Ex.No: 8 **UIT2602 WEB PROGRAMMING**
DATE:17/4/2024

CALCULATOR RESTFUL WEB SERVICE

Name: Sabarish Sankaran B

Reg No: 3122215002087

1. AIM:

The aim of this program is to create a RESTful web service for a calculator application, allowing users to perform arithmetic operations remotely via HTTP requests. The program will utilise AngularJS for the client-side interface and Express.js for the server-side implementation.

2. WEB TOOLS AND METHODOLOGY:

- Node . js: Node.js's asynchronous I/O model.
- Express: Express's minimalist framework for creating server-side applications."
- WebBrowser: To Test The Functionality Of the Website.

3. IMPLEMENTATION AND PROCEDURE:

a. Install Node.js:

- Ensure Node.js installed on the machine.
- Download and install it from the official Node.js website: Node.js Downloads.

b. Set Up Project Directory:

- Create a new directory for your project.
- Inside the project directory, create two files: index.html for the client-side code and server.js for the server-side code.

c. *Install Dependencies:*

- Run the following command to install the Express.js dependency for your server: npm install express (cmd) at project directory.

d. *Run the Server:*

- In the terminal, while still in the project directory, run the following command to start the Express.js server: node server.js (cmd)

e. *Access the Calculator App:*

- Access the Calculator App.
- Open a web browser and navigate to http://localhost:3000.
- This will load the HTML file with the AngularJS code.
- The calculator application interface with input fields for operands, a dropdown for the operator, and a "Calculate" button appears.
- Enter the operands and select an operator, then click the "Calculate" button to perform the calculation.
- The result should be displayed below.

4. SOURCE CODE:

❖ */public/Index.html:*

```
<!DOCTYPE html>
<html lang="en" ng-app="calculatorApp">
<head>
  <meta charset="UTF-8">
  <title>Calculator</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></sc
```

```
ript>
  <script src="calculator.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color:#eab676; /* Light gray background */
      padding: 20px;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh; /* Make the body full height of the viewport */
    }

    .calculator {
      background-color: #ffffff; /* White background */
      border-radius: 10px;
      padding: 20px;
      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); /* Soft shadow */
      width: 300px;
    }

    .calculator input[type="number"],
    .calculator select {
      padding: 8px;
      margin-right: 10px;
      border: 1px solid #cccccc; /* Light gray border */
      border-radius: 5px;
      font-size: 16px;
      width: 80px;
    }

    .calculator button {
      padding: 8px 16px;
      background-color: #4caf50; /* Green button */
```

```
            color: #ffffff; /* White text */
            border: none;
            border-radius: 5px;
            cursor: pointer;
            font-size: 16px;
        }

        .calculator button:hover {
            background-color: #45a049; /* Darker green button on hover */
        }

        .calculator p {
            font-size: 18px;
            margin-top: 10px;
        }
    </style>
</head>
<body ng-controller="CalculatorController as calcCtrl">
    <div class="calculator">
        <input type="number" ng-model="calcCtrl.firstOperand">
        <select ng-model="calcCtrl.operator">
            <option value="+">+</option>
            <option value="-">-</option>
            <option value="*">*</option>
            <option value="/">/</option>
        </select>
        <input type="number" ng-model="calcCtrl.secondOperand">
        <button ng-click="calcCtrl.calculate()">Calculate</button>
        <p>Result: {{ calcCtrl.answer }}</p>
    </div>
</body>
</html>
```

❖ */public/calculator.js:*

```
angular.module('calculatorApp', [])
  .controller('CalculatorController', ['$http', function($http) {
    var self = this;
    self.firstOperand = 0;
    self.secondOperand = 0;
    self.operator = '+';
    self.answer = '';

    self.calculate = function() {
      $http.get('/calculate', {
        params: {
          first: self.firstOperand,
          second: self.secondOperand,
          operator: self.operator
        }
      }).then(function(response) {
        if (response.data.result !== undefined) {
          self.answer = response.data.result;
        } else {
          self.answer = 'Error: ' + response.data.error;
        }
      }).catch(function(error) {
        console.error('Error occurred while calculating:', error);
        self.answer = 'Error occurred while calculating';
      });
    };
  }]);
```

❖ */server.js:*

```
const express = require('express');
const app = express();
const port = 3000;

// Serve static files from the public directory
app.use(express.static('public'))
// Route handler for GET request to /calculate
app.get('/calculate', (req, res) => {
    const { first, second, operator } = req.query;
    const num1 = parseFloat(first);
    const num2 = parseFloat(second);
    let result;

    switch(operator) {
        case '+':
            result = num1 + num2;
            break;
        case '-':
            result = num1 - num2;
            break;
        case '*':
            result = num1 * num2;
            break;
        case '/':
            result = num1 / num2;
            break;
        default:
            result = 'Invalid operator';
    }

    res.json({ result });
});

app.listen(port, () => {
    console.log(`Server is running on http://localhost:${port}`);
});
```

5. OUTPUT SCREENSHOTS:

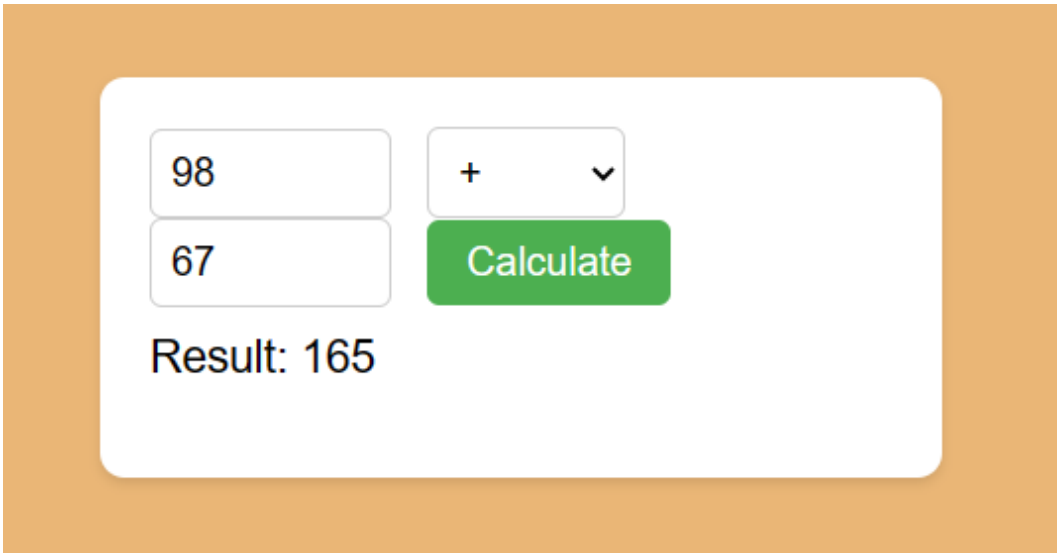
❖ *Install Express :*

```
PS C:\Users\LENOVO\Downloads\Calculator> npm install express

up to date, audited 66 packages in 833ms

12 packages are looking for funding
```

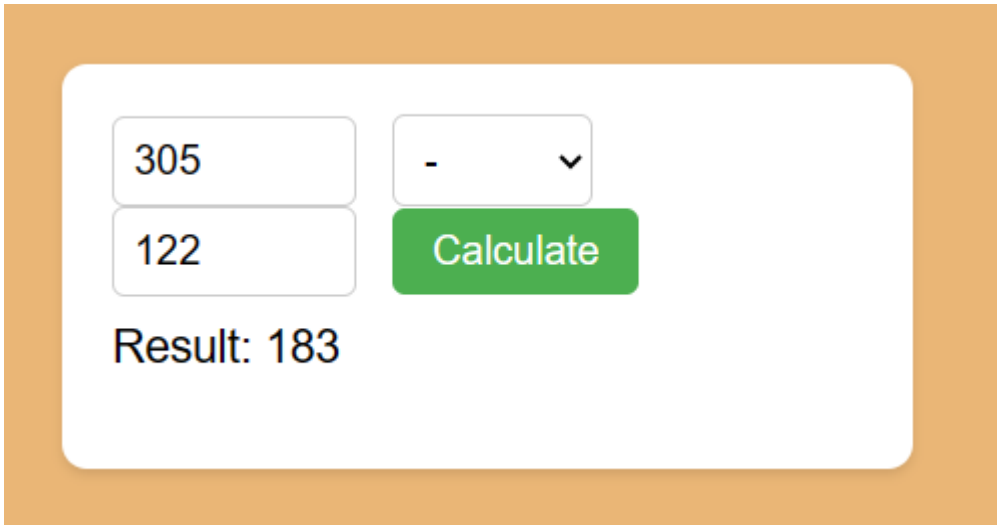
❖ *Run the server:*

```
PS C:\Users\LENOVO\Downloads\Calculator -1> node server.js
Server is running on http://localhost:3000
```

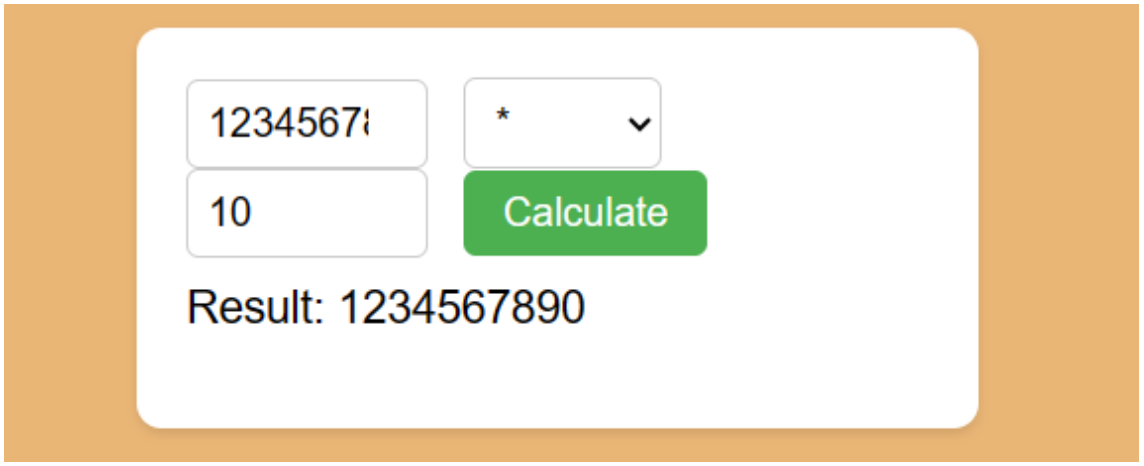
❖ *Addition:*



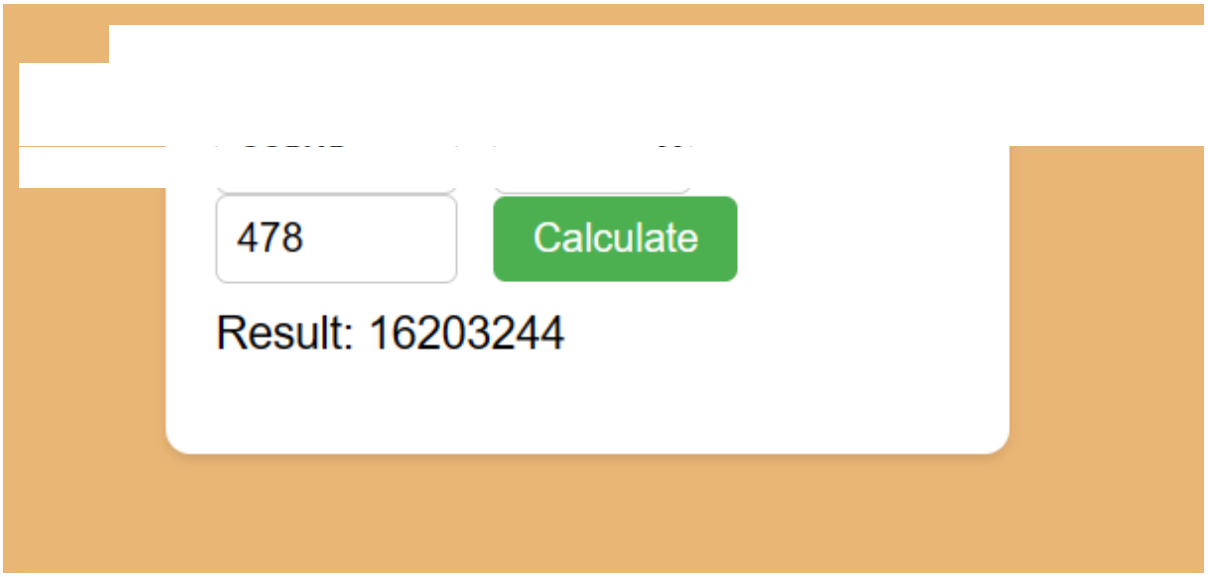
❖ *Subtraction:*



❖ *Multiplication:*



❖ *Division:*



6. CONCLUSION:

Users can input operands and select operators through the user interface, triggering a calculation process handled by the AngularJS controller. On the server side, Express.js routes are configured to handle incoming requests, extract operands and operators, perform the required arithmetic operations, and send back the result to the client. This program provides a seamless and efficient way to perform calculations over the web.