

Project: A "Simple" KR&SW Application

In this project, you will develop a meaningful application that utilizes concepts and technologies learned throughout the course. Your application should address a specific "real-world" problem for end users and incorporate advanced data processing techniques. Your application must go beyond merely displaying data. It should involve some advanced processing techniques, including, but not limited to, intelligent integration of diverse data sources, filtering and comparing data, reasoning and learning data, and summarizing complex data insights.

Project Team

Please sign up on Canvas for a project team. Each team consists of 3 to 5 students. It is recommended that the team has a good balance of students from the different programs, i.e., TGCD1, TAAI1, and EPTIECH.

Project Seminars

You must attend two project seminars to pass the project:

- **Idea presentation:** In this seminar, you should present a concrete idea for your project, which should include:
 - The domain, use cases and the functionalities of the application.
 - The concepts and technologies that will be used, focusing on those learned from this course.
 - The data sources to be used.
- **Project presentation:** In this seminar, you should:
 - Demonstrate the main functionalities of the application.
 - Present the fundamental structure of the application system.
 - Discuss the concepts and technologies that are used, focusing on those learned from this course.

Project Deliverables

- A **pdf** document with your written report where you describe the application, the design and implementation of the application, and some discussions and conclusion. The report should follow the provided LaTeX report template.

- A **zip** file or a **link** to the repository on platforms like github, containing the source code of your application implementation.

There are no restrictions regarding the project implementation: software platform(s), programming language(s), framework(s), libraries, public APIs, source-code management, etc. The code-source and specific content (data) must be available under the terms of the open source licenses, such as GPL, Creative Commons and/or Open Database.

- A **pdf** document specifying the contributions of each project team member, signed by all members. Please prepare the document using the provided project contract template.

Grading Criteria

Grading criteria is given in Table 1 below.

Criteria	Requirements	Points
Application	The application has been developed as described in the report and functions as intended.	35
Creativity	The more interesting the application is, the more points you will gain.	10
Report	The report should be easily understandable. The clearer, more concise, and consistent the report is, the more points you will gain.	30
Justification of Claims	The more consistently the claims you make are justified, the more points you will gain.	20
Formatting	The more correctly the required formatting guidelines are followed, the more points you will gain.	5
<i>Total</i>		100
<i>Bonus</i>	<i>A team with members from more than one program.</i>	5

Table 1: Grading Criteria and Points

The final grade will be assigned individually to each team member, based on individual contributions to the project and performance in project seminars. Specifically, the grade will be calculated as **project_points * individual_weight**, where project points are maximized at 100, and individual weight is in the range [0, 1]. The better the individual contribution and performance, the higher the individual weight. The grade will be 3 for points in [50, 69], 4 when points in [70, 84], and 5 when points in [85, 100].

Note: Any form of plagiarism, including the use of AI to generate code and write the report, will result in failing the project.

Example Project Ideas for Inspiration

Music Recommendation System

The application provides a personalized music recommendation system that suggests songs and artists based on user preferences. It uses ontolog(ies) to integrate data from multiple knowledge graphs (at least two) such as YAGO, MusicBrainz, Wikidata, and DBpedia. Rules are applied to suggest music based on user-defined criteria and to provide personalized recommendations by inferring relationships across data sources. A similar idea can be applied to other domains, such as movie and gaming.

Interactive Science Knowledge Base

The application allows users to explore relationships, trends, and insights in the scientific community. It uses ontolog(ies) to integrate data on researchers, their works, publications and related scientific concepts from multiple knowledge graphs (at least two), such as Open Research Knowledge Graph (ORKG), Wikidata and DBpedia. Rules are implemented to answer questions by inferring across data sources. The system could offer personalized answers by dynamically inferring relationships across data sources. A similar idea can be applied to other domains, such as culture and historical events.

Interactive Product Configurator

This project would involve building an interactive product configurator, where users can assemble a customizable product (such as a computer, car, or furniture). Components and constraints (e.g., compatibility, requirements, limits) are defined in XML, and ASP is used to check the feasibility of a user's selection. ASP also suggests valid completions or modifications if needed, ensuring the configuration meets all constraints.

Puzzle Game with ASP-based Hint System

In this project, the goal is to build an interactive puzzle game, such as Sokoban, where players move blocks to reach a goal. ASP would be used as a background solver to generate hints or check if the puzzle remains solvable after each player move. If no solution is possible due to player actions, the system can alert the player.