BalkanID

FTE Hiring Task

Submission Instructions

There are two tasks, one in **Engineering** and the other in **Data Science**. You are required to choose and attempt **only one of these tasks**.

You need to submit your assessment by **22nd July 2023, 7:30 PM IST.** No further extension will be given.

Join the GitHub classroom first using the link below:
https://classroom.github.com/a/YCCXVJKc

This will create a repository for you. Kindly make all commits and code pushes to this repository only. **No other repository submissions** will be considered.

After completing your task, fill your submission details on this Google Form:
https://forms.gle/EAiL3bgwFqo9pHxo8

Engineering Task

Problem Statement

Build a robust containerized task management system to handle user authentication, authorization and access management.

Key Features

- Secure user registration and authentication
- Account Deactivation and Deletion: Allow users to deactivate or delete their accounts, if applicable. Implement a mechanism to handle account deletion securely while considering data retention policies.
- Role-based and Group-based access management on resources(Tasks) with ability to create custom roles and groups (Need to make sure endpoints are secure)
- Protection against vulnerabilities like SQL injection attacks
- Support for bulk upload using CSV(Both users and tasks) making sure all the relationships are preserved accurately

Requirements

- Make the necessary APIs to expose data
- Make sure the entire application is dockerized using compose with proper use of networks and volumes
- SQL based database - **PostgreSQL** or **MySQL**
- Use a reverse-proxy of your choice.

Guidelines

- You can choose to write your program in **Golang only**. It is crucial to ensure that your code is **well organized and easy to understand**, and that you provide clear instructions on how to run your program.

- All plagiarized submissions will be disqualified. Please **ensure that you use a VCS Platform like Github** and commit and push all your contributions on time. Kindly share the same.

- The Github repository must have a file called "**README.md**" which contains information about how to install and run your project, along with a clear understanding of your project, including relevant diagrams, if any. If you have deployed your application, you may include a link in the README.

- The task judging metrics will be based on the following points:
    - **Correctness:** Does the code correctly implement all the features desired in the problem statement?
    - **Code Quality:** Does the code follow best practices to ensure that it is clean, readable, maintainable, and comments are provided wherever necessary?
    - **Efficiency:** Is the code efficient (i.e, it performs operations in a sufficiently performant manner with less use of resources)?
    - **Secure:** Does the code or the public repository have any security issues/vulnerabilities?

Bonus Points:

- You can use Docker and containerize your application code to run, including the database.
- Multi-tenant functionality

Data Science Task

Problem Statement

Knowledge graphs have emerged as powerful tools for representing and organizing information in a structured and interconnected manner. They enable a deeper understanding of complex domains by capturing relationships and contextual information between entities. By representing knowledge as a graph, with entities as nodes and relationships as edges, knowledge graphs provide a holistic view of the data, facilitating efficient data integration, knowledge discovery, and advanced analytics.

Dataset link: https://bit.ly/balkanid-ds-dataset-download

Task: Building a Knowledge Graph and Recommender System for Research Papers

Part 1: Charting the Research Landscape

> You are part of a team of researchers who have embarked on a quest to uncover hidden knowledge and insights within a vast collection of research papers. As you journey through the realm of research, you discover the importance of organizing and modeling metadata effectively. Utilizing the RDF framework and the expressive SPARQL language, build a knowledge graph that represents the interconnectedness of research papers, their authors, citations, and other relevant metadata. Showcase your skills by creating a powerful visualization tool that enables researchers to visualize the results of SPARQL queries as a subgraph.
>
> Note: Draw a diagram to illustrate the ontology of your knowledge graph depicting entities and their relationships using a diagramming software (like lucidcharts, powerpoint, draw.io, etc.). Also, provide at least 5 examples of the visualizations generated by executing SPARQL queries using the visualization tool that you have built.

Part 2: The Search for Knowledge

> To aid fellow researchers in their exploration, your task is to design and implement a recommender system that suggests 5 similar and important

research papers that a researcher could cite for a new research paper that he/she plans to write. Embrace the power of graph-related algorithms and learning models to generate meaningful recommendations. If implementing graph algorithms for recommendations proves challenging, use any other algorithm, but provide a detailed description of the steps you would take to incorporate graph algorithms in the recommender system.

In new_research_papers.jsonl (https://bit.ly/balkanid-new-data-download) we have provided the title, discipline and abstract of 3 papers that the researcher plans to write. Using your recommender system, for each new research paper, recommend 5 similar and important papers that the researcher could cite (in order).

Note: Importance of a research paper is measured not just by the number of citations that it receives but also by the quality of the citations. The quality of a citation is in turn determined by the number and quality of its own citations.

Part 3: Unveiling the Trailblazers

Delve into the heart of impact analysis and embark on a mission to identify the research works that have made the most significant impact on their peers. Explain and implement different methods of evaluating that you have incorporated to calculate how impactful a paper is, in a detailed manner. List the top 5 papers that you think are the most impactful.

Note: Document in a detailed manner, how the impact analysis was executed, along with justifications as to why you chose the methods used.

Guidelines

Choose one or more parts of the task based on your interest and abilities. Your implementation will be evaluated based on the level of completion and the quality of the implemented components. Remember to document your approaches and findings clearly, with easily accessible visualizations. In the documentation, cite or provide links to your code wherever relevant.

Some information on the structure of the dataset:

Papers (JSON objects saved as a single line in a JSONL file) have the following format.

- paper_id: arXiv ID of the paper
- _pdf_hash: always None
- _source_hash: SHA1 hash of the arXiv source file
- _source_name: name of the arXiv source file

- metadata: paper metadata from kaggle.com/datasets/Cornell-University/arxiv
- discipline: scientific discipline of the paper
- abstract: paper abstract copied from metadata
- body_text: list of paper content sections (paragraphs, listings, etc.)
    - section: section name
    - sec_number: section number
    - sec_type: section type (section, subsection, etc.)
    - content_type: content type (paragraph, listing, etc.)
    - text: text content
    - cite_spans: list of citation markers
        - start: starting character offset in text
        - end: ending character offset in text
        - text: surface text
        - ref_id: dictionary key for linked content in bib_entries
    - ref_spans: list of referenced non-textual content (figures, formulas, etc.)
        - start: starting character offset in text
        - end: ending character offset in text
        - text: surface text
        - ref_id: dictionary key for linked content in ref_entries
- bib_entries: list of bibliographic references
    - bib_entry_raw: raw bibliographic reference string
    - contained_arXiv_ids: list of linked arXiv papers
        - id: ID of linked arXiv paper
        - text: text segment in reference that the link was attached to
        - start: starting character offset in bib_entry_raw
        - end: ending character offset in bib_entry_raw
    - contained_links: list of embedded links
        - url: URL of link
        - text: text segment in reference that the link was attached to
        - start: starting character offset in bib_entry_raw
        - end: ending character offset in bib_entry_raw
    - discipline: scientific discipline of the cited paper
    - ids: matched identifiers of referenced paper
        - open_alex_id: referenced paper's OpenAlex ID
        - sem_open_alex_id: referenced paper's SemOpenAlex ID
        - pubmed_id: referenced paper's PubMed ID
        - pmc_id: referenced paper's PMC ID
        - doi: referenced paper's DOI
- ref_entries: list of non-textual content (figures, formulas, etc.)
    - type: content type

    - caption: table/figure caption (if type is table or figure)
    - latex: content of LaTeX math mode (if type is formula)

- All plagiarized submissions will be disqualified. Please **ensure that you use a VCS Platform like Github** and commit and push all your contributions on time. Kindly share the same.

- The Github repository must have a file called "**README.md**" which contains information about how to install and run your project, along with a clear understanding of your project and the approach you have taken, including relevant diagrams, if any.