

Powered by Crisp Analytics

JavaScript Documentation

### **Technical Design Document**

v1.1

**Document Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Published Date** | **Remarks** |
| 1.0 | Shaik Muhammad Irfan | Mar 23, 2023 | Initial Draft |

# 

# Table Of Contents

[Exercise 1](#_Toc1693455674) 4

[Exercise 2](#_Toc1693455674) 5

[Exercise 3](#_Toc1693455674)  6

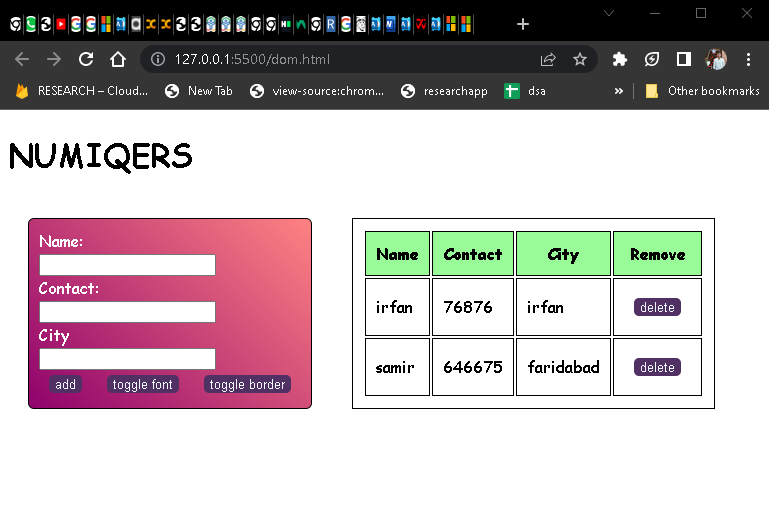
# Introduction

This document includes my assignment for javascript. I have attached the screenshots of the implementation in the next page while the link to the GitHub repo is given below.

Github repo link : https://github.com/Irfan622/javascript.git

# Exercise 1

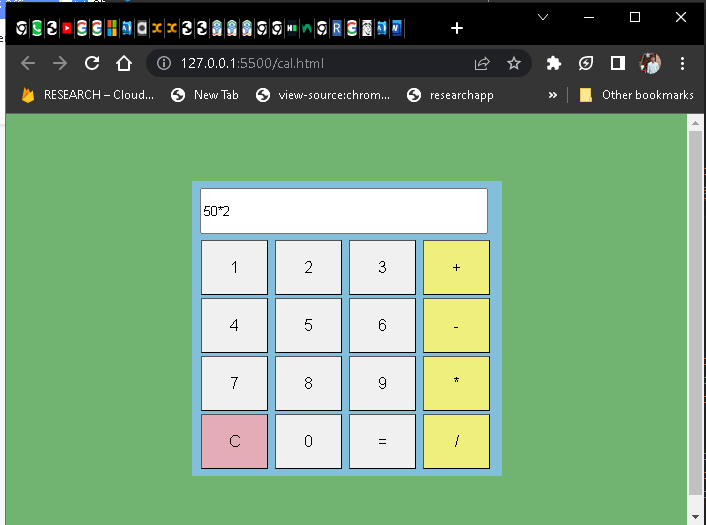
1. Dom manipulations on user clicks .

:

Actions performed by java script

* It adds an event listener to a button with the id "add", which prevents the default behavior of the button when clicked.
* It gets the values of three input fields with ids "form0", "form1", and "form2".
* It creates a new table row with the input values and a delete button.
* It appends the new table row to the table body.
* It adds an event listener to each delete button, which removes the closest table row when clicked.
* It adds an event listener to a button with the id "toggle", which toggles a class "secondFont" on the body element when clicked.
* It adds an event listener to a button with the id "tgl\_bdr", which toggles a class "bdr" on the table element when clicked.

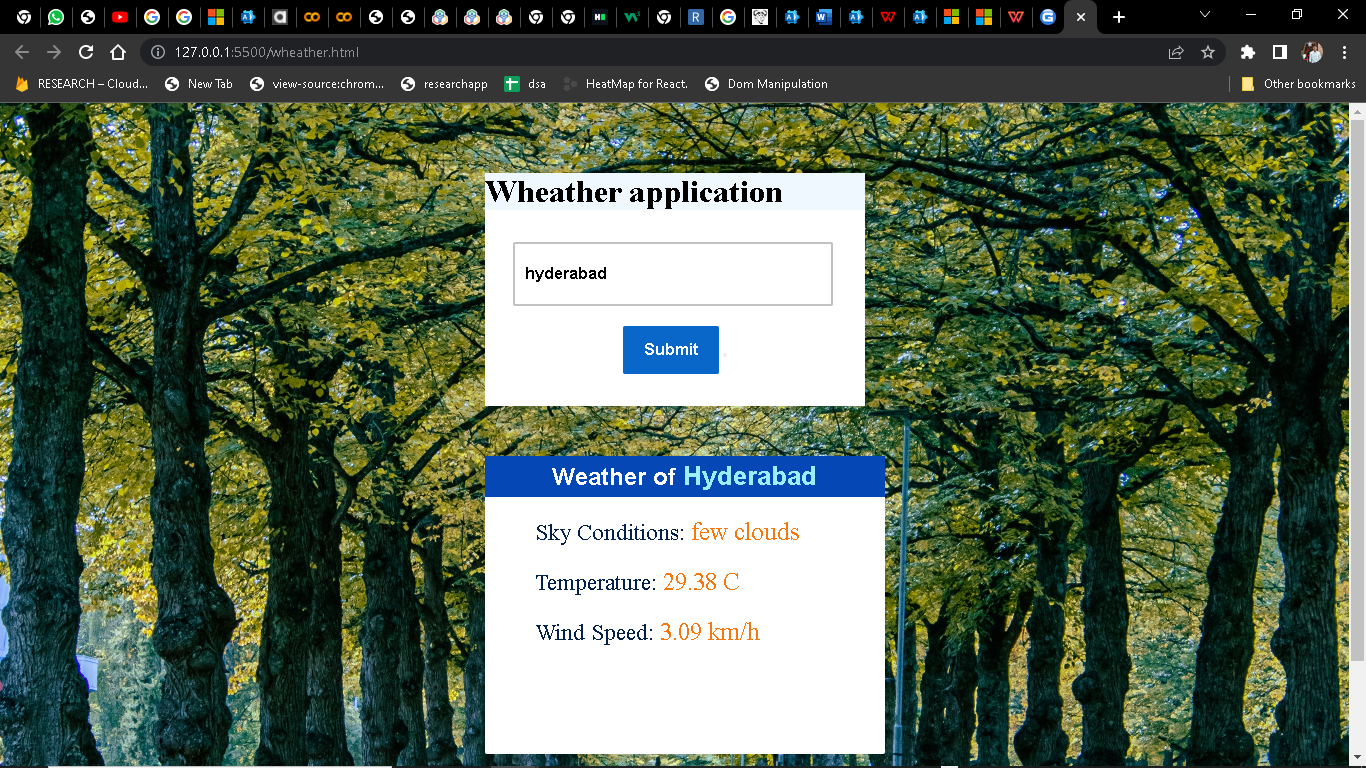
1. Calculator with custom ui.



Actions performed by java script

* It gets the element with the id "text", which is a text input element representing a calculator screen.
* It gets all the elements with the class "button" or "op", which are the calculator buttons.
* It initializes a variable "temp" to an empty string, which will store the input for the calculator.
* It loops through each button and adds an event listener for the "click" event.
* When a button is clicked, it gets the text content of the button and stores it in a variable "buttonText".
* If the button is the equals button "=", it evaluates the input stored in the "temp" variable and displays the result on the calculator screen.
* If the button is the clear button "C", it clears the calculator screen and resets the "temp" variable to an empty string.
* For all other buttons, it appends the text content of the button to the "temp" variable and displays it on the calculator screen.

1. Wheather application .



Actions performed by java script

* It gets the input element with the id "cityinput", which is the input field for the city name.
* It gets the button element with the id "add", which is the button to fetch the weather data.
* It gets the elements with the ids "cityoutput", "description", "temp" and "wind", which are the elements to display the weather data.
* It initializes a variable "apik" with the API key for the OpenWeatherMap API.
* It defines a function "conversion" that takes a value as input and returns the converted value from Kelvin to Celsius with two decimal points.
* It adds an event listener for the "click" event on the button element.
* When the button is clicked, it fetches the weather data from the OpenWeatherMap API using the city name entered in the input field and the API key.
* It parses the JSON data returned by the API and extracts the required weather information like the city name, weather description, temperature, and wind speed.
* It displays the extracted weather information on the HTML page by setting the text content of the corresponding HTML elements.
* If the API request fails or the entered city name is wrong, it displays an error message