Academic Resource Recommendation System

General Introduction

> Welcome to the exciting world of ***MLOps*** hackathon "***Open Source and Dataverse*"*!*** This document aims to present the framework and objectives of this unique event, which brings together creative minds to innovate and tackle concrete challenges related to open source and data management.Organized as part of the "***Open Source & Dataverse***" event, this hackathon aims to promote the use of open source solutions for data management and analysis. It addresses a crucial need: developing tools and algorithms that facilitate data access, sharing, and analysis, while encouraging collaboration between developers and researchers. Join us to advance innovation together!

Main Objective

> Develop an innovative web application to automate the recommendation of academic resources in an optimal way using a Machine Learning clustering model. The application must also provide advanced features for resource management and statistical analysis to enhance administrators decision-making processes while ensuring the integrity of the solution.

Hackathon Schedule and Phases

> **30 heures** intensives de développement en équipes, où les participants collaboreront pour développer une solution complète répondant aux besoins spécifiés.

Part	Start Time	**Review Time**
Machine Learning - Data Preparation and Modeling & **Database Restoration**	11:00 AM in the morning, 25/11	10:00 PM in the even 25/11
Backend Development	11:00 PM in the evening, 25/11	9:00 AM in the morn 26/11
Frontend Development & Integration & Testing	9:00 AM in the morning, 26/11	4:00 PM in the aftern 26/11
Documentation & Presentation	-	4:30 PM in the aftern 26/11

Target Participants

1. Target Audience :

- Students, professionals, and enthusiasts in web development and data science.

2. Required Skills:

- **Full-stack** development.
- **Artificial Intelligence** modeling and implementation.
- **Database** management.

Problem Statement

> In schools and universities, recommending academic resources is often difficult because there is no clear understanding of department courses, required skills, gained skills, and professors expertise. This can lead to poor resource suggestions and less effective learning. A recommendation system using clustering can solve this by matching students with the right resources based on their needs and goals.

Project Added Value

- **Personalized** recommendations improve students learning experience.
- **Optimized** use of academic resources reduces waste.
- **Data insights** help administrators make better decisions.
- **Aligns** required and gained skills for academic success.

Required Work

`Part N°1:` Machine Learning (Cluster-Based recommendation system)

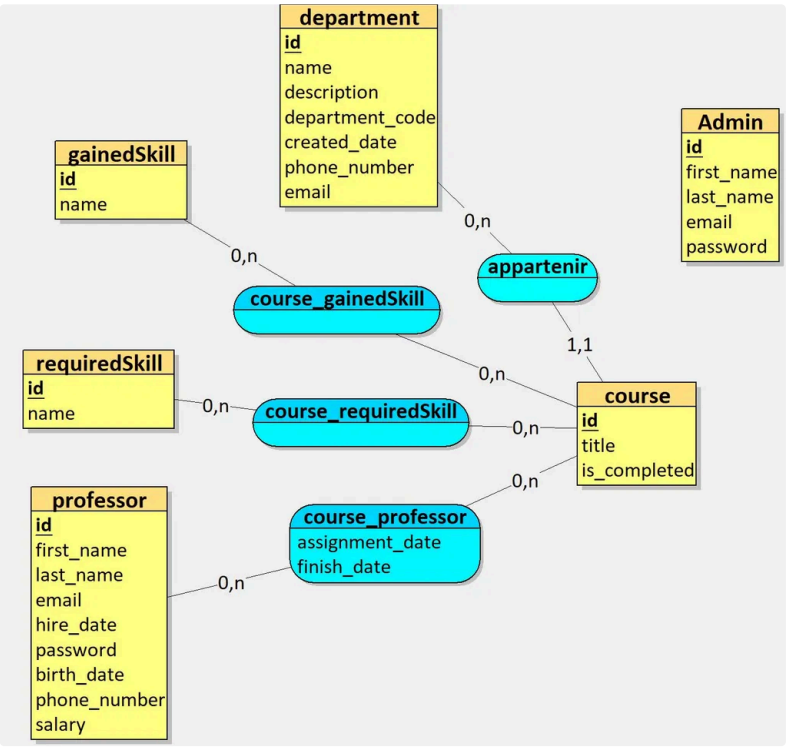
1. Data Preparation & Modeling :

> ***Objective:*** Prepare the shared dataset for optimal use in creating a Clustering-based recommendation system Machine Learning model.

- **Data Analysis and Preprocessing:**
  - Data Cleaning
  - Normalize data to ensure consistency in models.
  - Encode categorical variables
  - And other data integration and preprocessing as much as you need to be the top team
- **Feature Engineering:**
  - Identify key variables to improve model accuracy.
- **Model Training:**
  - Test multiple algorithms to get the best cluster based recommendation system.
  - Use Silhouette Score & Calinski-Harabasz Index & Davies-Bouldin Index to evaluate your model performance (*will be submitted to the leader board*).
- **Model Deployment:**
  - Export the trained model (e.g., *.pkl or .h5*)
  - Integrate an API to enable its use in the application.
- **judging criterias:**
  - You will submit your notebook (ipynb) to be evaluated by our teachers and give you a score accordingly.
  - Note that the this score will be added with the rest of the score to determine the 5 teams that will qualify to the finals

2. Database Restoration :

> We recently experienced a database incident that resulted in data loss. Fortunately, we have a complete dataset that shows the intended database structure and relationships. Your task is to use this dataset as a reference to restore the database to its proper state, ensuring all tables, relationships, and data are correctly implemented.



- For bonus points we also want to restore the completion time of each course given that a professor can't take more than 5 course (which means that the start time of the 6th course will be the end time of the first course and so as on.

Part N°2: Backend: API Development

> ***Objective:*** *Develop RESTful endpoints to enable the application to meet its functional requirements.*

1. Data Management:

> Develop RESTful endpoints for CRUD operations on courses and professors.

The expected APIs :

> *link to apis*

2. Machine Learning Prediction:

> Create an endpoint dedicated to sending course data and predicting other fields (department, professor, required skills, etc.) based on the course name, and predicting additional fields based on a specific input field.

3. Statistics (*link*) :

- Visualize the completion rate of courses by department.

  > **Graph Type**: Bar chart showing completion rate for each department.

- Number of courses offered by each department.

  > **Graph Type:** Bar chart showing the number of courses per department.

- Compare the number of required skills vs. gained skills for each course.

  > **Graph Type:** Stacked bar chart for each course showing required vs. gained skills.

- The number of courses offered by each department, broken down by year or month.

  > **Graph Type:** Line chart (graph function) showing the number of courses per department over time (year or month).

- Counts the total number of days a professor worked, with each day counted only once, even if the professor worked on multiple courses. Multiple assignments on the same day are counted as a single working day.

  > **Graph Type:** Bar Chart.

4. Security and Notifications:

- Set up token-based authentication in the login
- Encrypt the password (bcrypt) !
- Implement a notification service to alert professors when a course is assigned to them by a manager.

`Part N°3:` Frontend : Développement de l'Interface Utilisateur

1. Home Page :

- Presentation of the Platform's Objectives

2. Login Page :

- This page allows the user to authenticate.

3. The Other Pages :

a. **For the `admin`, there are three pages :**

   - Visualization of statistics with charts (pie chart, bar chart, etc ...)
   - Management of courses and professors through interactive tables (CRUD).
   - Page to test the solution (recommendations) of the modules.
   - profile admin

   > Note: When adding a course, the other fields will be automatically filled based on a recommendation system

b. **For the `professor`, there are two pages :**