Database Design

Page DB

Basic Features per Node (in their own dictionary called “Features”):

Feature	Type	Value Range	Why?
date_created	DateTime	Valid DateTimes since 2022	Should have access to the creation date to see if outdated.
date_last_modified (ACID)	DateTime	Valid DateTimes since 2022	Hackers
page_content_md	String (general type)	Content pre-validated for curse words and racist slurs	For Display - Autorenders everything except for prerequisite and more information links
page_rating	Float	0-5	For Recommendation
num_likes	Integer	>= 0	For Recommendation and Users
num_dislikes	Integer	>= 0	For Recommendation and Users
like_dislike_ratio	Integer	>= 0	For Recommendation
prerequisites	Map	Valid link map	Can map prerequisites to link so users can go explore prerequisites.
more_info	String/Map	Valid link map	May include interactive exercise links. In general, it is good to cite your sources.
is_draft	boolean	true/false	User may want to save as draft
course	string	Any course defined on website	Want to be able to group it with other members of same course

ui_dict	Map	Only valid ui attributes decided upon	Something will need to store page UI for consistency

Node Relationships:
We plan on defining two elementary relationships, these being:

1. **Prerequisite -** A certain page is a prerequisite for another page, and we want this known since we could then proceed to link to the page upon population. Also included in the Database for the time being for quick post, but for other types of queries we will likely need to link to these pages directly. This is a unidirectional relationship.

2. **Alternative (Optional) -** Bidirectional relationship for Nodes of the same topic. Way to recommend alternatives to users who did not find the content of one of the posts helpful.

Node Labels:
Neo4j allows topic labels, and we plan to utilize these to define the “fundamental premise” of the Node. The labels of a Node could work as a HashSet to define a set of topics applicable to the Node that we can use to group the pages into topic sets. Search results could work first by similarity of grouping, then by similarity of topics. Similarity will be treated in one other way, that way being via relationship. The starting labelings for math and Computer Science are given below.

Current Topic Labels in Math and Computer Science:

- M1 Algebra
- M2 Calculus and analysis
- M3 Geometry and topology
- M4 Combinatorics
- M5 Logic
- M6 Number theory
- M7: Dynamical systems and differential equations
- M8: Mathematical physics
- M9: Theory of computation
- M10: Information theory and signal processing
- M11:Probability and statistics
- M13: Game theory
- M14: Operations research

- C1:Computability
- C2: algorithms
- C3: computational complexity
- C4: computer design
- C5: programming language design

- C6: programming methodology
- C7: data structures
- C8: information retrieval
- C9: parallel and distributed computing
- C10: computer networks
- C11: cyber security
- C12: artificial intelligence

User DB

Feature	**Type**	**Restrictions**
username	String	Letters, Numbers, and Special Characters - to avoid problems (space illegal?)
password	String	Letters, Numbers, and Special Characters - to avoid problems (space illegal?)
posts_written	Map	Valid Posts
posts_disliked	Map	Valid Posts
posts_liked	Map	Valid Posts

User Relationships:

Naturally, attaching the **Following** (Unidirectional) and **Followed By** inevitably makes sense in a Github like format, but doesn't make sense to have initially.