

MSWD PROJECT

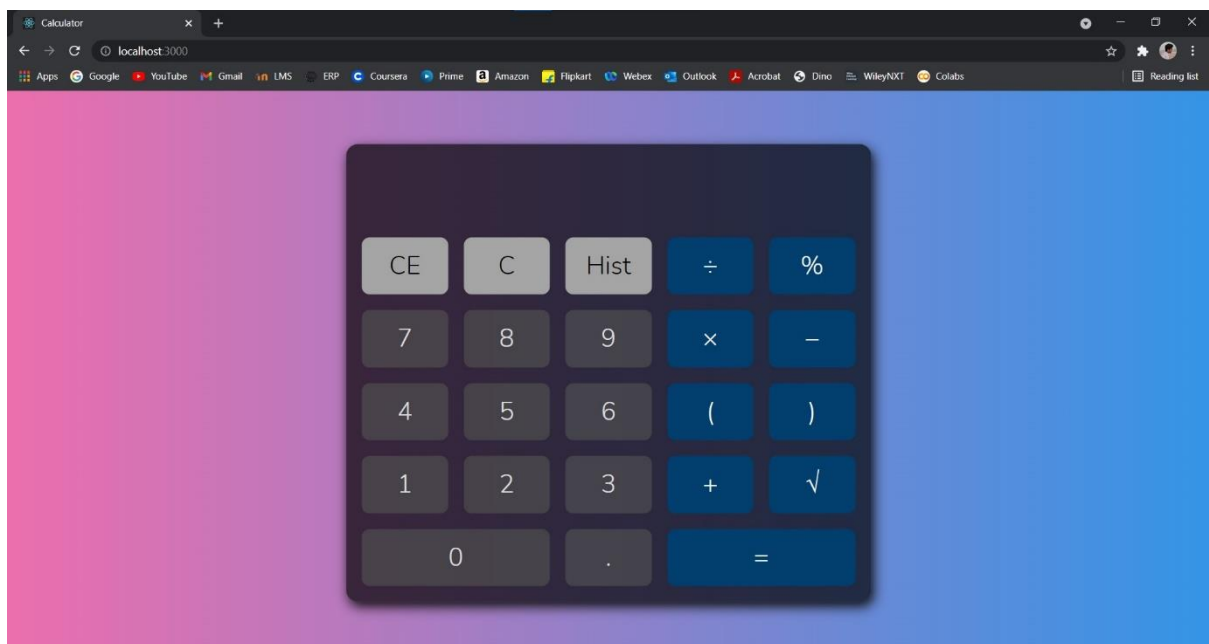
TEAM MEMBERS :

1. 190031328 - POLAVARAPU LAKSHMI HARI CHARAN
2. 190030844 - KONGARA JASWANTH

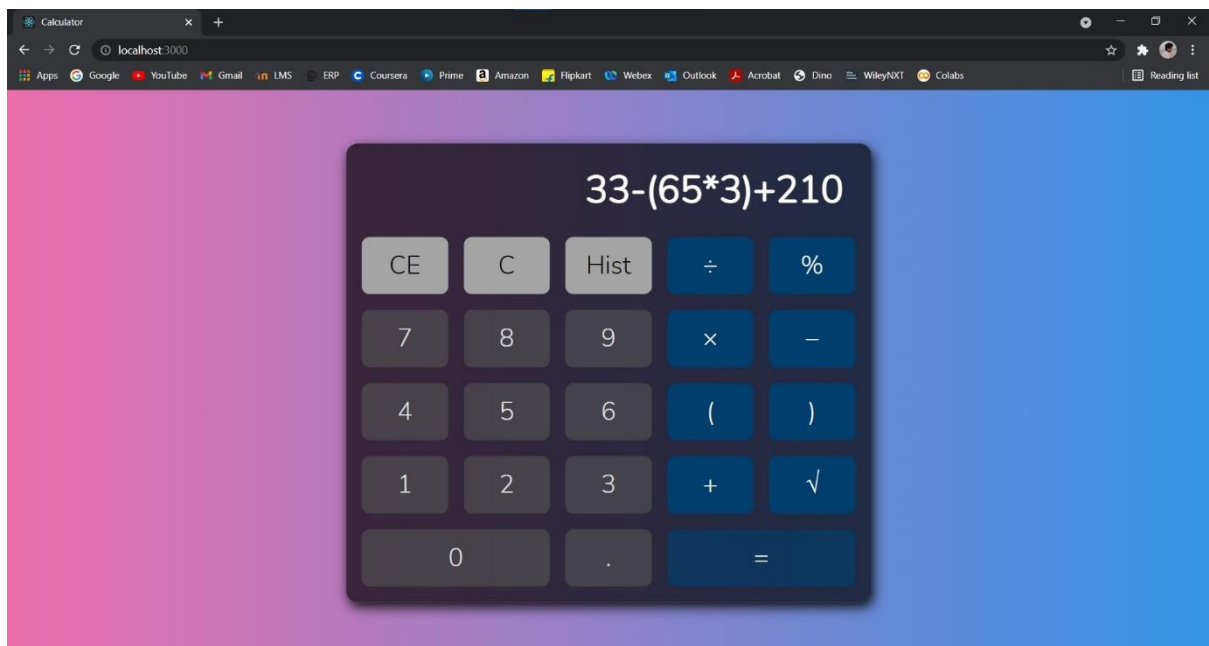
PROJECT TITLE : Calculator using React

OUTPUTS :

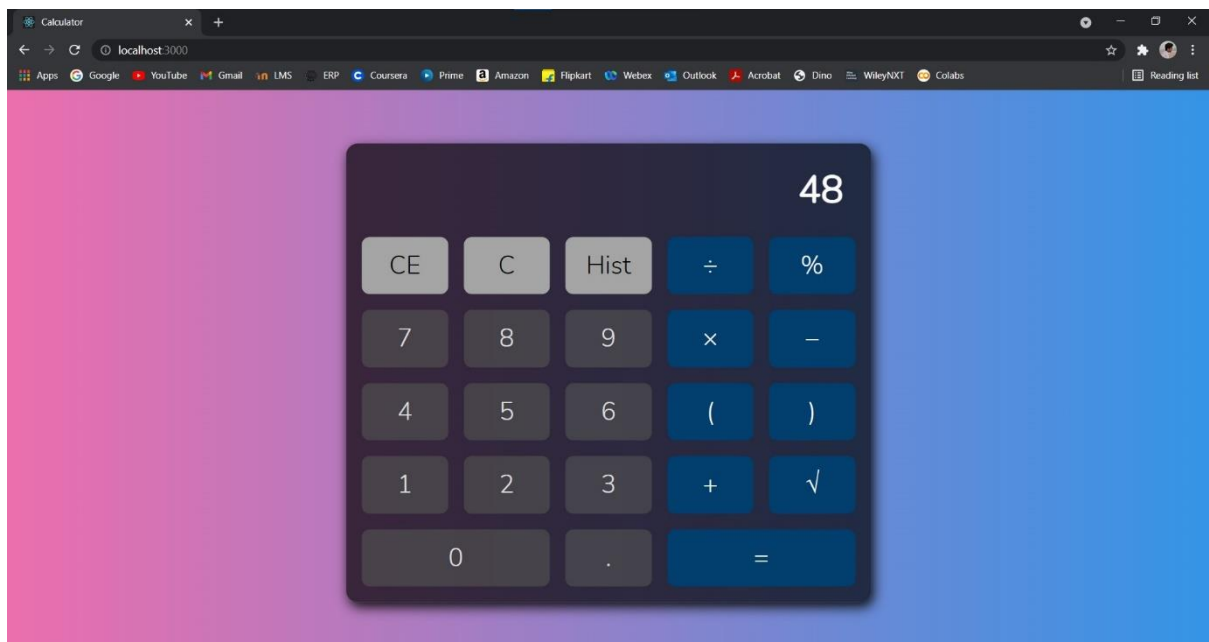
Calculator :



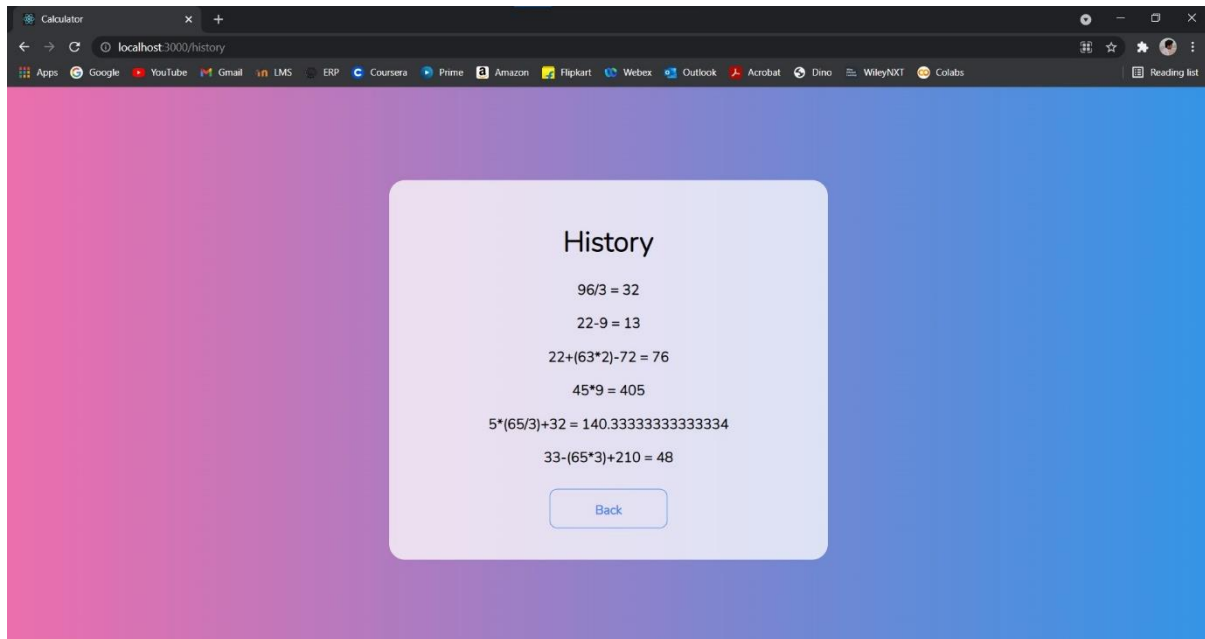
Input :



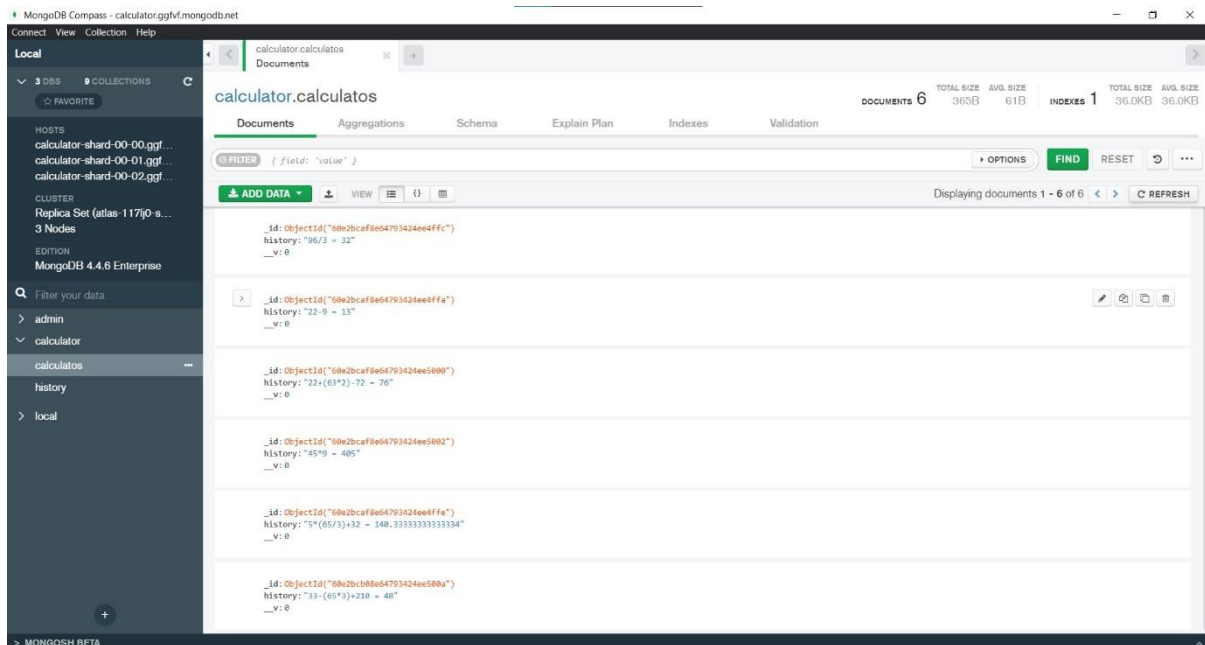
Output :



History :

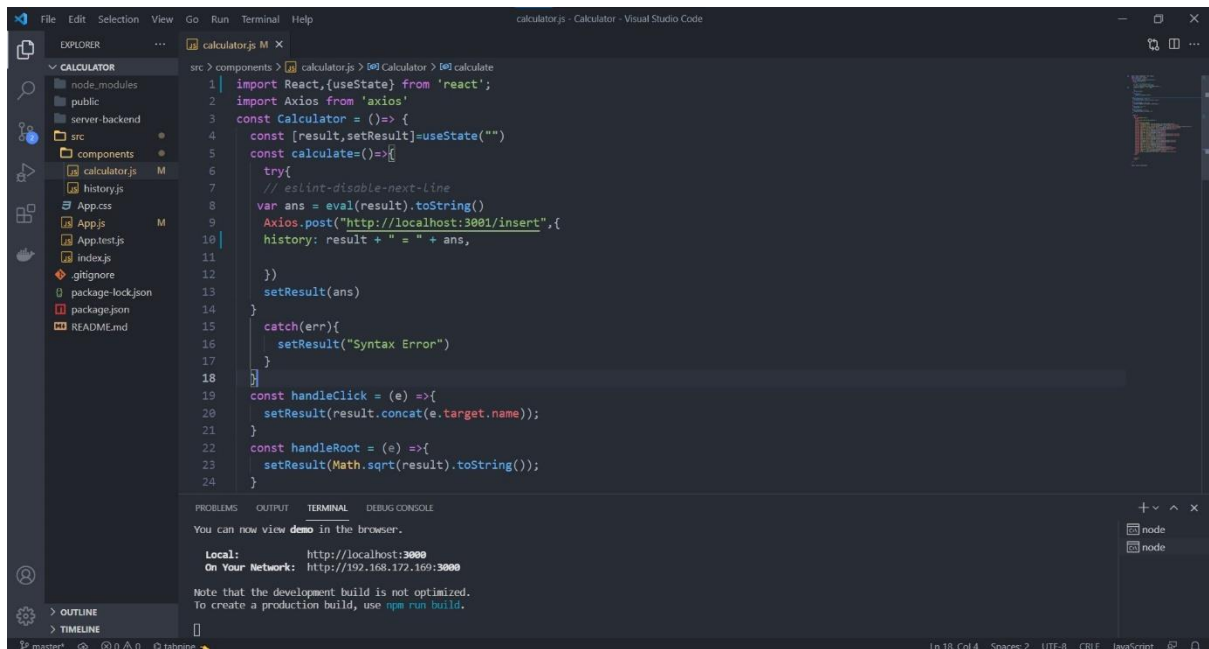


Backend Response (MongoDB):



Codes :

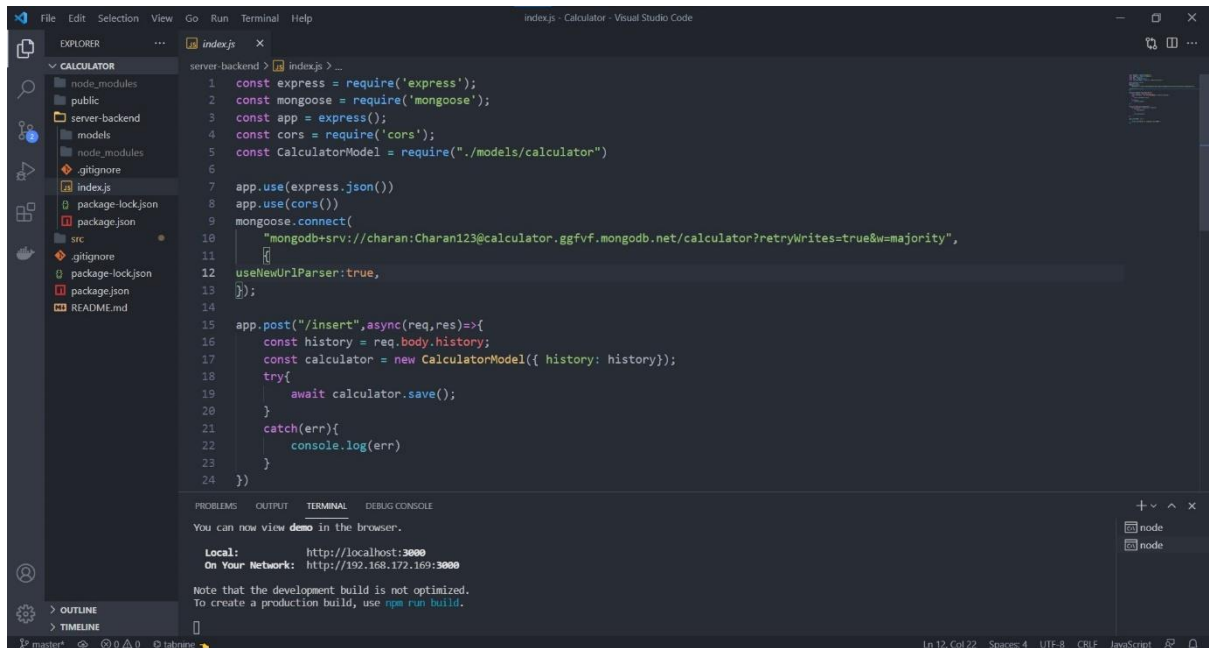
Frontend:



The screenshot shows the Visual Studio Code editor with the 'calculator.js' file open. The code is a React component that uses useState to manage the calculator's state. It includes a calculate function that evaluates the input and sends the result to a backend API via Axios. The terminal shows the local URL as http://localhost:3000.

```
1 import React, {useState} from 'react';
2 import Axios from 'axios'
3 const Calculator = () => {
4   const [result, setResult] = useState("")
5   const calculate = () => {
6     try {
7       // eslint-disable-next-line
8       var ans = eval(result).toString()
9       Axios.post("http://localhost:3000/insert", {
10         history: result + " = " + ans,
11       })
12     } catch (err) {
13       setResult("Syntax Error")
14     }
15   }
16   const handleClick = (e) => {
17     setResult(result.concat(e.target.name));
18   }
19   const handleRoot = (e) => {
20     setResult(Math.sqrt(result).toString());
21   }
22 }
```

Server Backend :



The screenshot shows the Visual Studio Code editor with the 'index.js' file open. The code is a Node.js server using Express and Mongoose. It connects to a MongoDB database and defines a POST endpoint for inserting calculator history. The terminal shows the local URL as http://localhost:3000.

```
1 const express = require('express');
2 const mongoose = require('mongoose');
3 const app = express();
4 const cors = require('cors');
5 const CalculatorModel = require("./models/calculator")
6
7 app.use(express.json())
8 app.use(cors())
9 mongoose.connect(
10   "mongodb+srv://charan123@calculator.ggfvf.mongodb.net/calculator?retryWrites=true&w=majority",
11   {
12     useNewUrlParser: true,
13   });
14
15 app.post("/insert", async (req, res) => {
16   const history = req.body.history;
17   const calculator = new CalculatorModel({ history: history });
18   try {
19     await calculator.save();
20   } catch (err) {
21     console.log(err)
22   }
23 })
```