**1. ReactJS-HOL**

**Program:**

Open the command prompt and do the following steps as mentioned below:

Step 1: **npm install -g create-react-app**

Step 2**: npx create-react-app myfirstreact**

Step 3: After the folder named myfirstreact is created now navigate the folder using command

**cd myfirstreact**

Step 4: Now open folder myfirstreact in Visual Studio and then open app.js file which is in src folder of myfirstreact.

Step 5: If already code is there in **App.js** file then replace code with following program.

**Code:**

**function App(){**

**return (**

**<h1> Welcome the first session of react </h1>**

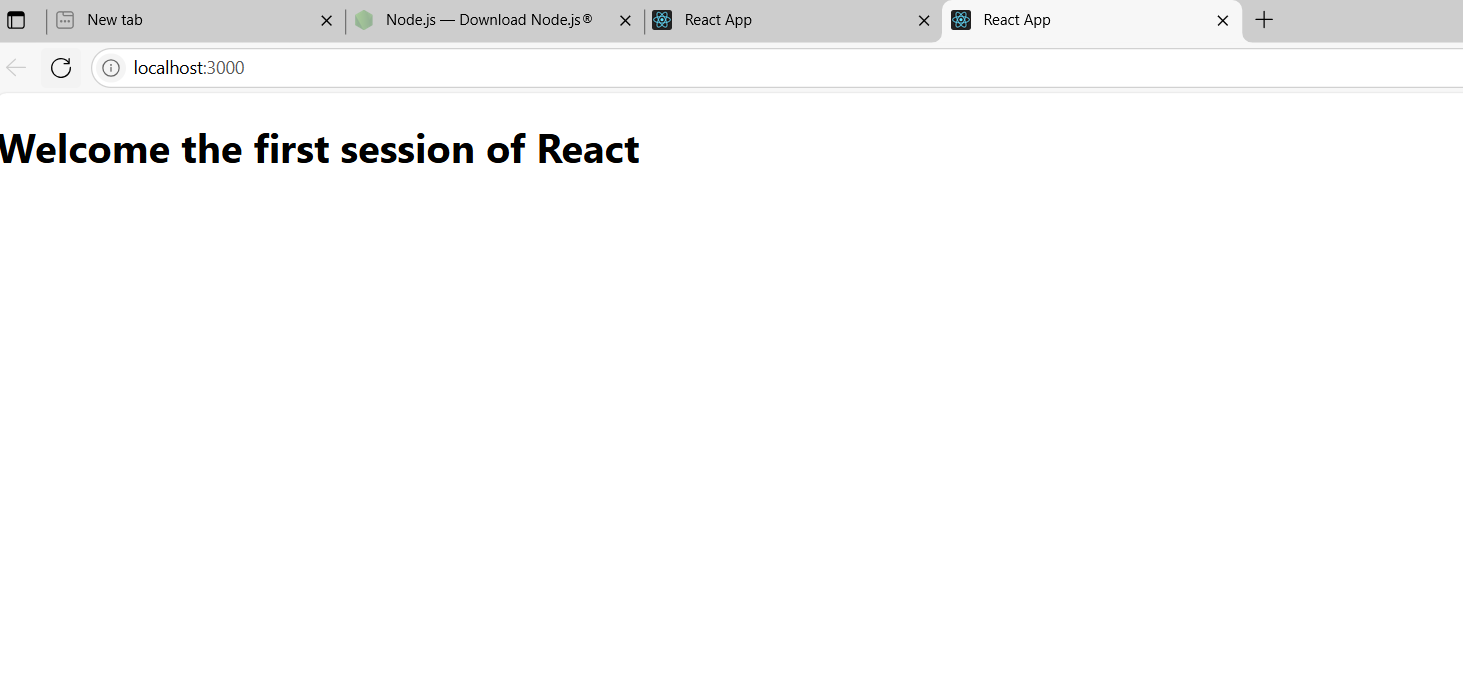
**);**

**}**

Step 6: Now again open command prompt and to execute the react application execute the following command

**npm start**

After sometime in a new browser window the react application opens and the output is as follows:



**2. ReactJS-HOL**

**Program:**

Open the command prompt and follow the given steps:

Step 1: Create a react app using command

**npx create-react-app studentapp**

Step 2: After the react app is created then you have to direct towards the folder named studentapp now using command:

**cd studentapp**

Step3: Now create a Components folder inside src folder which is in studentapp folder.

**mkdir src\Components**

Step 4: Now create Home.js Component

**notepad src\Components\Home.js**

Now if the Home.js is not existed before it will be created now and inside the javascript file type the following code:

**Code:**

**import React, { Component } from 'react';**

**class Home extends Component {**

**render() {**

**return (**

**<div>**

**<h2>Welcome to the Home page of Student Management Portal</h2>**

**</div>**

**);**

**}**

**}**

**export default Home;**

Step 6: Create About.js Component

**notepad src\Components\About.js**

if About.js file is already not created then the file will be created and then type the following.

**Code:**

**import React, { Component } from 'react';**

**class About extends Component {**

**render() {**

**return (**

**<div>**

**<h2>Welcome to the About page of Student Management Portal</h2>**

**</div>**

**);**

**}**

**}**

**export default About;**

Step 7: create Contact.js Component using command

**notepad src\Components\Contact.js**

and type the following command in Contact.javascript file.

**Code:**

**import React, { Component } from 'react';**

**class Contact extends Component {**

**render() {**

**return (**

**<div>**

**<h2>Welcome to the Contact page of Student Management Portal</h2>**

**</div>**

**);**

**}**

**}**

**export default Contact;**

Step 8: Now update App.js file in order to render all the components you created.

**notepad src\App.js**

Step 9: The App.js file opens and replace the code with following.

**Code:**

**import React from 'react';**

**import './App.css';**

**import Home from './Components/Home';**

**import About from './Components/About';**

**import Contact from './Components/Contact';**

**function App() {**

**return (**

**<div className="App">**

**<h1>Student Management Portal</h1>**

**<Home />**

**<About />**

**<Contact />**

**</div>**

**);**

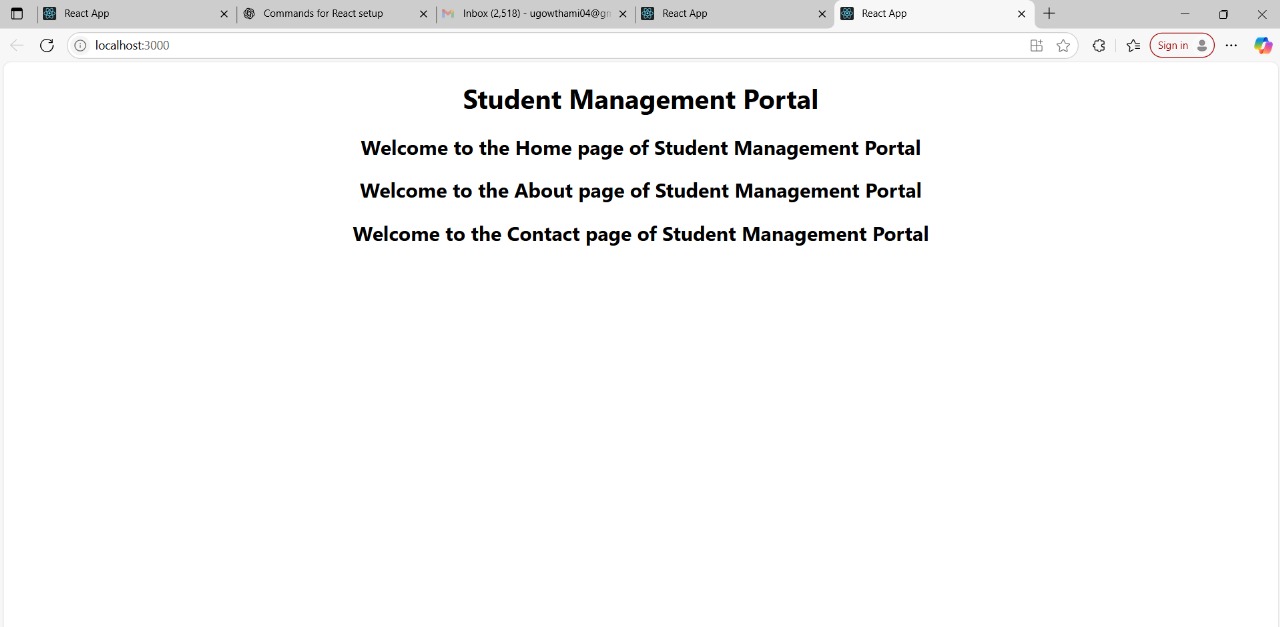
**}**

**export default App;**

Step 10: in order to execute react application open command prompt and execute following command

**npm start**

Step 11: Now the application opens in a new browser window automatically and output is as follows:



**3. ReactJS-HOL**

**Program:**

Open the command prompt and follow the below steps

Step 1: create the react project

**npx create-react-app scorecalculatorapp**

this command creates a folder named scorecalculatorapp

Step 2: Navigate to the folder you created

**cd scorecalculatorapp**

Step 3: Create a Components folder inside folder you created in above step

Inside src/ create the folder Components

**mkdir src\Components**

Step 4: Now create a file named CalculatorScore.js

**notepad src\Components\CalculateScore.js**

This will create the javascript file if it is not available and then tyoe the following code after the javascript file is created:

**Code:**

**import React from 'react';**

**import '../Stylesheets/mystyle.css';**

**function CalculateScore(props) {**

**const { name, school, total, goal } = props;**

**const average = total / goal**

**return (**

**<div className="score-card">**

**<h2>{name}</h2>**

**<p>School: {school}</p>**

**<p>Total Marks: {total}</p>**

**<p>Goal: {goal}</p>**

**<p>Average Score: {average.toFixed(2)}</p>**

**</div>**

**);**

**}**

**export default CalculateScore;**

Step 5: Now create Styles folder and in that folder create a file

**mkdir src\Stylesheets**

Step 6: now inside Styleds folder create a file

**notepad src\Stylesheets\mystyle.css**

In notepad the css file will be opened and type the following code:

**.score-card {**

**background-color: #f2f2f2;**

**border: 1px solid #ccc;**

**padding: 20px;**

**width: 300px;**

**margin: 20px auto;**

**font-family: Arial, sans-serif;**

**border-radius: 8px;**

**box-shadow: 0px 2px 8px rgba(0,0,0,0.1);**

**}**

Step 7: Now edit App.js file

**notepad src\App.js**

now replace contents in file using the below program.

**Code:**

**import React from 'react';**

**import './App.css';**

**import CalculateScore from './Components/CalculateScore';**

**function App() {**

**return (**

**<div className="App">**

**<h1>Student Score Calculator</h1>**

**<CalculateScore name="Madhavi" school="ABC School" total={450} goal={5} />**

**</div>**

**);**

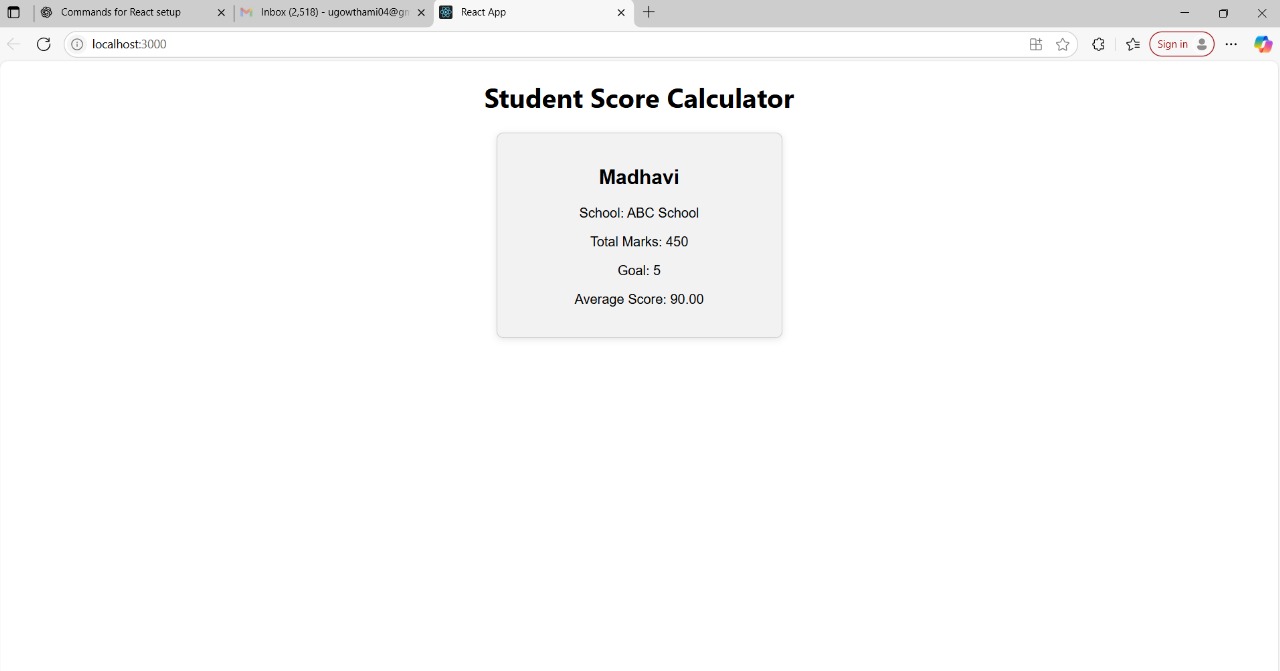
**}**

**export default App;**

Step 8: open command prompt and then in order to execute the react application tyoe the command

**npm start**

Step 9: Automatically the browser window opens which shows the output of the react application we created.



**4. ReactJS-HOL**

**Program:**

Open the command prompt and then do the below steps

Step 1: Create your react project

**npx create-react-app blogapp**

Step 2: Now navigate to the folder you just created

**cd blogapp**

Step 3: Create Post.js file in src/ folder

**notepad src\Post.js**

Now paste the following in the javascript file we created now.

**Code:**// src/Post.js

**class Post {**

**constructor(userId, id, title, body) {**

**this.userId = userId;**

**this.id = id;**

**this.title = title;**

**this.body = body;**

**}**

**}**

**export default Post;**

Step 4: Now in the javascript file paste the following class component with all hooks

**Code:**

**// src/Posts.js**

**import React, { Component } from 'react';**

**import Post from './Post';**

**class Posts extends Component {**

**constructor(props) {**

**super(props);**

**this.state = {**

**posts: [],**

**hasError: false,**

**errorInfo: null,**

**};**

**}**

**// Lifecycle Hook: Runs after component mounts**

**componentDidMount() {**

**this.loadPosts();**

**}**

**// Fetch Posts from API**

**loadPosts() {**

**fetch('https://jsonplaceholder.typicode.com/posts')**

**.then((response) => response.json())**

**.then((data) => {**

**const postsArray = data.map(**

**(item) => new Post(item.userId, item.id, item.title, item.body)**

**);**

**this.setState({ posts: postsArray });**

**})**

**.catch((error) => {**

**this.setState({ hasError: true, errorInfo: error });**

**});**

**}**

**// Lifecycle Hook: Catch errors in rendering**

**componentDidCatch(error, info) {**

**alert('Error: ' + error);**

**this.setState({ hasError: true, errorInfo: info });**

**}**

**render() {**

**if (this.state.hasError) {**

**return <h2>Something went wrong. Please try again later.</h2>;**

**}**

**return (**

**<div>**

**<h1>All Blog Posts</h1>**

**{this.state.posts.map((post) => (**

**<div key={post.id} style={{ border: '1px solid #ccc', margin: '10px', padding: '10px' }}>**

**<h2>{post.title}</h2>**

**<p>{post.body}</p>**

**</div>**

**))}**

**</div>**

**);**

**}**

**}**

**export default Posts;**

Step 6: Edit App.js file in order to include Posts component.

**notepad src\App.js**

Now the App.js opens and then replace the old code with following program.

**Code:**

// src/App.js

**import React from 'react';**

**import './App.css';**

**import Posts from './Posts';**

**function App() {**

**return (**

**<div className="App">**

**<Posts />**

**</div>**

**);**

**}**

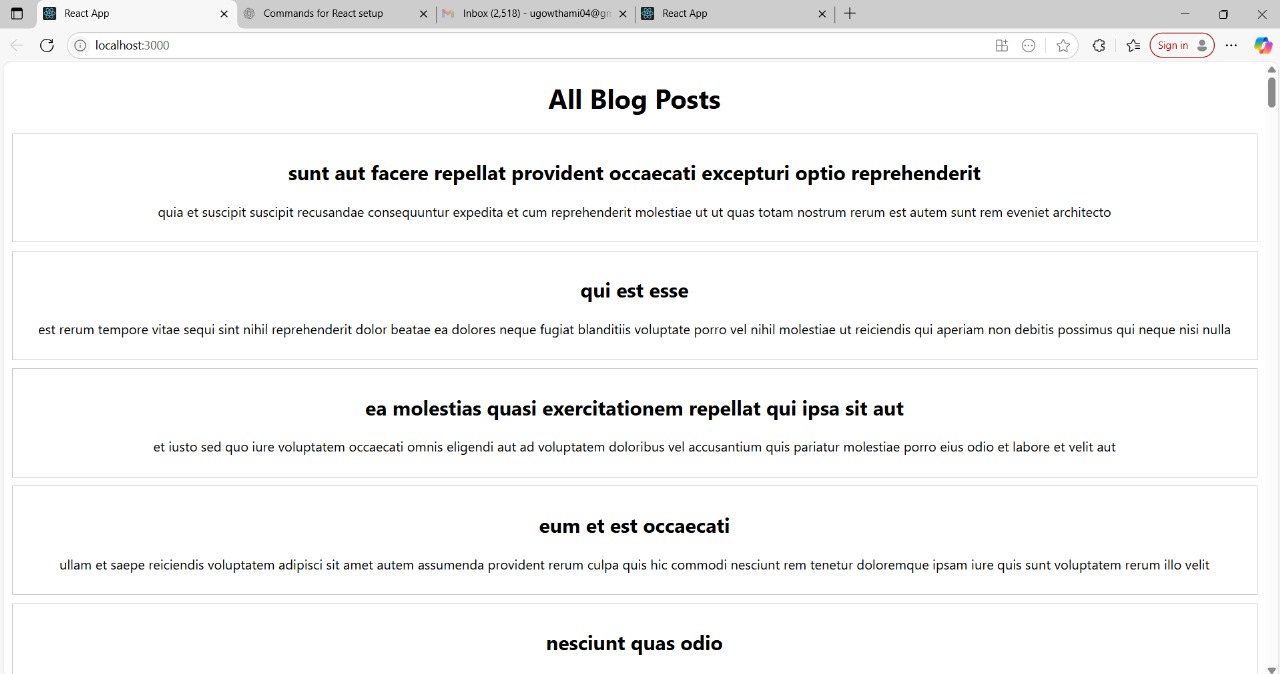
**export default App;**

Step 7: In order to execute the recat application

In command prompt use the command

**npm start**

Step 8: Now automatically new browser window opens and then output is as follows



**5.ReactJS-HOL**

**Objectives**

* Understanding the need for styling react component
* Working with CSS Module and inline styles

In this hands-on lab, you will learn how to:

* Style a react component
* Define styles using the CSS Module
* Apply styles to components using className and style properties

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **30 minutes.**

My Academy team at Cognizant want to create a dashboard containing the details of ongoing and completed cohorts. A react application is created which displays the detail of the cohorts using react component. You are assigned the task of styling these react components.

Download and build the attached react application.



1. Unzip the react application in a folder
2. Open command prompt and switch to the react application folder
3. Restore the node packages using the following commands



Figure 1: Restore packages

1. Open the application using VS Code
2. Create a new CSS Module in a file called “CohortDetails.module.css”
3. Define a css class with the name as “box” with following properties

*Width = 300px;*

*Display = inline block;*

*Overall 10px margin*

*Top and bottom padding as 10px*

*Left and right padding as 20px*

*1 px border in black color*

*A border radius of 10px*

1. Define a css style for html <dt> element using tag selector. Set the font weight to 500.
2. Open the cohort details component and import the CSS Module
3. Apply the box class to the container div
4. Define the style for <h3> element to use “green” color font when cohort status is “ongoing” and “blue” color in all other scenarios.
5. Final result should look similar to the below image



Figure 2: Final Result

**APP.css**

.App {

text-align: center;

}

.App-logo {

height: 40vmin;

pointer-events: none;

}

.app-container {

padding: 20px;

font-family: Arial, sans-serif;

text-align: center;

}

.heading {

font-size: 2rem;

color: #333;

margin-bottom: 20px;

}

.cohort-grid {

display: flex;

justify-content: center;

flex-wrap: wrap;

gap: 20px;

}

@media (prefers-reduced-motion: no-preference) {

.App-logo {

animation: App-logo-spin infinite 20s linear;

}

}

.App-header {

background-color: #282c34;

min-height: 100vh;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

font-size: calc(10px + 2vmin);

color: white;

}

.App-link {

color: #61dafb;

}

@keyframes App-logo-spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

**APP.js**

import './App.css';

import { CohortsData } from './Cohort';

import CohortDetails from './components/CohortDetails';

function App() {

return (

<div className="app-container">

<h1 className="heading">Cohort Details</h1>

<div className="cohort-grid">

{CohortsData.map((cohort, index) => (

<CohortDetails key={index} cohort={cohort} />

))}

</div>

</div>

);

}

export default App;

**Cohort.js**

// src/Cohort.js

export const CohortsData = [

{

name: "INTADMDF10 - .NET FSD",

startedOn: "22-Feb-2022",

status: "Scheduled",

coach: "Aathma",

trainer: "Jojo Jose"

},

{

name: "ADM21JF014 - Java FSD",

startedOn: "10-Sep-2021",

status: "Ongoing",

coach: "Apoorv",

trainer: "Elisa Smith"

},

{

name: "CDBJF21025 - Java FSD",

startedOn: "24-Dec-2021",

status: "Ongoing",

coach: "Aathma",

trainer: "John Doe"

},

{

name: "INTADMJF12 - Java FSD",

startedOn: "22-Feb-2022",

status: "Scheduled",

coach: "Ibrahim",

trainer: "To Be Assigned"

},

{

name: "CDE22JF011 - Java FSD",

startedOn: "24-Dec-2021",

status: "Ongoing",

coach: "Apoorv",

trainer: "Emma Swan"

},

{

name: "INTADMDF09 - Dataware Housing",

startedOn: "22-Feb-2022",

status: "Scheduled",

coach: "Aathma",

trainer: "Babjee Rao"

},

{

name: "ADM22DF001 - .NET FSD",

startedOn: "10-Sep-2021",

status: "Ongoing",

coach: "Ibrahim",

trainer: "Marie Curie"

}

];

**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';

ReactDOM.render(

<React.StrictMode>

<App />

</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;

**Outputs:**



