## *Environment Setup for Building React Application*

- You need a **Package Manager**
  - Package Manager is a software tool used by web developers to download and install various libraries required for project. Also you can run an application, uninstall any library using this.
    - NPM [Node Package Manager]
    - Yarn
    - NuGet
    - Bower etc..
  - **Download and install "NPM"**
  - **You can get NPM from "Node.js"**
    https://nodejs.org/en/download/
  - **Check Version from Command Prompt**
    **C:\>** node -v
    **C:\>** npm -v
- You need an Editor
  - Normal text-editors are not preferred to develop apps. You can use any advance Code-Editor/IDE. Here a link is provided, in case you don't have any.
  - Download and install "**Visual Studio Code**" editor
    https://code.visualstudio.com/
  - Install the following extensions in Visual Studio Code
    - Live Server
    - Vscode-icons [optional]

## Using React in an Existing Web Application

- Open your PC locations
- Create a new folder for project “**C:\WebApp”**
- Open “WebApp” folder in “Visual Studio Code”
- Open Terminal in VS-Code
- Run the command

  **> npm init -y**
- This will generate “package.json”
- Add a new folder by name **“public”**
- Add following file into “public” folder
  - Index.html
  - Home.html

**Index.html**

```
<!DOCTYPE html>

<html>

   <head>

      <title>Index</title>

   </head>

   <body>

      <h2>Welcome to React JS</h2>

      <p>This is an ordinary HTML page [without any
React]</p>

      <p>React is interated into Home Page - Click Below</p>

      <a href="home.html">Home</a>

   </body>

</html>
```

- ***To Integrate React environment into a HTML page you need following libraries***

    - React **It is React Core Library**
    - React DOM **It is used for Virtual DOM manipulations**
    - Babel Standalone **It is a compiler used to compile react code written in "JSX"**

- You can use **CDN** [Content Distribution Network] for Libraries or you can download using NPM
- Get the CDN of all 3 libraries and Add to your Home Page.

**Home.html**

```
<!DOCTYPE html>

<html>

  <head>

    <title>React Page</title>

    <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">

    <link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">
    <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
    <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
    <script type="text/babel">
      ReactDOM.render(
        "React JS - Shopping Store",
        document.getElementById("container")
      )
    </script>
  </head>
  <body class="container-fluid">
    <h2 class="bg-primary text-white text-center p-2">
     <span class="bi bi-cart4"></span>  Amazon Shopping
    </h2>
    <div id="container"></div>
  </body>
</html>
```

# *Web Pack*

- It is a module bundler.
- It is responsible for bundling all modules required for your application.
- So that your application can run and can be deployed on server.
- Web Pack is a module bundler used for various types of application.

## Create a React application using Web Pack

- Create a new folder for project **“React-Shopping-App”**
- Open in VS-Code
- Open Terminal
- Run the command to initialize your “**React-Shopping-App**” as a web application.

  **> npm init -y**
- This will generate “package.json”
- Run the command to install “React core” and “Virtual-DOM” library

  **> npm install react react-dom**
- This will generate **“node_modules”** folder and will copy your library into folder.
- Run the following commands to install compiler, file loader and web pack

  **> npm install -D @babel/core  @babel/preset-env @babel/preset-react  babel-loader**

  **> npm install file-loader  css-loader sass  sass-loader**

```
> npm install webpack webpack-cli
```

**package.json**

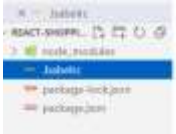
```
{
  "name": "React-Shopping-App",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "react": "^17.0.2",
    "react-dom": "^17.0.2"
  },
  "devDependencies": {
    "@babel/core": "^7.14.6",
    "@babel/preset-env": "^7.14.7",
    "@babel/preset-react": "^7.14.5",
    "babel-loader": "^8.2.2",
    "css-loader": "^5.2.7",
    "file-loader": "^6.2.0",
    "sass": "^1.35.2",
    "sass-loader": "^12.1.0",
    "webpack": "^5.44.0",
    "webpack-cli": "^4.7.2"
```

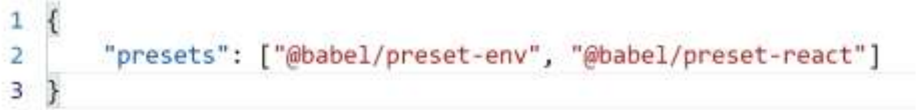
```
 }
}
```

- ***<u>You have to add babel configuration file</u>***

  - Add a new file into project by name **“.babelrc”**
  - Add the following code into “.babelrc” file

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





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

- **You have to add webpack configuration file**
  - Add a new file into project by name **“webpack.config.js”**

```
const path = require('path');


module.exports = {
   output : {
      path: path.join(__dirname, '/dist'),
  filename: 'index.bundle.js'
},
devServer: {
  port : 3010,
  watchContentBase: true
},
module : {
  rules : [
    {
      test: /\.(js|jsx)$/,
      exclude: /node_modules/,
      use : {
        loader: 'babel-loader'
      }
    },
    {
      test: /\.scss$/,
      use: [
        'css-loader',
        'sass-loader'
      ]
    }
```

```
            ]
        }
    }
```

- **Go to “package.json”**

```
"scripts": {
    "serve": "webpack serve --mode development",
    "build": "webpack --mode production"
  },
```

- **Add a new folder “src”**
- **Add following files into “src” folder**

***app.js***

```
import React from 'react';

export function App(){
   return(
      <div>
         <h2>React Application</h2>
         <p>App Designed with Webpack</p>
      </div>
   )
}
```

**Index.js**

```
require('file-loader?name=[name].[ext]!./index.html');
import React from 'react';
import ReactDOM from 'react-dom';
import { App } from './app';
```

```
ReactDOM.render(
    <App/>,
    document.getElementById("container")
)
```

**Index.html**

```
<!DOCTYPE html>
<html>
    <head>
        <title>React App</title>
    </head>
    <body>
        <div id="container"></div>
        <script src="index.bundle.js"></script>
    </body>
</html>
```

- **Now open terminal and run the command**

  ```
  > npm run serve
  ```

  [it requires server library – choose 'yes']
- **Your Application runs on**

  http://localhost:3010
- **Open any browser and request the given URL**

  http://localhost:3010