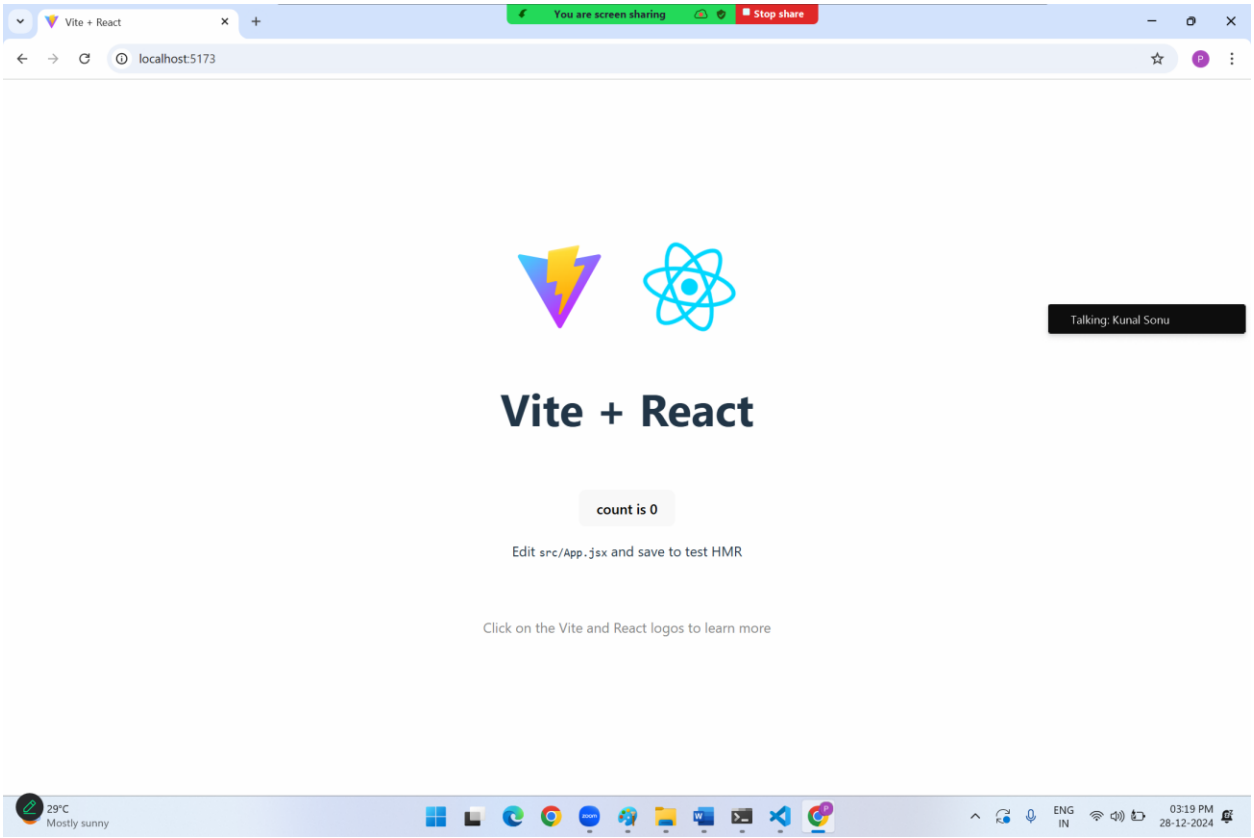


Sptep 12: - to run the project enter `npm run dev` command in terminal.



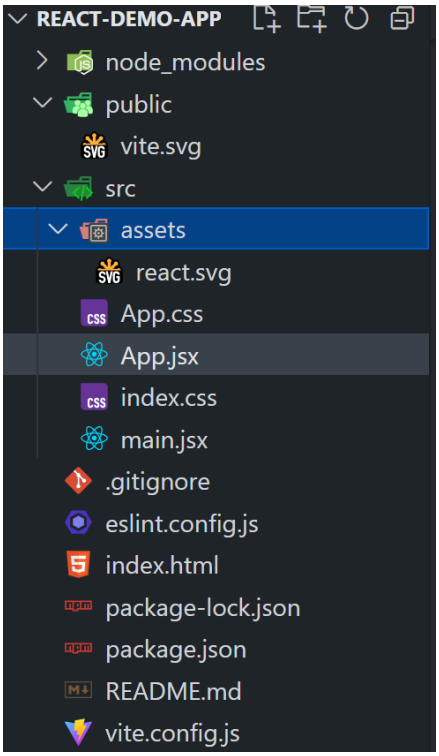
Step 12: - Now, Vite will execute run script to run the react application in dev environment and Vite will launch the react application on it's default port number 5173. We can visit to the application through web browser using

url: http://localhost:5173.



❖ React Folder Structure: -

When we create the React project using Vite tool, Vite add Project Structure Which includes some of the files and directories which are as below.



CJC
Complete Java Classes

1. **node_modules:** - this directory is generated after we execute `npm install` command. And it contains all the inbuild as well as third party libraries.
2. **Public: -** this directory contains the static content like image, videos ,audios which we may use on our index page
3. **Src : -** this directory includes all the source code of our react application, while developing the react application we will mostly interact with this directory.

a) **assets: -** assets directory also contains static files like image, videos and audio files, but this files are used in source code only.
b) **App.jsx:** it is the root component of our react application.
c) **App.css:** this file contains the external css for App component.
d) Index.css: it is the global CSS file and used for write common css code which will applied to app the React components.
e) **main.jsx: -** main.jsx is the important file in react application, this file get execute just after index.html, it contains the render method to render App component.
f) **.gitignore:** this is not react specific file, it related to version control.
g) **eslint.config.js: -** this file concerns to the code review and standards.
h) Index.html it is the only html page of our SPA.
i) **package.json : -** this file contains the project information and dependencies.
j) **package-lock.json:-** this file contains the information about nested dependencies.