REACT

01-REACT HANDSON

Create a new React Application with the name “myfirstreact”, Run the application to print “welcome to the first session of React” as heading of that page.

1. To create a new React app, Install Nodejs and Npm from the following link:

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

2. Install Create-react-app by running the following command in the command prompt:

```
C:>npm install -g create-react-app
```

3. To create a React Application with the name of “myfirstreact”, type the following command:

```
C:>npx create-react-app myfirstreact
```

4. Once the App is created, navigate into the folder of myfirstreact by typing the following command:

```
C:>cd myfirstreact
```

5. Open the folder of myfirstreact in Visual Studio Code
6. Open the App.js file in Src Folder of myfirstreact
7. Remove the current content of “App.js”
8. Replace it with the following:

```
function App() {
  return (
    <h1> Welcome the first session of React </h1>
  );
}
```

9. Run the following command to execute the React application:

```
C:\myfirstreact>npm start
```

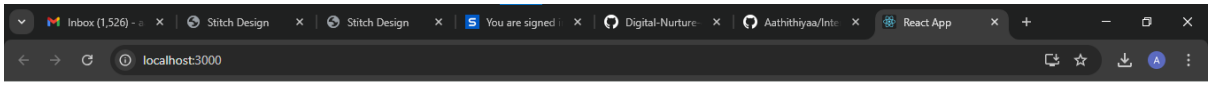
10. Open a new browser window and type “localhost:3000” in the address bar

Code:

App.js

```
import React from 'react';

function App() {
  return (
    <div>
      <h1>welcome to the first session of React</h1>
    </div>
  );
}

export default App;
```

Output:



welcome to the first session of React

02-REACTJS HANDSON

1. Create a React project named “StudentApp” type the following command in terminal of Visual studio:

```
C:>npx create-react-app StudentApp
```

2. Create a new folder under Src folder with the name “Components”. Add a new file named “Home.js”

3. Type the following code in Home.js

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

import class Home extends Component{
   render(){
      <div>
      <h3> Welcome to the Home Page of Student Management Portal </h3>
      </div>
   }
}
```

4. Under Src folder add another file named “About.js”

5. Repeat the same steps for Creating “About” and “Contact” component by adding a new file as ”About.js”, “Contact.js” under “Src” folder and edit the code as mentioned for “Home” Component.

6. Edit the App.js to invoke the Home, About and Contact component as follows:

```
import logo from './logo.svg';
import './App.css';
import {Home} from './Components/Home';
import {About} from './Components/About';
import {Contact} from './Components/Contact';

function App() {
  return (
    <div className="container">
     <Home/>
     <About/>
     <Contact/>
    </div>
  );
}

export default App;
```

7. In command Prompt, navigate into StudentApp and execute the code by typing the following command:

```
C:\studentapp>npm start
```

8. Open browser and type “localhost:3000” in the address bar:

Code:

About.js

```
import React from 'react';

class About extends React.Component {
  render() {
    return (
      <div>
        <h2>Welcome to the About page of the Student Management Portal</h2>
      </div>
    );
  }
}

export default About;
```

Contact.js

```
import React from 'react';

class Contact extends React.Component {
  render() {
    return (
      <div>
        <h2>Welcome to the Contact page of the Student Management Portal</h2>
      </div>
    );
  }
}

export default Contact;
```

Home.js

```
import React from 'react';
```

```
class Home extends React.Component {
  render() {
    return (
      <div>
        <h2>Welcome to the Home page of Student Management Portal</h2>
      </div>
    );
  }
}

export default Home;
```

App.js

```
import React from 'react';

import Home from './Components/Home';
import About from './Components/About';
import Contact from './Components/Contact';

function App() {
  return (
    <div className="App">
      <Home />
      <About />
      <Contact />
    </div>
  );
}

export default App;
```

Output:

Welcome to the Home page of Student Management Portal

Welcome to the About page of the Student Management Portal

Welcome to the Contact page of the Student Management Portal

03-REACTJS-HANDSON

Create a react app for Student Management Portal named scorecalculatorapp and create a function component named “CalculateScore” which will accept Name, School, Total and goal in order to calculate the average score of a student and display the same.

1. Create a React project named “scorecalculatorapp” type the following command in terminal of Visual studio:

   ```
   C:>npx create-react-app scorecalculatorapp
   ```

2. Create a new folder under Src folder with the name “Components”. Add a new file named “CalculateScore.js”

3. Type the following code in CalculateScore.js

   ```
   export const CalculateScore = ({Name,School, total, goal}) => (
       <div className="formatstyle">
         <h1><font color="Brown">Student Details:</font></h1>
         <div className="Name">
           <b> <span> Name:  </span> </b>
           <span>{Name}</span>
         </div>
         <div className="School">
           <b> <span> School: </span> </b>
           <span>{School}</span>
         </div>
         <div className="Total">
           <b><span>Total:</span> </b>
           <span>{total}</span>
           <span>Marks</span>
         </div>
         <div className="Score">
           <b>Score:</b>
           <span>
             {calcScore(
               total,
               goal
             )}
           </span>
         </div>
       </div>
   ```

4. Create a Folder named Stylesheets and add a file named “mystyle.css” in order to add some styles to the components:

```
.Name
{
  font-weight:300;
 color:blue;
}
.School
{
 color:crimson;
}
.Total
{
 color:darkmagenta;
}
.formatstyle
{
   text-align:center;
   font-size:large;
}
.Score
{
   color:forestgreen;
}
```

5. Edit the App.js to invoke the CalculateScore functional component as follows:

```
.Name
{
  font-weight:300;
 color:blue;
}
.School
{
 color:crimson;
}
.Total
{
 color:darkmagenta;
}
.formatstyle
{
   text-align:center;
   font-size:large;
}
.Score
{
   color:forestgreen;
}
```

6. Edit the App.js to invoke the CalculateScore functional component as follows:

```
import {CalculateScore} from '../src/components/CalculateScore';

function App()
{

return(
<div>
   <CalculateScore Name={"Steeve"}
   School={"DNV Public School"}
   total={284}
   goal={3}
   />
</div>
)
}

export default App;
```

7. In command Prompt, navigate into scorecalculatorapp and execute the code by typing the following command:

```
C:\scorecalculatorapp>npm start
```

8. Open browser and type “localhost:3000” in the address bar:

Output:

CalculateScore.js

```
import React from 'react';
import '../Stylesheets/mystyle.css';

function CalculateScore() {
  const name = "Aathithiyaa";
  const school = "Sunshine School";
  const total = 480;
  const goal = "To score above 90%";

  const average = total / 5;

  return (
    <div className="score-card">
      <h2>Student Score Card</h2>
      <p><strong>Name:</strong> {name}</p>
      <p><strong>School:</strong> {school}</p>
      <p><strong>Total Marks:</strong> {total}</p>
```

```
      <p><strong>Average Score:</strong> {average}</p>
      <p><strong>Goal:</strong> {goal}</p>
    </div>
  );
}

export default CalculateScore;
```

mystylesheets

```
.score-card {
  border: 2px solid #4caf50;
  padding: 20px;
  margin: 20px;
  background-color: #f0f9f0;
  border-radius: 10px;
  font-family: Arial, sans-serif;
}

.score-card h2 {
  color: #2e7d32;
}

.score-card p {
  font-size: 16px;
  margin: 5px 0;
}
```

App.js

```
import React from 'react';

import CalculateScore from './Components/CalculateScore';

function App() {
  return (
    <div className="App">
      <CalculateScore />
    </div>
  );
}

export default App;
```

Output:

Student Score Card

Name: Aathithiyaa
School: Sunshine School
Total Marks: 480
Average Score: 96
Goal: To score above 90%

04-REACTJS-HANDSON

1. Create a new react application using *create-react-app* tool with the name as "blogapp"
2. Open the application using VS Code
3. Create a new file named as **Post.js** in **src folder** with following properties

```
Post.js U
1   class Post {
2       constructor(id, title, body){
3           this.id=id;
4           this.title=title;
5           this.body=body;
6       }
7   }
8   export default Post;
```

Figure 1: Post class

4. Create a new class based component named as **Posts** inside **Posts.js** file

```
Posts.js U
1  class Posts extends React.Component {
2      constructor(props){
3          super(props);
4      }
5  }
```

Figure 2: Posts Component

5. Initialize the component with a list of Post in state of the component using the constructor
6. Create a new method in component with the name as **loadPosts()** which will be responsible for using Fetch API and assign it to the component state created earlier. To get the posts use the url (https://jsonplaceholder.typicode.com/posts)

7. Implement the **componentDidMount()** hook to make calls to **loadPosts()** which will fetch the posts

```
JS Posts.js U
1  class Posts extends React.Component {
2      constructor(props){
3          super(props);
4          //code
5      }
6      loadPosts() {
7          //code
8      }
9      componentDidMount() {
10         //code
11     }
12 }
```

Figure 3: componentDidMount() hook

8. Implement the **render()** which will display the title and post of posts in html page using heading and paragraphs respectively.

```
JS Posts.js U
1  class Posts extends React.Component {
2      constructor(props) {…
5      }
6      loadPosts() {…
8      }
9      componentDidMount() {…
11     }
12     render() {
13         //code
14     }
15 }
```

Figure 4: render() method

Define a **componentDidCatch()** method which will be responsible for displaying

Figure 5: render() method

9. Define a **componentDidCatch()** method which will be responsible for displaying any error happing in the component as alert messages.

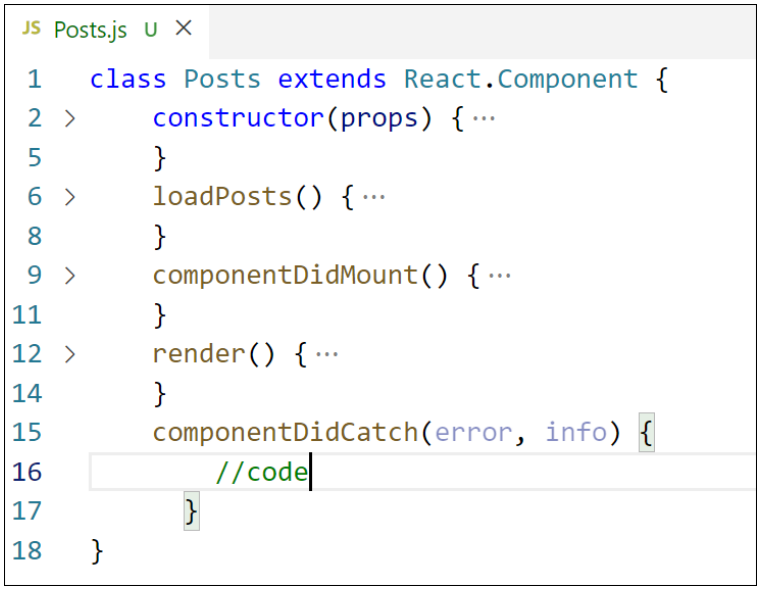


Figure 6: componentDidCatch() hook

10. Add the Posts component to App component.
11. Build and Run the application using *npm start* command.

Code:

App.js

```
import React from 'react';

import Posts from './Posts';

function App() {
  return (
    <div className="App">
      <Posts />
    </div>
  );
}

export default App;
```

Posts.js

```
class Post {

  constructor(userId, id, title, body) {
    this.userId = userId;
    this.id = id;
    this.title = title;
    this.body = body;
  }
```

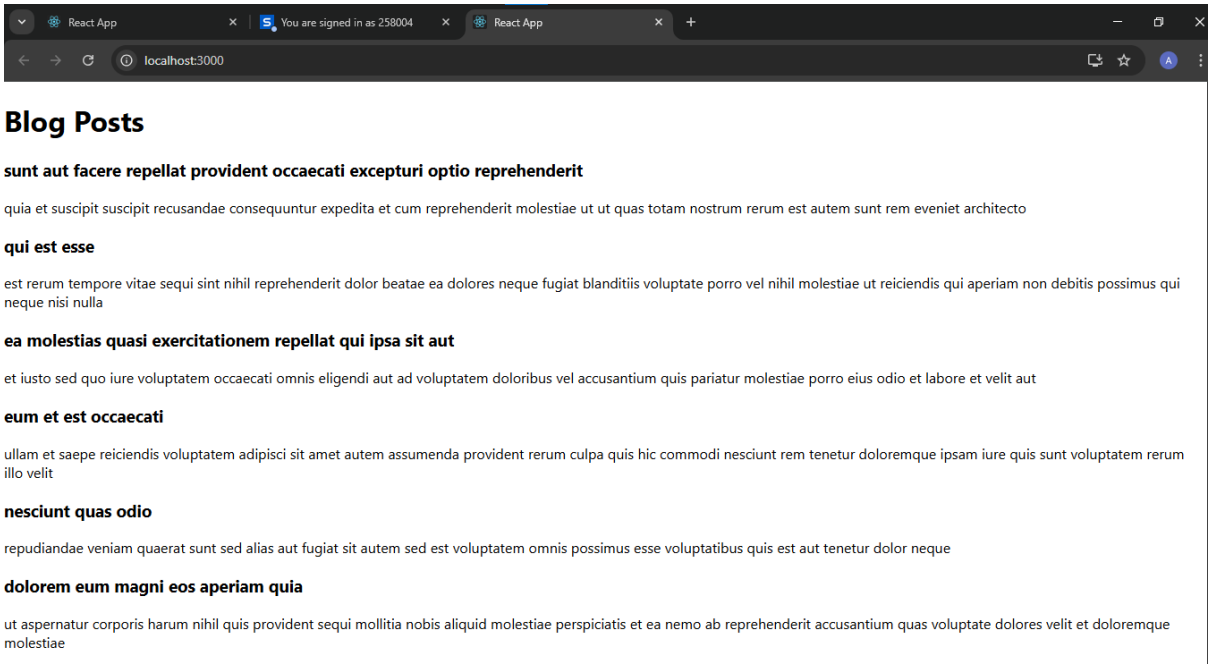
```
}

export default Post;
```

Post.js

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

import Post from './Post';

class Posts extends Component {
  constructor(props) {
    super(props);
    this.state = {
      posts: [],
      hasError: false,
    };
  }

  loadPosts = () => {
    fetch('https://jsonplaceholder.typicode.com/posts')
      .then(response => response.json())
      .then(data => {
        const postObjects = data.map(p => new Post(p.userId, p.id, p.title,
p.body));
        this.setState({ posts: postObjects });
      })
      .catch(error => {
        this.setState({ hasError: true });
        console.error("Error fetching posts:", error);
      });
  };

  componentDidMount() {
    this.loadPosts();
  }

  componentDidCatch(error, info) {
    alert("Something went wrong: " + error.toString());
    this.setState({ hasError: true });
  }

  render() {
    if (this.state.hasError) {
      return <h2>Unable to load posts. Please try again later.</h2>;
    }

    return (
      <div>
```

```
        <h1>Blog Posts</h1>
        {this.state.posts.map(post => (
          <div key={post.id} style={{ marginBottom: '20px' }}>
            <h3>{post.title}</h3>
            <p>{post.body}</p>
          </div>
        ))}
      </div>
    );
  }
}

export default Posts;
```

Output:



05-REACTJS-HANDSON

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

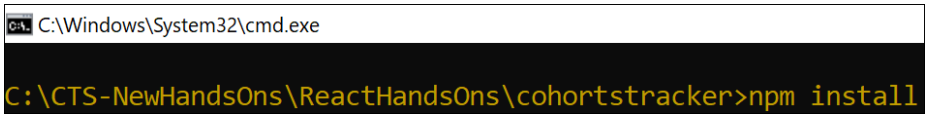


Figure 7: Restore packages

4. Open the application using VS Code
5. Create a new CSS Module in a file called “CohortDetails.module.css”
6. 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
7. Define a css style for html <dt> element using tag selector. Set the font weight to 500.
8. Open the cohort details component and import the CSS Module
9. Apply the box class to the container div
10. Define the style for <h3> element to use “green” color font when cohort status is “ongoing” and “blue” color in all other scenarios.
11. Final result should look similar to the below image

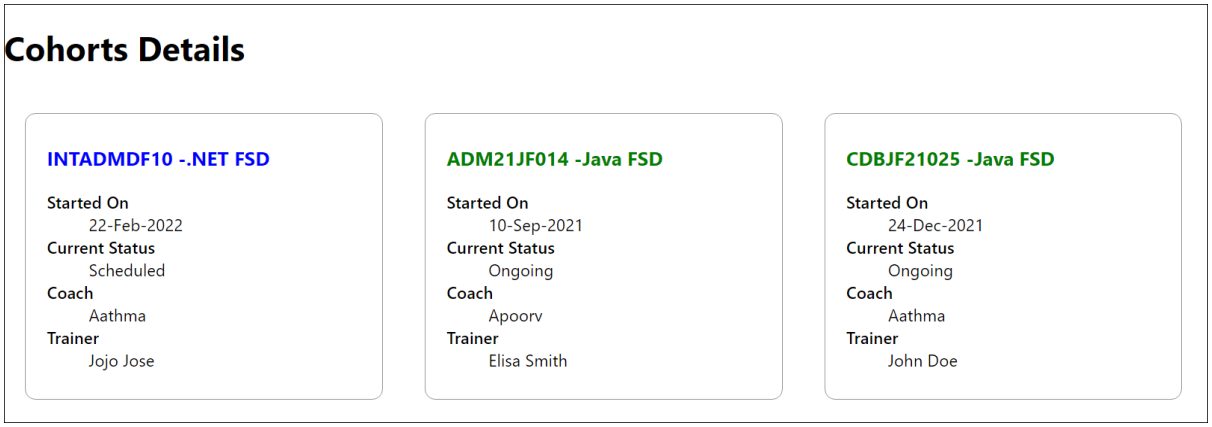


Figure 8: Final Result

Code:

App.js

```
import React from 'react';

import CohortDetails from './Cohorts/CohortDetails';

function App() {
  const cohorts = [
    {
      name: 'React Bootcamp',
      startDate: '2025-07-01',
      endDate: '2025-08-01',
      status: 'ongoing',
    },
    {
      name: 'Python Basics',
      startDate: '2025-06-01',
      endDate: '2025-06-30',
      status: 'completed',
    },
  ];

  return (
    <div className="App">
      <h2>My Academy Dashboard</h2>
      {cohorts.map((cohort, index) => (
        <CohortDetails key={index} cohort={cohort} />
      ))}
    </div>
  );
}

export default App;
```

CohortDetailsModule.css

```css
.box {
  width: 300px;
  display: inline-block;
  margin: 10px;
  padding: 10px 20px;
  border: 1px solid black;
  border-radius: 10px;
}

dt {
  font-weight: 500;
}
```

CohortDetails.js

```jsx
import React from 'react';
import styles from './CohortDetails.module.css';

function CohortDetails({ cohort }) {
  const titleStyle = {
    color: cohort.status === 'ongoing' ? 'green' : 'blue',
  };

  return (
    <div className={styles.box}>
      <h3 style={titleStyle}>{cohort.name}</h3>
      <dl>
        <dt>Start Date:</dt>
        <dd>{cohort.startDate}</dd>
        <dt>End Date:</dt>
        <dd>{cohort.endDate}</dd>
        <dt>Status:</dt>
        <dd>{cohort.status}</dd>
      </dl>
    </div>
  );
}

export default CohortDetails;
```

Output:

My Academy Dashboard

React Bootcamp

Start Date:
2025-07-01
End Date:
2025-08-01
Status:
ongoing

Python Basics

Start Date:
2025-06-01
End Date:
2025-06-30
Status:
completed