|  |  |
| --- | --- |
| Roll No. | A-75 |
| Student Name | Prasad Kashid |
| Subject | Internet Programming |
| Date Of Experiment |  |
| Date Of Submission |  |

**SUBJECT: INTERNET PROGRAMMING EXPERIMENT NO. 8**

**AIM:** Bundling React app using Webpack.

**OBJECTIVE:**

To orient students to Node.js for developing frontend application. To orient students to Node.js for developing backend applications.

**LAB OUTCOMES:** LO5 & LO6

**LO5:** Construct front end applications using React

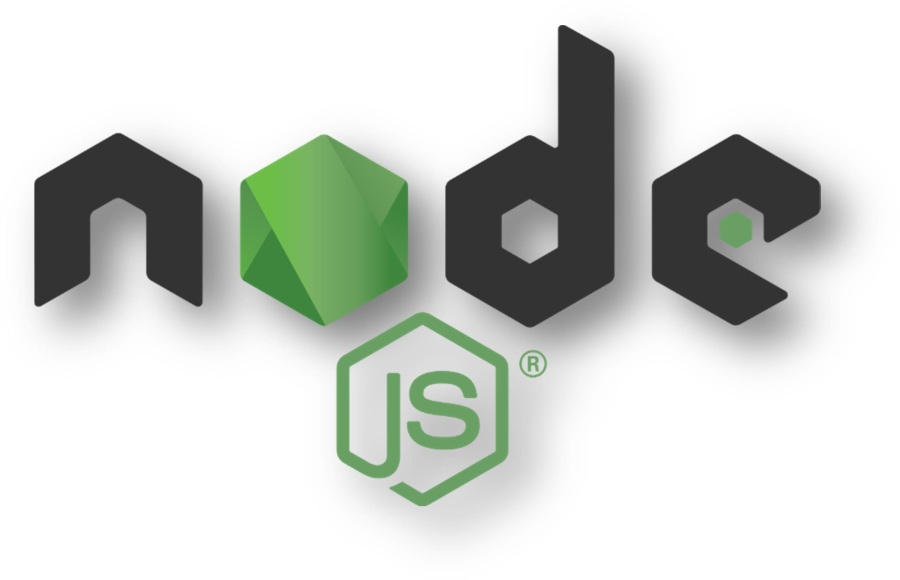
**LO6:** Construct back end applications using Node.js/Express

**THEORY:**

***Node.js***

* Node.js is actually not a framework or a library, but a runtime environment, based on Chrome’s V8 JavaScript engine.
* It is used for server-side programming, and primarily deployed for non- blocking, event-driven servers, such as traditional web sites and back-end API services, but was originally designed with real-time, push-based architectures in mind. Every browser has its own version of a JS engine, and node.
* Node.js provides a minimal interface to build our applications. It provides us the tools that are required to build our app. It is flexible as there are numerous modules available on npm, which can be directly plugged into Node.js.
* As an open-source project, Node.js was sponsored by Joyent, a cloud computing and hosting solutions provider. The company invested in a number of other technologies, such as Ruby on Rails framework, and provided hosting services to Twitter and LinkedIn. The latter also became one of the first

companies to use Node.js for its mobile application backend. The technology was later adopted by a number of technology leaders, such as Uber, eBay, Walmart, and Netflix, to name a few.



## Node.js IDEs:

Almost any popular code editor has support and plugins for JavaScript and Node.js, so it only matters how you customize your IDE to your coding needs. But, many developers highly praise special tools from VS Code, Brackets, Atom, and WebStorm.

## Advantages of Node.js

* Robust technology stack.
* Fast-processing and event-based model.
* Scalable technology for microservices.
* Rich ecosystem.
* Strong corporate support.
* Seamless JSON support.
* Performance bottlenecks with heavy computation tasks.
* Callback hell issue.

## Webpack

At its core, webpack is a static module bundler for modern JavaScript applications. When webpack processes your application, it internally builds a dependency graph from one or more entry points and then combines every module your project needs into one or more bundles, which are static assets to serve your content from. Webpack 5 runs on Node.js version 10.13.0+.



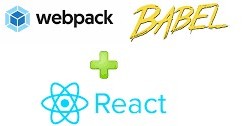
## Babel:

Babel is a free and open-source JavaScript transcompiler that is mainly used to convert ECMAScript 2015+ (ES6+) code into a backwards compatible version of JavaScript that can be run by older JavaScript engines. Babel is a popular tool for using the newest features of the JavaScript programming language.

Developers can use new JavaScript language features by using Babel to convert their source code into versions of JavaScript that Web browsers are able to process. The core version of Babel was downloaded 5 million times a month as of 2016, increasing to 16 million times per week as of 2019.

Babel plugins are used to transform syntax that is not widely supported into a backwards-compatible version. For example, arrow functions, which are specified in ES6, are converted into regular function declarations. Non-standard JavaScript syntax such as JSX can also be transformed.

Babel can automatically inject polyfills provided by core-js for support features that are missing entirely from JavaScript environments. For example, static methods like Array.from and built-ins like Promise are only available in ES6+, but they can be used in older environments if core-js is used.



**OUTCOMES:**

**CODE:**

This is a Node.js experiment. It uses NPM to manage its dependencies. You need to create a new project directory and initialize the node app.

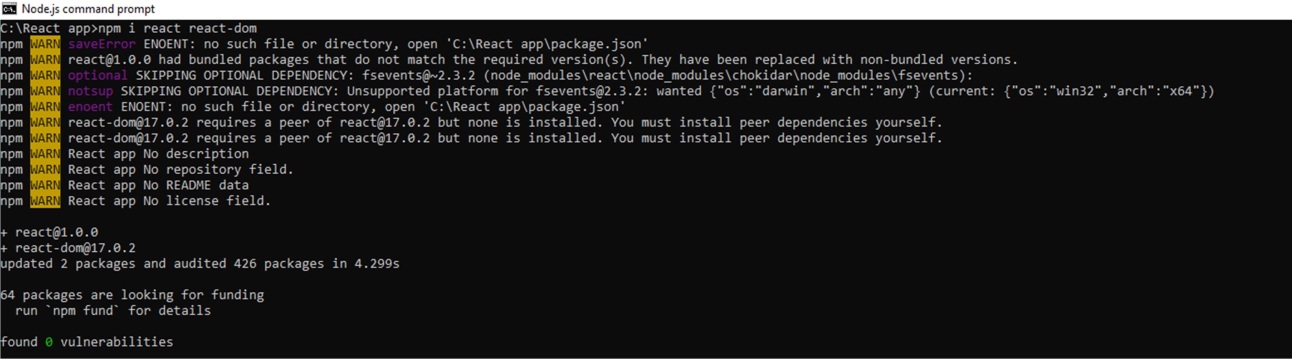
**Create package.json file:**

Now we have created a directory react app using mkdir command , by using npm init

-y we have skipped all the questions asked by npm to initialize the package.json file to default.

# Install react and react-dom:

***npm i react react-dom*** command installs react & react-dom.



# Install Babel and the required presets and plugins:

@babel/preset-react is preset for react,

@babel/preset-env is a smart preset that allows you to use the latest JavaScript without needing to micromanage which syntax transforms are needed by your target environment(s).

@babel/core contains the core functionality of Babel.

babel-loader will be used by webpack to transpile Modern JS into the JS code that browsers can understand.

Since all browsers don’t understand javascript’s static class properties feature @babel/plugin-proposal-class-properties plugin transforms static class properties as well as properties declared with the property initializer syntax.

# Create a babel config file: .barbelrc into root directory /react app/..

Here we tell babel to use @babel/preset-env target the last few versions of browsers and support for them. This will ensure that when the browser is updated it will stop transpiling of the old browser version and will transpile for the new one.

modules: false means hey babel! don’t do anything with the modules let webpack handle it. We also tell webpack to use @babel/preset-react for React and @babel/plugin-proposal-class-properties to transform static class properties

# Install Webpack and Webpack Dev Server:

**Create directories and files for the App:**

Create directories called src and public .Create our HTML file public/index.htm , entry file src/index.js and a component file src/App.js inside of it.

# Set up Webpack configuration file webpack.config.js



Here html-webpack-plugin will use your custom index.html that will be rendered

by webpack-dev-server.

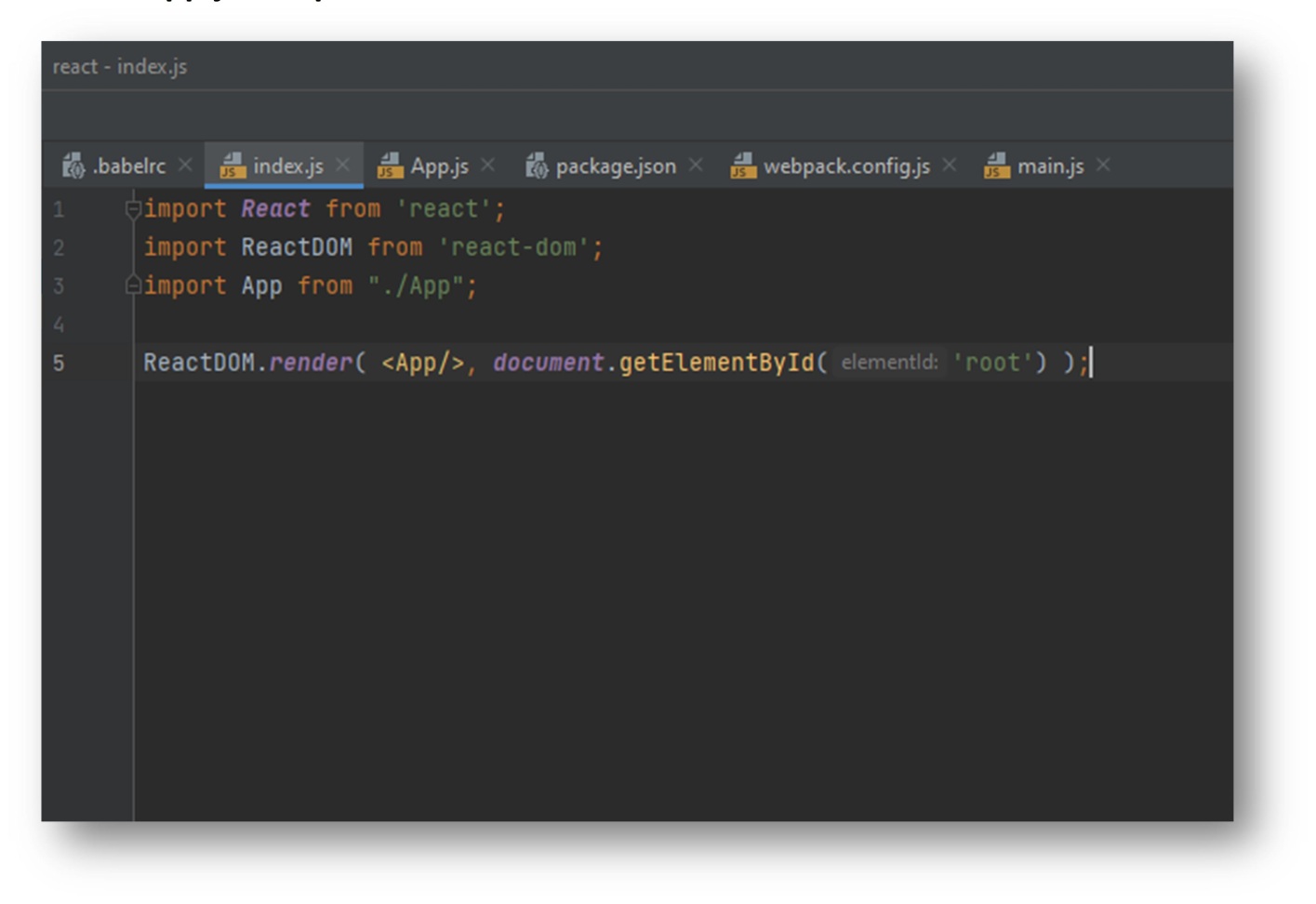
Please note that if you don’t pass any param in new HTMLWebpackPlugin() , then the html-webpack-plugin plugin will generate an HTML5 file for you that includes all your webpack bundles in the body using script tags.

Also add the style loader, css loader and file-loader for styles and images. As webpack understands javascript so we need to convert the styles and images in javascript using these loaders.

# Create a React Component src/App.js:

Create a class inside src/App.js and export it.

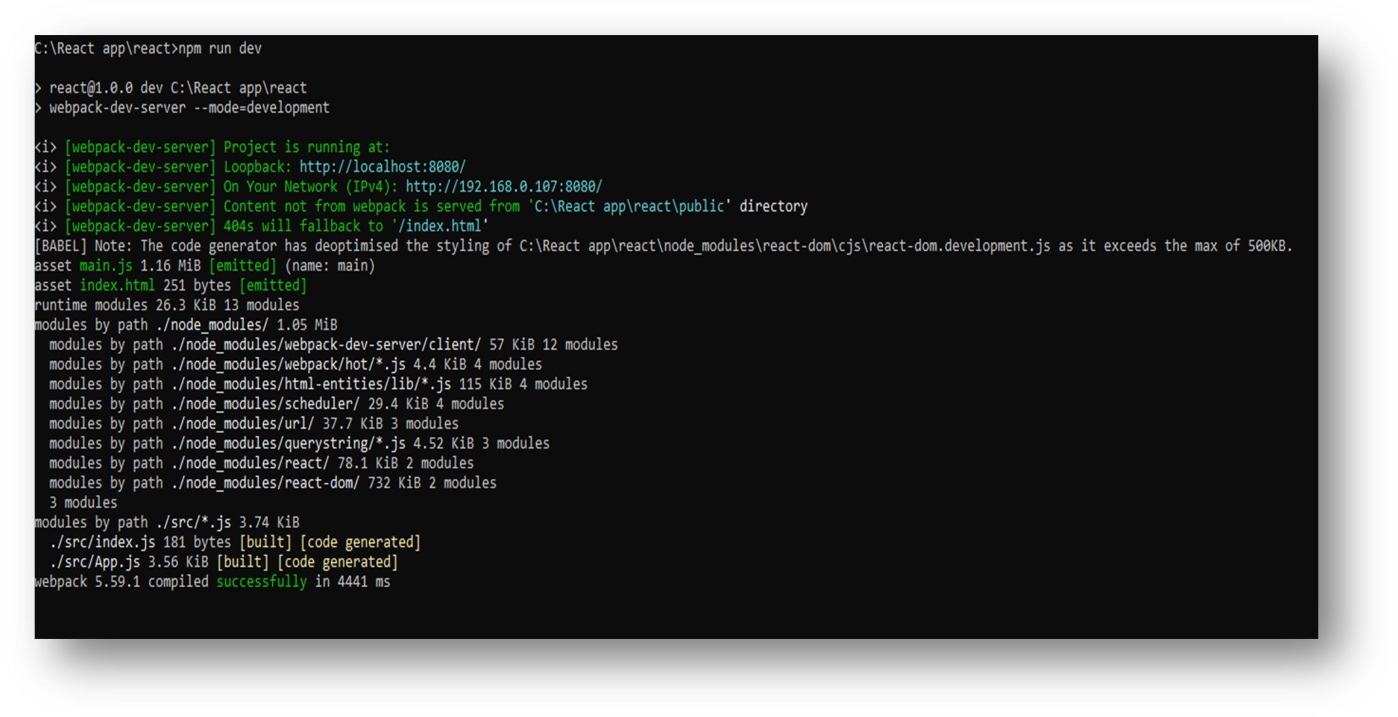
# Create a div#root inside public/index.html:

**Insert App.js component into the DOM:**

Now, insert the App.js component into div with the id root that exists public/index.html file

# Add scripts in the package.json:

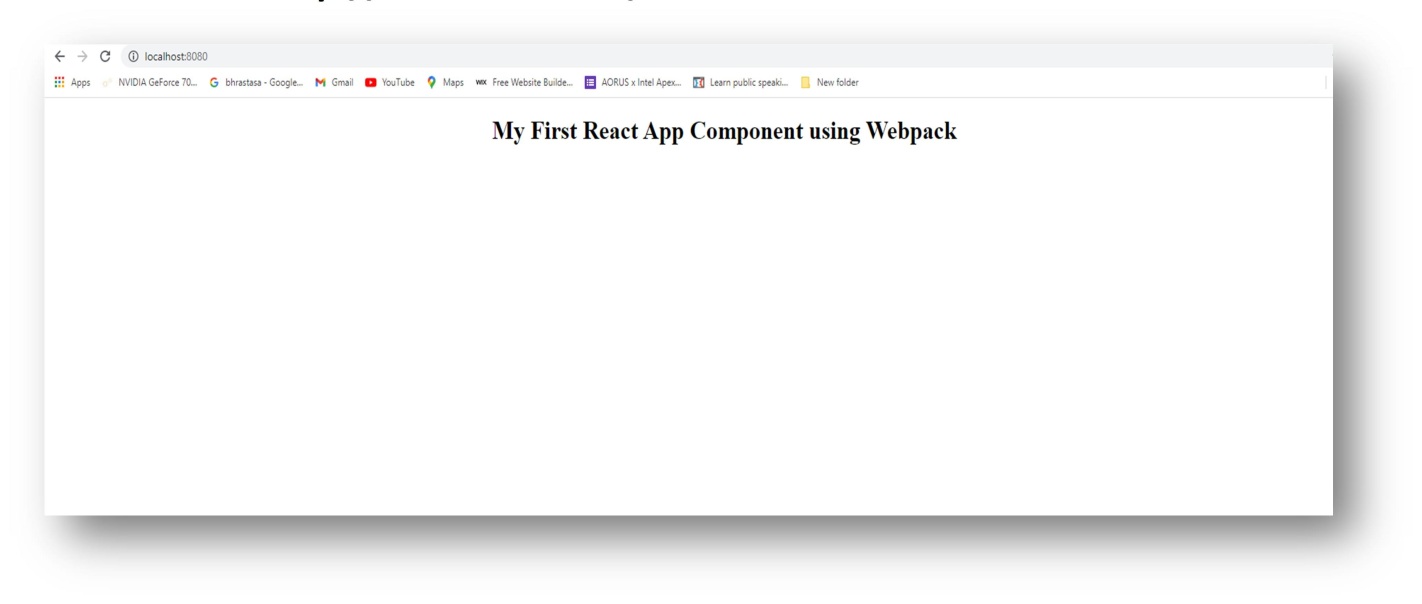
**Now running the webpack dev server:**



Command: “npm run dev” , as “dev” is used in the scripts to run webpack server in development mode, which can also be used to run in production mode using command: npm run prod.

Now the server is running on **http://localhost:8080/** in browser.

**OUTPUT:**

***In Browser @http://localhost:8080/***

**CONCLUSION:** In the above experiment, we studied about how to create React App using Webpack.