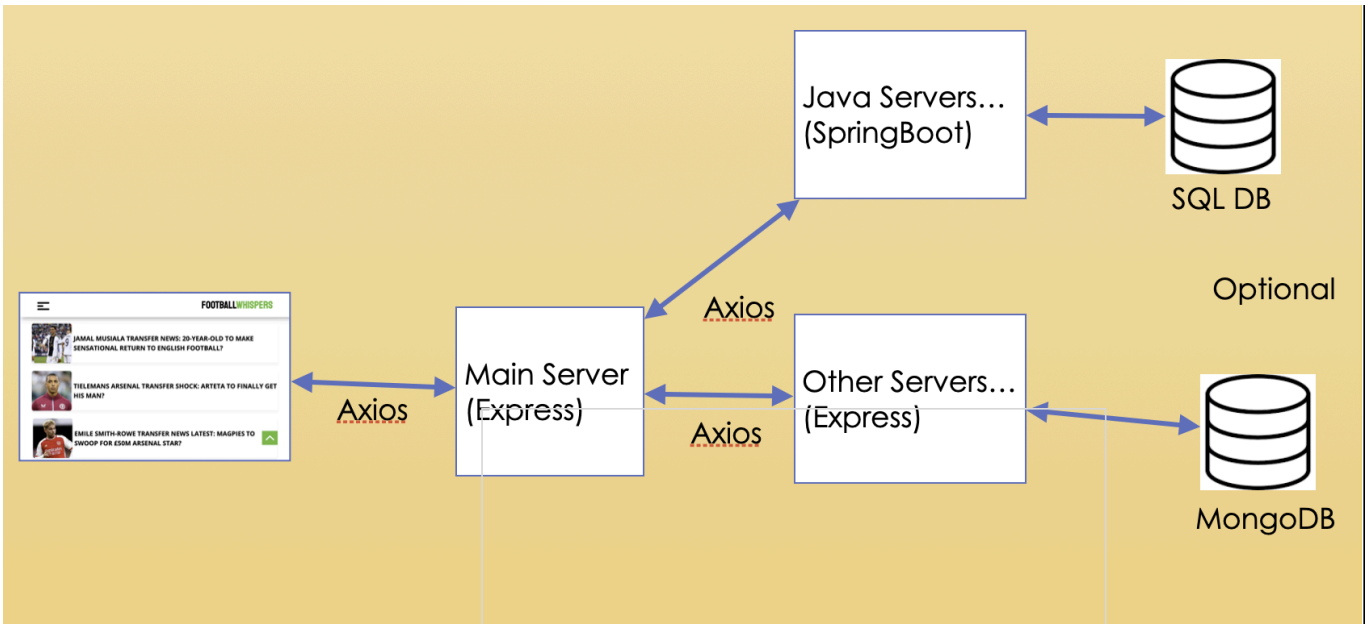
Assignment TWEB for 2025-2026

1. Introduction

This assignment is primarily concerned with applying the ideas that are being presented in the module on methods for accessing the Web and making sense of its content. In providing a solution, you are **required to use the methods and techniques taught in the module**.
You are given a set of data about animes over the years.
The data is provided as a set of CSV files.
You are requested to create a system able to support access and analysis by fans and journalists (experts).

2. Requirements

Here is the outline of the required architecture.
The architecture is composed of a number of servers, some of them will be written in Javascript (Express), and some of them in Java Spring Boot. They interact with two types of databases: MongoDB and Postgres where the Transfermarkt data is to be stored.



Functionalities:
The goal of the work is to provide the possibility for fans and journalists to access data about anime. You must design service with the following features:

- An HTML+Javascript+CSS+Bootstrap+Handlebars page enables querying and exploring the data.
    - Note: you must use Handlebars or - if you really must - React. No other framework is allowed
- The servers:

    - Queries are sent to a central server implemented in Express which will communicate to two other servers for database access, at least one written in Express and one in Java Spring Boot. The central server must be fast and be able to serve thousands of users, so it must not perform any heavy duties.

- The data will have to be divided into two subsets:
    - the dynamic data with a reasonably fast change rate e.g. the reviews,
        - This should be stored in a MongoDB.
    - the more static data such as the data about the features, the people details, etc. This should be stored
        - in a PostGres database

3. The Data Provided

You are provided with the following data files:

- Characters.csv: 209,963 characters details
- Character_anime_works.csv: it contains 236,816 characters and roles in the database
- Character_nicknames.csv: 37,080 nicknames of characters
- Details.csv: 28.955 anime details
- Favs.csv: 4,178,747 likes of users
- Person_alternate_names.csv: 20,465 alt names for people (e.g. actors)
- Person_anime_works.csv: 458,091 roles of people in anime (e.g. singing the theme song)
- Person_details.csv: 76,699 people details
- Profiles.csv: 337,155 user profiles
- Ratings.csv: 124,298,357 ratings given by users
- Recommendations.csv: 105,249 user recommendations
- Stats.csv: 28,955 user stats (ratings, number of completed series, etc.)

5. Groups

The assignment is to be done in groups. Experience has demonstrated that it is not really possible for one person to do all the work for the assignment, unless they are an exceptionally competent programmer with previous knowledge of Web programming. Therefore, the assignment is organised on the basis that people should work in groups. Groups must be composed of

- **a maximum of 3 members**

Students are also allowed to do the assignment in pairs or on their own.

6. An important Remark

The assignment per se is open ended. You could spend a year building a fantastic website or producing data analytics and visualisations, inspecting several nuances of the data or the tasks.
That is definitely not the point of the assignment. The point of the assignment is to test if you master the technologies and methodologies introduced by the module. Therefore be sure to check the self assessment forms that tells you in detail what you will be evaluated upon.

7. Material Allowed - Plagiarism

You can freely use the lecture notes, lab classes examples and all materials used during the module. No third party code can be used in the assignment, except what has been explicitly provided in the lectures or lab classes. For example you are allowed to use some code given in the lecture slides but **you are not allowed to download any code from the Web or to use any other software that will perform a considerable part of the assignment**. Unauthorised re-use of third party software will be considered plagiarism. In case of doubt ask the lecturer for permission before using any third party code.
You must not use any support for server implementation or communication with the Express or Spring Boot servers. For example you cannot use any functions that would replace an Axios call.
For other libraries, please ask the lecturer before using them.
The use of Generative AI tools is allowed, actually encouraged. However you are required to explain in your report how you used it. Also be sure to fully understand what you use because the oral examination will test your understanding.

8. Marking schema

Each part of the assignment will carry marks divided as follows:

- 25% for the documentation in the project report.
- 50% for the implementation of your solution (quality of code and data analysis). See appendix A for further details on the specific marking schema.
- 25% for the correctness of results.

9. Handing in

Your solution must be submitted **at least one week prior to the exam date** in a self-contained folder named after the group name (<MainDirectory> in the following).

The directory must contain:

1. The code of the solution (please note that we will both inspect and run the code). We must be able to run your solution without problems on a standard computer. It should not need any applications or code or libraries to be installed in order to run. Use *npm* or *pip* or *Gradle* to make sure the system knows what libraries to load automatically. All the code should be in the directory **`<MainDirectory>/solution`**. Please note that the quality of the code carries a relevant portion of marks, so be sure to write it properly.
2. A report documenting your work. The report must be contained in the directory **`<MainDirectory>/report/`**. An outline and set of requirements for the report are provided in Appendix A.
3. The documentation within the code (Javascript/HTML/Java/etc.) files must be of very high quality, both in terms of Javadoc (or equivalent Javascript) or in terms of Swagger documentation for the ALL the servers. Please note that this documentation carries a relevant portion of marks, so be sure to write it properly.
4. The filled self assessment form. The self assessment forms are available at this link:

To submit your work, you need to upload all the developed materials, including self-assessment forms and the project description report, to your **GitHub or GitLab repository**. This repository must be shared with the lecturer and lab assistants as explained above.

Afterward, you need to register for the exam (see the next point) by the submission deadline and clearly indicate the link to your repository in the spreadsheet under column 4. This will count as the submission of your project.

Simply sharing the repository with us is not sufficient as a submission because:

- Most of the shared repositories do not clearly identify the author (many times the message is something like "Lillo194 has shared a repository with you").
- Often, the repository is shared at the beginning of the work, so it is unclear which exam session it is for (we risk evaluating incomplete projects).

9.1. Exam Booking and Submission

You must register for the exam through the UniTo system as usual, but you also need to take the following action:

Go to the booking spreadsheet by the submission deadline (one week before the exam) and book a slot:

21/01/2025 Room E

Time	Group Members (name and surname)	Notes (e.g., course and credits)	GitHub/GitLab Repository (must be private and shared with me and the assistants)
09:00			
09:30			
10:00			
10:30			
Break			

Here, the repository link must be added. **This will count as the project submission**.

9.2. Division of work

It is required that every member of the group works on all parts of the solution.

9.3. Private Github/Gitlab Repo

The solution must be provided via a link to a Gitlab or Github repository **at least one week prior to the exam date (i.e. if the exam starts on the 27th of September, the deadline is 23:59 of the 20th of September). Any solution submitted for whatever reason after the deadline will be rejected**. The repository must be used **regularly** during the development of the project by **all members** of the group. The repository will be used during the examination phase to:

- check that the project has actually been developed by the group
- provide clear indications of the contribution of each group member

For this reason, each group member will have to upload (commit) **personally** all the changes to the repository, ideally for each day of work on the project. That is, (i) the group must not delegate the task of

making commits to a single member and (ii) commits must not be made only when the code (or a part of the code) is fully completed. We should see the intermediate work, ideally in separate branches.
The name of the group members must be clearly identifiable with name and surname (e.g. Giovanna Rossi instead of xyzaa). Be very careful when you have more than one user e.g. in Github not to submit with multiple identities.
You are allowed to use several github/gitlab repositories, each for a different part of the project, e.g. one repo for each of the servers, etc. However please provide just one point of download (i.e. one repo allowing downloading all others).

Please note: if the group does not provide a satisfactory recording of the commits to demonstrate the actual progress of the project, the project will be rejected and the assignment redone from scratch.

The repositories must be shared with

- **Fabcira, LuisBarrios03** on Github
- Ciravegna, https://gitlab2.educ.di.unito.it/ciravegna, and Luis Andre' Barrios Luna Victoria https://gitlab2.educ.di.unito.it/st224596 on the University Gitlab.

Important Note: on Gitlab you must include the lecturer and markers as **full team members** as any other role (e.g. guest) would not allow access to your code.

10. Appendix A: Report Outline and Marking Schema

It is important that you produce a high quality report. Be sure not to leave the report at the very last minute. It is important that you do not ignore techniques and examples provided to you during the lectures and lab classes. Referring back to them during your planning/implementation stages will give you the base from where to start, as well as a point of comparison/discussion where your ideas differ from what already presented to you (i.e., do not reinvent the wheel – if it has been done before, reuse it and complement it where it lacks functionality). You may use diagrams to aid your explanation in these sections, but please note that a diagram alone is not acceptable and must be accompanied by an explanation/discussion.
Your report must be organised according to the following outline, which is designed to help you ensure that all the required information is provided.
Length: **maximum 5 pages**. Any additional pages will be ignored.

Index Table

[No marks – OPTIONAL] - You may or may not want to include an index table on your report. This is completely up to you. This does not count against the page limit.

Introduction

[no marks] – This should give the marker a guide as to what to expect from your report. Please make the marker's work easier by being honest.

For each technical tasks:
Create a subsection in your report for each of the technical requirements. It is very important that they are documented individually by explicitly following the organisation below.

- **Solution:**
  - Design and its motivations: explain how your solution works and explain why you chose to design your solution in this particular way. Does it have advantages/disadvantages over other design choices?
- **Issues**

    - Introduce the task this section refers to and the challenges that you were faced with.
- **Requirements**:
    - How does this design comply with the requirements specified in the original assignment sheet? Are you meeting all requirements?
- **Limitations**:
    - Have you thought about exceptional situations that may limit your solution? Is your solution extensible? Can it be easily adapted for other requirements? Remember that no design is flawless and that is ok!

Moreover you should add:

- **Conclusions**
    - [no marks] - You should include here any relevant conclusions you've collectively arrived at regarding the process of designing the solution for this part of the assignment as well as any lessons learned.
- **Division of Work**
    - [No marks – MANDATORY] – Have all the group members shared the workload in a balanced way? In what way? E.g. what has each member provided – be precise!
- **Extra Information**
    - [No marks – MANDATORY] – This section should include any extra details needed to run your code. If no extra configuration is needed, please explicitly say so in this section.
- **Bibliography**
    - [no marks] - Do not forget to cite any sources and reference these within the text where appropriate (e.g., “(...) we have used the techniques as per [1].”).Make sure to use a standard format style. No need to cite the lecture notes or lab classes.