UNIVERSITY OF DAR ES SALAAM

COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGIES

UDSM DHIS2 LAB DEVELOPER CHALLENGE

Title: **SearchIT**

Student name: **EDWARD, EDITH NAIKE**

Degree: **BSC IN COMPUTER SCIENCE**

Contact (email): edithnaike@gmail.com

Contact (phone number): **+255 753 040639**

INTRODUCTION

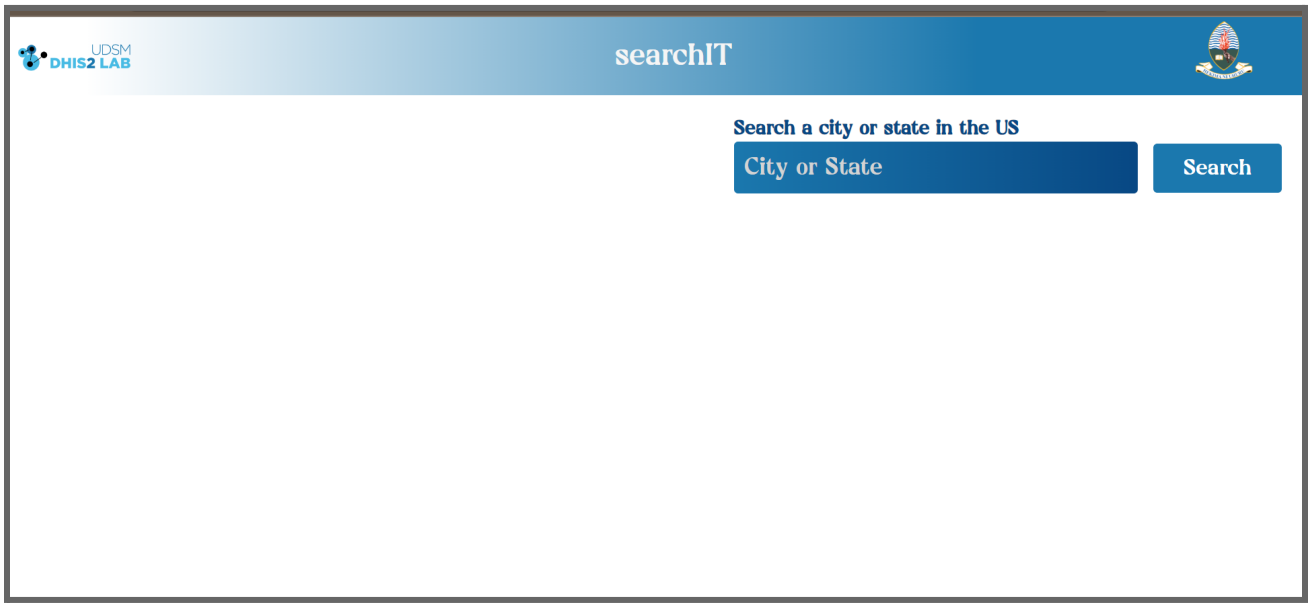
This is my response to the UDSM DHIS2 Lab Developer Challenge, named "SearchIT". This challenge started on the 8th of June 2023 with the deadline on the 30th of June 2023. I am pleased to present my “ability to build a functional web application without the use of any frameworks and later apply for Practical Training” offered by UDSM DHIS2.

I have used the given dataset of US cities and their details to create a simple search engine web application.

DESIGN

The current UDSM DHIS2 Lab website (https://dhis2.udsm.ac.tz/#/home) inspired the design approach. I adopted the header from the website’s navigation bar as well as the colour scheme and button.

Below is an image of the website for visualisation.



I strictly followed the UI design task guidelines.

With the aim of providing simplicity and usability by maintaining a minimal and straightforward layout.

Please click the attached link to the GitHub repository https://github.com/Edith-Naike-Edward/DHIS2challenge.git.

IMPLEMENTATION CHOICES

I implemented the code by using HTML5, CSS3 and Javascript to fetch data from https://tinyurl.com/5bzzvh6u.

The implementation choices I made when doing the Javascript are:

1. To fetch data from the provided URL using a fetch() function. It returns a promise that resolves to the parsed JSON response. If there is an error, it throws an error and responds “An error occurred”.
2. The retrieved data and the search query are passed as inputs to the filterCities(data, searchQuery) function. It filters the cities based on the search query by matching the lowercase versions of the city and state names with the query's lowercase version. It uses the Array.prototype.filter() function to create a new array containing only the cities that have the search query in the city or state name.
3. The displayResults(results) function takes as input the filtered cities and shows them on the website. It begins by clearing the searchResults element, which clears the existing search results. If no results are discovered, the message "No results found" is shown. Otherwise, it uses the Array.prototype.forEach() function to loop over the results. It generates an HTML element for each city and populates it with the city name, state name, population, rank, and growth. Based on the growth percentage, the growth value is stylized with a colour (green for positive growth, red for negative growth).
4. The searchCities() method, written in Vanilla Javascript, is an async function that searches for cities. I used it to obtain the search input value, trim it, and determine whether it is empty. If it is null, the method returns immediately. Otherwise, it employs await to obtain data from the URL through the fetchData() method. The filtered cities are then returned by using the filterCities() method with the obtained data and the search query. Finally, the displayResults() method is used to display the filtered cities on the page.

The implementation choices I made when doing the UI design were to use HTML to create the overall layout of the web application and CSS to define the visual appearance of the elements to be aesthetically appealing. And an external JavaScript file named "script.js" which is responsible for handling the search functionality and displaying the search results.

THE CHALLENGES I FACED

The challenges I faced when completing this challenge were to accurately filter and search for data using the appropriate functions.

HOW I OVERCAME THIS CHALLENGE

I overcame this challenge by informing myself on how to properly implement searching data and filtering data using Javascript from online sources.