**Demonstrate connecting frontend and backend by developing a frontend in react that connects to the REST Controller backend developed previously.**

**Frontend:**

**App.js:**

import { useState } from 'react';

import './App.css';

function App() {

  const [num1, setNum1] = useState('');

  const [num2, setNum2] = useState('');

  const [op, setOp] = useState('+');

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

  const handleNum1 = (e) => setNum1(e.target.value);

  const handleNum2 = (e) => setNum2(e.target.value);

  const handleOp = (e) => setOp(e.target.value);

  const handleCalculate = () => {

    let opr='';

      if (op === '+') {

        opr = 'Add' ;

      } else if (op === '-') {

        opr = 'Sub';

      } else if (op === '\*') {

        opr = 'Mul';

      } else if (op === '/') {

        opr =  'Div' ;

      } else {

        opr = 'Invalid operation';

      }

    fetch(`http://localhost:8080/${num1}/${num2}/${opr}`)

      .then(response => response.json())

      .then(data => setResult(data.Result))

      .catch(error => console.error('Error:', error));

  };

  const handleGreetClick = () => {

    fetch('http://localhost:8080')

      .then(response => response.text())

      .then(data => alert(data))

      .catch(error => console.error('Error:', error));

  };

  const handleGreetObjClick = () => {

    fetch('http://localhost:8080/greet')

      .then(response => response.json())

      .then(data => alert(data.Message))

      .catch(error => console.error('Error:', error));

  };

  return (

    <div className="App">

      <center>

        <h1>Full Stack</h1>

        <button onClick={handleGreetClick}>Greet</button>

        <button onClick={handleGreetObjClick}>Message</button>

        <div>

          <label htmlFor="num1">Num1</label>

          <input type="text" value={num1} id="num1" onChange={handleNum1} /><br />

          <label htmlFor="num2">Num2</label>

          <input type="text" value={num2} id="num2" onChange={handleNum2} />

        </div>

        <div>

          <label htmlFor="op">Operation</label>

          <select value={op} onChange={handleOp}>

            <option value="+">+</option>

            <option value="-">-</option>

            <option value="\*">\*</option>

            <option value="/">/</option>

          </select>

        </div>

        <button onClick={handleCalculate}>Calculate</button>

        <div>

          <label htmlFor="result">Result</label>

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

        </div>

      </center>

    </div>

  );

}

export default App;

**App.css:**

.App {

  text-align: center;

}

.App {

  font-family: Arial, sans-serif;

  margin-top: 50px;

  background-color: aqua;

}

h1 {

  font-size: 2.5em;

  color: #333;

}

button {

  background-color: #2e56c4;

  color: white;

  padding: 10px 20px;

  margin: 10px 5px;

  border: none;

  border-radius: 5px;

  cursor: pointer;

  font-size: 1.2em;

  transition: background-color 0.3s ease;

}

button:hover {

  background-color: #45a049;

}

input[type="text"] {

  width: 200px;

  padding: 10px;

  margin: 10px 0;

  box-sizing: border-box;

  border: 2px solid #ccc;

  border-radius: 5px;

  font-size: 1em;

}

select {

  padding: 10px;

  margin: 10px 0;

  font-size: 1em;

  border: 2px solid #ccc;

  border-radius: 5px;

  cursor: pointer;

}

label {

  display: block;

  font-size: 1.2em;

  margin-top: 20px;

}

#result {

  font-size: 1.5em;

  padding: 10px;

  margin-top: 20px;

  width: 200px;

  text-align: center;

  border: 2px solid #4CAF50;

  border-radius: 5px;

  background-color: #f9f9f9;

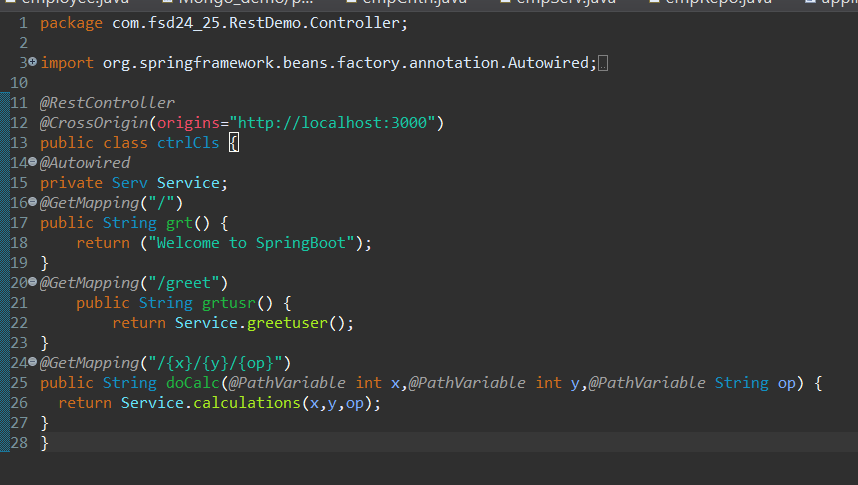
}

.center {

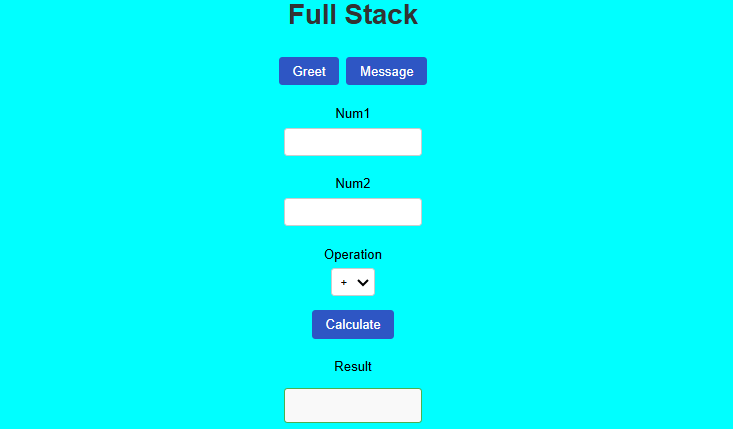
  text-align: center;

}

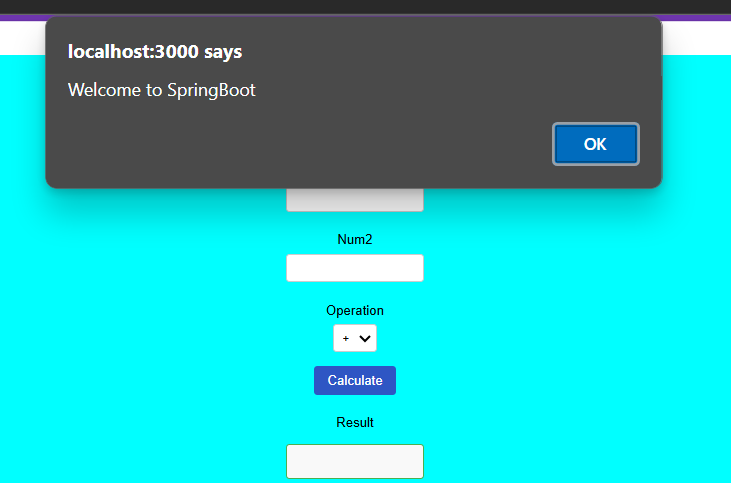
**Backend:**



**Output:**



When we click on Greet



When we click on Message

