Restful Webservice For Calculator Application

Exp No.11 **Date:29-04-21**

Objective:

- To make a calculator using Restful webservice using REST API , Express and NodeJS.

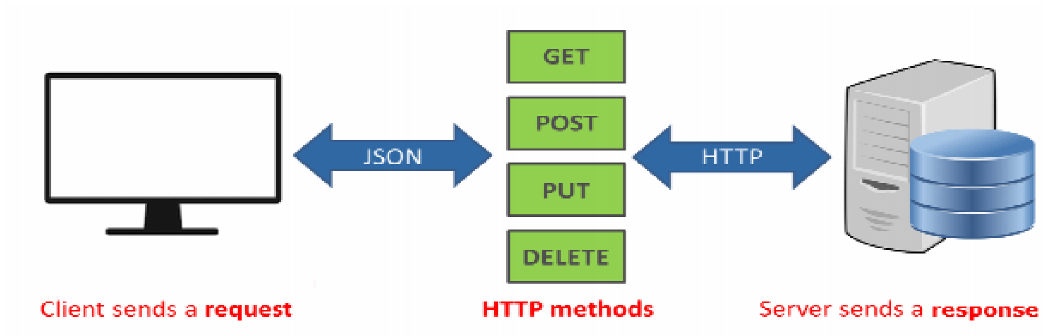
Requirements:

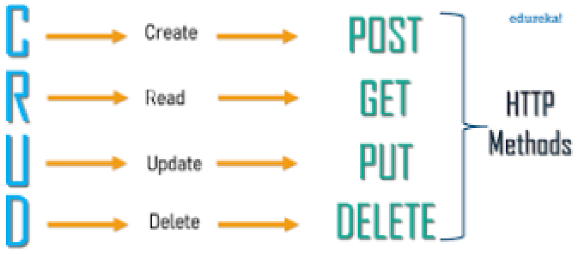
- **Front End Requirements:**
  1. Web Browser HTML support
  2. HTML
  3. CSS
- **Back End Requirements:**
  1. NodeJS
  2. Express
  3. Hapi
- **Text Editor/IDE**

Theory:

Rest-API

EXPRESS – REST API EXAMPLES





- Restful Web Services is a lightweight, maintainable, and scalable service that is built on the REST architecture. Restful Web Service, expose API from your application in a secure, uniform, stateless manner to the calling client.
- The calling client can perform predefined operations using the Restful service.
- The underlying protocol for REST is HTTP. REST stands for representational State Transfer.

- **Node js:**
    - Node js is a free and an open source environment
    - It used javascript to run on server.
    - Nodejs is used to generate dynamic page content.
    - It executes javascript code outside browser.

- **Express:**
    - Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications. Following are some of the core features of Express framework –
    - Allows to set up middlewares to respond to HTTP Requests.
    - Defines a routing table which is used to perform different actions based on HTTP Method and URL.

- **Hapi:**
    - **hapi.js** (short for Http-API, pronounced *happy* and also known as *hapi*) is an open-source framework for developing scalable web applications. One of the most basic use case of hapi is to build REST APIs. You can build application programming interface (API) servers, websites, and HTTP proxy applications with hapi.js**.**

Procedure:

1. Using following shell commands – create working directory
   mkdir calc-service
   cd calc-service
2. Create project.json in **Visual Studio Code**
3. Create a our project.json file, by typing this command: npm init
4. Enter the following command:
   npm install --save hapi@17.x.x

5.Create the Server.js file.

6. Create all routes in routes.js.

7.Above code creates a route to a give path of /calculator/about using a GET method. In the line 4 handler function handles the request and send back a reply as the response.

8.Create calUI.html with textboxes to input number with 4 buttons +,-,*,/.
9. Create onclick for all buttons and generate dynamic URL .
10.We have used GET method to send the request an the URL matches the path of the route. We have a successful return from the server as a JSON string.
11.Start the server using node server.js.
12.Open calc.html in the web browser and give the numbers as input and click the appropriate button and you will get the result on the button click.

Implementation:

Sever.js

```
const Hapi = require('hapi');

const host = 'localhost';
const port = 3000;

const server = Hapi.Server({
   host: host,
   port: port
});

const init = async () => {

   await server.start();
   console.log("Server up and running at port: " + port);

}

//Setup the routes
require('./routes/routes')(server);

init();
```

routes.js

```
module.exports = function(server) {
   //About route
   server.route({
      method: 'GET',
      path: '/calculator/about',
      handler: function (request, h) {

         var data = {
            message: 'Welcome to our Calculator Service'
         };

         return data;
      }
```

```
});

//Add route
server.route({
    method: 'GET',
    path: '/calculator/add/{num1}/{num2}',
    handler: function (request, h) {

        const num1 = parseInt(request.params.num1);
        const num2 = parseInt(request.params.num2);

        var data = {
            answer: num1 + num2
        };

        return data;
    }
});

//Subtract route
server.route({
    method: 'GET',
    path: '/calculator/sub/{num1}/{num2}',
    handler: function (request, h) {

        const num1 = parseInt(request.params.num1);
        const num2 = parseInt(request.params.num2);

        var data = {
            answer: num1 - num2
        };

        return data;
    }
});

//Multiply route
server.route({
    method: 'GET',
    path: '/calculator/multi/{num1}/{num2}',
    handler: function (request, h) {

        const num1 = parseInt(request.params.num1);
        const num2 = parseInt(request.params.num2);

        var data = {
            answer: num1 * num2
        };
        return data;
      }
   });

   //Division route
   server.route({
      method: 'GET',
      path: '/calculator/div/{num1}/{num2}',
      handler: function (request, h) {

         const num1 = parseInt(request.params.num1);
         const num2 = parseInt(request.params.num2);

         var data = {
            answer: num1 / num2
         };

         return data;
      }
   });
}
```

Calcui.html

```
<html>
   <head>
      <link rel="stylesheet" href="calc.css">
   </head>
   <title>Calculator using Rest API</title>
   <body>
      <center><h1>Calculator</h1></center>
      <div id="calform">
      <form>
         <p>Enter First Number:<input type="number" id="num1"></p>
         <p>Enter Second Number:<input type="number" id="num2"></p>
         <input type="button" id="btn1" value="+" onclick="add()">
         <input type="button" id="btn2" value="-" onclick="sub()">
         <input type="button" id="btn3" value="*" onclick="mul()">
         <input type="button" id="btn4" value="/" onclick="div()">

      </form>
      </div>
      <script>

         function add()
         {
            var x=document.getElementById('num1').value;
            var y=document.getElementById('num2').value;
            document.getElementById("btn1").onclick = function () {
```

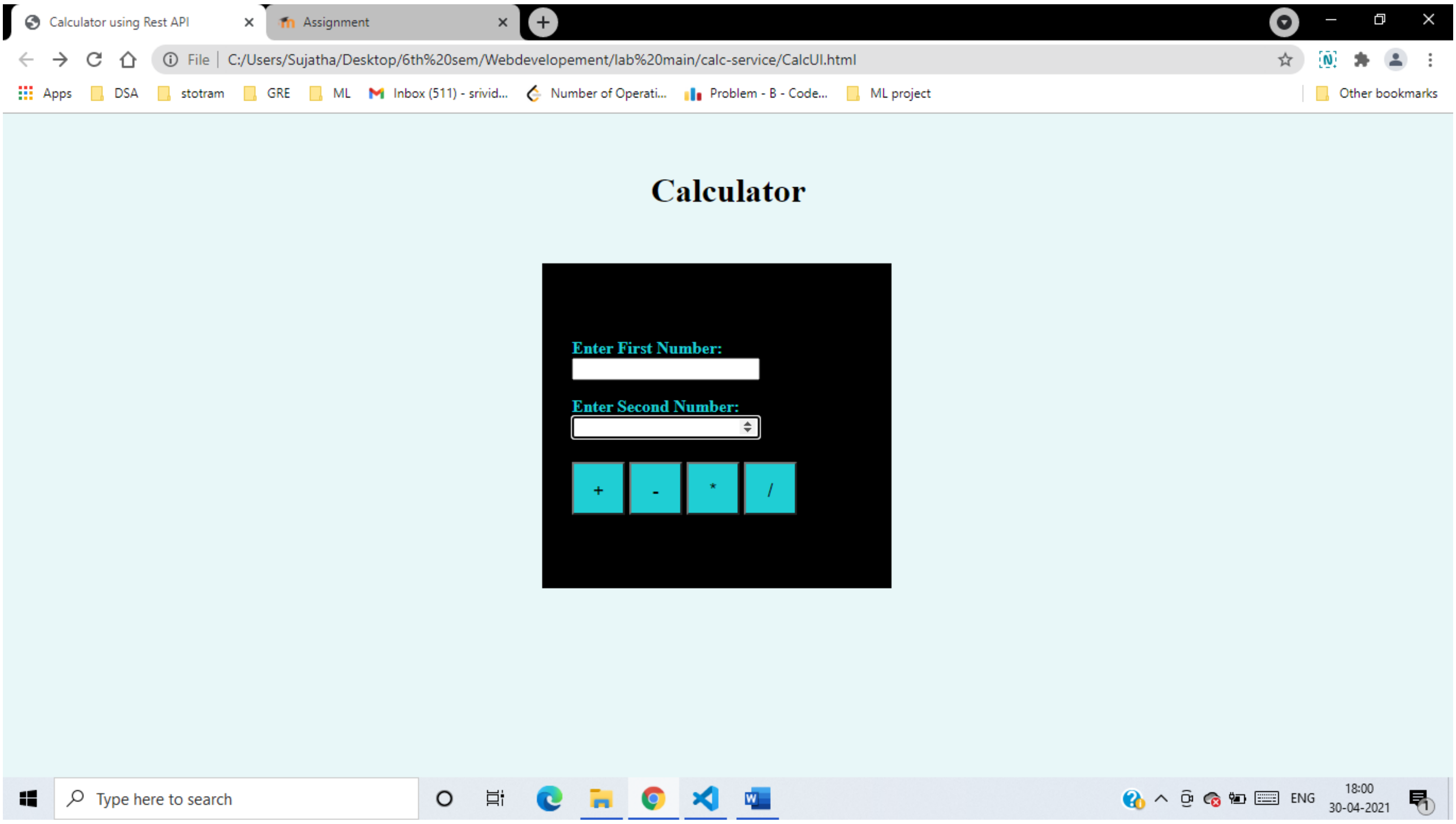
```
          location.href = "http://localhost:3000/calculator/add"+"/"+x+"/"+y;
        };
      }
      function sub()
      {
        var x=document.getElementById('num1').value;
        var y=document.getElementById('num2').value;
        document.getElementById("btn2").onclick = function () {
          location.href = "http://localhost:3000/calculator/sub"+"/"+x+"/"+y;
        };
      }
      function mul()
      {
        var x=document.getElementById('num1').value;
        var y=document.getElementById('num2').value;
        document.getElementById("btn3").onclick = function () {
          location.href = "http://localhost:3000/calculator/multi"+"/"+x+"/"+y;
        };
      }
      function div()
      {
        var x=document.getElementById('num1').value;
        var y=document.getElementById('num2').value;

        document.getElementById("btn4").onclick = function () {
          location.href = "http://localhost:3000/calculator/div"+"/"+x+"/"+y;
        };
      }
    </script>
  </body>
</html>
```

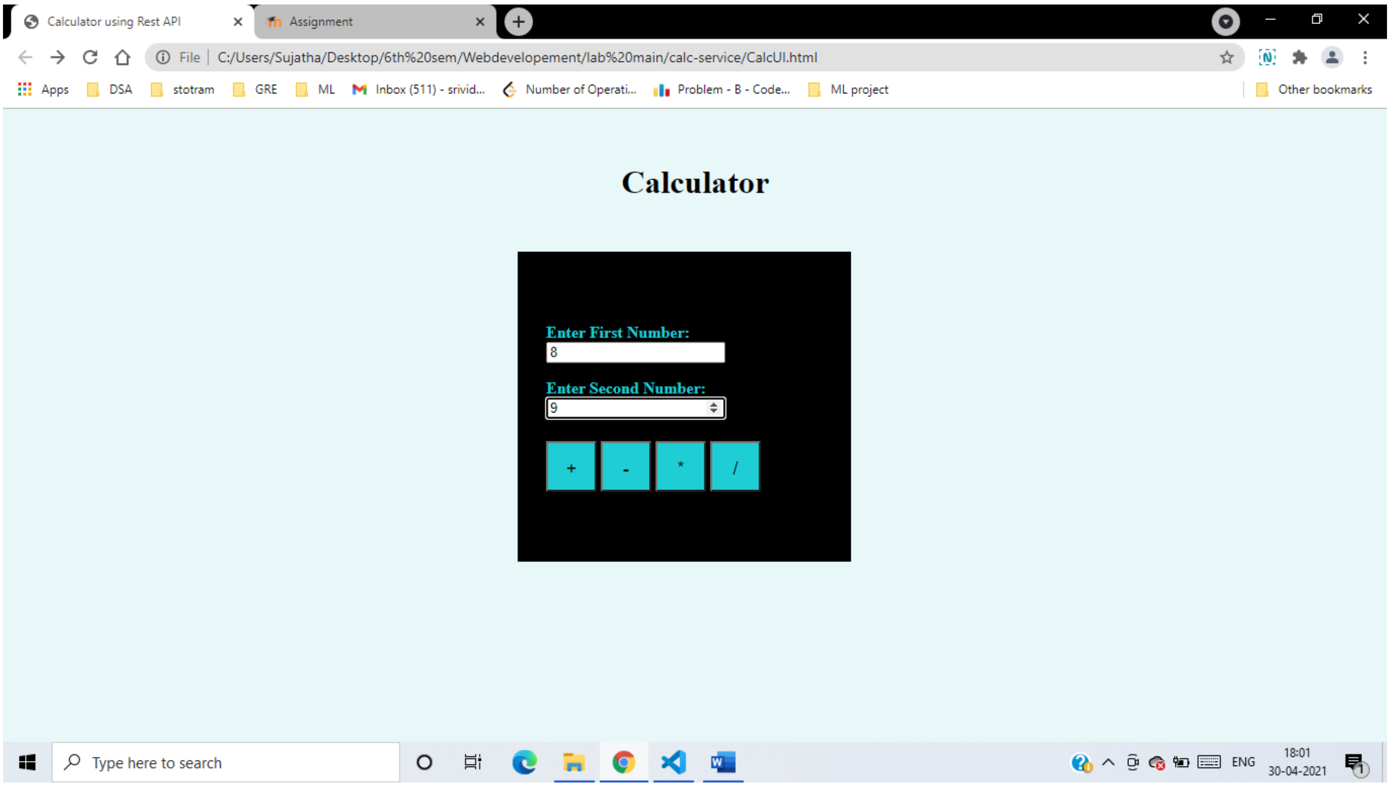
Calc.css

```
body{
  background-color: rgb(232, 247, 248);
}
#calform{
  border:1px black solid;
  margin-top:50px;
  margin-left:500px;
  padding-left:2%;
  padding-top:4%;
  height:250px;
  width:300px;
  background-color: black;
  color: rgb(31, 206, 212);
  font-weight: bold;
}
```

```
#btn1,#btn2,#btn3,#btn4{
    height:50px;
    width:50px;
    font-size: large;
    margin-top:2%;
    padding-top:2%;
    background-color:  rgb(31, 206, 212);
}
h1{
    margin-top:4%;
}
```

Output:




Calculator using Rest API
Assignment
File | C:/Users/Sujatha/Desktop/6th%20sem/Webdevelopement/lab%20main/calc-service/CalcUI.html
Apps DSA stotram GRE ML Inbox (511) - srivid... Number of Operati... Problem - B - Code... ML project Other bookmarks
Type here to search
ENG 18:01 30-04-2021

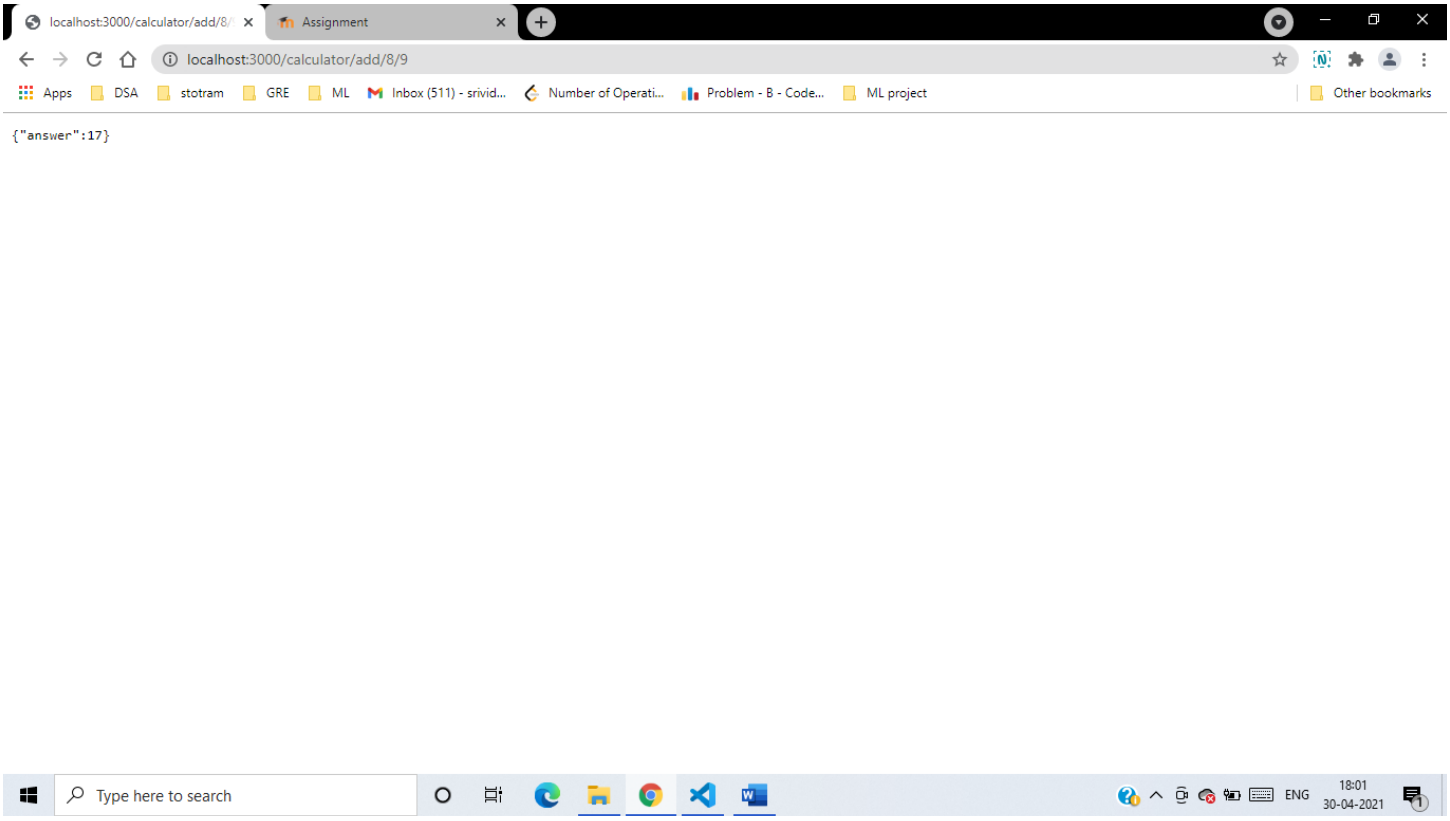
Calculator

Enter First Number:
8

Enter Second Number:
9

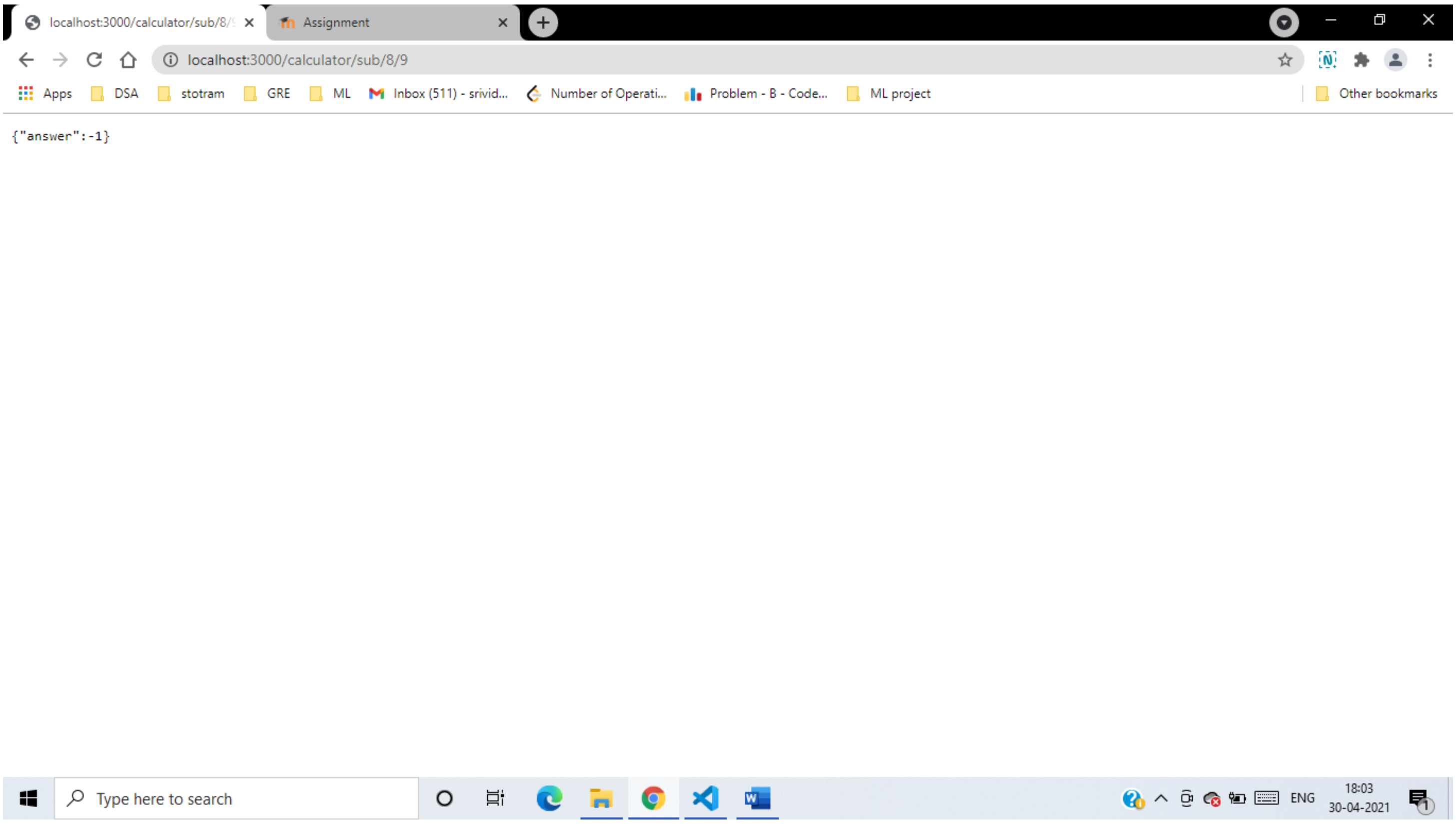
+ - * /

We give two numbers as input as on clicking add(+) button the result is displayed.


localhost:3000/calculator/add/8/
Assignment
localhost:3000/calculator/add/8/9
Apps DSA stotram GRE ML Inbox (511) - srivid... Number of Operati... Problem - B - Code... ML project Other bookmarks
Type here to search
ENG 18:01 30-04-2021

```
{"answer":17}
```

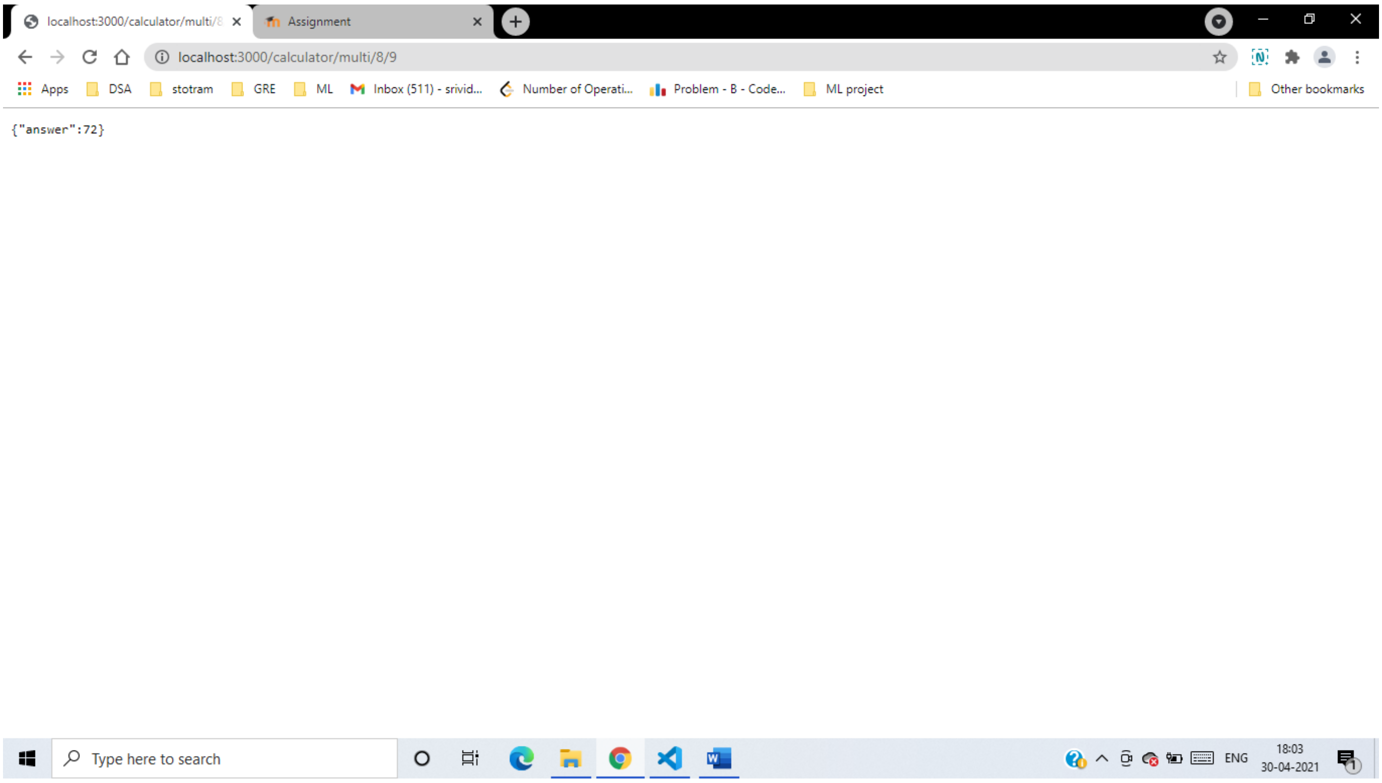
On clicking subtract(-) button the result is displayed.



localhost:3000/calculator/sub/8/9

```
{"answer":-1}
```

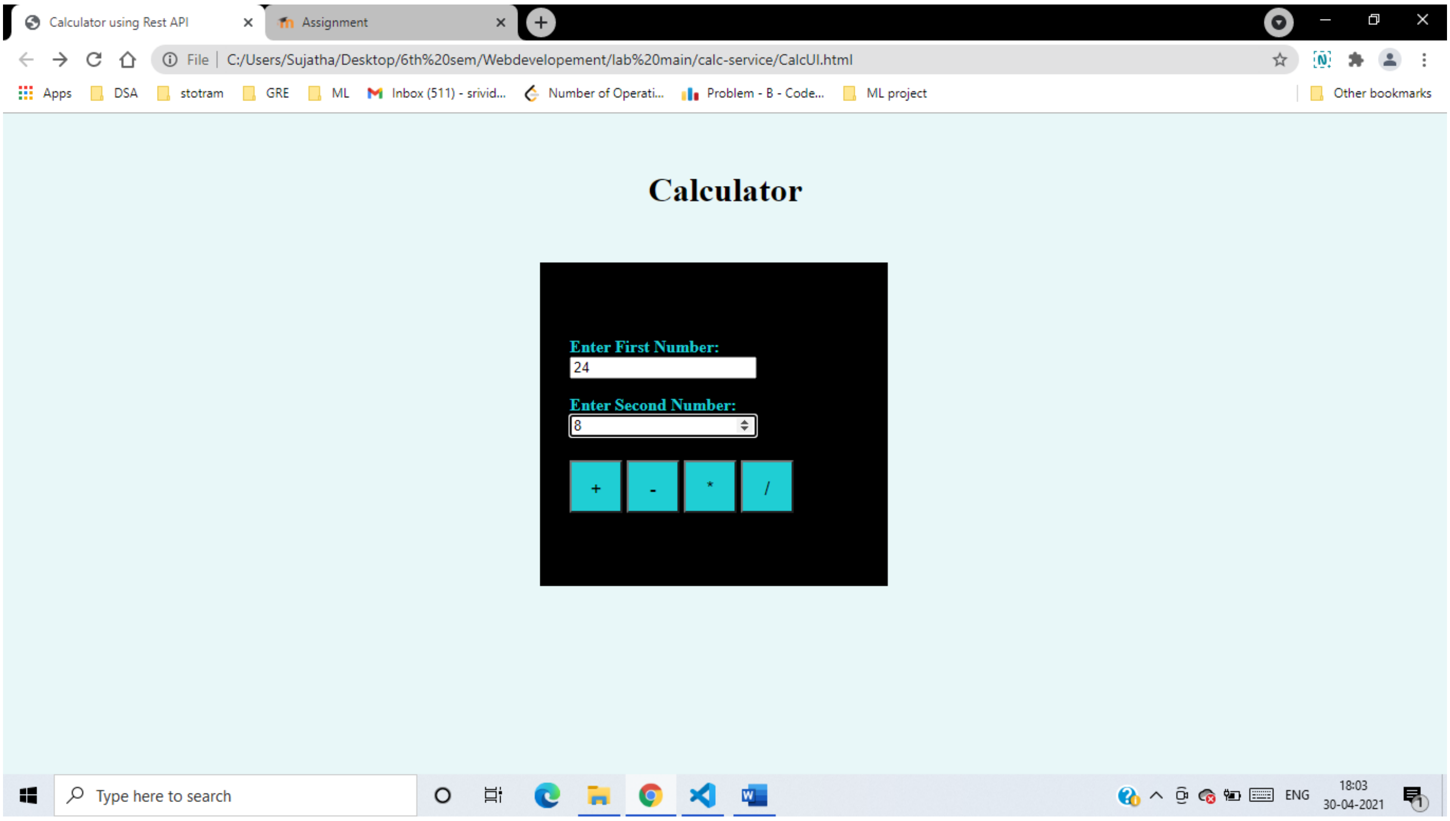
On clicking multiply button(*) the result is displayed.



localhost:3000/calculator/multi/8/9

```
{"answer":72}
```

We give two numbers as input as on clicking division(/) button the result is displayed.

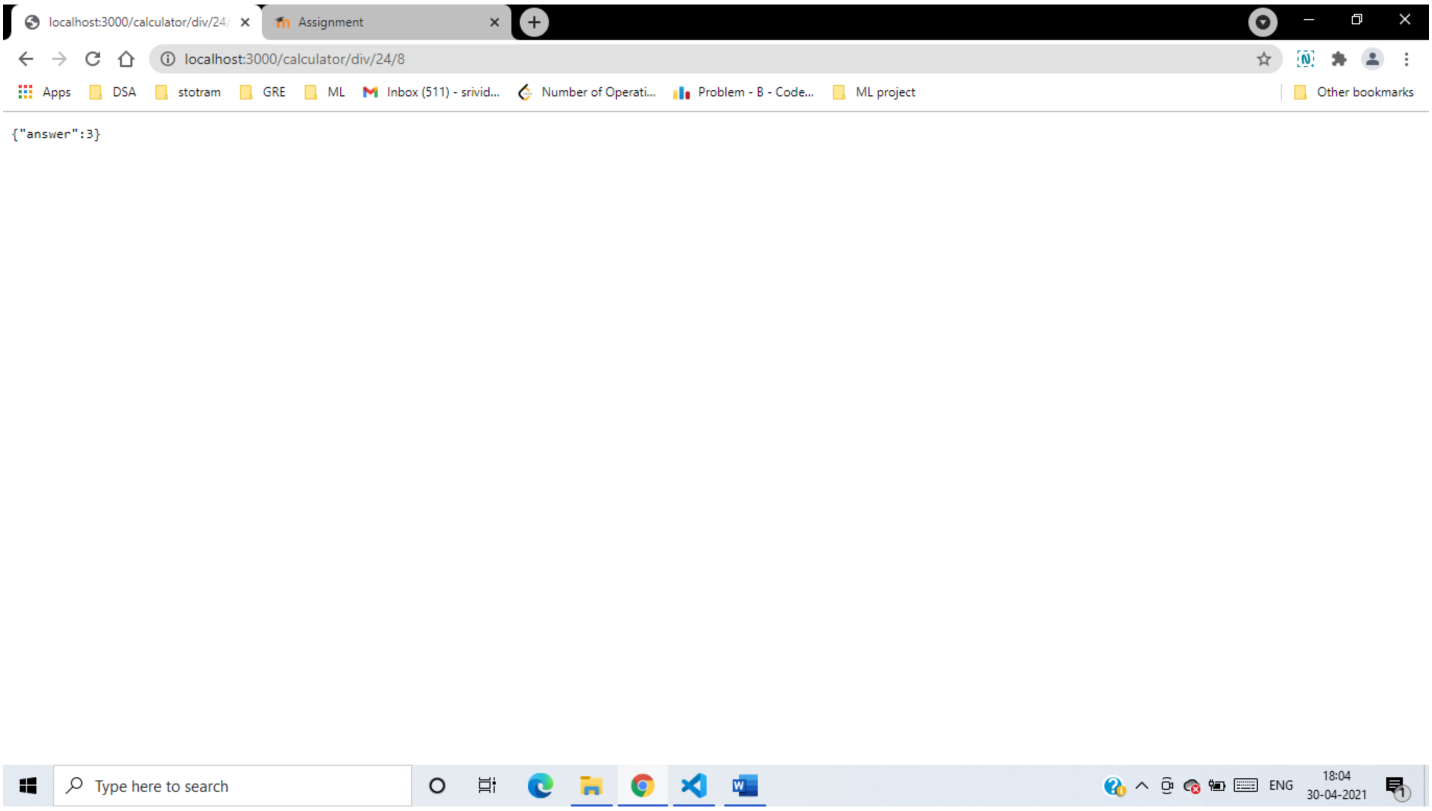


Calculator

Enter First Number: 24

Enter Second Number: 8

Result of Division.



```
{"answer":3}
```

Result:

The program for calculator application using Rest API has been successfully written and executed.