

Simple web application using MEAN Stack

Name:	M Badri Narayanan
Reg No:	185002018
Semester:	VI
Exercise Number:	12
Date:	April 27, 2021

Aim

To write a program to implement RESTful web service for calculator application.

Procedure

- Get the operands using input tag and the operator using option tag.
 - The div enclosing the calculator is directed to the corresponding ng-app and ng-controller.
 - Have a calculate button whose ng-click is directed to the calculate function in the controller.
 - Have a tag for answer which is linked to the result passed by the controller.
 - Inside the controller get the operator and pass it to the \$http.get function and pass the first and second operands as query params in the get function.
 - Store the result in \$scope.answer.
-

For the Server

- Connect to express.
 - Based on different paths, route app.get and perform the respective operations.
 - Store query first and query second in two variables.
 - Convert the variables to numbers.
 - Perform the required operation using the two numbers and send the result.
-

Java Script Code (Calc.js)

```
const express = require("express");
const app = express();
app.use(express.static("public"));
app.get("/", (req, res) => {
    res.sendFile(__dirname + "/RESTCalc.html");
});
app.get("/add", (req, res) => {
    var first = req.query.first;
    var second = req.query.second;
    var value = Number(first) + Number(second);
    var result = {result: value};
    res.send(result);
});
app.get("/subtract", (req, res) => {
    var first = req.query.first;
    var second = req.query.second;
    var value = Number(first) - Number(second);
    var result = {result: value};
    res.send(result);
});
app.get("/multiply", (req, res) => {
    var first = req.query.first;
    var second = req.query.second;
    var value = Number(first) * Number(second);
    var result = {result: value};
    res.send(result);
});
app.get("/divide", (req, res) => {
    var first = req.query.first;
    var second = req.query.second;
    var value = Number(first) / Number(second);
    var result = {result: value};
    res.send(result);
});
app.listen(7000);
```

HTML Code (RESTCalc.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Calclator</title>
    <style>
        body{
        }
    </style>
</head>
<body>
<script
    ↪ src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
<center><h3>CALCULATOR</h3>
    <div ng-app="CalculatorApp" ng-controller="CalculatorController">
        <p><b>0</b>operand 1</b></p>
        <p><input type="number" ng-model="first"></p>
```

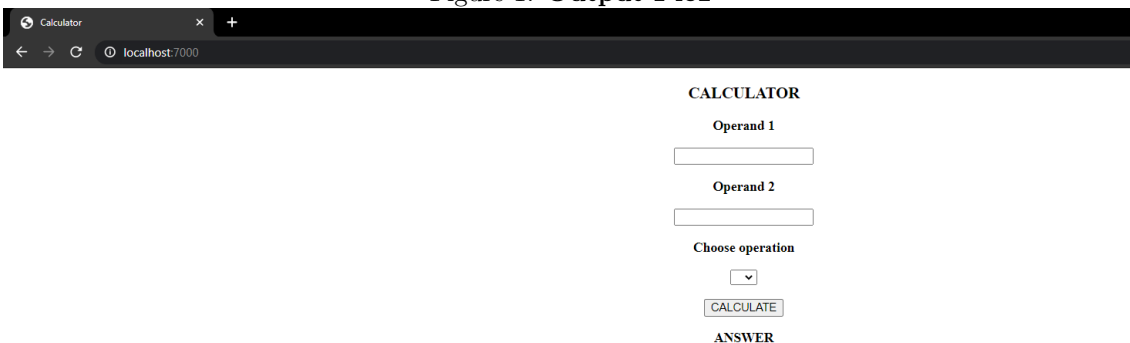
```

<p><b>Operand 2</b></p>
<p><input type="number" ng-model="second"></p>
<p><b>Choose operation</b></p>
<p><select ng-model="operator">
  <option value="add">+</option>
  <option value="subtract">-</option>
  <option value="multiply">*</option>
  <option value="divide">/</option>
</select></p>
<button ng-click="calculate()">CALCULATE</button>
<p><b>ANSWER</b></p>
<p ng-model="answer">{{answer}}</p>
</div>
</center>
<script>
  angular.module('CalculatorApp', [])
    .controller('CalculatorController', function ($scope, $http) {
      $scope.calculate = function () {
        ops = $scope.operator;
        $http.get(ops,
          {
            params: {
              first: $scope.first,
              second: $scope.second
            }
          })
          .success(function (res) {
            console.log('Exit status ' + JSON.stringify(res));
            $scope.answer = res.result;
          });
      };
    });
</script>
</body>
</html>

```

Output

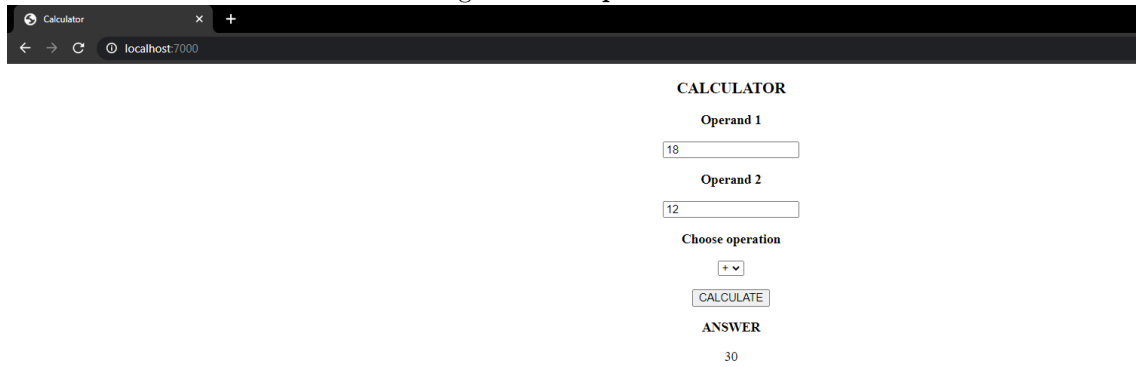
Figure 1: Output Pic1



The screenshot shows a web browser window with the title 'Calculator' and the address bar displaying 'localhost:7000'. The page content is a simple calculator interface with the following elements:

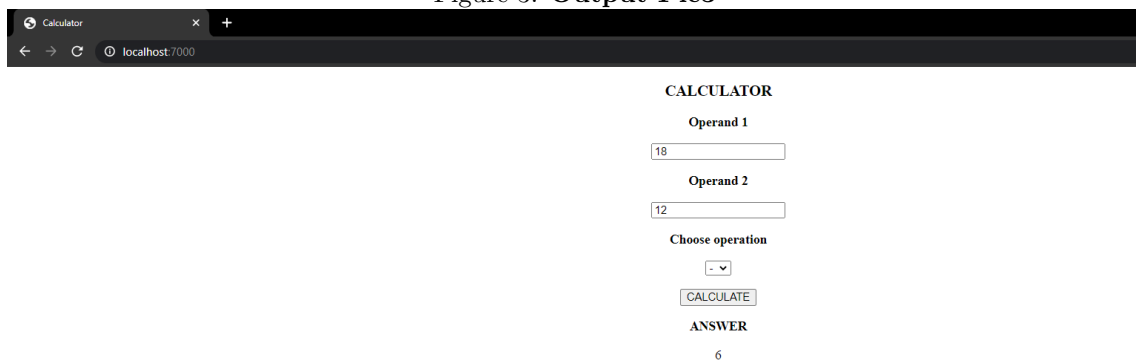
- CALCULATOR**: The main heading of the application.
- Operand 1**: A text input field for the first operand.
- Operand 2**: A text input field for the second operand.
- Choose operation**: A dropdown menu with a downward arrow icon.
- CALCULATE**: A button to perform the calculation.
- ANSWER**: A field to display the result of the calculation.

Figure 2: Output Pic2



A screenshot of a web browser window titled "Calculator" with a single tab. The address bar shows "localhost:7000". The page content is a simple calculator interface. It has two input fields: "Operand 1" containing the number "18" and "Operand 2" containing the number "12". Below these is a "Choose operation" section with a dropdown menu showing a plus sign (+) and a "CALCULATE" button. At the bottom, the "ANSWER" is displayed as "30".

Figure 3: Output Pic3



A screenshot of a web browser window titled "Calculator" with a single tab. The address bar shows "localhost:7000". The page content is a simple calculator interface. It has two input fields: "Operand 1" containing the number "18" and "Operand 2" containing the number "12". Below these is a "Choose operation" section with a dropdown menu showing a minus sign (-) and a "CALCULATE" button. At the bottom, the "ANSWER" is displayed as "6".

Result

A RESTful web service for calculator application was implemented successfully using node and express.
