

# Individual Assignment:

## Build an Informative Website

---

The student will build a website aimed at informing its visitors about an aspect of either climate change or surveillance. The website will present a combination of qualitative and quantitative information, supported by publicly available datasets, through textual, numerical, and visual communication. The student will create their own relational database from a couple of selected datasets. They will clean and analyse the data, and present visualisations on the website supporting its purpose. Finally, the student will integrate a user-tracking functionality into their website and reflect on its use and implications.

*Note: Collaboration is encouraged; however, every student must create their own website using their own database and present it in their own way.*

### Deadlines

Friday, March 14, 2025 (5 p.m.)	Progress Check – Part A
Thursday, April 17, 2025 (5 p.m.)	Progress Check – Part B
Monday, May 12, 2025 (2 p.m. - 4 p.m.)	Walk-in Feedback Session – Part C
Friday, May 16, 2025 (5 p.m.)	Final Submission

The Final Submission deadline is for the final submission of the complete assignment, parts A, B, and C together. It can be the improved versions of your Progress Check submissions. With this final submission, you will receive a grade for your complete DE assignment.

### Tasks

The assignment is composed of three major steps:

#### Part A: Developing the website structure

- Determine the topic and purpose of your website within the scope of available datasets
- Create an HTML/CSS structure for a number of web pages that are designed for the purpose
- Turn your website into a web application using Python Flask

#### Part B: Creating a relational database and presenting the information on your website

- Select multiple datasets\* that can convey the intended information
- Clean and prepare the data using Python and/or SQL for presentation on the website
- Design your relational database and draw its entity-relationship diagram
- Build the relational database according to your design using SQL
- Present and visualise the data stored in your relational database on your website, either as static content or dynamically through a backend connection

*\*Datasets will be selected from the resource list provided to the students or the student can use other appropriate and open-access datasets of their choice. In the latter case, datasets must be approved by a Core Lecturer.*

### Part C: Integrating user-tracking functionality and reflecting on its implications

- Determine the metrics that you want to track in your website users (e.g. time spent on a page, bounce rate, click rate)
- Add user-tracking functionality to your website via one of the two options:
  - integrating an off-the-shelf web analytics tool
  - writing Python code to create your own trackers
- Test your tracker with your peers
- Write a reflection report:
  - explaining your reasoning for the metric selection
  - sharing the insights you gained from the test run with your peers
  - discussing the potential benefits and harms of user-tracking
  - discussing the implications of
    - i. collecting user data considering your position
    - ii. using an off-the-shelf tool from a third-party

## Final Deliverables

A single .zip file containing these items:

- (1) Brief description of the website's purpose (1-3 sentences)
- (2) All the code and data files that are used and generated (i.e. while building the website, creating the database, cleaning and visualising data, and generating the user analytics)
- (3) Entity-relationship diagram for the relational database
- (4) Reflection report on user-tracking (word count: 500 ± 10%)

## Assessment Criteria

- Demonstration of proficiency in web development with HTML/CSS and Python:
  - Functionality, efficiency, and reproducibility of the web application
  - Ability to search, explore, and utilise appropriate functions/properties in addition to the effective use of class information
- Implementation of appropriate strategies to improve data quality:
  - Proper cleaning and preparation of the datasets for their use
  - Correct and effective use of Python and/or SQL
- Application of fundamental concepts for the design and implementation of the relational database:
  - Efficient and logical organisation of the entity-relationship diagram
  - Correct translation of the intended design into a relational database using SQL
- Effective communication of the intended message:
  - Effectiveness of the variables, data, and data presentation forms used in the website to deliver the intended information
  - Clarity of the information and the website for its users
- Comprehension of the use and implications of user-tracking applications
  - Correct implementation of user tracking functionality for generating the intended metrics
  - Critical and thorough reflections on the tracking experience
- Proper referencing of the resources used for the website

## Learning Objectives

- Explain the relevance of the components in the web development stack

- Explain the difference between client-side and server-side applications
- Apply basic web development principles to devise a web-based prototype
- Apply fundamental concepts for the design and implementation of databases, including tasks like data storage, modification, querying, and exporting in different formats
- Apply fundamental techniques of data cleaning (for instance, finding inconsistencies, duplicates, and replacement using regular expressions)
- Integrate a user-tracking system

Version History:

Jan. 20, 2025: V1.0