Calculator Application

IIHT

Time To Complete: 10 to 12 hr

CONTENTS

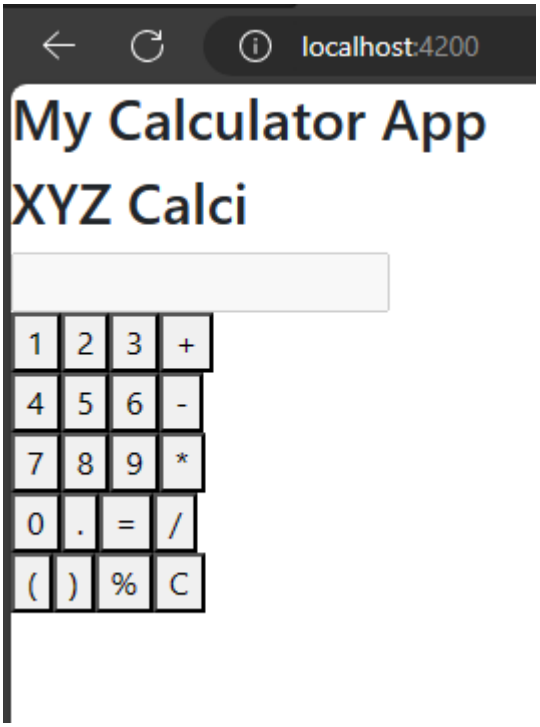
1 Problem Statement

Calculator application is SPA (Single Page Application) allows users to have a conventional calculator online. Users are able to do all basic calculating operations here..

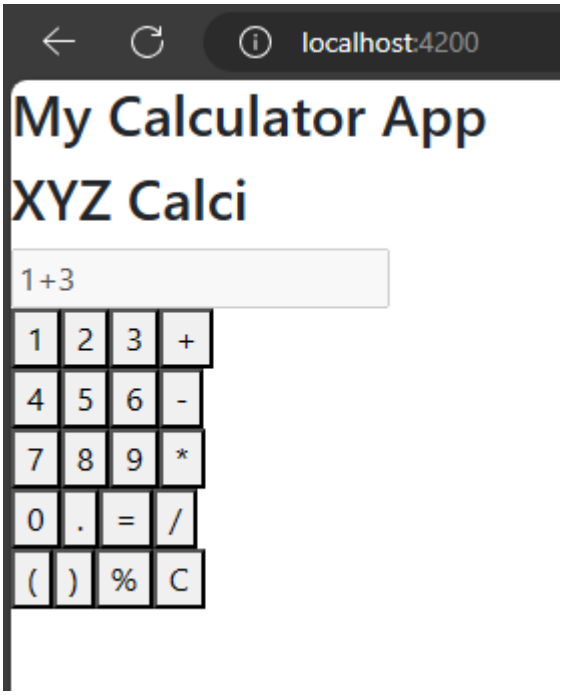
2 Proposed Calculator Application Wireframe

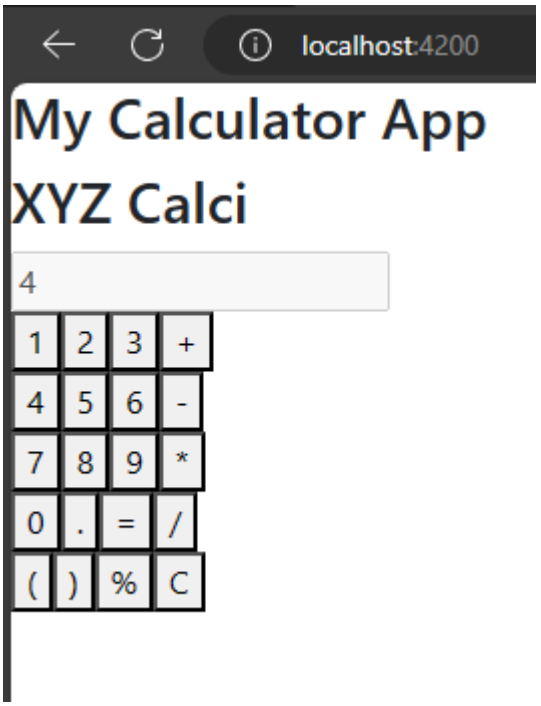
UI needs improvisation and modification as per given use case and to make test cases passed.

2.1 Welcome page



2.2 ScreenShots





3 BUSINESS-REQUIREMENT:

As an application developer, develop the Calculator Application (Single Page App) with below guidelines:

User Story #	User Story Name	User Story
US_01	Welcome Page	As a user I should be able to visit the welcome page as default page. Acceptance criteria: **App Overview: Angular Calculator App:** **Purpose** A simple Angular-based calculator that allows users to input two numbers and perform basic arithmetic operations. It's built using a component-driven architecture and uses `FormsModule` and `ReactiveFormsModule`. **HTML Structure (app.component.html)** 1. **Heading** ○ `<h2>` — Displays: **"My Calculator App"** 2. **Component Inclusion** ○ `<app-calculator>` — Embeds the `CalculatorComponent`, where all calculator functionality resides. **Angular Calculator Component Summary** **HTML Structure** 1. **Heading** ○ Displayed in an `<h2>` tag with the text: `XYZ Calci`. 2. **Result Display** ○ A disabled `<input type="text">` shows the current expression/result. ○ Dynamically bound to `result` using `[value]`. 3. **Buttons Section** ○ Buttons for digits `0–9`. ○ Operators: `+`, `-`, `*`, `/`, `%`, `.`, `(`, `)`. ○ Control buttons: ■ `=`: for calculation. ■ `C`: to clear input.

		○ Each button uses a `<button>` element with `(click)` bindings. 4. **Layout** ○ Buttons are grouped into rows using ` ` for visual separation. **Functions & Responsibilities** **`appendToExpression(value: string)`** ● Adds the clicked value (number or operator) to the expression string. ● Updates the result field to reflect the current expression. **`calculateResult()`** ● Evaluates the current expression using `eval()`. ● If the expression is valid, displays the result. ● If the expression is invalid (e.g., unmatched parentheses), shows the string `'Error'`. **`clearExpression()`** ● Resets both `expression` and `result` to empty strings. ● Effectively clears the calculator screen. **** Kindly refer to the screenshots for any clarifications. ****

4 CONSTRAINTS

2. You should be able to press the “TAB” key and “SHIFT + TAB” to navigate from top field to bottom field and vice-versa.
3. By default the label should be non-editable.

5 MANDATORY ASSESSMENT GUIDELINES

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to
   Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.
3. This editor Auto Saves the code.
4. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
5. This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.
   Note: The application will not run in the local browser
6. You can follow series of command to setup Angular environment once you are in your project-name folder:
   a. npm install -> Will install all dependencies -> takes 10 to 15 min
   b. npm run start -> To compile and deploy the project in browser. You can press <Ctrl> key while clicking on localhost:4200 to open project in browser -> takes 2 to 3 min
   c. npm run test -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min
7. Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.