

Tutorial No. 08

**Title: Introduction to React
JS.**

Batch:A2

Roll No.:16010421063

Tutorial No:8

Aim: To implement methods, functions to manipulate DOM element using React JS

Resources needed: Notepad++, Web Browser

Theory:

React JS

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”. 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. React does not manipulate the browser's DOM directly. Instead, React creates a virtual DOM in memory, where it does all the necessary manipulating, before making the changes in the browser DOM.

We will start with React .Component example

```
class ShoppingList extends React.Component {  
  render() {  
    return (  
      <div className="shopping-list">  
        <h1>Shopping List for {this.props.name}</h1>  
        <ul>  
          <li>Instagram</li>  
          <li>WhatsApp</li>  
          <li>Oculus</li>  
        </ul>  
      </div>  
    );  
  }  
}
```

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:

```
return React.createElement('div', {className: 'shopping-list'},
  React.createElement('h1', /* ... h1 children ... */),
  React.createElement('ul', /* ... ul children ... */)
);
```

The createElement is an inbuilt method. There are lots of methods and functions included in the reactjs API. see the details on reactjs.org API Reference.

JSX comes with the full power of JavaScript. You can put any JavaScript expressions within braces inside JSX. Each React element is a JavaScript object that you can store in a variable or pass around in your program.

Hello World Example

```
<!DOCTYPE html>
<html>
  <script src="https://unpkg.com/react@16/umd/react.production.min.js"></script>
  <script src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js"></script>
  <script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
  <body>

    <div id="mydiv"></div>

    <script type="text/babel">
      class Hello extends React.Component {
        render() {
          return <h1>Hello World!</h1>
        }
      }

      ReactDOM.render(<Hello />, document.getElementById('mydiv'))
    </script>
  </body>
</html>
```

Setting up a React Environment

First of all you need to download NodeJs for your operating system version.

<https://nodejs.org/en/download/>

npm(node packet manager once you download NodeJs

Then follow the following commands:

1) run the command “C:\Users\Your Name>npm install -g create-react-app”

2) run the command to create an application name

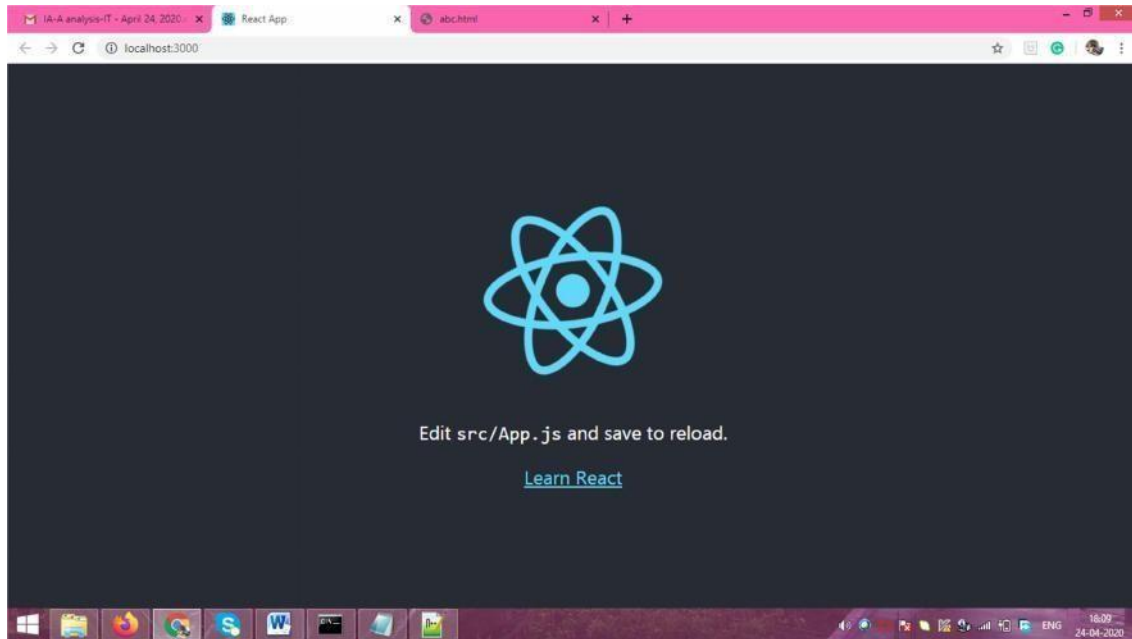
helloworld “C:\Users\Your Name>npx create-react-app

Helloworld”

3) run this command to get to the current directory “C:\Users\Your Name>cd helloworld”

4) run this command to start the react application “C:\Users\Your Name\helloworld >npm start”

A new browser window will pop up with your newly created React App! If not, open your browser and type localhost:3000 in the address bar.



Modify the React Application

Look in the helloworld directory, and you will find a src folder. Inside the src folder there is a file called App.js, open it and make changes to any HTML part. You will be able to see the change on the newly opened browser

App.js File

```
import React, { Component } from 'react';
import logo from './logo.svg';
import './App.css';

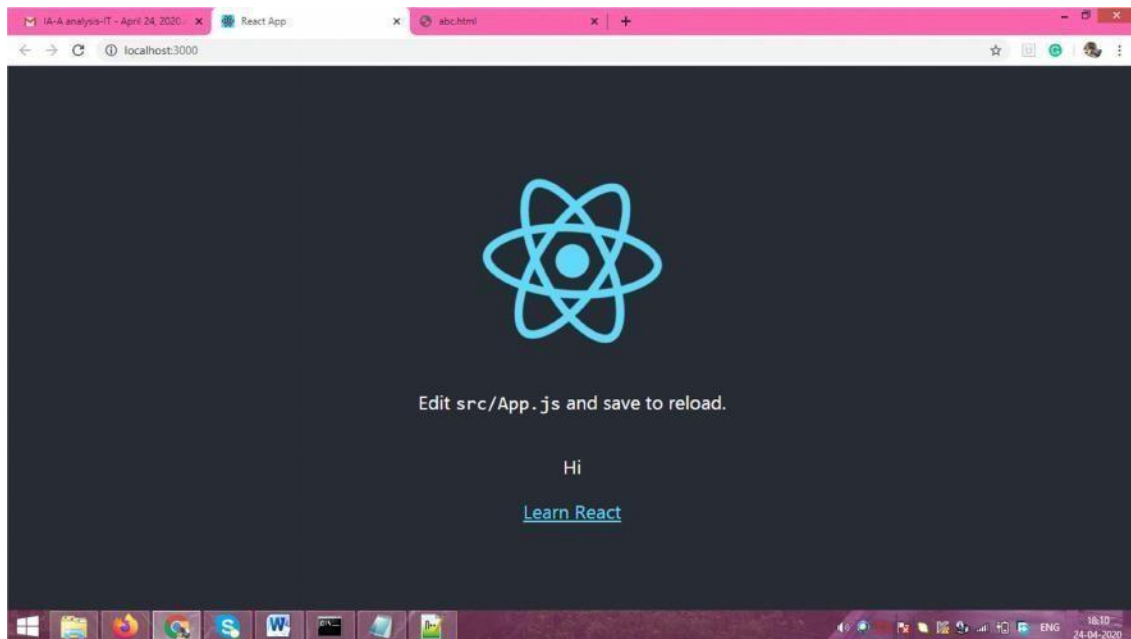
class App extends Component {
  render() {
    return (
      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <p>
            Edit <code>src/App.js</code> and save to reload.
          </p>
        <p> Hi </p>
        <a
          className="App-link"
          href="https://reactjs.org"
```

```
        target="_blank"  
        rel="noopener noreferrer"  
      >  
        Learn React  
      </a>  
    </header>  
  </div>  
}  
}
```

export default App;

in the file “Hi” is added inside paragraph tag.

The result browser screen is as follows



Activity:

- Use React JS to change any of the HTML content.

Results: (Program printout with output)

```
import axios from "axios";  
import { useSnackbar } from "notistack";  
import React, { useEffect, useState } from "react";  
import { useParams } from "react-router-dom";  
import { url } from "../../constants/baseUrl";  
import getCommonOptions from "../../helpers/getCommonOptions";  
import ReactMarkdown from "react-markdown";
```

```
import styles from "./BlogDetails.module.css";
import CircularProgress from "@mui/material/CircularProgress";
const BlogDetails = () => {
  let { blogId } = useParams();
  const [data, setData] = useState({});
  const { enqueueSnackbar } = useSnackbar();
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    const getData = async () => {
      axios
        .post(
          `${url}blogs/getDetailBlog/`,
          {
            blog_id: blogId,
          },
          getCommonOptions()
        )
        .then((res) => {
          setLoading(false);

          setData(res.data);
        })
        .catch(() =>
          enqueueSnackbar("Internal Server Error", {
variant: "error" })
        );
    };

    getData();
  }, []);
  useEffect(() => {
    console.log(data.category);
  }, [data]);

  if (loading) {
    return (
      <div className={styles.loader}>
        { /* helllloo */ }
        <CircularProgress color="primary" determinate={false}
size="lg" />
      </div>
    );
  }
}
```

```
    return (
      <div className={styles.container}>
        <img src={`_${url}_${data.image}`} className={styles.image}
      />
        <div
          style={{
            marginTop: "1rem",
          }}
        >
          <div className={styles.title}>{data.title}</div>
          <div className={styles.author}>Author-
{data.authorName}</div>
          </div>
          <div className={styles.content}>
            <ReactMarkdown>{data.content}</ReactMarkdown>
          </div>
          <div className={styles.categoryContainer}>
            <span
              style={{
                fontSize: "1.2rem",
              }}
            >
              Category-
            </span>
            {data?.category?.map((x) => (
              <div className={styles.category}>{x.name}</div>
            ))}
          </div>
          <div className={styles.creatorSupport}>
            Hey do you like the content Creator? Now you can
support{" "}
            {data.authorName} by donating Matic
          </div>
        </div>
      );
    };
  };

export default BlogDetails;
```

SHOW ME THE CODE

LINUS TORVALDS

AI in healthcare

Author- arya2

Artificial intelligence (AI) has already made significant strides in many industries, including healthcare. From identifying cancer to predicting patient outcomes, AI has the potential to revolutionize healthcare in the coming years.

Current Applications of AI in Healthcare

AI is already being used in a number of ways in healthcare. For example:

- **Diagnosis:** AI algorithms can be trained to identify patterns in medical images, such as X-rays, MRIs, and CT scans, to help diagnose diseases and conditions.
- **Patient monitoring:** AI can help healthcare providers monitor patients remotely, using wearables and other devices to collect and analyze data.
- **Drug discovery:** AI can help identify new drug targets, analyze chemical structures, and predict the efficacy of drugs.
- **Administrative tasks:** AI can help automate administrative tasks, such as appointment scheduling and billing.

The Future of AI in Healthcare

AI has the potential to revolutionize healthcare in the coming years. Here are just a few ways it could be used:

- **Predictive analytics:** AI can help predict patient outcomes, identify high-risk patients, and recommend treatment plans.
- **Precision medicine:** AI can help identify personalized treatment plans based on a patient's genetics, lifestyle, and other factors.
- **Robot-assisted surgery:** AI can be used to guide surgical robots, improving accuracy and reducing the risk of complications.
- **Virtual assistants:** AI-powered virtual assistants could help patients manage their health, reminding them to take medication and providing other support.

Challenges and Concerns

Despite the potential benefits of AI in healthcare, there are also concerns about its use. Here are a few challenges that need to be addressed:

- **Data privacy:** AI relies on large amounts of data to be effective, which raises concerns about patient privacy.

Home

```
import React from "react";
import RecentPosts from "../components/RecentPosts/RecentPosts";
import TopPosts from "../components/TopPosts/TopPosts";
import styles from "./Home.module.css";

const Home = () => {
  return (
    <div className={styles.container}>
      <section className={styles.section}>
        <div>
          
        </div>
        <div>
          <h1>Making Blogging profitable with
Blockchain</h1>
        </div>
      </section>
      <RecentPosts />
      <TopPosts />
    </div>
  );
};

export default Home;
```

Recent Post


```
import React, { useEffect, useState } from "react";
import styles from "../RecentPosts.module.css";
import axios from "axios";
import { url } from "../../constants/baseUrl";
import getCommonOptions from "../../helpers/getCommonOptions";
import PostItem from "../PostItem/PostItem";

const RecentPosts = () => {
  const [blogList, setBlogList] = useState([]);

  useEffect(() => {
    const getBlogs = async () => {
      axios
        .get(`${url}blogs/getRecentBlog`, getCommonOptions())
        .then((res) => {
          setBlogList(res.data);
        })
        .catch((err) => console.log(err));
    };
    getBlogs();
  }, []);

  return (
    <div className={styles.container}>
      <div className={styles.containerHeader}>
        <div>
          <h3>Recent Posts</h3>
        </div>
        <div style={{ textAlign: "right", textDecoration:
"underline" }}>
          See More
        </div>
      </div>
      <div className={styles.container2}>
        {blogList.map((item) => (
          <PostItem item={item} />
        ))}
      </div>
    </div>
  );
};
```

```
export default RecentPosts;
```

Top Post

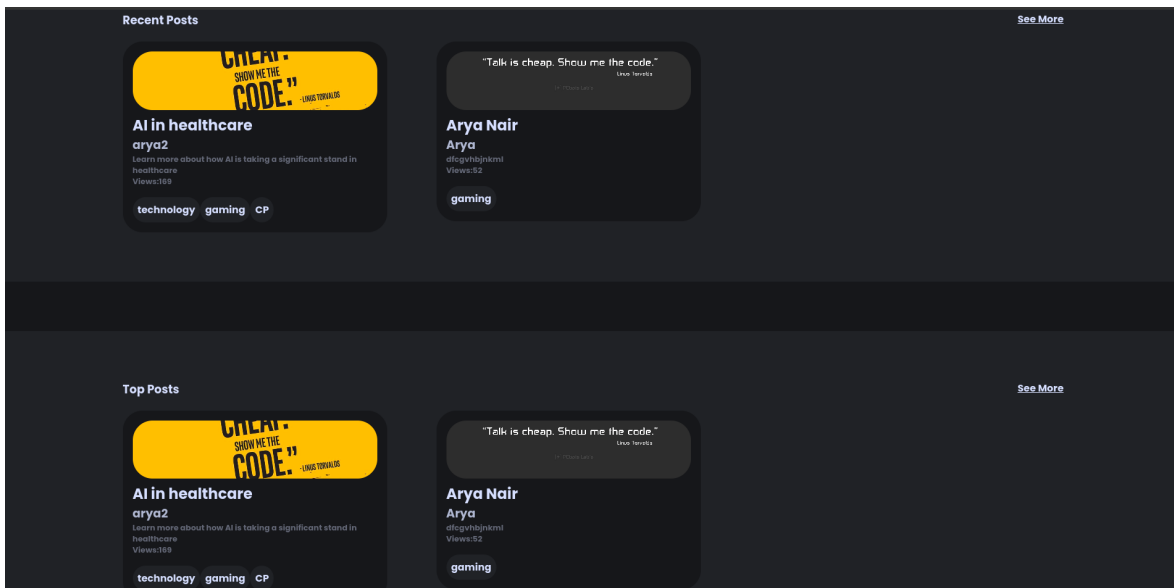
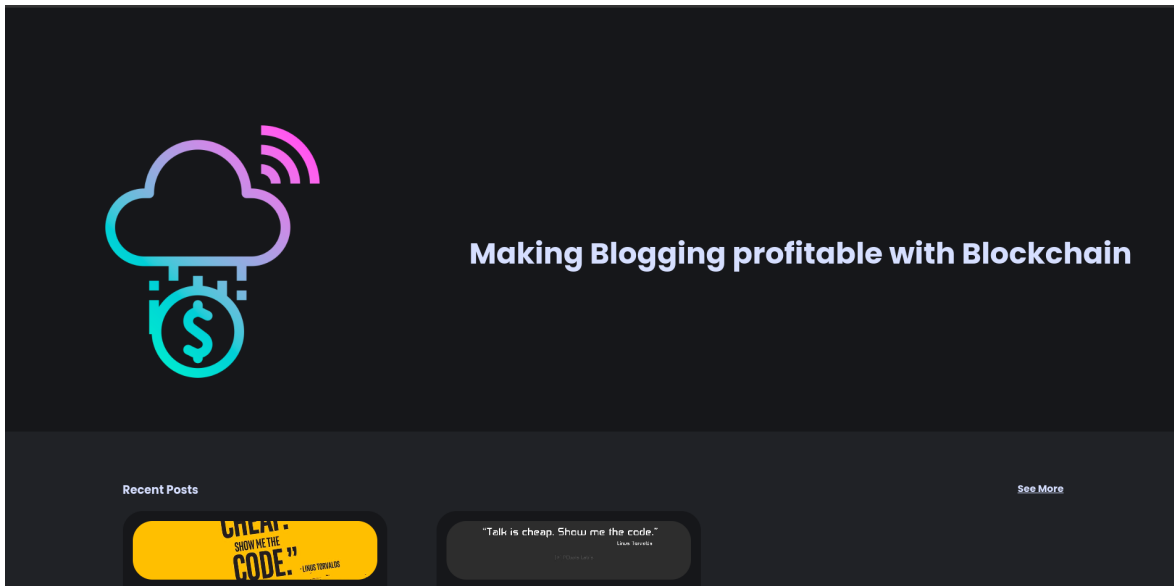
```
import React, { useEffect, useState } from "react";
import styles from "../TopPosts.module.css";
import axios from "axios";
import { url } from "../../constants/baseUrl";
import getCommonOptions from "../../helpers/getCommonOptions";
import PostItem from "../PostItem/PostItem";

const TopPosts = () => {
  const [blogList, setBlogList] = useState([]);

  useEffect(() => {
    const getBlogs = async () => {
      axios
        .get(`${url}blogs/getTopBlog`, getCommonOptions())
        .then((res) => {
          setBlogList(res.data);
        })
        .catch((err) => console.log(err));
    };
    getBlogs();
  }, []);

  return (
    <div className={styles.container}>
      <div className={styles.containerHeader}>
        <div>
          <h3>Top Posts</h3>
        </div>
        <div style={{ textAlign: "right", textDecoration:
"underline" }}>
          See More
        </div>
      </div>
      <div className={styles.container2}>
        {blogList.map((item) => (
          <PostItem item={item} />
        ))}
      </div>
    </div>
  );
};
```

```
);  
};  
  
export default TopPosts;
```



Outcomes:CO 4 Implement web application using React JS, JSON and CBOR

Conclusion: (Conclusion to be based on the outcomes achieved) completed and implemented react

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

<http://www.w3schools.com>

https://www.tutorialspoint.com/angularjs/angularjs_tutorial.pdf

<https://angularjs.org>
