BalkanID

Summer Internship Task

SSN Chennai - Batch of 2025

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 **26th September 2023, 11:59 PM IST.** No further extension will be given.

Join the GitHub classroom first using the link below:

Classroom Link: https://classroom.github.com/a/rd0kEjs3

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

Engineering Task

Problem Statement

Build a robust online book store 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.
- Protection against vulnerabilities like SQL injection attacks
- Have Proper system logging with retention policies upon system failure
- Users can easily search and filter books and add them to shopping cart
- Users can easily download their bought books and leave a review on the books they bought
- Admins have the ability to manage inventory and others.
- A simple frontend using Next.js and React.js with MaterialUI or TailwindCSS as the CSS Library.

Requirements

- Make the necessary APIs to expose
- SQL based database - PostgreSQL
- 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
- You can test your code by adding **unit test cases** and **workflow test cases**
- You can add a recommendation **system** to recommend books to user.
- Unit test cases for your frontend using a framework like **Jest**.

Data Science Task

Knowledge graphs have become influential tools in arranging and portraying information in an interconnected and systematic way. They offer a more profound comprehension of intricate areas by documenting the connections and context among various entities. Through the depiction of knowledge in a graph form, where entities act as nodes and their interconnections as edges, knowledge graphs offer a comprehensive perspective on data. This aids in seamless data amalgamation, the unveiling of new knowledge, and sophisticated analysis.

Graph Neural Networks (GNNs) have risen as groundbreaking computational models for processing information directly on graphs. By operating on a structure where data points (nodes) and their interactions (edges) form the foundation, GNNs can learn and capture intricate patterns within relational data. GNNs have found applications in various fields, enabling superior data interpretation, pattern recognition, and predictive analytics in domains where traditional neural networks might fall short.

Task: Providing Authors with Co-author suggestions

Authors often collaborate to co-write publications, seeking collaborators who either resonate with their writing style, can expand their readership, or make for harmonious teamwork. A collective of such authors has approached you, each eager to discover fresh co-authoring opportunities. While some within this group have previously joined forces to produce works, they're now on the lookout for new and intriguing collaborative ventures.

Link to dataset: https://bit.ly/BalkanID_DSTask_dataset
The provided file is a dump file from a Neo4j Database (A Graph Database). This dump file was created with Neo4j version 5.3.0. The Nodes in the graph represent an Author, and the edge (undirected) shows that the two authors it connects have already co-authored a publication together. Each node or author has 224 features associated with it, which has been anonymized.

Part 1: Unveiling Patterns **(Exploring the Landscape of Author Collaborations)**

For the first phase of this task, you are tasked with performing exploratory analysis on the given graph. Include any relevant Cypher queries along with your EDA. We encourage creativity in your EDA approach, but remember to meticulously document each step. Evaluation will be based on the relevance and thoroughness of the analysis. Please elucidate your chosen methodologies and substantiate your reasons for selecting specific methods.

Part 2: Crafting Tomorrow's Chapters **(Forecasting Potential Writing Collaborations)**

For this phase, given an author (who is already present in the graph), your task is to predict the 5 most likely authors from the graph, who could co-author with the given author. You are required to use graph neural networks for this section. This section will be evaluated based on the quality of results.

Important: You must include results of at least 3 authors' recommended co-authors in your writeup. You are free to use PyTorch Geometric, Deep Graph Library, or any other library you see fit to implement your GNN.

Part 3: Cloud Chronicles

Transcending Terrestrial Limits with Deployment

This phase of the task requires you to deploy your model to the cloud. Dockerization of the model is optional. When a GET request is done to the deployed endpoint, the endpoint must return a JSON output as shown below. If your method of calculating potential coauthors does not provide likeliness, the output can contain a 1 in place of the value.

GET Request:

```
curl -X GET "<url>?id=authorID_45235_40f15_04cd1_7100c_4835e"
```

Response:


```
[{
    "authorID": "...",
    "likeliness": 0,
    "rank": 1
},
{
"authorID": "...",
"likeliness": 0,


"rank": 2 },
 {
"authorID": "...",
"likeliness": 0,
"rank": 3

} ]
```


General Guidelines:

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.