Week-6

# React

## Exercise-1

**App.js**

import logo from './logo.svg';

import './App.css';

function App() {

  return (

    <h1 id='title'>Welcome the first session of React</h1>

  );

}

export default App;

**index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

  <React.StrictMode>

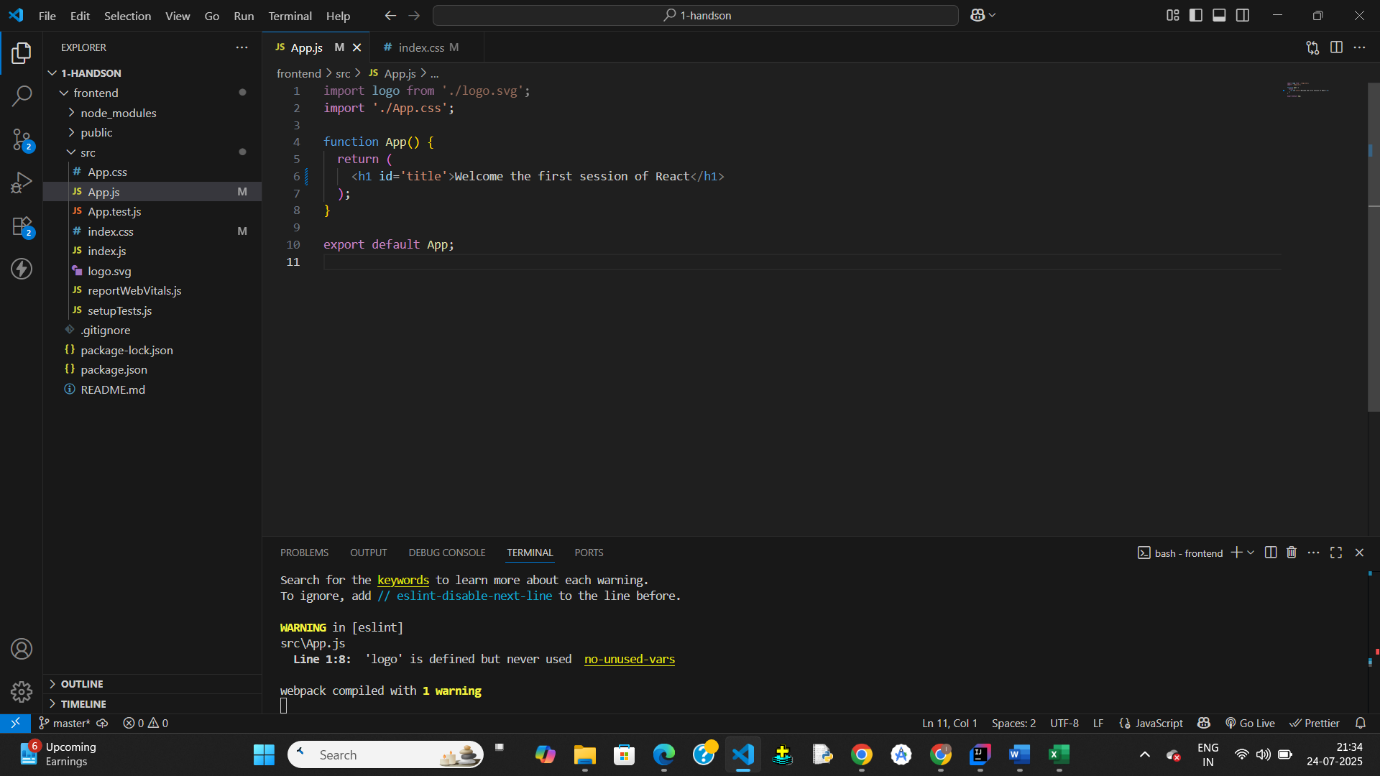
    <App />

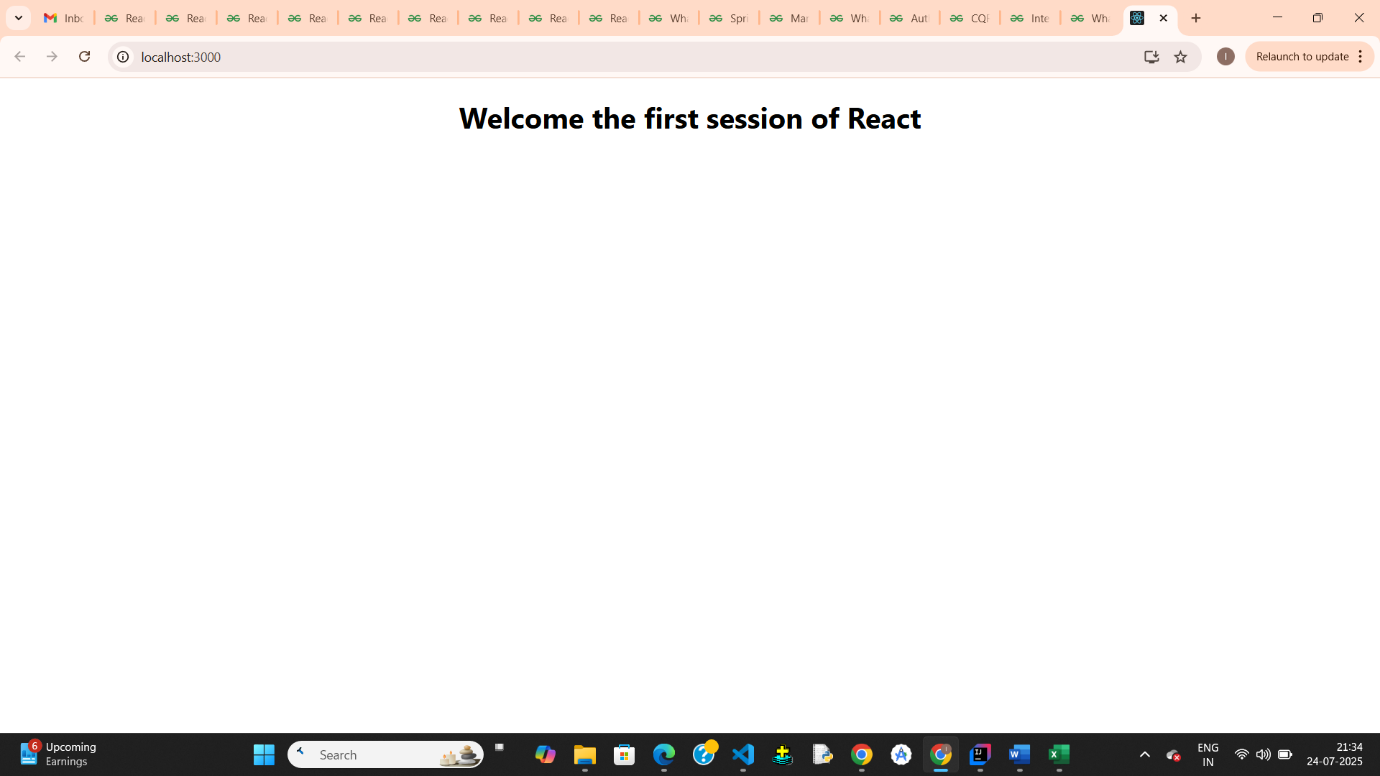
  </React.StrictMode>

);

reportWebVitals();

**Output:**

****



## Exercise-2

**App.js**

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>

      <Home></Home>

      <About></About>

      <Contact></Contact>

    </div>

  );

}

export default App;

**Home.js**

function Home()

{

    return(

        <div>

            <h1>Welcome to Homepage of student management system</h1>

        </div>

    )

}

export default Home;

**About.js**

function About()

{

    return(

        <div>

            <h1>Welcome to AboutPage of student management system</h1>

        </div>

    )

}

export default About;

**Contact.js**

function Contact()

{

    return(

        <div>

            <h1>Welcome to ContactPage of student management system</h1>

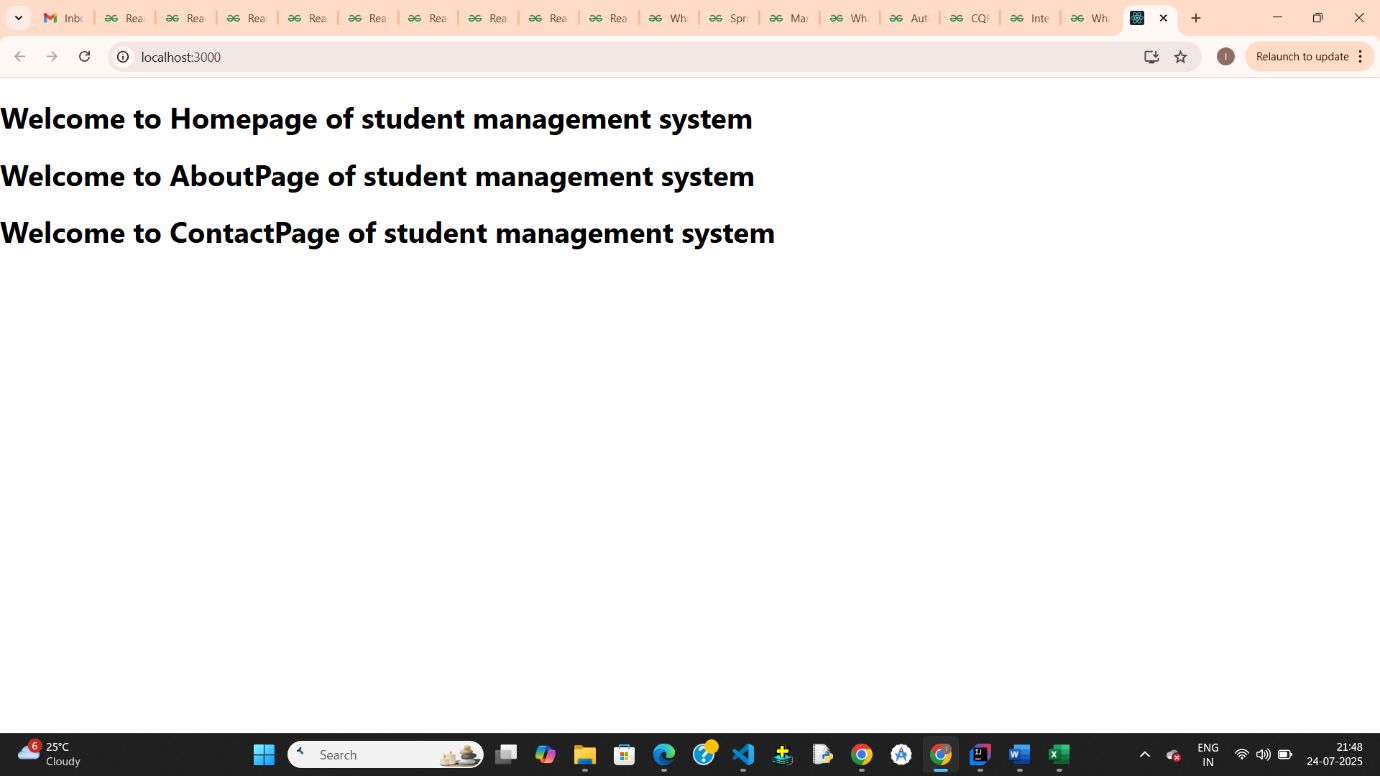
        </div>

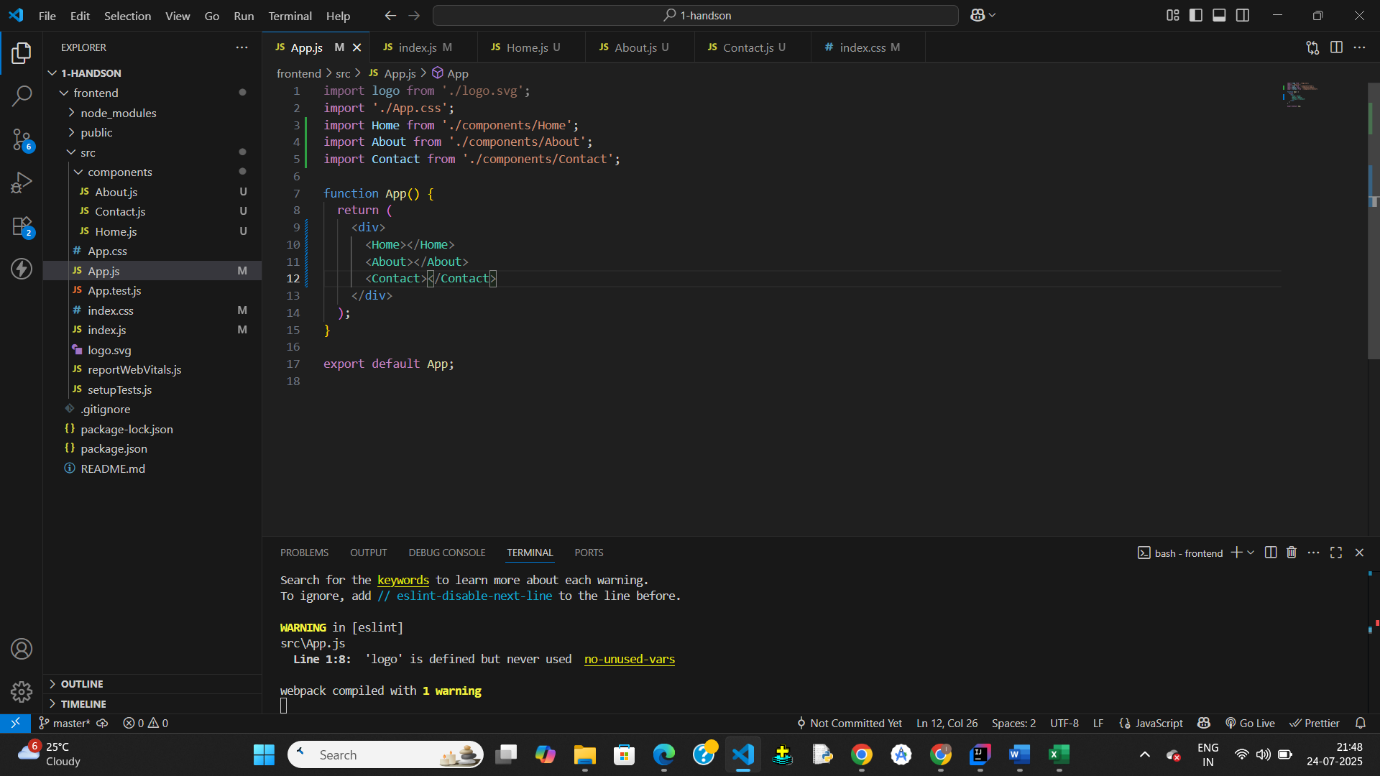
    )

}

export default Contact;

**Output:**

****

****

## Exercise-3

**App.js**

// src/App.js

import React from 'react';

import CalculateScore from './components/CalculateScore'

function App() {

  return (

    <div className="App">

      <CalculateScore name="Iswarya Gali" school="GPREC" total={450} goal={5} />

    </div>

  );

}

export default App;

**CalculateScore.js**

// src/Components/CalculateScore.js

import React from 'react';

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

function CalculateScore({ name, school, total, goal }) {

  const average = total / goal;

  return (

    <div className="score-container">

      <h2>Student Score Details</h2>

      <p><strong>Name:</strong> {name}</p>

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

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

      <p><strong>Subjects (Goal):</strong> {goal}</p>

      <p className="average"><strong>Average Score:</strong> {average.toFixed(2)}</p>

    </div>

  );

}

export default CalculateScore;

**Style.css**

/\* src/Stylesheets/mystyle.css \*/

.score-container {

  background-color: #f2f2f2;

  border: 2px solid #ddd;

  padding: 20px;

  margin: 30px auto;

  width: 50%;

  border-radius: 8px;

  box-shadow: 0 0 10px #ccc;

  font-family: Arial, sans-serif;

}

.score-container h2 {

  color: #3f51b5;

  text-align: center;

}

.score-container p {

  font-size: 18px;

  margin: 10px 0;

}

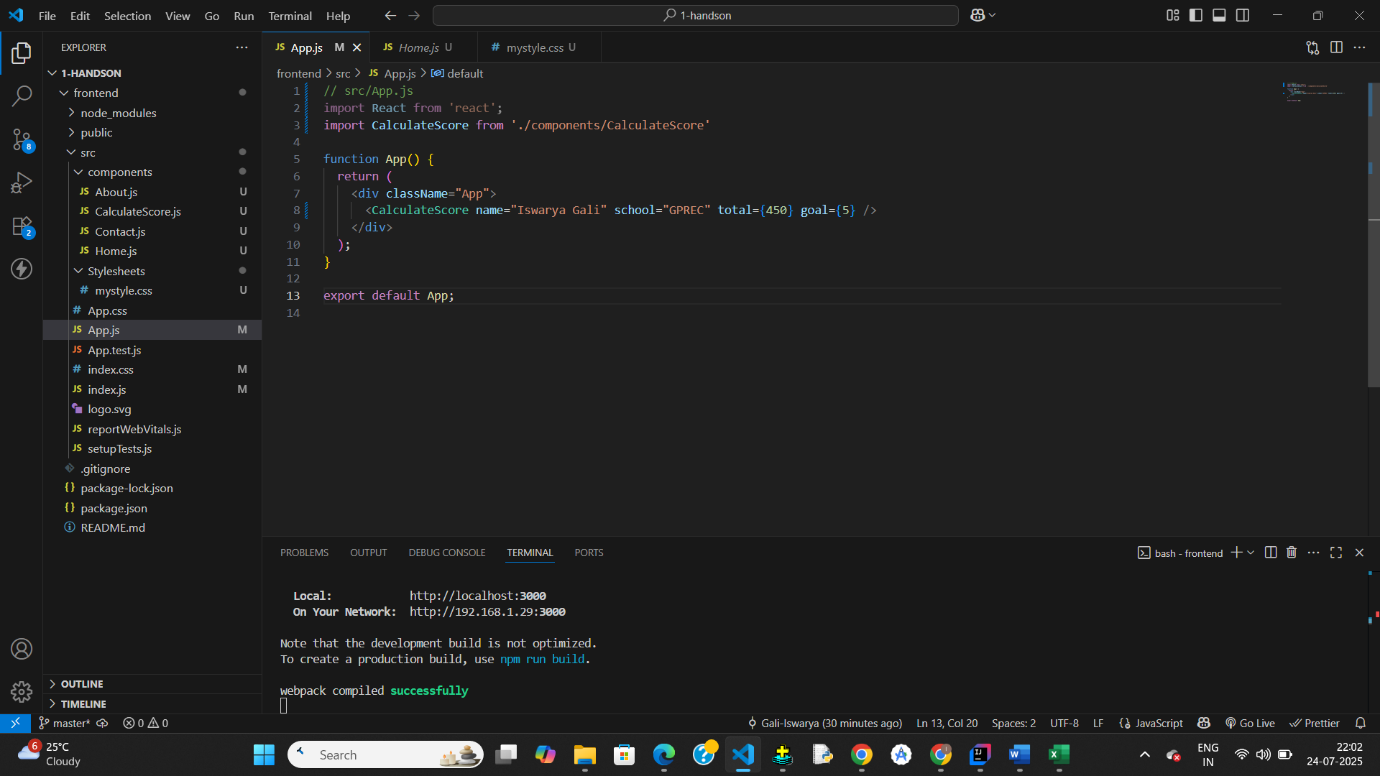
.average {

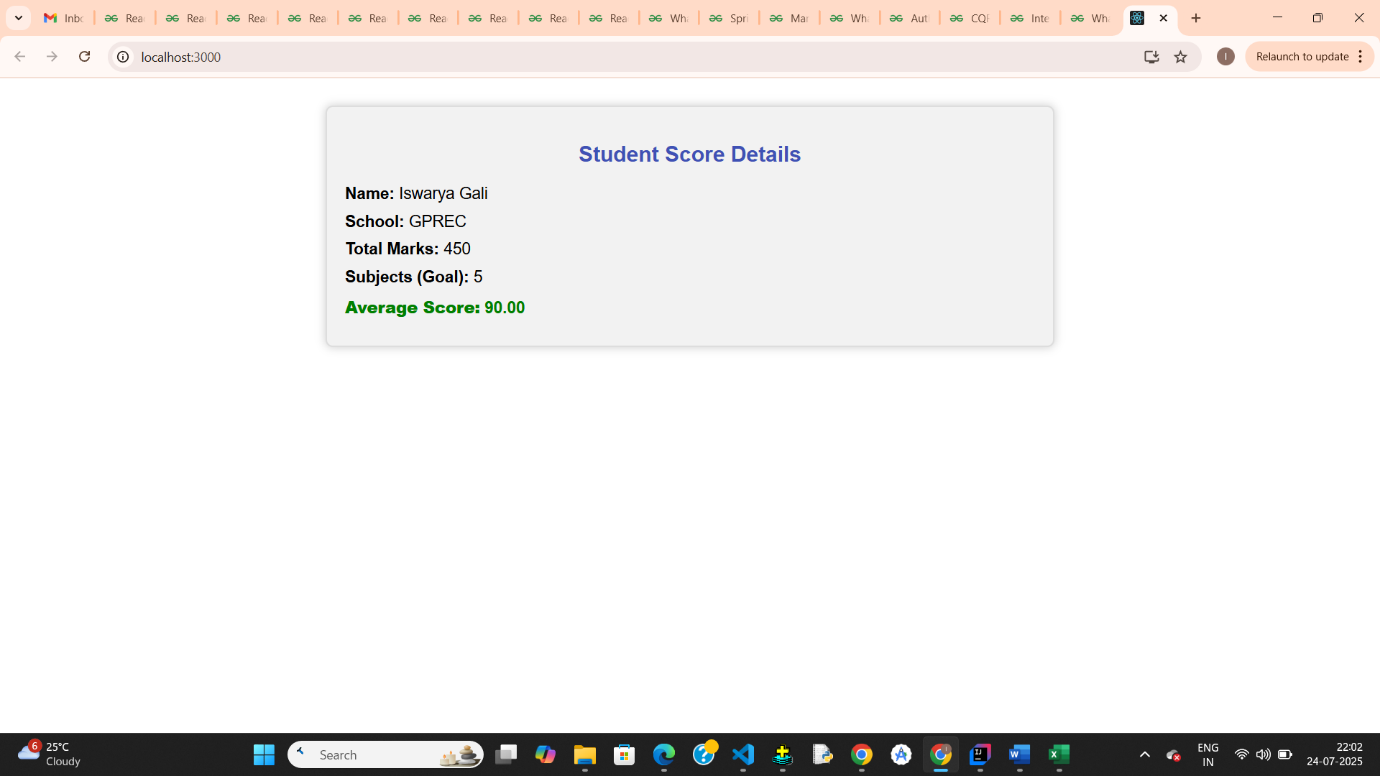
  color: green;

  font-weight: bold;

}

**Output:**

****

****

# Exercise-4

**App.js**

// src/App.js

import React from 'react';

import Posts from './Posts';

function App() {

  return (

    <div className="App">

      <Posts />

    </div>

  );

}

export default App;

**Post.js**

// src/Post.js

import React from 'react';

class Post extends React.Component {

  render() {

    const { title, body } = this.props;

    return (

      <div style={{ marginBottom: "20px" }}>

        <h2>{title}</h2>

        <p>{body}</p>

      </div>

    );

  }

}

export default Post;

**Posts.js**

// src/Posts.js

import React from 'react';

import Post from './Post';

class Posts extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      posts: [],

      hasError: false

    };

  }

  // Method to fetch posts from API

  loadPosts = () => {

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

      .then(res => res.json())

      .then(data => this.setState({ posts: data }))

      .catch(err => {

        throw new Error('Fetching failed');

      });

  };

  // Lifecycle method for data fetch

  componentDidMount() {

    this.loadPosts();

  }

  // Error boundary method

  componentDidCatch(error, info) {

    this.setState({ hasError: true });

    alert("An error occurred: " + error.message);

  }

  render() {

    if (this.state.hasError) {

      return <h1>Something went wrong while loading posts.</h1>;

    }

    return (

      <div style={{ padding: "30px" }}>

        <h1>Blog Posts</h1>

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

          <Post key={post.id} title={post.title} body={post.body} />

        ))}

      </div>

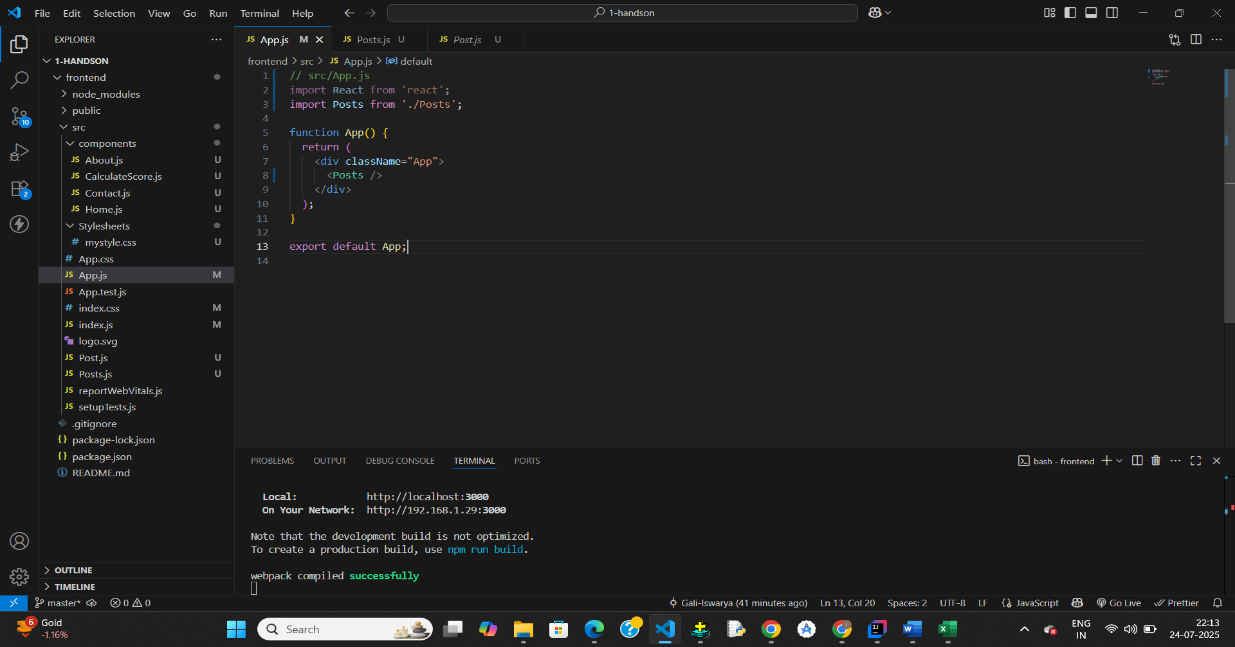
    );

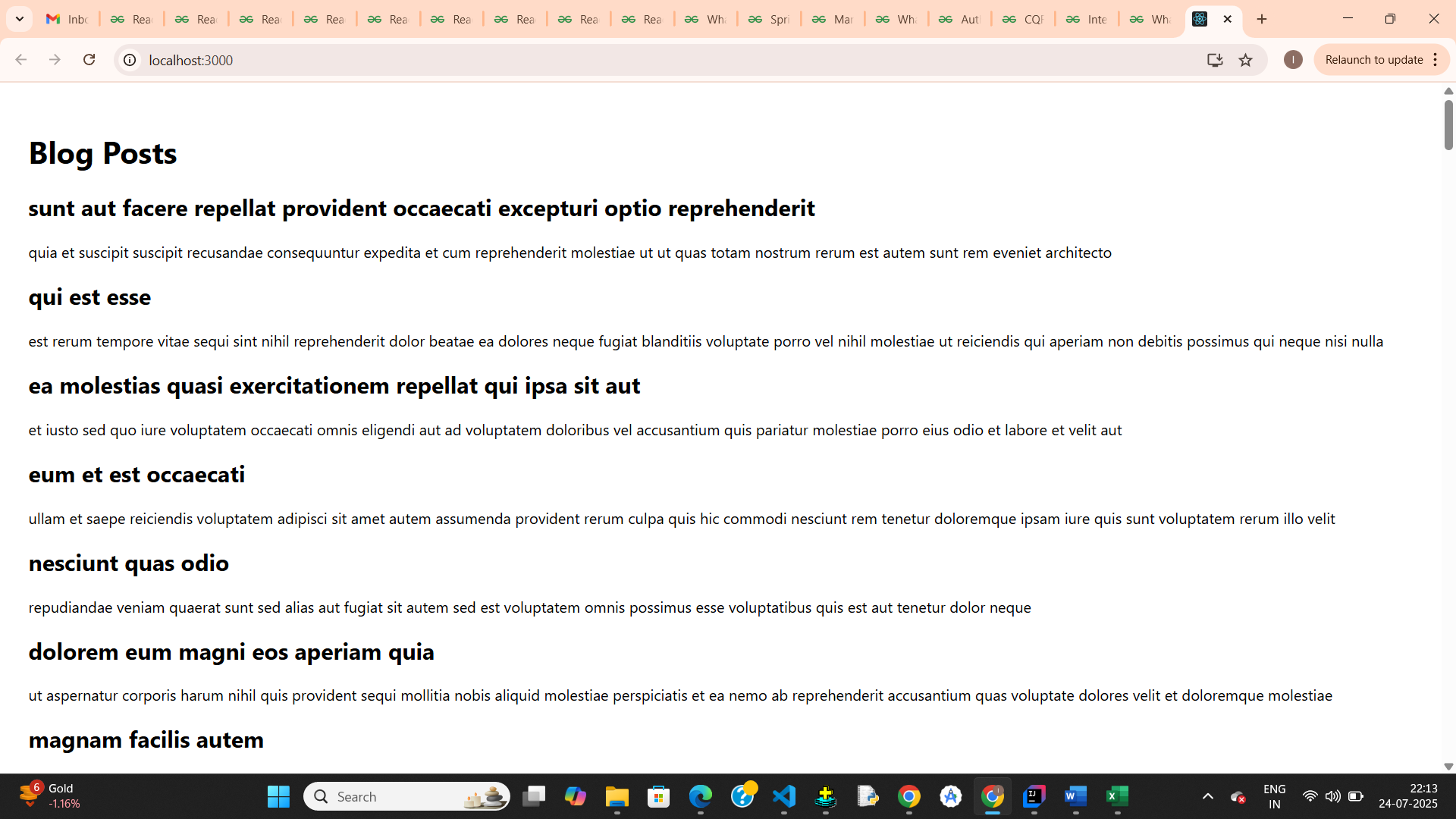
  }

}

export default Posts;

**Output:**

****

****

# Exercise-5

**App.js**

// src/App.js or src/components/Dashboard.js

import React from 'react';

import CohortDetails from './components/CohortDetails';

function App() {

const cohorts = [

{ name: 'React Bootcamp', status: 'Ongoing', duration: '6 Weeks', trainer: 'John' },

{ name: 'Java Fundamentals', status: 'Completed', duration: '4 Weeks', trainer: 'Alice' },

];

return (

<div>

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

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

))}

</div>

);

}

export default App;

**CohortDetails.js**

// src/components/CohortDetails.js

import React from 'react';

import styles from './CohortDetails.module.css';

function CohortDetails({ cohort }) {

  const { name, status, duration, trainer } = cohort;

  // Conditional inline style for heading color

  const titleStyle = {

    color: status.toLowerCase() === 'ongoing' ? 'green' : 'blue',

  };

  return (

    <div className={styles.box}>

      <h3 style={titleStyle}>{name}</h3>

      <dl>

        <dt>Status:</dt>

        <dd>{status}</dd>

        <dt>Duration:</dt>

        <dd>{duration}</dd>

        <dt>Trainer:</dt>

        <dd>{trainer}</dd>

      </dl>

    </div>

  );

}

export default CohortDetails;

CohortDetails**.module.css**

/\* src/components/CohortDetails.module.css \*/

.box {

  width: 300px;

  display: inline-block;

  margin: 10px;

  padding: 10px 20px;

  border: 1px solid black;

  border-radius: 10px;

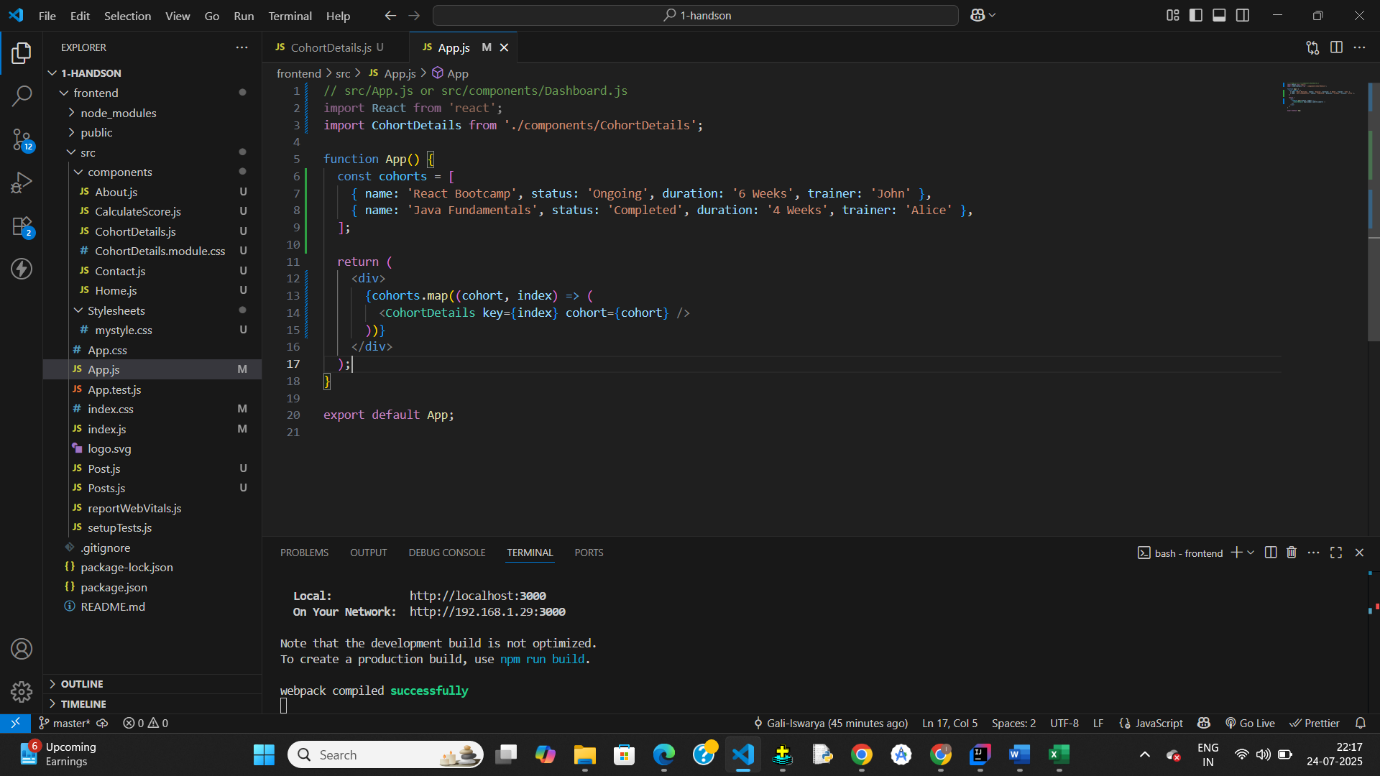
}

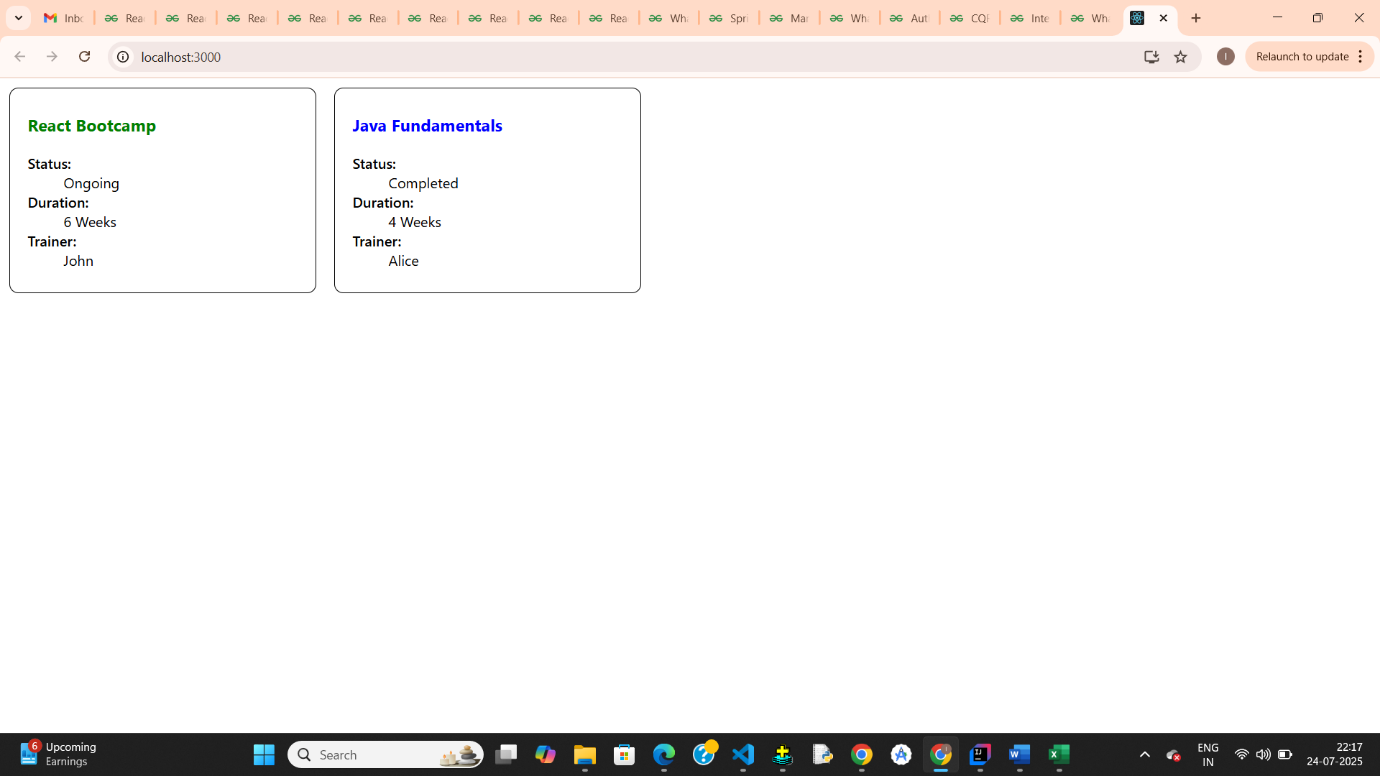
dt {

  font-weight: 500;

}

**Output:**

****

****