School of Information Technology and Engineering

BITE304P– Web Technologies Laboratory

Assessment-5

AJAX, AngularJS and ReactJS

NAME : DHAYANIDHI S

REGNO : 23BIT0214

1. Develop a AJAX web application to display a list of possible flight prices and carriers using JSON data.
    a. The page contains two text boxes where the user can specify the start location and end location, a checkbox that they can check if they want to only see non-stop flights (flights with 0 stops) and a “Go!” button.
    b. When the “Go!” button is clicked, clear previous results, and read the JSON data.
        i. Add an h1 to the results div containing the text “Flights from” and then the start and destination locations.
        ii. Turn each flight's data into a paragraph in the results div.
        iii. In each paragraph, write the price of the ticket (with a “$”) followed by the word “from”, the carrier’s name, and then the word “with”, the number of stops, and the word “stops”. If the flight has a price below 1000 display its row in bold.
    c. The JSON data returned has a list of flights associated with it. The list of flights contains lists that each contain a price, a carrier, and the number of stops.


```
{ "Edinburgh":{
"start":"Seattle",
"flights":[{"carrier":"Delta","price":812,"stops":2},
{"carrier":"Air France","price":1020,"stops":0},
{"carrier":"Air France","price":1190,"stops":3}]
},
"New York":{
"start":"Seattle",
"flights":[{"carrier":"British Airlines","price":782,"stops":1},
{"carrier":"Delta","price":1562,"stops":2},
{"carrier":"United","price":957,"stops":1},
{"carrier":"KLM","price":687,"stops":3},
{"carrier":"KLM","price":1458,"stops":1}]
}
```

}

Start location: seattle End location: Edinburgh Non-stop? ☐ Go!

Flights from seattle to Edinburgh

$812 from Delta with 2 stops

$1020 from Air France with 0 stops

$1190 from Air France with 3 stops

Start location: seattle End location: Edinburgh Non-stop? ☑ Go!

Flights from seattle to Edinburgh

$1020 from Air France with 0 stops

CODE :

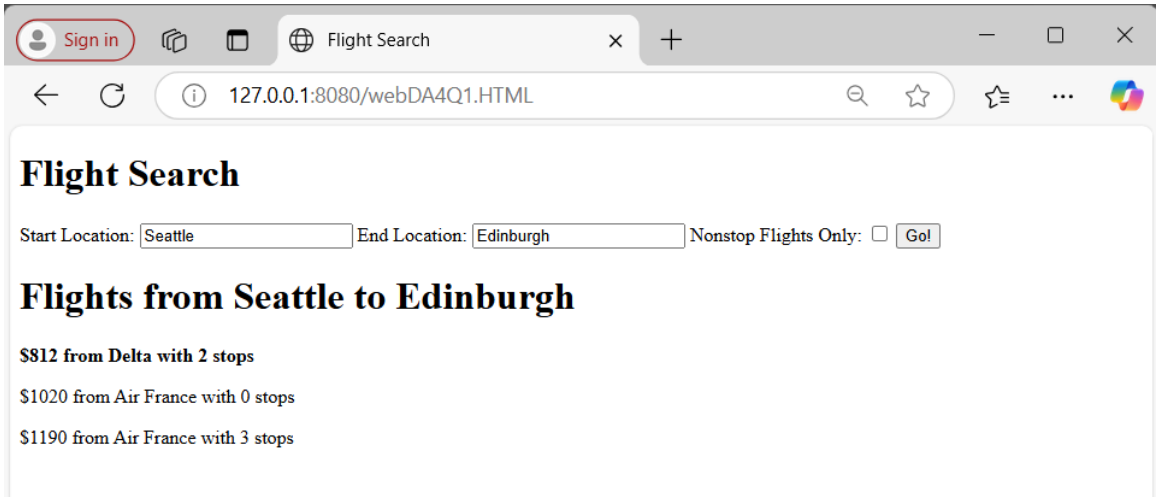
WebDA4Q1.html

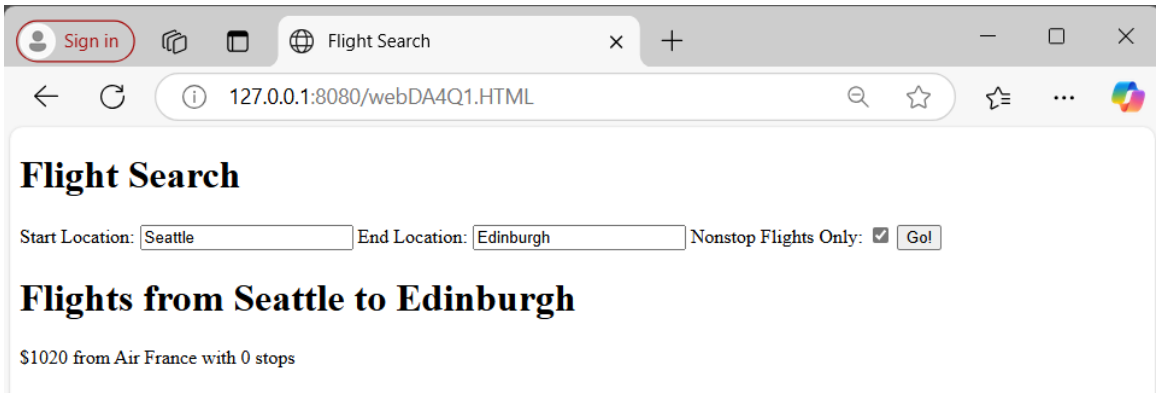
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flight Search</title>
    <style>
        #results {
            margin-top: 20px;
        }
    </style>
</head>
<body>
    <h1>Flight Search</h1>
    <div>
        <label for="start">Start Location:</label>
        <input type="text" id="start" placeholder="e.g. Seattle">
        <label for="end">End Location:</label>
        <input type="text" id="end" placeholder="e.g. Edinburgh">
        <label for="nonstop">Nonstop Flights Only:</label>
        <input type="checkbox" id="nonstop">
        <button id="goButton">Go!</button>
    </div>
    <div id="results"></div>

    <script src="webDA4Q1.js"></script>
</body>
</html>
```

WebDA4Q1.js

```
document.getElementById('goButton').addEventListener('click', function() {
    const resultsDiv = document.getElementById('results');
    resultsDiv.innerHTML = '';
    const startLocation = document.getElementById('start').value.trim();
    const endLocation = document.getElementById('end').value.trim();
    const nonstopOnly = document.getElementById('nonstop').checked;
    const flightData = {
        "Edinburgh": {
            "start": "Seattle",
            "flights": [
                {"carrier": "Delta", "price": 812, "stops": 2},
                {"carrier": "Air France", "price": 1020, "stops": 0},
                {"carrier": "Air France", "price": 1190, "stops": 3}
            ]
        },
        "New York": {
            "start": "Seattle",
            "flights": [
                {"carrier": "British Airlines", "price": 782, "stops": 1},
                {"carrier": "Delta", "price": 1562, "stops": 2},
                {"carrier": "United", "price": 957, "stops": 1},
                {"carrier": "KLM", "price": 687, "stops": 3},
                {"carrier": "KLM", "price": 1458, "stops": 1}
            ]
        }
    };
    if (flightData[endLocation]) {
        const flights = flightData[endLocation].flights;

        const header = document.createElement('h1');
        header.textContent = `Flights from ${startLocation} to
${endLocation}`;
        resultsDiv.appendChild(header);
        flights.forEach(flight => {
            if (!nonstopOnly || flight.stops === 0) {
                const flightParagraph = document.createElement('p');
                flightParagraph.textContent = `$${flight.price} from
${flight.carrier} with ${flight.stops} stops`;
                if (flight.price < 1000) {
                    flightParagraph.style.fontWeight = 'bold';
                }
                resultsDiv.appendChild(flightParagraph);
            }
        });
    } else {
        resultsDiv.innerHTML = `<p>No flights available from
${startLocation} to ${endLocation}.</p>`;}
});
```

OUTPUT :





2. Develop a web application using AngularJS and add behavior to the following HTML code. The page contains a text input box with an id of foodname, and a drop-down list with an id of foodgroup.
   i. The user types a name of a food item into the foodname text box, such as apple or Cookie, selects a food group from the drop-down list, such as dairy or fruit, and then clicks the button with an id of eat.
   ii. When the eat button is clicked, any element on the page that matches all of the following criteria will be removed from the page:
      a. The element is an img element that has a class of food.
      b. The element's food group matches the group chosen in the foodgroup drop-down list. Food groups are represented as class attributes. This is in addition to the food class; recall that class attributes can specify multiple classes separated by spaces. For example, a jug of milk would have the following element:

c. The food item's name is the same as the text in the foodname box. The food's name is stored as the image's alt attribute. For example, if the user types cookie, img elements with an alt of cookie will be removed. Your code should be case-insensitive; for example, coOKIe should match images with an alt of cookie.

Assume that the Prototype is also included on the page.

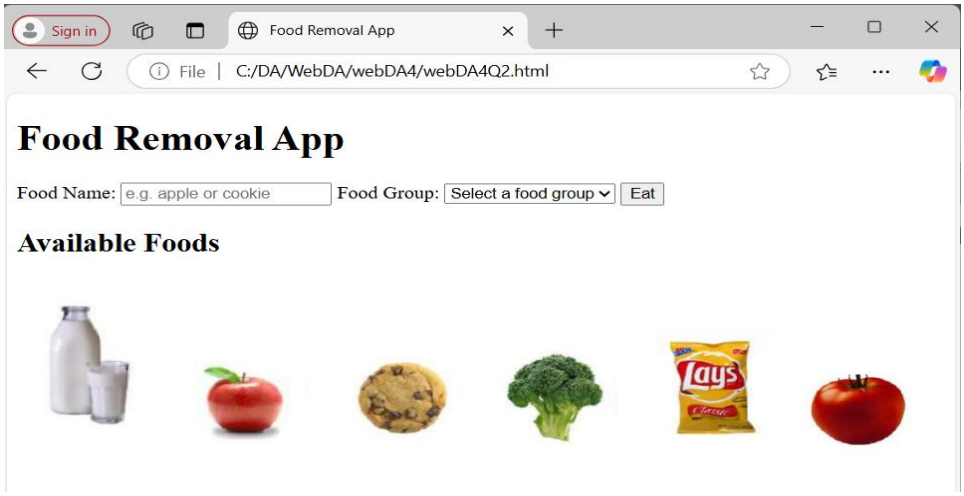
CODE :

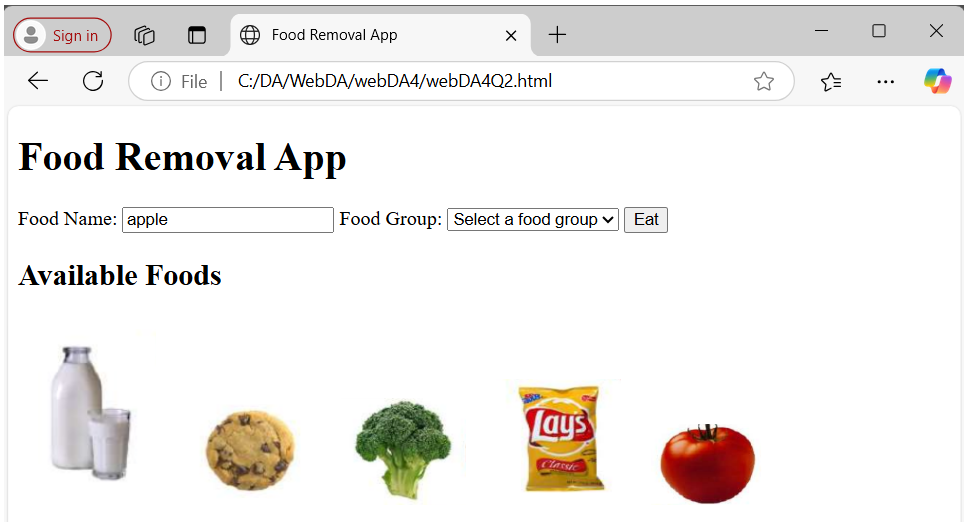
WebDA4Q2.html

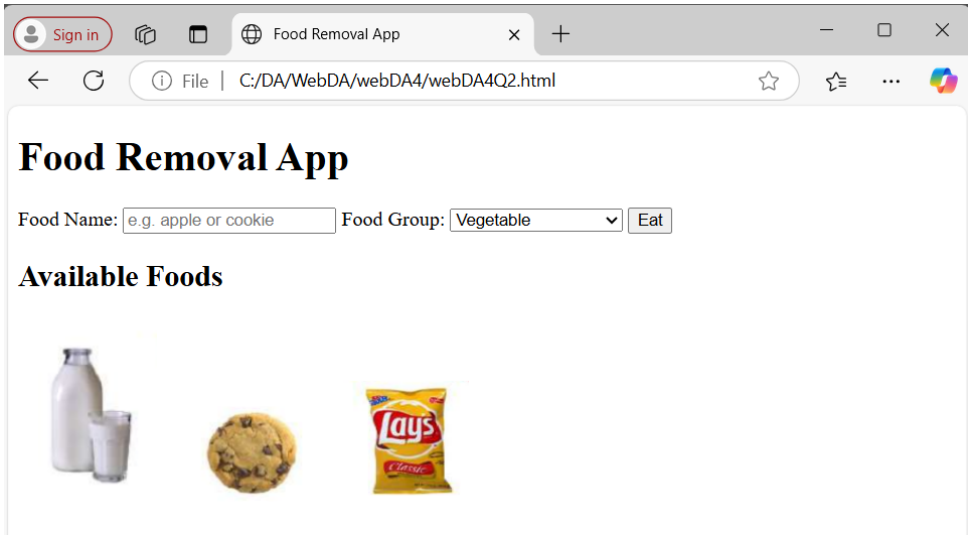
```
<!DOCTYPE html>
<html lang="en" ng-app="foodApp">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Food Removal App</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"
></script>
    <style>
        .food {
            width: 100px;
            height: auto;
            margin: 10px;
        }
    </style>
</head>
<body ng-controller="FoodController">
```

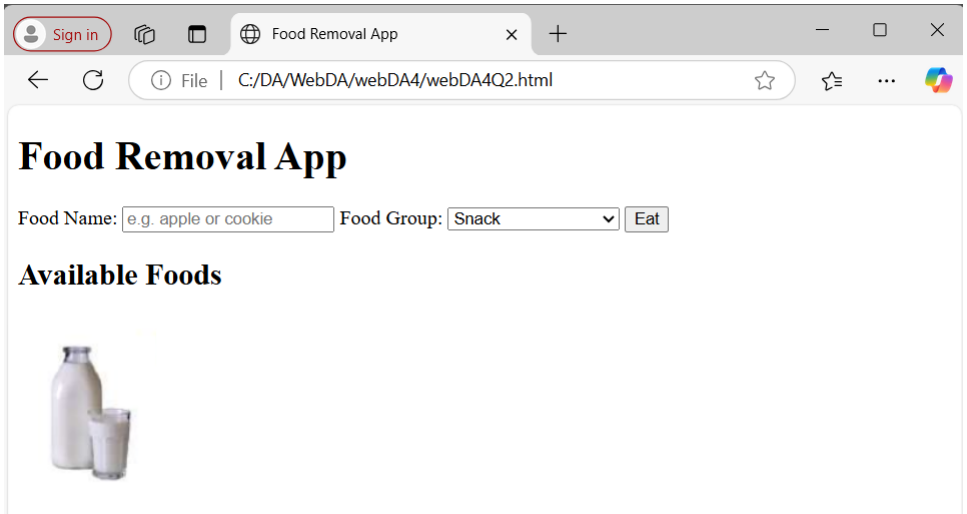
```
    <h1>Food Removal App</h1>
    <div>
        <label for="foodname">Food Name:</label>
        <input type="text" id="foodname" ng-model="foodName"
placeholder="e.g. apple or cookie">

        <label for="foodgroup">Food Group:</label>
        <select id="foodgroup" ng-model="selectedGroup">
            <option value="">Select a food group</option>
            <option value="dairy">Dairy</option>
            <option value="fruit">Fruit</option>
            <option value="vegetable">Vegetable</option>
            <option value="snack">Snack</option>
        </select>
        <button id="eat" ng-click="removeFood()">Eat</button>
    </div>
    <div>
        <h2>Available Foods</h2>
        <img src="images/milk.png" class="food dairy" alt="milk" />
        <img src="images/apple.png" class="food fruit" alt="apple" />
        <img src="images/cookie.png" class="food snack" alt="cookie" />
        <img src="images/broccoli.png" class="food vegetable"
alt="broccoli" />
        <img src="images/lays.png" class="food snack" alt="lays" />
        <img src="images/tomato.png" class="food vegetable" alt="tomato"/>
    </div>
    <script>
        angular.module('foodApp', [])
            .controller('FoodController', ['$scope', function($scope) {
                $scope.foodName = '';
                $scope.selectedGroup = '';
                $scope.removeFood = function() {
                    var foodImages =
document.querySelectorAll('img.food');
                    foodImages.forEach(function(img) {
                        var foodAlt =
img.getAttribute('alt').toLowerCase();
                        var foodGroup = Array.from(img.classList).find(cls
=> cls !== 'food');
                        if (foodAlt === $scope.foodName.toLowerCase() ||
foodGroup === $scope.selectedGroup) {
                            img.style.display = 'none';
                        }
                    });
                };
            }]);
    </script>
</body>
</html>
```

OUTPUT :


Sign in
Food Removal App
File | C:/DA/WebDA/webDA4/webDA4Q2.html
Food Removal App
Food Name: e.g. apple or cookie
Food Group: Select a food group
Eat
Available Foods


Sign in
Food Removal App
File | C:/DA/WebDA/webDA4/webDA4Q2.html
Food Removal App
Food Name: apple
Food Group: Select a food group
Eat
Available Foods


Sign in
Food Removal App
File | C:/DA/WebDA/webDA4/webDA4Q2.html
Food Removal App
Food Name: e.g. apple or cookie
Food Group: Vegetable
Eat
Available Foods



3. Develop a Simple Calculator Application using React JS that would contain the components for general mathematical calculations

CODE :

Calculator.js

```
import React, { useState } from 'react';
const Calculator = () => {
    const [input, setInput] = useState('');
    const [result, setResult] = useState('');
    const handleButtonClick = (value) => {
        setInput(input + value);
    };
    const handleClear = () => {
        setInput('');
        setResult('');
    };
    const handleCalculate = () => {
        try {
            const evalResult = eval(input);
            setResult(evalResult);
        } catch (error) {
            setResult('Error');
        }
    };
    return (
        <div style={{ padding: '20px', maxWidth: '300px', margin: 'auto'
}}>
            <h2>Simple Calculator</h2>
            <div>
```

```
                <input type="text" value={input} readOnly />
                <div>{result}</div>
            </div>
            <div>
                <button onClick={() => handleButtonClick('1')}>1</button>
                <button onClick={() => handleButtonClick('2')}>2</button>
                <button onClick={() => handleButtonClick('3')}>3</button>
                <button onClick={() => handleButtonClick('+')}>+</button>
            </div>
            <div>
                <button onClick={() => handleButtonClick('4')}>4</button>
                <button onClick={() => handleButtonClick('5')}>5</button>
                <button onClick={() => handleButtonClick('6')}>6</button>
                <button onClick={() => handleButtonClick('-')}>-</button>
            </div>
            <div>
                <button onClick={() => handleButtonClick('7')}>7</button>
                <button onClick={() => handleButtonClick('8')}>8</button>
                <button onClick={() => handleButtonClick('9')}>9</button>
                <button onClick={() => handleButtonClick('*')}>*</button>
            </div>
            <div>
                <button onClick={handleClear}>C</button>
                <button onClick={() => handleButtonClick('0')}>0</button>
                <button onClick={handleCalculate}>=</button>
                <button onClick={() => handleButtonClick('/')}>/</button>
            </div>
        </div>
    );
};
export default Calculator;
```

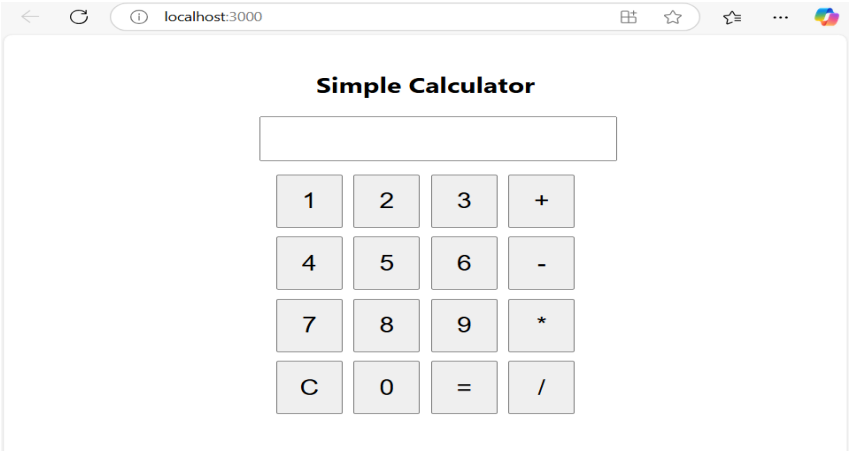
App.js

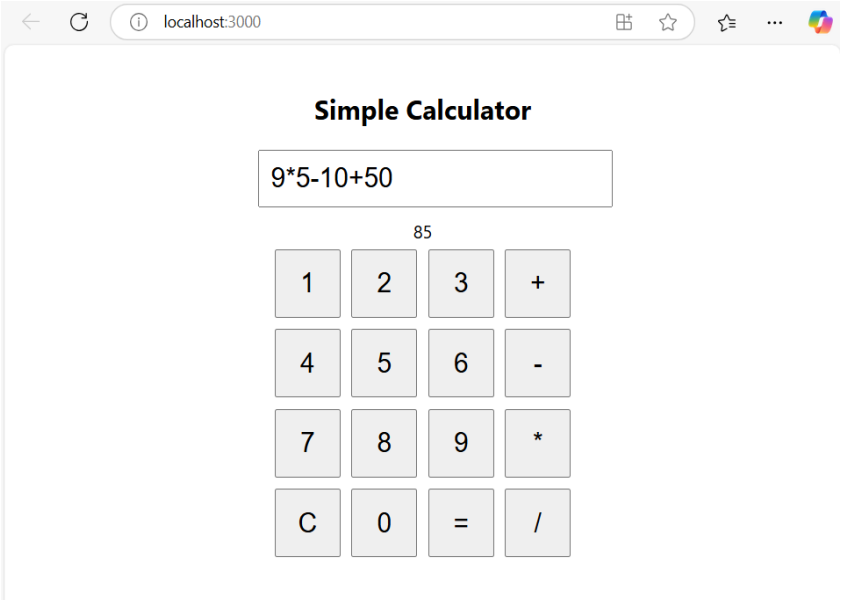
```
import React from 'react';
import './App.css';
import Calculator from './Calculator';
function App() {
    return (
        <div className="App">
            <Calculator />
        </div>
    );
}

export default App;
```

App.css

```
.App {
  text-align: center;
}
input {
  width: 100%;
  padding: 10px;
  font-size: 24px;
  margin-bottom: 10px;
}
button {
  width: 60px;
  height: 60px;
  font-size: 24px;
  margin: 5px;
}
```

OUTPUT :





4. Create a Class Component called 'Employee' that would render the details only when the age of the employee is above 25 else display 'Invalid Data'. Apply both inline styling and external styles.

CODE :

Employee.js

```
import React, { Component } from 'react';
import './Employee.css';

class Employee extends Component {
    render() {
        const { name, age, position } = this.props;
        const employeeStyle = {
            border: '1px solid #ccc',
            borderRadius: '5px',
            padding: '20px',
            margin: '10px',
            backgroundColor: '#f9f9f9',
            boxShadow: '0 2px 5px rgba(0, 0, 0, 0.1)',
        };
        if (age > 25) {
            return (
                <div style={employeeStyle}>
                    <h2>Employee Details</h2>
                    <p><strong>Name:</strong> {name}</p>
                    <p><strong>Age:</strong> {age}</p>
                    <p><strong>Position:</strong> {position}</p>
                </div>
            );
        } else {
            return (
                <div style={employeeStyle}>
                    <h2>Invalid Data</h2>
                </div>
            );
        }
    }
}
export default Employee;
```

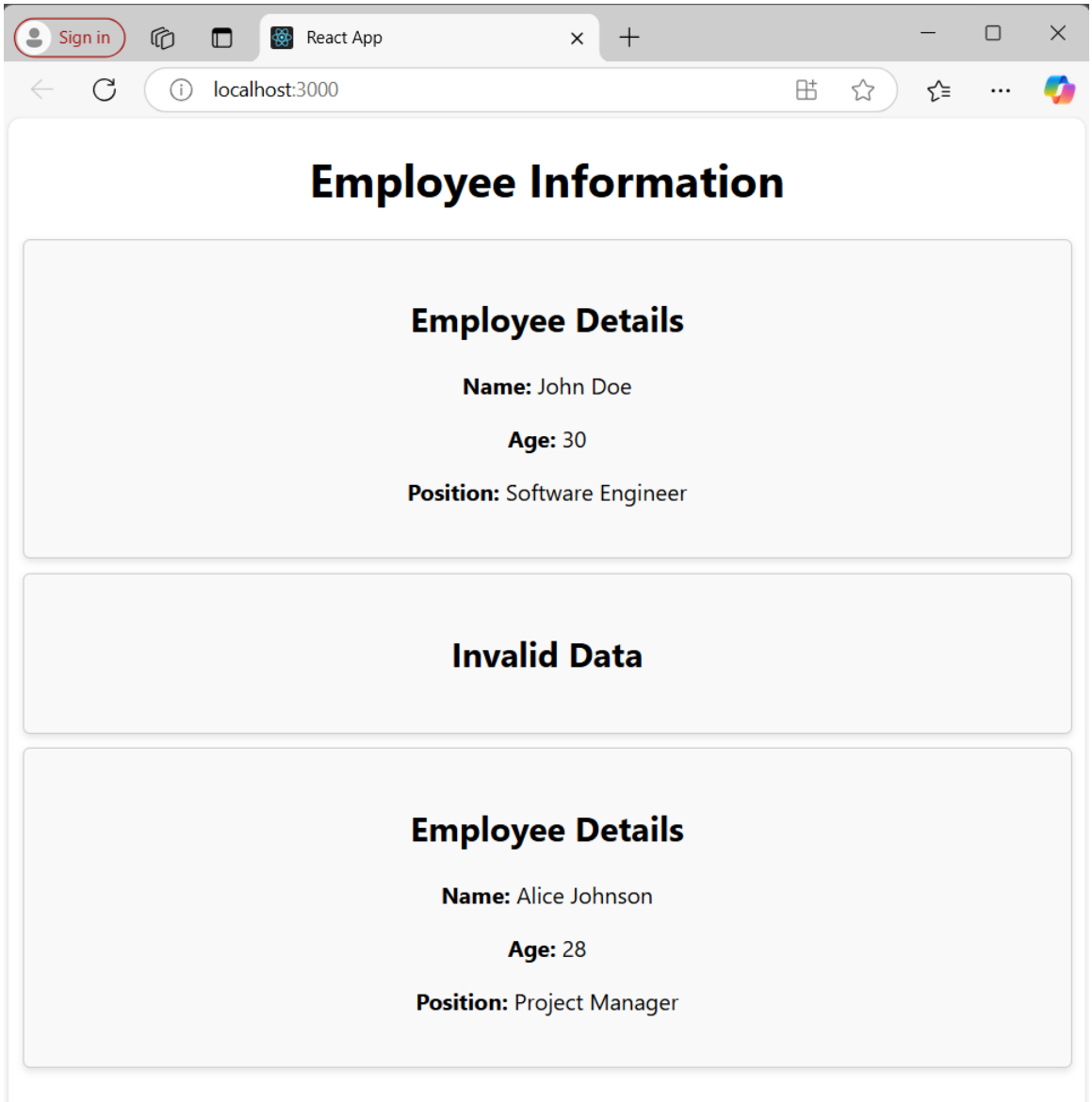
Employee.css

```
.employee-container {
    border: 1px solid #ccc;
    border-radius: 5px;
    padding: 20px;
    margin: 10px;
    background-color: #f9f9f9;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

.employee-container h2 {
    color: #333;
}

.employee-container p {
    color: #555;
}
```

App.js

```
import React from 'react';
import ./App.css';
import Employee from ./Employee';
function App() {
    return (
        <div className="App">
            <h1>Employee Information</h1>
            {/* Example Employee Data */}
            <Employee name="John Doe" age={30} position="Software
Engineer" />
            <Employee name="Jane Smith" age={22} position="Intern" />
            <Employee name="Alice Johnson" age={28} position="Project
Manager" />
        </div>
    );
}
export default App;
```

OUTPUT :



5. Create class component for the following:
    a. Represent the book details like ISBN, title, author, publisher, category(Maths/Science/Novel/GK) and price of the book. Set the default values for these properties. Create few objects with the values and one object with default value. Print all books in HTML table format.

CODE :

Book.js

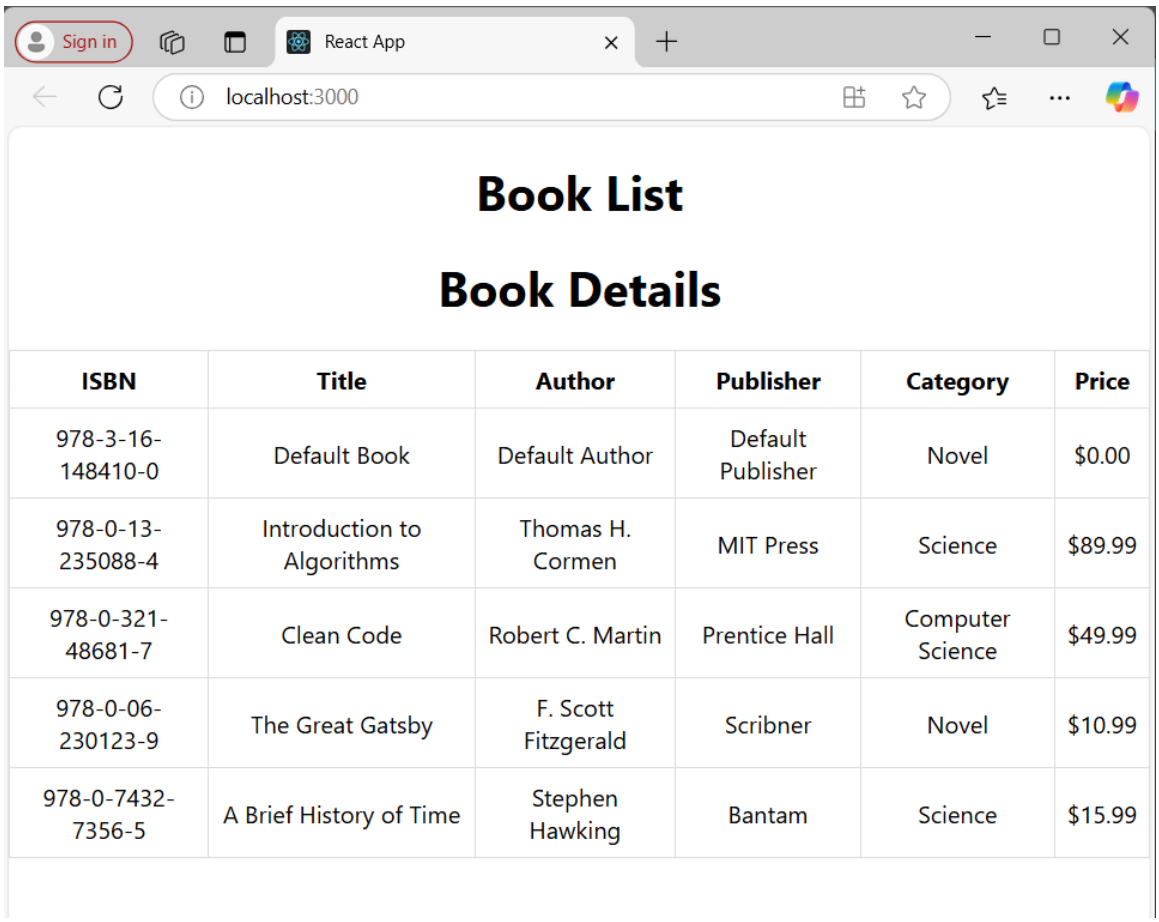
```
import React, { Component } from 'react';
class Book extends Component {
    constructor(props) {
        super(props);
        this.state = {
            books: [
                {
                    ISBN: '978-3-16-148410-0',
                    title: 'Default Book',
                    author: 'Default Author',
                    publisher: 'Default Publisher',
                    category: 'Novel',
                    price: 0,
                },
                {
                    ISBN: '978-0-13-235088-4',
                    title: 'Introduction to Algorithms',
                    author: 'Thomas H. Cormen',
                    publisher: 'MIT Press',
                    category: 'Science',
                    price: 89.99,
                },
                {
                    ISBN: '978-0-321-48681-7',
                    title: 'Clean Code',
                    author: 'Robert C. Martin',
                    publisher: 'Prentice Hall',
                    category: 'Computer Science',
                    price: 49.99,
                },
                {
                    ISBN: '978-0-06-230123-9',
                    title: 'The Great Gatsby',
                    author: 'F. Scott Fitzgerald',
                    publisher: 'Scribner',
                    category: 'Novel',
                    price: 10.99,
                },
                {
                    ISBN: '978-0-7432-7356-5',
                    title: 'A Brief History of Time',
                    author: 'Stephen Hawking',
                    publisher: 'Bantam',
                    category: 'Science',
                    price: 15.99,
                },
```

```
            ],
        };
    }

    render() {
        return (
            <div>
                <h1>Book Details</h1>
                <table style={{ width: '100%', borderCollapse: 'collapse'
}}>
                    <thead>
                        <tr>
                            <th style={{ border: '1px solid #ddd',
padding: '8px' }}>ISBN</th>
                            <th style={{ border: '1px solid #ddd',
padding: '8px' }}>Title</th>
                            <th style={{ border: '1px solid #ddd',
padding: '8px' }}>Author</th>
                            <th style={{ border: '1px solid #ddd',
padding: '8px' }}>Publisher</th>
                            <th style={{ border: '1px solid #ddd',
padding: '8px' }}>Category</th>
                            <th style={{ border: '1px solid #ddd',
padding: '8px' }}>Price</th>
                        </tr>
                    </thead>
                    <tbody>
                        {this.state.books.map((book, index) => (
                            <tr key={index}>
                                <td style={{ border: '1px solid #ddd',
padding: '8px' }}>{book.ISBN}</td>
                                <td style={{ border: '1px solid #ddd',
padding: '8px' }}>{book.title}</td>
                                <td style={{ border: '1px solid #ddd',
padding: '8px' }}>{book.author}</td>
                                <td style={{ border: '1px solid #ddd',
padding: '8px' }}>{book.publisher}</td>
                                <td style={{ border: '1px solid #ddd',
padding: '8px' }}>{book.category}</td>
                                <td style={{ border: '1px solid #ddd',
padding: '8px' }}>${book.price.toFixed(2)}</td>
                            </tr>
                        ))}
                    </tbody>
                </table>
            </div>
        );
    }
}
export default Book;
```

App.js

```
import React from 'react';
import './App.css';
import Book from './Book';
function App() {
    return (
        <div className="App">
            <h1>Book List</h1> {/* Title for the application */}
            <Book /> {/* Render the Book component */}
        </div>
    );
}
export default App;
```

OUTPUT :



ISBN	Title	Author	Publisher	Category	Price
978-3-16-148410-0	Default Book	Default Author	Default Publisher	Novel	$0.00
978-0-13-235088-4	Introduction to Algorithms	Thomas H. Cormen	MIT Press	Science	$89.99
978-0-321-48681-7	Clean Code	Robert C. Martin	Prentice Hall	Computer Science	$49.99
978-0-06-230123-9	The Great Gatsby	F. Scott Fitzgerald	Scribner	Novel	$10.99
978-0-7432-7356-5	A Brief History of Time	Stephen Hawking	Bantam	Science	$15.99

6. Design a React JS Form to capture Employee ID, Employee Name, Gender, Date of Birth, Date of Joining, Basic Pay, Allowances and Deductions. Trigger submit button click event to display the below requirement:
    a. Compute Net Pay as Basic Pay + Allowance-Deductions
    b. Compute total years of experience as on date.
    c. Display the output as "Mr./Ms. (based on gender) <Employee Name> (<Emp Id> having <total years of experience> years is earning <Net Pay> as salary"
    d. Implement the above form functionalities in both controlled and uncontrolled form components version.
    e. On click oCreate controlled and uncontrolled form to collect the input for book class.Update the App.js, index.js and index.html files according to the questions given here. Represent the class component as a separate file.Include codes of all these files in your PDF document under the title of each file names. Also, include the outputs.

CODE :

ControlledForm.js

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

class ControlledForm extends Component {
    constructor(props) {
        super(props);
        this.state = {
            empId: '',
            empName: '',
            gender: 'Male',
            dob: '',
            doj: '',
            basicPay: '',
            allowances: '',
            deductions: '',
            output: '',
        };
    }

    handleChange = (e) => {
        this.setState({ [e.target.name]: e.target.value });
    };

    handleSubmit = (e) => {
        e.preventDefault();
```

```
        const { empId, empName, gender, dob, doj, basicPay, allowances,
deductions } = this.state;

        const netPay = parseFloat(basicPay) + parseFloat(allowances) -
parseFloat(deductions);

        const joiningDate = new Date(doj);
        const currentDate = new Date();
        const totalYears = currentDate.getFullYear() -
joiningDate.getFullYear();

        const genderPrefix = gender === 'Male' ? 'Mr.' : 'Ms.';
        const output = `${genderPrefix} ${empName} (${empId}) having
${totalYears} years is earning $${netPay.toFixed(2)} as salary.`;

        this.setState({ output });
    };
    render() {
        return (
            <div>
                <h2>Controlled Form</h2>
                <form onSubmit={this.handleSubmit}>
                    <input type="text" name="empId" placeholder="Employee
ID" onChange={this.handleChange} required />
                    <input type="text" name="empName"
placeholder="Employee Name" onChange={this.handleChange} required />
                    <select name="gender" onChange={this.handleChange}>
                        <option value="Male">Male</option>
                        <option value="Female">Female</option>
                    </select>
                    <input type="date" name="dob"
onChange={this.handleChange} required />
                    <input type="date" name="doj"
onChange={this.handleChange} required />
                    <input type="number" name="basicPay"
placeholder="Basic Pay" onChange={this.handleChange} required />
                    <input type="number" name="allowances"
placeholder="Allowances" onChange={this.handleChange} required />
                    <input type="number" name="deductions"
placeholder="Deductions" onChange={this.handleChange} required />
                    <button type="submit">Submit</button>
                </form>
                {this.state.output && <p>{this.state.output}</p>}
            </div>
        );
    }
}

export default ControlledForm;
```

UncontrolledForm.js

```
import React, { Component } from 'react';
class UncontrolledForm extends Component {
    constructor(props) {
        super(props);
        this.empIdRef = React.createRef();
        this.empNameRef = React.createRef();
        this.genderRef = React.createRef();
        this.dobRef = React.createRef();
        this.dojRef = React.createRef();
        this.basicPayRef = React.createRef();
        this.allowancesRef = React.createRef();
        this.deductionsRef = React.createRef();
        this.state = {
            output: '',
        };
    }
    handleSubmit = (e) => {
        e.preventDefault();
        const empId = this.empIdRef.current.value;
        const empName = this.empNameRef.current.value;
        const gender = this.genderRef.current.value;
        const dob = this.dobRef.current.value;
        const doj = this.dojRef.current.value;
        const basicPay = parseFloat(this.basicPayRef.current.value);
        const allowances = parseFloat(this.allowancesRef.current.value);
        const deductions = parseFloat(this.deductionsRef.current.value);
        const netPay = basicPay + allowances - deductions;
        const joiningDate = new Date(doj);
        const currentDate = new Date();
        const totalYears = currentDate.getFullYear() -
joiningDate.getFullYear();
        const genderPrefix = gender === 'Male' ? 'Mr.' : 'Ms.';
        const output = `${genderPrefix} ${empName} (${empId}) having
${totalYears} years is earning $${netPay.toFixed(2)} as salary.`;
        this.setState({ output });
    };
    render() {
        return (
            <div>
                <h2>Uncontrolled Form</h2>
                <form onSubmit={this.handleSubmit}>
                    <input type="text" ref={this.empIdRef}
placeholder="Employee ID" required />
                    <input type="text" ref={this.empNameRef}
placeholder="Employee Name" required />
                    <select ref={this.genderRef}>
                        <option value="Male">Male</option>
                        <option value="Female">Female</option>
```

```
                    </select>
                    <input type="date" ref={this.dobRef} required />
                    <input type="date" ref={this.dojRef} required />
                    <input type="number" ref={this.basicPayRef}
placeholder="Basic Pay" required />
                    <input type="number" ref={this.allowancesRef}
placeholder="Allowances" required />
                    <input type="number" ref={this.deductionsRef}
placeholder="Deductions" required />
                    <button type="submit">Submit</button>
                </form>
                {this.state.output && <p>{this.state.output}</p>}
            </div>
        );
    }
}

export default UncontrolledForm;
```

App.js

```
import React from 'react';
import './App.css';
import ControlledForm from './ControlledForm';
import UncontrolledForm from './UncontrolledForm';

function App() {
    return (
        <div className="App">
            <h1>Employee Form</h1>
            <ControlledForm />
            <UncontrolledForm />
        </div>
    );
}

export default App;
```

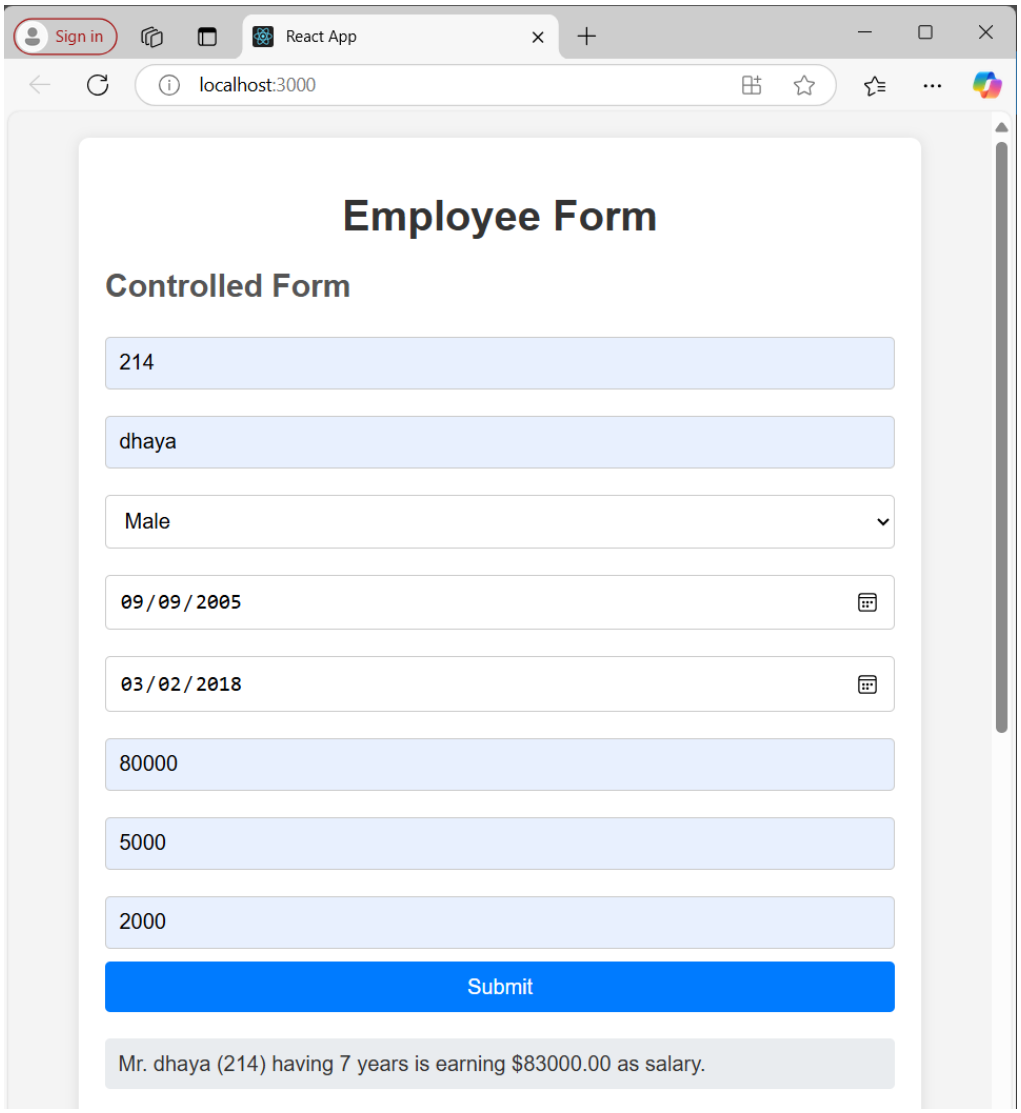
App.css

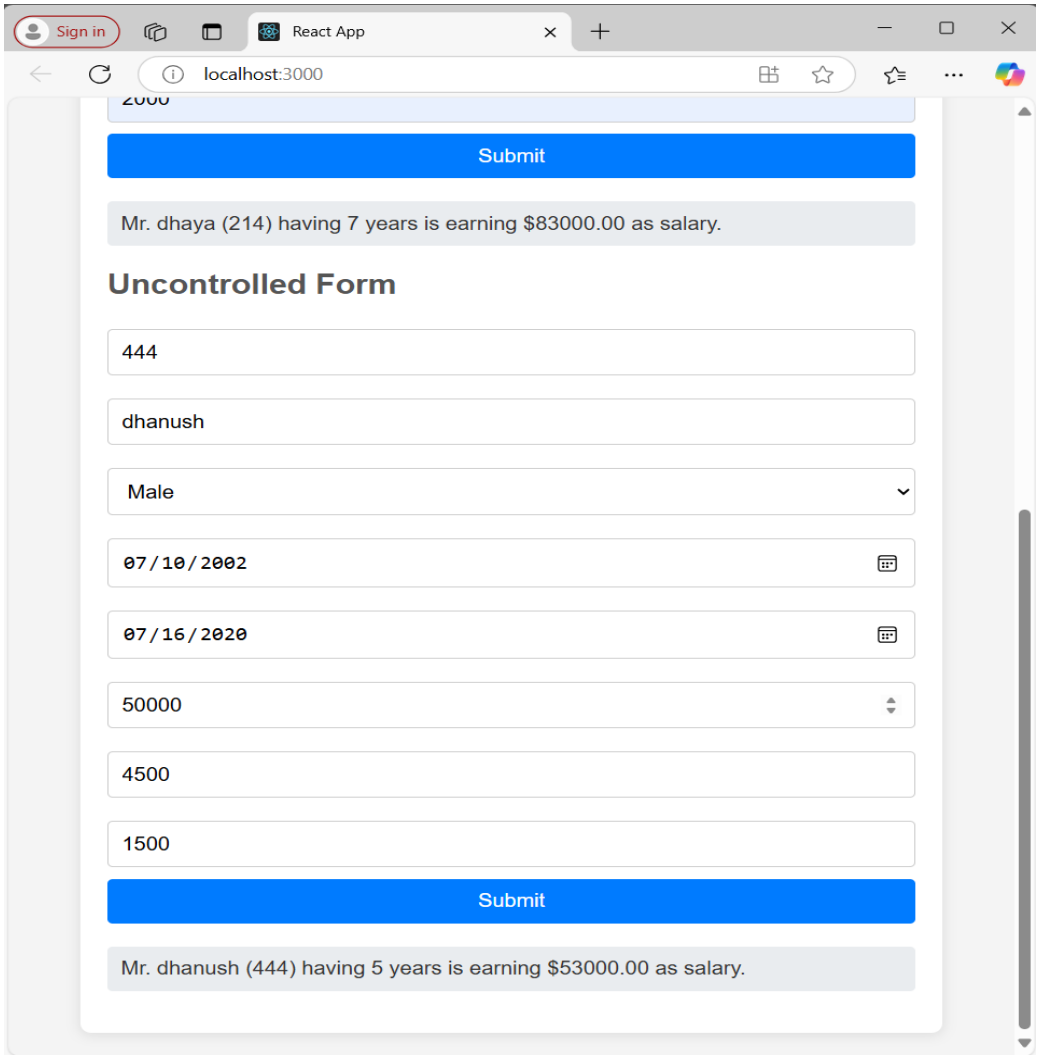
```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  margin: 0;
  padding: 20px;
}
```

```css
.App {
  max-width: 600px;
  margin: auto;
  background: white;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}

h1 {
  text-align: center;
  color: #333;
}

h2 {
  color: #555;
  margin-bottom: 15px;
}

form {
  display: flex;
  flex-direction: column;
}

input[type="text"],
input[type="date"],
input[type="number"],
select {
  padding: 10px;
  margin: 10px 0;
  border: 1px solid #ccc;
  border-radius: 4px;
  font-size: 16px;
}

input[type="text"]:focus,
input[type="date"]:focus,
input[type="number"]:focus,
select:focus {
  border-color: #007bff;
  outline: none;
}

button {
  padding: 10px;
  background-color: #007bff;
  color: white;
  border: none;
  border-radius: 4px;
  font-size: 16px;
```

```
  cursor: pointer;
  transition: background-color 0.3s;
}

button:hover {
  background-color: #0056b3;
}

p {
  margin-top: 20px;
  font-size: 16px;
  color: #333;
  background-color: #e9ecef;
  padding: 10px;
  border-radius: 4px;
}
```

OUTPUT :





7. Develop a Component called 'Login' having the username and password as it members. Validate the rules for the password as it should contain atleast one uppercase, lowercase, numbers, special symbol and the length between 8 and 15. On successful validation display 'Welcome <Username>'

CODE :

Login.js

```
import React, { Component } from 'react';
import './Login.css';
class Login extends Component {
    constructor(props) {
        super(props);
        this.state = {
            username: '',
            password: '',
```

```
            message: '',
            error: '',
        };
    }
    handleChange = (e) => {
        this.setState({ [e.target.name]: e.target.value });
    };
    validatePassword = (password) => {
        const passwordRegex = /^(?=.*[a-z])(?=.*[A-
Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,15}$/;
        return passwordRegex.test(password);
    };
    handleSubmit = (e) => {
        e.preventDefault();
        const { username, password } = this.state;
        if (this.validatePassword(password)) {
            this.setState({ message: `Welcome ${username}`, error: '' });
        } else {
            this.setState({ error: 'Password must contain at least one
uppercase letter, one lowercase letter, one number, one special character,
and be between 8 to 15 characters long.', message: '' });}
    };
    render() {
        return (
            <div>
                <h2>Login</h2>
                <form onSubmit={this.handleSubmit}>
                    <input
                        type="text"
                        name="username"
                        placeholder="Username"
                        value={this.state.username}
                        onChange={this.handleChange}
                        required/>
                    <input
                        type="password"
                        name="password"
                        placeholder="Password"
                        value={this.state.password}
                        onChange={this.handleChange}
                        required/>
                    <button type="submit">Login</button>
                </form>
                {this.state.message && <p>{this.state.message}</p>}
                {this.state.error && <p style={{ color: 'red'
}}>{this.state.error}</p>}
            </div>
        );
    }}
export default Login;
```

Login.css

```
/* Login.css */
body {
    background-color: #f0f0f0;
    font-family: Arial, sans-serif;
}

h2 {
    text-align: center;
    color: #333;
}

form {
    background-color: white;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    max-width: 400px;
    margin: 50px auto;
}

input[type="text"],
input[type="password"] {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    border: 1px solid #ccc;
    border-radius: 4px;
}

button {
    width: 100%;
    padding: 10px;
    background-color: #007bff;
    color: white;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    transition: background-color 0.3s;
}

button:hover {
    background-color: #0056b3;
}

p {
    text-align: center;
    color: red;
}
```

App.js

```
import React from 'react';
import './App.css';
import Login from './Login';

function App() {
    return (
        <div className="App">
            <h1>Login Form</h1>
            <Login />
        </div>
    );
}
export default App;
```

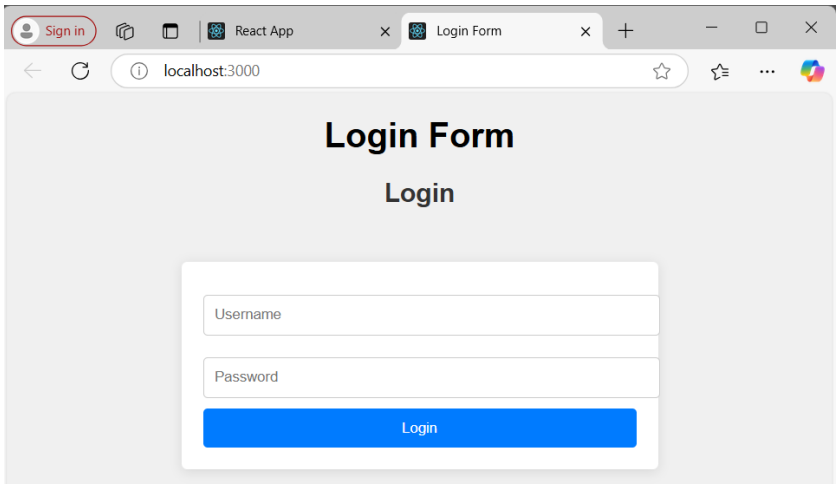
index.js

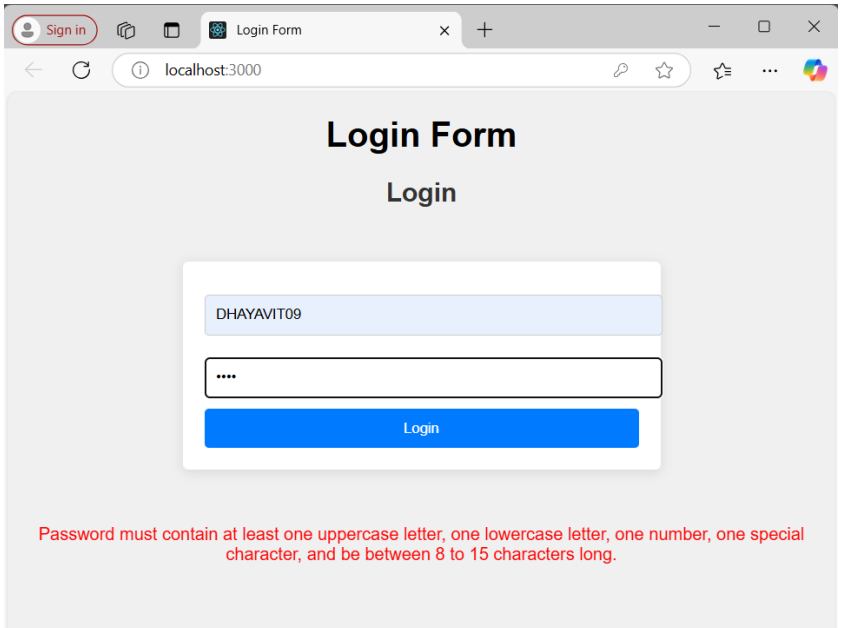
```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
    <React.StrictMode>
        <App />
    </React.StrictMode>
);
```

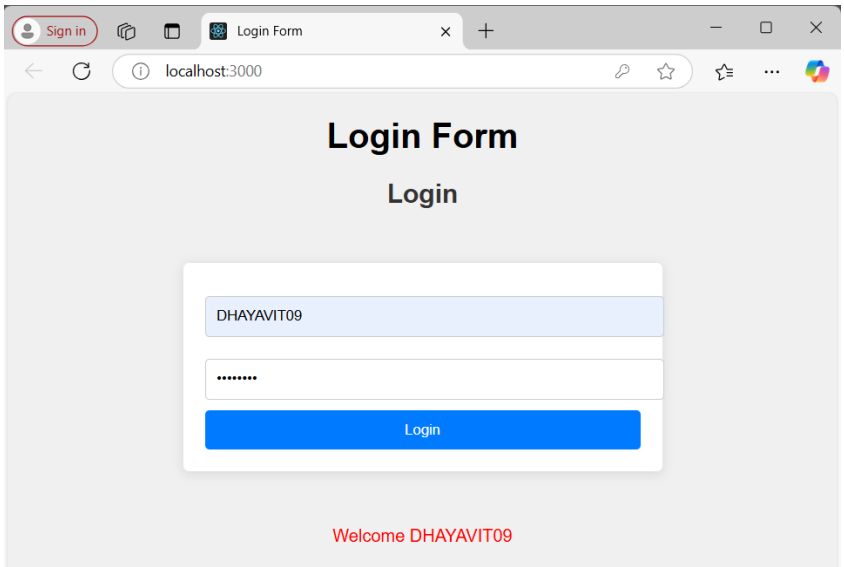
public/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Login Form</title>
</head>
<body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
</body>
</html>
```

OUTPUT :


Sign in
React App
Login Form
localhost:3000
Login Form
Login
Username
Password
Login


Sign in
Login Form
localhost:3000
Login Form
Login
DHAYAVIT09
••••
Login
Password must contain at least one uppercase letter, one lowercase letter, one number, one special character, and be between 8 to 15 characters long.


Sign in
Login Form
localhost:3000
Login Form
Login
DHAYAVIT09
••••••••
Login
Welcome DHAYAVIT09