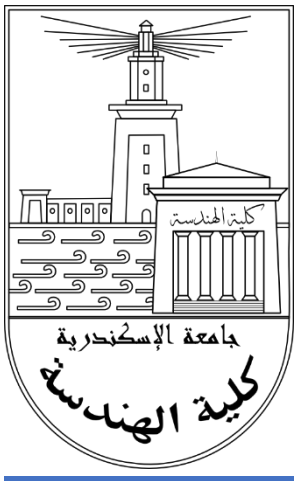


Programming ||: Lab 2 report

Name: Adel Mahmoud Mohamed

ID: 20010769

➢ Design layout:

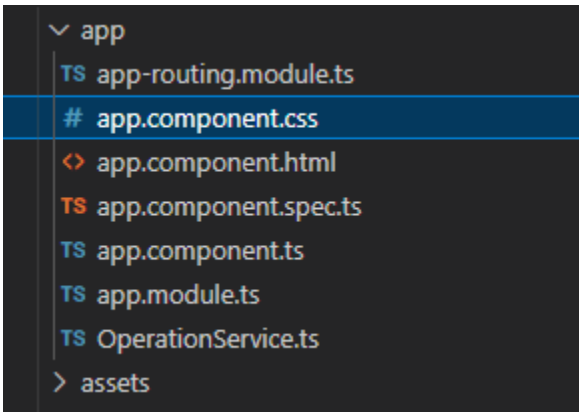
- The calculator has two screens the upper one is where the operation is displayed, and the lower one is where the current operand and the final result are displayed.
- You can use the keyboard to insert simple values like **(=, +, -, x, dot, delete, backspace, and the digits from 1 up to 9)**
- The calculations are made on the server side.
- It is so much like the Windows standard calculator.

➢ Design steps:

- The application is built using **MVC** paradigm.
- Angular framework is used the buildup the **View**, where the HTML, CSS, and typescript components are written.
- Spring Boot framework is used to build up the **Controller** where the requests to the server and the responses are handled, and the **Model** which makes the calculations and send the results to the controller which send it to the view to be rendered.

➢ Angular main components(file name is: calculator-app):

- **App.component.html:** this is where the html template is written.
- **App.component.css:** this is where the CSS is written.
- **App.component.ts:** this is where the user's actions are handled.
- **OperationService.ts** this is the link between the view and the controller and from which the URL of http request is sent to the server.



➢ Spring boot main classes(file name: calculator_backend):

- **Expression_handler class:** this is where the calculations are done for both the unary operators by the function

```
public String  handles_unary(String op, String operand)
```

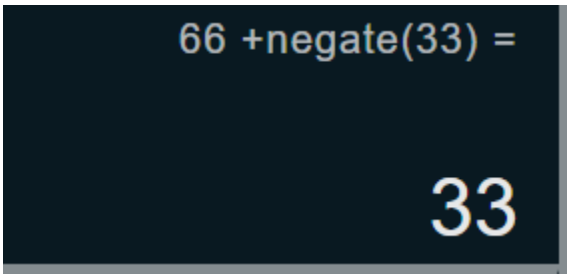
and the binary operators by the function:

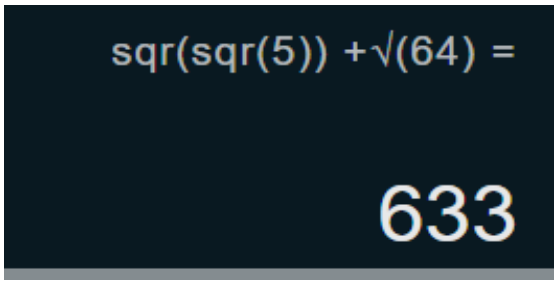
```
public String evaluate(String first_op,String operator,String second_op)
```

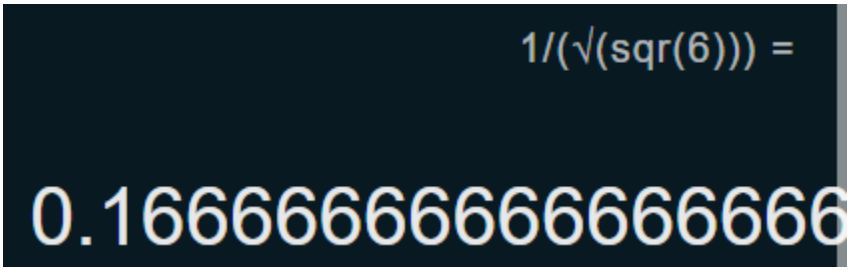
then the result is sent to the controller class to deal with it.

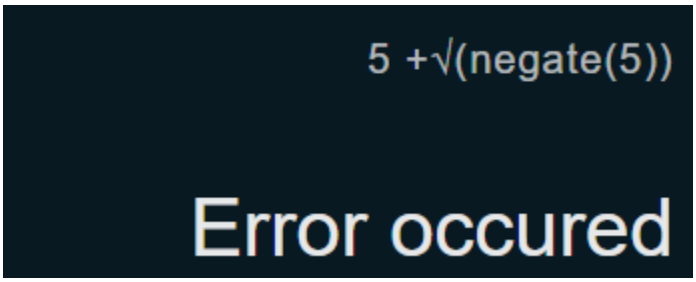
- **Controller class:** handles the request and responses and send the calculation results to the front-end to view it

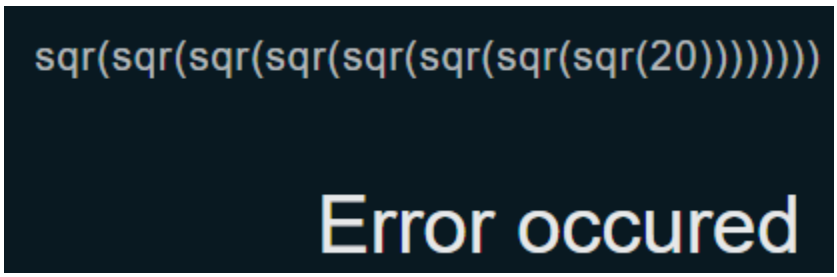
➢ Sample runs:

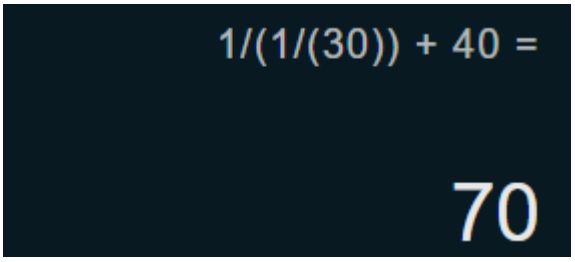


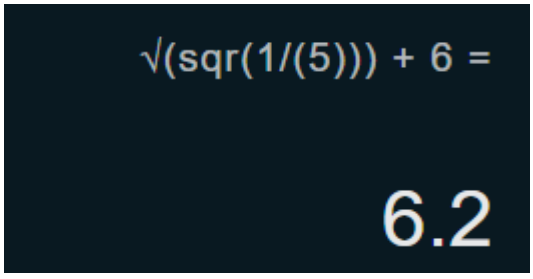












➢ Assumptions:

- The “%” operator operates just like the windows calculator so when the expression has one operand it gives zero, else it gives (second operand / 100) * first operand.
- CE and C have the same effect as mentioned in the lab requirements that it’s okay if they perform the same thing.
- The message “Error Occurred” is displayed whenever an error occurs like:
  - ✓ Division by zero.
  - ✓ Square root to a non-positive operand.
  - ✓ Overflow: that’s when the result of an operation is too large for a numeric data type.
- After the “=” is pressed then the new result can be used again for a second operation.
- If the user entered binary operator and he wants to change it, then he can and the last one is taken in the expression.
- You can perform successive operations at a time like 3 + 5 + 7 + … without having to click “=” every time.

Kindly Note:

I have deleted the folder node_modules from the angular folder so that the folder size is decreased so please install the modules when you open the front-end folder in order to perform ng serve.