React Guide

Building First React.js Application

1. How does React.js work ?
2. What are the advantages of using React.js ?
3. Who are use the React.js ?
4. Prerequisites / Set up the development environment
5. Set up the project
6. Explore the project structure
7. Create first component
8. Render component
9. Start the application

React.js is a popular open-source frontend JavaScript library for building attractive and reusable user interfaces based on components by Facebook Inc. It is maintained by Meta. This article provides a little introduction to React and guides you to build a very simple React.js application.

How does React.js work?

Rather of working directly with the browser's DOM (Document Object Model), React uses an in-memory virtual DOM. This virtual DOM is a minimal copy of the real DOM that React updates first when changes happen. It then detects what has changed by comparing the virtual DOM to the previous version. The process is made more efficient and ensures that only the sections that require updating are updated when updates are made to the real DOM. This method helps React in providing smooth, rapid user experiences.

What are the advantages of using React.js ?

- Reusable components: React allows creating reusable components, which can be used throughout the application. These reusable component increases readability and stability of the code.
- Faster debugging and rendering.
- Syntaxes are simple, because of that anyone can easy to learn.

- Strong community support: React has a large, active community with plenty of resources, libraries and tools. Because of that, developers can find solutions and getting supports quickly.

Who are use the React.js ?

- Facebook
- Instagram
- Netflix
- Uber
- Reddit
- Discord
- SoundCloud
- WhatsApp Web and more …

Prerequisites / Set up the development environment

Before we get started to develop our React application, the followings must be installed on your machine:

- Download and install Node.js is required. (check node -v)
- Install NPM (Node Package Manager): It comes bundled with Node.js (check npm -v)
- Open command prompt and type `npm install upper-case`
- Download and install Visual Studio Code Editor.

You can use an any other code editor (e.g. notepad, notepad++, sublime text etc.) that you're familiar with for the implementations. But, we're using VS Code editor for the implementations in this article.

Alternative Installation:

Set up the project

Create a new folder for the project in any desired location on your local machine. You can use any name for the project folder. I have named the project folder as first-react-app

Next, open the created folder using VS Code editor.

Now, open a new terminal in the VS Code editor and type the below command to create react app. I'm going to generate the React app template inside the `my-react-app` folder. You can replace `my-react-app` part with your desired project name in the command.

```
npx create-react-app my-react-app
```

The installation process will take few minutes. Don't be hurried! Wait until completed installation. At the end of the installation, it will display the "**Happy Hacking!"** message.

Navigate inside to the generated React project using below command.

```
cd my-react-app
```

Installations with Vite:

Run following commands

```
npm create vite@latest
npm install
```

Run the project by command:

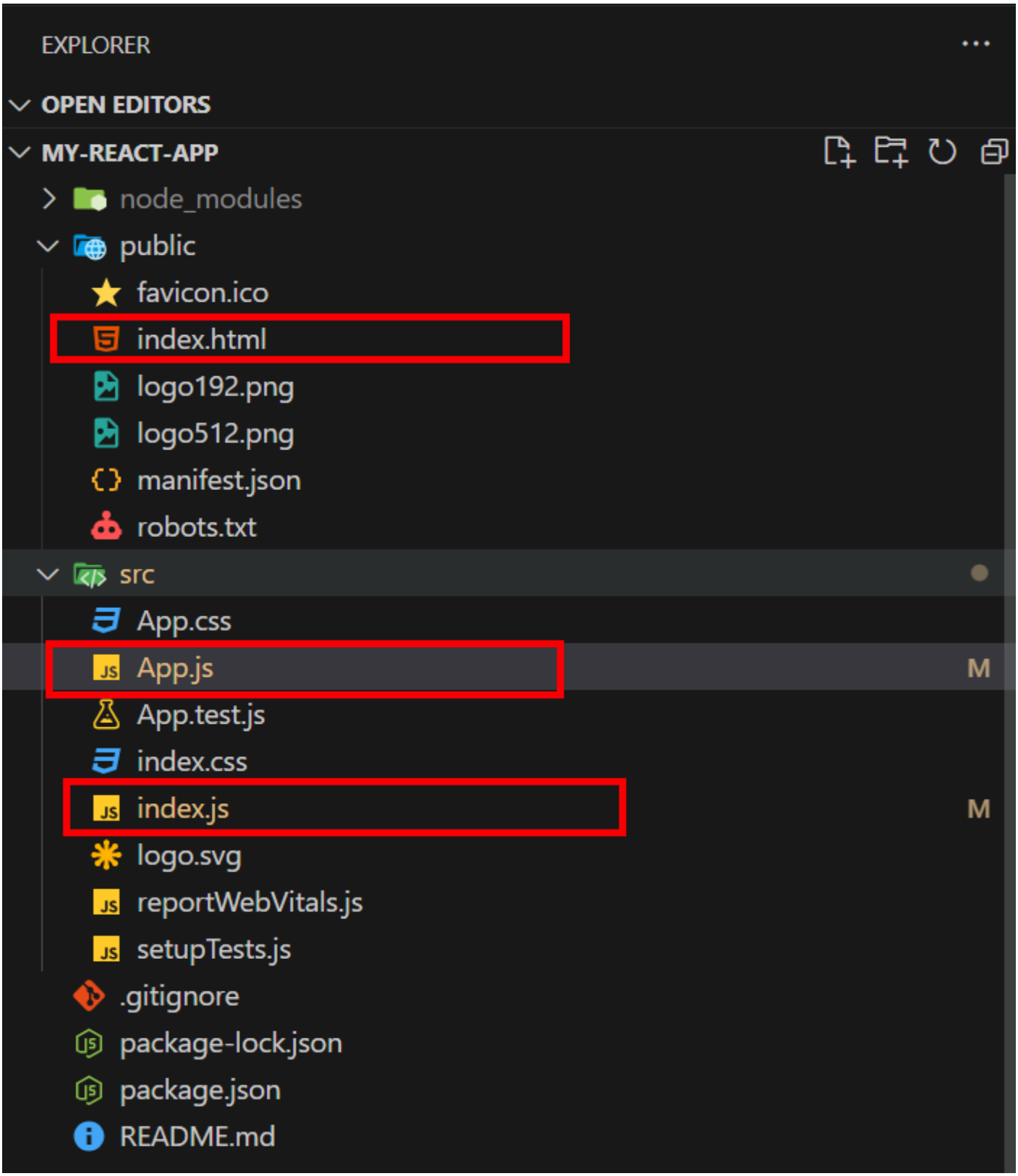
```
npm run dev
```

Explore the project structure

Explore the created files and folders by navigating to your project folder. The main files you will handle in this article are :

- **index.js :** This is where your application starts.
- **App.js :** The core component of your application is located in this file.

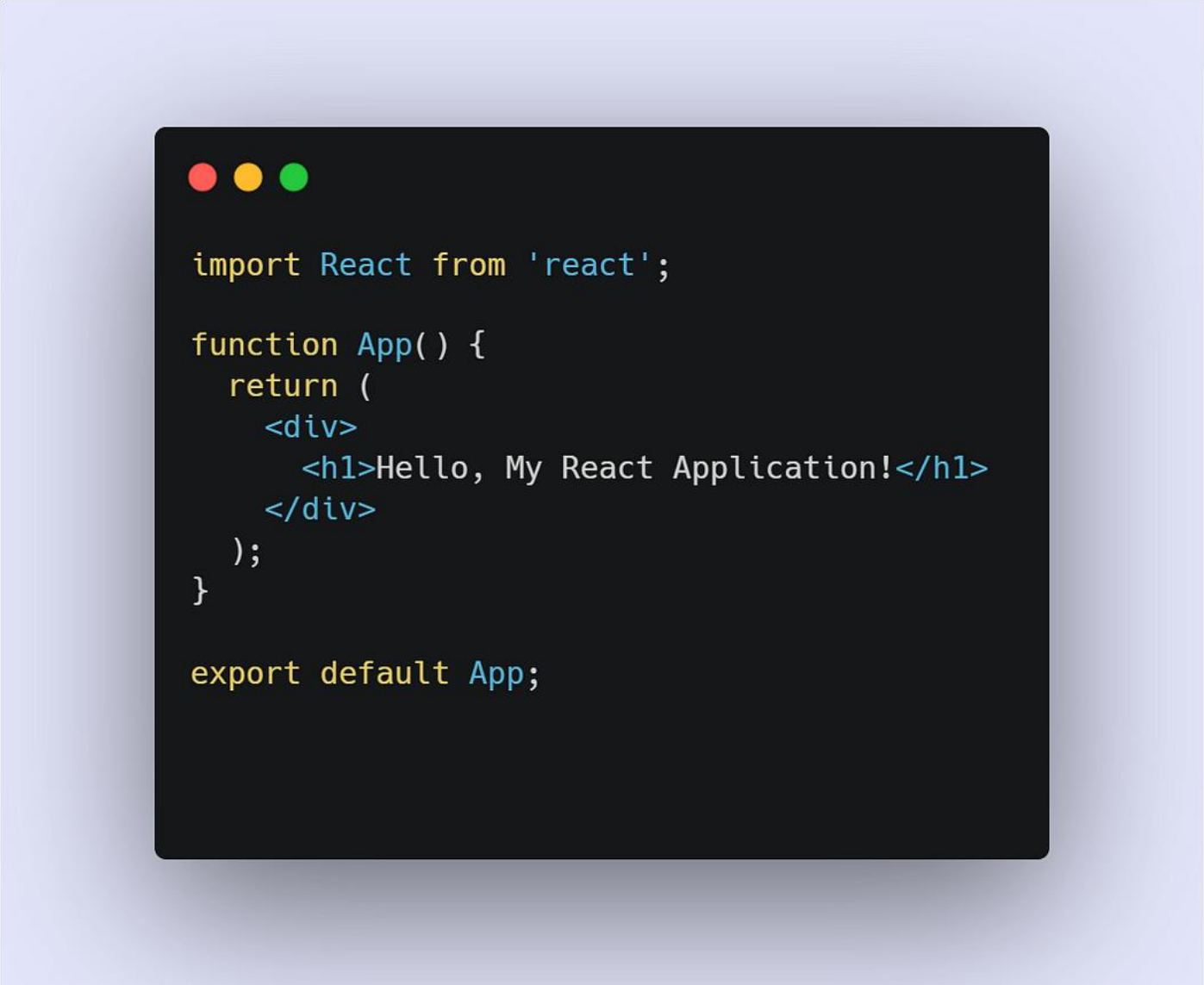
- **index.html :** Containing your mounted React application.



Project folder structure

Create first component

In created React application, open App.js file. Use the following code to replace the existing content:

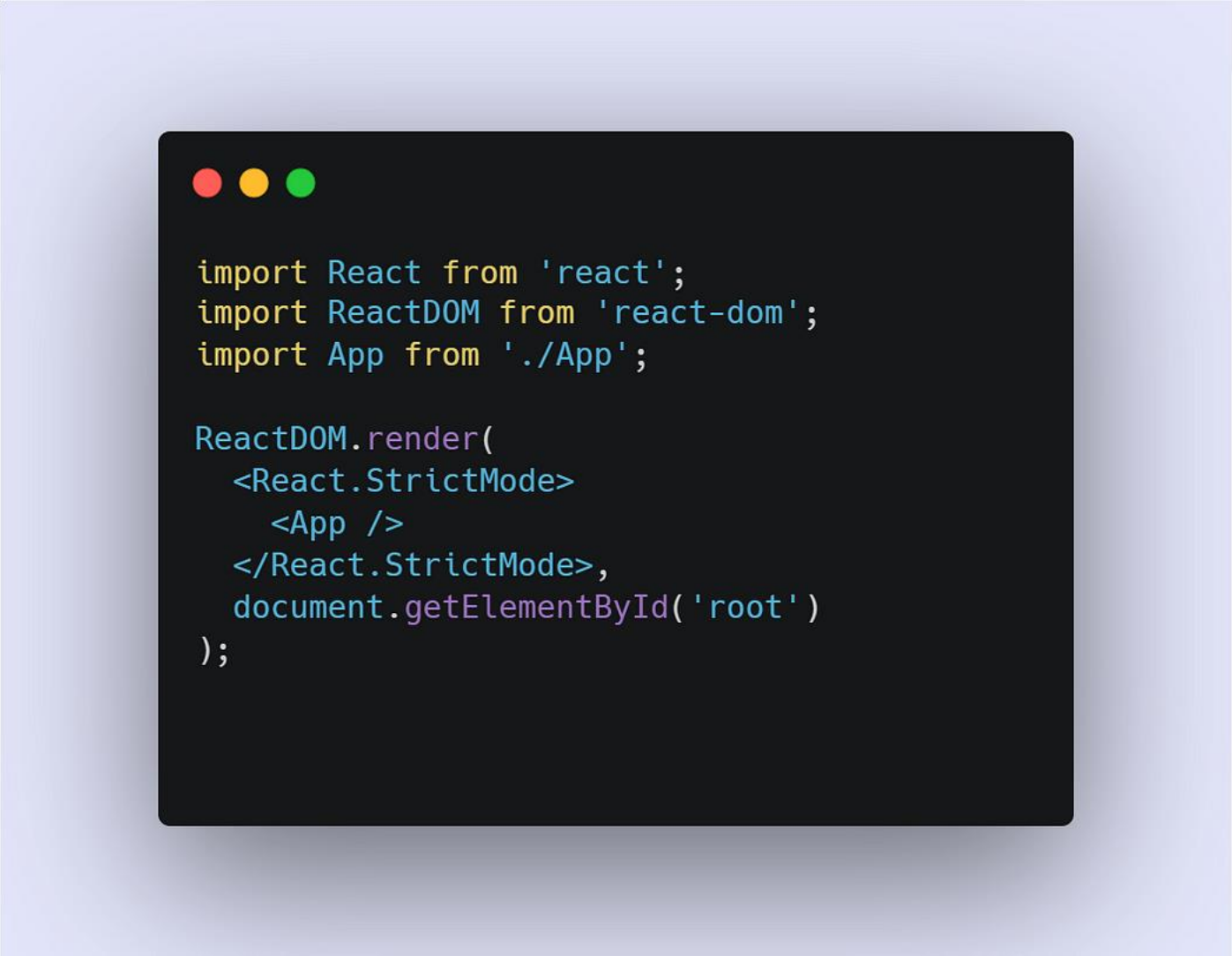


```
import React from 'react';

function App() {
  return (
    <div>
      <h1>Hello, My React Application!</h1>
    </div>
  );
}

export default App;
```

App.js file

Render component

To render the created component, open index.js file and update the existing content as shown in the below snapshot.



```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
```

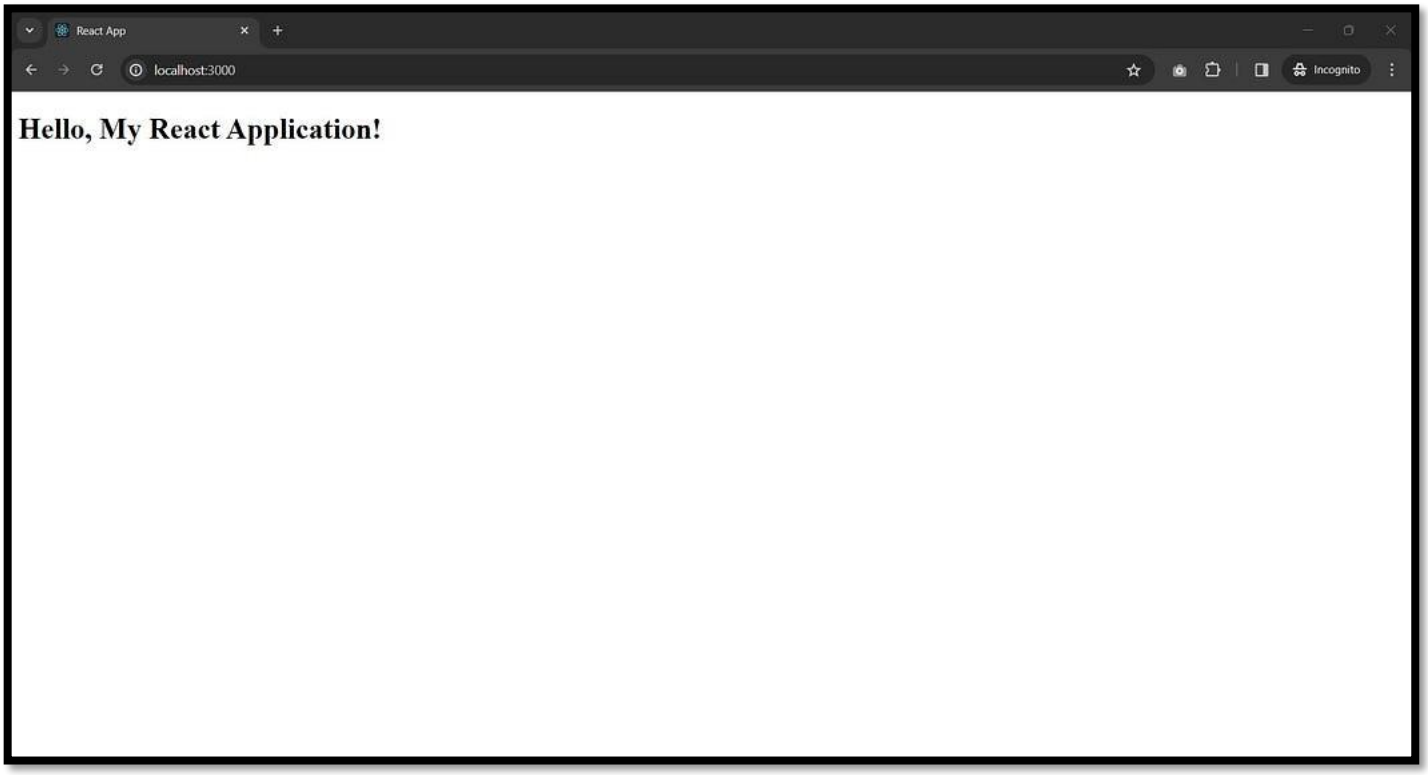
index.js file

Start the application

To launch the application, type the following command in the terminal of your VS Code editor;

```
npm start
```

After a few seconds, your default web browser will automatically open and display **"Hello, My React Application!"** heading at http://localhost:3000.



Expected final output.

Home Task: **:** Create a React-based Todo List application to help users to manage their daily tasks efficiently:

Develop a functional Todo List application with the following features:

1. **Add new tasks** (with input validation)
2. **Display all tasks** in an organized list
3. **Delete tasks** (individual item removal)