Module 5: React.js Webpack Assignment 06/01/2023
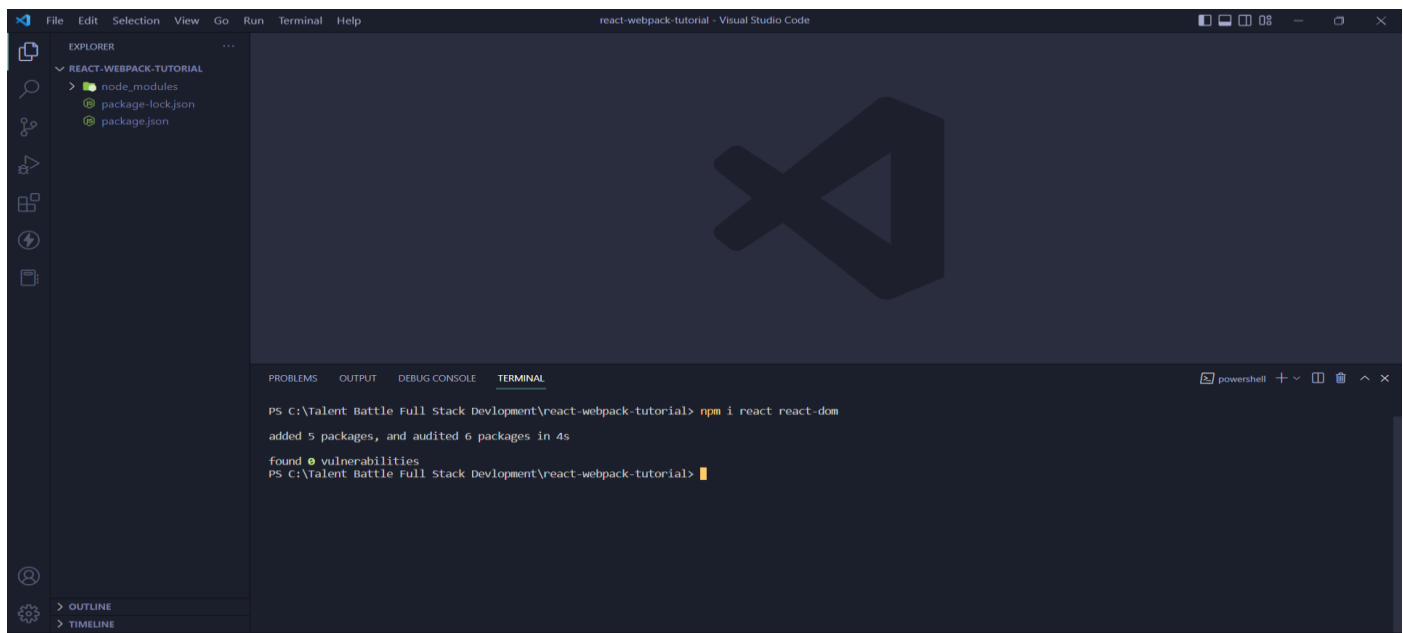
Step 1: npm init -y

Step 2: npm i react react-dom

Step 3: npm i -D @babel/core @babel/preset-env @babel/preset-react babel-loader css-loader html-webpack-plugin sass sass-loader style-loader url-loader webpack webpack-cli webpack-dev-server
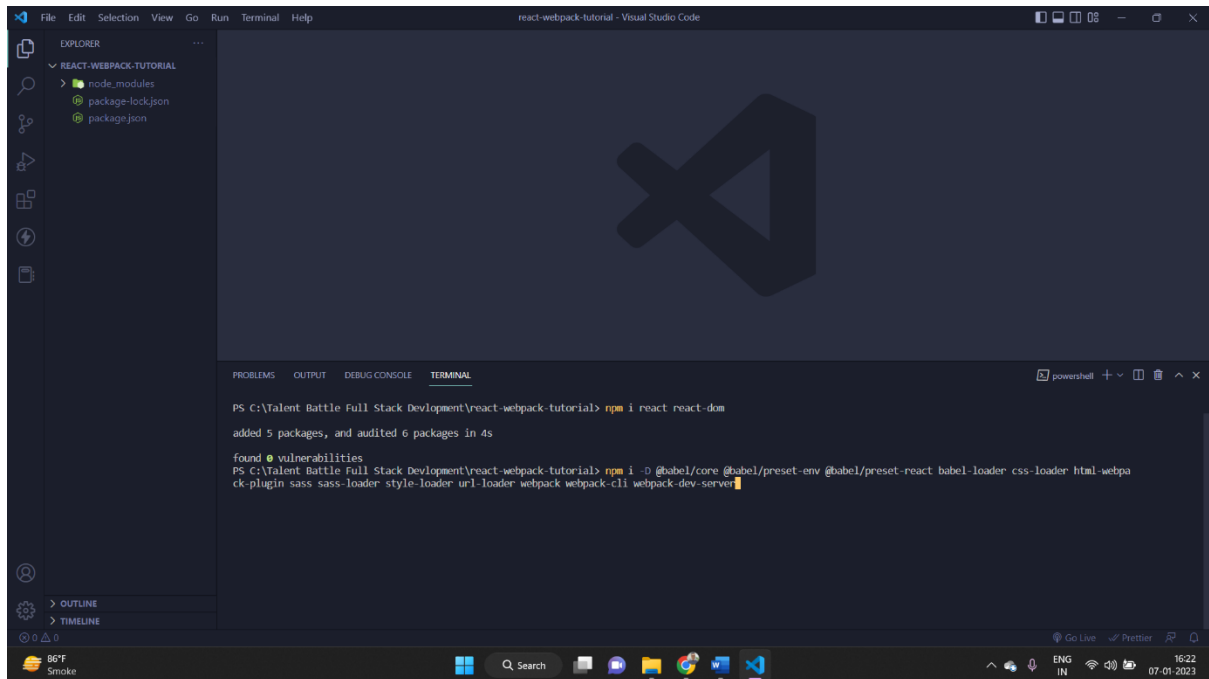
Step 4: Create. babelrc File in Root Directory

{

  "presets": ["@babel/preset-env", "@babel/preset-react"]

}
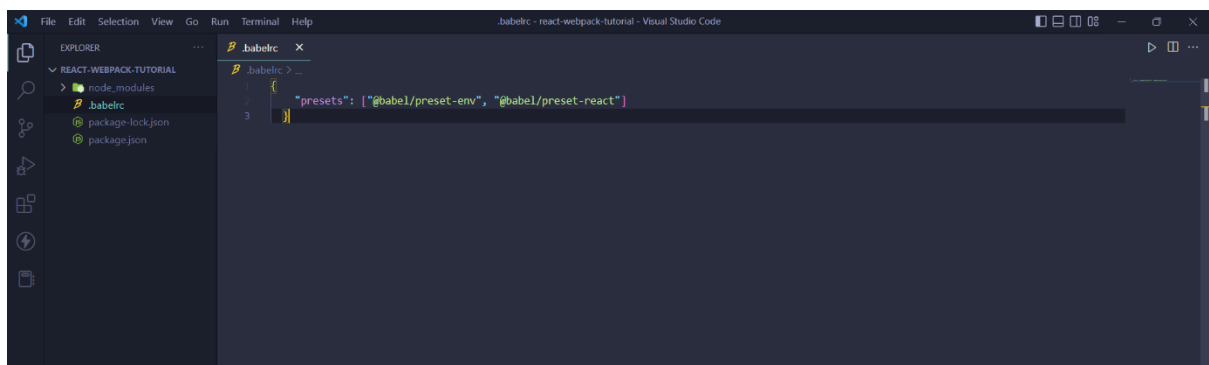
Step 5. Create a webpack.config.js file in root Directory

```js
const path = require("path");
const HtmlWebpackPlugin = require("html-webpack-plugin");

module.exports = {
  output: {
    path: path.join(__dirname, "/dist"), // the bundle output path
    filename: "bundle.js", // the name of the bundle
  },
  plugins: [
    new HtmlWebpackPlugin({
      template: "src/index.html", // to import index.html file inside index.js
    }),
  ],
  devServer: {
    port: 3030, // you can change the port
  },
  module: {
    rules: [
      {
        test: /\.(js|jsx)$/, // .js and .jsx files
        exclude: /node_modules/, // excluding the node_modules folder
        use: {
          loader: "babel-loader",
        },
      },
      {
        test: /\.(sa|sc|c)ss$/, // styles files
        use: ["style-loader", "css-loader", "sass-loader"],
      },
      {
        test: /\.(png|woff|woff2|eot|ttf|svg)$/, // to import images and fonts
        loader: "url-loader",
        options: { limit: false },
      },
    ],
  },
};
```
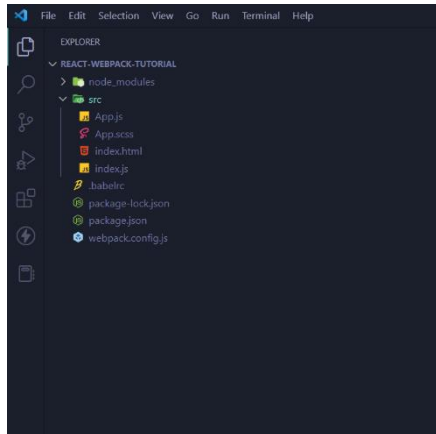
Step 6. Create an /src folder and create the following files inside it

```
|-- src
  |-- App.js
  |-- App.scss
  |-- index.html
  |-- index.js
```


Your Folder Structure Will Be Like This

Step 7: Insert Following Things Inside Corresponding Files

1. App.js

```jsx
import React from "react";
const App = () => {
  return(
    <div>
    <h1>
        Welcome To Talent Battle React Webpack Session
    </h1>
    <h2>
    After all – necessity is the mother of all invention…
    </h2>
    </div>
  );
};

export default App;
```

2. App.scss

```scss
//! Advantage Of SASS We Can Have Nesting Of Tags

body{
    h1 {
        color: blue;
        text-align: center;

    }

    h2{
        color: red;
        text-align: center;
    }
}
```

3. Index.html

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>React with Webpack</title>
  </head>
  <body>
    <div id="root"></div>
    <script src="bundle.js"></script>
  </body>
</html>
```

4. Index.js

```javascript
import React from "react";

import ReactDOM from "react-dom";
import App from "./App";
import "./App.scss";

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

Step 8: Edit & Add Following Lines In "scripts" inside Package.json

```json
{
  "name": "react-webpack-tutorial",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "serve": "webpack serve --mode development",
    "build": "webpack --mode production"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "react": "^18.2.0",
    "react-dom": "^18.2.0"
  },
  "devDependencies": {
    "@babel/core": "^7.20.12",
    "@babel/preset-env": "^7.20.2",
    "@babel/preset-react": "^7.18.6",
    "babel-loader": "^9.1.2",
    "css-loader": "^6.7.3",
    "html-webpack-plugin": "^5.5.0",
    "sass": "^1.57.1",
    "sass-loader": "^13.2.0",
    "style-loader": "^3.3.1",
    "url-loader": "^4.1.1",
    "webpack": "^5.75.0",
    "webpack-cli": "^5.0.1",
    "webpack-dev-server": "^4.11.1"
  }
}
```

Step 9: Run and modify your app

Run the app with npm run serve. Open your browser on http://localhost:3030/. Try to modify it and see the changes on the fly.

Step 10: Build the App

Run npm run build in the terminal. You will see the following output:

```
|-- dist
  |-- bundle.js
  |-- bundle.js.LICENSE.txt
  |-- index.html
```

Finally, open the index.html file and you should see your app.

THANKYOU AND HAVE A NICE DAY 😊

Ameya Belvalkar