# MINI PROJECT (2020-21)

“**E-commerce site**-**IndiaZon”**

**FINAL-PROJECT-REPORT**



**Institute of Engineering & Technology**

**Submitted By: Submitted To:**

Name : Kritarth Sharma Faculty Name: Mr. Akash Choudhary

Uid:181500335

Name : Priyanka Nigam

Uid:181500509

**Contents**

|  |  |
| --- | --- |
| **Declaration** | **4** |
| **1. Introduction** | **6** |
| 1.1 project implementation | **7** |
| 1.2 use of the project |  |
|  |  |
| **2. Data flow diagram** | **9** |
| **3. ER-diagram** | **14** |
|  |  |
| **5. Screeshots** | **25** |
| **6. What is React** | **30** |
| **7. Firewall authentication** | **44** |

|  |  |
| --- | --- |
| 4**. Creating products** | **47** |
| **5. Advantages of Firebase** | **48** |
| **6. Conclusion** | **49** |
| **7. References** | **50** |

**DECLARATION**

We declare that the project report is based on our own work carried out during the course of our study under the supervision of Mr. Akash Chaudhary. We assert the statements made and conclusions drawn are an outcome of my research work. The work contained in the report is original and has been done by me under the general supervision of my supervisor. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad. We have followed the guidelines provided by the university in writing the synopsis. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

**ACKNOWLEDGEMENT**

Our sincere gratitude and thanks towards our project guide Mr. Akash Kumar Chaudhary. It was only with his backing and support that We could start the project. He provided us all sorts of help and corrected us if ever seemed to make mistakes. We have no such words to express my gratitude. We acknowledge my dearest parents for being such a nice source of encouragement and moral support that helped me tremendously in this aspect. We also declare to the best of my knowledge and belief that the Project Work has not been submitted anywhere else.

**INTRODUCTION**

Ecommerce, also known as electronic commerce or internet commerce, refers to the buying and selling of goods or services using the internet, and the transfer of money and data to execute these transactions. Ecommerce is often used to refer to the sale of physical products online, but it can also describe any kind of commercial transaction that is facilitated through the internet.

Whereas e-business refers to all aspects of operating an online business, ecommerce refers specifically to the transaction of goods and services.

The history of ecommerce begins with the first ever online sale: on the August 11, 1994 a man sold a CD by the band Sting to his friend through his website NetMarket, an American retail platform. This is the first example of a consumer purchasing a product from a business through the World Wide Web—or “ecommerce” as we commonly know it today.

Since then, ecommerce has evolved to make products easier to discover and purchase through online retailers and marketplaces. Independent freelancers, small businesses, and large corporations have all benefited from ecommerce, which enables them to sell their goods and services at a scale that was not possible with traditional offline retail.

Global retail ecommerce sales are projected to reach $27 trillion by 2020.

The main theme of the project is to build a social media website where the user can do the following things:

* Register if new user
* Login
* View various items
* Add to cart
* Purchase items

**Project Implementation:**

The project will have the following modules:

* Post module: All the functionalities related to posts will be in this module.
* Cart module: Manages operations related to cart.
* User module: Manages the users.
* Purchase module: All the functionalities related to purchase such as insert, view, delete will be in this module.

The project should provide the functionality for generating the report related to cart, users, purchase, and statistics of visitors.

Admin of the site should have control over:

* Designing and developing the website.
* Creating and updating the database
* Inserting, viewing, updating, deleting the data in the database.

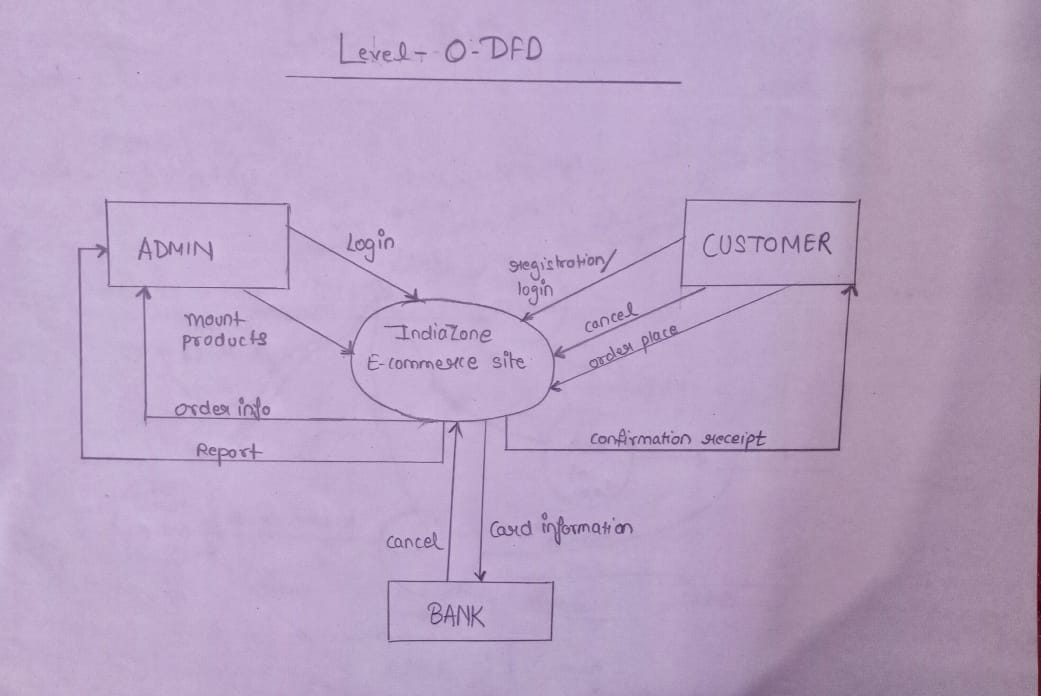
**USE OF THE PROJECT**

The online marketplace is a good platform for you to expand your business. We are going to explain what kind of advantages there are by sharing what we know about online selling. In brief, these are the plus points we will talk about.  
  
1. Faster buying process  
2. Store and product listing creation  
3. Cost reduction  
4. Affordable advertising and marketing  
5. Flexibility for customers  
6. No reach limitations  
7. Product and price comparison  
8. Faster response to buyer/market demands  
9. Several payment modes

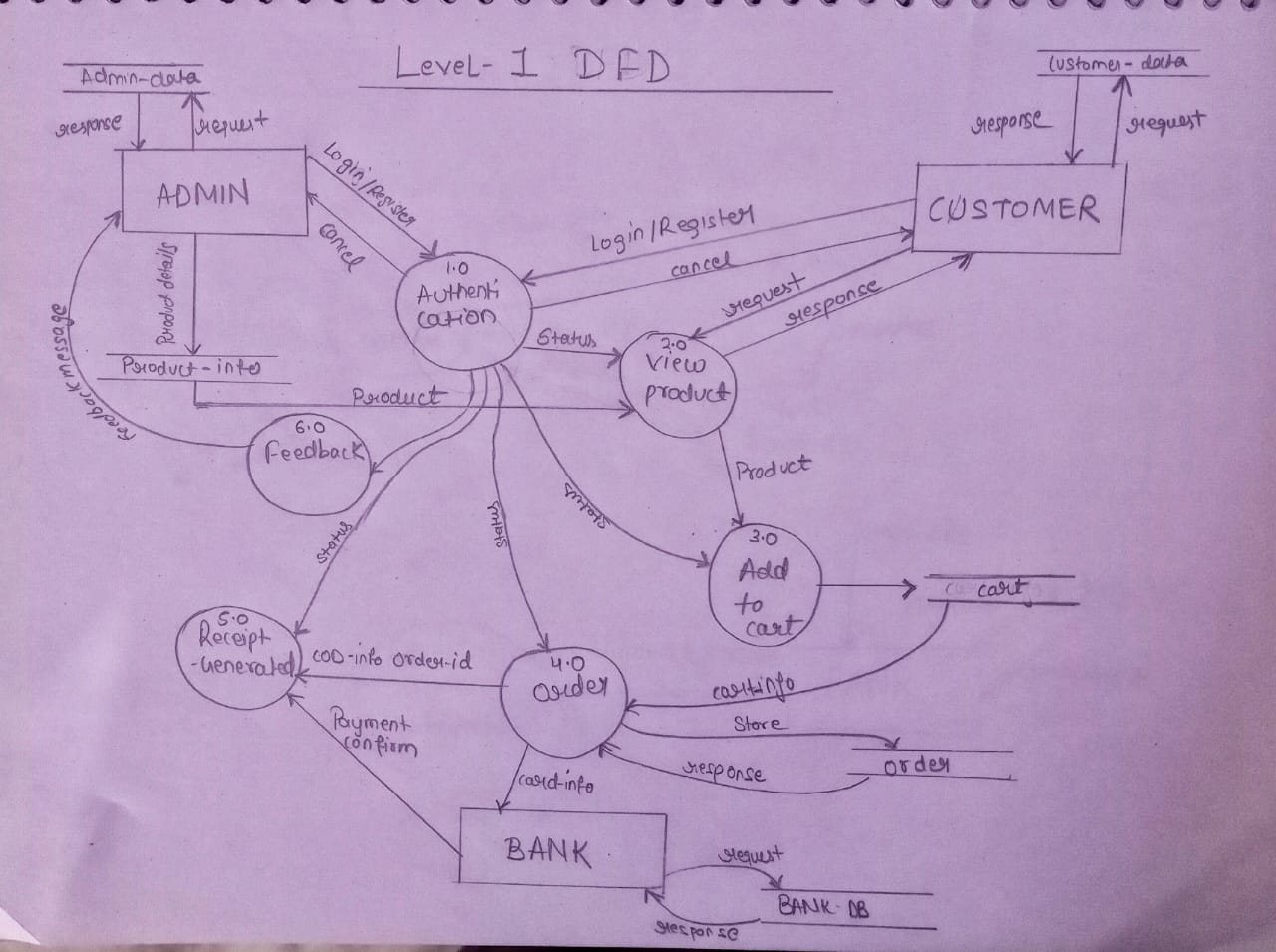
**Data flow Diagram**

A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.

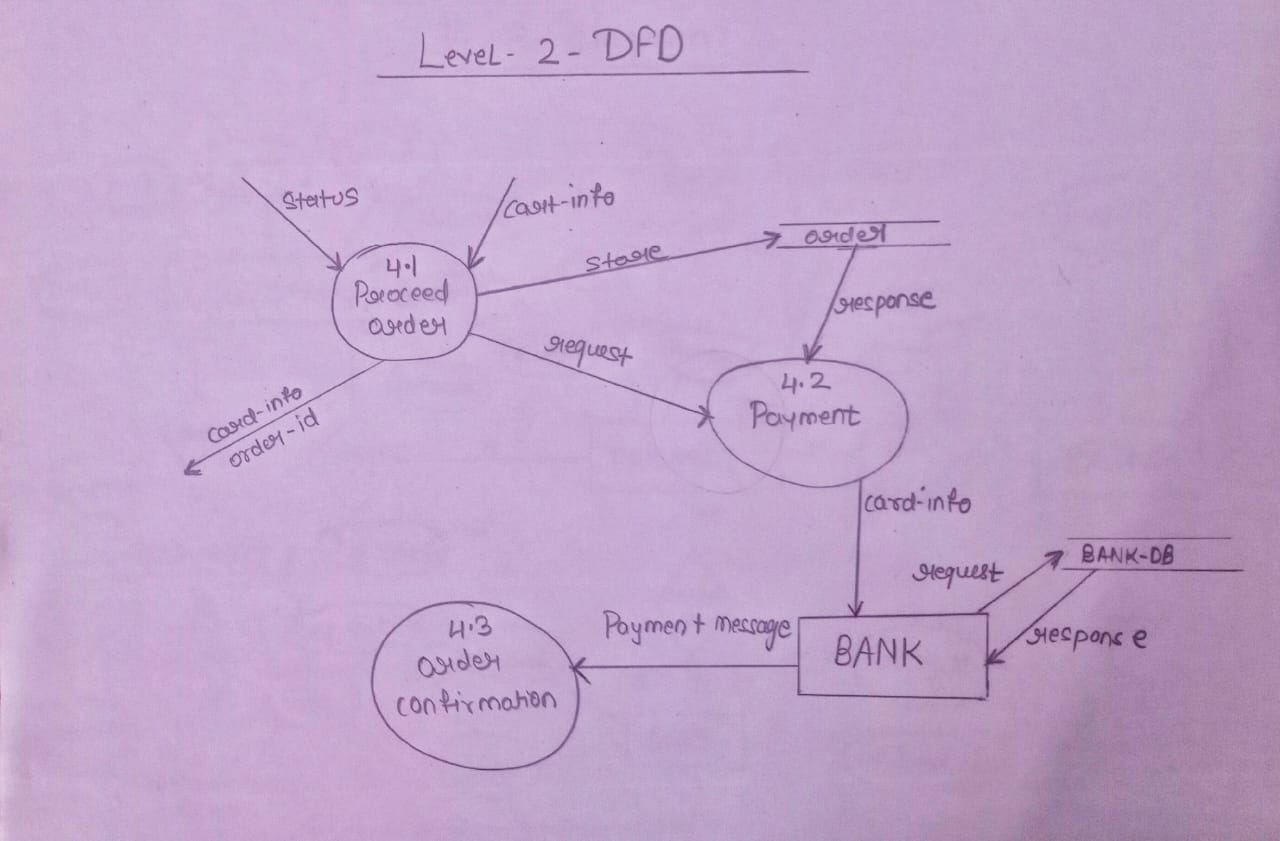
0 Level DFD: -DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analyzed or modeled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.



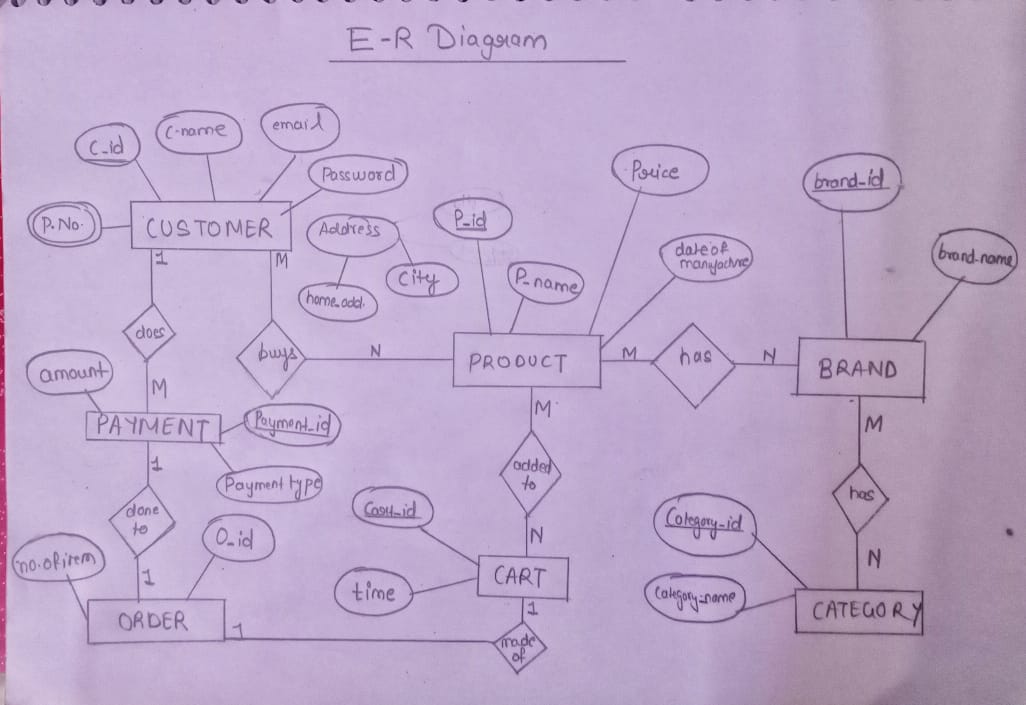
**1 level DFD**: - DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub processes.



**2 level DFD**: - DFD Level 2 provides a deatails of the process presented in the level-1 DFD in expanded form.



**ER-Diagram**

****

We have created a react app named as client where we write all the code of front-end Which is visible to the client side on our website “IndiaZon”

App.js

import React from "react";

import { Switch, Route } from "react-router-dom";

import Login from "./pages/auth/Login";

import Register from "./pages/auth/Register";

import Home from "./pages/Home";

import Header from "./components/nav/Header";

const App = () => {

return (

<>

<Header />

<Switch>

<Route exact path="/" component={Home} />

<Route exact path="/login" component={Login} />

<Route exact path="/register" component={Register} />

</Switch>

</>

);

};

export default App;

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<meta name="theme-color" content="#000000" />

<meta name="description" content="Web site created using create-react-app" />

<link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

<link rel="stylesheet"

href="https://unpkg.com/bootstrap-material-design@4.1.1/dist/css/bootstrap-material-design.min.css"

integrity="sha384-wXznGJNEXNG1NFsbm0ugrLFMQPWswR3lds2VeinahP8N0zJw9VWSopbjv2x7WCvX" crossorigin="anonymous">

<link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

<link rel="stylesheet"

href="https://unpkg.com/bootstrap-material-design@4.1.1/dist/css/bootstrap-material-design.min.css"

integrity="sha384-wXznGJNEXNG1NFsbm0ugrLFMQPWswR3lds2VeinahP8N0zJw9VWSopbjv2x7WCvX" crossorigin="anonymous">

<title>React App</title>

</head>

<body>

<noscript>You need to enable JavaScript to run this app.</noscript>

<div id="root"></div>

</body></html>

Login.js

import React from "react";

const Login = () => {

return (

<div>

<p>Login</p>

</div>

);

};

export default Login;

Register.js

import React from "react";

const Register = () => {

return (

<div>

<p>Register</p>

</div>

);

};

export default Register;

Home.js

import React from "react";

const Home = () => (

<div>

<p>react home</p>

</div>

);

export default Home;

Header.js

import React, { useState } from "react";

import { Menu } from "antd";

import {

AppstoreOutlined,

SettingOutlined,

UserOutlined,

UserAddOutlined,

} from "@ant-design/icons";

import { Link } from "react-router-dom";

const { SubMenu, Item } = Menu;

const Header = () => {

const [current, setCurrent] = useState("home");

const handleClick = (e) => {

// console.log(e.key);

setCurrent(e.key);

};

return (

<Menu onClick={handleClick} selectedKeys={[current]} mode="horizontal">

<Item key="home" icon={<AppstoreOutlined />}>

<Link to="/">Home</Link>

</Item>

<Item key="register" icon={<UserAddOutlined />} className="float-right">

<Link to="/register">Register</Link>

</Item>

<Item key="login" icon={<UserOutlined />} className="float-right">

<Link to="/login">Login</Link>

</Item>

<SubMenu icon={<SettingOutlined />} title="Username">

<Item key="setting:1">Option 1</Item>

<Item key="setting:2">Option 2</Item>

</SubMenu>

</Menu>

);

};

export default Header;

Index.css

body {

margin: 0;

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

code {

font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

monospace;

}

Index.js

import React from "react";

import ReactDOM from "react-dom";

import "./index.css";

import App from "./App";

import reportWebVitals from "./reportWebVitals";

import { BrowserRouter } from "react-router-dom";

import "antd/dist/antd.css";

ReactDOM.render(

<React.StrictMode>

<BrowserRouter>

<App />

</BrowserRouter>

</React.StrictMode>,

document.getElementById("root")

);

reportWebVitals();

reportWebVitals.js

const reportWebVitals = onPerfEntry => {

if (onPerfEntry && onPerfEntry instanceof Function) {

import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {

getCLS(onPerfEntry);

getFID(onPerfEntry);

getFCP(onPerfEntry);

getLCP(onPerfEntry);

getTTFB(onPerfEntry);

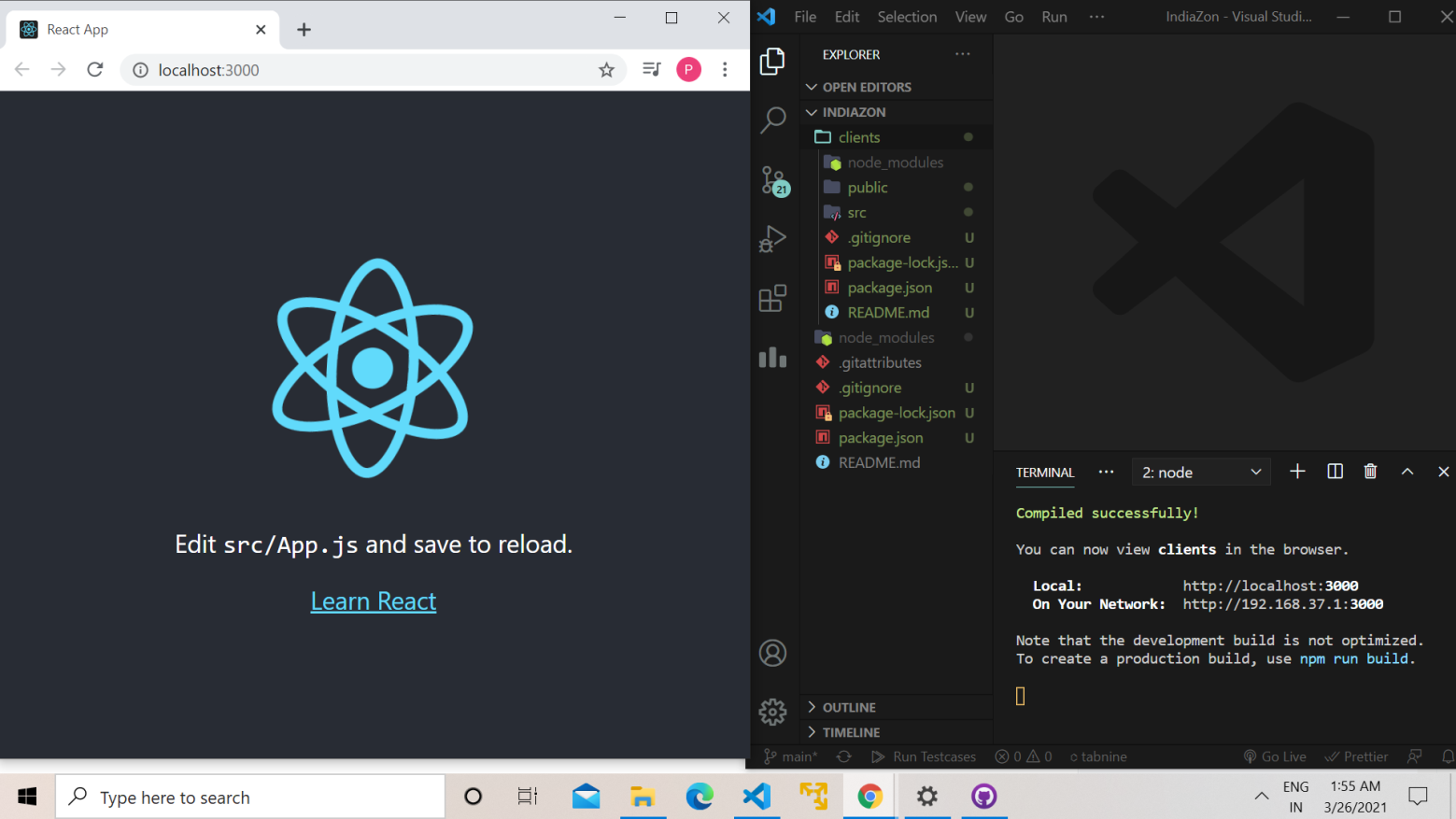
});

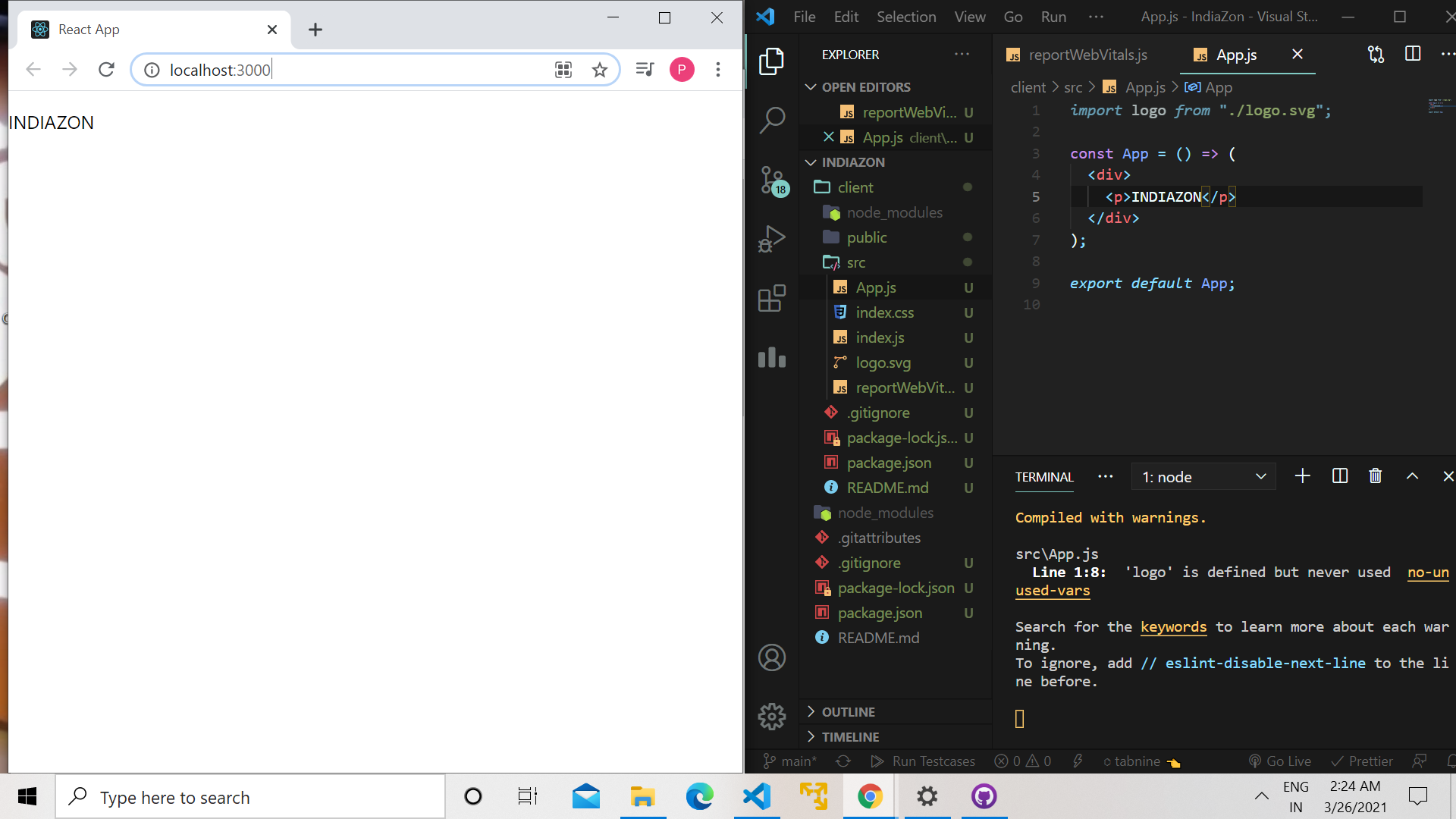
}

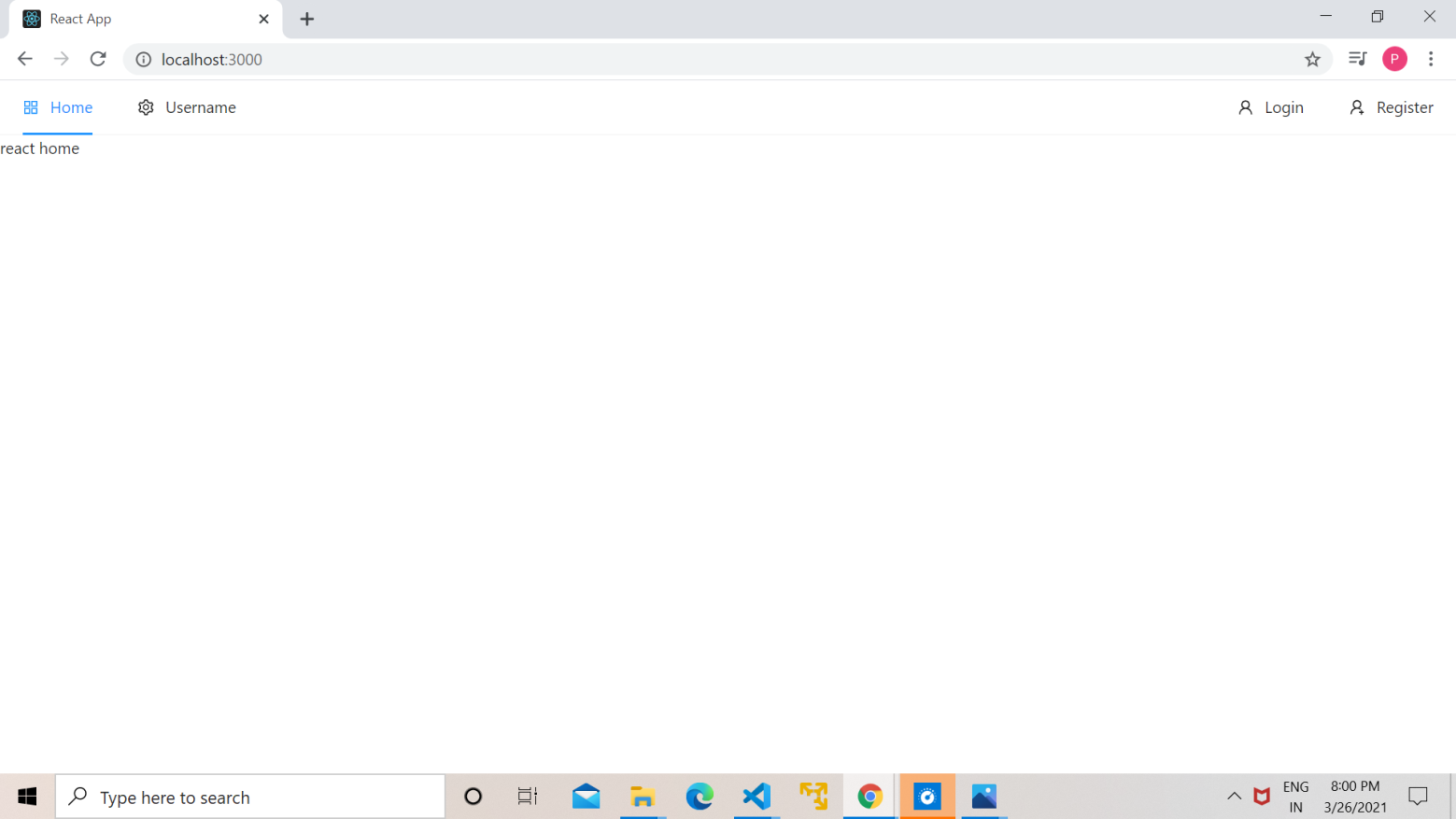
};

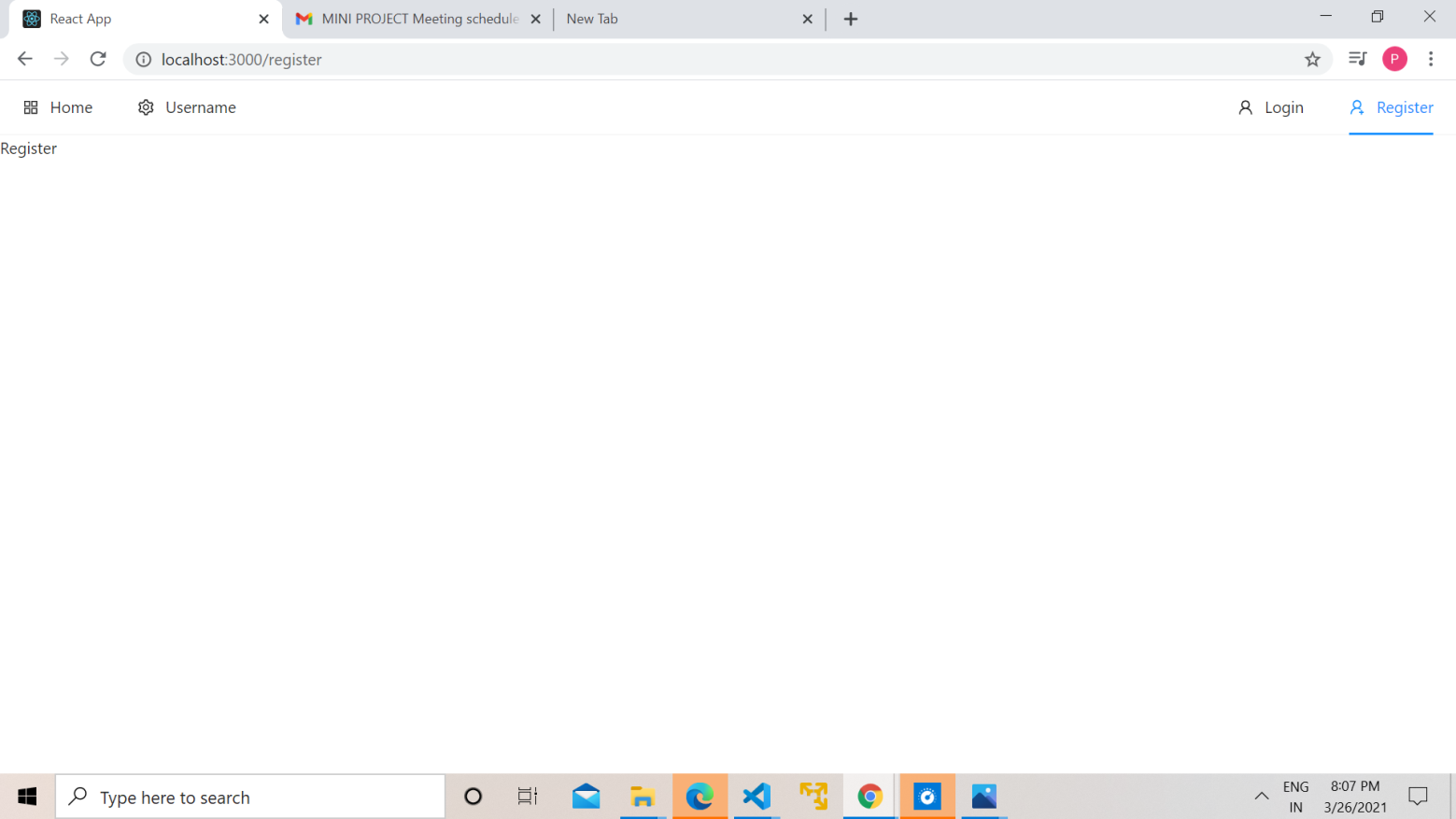
export default reportWebVitals;

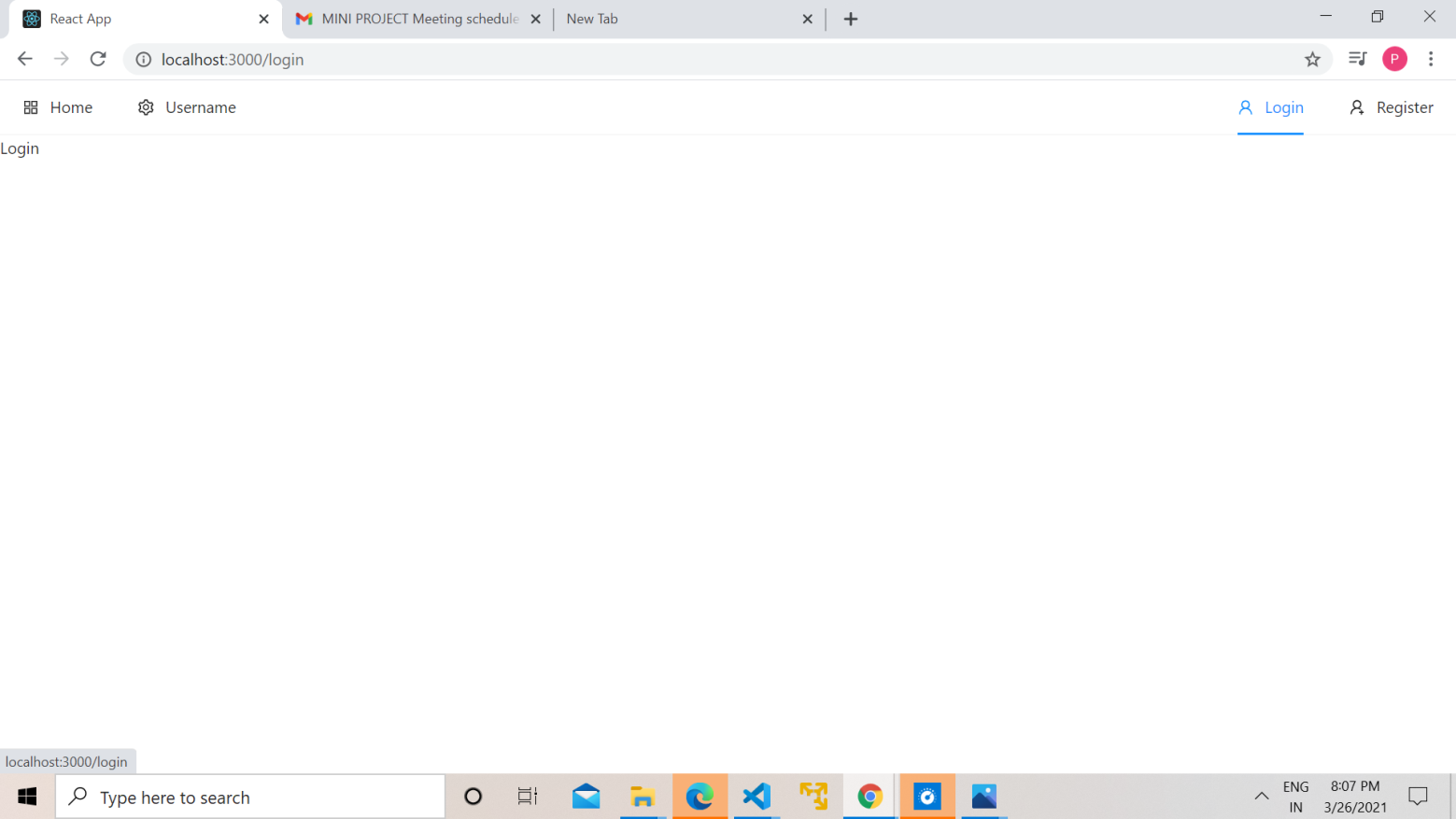
**Screenshots**











### What Is React?

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”.

React has a few different kinds of components, but we’ll start with React.Component subclasses:

We’ll get to the funny XML-like tags soon. We use components to tell React what we want to see on the screen. When our data changes, React will efficiently update and re-render our components.

Here, ShoppingList is a **React component class**, or **React component type**. A component takes in parameters, called props (short for “properties”), and returns a hierarchy of views to display via the render method.

The render method returns a description of what you want to see on the screen. React takes the description and displays the result. In particular, render returns a **React element**, which is a lightweight description of what to render. Most React developers use a special syntax called “JSX” which makes these structures easier to write. The <div /> syntax is transformed at build time to React.createElement('div'). The example above is equivalent to:

## **What is Babel?**

Babel is a JavaScript compiler that can translate markup or programming languages into JavaScript.

With Babel, you can use the newest features of JavaScript (ES6 - ECMAScript 2015).

Babel is available for different conversions. React uses Babel to convert JSX into JavaScript.

## **What is JSX?**

JSX stands for **J**ava**S**cript **X**ML.

JSX is an XML/HTML like extension to JavaScript.

As you can see above, JSX is not JavaScript nor HTML.

JSX is a XML syntax extension to JavaScript that also comes with the full power of ES6 (ECMAScript 2015).

Just like HTML, JSX tags can have a tag names, attributes, and children. If an attribute is wrapped in curly braces, the value is a JavaScript expression.

## **React DOM Render**

The method ReactDom.render() is used to render (display) HTML elements:

## **JSX Expressions**

Expressions can be used in JSX by wrapping them in curly **{}** braces.

## **React Elements**

React applications are usually built around a single **HTML element**.

React developers often call this the **root node** (root elemen)

Elements are **rendered** (displayed) with the ReactDOM.render() method

React elements are **immutable**. They cannot be changed.

The only way to change a React element is to render a new element every time:

**React Fragment**

We know that we make use of the render method inside a component whenever we want to render something to the screen. We may render a single element or multiple elements, though rendering multiple elements will require a ‘div’ tag around the content as the render method will only render a single root node inside it at a time.

**Reason to use Fragments**: As we saw in the above code when we are trying to render more than one root element we have to put the entire content inside the ‘div’ tag which is not loved by many developers. So in React 16.2 version, Fragments were introduced, and we use them instead of the extraneous ‘div’ tag.

**React Props**

Props are an optional input, and can be used to send data to the component. Props are a way of making components easily and dynamically customizable. It’s important to note that props are read-only and that a component must never modify the props passed to it. This also makes them come in handy when you want to display fixed values.

Now that you know about props, make use of them in the components that we have just created with a custom name appended to it.

### ReactJS | Components

### A Component is one of the core building blocks of React. In other words, we can say that every application you will develop in React will be made up of pieces called components. Components make the task of building UIs much easier. You can see a UI broken down into multiple individual pieces called components and work on them independently and merge them all in a parent component which will be your final UI.

### You can see in the below image we have broken down the UI of GeeksforGeeks’s homepage into individual components.

### Functional Components: Functional components are simply javascript functions. We can create a functional component in React by writing a javascript function. These functions may or may not receive data as parameters, we will discuss this later in the tutorial. Below example shows a valid functional component in React:

**Class Components**: The class components are a little more complex than the functional components. The functional components are not aware of the other components in your program whereas the class components can work with each other. We can pass data from one class component to other class components. We can use JavaScript ES6 classes to create class-based components in React. Below example shows a valid class-based component in React: 

### [React Router](https://www.beta-labs.in/2021/03/react-router.html)

As opposed to traditional multi-page applications, SPAs only maintain one HTML file, commonly index.html. Instead of serving a different HTML file for each path, SPAs depend on client-side routing. React Router is a popular client-side routing library.

## **Routing is a process in which a user is directed to different pages based on their action or request. ReactJS Router is mainly used for developing** **Single Page Web Applications (SPA).  React Router is used to define multiple routes in the application. When a user types a specific URL into the browser, and if this URL path matches any 'route' inside the router file, the user will be redirected to that particular route.**

**Why use React router?**

React Router plays an important role to display multiple views in a single page application. Without React Router, it is not possible to display multiple views in React applications. Most of the social media websites like Facebook, Instagram uses React Router for rendering multiple views.

React Router uses component structure to call components, which display the appropriate information. React router also allows the user to utilize browser functionality like the back button, and the refresh page, all while maintaining the correct view of the application.

**React Router Installation**

React contains three different packages for routing. These are:

1.  **react-router**: It provides the core routing components and functions for the React Router applications.

2.    **react-router-dom**: It is used for web applications design.

3.    **react-router-native**: It is used for mobile applications.

It is not possible to install react-router directly in your application. To use react routing, first, you need to install react-router-dom modules in your application. The below command is used to install react router dom.

npm install react-router-dom

**BrowserRouter as Router:**

First, you'll need to set up your app to work with React Router. Everything that gets rendered will need to go inside the <BrowserRouter> element, but we just rename BrowserRouter as Router only to reduce the long name of the BrowserRouter. so wrap your App in those first. It's the component that does all the logic of displaying various components that you provide it with.

**Switch**

Next, in your App component, add the Switch element (open and closing tags). These ensure that only one component is rendered at a time. It checks each route for a match sequentially and stops once the first match is found.

## **Route**

It's now time to add your <Route> tags. These are the links between the components and should be placed inside the <Switch> tags.

To tell the <Route> tags which component to load, simply add a path attribute and the name of the component you want to load with component attribute. The <Route> will return null in case the specified URL doesn’t match the defined path.

## **Link**

Sometimes, we want to need multiple links on a single page. When we click on any of that particular Link, it should load that page which is associated with that path without reloading the web page. To do this, we need to import <Link> component in the Navbar.js file. The Link component is used to trigger new routes. You import it from react-router-dom, and you can add the Link components to point at different routes, with the to attribute. It is the equivalent of anchor tags: <a> </a>.

So now we should update Navbar.js for providing Navigation link in react application by replacing <a> tag to <Link> Tag and 'href' to 'to' attribute.

**React Hooks**

**With the release of React 16.8 in 2019, React Hooks have finally become available to use in our production applications. This allows React developers to make functional components stateful. Instead of using a class component to hold stateful logic, we can use functional components.**

**React Hooks are not available in class-based components — but instead they allow you to use React without JavaScript classes. If you prefer to use functions instead of classes (which is recommended in JavaScript) you can simply start using Hooks without worrying about migrating your whole apps to classes.**

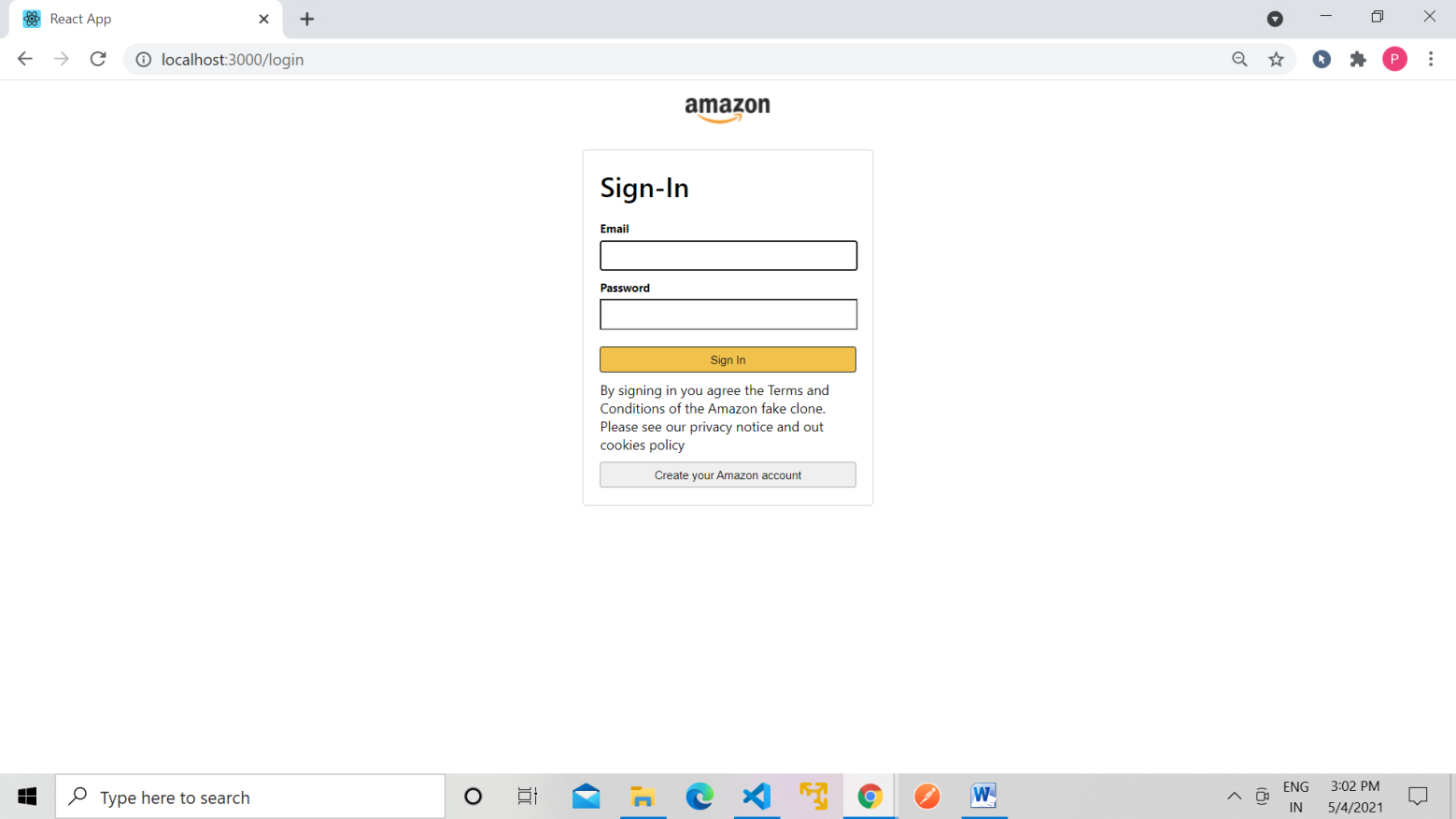
**Hooks enable you to use “class-features” In React by providing a set of built-in functions such as:**

**The useState() hook for using states from function components,**

**The useEffect() hook for performing side effects from function components (It's equivalent to life-cycle methods like componentDidMount, componentDidUpdate, and componentWillUnmount in React classes).**

**Firewall Authentication**

A firewall can support various authentication methods. Authentication in basic definition means a user is claiming to be who they say they are and are allowed access to the resources they are authenticating for.

****

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| **Creating products** |  |
| **Advantages of Firebase:**   1. It is very simple and user friendly. 2. No need for complicated configuration. 3. Firebase offers simple control dashboard. 4. Offers number of useful services. 5. The data is real-time, which means that every change will automatically update connected clients. |  |
| **Checkout**    **Conclusion** |  |

* Faster buying process.
* Store and product listing creation.
* **Cost** reduction.
* Affordable advertising and marketing.
* **Flexibility** for customers.
* No **reach** limitations.
* Product and price comparison.
* Faster response to buyer/market demands.

**References:**

**Githublink:**[**https://github.com/kritarth2121/Ecommerce-website**](https://github.com/kritarth2121/Ecommerce-website)

<https://reactjs.org/docs/getting-started.html>

<https://www.w3schools.com/>

<https://www.youtube.com/>