**Skill:-React**

**Filename:- 9. ReactJS-HOL**

**Solution:-**

**1. List the features of ES6**

**Answer:-**

- let and const  
- Arrow functions  
- Template literals  
- Classes  
- Modules  
- Default parameters  
- Destructuring  
- Promises  
- Spread and rest operators

**2. Explain JavaScript let**

**Answer:-**

let declares a block-scoped variable, meaning it's only accessible within the block it's defined.

**3. Identify the differences between var and let**

**Answer:-**

var is function-scoped, while `let` is block-scoped. let does not allow re-declaration in the same scope, while var does.

**4. Explain JavaScript const**

**Answer:-**

const declares a constant, block-scoped variable whose value cannot be reassigned.

**5. Explain ES6 class fundamentals**

**Answer:-**

ES6 classes are syntactic sugar over JavaScript's prototype-based inheritance. They use the class keyword and contain constructor and methods.

**6. Explain ES6 class inheritance**

**Answer:-**

Use the extends keyword to inherit from another class and super() to call the parent constructor.

**7. Define ES6 arrow functions**

**Answer:-**

Arrow functions provide a shorter syntax and do not have their own this, arguments, or super.

**8. Identify set(), map()**

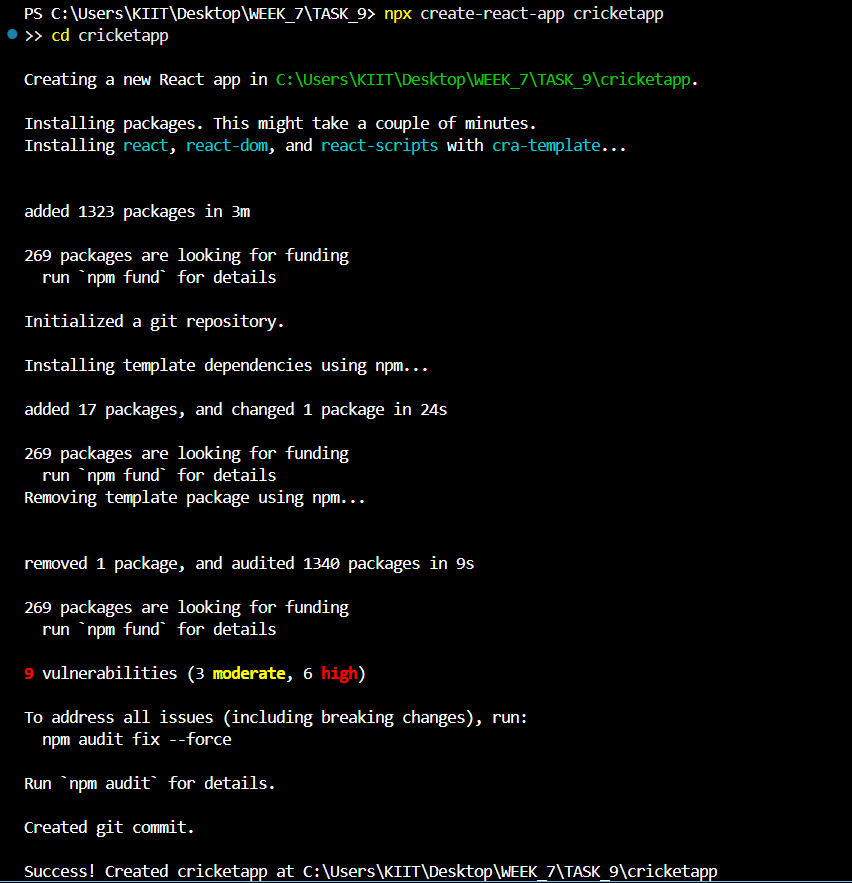
**Answer:-**

Set stores unique values. Map stores key-value pairs where keys can be any type.

***Created a New React App***

***Command:-* npx create-react-app cricketapp**

**cd cricketapp**



***Go to src/ Folder Create 2 New Components thar are:***

1. ***Created a New ListofPlayers.js and below is the code for it.***

import React from 'react';

function ListofPlayers() {

  const players = [

    { name: 'Player 1', score: 95 },

    { name: 'Player 2', score: 45 },

    { name: 'Player 3', score: 75 },

    { name: 'Player 4', score: 88 },

    { name: 'Player 5', score: 62 },

    { name: 'Player 6', score: 30 },

    { name: 'Player 7', score: 91 },

    { name: 'Player 8', score: 52 },

    { name: 'Player 9', score: 100 },

    { name: 'Player 10', score: 77 },

    { name: 'Player 11', score: 64 }

  ];

  const lowScorers = players.filter(p => p.score < 70);

  return (

    <div>

      <h2>All Players</h2>

      {players.map((player, index) => (

        <p key={index}>{player.name} - {player.score}</p>

      ))}

      <h2>Players with score below 70</h2>

      {lowScorers.map((player, index) => (

        <p key={index}>{player.name} - {player.score}</p>

      ))}

    </div>

  );

}

export default ListofPlayers;

1. ***Created a New IndianPlayers.js and below is the code for it.***

import React from 'react';

function IndianPlayers() {

  const oddPlayers = ['Player 1', 'Player 3', 'Player 5', 'Player 7', 'Player 9', 'Player 11'];

  const evenPlayers = ['Player 2', 'Player 4', 'Player 6', 'Player 8', 'Player 10'];

  const [T20players] = [['Kohli', 'Rohit']];

  const [RanjiTrophyPlayers] = [['Player A', 'Player B']];

  const mergedPlayers = [...T20players, ...RanjiTrophyPlayers];

  return (

    <div>

      <h2>Odd Team Players</h2>

      {oddPlayers.map((p, i) => <p key={i}>{p}</p>)}

      <h2>Even Team Players</h2>

      {evenPlayers.map((p, i) => <p key={i}>{p}</p>)}

      <h2>Merged Players</h2>

      {mergedPlayers.map((p, i) => <p key={i}>{p}</p>)}

    </div>

  );

}

export default IndianPlayers;

***Update the code of App.js in src/App.js (if const flag = true)***

import React from 'react';

import ListofPlayers from './ListofPlayers';

import IndianPlayers from './IndianPlayers';

function App() {

  const flag = true;

  return (

    <div className="App">

      {flag ? <ListofPlayers /> : <IndianPlayers />}

    </div>

  );

}

export default App;

***Update the code of App.js in src/App.js (if const flag = false)***

import React from 'react';

import ListofPlayers from './ListofPlayers';

import IndianPlayers from './IndianPlayers';

function App() {

  const flag = false;

  return (

    <div className="App">

      {flag ? <ListofPlayers /> : <IndianPlayers />}

    </div>

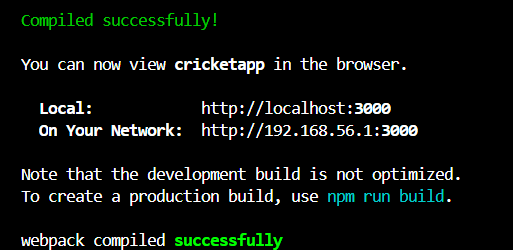
  );

}

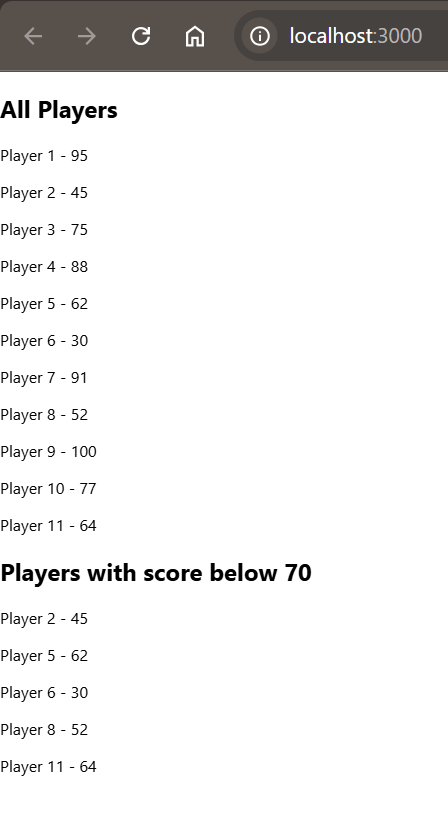
export default App;

***Starting the React Application on local Server by* npm start**

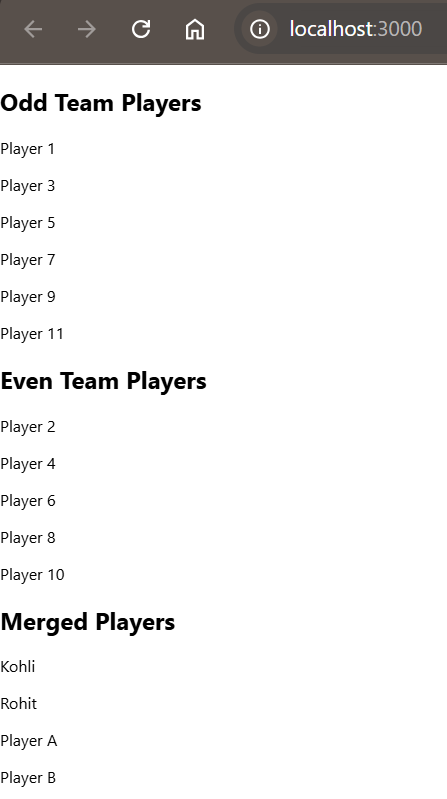
***Output:-***



***Output on Local Server localhost:3000 (when const flag = true)***



***Output on Local Server localhost:3000(when const flag = false)***



**Skill:-React**

**Filename:- 10. ReactJS-HOL**

**Solution:-**

**1. Define JSX**

**Answer:-**

JSX stands for JavaScript XML. It allows writing HTML-like syntax in JavaScript, which React transforms into elements.

**2. Explain about ECMA Script**

**Answer:-**

ECMAScript is the standard specification that JavaScript follows. ES6 refers to the 6th edition of ECMAScript.

**3. Explain React.createElement()**

**Answer:-**

React.createElement() is a method to create React elements manually without using JSX.

**4. Explain how to create React nodes with JSX**

**Answer:-**

You create React nodes by writing HTML-like tags inside JavaScript. JSX is then transpiled into React.createElement() calls.

**5. Define how to render JSX to DOM**

**Answer:-**

Use ReactDOM.render(element, container) to render JSX into a real DOM node.

**6. Explain how to use JavaScript expressions in JSX**

**Answer:-**

Wrap JavaScript expressions inside {} inside JSX, e.g., <p>{name}</p>.

**7. Explain how to use inline CSS in JSX**

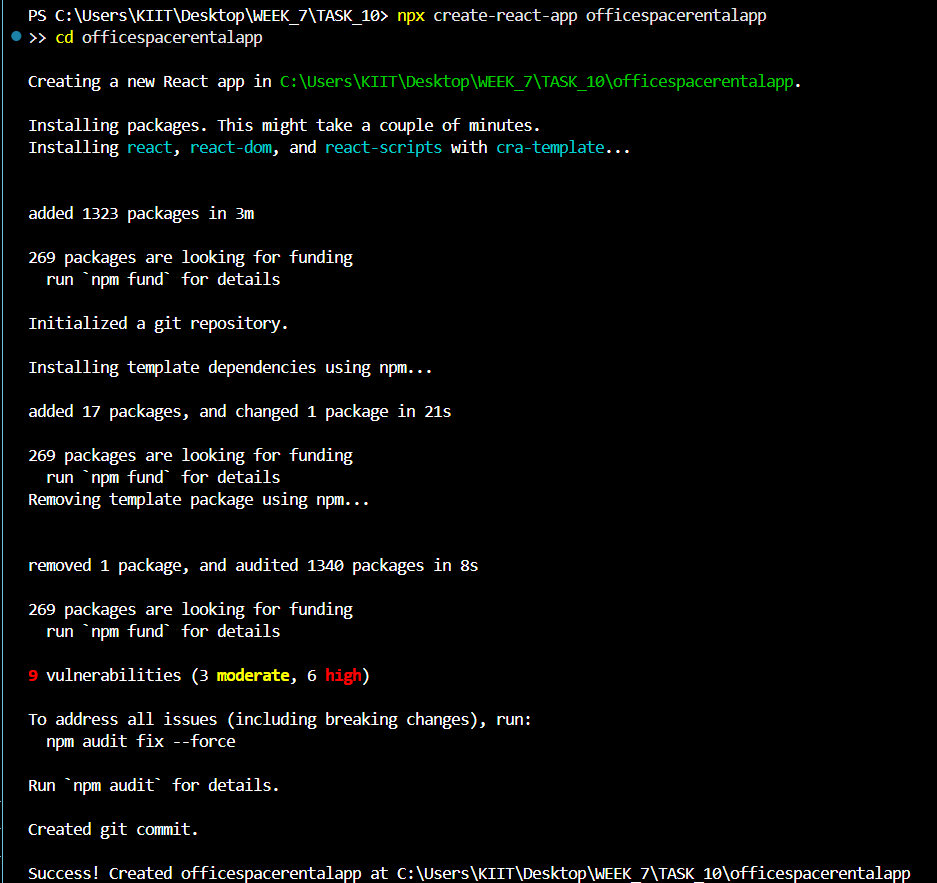
**Answer:-**

Use the style attribute and pass a JavaScript object with camelCase properties, e.g., <div style={{backgroundColor: 'blue'}}></div>.

***Created a New React App***

***Command:-* npx create-react-app officespacerentalapp**

**cd officespacerentalapp**



***Update the code of App.js in src/App.js***

import React from 'react';

function App() {

  const office1 = {

    name: "Orchid Workspace",

    rent: 75000,

    address: "BTM Layout, Bangalore"

  };

  const officeList = [

    {

      name: "Global Tech Hub",

      rent: 50000,

      address: "HSR Layout, Bangalore"

    },

    {

      name: "Innovation Tower",

      rent: 85000,

      address: "Indiranagar, Bangalore"

    }

  ];

  const getRentStyle = (rent) => {

    return {

      color: rent < 60000 ? 'red' : 'green'

    };

  };

  return (

    <div>

      <h1>Office Space Rental</h1>

      <img src="office1.webp" alt="Office" width="300" />

      <h2>{office1.name}</h2>

      <p style={getRentStyle(office1.rent)}>Rent: ₹{office1.rent}</p>

      <p>Address: {office1.address}</p>

      <h2>Other Office Spaces</h2>

      {officeList.map((office, index) => (

        <div key={index}>

          <h3>{office.name}</h3>

          <p style={getRentStyle(office.rent)}>Rent: ₹{office.rent}</p>

          <p>Address: {office.address}</p>

        </div>

      ))}

    </div>

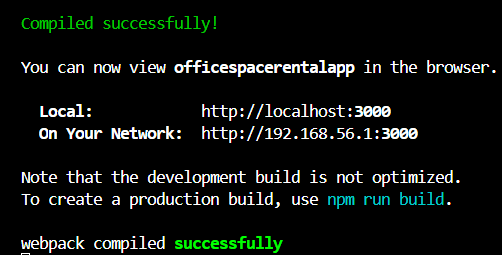
  );

}

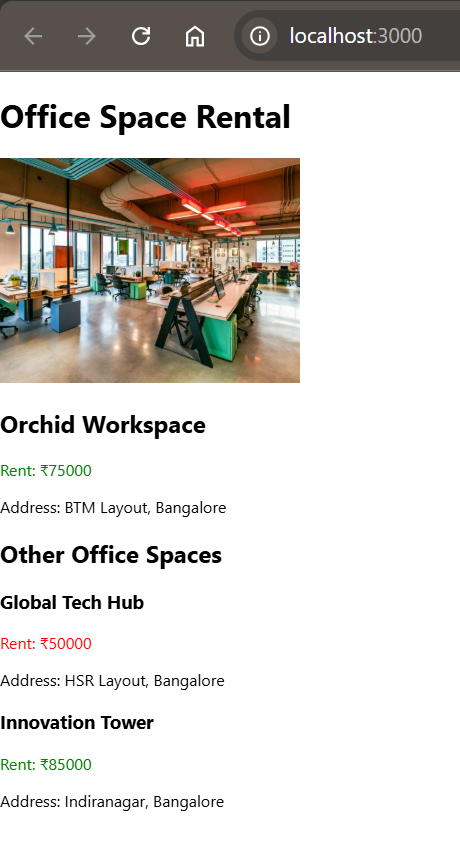
export default App;

***Starting the React Application on local Server by* npm start**

***Output:-***



***Output on Local Server localhost:3000***



**Skill:-React**

**Filename:- 11. ReactJS-HOL**

**Solution:-**

**1. Explain React events**

**Answer:-**

React handles events similarly to DOM events but uses camel-case syntax.

**2. Explain about event handlers**

**Answer:-**

Event handlers are functions that handle user interactions like clicks or keypresses in React.

**3. Define Synthetic event**

**Answer:-**

A SyntheticEvent is a cross-browser wrapper around native events provided by React.

**4. Identify React event naming convention**

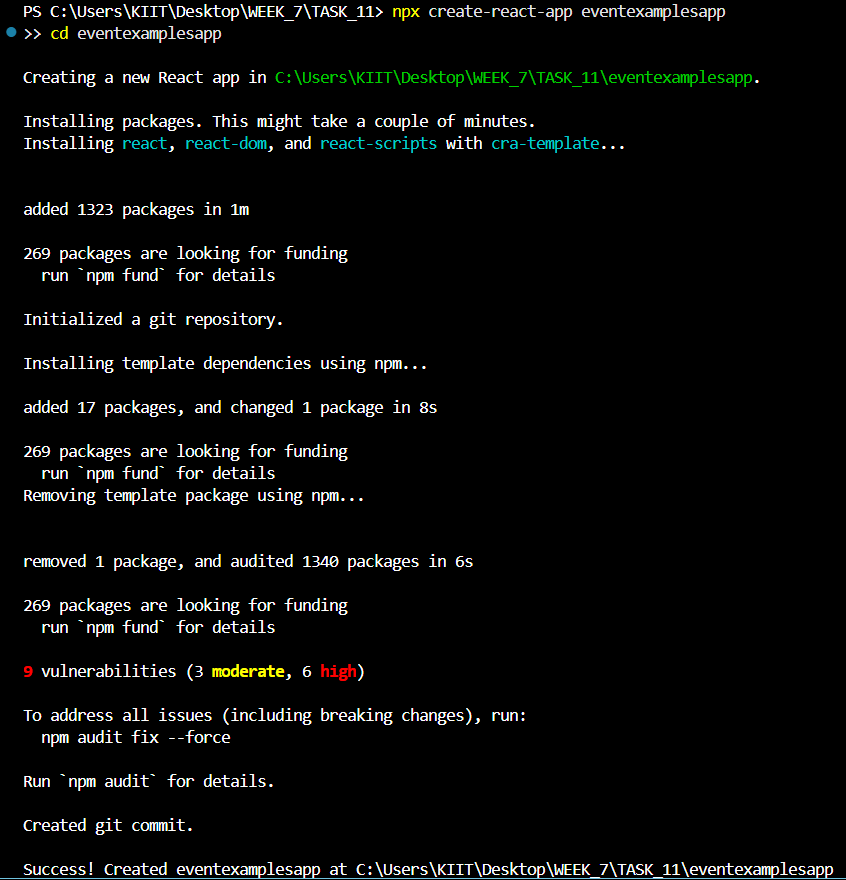
**Answer:-**

React uses camelCase for event names, e.g., `onClick`, `onChange`.

***Created a New React App***

***Command:-* npx create-react-app eventexamplesapp**

**cd eventexamplesapp**



***Update the code of App.js in src/App.js***

import React, { useState } from 'react';

import CurrencyConvertor from './CurrencyConvertor';

function App() {

  const [count, setCount] = useState(0);

  const sayHello = () => {

    console.log("Hello - This is a static message");

  };

  const increment = () => {

    setCount(count + 1);

    sayHello();

  };

  const decrement = () => {

    setCount(count - 1);

  };

  const sayWelcome = (message) => {

    alert(message);

  };

  const handleSyntheticEvent = (event) => {

    alert("I was clicked");

  };

  return (

    <div>

      <h1>React Event Handling</h1>

      <h2>Counter: {count}</h2>

      <button onClick={increment}>Increment</button>

      <button onClick={decrement}>Decrement</button>

      <br /><br />

      <button onClick={() => sayWelcome("Welcome")}>Say Welcome</button>

      <br /><br />

      <button onClick={handleSyntheticEvent}>OnPress</button>

      <br /><br />

      <CurrencyConvertor />

    </div>

  );

}

export default App;

***Created a New File Name as CurrencyConvertor.js in src/and below is the code for it.***

import React, { useState } from 'react';

function CurrencyConvertor() {

  const [rupees, setRupees] = useState('');

  const [euro, setEuro] = useState('');

  const handleSubmit = () => {

    const converted = parseFloat(rupees) / 90;

    setEuro(converted.toFixed(2));

  };

  return (

    <div>

      <h2>Currency Convertor</h2>

      <input

        type="text"

        value={rupees}

        onChange={(e) => setRupees(e.target.value)}

        placeholder="Enter amount in INR"

      />

      <button onClick={handleSubmit}>Convert</button>

      <p>Amount in Euro: €{euro}</p>

    </div>

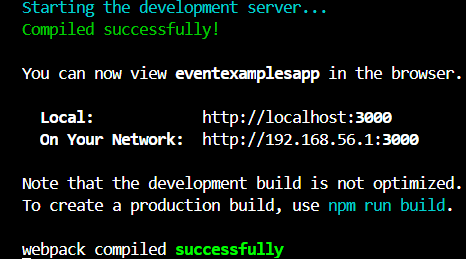
  );

}

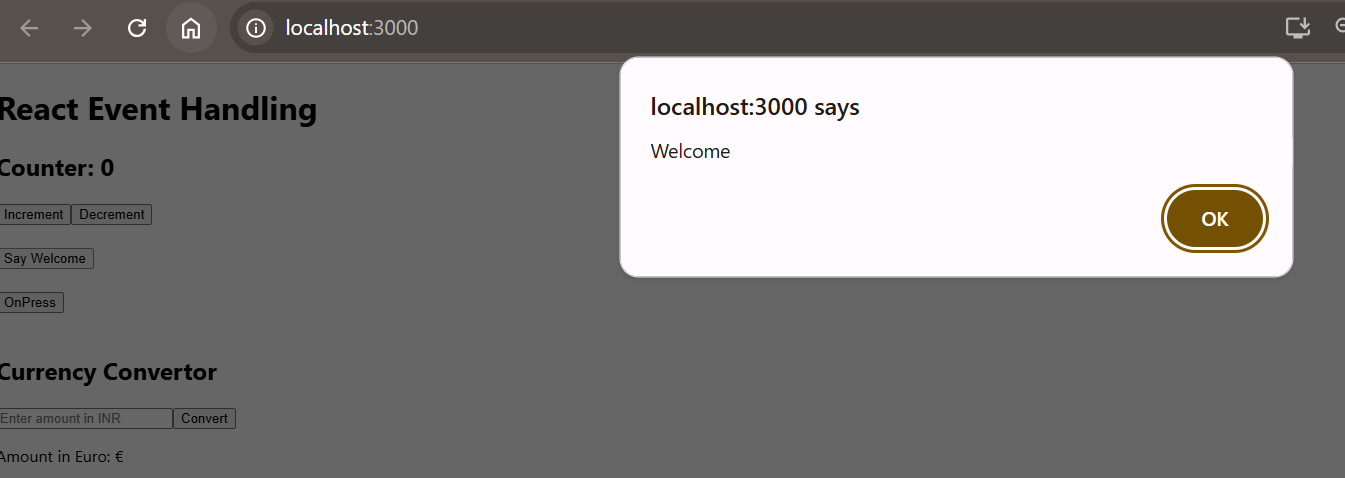
export default CurrencyConvertor;

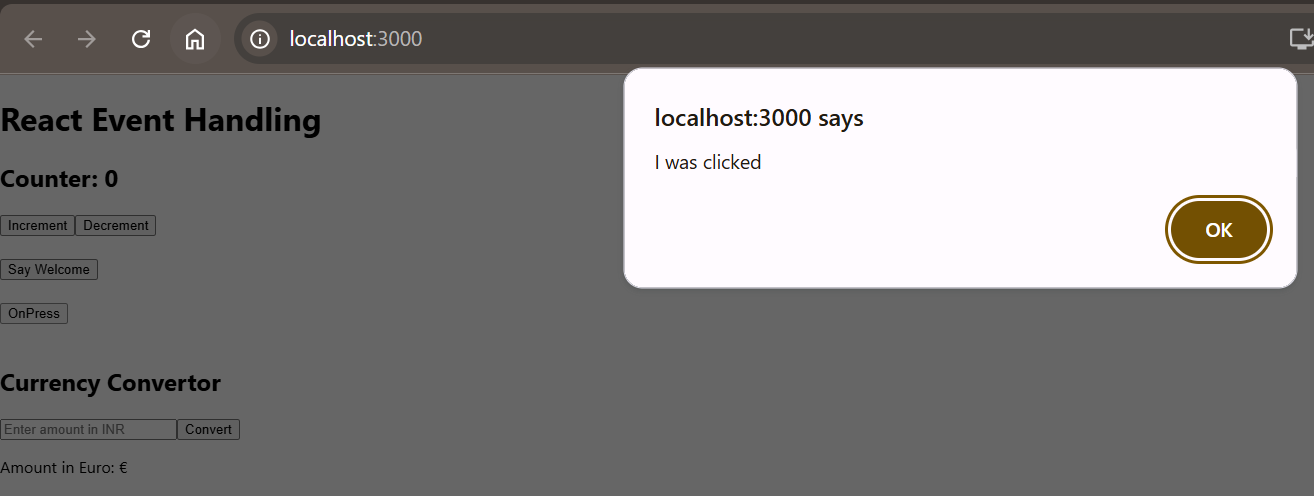
***Starting the React Application on local Server by* npm start**

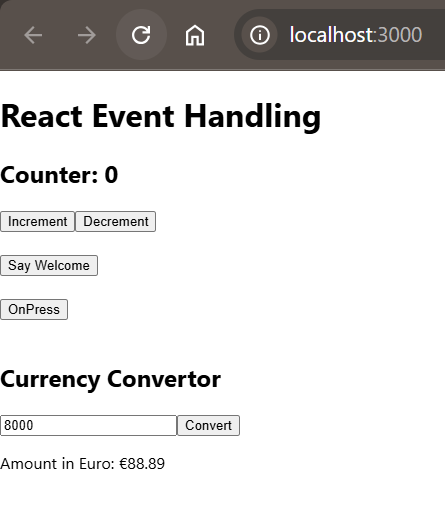
***Output:-***



***Output on Local Server localhost:3000***







**Skill:-React**

**Filename:- 12. ReactJS-HOL**

**Solution:-**

**1. Explain about conditional rendering in React**

**Answer:-**

Conditional rendering lets you show elements based on conditions using `if`, ternary, or logical `&&` operators.

**2. Define element variables**

**Answer:-**

Element variables store elements in variables, allowing dynamic control over rendering.

**3. Explain how to prevent components from rendering**

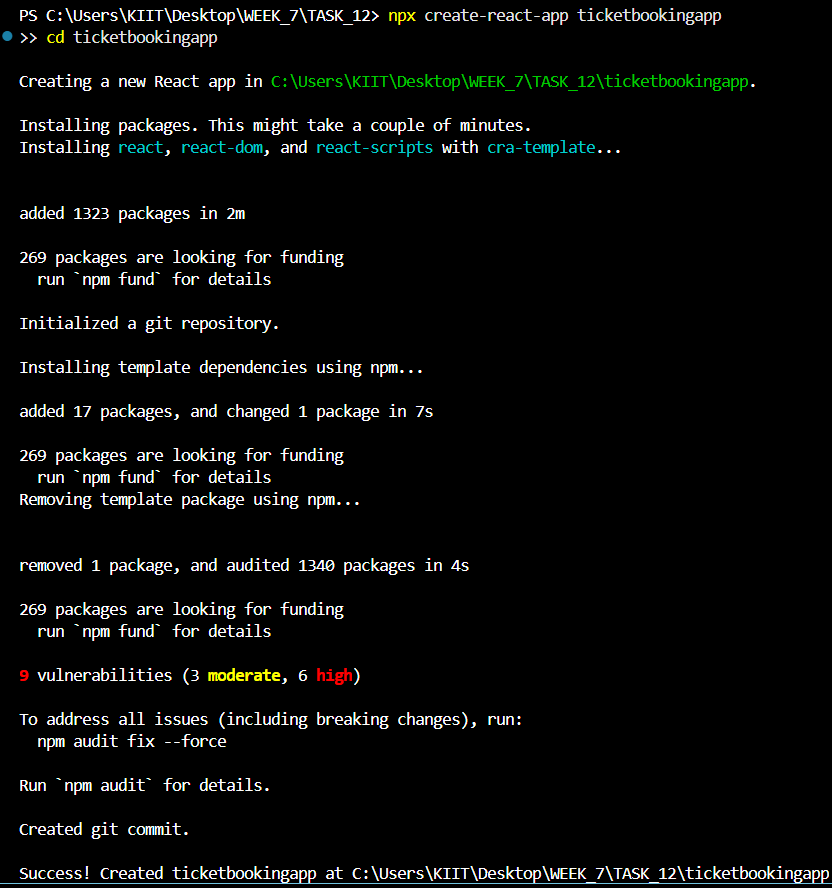
**Answer:-**

Return `null` in the component’s render method to prevent rendering.

***Created a New React App***

***Command:-* npx create-react-app ticketbookingapp**

**cd ticketbookingapp**



***Go to src/ Folder Create 2 New Components thar are:***

1. ***Created a New Guest.js and below is the code for it.***

import React from 'react';

function Guest() {

  return (

    <div>

      <h2 style={{ color: 'green' }}>Flight Details</h2>

      <p>Guests can only view the available flights.</p>

      <p>Flight: Air India AI-101</p>

      <p>Time: 1:45 PM</p>

      <p>From: Delhi To: Chennai</p>

    </div>

  );

}

export default Guest;

1. ***Created a New File User.js and below is the code for it.***

import React from 'react';

function User() {

  return (

    <div>

      <h2 style={{ color: 'green' }}>Welcome, User!</h2>

      <p>You can now book your tickets.</p>

      <p>Flight: Indigo 6E-245</p>

      <p>Time: 10:30 AM</p>

      <p>From: Bangalore To: Mumbai</p>

    </div>

  );

}

export default User;

***Update the code of App.js in src/App.js***

import React, { useState } from 'react';

import Guest from './Guest';

import User from './User';

function App() {

  const [isLoggedIn, setIsLoggedIn] = useState(false);

  const handleLoginLogout = () => {

    setIsLoggedIn(!isLoggedIn);

  };

  return (

    <div style={{ textAlign: 'center', marginTop: '50px' }}>

      <h1 style={{ color: 'blue' }}>Ticket Booking App</h1>

      <button onClick={handleLoginLogout}>

        {isLoggedIn ? 'Logout' : 'Login'}

      </button>

      <br /><br />

      {isLoggedIn ? <User /> : <Guest />}

    </div>

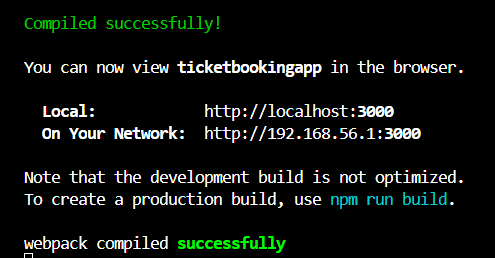
  );

}

export default App;

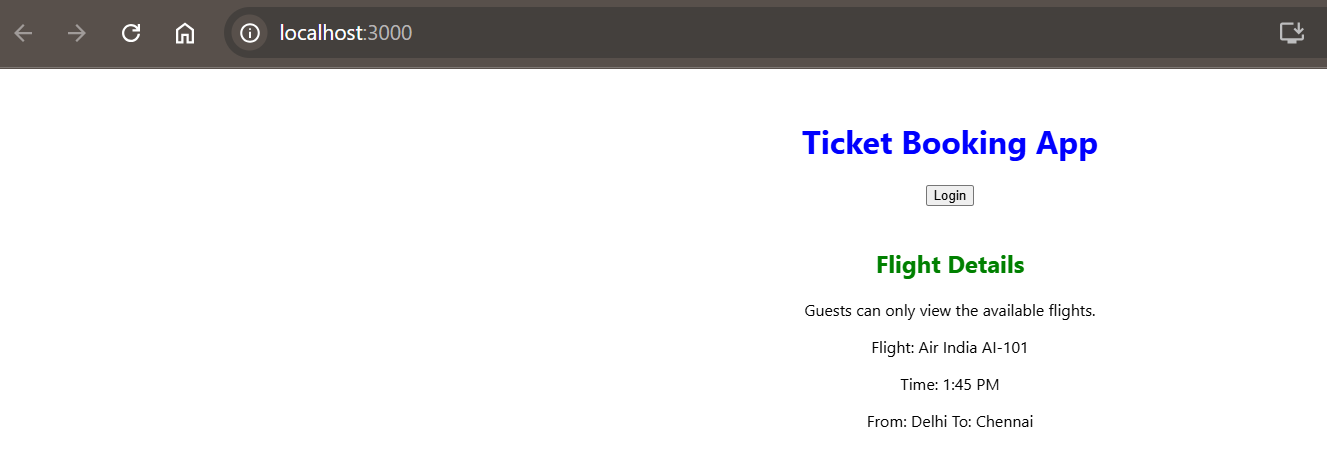
***Starting the React Application on local Server by* npm start**

***Output:-***

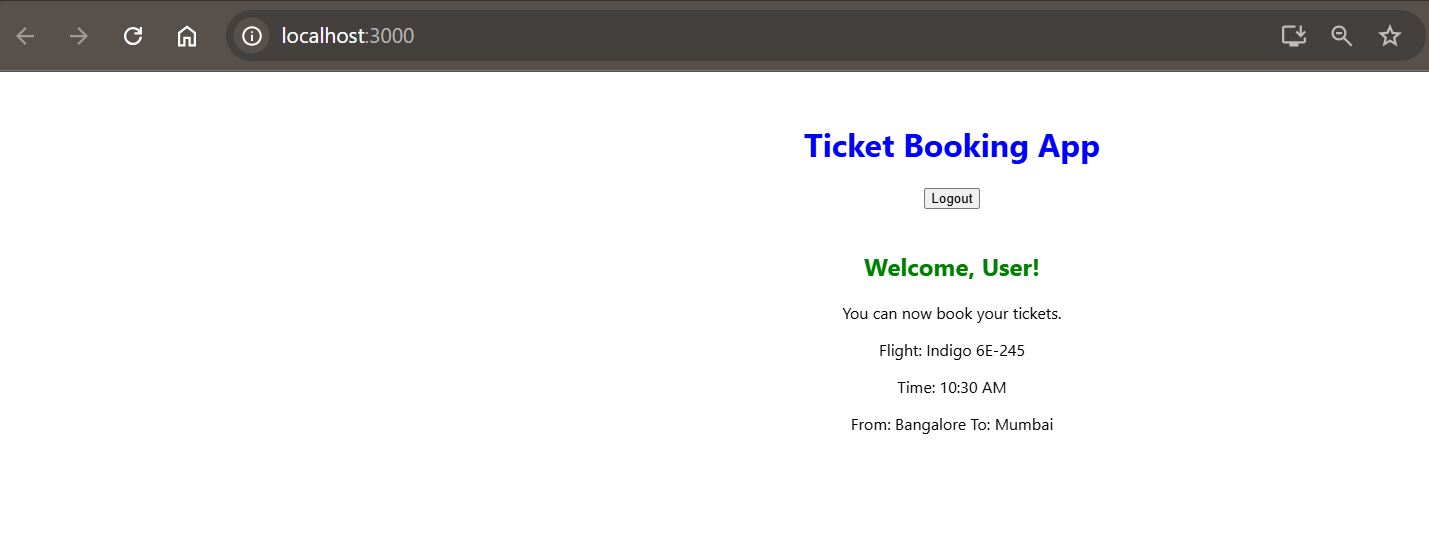


***Output on Local Server localhost:3000***

***For GUEST USER***



***For Login USER***



**Skill:-React**

**Filename:- 13. ReactJS-HOL**

**Solution:-**

**1. Explain various ways of conditional rendering**

**Answer:-**

- Using if/else  
- Using ternary operator  
- Using logical && operator  
- Storing components in variables

**2. Explain how to render multiple components**

**Answer:-**

You can render multiple components using fragments (`<> </>`) or return an array of components.

**3. Define list component**

**Answer:-**

A list component renders a list of data by mapping over it and displaying components for each item.

**4. Explain about keys in React applications**

**Answer:-**

Keys help React identify which items have changed, added, or removed, improving list rendering performance.

**5. Explain how to extract components with keys**

**Answer:-**

Use keys when extracting components from a list to ensure stable identity across re-renders.

**6. Explain React Map, map() function**

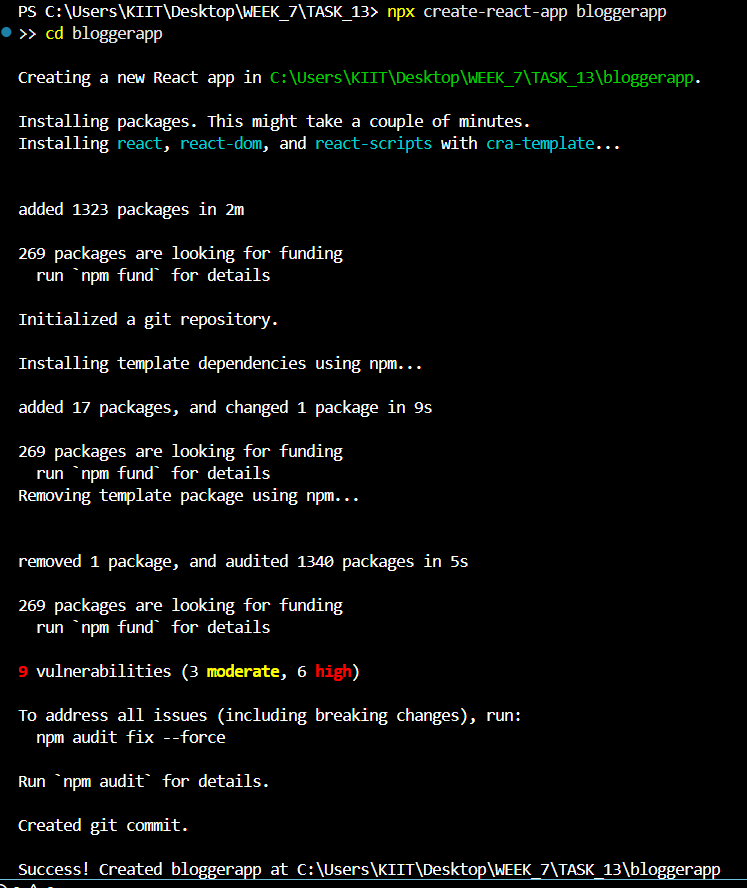
**Answer:-**

The `map()` function transforms an array into a new array of React elements, useful for rendering lists.

***Created a New React App***

***Command:-* npx create-react-app bloggerapp**

**cd bloggerapp**



***Go to src/ Folder Create 3 New Components thar are:***

1. ***Created a New BookDetails.js and below is the code for it.***

function BookDetails({ show }) {

  return (

    <>

      {show && (

        <div>

          <h2>Book Details</h2>

          <h3>Master React</h3>

          <p>670</p>

          <h3>Deep Dive into Angular 11</h3>

          <p>800</p>

          <h3>Mongo Essentials</h3>

          <p>450</p>

        </div>

      )}

    </>

  );

}

export default BookDetails;

1. ***Created a New BlogDetails.js and below is the code for it.***

function BlogDetails({ show }) {

  return show ? (

    <div>

      <h2>Blog Details</h2>

      <h3>React Learning</h3>

      <p><strong>Stephen Biz</strong></p>

      <p>Welcome to learning React!</p>

      <h3>Installation</h3>

      <p><strong>Schwezdzenier</strong></p>

      <p>You can install React from npm.</p>

    </div>

  ) : null;

}

export default BlogDetails;

1. ***Created a New BlogDetails.js and below is the code for it.***

function CourseDetails({ show }) {

  let content;

  if (show) {

    content = (

      <div>

        <h2>Course Details</h2>

        <h3>Angular</h3>

        <p>4/5/2021</p>

        <h3>React</h3>

        <p>6/3/2021</p>

      </div>

    );

  } else {

    content = null;

  }

  return content;

}

export default CourseDetails;

***Update the code of App.js in src/App.js***

import React, { useState } from 'react';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

import CourseDetails from './CourseDetails';

import './App.css';

function App() {

  const [showBook, setShowBook] = useState(true);

  const [showBlog, setShowBlog] = useState(true);

  const [showCourse, setShowCourse] = useState(true);

  return (

    <div>

      <h1>Blogger App</h1>

      <div style={{ textAlign: 'center' }}>

        <button onClick={() => setShowBook(!showBook)}>Toggle Book</button>

        <button onClick={() => setShowBlog(!showBlog)}>Toggle Blog</button>

        <button onClick={() => setShowCourse(!showCourse)}>Toggle Course</button>

      </div>

      <div className="container">

        <div className="section">

          <BookDetails show={showBook} />

        </div>

        <div className="section">

          <BlogDetails show={showBlog} />

        </div>

        <div className="section">

          <CourseDetails show={showCourse} />

        </div>

      </div>

    </div>

  );

}

export default App;

***Update the Code of App.css in src/App.css***

body {

  font-family: Arial, sans-serif;

  background-color: #ffffff;

  margin: 0;

  padding: 0;

}

h1 {

  text-align: center;

  color: #333;

}

.container {

  display: flex;

  justify-content: space-around;

  align-items: flex-start;

  padding: 20px;

  gap: 20px;

}

.section {

  flex: 1;

  border-right: 3px solid green;

  padding-right: 20px;

}

.section:last-child {

  border-right: none;

}

button {

  margin: 10px;

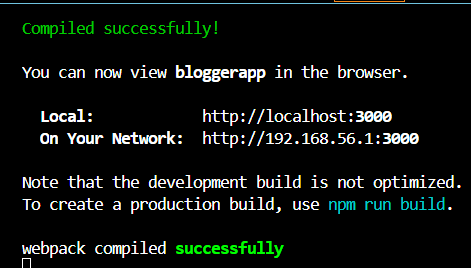
  padding: 8px 12px;

  cursor: pointer;

}

***Starting the React Application on local Server by* npm start**

***Output:-***



***Output on Local Server localhost:3000***

