**REACT Start-Up**

* C:\Sujeet\React\_Workspace\IndecisionAPP>node --version
* C:\Sujeet\React\_Workspace\IndecisionAPP>npm --version
* C:\Sujeet\React\_Workspace\IndecisionAPP>npm init
* C:\Sujeet\React\_Workspace\IndecisionAPP>npm install -g yarn (optional)
* C:\Sujeet\React\_Workspace\IndecisionAPP>npm install -g live-server

**CDN Link for React and ReactDOM:**

<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>

**Babel:**

Babel is used to convert JSX code to plain JavaScript code it act as a JavaScript compiler. Babel uses some plugins called presets to compile your JSX code. We will use 2 of them.

**env preset:** This will give access to es6,es7 features, const, arrow function, spread operator etc.

**react preset:**

C:\Sujeet\React\_Workspace\IndecisionAPP>npm install -g babel-cli@6.24.1 C:\Sujeet\React\_Workspace\IndecisionAPP>npm install babel-preset-react@6.24.1 babel-preset-env@1.5.2

Watch the JSX Code changes in app.js and compile it down to JavaScript code and copy it into index.js from where live server is running the file for the output.

C:\Sujeet\React\_Workspace\IndecisionAPP>babel src/app.js --out-file=public/scripts/index.js --presets=env,react –watch

C:\Sujeet\React\_Workspace\IndecisionAPP>live-server public

**WEBPACK:**

WebPack is used to bundle our application.js file into bundle.js file which will be available in public folder from where your application will run. It will combine all the file that is being used in your file and compress it into a single file. Also it will run babel for compiling JSX code into JavaScript code. It help you to break your application into separate module.

C:\Sujeet\React\_Workspace\IndecisionAPP>npm install live-server

C:\Sujeet\React\_Workspace\IndecisionAPP>npm install babel-cli@6.24.1

You can use script to run long command in short-hand if you are not using webpack as below:

 "scripts": {

    "server" : "live-server public/",

    "babel-build" : "babel src/app.js --out-file=public/scripts/index.js --presets=env,react –watch"

  },

Using webpack you have to configure everything into a root level file “webpack.config.js” as shown below:

C:\Sujeet\React\_Workspace\IndecisionAPP>npm install [webpack@3.1.0](mailto:webpack@3.1.0)

C:\Sujeet\React\_Workspace\IndecisionAPP>npm install babel-core@6.25.0 babel-loader@7.1.1 react@18 react-dom@18

babel-cli allow you to run babel from command line and babel-core allow you to run babel from tools like webpack

babel-loader is a webpack plugin it will teach webpack how to run babel when webpack sees certain files

As we remove global module for react and react-dom we will install it locally by running the below command :

**Webpack.config.js:**

const path = require('path');

module.exports = {

    entry : './src/app.js',

    output : {

        path : path.join(\_\_dirname, 'public'),

        filename : 'bundle.js'

    },

    module : {

        rules : [{

            loader : 'babel-loader',

            test : /\.js$/,

            exclude : /node\_modules/

        }]

    }

}

**Index.html:**

<html>

    <title> Indecision App</title>

    <body>

        <div id="appRoot"></div>

        <script src="./bundle.js"></script>

    </body>

</html>

**package.json:**

"scripts": {

    "webpack-build" : "webpack --watch",

  },

.**babelrc:**

{

    "presets": ["env", "react"]

}

src/app.js:

import React from 'react';

import ReactDOM from 'react-dom';

console.log("running !!!!")

ReactDOM.render(<p>Hello</p>, document.getElementById("appRoot"))