5th Vote:

Index.js

import React from 'react'

import ReactDOM from 'react-dom/client'

import App from './App.js'

import './index.css'

ReactDOM.createRoot(document.getElementById('root')).render(

  <React.StrictMode>

    <App />

  </React.StrictMode>,

)

Package.json

{

  "name": "client",

  "version": "0.1.0",

  "private": true,

  "dependencies": {

    "@testing-library/jest-dom": "^5.17.0",

    "@testing-library/react": "^13.4.0",

    "@testing-library/user-event": "^13.5.0",

    "react": "^18.2.0",

    "react-dom": "^18.2.0",

    "react-scripts": "5.0.1",

    "web-vitals": "^2.1.4"

  },

  "scripts": {

    "start": "react-scripts start",

    "build": "react-scripts build",

    "test": "react-scripts test",

    "eject": "react-scripts eject"

  },

  "eslintConfig": {

    "extends": [

      "react-app",

      "react-app/jest"

    ]

  },

  "browserslist": {

    "production": [

      ">0.2%",

      "not dead",

      "not op\_mini all"

    ],

    "development": [

      "last 1 chrome version",

      "last 1 firefox version",

      "last 1 safari version"

    ]

  }

}

App.js

import  { useState } from 'react';

import './App.css';

function App() {

  const [name, setName] = useState('');

  const [age, setAge] = useState('');

  const [message, setMessage] = useState('');

  const checkEligibility = () => {

    if (age >= 18) {

      setMessage(`${name}, you are eligible to vote!`);

    } else {

      setMessage(`${name}, you are not eligible to vote yet.`);

    }

  };

  return (

    <div className="container">

      <h1>Check Your Voting Eligibility</h1>

      <form>

        <div className="form-group">

          <label htmlFor="name">Name:</label>

          <input

            type="text"

            className="form-control"

            id="name"

            value={name}

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

          />

        </div>

        <div className="form-group">

          <label htmlFor="age">Age:</label>

          <input

            type="number"

            className="form-control"

            id="age"

            value={age}

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

          />

        </div>

        <button

          type="button"

          className="btn btn-primary"

          onClick={checkEligibility}

        >

          Check Eligibility

        </button>

      </form>

      {message && <p className="message">{message}</p>}

    </div>

  );

}

export default App;

app.css

.container {

  max-width: 400px;

  margin: 50px auto;

  padding: 20px;

  border: 1px solid #ccc;

  border-radius: 8px;

  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

.form-group {

  margin-bottom: 20px;

}

label {

  font-weight: bold;

}

.input {

  width: 100%;

  padding: 10px;

  font-size: 16px;

  border: 1px solid #ccc;

  border-radius: 4px;

}

.btn-primary {

  display: block;

  width: 100%;

  padding: 10px;

  background-color: #007bff;

  color: #fff;

  border: none;

  border-radius: 4px;

  cursor: pointer;

}

.btn-primary:hover {

  background-color: #0056b3;

}

.message {

  margin-top: 20px;

  font-size: 18px;

  font-weight: bold;

}

**login form:**

index.js

import React from 'react'

import ReactDOM from 'react-dom/client'

import App from './App.js'

import './index.css'

ReactDOM.createRoot(document.getElementById('root')).render(

  <React.StrictMode>

    <App />

  </React.StrictMode>,

)

Package.json

{

  "name": "login",

  "version": "0.1.0",

  "private": true,

  "dependencies": {

    "@testing-library/jest-dom": "^5.17.0",

    "@testing-library/react": "^13.4.0",

    "@testing-library/user-event": "^13.5.0",

    "react": "^18.2.0",

    "react-dom": "^18.2.0",

    "react-scripts": "5.0.1",

    "web-vitals": "^2.1.4"

  },

  "scripts": {

    "start": "react-scripts start",

    "build": "react-scripts build",

    "test": "react-scripts test",

    "eject": "react-scripts eject"

  },

  "eslintConfig": {

    "extends": [

      "react-app",

      "react-app/jest"

    ]

  },

  "browserslist": {

    "production": [

      ">0.2%",

      "not dead",

      "not op\_mini all"

    ],

    "development": [

      "last 1 chrome version",

      "last 1 firefox version",

      "last 1 safari version"

    ]

  }

}

App.js

import { useState } from 'react';

import './App.css';

function App() {

  const [username, setUsername] = useState('');

  const [password, setPassword] = useState('');

  const [loggedIn, setLoggedIn] = useState(false);

  const handleSubmit = (e) => {

    e.preventDefault();

    // For simplicity, just log in with any username and password

    if (username === 'user' && password === 'password') {

      setLoggedIn(true);

    } else {

      alert('Invalid username or password');

    }

  };

  const handleLogout = () => {

    setLoggedIn(false);

    setUsername('');

    setPassword('');

  };

  if (loggedIn) {

    return (

      <div className="container">

        <h2>Welcome, {username}!</h2>

        <button className="btn btn-primary" onClick={handleLogout}>

          Logout

        </button>

      </div>

    );

  }

  return (

    <div className="container">

      <h2>Login</h2>

      <form onSubmit={handleSubmit}>

        <div className="form-group">

          <label htmlFor="username">Username:</label>

          <input

            type="text"

            className="form-control"

            id="username"

            value={username}

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

          />

        </div>

        <div className="form-group">

          <label htmlFor="password">Password:</label>

          <input

            type="password"

            className="form-control"

            id="password"

            value={password}

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

          />

        </div>

        <button type="submit" className="btn btn-primary">

          Login

        </button>

      </form>

    </div>

  );

}

export default App;

**2nd MEGHANA’S –**

**Frontend:**

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>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

reportWebVitals();

package.json

{

  "name": "restaurant-order-app",

  "version": "0.1.0",

  "private": true,

  "dependencies": {

    "@testing-library/jest-dom": "^5.17.0",

    "@testing-library/react": "^13.4.0",

    "@testing-library/user-event": "^13.5.0",

    "axios": "^1.6.7",

    "cors": "^2.8.5",

    "react": "^18.2.0",

    "react-dom": "^18.2.0",

    "react-scripts": "5.0.1",

    "web-vitals": "^2.1.4"

  },

  "scripts": {

    "start": "react-scripts start",

    "build": "react-scripts build",

    "test": "react-scripts test",

    "eject": "react-scripts eject"

  },

  "eslintConfig": {

    "extends": [

      "react-app",

      "react-app/jest"

    ]

  },

  "browserslist": {

    "production": [

      ">0.2%",

      "not dead",

      "not op\_mini all"

    ],

    "development": [

      "last 1 chrome version",

      "last 1 firefox version",

      "last 1 safari version"

    ]

  }

}

App.js

import React, { useState } from 'react';

import axios from 'axios';

import './App.css'; // Import the CSS file

function App() {

  const initialOrderState = {

    dish: '',

    quantity: 0,

    specialInstructions: ''

  };

  const [order, setOrder] = useState(initialOrderState);

  const [orderSent, setOrderSent] = useState(false);

  const handleChange = (e) => {

    const { name, value } = e.target;

    setOrder({ ...order, [name]: value });

  };

  const handleSubmit = (e) => {

    e.preventDefault();

    axios.post('http://localhost:5000/order', order)

      .then(response => {

        console.log(response.data);

        setOrderSent(true);

        setOrder(initialOrderState); // Reset the form fields

      })

      .catch(error => {

        console.error('Error:', error);

      });

  };

  return (

    <div className="container">

      <h1>Meghna's Restaurant</h1>

      <form onSubmit={handleSubmit} className="form-container">

        <label>

          Dish:

          <input

type="text"

name="dish"

value={order.dish}

onChange={handleChange} />

        </label>

        <br />

        <label>

          Quantity:

          <input type="number" name="quantity" value={order.quantity} onChange={handleChange} />

        </label>

        <br />

        <label>

          Special Instructions:

          <textarea name="specialInstructions" value={order.specialInstructions} onChange={handleChange} />

        </label>

        <br />

        <button type="submit">Place Order</button>

      </form>

      {orderSent && <p className="success-message">Order has been sent!</p>}

    </div>

  );

}

export default App;

App.css

body {

  font-family: Arial, sans-serif;

  text-align: center;

  background-color: #222;

  color: #fff;

}

.container {

  padding-top: 50px;

}

.form-container {

  max-width: 400px;

  margin: 0 auto;

  padding: 20px;

  border: 1px solid #ccc;

  border-radius: 5px;

  background-color: #00f2ff8e;

}

.form-container label {

  display: block;

  margin-bottom: 10px;

  color: #fff;

}

.form-container input[type="text"],

.form-container input[type="number"],

.form-container textarea {

  width: calc(100% - 20px);

  margin-left: 10px;

  padding: 5px;

  border-radius: 5px;

  border: 1px solid #ccc;

  background-color: #ffffff8e;

  color: #fff;

}

.form-container textarea {

  resize: vertical;

}

.form-container button {

  background-color: #007bff;

  color: #fff;

  padding: 10px 20px;

  border-radius: 5px;

  border: none;

  cursor: pointer;

}

.form-container button:hover {

  background-color: #0056b3;

}

.success-message {

  margin-top: 20px;

  color: green;

}

**BACKEND**

* **Create folder backend**
* **cd backend in terminal**
* **run the commands in terminal - npm init -y**
* **npm install express**
* **npm install -g nodemon**
* **create file server.js**

**server.js (code ):**

const express = require('express');

const cors = require('cors');

const app = express();

const PORT = 5000;

app.use(express.json());

app.use(cors());

app.post('/order', (req, res) => {

  console.log('Received order:', req.body);

  // Handle the order here, e.g., save it to a database

  res.send('Order received successfully');

});

app.listen(PORT, () => {

  console.log(`Server is running on port ${PORT}`);

});

**In package.json(of backend folder) , in scripts add -**

"scripts": {

    "start": "node server.js"

  },

**STUDENT LIST**

**App.js**

import React, { useState, useEffect } from 'react';

import './App.css';

function App() {

  const [students, setStudents] = useState([]);

  useEffect(() => {

    fetchStudents();

  }, []);

  const fetchStudents = async () => {

    try {

      const response = await fetch('http://localhost:5000/students');

      const data = await response.json();

      setStudents(data);

    } catch (error) {

      console.error('Error fetching students:', error);

    }

  };

  return (

    <div className="App">

      <h1>Student Details</h1>

      <ul>

        {students.map(student => (

          <li key={student.id}>

            <strong>Name:</strong> {student.name} <strong>USN:</strong> {student.usn}, <strong>Hobby:</strong> {student.hobby}

          </li>

        ))}

      </ul>

    </div>

  );

}

export default App;

**app.css**

.App {

  text-align: center;

  font-family: Arial, sans-serif;

}

h1 {

  margin-bottom: 20px;

}

ul {

  list-style: none;

  padding: 0;

}

li {

  margin-bottom: 10px;

  border: 1px solid #ccc;

  padding: 10px;

  border-radius: 5px;

  background-color: #f9f9f9;

}

strong {

  font-weight: bold;

  margin-right: 5px;

}

**Backend – server.js**

const express = require('express');

const app = express();

const cors = require('cors');

const fs = require('fs');

app.use(cors());

app.use(express.json());

const studentsData = fs.readFileSync('student.json');

let students = JSON.parse(studentsData);

app.get('/students', (req, res) => {

  res.json(students);

});

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => {

  console.log(`Server is running on port ${PORT}`);

});

**Student.json**

[

    {

        "name":"Faisal Tanveer Khan",

        "usn":"1ds21ai099",

        "hobby":"Time waste"

    },

    {

        "name":"Anamika Jha",

        "usn":"1ds21ai097",

        "hobby":"Reading books"

    },

    {

        "name":"Shadan Siddique",

        "usn":"1ds21ai095",

        "hobby":"Youtube video creation"

    },

    {

        "name":"Rohan Kumar",

        "usn":"1ds21ai093",

        "hobby":"Stand-up comedy"

    },

    {

        "name":"Ujjwala Sinha",

        "usn":"1ds21ai091",

        "hobby":"Painting"

    }

]

{

  "name": "server",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

    "start": "node server.js"

  },

  "keywords": [],

  "author": "",

  "license": "ISC",

  "dependencies": {

    "cors": "^2.8.5",

    "express": "^4.18.2"

  }

}

**6th Mongo DB:**

Initial steps

use student1

db.stud1col1.insert({srn:110,sname:"Rahul",degree:"BCA",sem:6,CGPA:7.9}) //Insert is 10 times, change the values

1. Display all the documents:

```javascript

db.stud1col1.find()

```

2. Display all the students in BCA:

```javascript

db.stud1col1.find({ degree: "BCA" })

```

3. Display all the students in ascending order:

```javascript

db.stud1col1.find().sort({ CGPA: 1 })

```

4. Display first 5 students:

```javascript

db.stud1col1.find().limit(5)

```

5. Display students 5, 6, 7:

```javascript

db.stud1col1.find().skip(4).limit(3)

```

6. List the degree of student "Rahul":

```javascript

db.stud1col1.find({ sname: "Rahul" }, { degree: 1, \_id: 0 })

```

7. Display students' details of 5, 6, 7 in descending order of percentage:

```javascript

db.stud1col1.find().skip(4).limit(3).sort({ CGPA: -1 })

```

8. Display the number of students in BCA:

```javascript

db.stud1col1.find({ degree: "BCA" }).count()

```

**1st CALCULATOR**

**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>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

reportWebVitals();

**App.js**

import React, { useState } from 'react';

import './App.css';

function Calculator() {

  const [result, setResult] = useState('');

  const handleClick = (event) => {

    setResult(result + event.target.value);

  };

  const calculate = () => {

    try {

      setResult(eval(result) || 'Error');

    } catch (error) {

      setResult('Error');

    }

  };

  const clear = () => {

    setResult('');

  };

  return (

    <div className="calculator">

      <input type="text" value={result} />

      <div className="keypad">

        <button onClick={handleClick} value="7">7</button>

        <button onClick={handleClick} value="8">8</button>

        <button onClick={handleClick} value="9">9</button>

        <button onClick={handleClick} value="/">/</button>

        <button onClick={handleClick} value="4">4</button>

        <button onClick={handleClick} value="5">5</button>

        <button onClick={handleClick} value="6">6</button>

        <button onClick={handleClick} value="\*">\*</button>

        <button onClick={handleClick} value="1">1</button>

        <button onClick={handleClick} value="2">2</button>

        <button onClick={handleClick} value="3">3</button>

        <button onClick={handleClick} value="-">-</button>

        <button onClick={handleClick} value="0">0</button>

        <button onClick={handleClick} value=".">.</button>

        <button onClick={calculate}>=</button>

        <button onClick={handleClick} value="+">+</button>

      </div>

      <button onClick={clear} className="clear">Clear</button>

    </div>

  );

}

function App() {

  return (

    <div className="App">

      <h1>Simple Calculator</h1>

      <Calculator />

    </div>

  );

}

export default App;

**app.css**

.calculator {

  width: 300px;

  margin: 0 auto;

  padding: 20px;

  border: 1px solid #ccc;

  border-radius: 5px;

}

.calculator input[type="text"] {

  width: calc(100% - 10px);

  margin-bottom: 10px;

  padding: 10px;

  font-size: 1.5em;

}

.keypad {

  display: grid;

  grid-template-columns: repeat(4, 1fr);

  gap: 5px;

}

button {

  padding: 10px;

  font-size: 1.2em;

  border: 1px solid #ccc;

  border-radius: 5px;

  background-color: #f9f9f9;

  cursor: pointer;

}

button:hover {

  background-color: #e0e0e0;

}

.clear {

  width: calc(100% - 10px);

  margin-top: 10px;

  padding: 10px;

  font-size: 1.2em;

  border: none;

  border-radius: 5px;

  background-color: #ff0000;

  color: #fff;

  cursor: pointer;

}

.clear:hover {

  background-color: #cc0000;

}

**4th NUMBER**

const readline = require('readline');

const rl = readline.createInterface({

  input: process.stdin,

  output: process.stdout

});

// Function to check if a number is even or odd

function checkEvenOrOdd(number) {

  return number % 2 === 0 ? 'even' : 'odd';

}

// Function to find the greatest number among three numbers

function findGreatestNumber(num1, num2, num3) {

  return Math.max(num1, num2, num3);

}

// Function to handle input

function checkNumbers() {

  rl.question("Enter number 1: ", (num1) => {

    rl.question("Enter number 2: ", (num2) => {

      rl.question("Enter number 3: ", (num3) => {

        // Convert input strings to numbers

        num1 = parseInt(num1);

        num2 = parseInt(num2);

        num3 = parseInt(num3);

        // Check if numbers are even or odd

        console.log(`${num1} is ${checkEvenOrOdd(num1)}`);

        console.log(`${num2} is ${checkEvenOrOdd(num2)}`);

        console.log(`${num3} is ${checkEvenOrOdd(num3)}`);

        // Find the greatest number

        const greatest = findGreatestNumber(num1, num2, num3);

        console.log(`The greatest number is ${greatest}.`);

        // Close the readline interface

        rl.close();

      });

    });

  });

}

// Call the function

checkNumbers();

**🡪8th mongoDB Connection**

**Frontend**

**Index.js (same)**

**App.js**

import React, { useEffect, useState } from 'react';

function App() {

    const [users, setUsers] = useState([]);

    useEffect(() => {

        fetchData();

    }, []);

    const fetchData = async () => {

        try {

            const response = await fetch('http://localhost:3000/users');

            const users = await response.json();

            setUsers(users);

        } catch (error) {

            console.error("Error fetching users:", error);

        }

    };

    return (

        <div>

            <h1>Users</h1>

            <ul>

                {users.map(user => (

                    <li key={user.id}>{user.name}</li>

                ))}

            </ul>

        </div>

    );

}

export default App;

**app.css**

.container {

  max-width: 600px;

  margin: 10px;

  padding: 20px;

}

h1 {

  text-align: center;

  font-size: 24px;

  margin-bottom: 20px;

}

ul {

  list-style-type: none;

  padding: 0;

}

li {

  background-color:#f9f9f9;

  padding: 10px;

  margin-bottom: 5px;

  border-radius: 5px;

  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

li:hover {

  background-color: #e0e0e0;

}

**BACKEND**

**npm init -y**

**npm install mongoose cors express**

**Index.js**

const express = require('express');

const cors = require('cors');

const mongoose = require('mongoose');

const app = express();

app.use(express.json());

app.use(cors());

// Database connection

mongoose.connect('mongodb://localhost:27017/mydatabase', {

    useNewUrlParser: true,

    useUnifiedTopology: true

}).then(() => {

    console.log("Connected to database");

}).catch(err => {

    console.error("Connection to database failed:", err);

});

// Schema

const userSchema = new mongoose.Schema({

    id: Number,

    name: String

});

const User = mongoose.model('User', userSchema);

// Route to get users

app.get('/users', async (req, res) => {

    try {

        const users = await User.find();

        res.json(users);

    } catch (error) {

        console.error("Error fetching users:", error);

        res.status(500).json({ error: "Internal Server Error" });

    }

});

// Start the server

const PORT = process.env.PORT || 3000;

app.listen(PORT, () => {

    console.log(`Server started on port ${PORT}`);

});

**7TH API Fetch()**

**Frontend**

**App.js**

import React, { useState, useEffect } from 'react';

import './App.css'; // Import CSS file

function App() {

  const [jsonData, setJsonData] = useState(null);

  useEffect(() => {

    const fetchData = async () => {

      try {

        const response = await fetch('http://localhost:3000/data');

        const data = await response.json();

        setJsonData(data);

      } catch (error) {

        console.error('Error fetching data:', error);

      }

    };

    fetchData();

  }, []);

  return (

    <div className="container"> {/\* Apply container class \*/}

      <div className="data-container"> {/\* Apply data-container class \*/}

        <h2>JSON Data</h2>

        <pre>{JSON.stringify(jsonData, null, 2)}</pre>

      </div>

    </div>

  );

}

export default App;

**app.css**

.container {

  max-width: 800px;

  margin: 0 auto;

  padding: 20px;

}

.data-container {

  background-color: #f0f0f0;

  border-radius: 5px;

  padding: 20px;

  margin-top: 20px;

}

h2 {

  margin-bottom: 10px;

}

pre {

  background-color: #e0e0e0;

  padding: 10px;

  border-radius: 5px;

  overflow-x: auto;

  font-family: 'Courier New', Courier, monospace;

}

**Backend**

**Server.js**

const express= require('express')

const cors=require('cors')

const data=require('./data.json')

const app=express()

app.use(cors())

app.use(express.json())

const mongoose=require('mongoose')

mongoose.connect('mongodb://localhost:27017/mydatabase').then(()=>{

    console.log("connected to database");

})

app.get('/data',(req,res)=>{

         res.json(data)

})

app.listen(3000,()=>{

    console.log("Server is listening on port 3000");

})

**Data.json**

[{

    "name":"Krutika",

    "usn":"28",

    "hobby":"Drawing"

},

{

    "name":"K",

    "usn":"228",

    "hobby":"Poetry"

}

]

**3rd API-STUDENTS**

**App.js**

import React, { useState, useEffect } from 'react';

import './App.css';

function App() {

  const [students, setStudents] = useState([]);

  useEffect(() => {

    fetchData();

  }, []);

  const fetchData = async () => {

    try {

      const response = await fetch('http://localhost:5000/api/students');

      const data = await response.json();

      setStudents(data);

    } catch (error) {

      console.error('Error fetching data:', error);

    }

  };

  return (

    <div className="App">

      <h1>Student Details</h1>

      <ul>

        {students.map((student, index) => (

          <li key={index}>

            <strong>Name:</strong> {student.name}, <strong>USN:</strong> {student.usn}, <strong>Hobby:</strong> {student.hobby}

          </li>

        ))}

      </ul>

    </div>

  );

}

export default App;

**App.css**

.App {

  font-family: Arial, sans-serif;

  text-align: center;

  margin-top: 50px;

}

h1 {

  font-size: 24px;

  margin-bottom: 20px;

}

ul {

  list-style: none;

  padding: 0;

}

li {

  margin-bottom: 10px;

  padding: 10px;

  border: 1px solid #ccc;

  border-radius: 5px;

  background-color: #f9f9f9;

}

strong {

  font-weight: bold;

  margin-right: 5px;

}

**Backend**

**Server.js**

const express = require('express');

const cors = require('cors');

const fs = require('fs');

const app = express();

app.use(cors());

app.get('/api/students', (req, res) => {

  try {

    const studentsData = fs.readFileSync('student.json', 'utf8');

    const students = JSON.parse(studentsData);

    res.json(students);

  } catch (error) {

    console.error('Error reading students data:', error);

    res.status(500).send('Internal Server Error');

  }

});

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => {

  console.log(`Server is running on port ${PORT}`);

});

**Students.json**

[

    { "name": "A", "usn": "001", "hobby": "Reading" },

    { "name": "B", "usn": "002", "hobby": "Gaming" },

    { "name": "C", "usn": "003", "hobby": "Sports" },

    { "name": "D", "usn": "004", "hobby": "Music" },

    { "name": "E", "usn": "005", "hobby": "Traveling" }

  ]