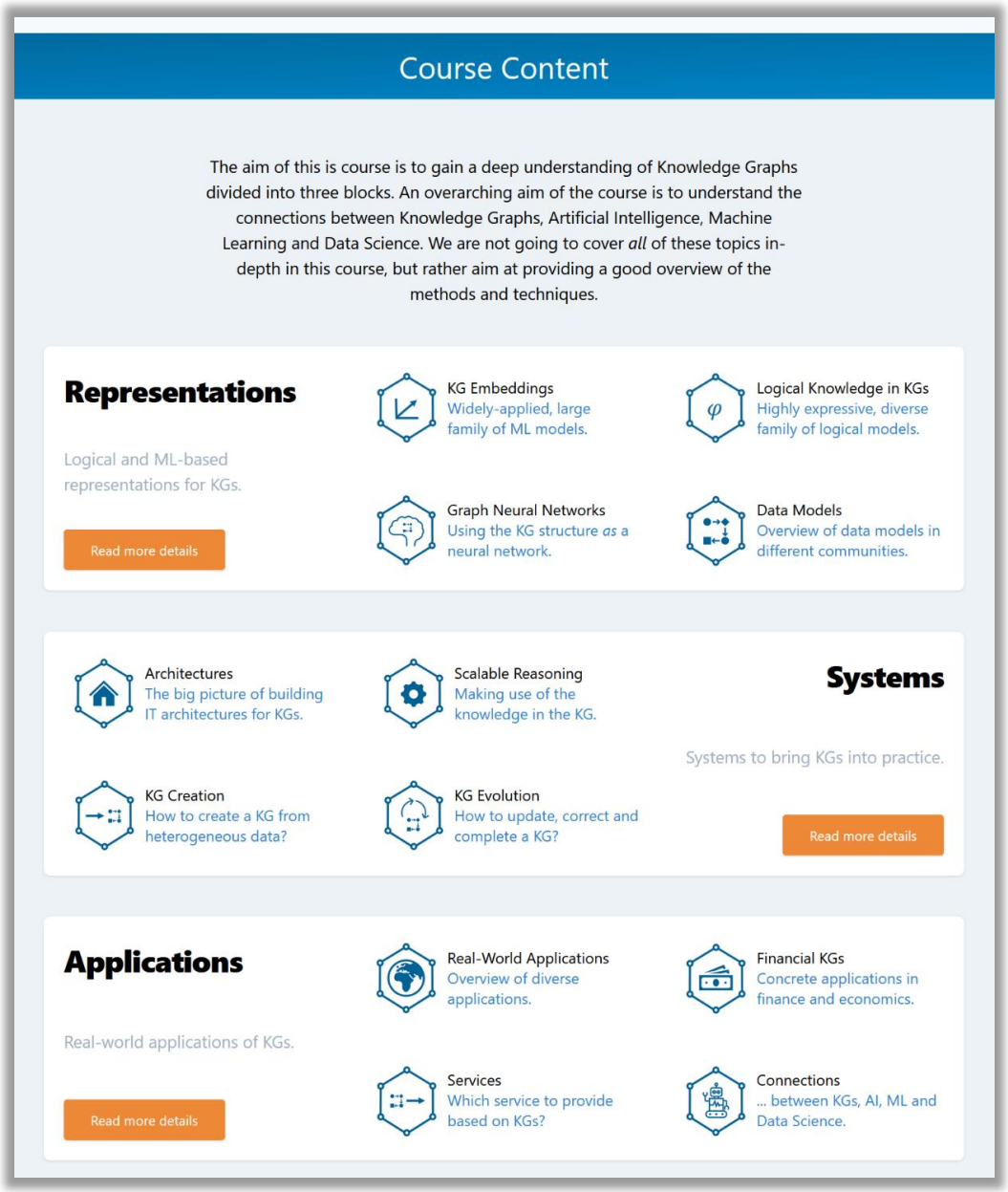
Portfolio Example Structure

Recall the learning outcomes of the course (here listed as an overview):



The main goal of this document is to give you an **idea** how you could **structure your portfolio** while **covering the learning outcomes** you wish to demonstrate. Please note that this is an example, and will differ depending on the project you chose and the learning outcomes you want to demonstrate. You *do not* need to adhere to this example.

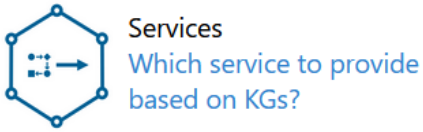
If you choose this structure, please indicate it in the filename of your PDF file such that it ends in "-structured.pdf". Please do include the portfolio cover pages as in the pro forma - but it should be very easy and quick to refer to the respective pages (see an example in the end).

1. Scenario (recommended to be no more than 1 page)

The first step in this is to **succinctly describe the domain** of your knowledge graph application. The corresponding learning outcomes are:

Note that you are not required to pick the financial domain, it is mostly to give you an example of how knowledge graph applications in particular domains can work.

The second step is to **succinctly describe the service** that your knowledge graph provides. The corresponding learning outcome is:



Examples (but not an exhaustive list) are:

- A particular **query** that you want to execute over your knowledge graph
  (this can also be a class of queries if a single query is too constrained)
- A **recommendation** to a user that should be provided by your knowledge graph
- A particular **dataset** that is constructed from your knowledge graph
- …

Consider "service" as general as described in the course and learning outcomes, i.e., simply write down here what the outcome of your project is.

2. KG Construction

In this part, your goal is to describe how you constructed your KG. Keep this writing reasonably short, it is just one part of your project (if you did construct a KG).

The first step is to **succinctly describe the datasets** your knowledge graph is based on. This will differ based on if it is one or many datasets you use, whether you use publicly available, non-publicly available, or self-constructed datasets. See some guidance below as well.

The second step is to **succinctly describe the technologies** you are using to store and process your knowledge graph. The corresponding learning outcome is:

Architectures
The big picture of building IT architectures for KGs.

Give a succinct overview, this is just for you to show what technologies introduced in the course (or not) you have chosen – unless this is one of the LOs you want to particular focus on.

The third step is to **design and succinctly describe how you constructed** your knowledge graph. The corresponding learning outcome is:

KG Creation
How to create a KG from heterogeneous data?

You could e.g.:

- Include *3 concrete examples* of how a particular type of node or edge of the graph was constructed based on the datasets you are using.

We suggest that you create one subsection for each of the above three steps (i.e., Subsection 2.1 is on the description of the datasets) – similar also for the entire assignment.

Guidance on Code

Include all of the code you are using in the ZIP file (in a subfolder named "2 - construction"). If appropriate, feel free to give some examples of your code in the appendix of your assignment report (but please refrain from simply copying full code if there is no particular point you want to give evidence for with it).

- For public datasets: Make sure that you provide a (stable) link in your assignment report to the public dataset that allows for reproducibility of your results.
- For non-public/self-constructed datasets (or if no stable link exists): Include the dataset in the single ZIP file that you can provide in addition to the single PDF file that is part of your submission (in a subfolder named "2 - construction"). If the dataset is larger than possible in the ZIP file, provide in the report a link to a shared file, and include in the ZIP file a subset of the datasets that allows partial reproducibility.

Please, if not self-explanatory, include a brief readme.txt (or readme.md) file in the ZIP file (in a subfolder named "2 - construction") for how to execute your code.

3. ML-based Representation

In this part, your goal is to demonstrate proficiency of the following learning outcomes:

This is of course only the case if you chose one (or both) of these .

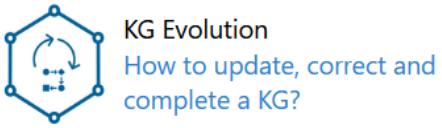
The first step is to **succinctly describe the ML-based representation** you chose, and how you trained it. For example you can:

- Explicitly give 5 examples of how particular nodes or edges are represented in your chosen ML-based representation.
- Include one example of a node or edge predicted by your representation that should be there (true positive).
- Include one such example that should not be there (false positive).

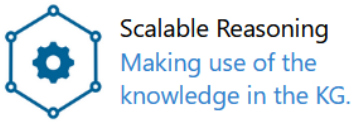
The second step is to **succinctly describe** to how your ML-based representation is used to **evolve your knowledge graph**:

- Is it used to complete your KG?
- Is it used to update/correct your KG?

It is sufficient to pick one of the above, but make sure to describe if it has an effect on evolving the knowledge graph (adding something / removing something / changing something). The corresponding learning outcome is:



The third step is to **succinctly describe the context and limitations** of your approach. The corresponding learning outcome is:



Some of the questions you may ask yourself are:

- Will the approach scale in different dimensions? (volume, veracity, variety, etc.)
- How will it scale in situations where there is little training data?

Follow the guidance on datasets and code as in part 2, use the ZIP file folder “3 - ML”.

4. Logic-based Representation

In this part, your goal is to demonstrate proficiency of the following learning outcome:

The first step is to **design and succinctly describe the logic-based representation** you chose. For example you can:

- Explicitly give 5 examples (queries, rules, …, depending on what you chose).
- At least one of these examples could include *recursive exploration*, i.e., following connections in the graph which could be connected through arbitrarily many edges.
- At least one of these examples could *create* some new node or edge in the graph.

The second step is, as in part 3, to **succinctly describe** to how your logic-based representation is used to **evolve your knowledge graph**:

- Is it used to complete your KG?
- Is it used to update/correct your KG?

It is sufficient to pick one of the above, but make sure to describe if it has an effect on evolving the knowledge graph (adding something / removing something / changing something). The corresponding learning outcome is:

The third step is to **succinctly describe the context and limitations** of your approach. The corresponding learning outcome is:

Some of the questions you may ask yourself are:

- Will the approach scale in different dimensions (volume, veracity, variety, etc.)
- Could you improve your approach to scale in some of these dimensions?

Make sure you make explicit reference to your concretely chosen scenario in your discussion, and do not only generically describe these aspects.

Follow the guidance on datasets and code as in part 2, use the ZIP file folder "4 - logic".

5. Reflection (recommended to be no more than 1.5 pages)

In this part, your goal is to reflect on the different parts of your exercise.

The first step is to **reflect on and succinctly describe the outcome of the service** you provided through your knowledge graph (which you initially formulated in part 1):

Follow the guidance on datasets and code as in part 2, use the ZIP file folder "5 - reflection".

The second step is to **briefly reflect on and the data model you chose** and how it compares to possible alternatives:

The third step is to **reflect on and the connections between the different forms of AI** you have used during this assignment:

Some of the questions you may ask yourself are:

- How is the interaction between the ML-based knowledge and the logic-based knowledge in the overall knowledge graph?
- Could this interaction be improved?
- Would it be possible to learn some of the logical representations automatically?
- Are there ML-based representations you used whose accuracy could be improved via logical representations?

Stay concrete in this discussion relative to your scenario and do not only generically discuss your chosen aspects.

EXAMPLE LO SECTION OF YOUR COVER PAGES

(i.e., if you use the example structure, you can use a briefer formulation of the cover pages as in the example below)

Representations

(LO1) Understand and apply **Knowledge Graph Embeddings**	☐ I showed **basic** proficiency ☐ **exceeded** basic proficiency
I constructed a KGE, described in a dedicated section.	Section 3 and particularly 3.1 (page X)

(LO2) Understand and apply **logical knowledge** in KGs	☐ I showed **basic** proficiency ☐ **exceeded** basic proficiency
I constructed a logic-based representation, described in a dedicated section.	Section 4 and particularly 4.1 (page X)

(LO3) Understand and apply **Graph Neural Networks**	☐ I showed **basic** proficiency ☐ **exceeded** basic proficiency
I constructed a GNN, described in a dedicated section.	Section 3 and particularly 3.1 (page X)

(LO4) Compare different Knowledge Graph **data models** from the database, semantic web, machine learning and data science communities.	☐ I showed **basic** proficiency ☐ **exceeded** basic proficiency
I reflected on this in a dedicated section.	Section 5.2 (page X)

Systems

(LO5) Design and implement **architectures** of a Knowledge Graph	☐ I showed **basic** proficiency ☐ **exceeded** basic proficiency
I described my architecture in a dedicated section.	Section 2.2 (page X)

(LO6) Describe and apply **scalable reasoning** methods in Knowledge Graphs	☐ I showed **basic** proficiency ☐ **exceeded** basic proficiency
I described this in a dedicated section together with the KGE/GNN/logic-based approach.	Section 3.3 and 4.3 (pages X,Y)

(LO7) Apply a system to **create** a Knowledge Graph	☐ I showed **basic** proficiency ☐ **exceeded** basic proficiency
I described this in a dedication section.	Section 2 (page X)

(LO8) Apply a system to **evolve** a Knowledge Graph	☐ I showed **basic** proficiency ☐ **exceeded** basic proficiency
I described this in a dedicated section together with the KGE/GNN/logic-based approach.	Section 3.2 and 4.2 (pages X,Y)

Applications

(LO9) Describe and design **real-world applications** of Knowledge Graphs	☐ I showed **basic** proficiency ☐ **exceeded** basic proficiency
I described this in a dedication section.	Section 1.1 (page X)

(LO10) Describe **financial Knowledge Graph** applications	☐ I showed **basic** proficiency ☐ **exceeded** basic proficiency
I described this in a dedication section.	Section 1.1 (page X)

(LO11) Apply a system to provide **services** through a Knowledge Graph	☐ I showed **basic** proficiency ☐ **exceeded** basic proficiency
I described this in dedication sections.	Section 1.2 and 5.1 (pages X,Y)

(LO12) Describe the **connections** between Knowledge Graphs (KGs), Machine Learning (ML) and Artificial Intelligence (AI)	☐ I showed **basic** proficiency ☐ **exceeded** basic proficiency
I reflected on this in a dedicated section.	Section 5.3 (page X)