



# Web-based Calculator

---

## Student Data:-

Mark Nader Fathy Beshara

ID : 18011305

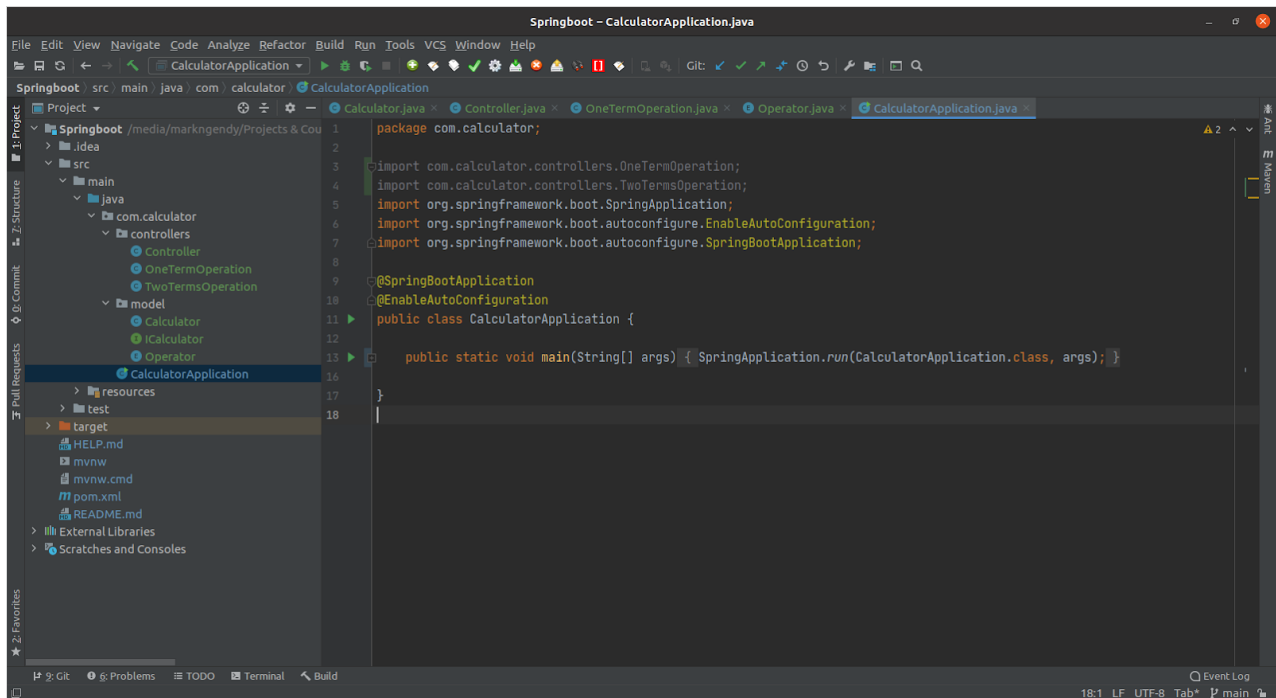
## Problem Statement

Build a web-based Calculator similar to that of windows.

- The buttons should be web buttons.
- No need for fancy styling
- Calculation should be done server-side.
- For simplicity, You can ignore the difference between the C and CE buttons.
- Repeating pressing the = button does not issue new calculations
- Handle exceptions such as dividing by 0, by displaying an E.

## How To Run The App

1. First you should open the backend project using intellij IDE.



- Then you should build then run the backend project and it should run at Port **localhost 8080** with Tomcat server.

The screenshot shows an IDE window titled "Springboot - CalculatorApplication.java". The code editor displays the following Java code:

```

@SpringBootApplication
@EnableAutoConfiguration
public class CalculatorApplication {

    public static void main(String[] args) { SpringApplication.run(CalculatorApplication.class, args); }
}

```

The Run console shows the output of the application, including the Spring Boot logo and the following log messages:

```

Spring Boot :: (v2.3.4.RELEASE)

-11-28 21:01:43.460 INFO 241040 --- [main] com.calculator.CalculatorApplication : Starting CalculatorApplication on markngendy-Inspiron-5570 with PID 241040
-11-28 21:01:43.465 INFO 241040 --- [main] com.calculator.CalculatorApplication : No active profile set, falling back to default profiles: default
-11-28 21:01:44.792 INFO 241040 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
-11-28 21:01:44.806 INFO 241040 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
-11-28 21:01:44.807 INFO 241040 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.38]
-11-28 21:01:44.893 INFO 241040 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
-11-28 21:01:44.893 INFO 241040 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1288 ms
-11-28 21:01:45.162 INFO 241040 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
-11-28 21:01:45.554 INFO 241040 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
-11-28 21:01:45.688 INFO 241040 --- [main] com.calculator.CalculatorApplication : Started CalculatorApplication in 2.695 seconds (JVM running for 3.1s)

```

- Now you should open the frontend project using a suitable IDE (Visual Studio Code)

The screenshot shows the Visual Studio Code editor with the "Calculator.vue" file open. The code editor displays the following Vue.js code:

```

<script>
export default {
  data() {
    return {
      firstOp: "",
      secondOp: "",
      operator: "",
      number: ""
    };
  },
  methods: {
    clear() {
      this.firstOp = "";
      this.secondOp = "";
      this.operator = "";
      this.number = "";
    },
    validateMinus() {
      if (this.number === "" || this.operator === "-") {
        this.number = "";
      } else {
        this.appendOp("SUBTRACT");
      }
    },
    async appendOp(operation) {
      if (this.firstOp === "") {
        this.firstOp = this.number;
        this.number = "";
      } else if (this.secondOp === "" && this.operator === "") {
        this.secondOp = this.number;
        this.number = "";
      } else if (this.operator !== "") {
        await this.equal();
        this.number = "";
      }
      this.operator = operation;
    }
  }
};

```

#### 4. Run the commands in the terminal of the README.md file

```

1 # vue
2
3 ## Project setup
4 ...
5 yarn install
6 ...
7
8 ## Compiles and hot-reloads for development
9 ...
10 yarn serve
11 ...
12
13 ## Compiles and minifies for production
14 ...
15 yarn build
16 ...
17
18 ## Run your unit tests
19 ...
20 yarn test:unit
21 ...
22
23 ## Lints and fixes files
24 ...
25 yarn lint
26 ...
27
28 ## Customize configuration
29 ...

```

Terminal output:

```

1: bash
markngendy@markngendy-Inspiron-5570:~/Downloads/Web-based-Calculator-master/vue$

```

#### 5. To run the Vue project type (yarn serve) command in terminal.

#### 6. The frontend project will run at port **localhost 8081**.

```

143 if (
144   !(this.firstOp === "") &&
145   this.secondOp === "" &&
146   this.operator === ""
147 ) {
148   this.clear();
149   this.number = number;
150 } else {
151   this.number = `${this.number}${number}`;
152 }

```

Terminal output:

```

2: npm
markngendy@markngendy-Inspiron-5570:/media/markngendy/Projects & Courses/College Labs/OOP Labs/Calculator/vue$ npm run serve
> vue@0.1.0 serve /media/markngendy/Projects & Courses/College Labs/OOP Labs/Calculator/vue
> vue-cli-service serve

INFO Starting development server...
98% after emitting CopyPlugin

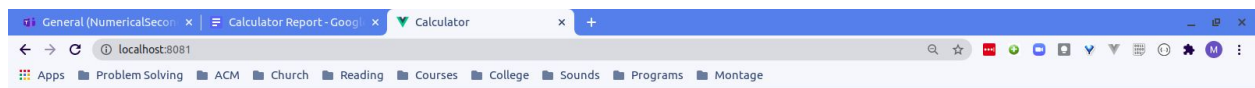
DONE Compiled successfully in 1640ms

App running at:
- Local: http://localhost:8081/
- Network: http://192.168.1.12:8081/

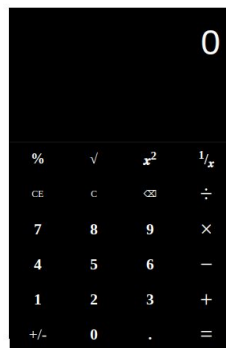
Note that the development build is not optimized.
To create a production build, run yarn build.

```

7. Open the browser on Url <http://localhost:8081/>



Online Calculator



## Sample Runs

First Sample run we will add 5 & 8

1- Click to write the first operand (5)



Online Calculator



2- Click the plus button (+)



## Online Calculator



3- Click to write the second operand (8)



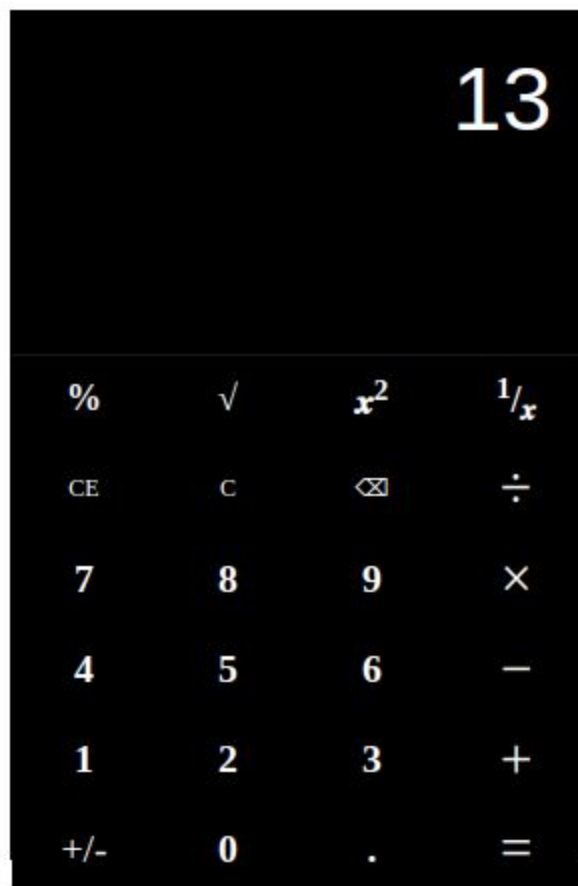
## Online Calculator



4- Click equal to get the result displayed (13)



## Online Calculator



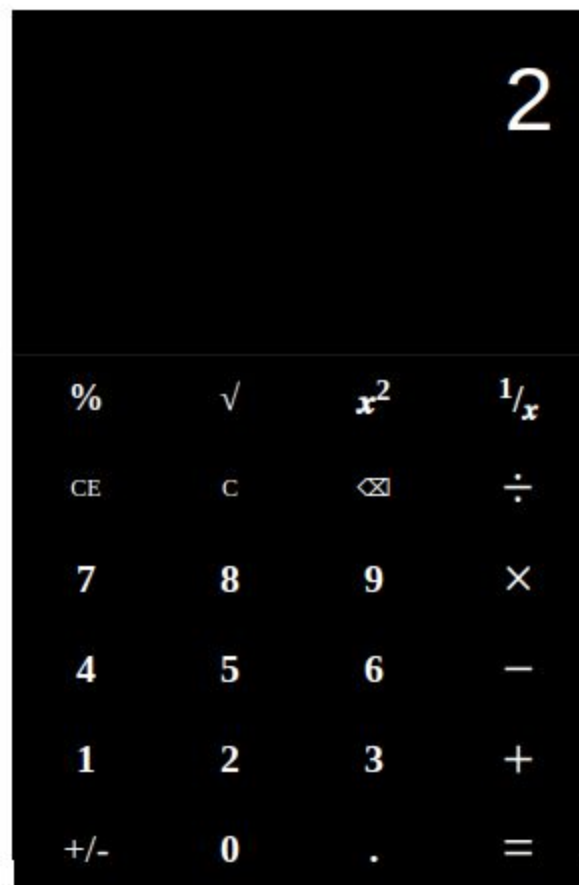


Second Sample run we will calculate  $(2 + -25 + 8)$

1- Enter number 2



## Online Calculator



2- Press Add +



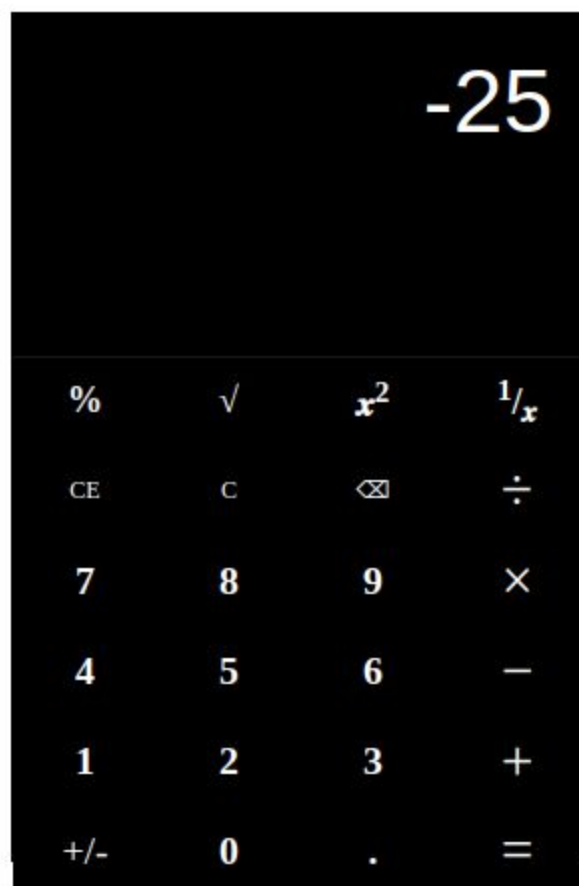
## Online Calculator



3- Enter number -25



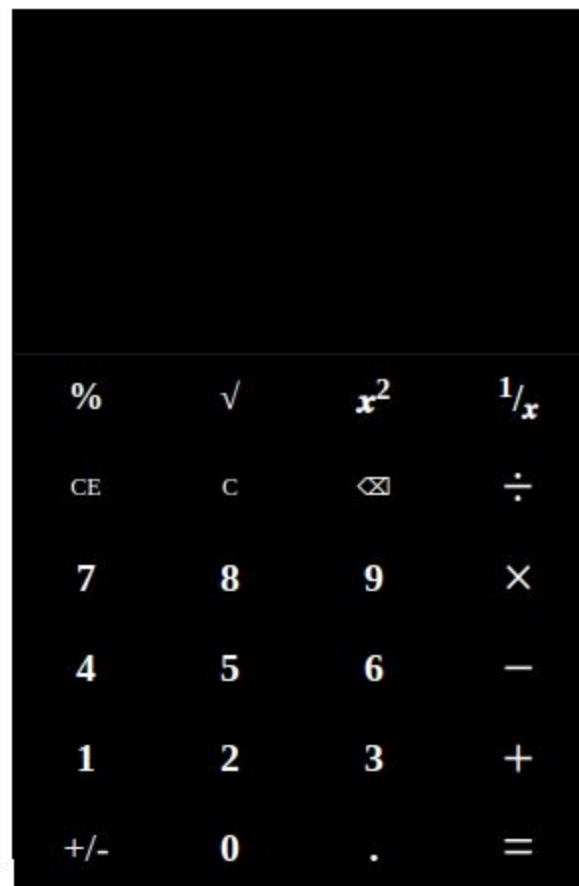
## Online Calculator



4- Press Add +



## Online Calculator



5- Enter number 8



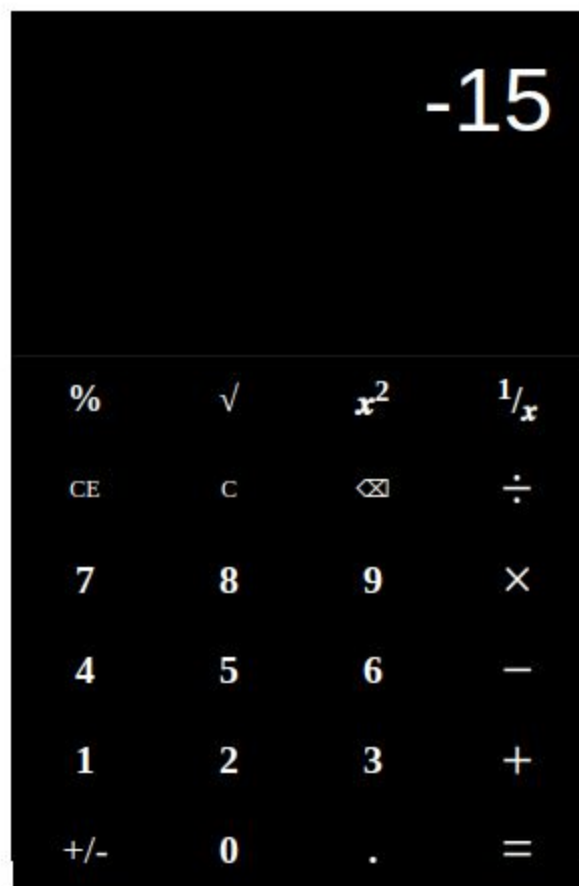
## Online Calculator



6- Press equal to get the result



## Online Calculator



## Features

### I. Multiple Operations

This calculator application performs multiple operations like (addition, squaring, division, calculating inverse, calculating absolute value of a number, ....)

### II. Accumulative Operations

You can perform sequential operations one after another without clearing the result

### III. Error & Difference Validations

When you divide with irregular operation (divide by zero), An error message will be displayed on the screen.

The calculator can distinguish the difference sign for performing operation and the difference sign for negative number so it deals with negative numbers without errors.