



Programming Studio 1

COSC2803 | Semester 1 2023

Climate Change Awareness

Requirements Document

1 Overview

This document outlines the tasks for the “Climate Change Awareness” social challenge. You can think of this document representing the **product idea** (from a UX/UI perspective) and the **business case** (from the database perspective).

1.1 Challenge A: Climate Change Awareness

Develop a web-application to **help governments, scientists and the general public analyse patterns of change in ocean and land temperatures over a 260-year period**. You will need to present different statistics and calculated information about observed temperatures from across various countries using data in an informative manner. You will need to cater for diverse users who are interested in high-level concepts or an in-depth analysis of the data. A core focus should be on identifying trends across various time scales and geographic locations.

1.2 Notes on the provided Datasets

You have been provided datasets that relate to historic (yearly) temperature observations for different geographic locations. In addition, you are provided a dataset of historic (yearly) population statistics.

The temperature dataset is from **1750 – 2013** at a **worldwide, country, state** and **city** level granularity, while the population data is from **1960 – 2013** at a **worldwide** and **country** level granularity.

Note that early data for some countries is unavailable – for example Australia has no data prior to 1852.

The original data was based on daily average temperature observations. This has been greatly reduced by aggregating down to a yearly frequency. This was done by averaging the daily average temperature observations, in addition the minimum and maximum of the daily average temperature for each year was calculated.

More details about each dataset can be found in the ‘description.txt’ file.

1.3 Note about temperature and population data variance

The temperature dataset and population datasets are from different sources, as such there is no population data for dates prior to 1960. Queries that require both population and temperature data will need to deal with this in a reasonable manner.

2 Project Level Details

The tasks for each level are outlined in this section.

2.1 Level 1 (GREEN) “Big Picture” Content

Level 1 sub-tasks capture the attention of the users and show how to use your website. It is up to you to decide the best way to:

- Show the users the required information, that is, complete the UX/UI design.
- Retrieve information from your database, where required. That is, you cannot “hard-code” values that must be computed from the database.
- You should *carefully consider* how to best complete your UX/UI and ER designs.

Sub-Task A: Landing Page

This is the first page users see when they visit your website. This should:

- Capture the attention of all users of your website.
- Present the year range (first and last year) for available data (population and temperature) and the corresponding world population and temperature during these years.
- Identify the total number of years of available data for both global population and temperature.
- **Both must be dynamically calculated from data stored in, and retrieved from, your database.**

Sub-Task B: Mission Statement

This presents the overall purpose of your website. This should:

- Present your perspective on how your website addresses the social challenge.
- Describe how your site can be used.
- Present the Personas that your website targets. **These must be stored in and retrieved from your database.**
- List the names and student numbers of all team members. **These must be stored in and retrieved from your database.**

2.2 Level 2 (ORANGE) “Shallow Glance” of the data

Level 2 sub-tasks explore, at a shallow level, the issue of climate change and population at various geographic levels. You should *carefully consider* how to complete your UX/UI designs and the ER model to store this business case. The sub-tasks include both UX/UI design and ERM design challenges.

Sub-Task A: Focused view of temperature and population change by Country/Global

Present information on population changes between two user selected years for at either the granularity of countries, or for the world. You should:

- Enable a user to choose to view by Countries or the world.
- If displaying by countries, show each countries name.

- If displaying by the world region, identify its name as 'World'.
- Enable the user to choose a start year and end year.
- Display the change in temperature and population between the start year and end year (where available). This must be computed from the database using SQL queries. If data is not available for some years, your website should handle this in a suitable manner.
- Display temperature and population changes between the start year and end year using either:
 - Raw values, for example the population of the country and the average temperature; or
 - Proportional values, for example, the *percentage* change in the country population and *percentage* change in average temperature.
 - (Where possible) Calculate 'The correlation' between temperature change and population change, using the proportional values. You will need to determine how to effectively compute 'The correlation'.
- When displaying regions, enable a user to sort by temperature or population:
 - The user may choose to sort on either criterion.
 - The user may choose to sort the criteria in ascending or descending order.
 - This sorting must be conducted using SQL queries. That is, we require you to show knowledge of SQL sorting methods and cannot sort using Java or JavaScript.

Sub-Task B: Focused view of temperature change by City/State

Present focused information on the temperature change between *two* user selected years for the *States* or *Cities* of a country that the user has also selected. You should:

- Enable a user to choose a Country.
- Enable the user to choose a start year and end year.
- Enable the user to choose to see temperatures for either *Cities* or *States*.
- Display the change in temperature between the start year and end year (where available) or all cities or states (depending on the user's choice) for the country the user selected. This must be computed from the database using SQL queries.
- Display temperature changes between the start and end year using either:
 - Raw values, for example the *average/minimum/maximum temperature*; or
 - Proportional values, for example, the *percentage change* in the average/minimum/maximum temperature.
- For the Country:
 - Compute the Ranking of the cities or states (1st, 2nd, 3rd, etc.) from **largest to smallest change by proportion** for each statistic (that is, average, minimum, or maximum temperatures). For example, the city or state with the largest change by proportion for average temperature is given rank 1.
 - Display the three rankings (average/minimum/maximum).
 - You may compute the ranking, either in SQL or through Java.

2.3 Level 3 (RED) "Deep-dive" of the data

Level 3 sub-tasks explore a deeper view of the issue of climate change and population at various geographic levels. You should *carefully consider* how to complete your UX/UI designs and the ER model necessary to store this business case. These sub-tasks include some quite complex UX/UI design and ERM design challenges. For top-marks you will need to take a *very well-considered approach*.

Sub-Task A: Identify changes in temperature over extended periods

Identify the average temperature over selected period of time beginning at a selected year. You should:

- Enable the user to select a starting year, a time period (in years), and a geographic region (global, country, state, or city).
- Calculate the average temperature across the selected period, for the given region beginning from the starting year.
- Enable the user to select multiple starting years and show the difference in averages for the same region and time period (you will need to think carefully about how to do this effectively when the user selects 3 or more starting years).
- Enable the user to compare the average, and difference in averages, for multiple regions - country, state or city only (you will need to think carefully about how to do this effectively when the user selects 3 or more starting years).
- Enable the user to sort all regions by the difference in averages (you will need to think carefully about how to do this effectively when the user selects 3 or more starting years)
- Enable the user to filter country results, to only show results for regions (countries) that fall within:
 - A selected range of population.
 - A select average temperature change
- Retrieve the *exact* information from your database using as few queries as possible:
 - Ideally, use only a single query (with column selections and JOIN operations)
 - Unnecessary information should not be returned in the SQL queries.
 - Performing the sorting through SQL queries if possible.
 - Note that this is challenge in making suitable SQL queries.

Sub-Task B: Finding time periods with similar temperature and/or population

Enable a user to find time periods that have similar temperature and population. You should:

- Enable the user to select a starting year, a time period (in years), and a geographic region (country, state, or city), then find other time periods that are “the most similar” to the chosen combination on a set of user selected properties, such that users may:
 - Select similarity in terms of temperature alone
 - Select similarity in terms of population alone (for country only)
 - Select similarity in terms of both temperature and population (for country only)
 - The user can then select to determine similarity in terms of:
 - the absolute values
 - the relative change in values
- From the user selection, your system must find the <x> most similar results, identifying the region, the start year, end year, the values used to determine its similarity to the selected region, as well as the similarity score, using the user's selection of constraints, where the user chooses the number of regions <x> to locate:
 - For example, the user selects the starting year 1900, a time period of 10 years, and the region ‘Australia’. The user selects similarity by temperature alone based on the absolute value, and to find the 5 most similar regions. Since the user has selected a country (Australia), the system should then find countries (including Australia) and time

periods (of 10 years) that have similar start and end temperatures, then display the 5 most similar results.

- Present the selected region and most similar regions, along with the relevant data that was used to determine the similarity.
- Sort the regions that are found by the 'most similar' to the 'least similar'.
- Retrieve the *exact* information from your database using as few queries as possible:
 - Ideally, use only a single query (with column selections and JOIN operations)
 - Unnecessary information should not be returned in the SQL queries.
 - Performing the sorting through SQL queries if possible.
 - Note that this is challenge in making suitable SQL queries.