Projet AI & Cognition

Conceptual graph-based Recommendation System

for PRM issues: OL approach

Knowledge Extraction

Elaborated by

Ayed Sarra nour benaouicha Yosser Kaddour

Adnene Mhiri Skander Brahim Guermaz Mouhamed Ali Rommene

Supervised By

Mr.Bilel FARAH

ABSTRACT

The goal of this project was to collect knowledge from PMI's Practice Standard for Project Risk Management and the PMBOK's Risk Management section. By utilizing natural language processing (NLP) and regular expressions, we transformed this data into a structured framework for easy extraction of essential project risk management insights.

Our approach involved analyzing text to identify concepts, properties, and relationships, creating a cohesive conceptual graph. This report outlines our methods, challenges faced, and how structured data is essential to improving decision-making and understanding in project risk management.

This project is a major step toward helping project and risk management professionals use industry standards.

TABLE OF CONTENTS

TABLE OF FIGURES

I. GENERAL CONTEXT

1. Introduction

The objective of this project was to gather, organize and structure information from two known sources, in the project management field; the Risk Management section of the 6th Edition of the Project Management Body of Knowledge (PMBOK) and the Project Management Institutes (PMI) Practice Standard for Project Risk Management. The main aim was to convert the content from these sources into a dataframe making it easier to extract important insights related to project risk management.

To accomplish this we used a combination of natural language processing (NLP) techniques and regular expressions. These methods allowed us to analyze the text and identify concepts, taxonomy, properties related to objects and data as well as various assertions like axioms, synonyms and definitions. By transforming the information into a conceptual graph our goal was to create a knowledge repository that can be easily accessed and utilized by project management professionals and researchers.

This report provides information on the methods employed during the knowledge extraction process from both PMBOK and PMI sources. It also discusses the challenges we encountered along with insights gained throughout this process. Emphasizing on how structuring information can significantly improve decision making capabilities and understanding, in the field of project risk management.

This project is a leap in utilizing the vast amount of knowledge contained in industry standards and best practices to help project managers and risk management professionals.

2. Problem description

The effective management of project risks is crucial for successful project execution. It involves identifying, analyzing, and mitigating potential risks to ensure projects are completed on time, within scope, and within budget. The Project Management Body of Knowledge (PMBOK) and the Project Management Institute's (PMI) Practice Standard for Project Risk Management are authoritative sources that provide best practices and guidelines in this area.

However, these sources are presented in an unstructured, text-based format, making it challenging for project managers, researchers, and decision-makers to access and utilize this valuable knowledge efficiently. This project aims to address this problem by creating a structured and easily navigable framework for extracting insights and actionable information from the PMBOK and PMI documents.

The key problems that need to be addressed include:

Unstructured Information: The PMBOK and PMI documents lack a clear structure for organizing concepts, definitions, and relationships related to project risk management. This makes it difficult for users to quickly locate the information they need.

Knowledge Extraction: Extracting relevant information and organizing it into a structured format is a complex task. It involves parsing the text, identifying key concepts, and establishing relationships between them using advanced natural language processing (NLP) techniques and regular expressions.

Knowledge Accessibility: Once the knowledge is extracted, it needs to be made easily searchable and retrievable. Without a structured format, users may struggle to find answers to their specific risk management questions.

Practical Utility: The ultimate goal is to create a structured dataframe that not only presents the information in a structured manner but also enhances its practical utility. This will enable project managers, risk analysts, and decision-makers to access and apply this knowledge effectively in their work.

By addressing these challenges, this project aims to make the wealth of knowledge contained within the PMBOK and PMI's Practice Standard for Project Risk Management more accessible, actionable, and valuable to professionals in the field.

3. Methodology

The methodology employed to extract knowledge from the Risk Management section of the PMBOK 6th edition and PMI's Practice Standard for Project Risk Management followed a systematic approach. It involved several steps, starting with data collection from the selected sources. Preprocessing and cleaning were performed to eliminate formatting artifacts and non-content elements. Natural Language Processing (NLP) techniques, including named entity recognition, tokenization, and part-of-speech tagging, were utilized. Information retrieval techniques, such as regular expressions, were applied to extract specific knowledge elements like concepts, definitions, synonyms, and axioms. These extracted components were then organized into a hierarchical taxonomy. Quality validation was conducted by subject matter experts to ensure accuracy. The structured knowledge was integrated into a dataframe, and a conceptual graph was constructed to visualize the relationships between concepts. Finally, data analysis and interpretation were carried out to gain insights into project risk management best practices. This methodology guarantees the accuracy and reliability of the extracted structured knowledge from the corpora.

II. DATA EXTRACTION AND PREPROCESSING

1. PMBOK

1.1. Description of the source document

The Project Management Body of Knowledge (PMBOK): is a globally recognized guide published by the Project Management Institute (PMI). It provides a comprehensive framework of project management best practices, principles, and processes. The PMBOK serves as a standard reference for project management professionals and organizations worldwide. **[1]**

For this study, the focus was on the " Risk management chapter" (Chapter 11) within the PMBOK which has a particular importance as it provides guidance on identifying, analyzing, and responding to project risks. It outlines processes, tools, and techniques for effective risk management throughout the project lifecycle. Extracting knowledge from this chapter can provide valuable insights into industry-standard best practices for managing project risks.

1.2. Data Workflow for Text Data Preparation

1.2.1. Extracting Section Titles from the PDF

In our data workflow, the first step was to extract section titles from the PDF documents. This process was essential for identifying and capturing the primary headings and subheadings within the source documents. These section titles play a crucial role in structuring and categorizing the content during the subsequent knowledge extraction phase. By extracting these titles, we were able to create a well-organized framework for further analysis and exploration of the document's content. The output of this step as shown in the figure below is a list of section titles .

1.2.1. Conversion of PDF to Text Format for Accurate Content Extraction

This conversion step is crucial for improving the accuracy of content extraction for further analysis.

1.2.2. Extraction and Structuring of Section Content

This step includes extracting content based on section titles and organizing it into a DataFrame as well as the additional step of title removal for improved data clarity as shown in figure 2 . The extracted content, aligned with the respective section titles, is structured in a tabular format for efficient analysis.

1.2.3. Data cleaning and preprocessing

Within this section, we provide a thorough explanation of the extensive data cleaning and preprocessing procedures implemented on the extracted text data. These meticulous steps are crucial in guaranteeing the utmost quality and relevance of the text, eliminating any unwanted noise, and ensuring its suitability for subsequent analysis. The subsequent data cleaning and preprocessing actions were executed:

- Correction of Misspelling and Grammatical Errors
- Stop Word Removal
- Lemmatization
- Tokenization
- (POS) Tagging and Lemmatization

The DataFrame as shown in Fig 3 resulting from these data cleaning and preprocessing steps is well-structured and ready for further analysis.

1.3. Adding new columns

Once the cleaning process is complete, we can proceed with extracting additional columns from the content. This step aims to uncover further insights and information by identifying and extracting relevant data points or features from the cleaned text.

1.3.1. Extract the “References” , “Figures/tables” column

To find and extract references, figures and tables positions as shown in the figure below regular expressions were used.

This process involved the precise detection of patterns matching each preferred section , which were then extracted for further analysis. The extraction of references , figures and table positions enhanced the clarity and structure of the extracted data, contributing to the overall success of the knowledge extraction workflow.

1.3.2. Reorder the Dataframe’s sections

In this section, we first divided the sections based on their levels ,then we reorder them according to the desired order .The reordered DataFrame was then saved as a new CSV file named "extracted_sections.csv".The reordered DataFrame was then saved as a new CSV file named "extracted_sections.csv".

1.3.3. Add the “Type “ column

We have categorized the rows in your DataFrame based on the 'title' column using a function named categorize_type.

In this function:

- If the title matches the pattern '11.x.3.x', it is categorized as 'output'.
- If the title matches the pattern '11.x.2.x', it is categorized as 'tool and techniques'.
- If the title matches the pattern '11.x.1.x', it is categorized as 'input'.
- For all other cases, the type is set to 'NAN'.

1.3.4. The summarization of the “Content” column

We defined a function named **generate_summary** that takes a text (presumably a section content) as input and generates a summary. The summarization is based on word frequency, and the function selects the top N sentences with the highest scores to form the summary.

1.3.5. Extraction of keywords

NLTK library was used to tokenize words, remove stopwords, and apply stemming to extract keywords from a given summary. The results are then added to a new 'keywords' column in a DataFrame, and the updated DataFrame is saved to a CSV file named 'extracted_sections.csv'.

1.3.6. Extraction of synonyms

We used the NLTK library to tokenize words and find synonyms for each word using WordNet. The results are then added to a new 'synonyms' column in a DataFrame, and the updated DataFrame is saved to a CSV file named 'extracted_sections.csv'.

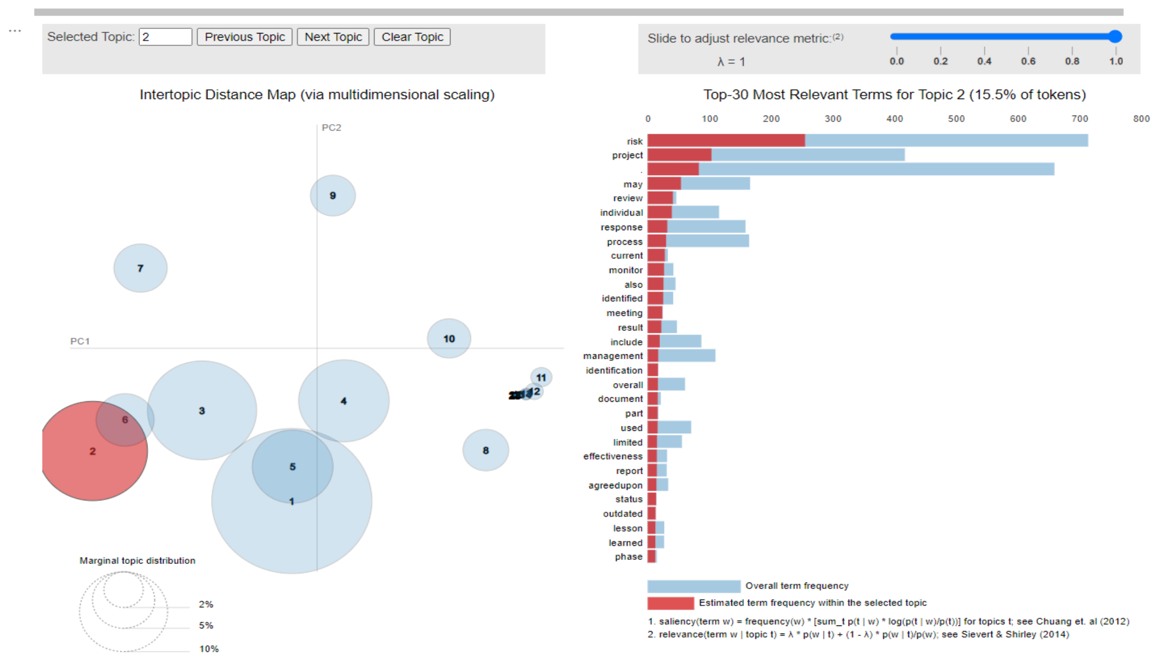
1.3.7. Extraction of topics

In this step we have proceeded by tokenizing the content into words, removing stopwords, counting word frequencies, and then extracting the most common words as topics. The results are added to a new 'topics' column in a DataFrame.

1.3.8. Visualize topics

We used the Gensim library to train an LDA (Latent Dirichlet Allocation) topic model our section contents and visualized the topics using PyLDAvis. This is a common approach for topic modeling in natural language processing. The purpose was to generate a visualization of the topics using PyLDAvis, allowing us to interactively explore the topics and their relationships.

We finally added the content topics to the DataFrame in a new column ('Content_Topics') and saved the updated DataFrame with the new column to a CSV file ('extracted_sections.csv').



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 1: the generated visualization of the topics using PyLDAvis

2. Practice Standard for Project Risk Management

2.1. Description of the source document

The *"Practice Standard for Project Risk Management"* is a publication by the Project Management Institute (PMI) that provides guidance and best practices for managing risks in project management. It is designed to help project managers and professionals understand and effectively manage risks throughout the project lifecycle. **[2]**
The PDF document consists of 9 chapters and 5 appendices. Each chapter holds its own significance, addressing various key topics, including the Risk Management Framework, Risk Identification, Risk Assessment, Risk Response Planning, Risk Monitoring and Control, and Reporting and Documentation. Additionally, the appendices provide further clarification of complex concepts and offer additional resources.

2.2. Data Workflow for Text Data Preparation

The work was divided into 3 main sections; images extraction, chapters preprocessing and text extraction, and appendices text extraction and content cleaning.

2.2.1. Images Extraction

The image extraction process started with the use of the ***PyMuPDF*** library, simplifying the extraction of embedded images from the PMI PDF file. A simple function was developed to extract every image in an organized way. These images were then saved into a designated folder for further processing and categorization. Simultaneously, the code scanned text files for figure and table titles and organized them according to content patterns.

Following this, every image in the output folder was renamed to correspond with these titles, ensuring a clear relationship between images and their titles. The code also included checks for accuracy and managed potential issues during the image-handling process. ***PyMuPDF*** for PDF parsing, ***regular expressions*** for text parsing, and basic file operations with the ***OS module*** were used, resulting in a well-structured technique for image extraction and organization from source documents.

2.2.2. Chapters preprocessing and text extraction

The PDF text extraction in this section was performed using three libraries:

- ***PyMuPDF (Fitz):*** is a popular library for working with PDF files in Python. It is used to extract text from a PDF document. PyMuPDF
- ***pdf2image*** and ***pytesseract***: These two libraries are used in combination to first convert each page of the PDF file into an image and then extract text from the images. The convert_from_path function from the pdf2image library is used to convert the PDF into images, and pytesseract is employed to extract text from the images.

The obtained text is then cleaned and stored in a txt file which will be used in the next step to extract each section alone meaning the chapters, titles, subtitles, content and figures and for this we used the regular express.

The information extracted will then be organized into a structured DataFrame using the Pandas library, in which it categorizes the text into columns for chapters, titles, subtitles, content, and figures based on specific patterns present in the text.

2.2.3. Appendices Text Extraction and Content Cleaning

This process started by cleaning the text, removing certain symbols to enhance readability. Once cleaned, this refined text was then saved into a new file. Following the cleaning phase, specialized patterns were used to identify various parts within the text, such as chapters, titles, and subtitles. These patterns were like unique markers in the text, helping to distinguish different sections effectively.

After this identification, the code sorted the text into specific groups, organizing the chapters, titles, and subtitles, along with their associated content. Additionally, the code identified and matched figures related to each section, creating a structured and organized format. The final result of this process was the formation of a well-structured DataFrame that neatly organized the diverse sections extracted from the original text.

This DataFrame is set to be merged with other existing DataFrames. The combined data will be utilized in building a conceptual graph for modeling purposes. The merging of these datasets and their subsequent application in constructing a conceptual graph will contribute to a more comprehensive understanding and representation of the information derived from the appended text.

3. Merging both dataframes

After collecting all the information pertaining to the chapters and appendices in two separate dataframes, the next step is to merge them into a single dataset based on text resemblance using cosine similarity. Subsequently, keyword extraction is performed using the pretrained Keybert model, followed by lemmatization and synonym finding for words within the content using the WordNet Lexical Database, ultimately returning the final dataframe comprising the following features: chapter, title, subtitle, content, figures, keywords and synonyms.

III. Modeling the graph

1. Extract Taxonomic Relations

We have loaded a pre-trained REBEL model, extracting taxonomic and non-taxonomic relations from text,and storing the extracted relations for further processing. This kind of approach is often used in natural language processing tasks related to information extraction and knowledge base construction from unstructured text data.

2. Extract non-Taxonomic Relations

We focused on extracting non-taxonomic features from the unstructured text data within the "Section Content" column.

The implemented code leverages the textacy and spaCy libraries to perform subject-verb-object (SVO) triple extraction, providing valuable insights into relationships within the text.

By identifying and categorizing non-taxonomic features through SVO triple extraction, we gain a deeper understanding of relationships and entities within the text data. The resulting unique DataFrame provides a clean and concise representation of these relationships, setting the stage for further analysis and integration into the broader knowledge graph.

3. Construct the Graph

We used NetworkX and Matplotlib libraries to visualize a directed graph based on a sampled subset of triplets to represent relationships between entities. This step includes multiple purposes

- Stemming and Concept Association: Enhancing text processing by stemming verbs and associating them with corresponding objects, providing a more normalized representation of processes.
- Fuzzy Matching and Type Relation Assignment: Associating types and concepts based on fuzzy matching, enabling the assignment of relevant type relations to entities in the main dataset.
- Graph Visualization: Visualizing a sampled subset of triplets in a directed graph to gain a visual understanding of relationships between entities in the dataset.

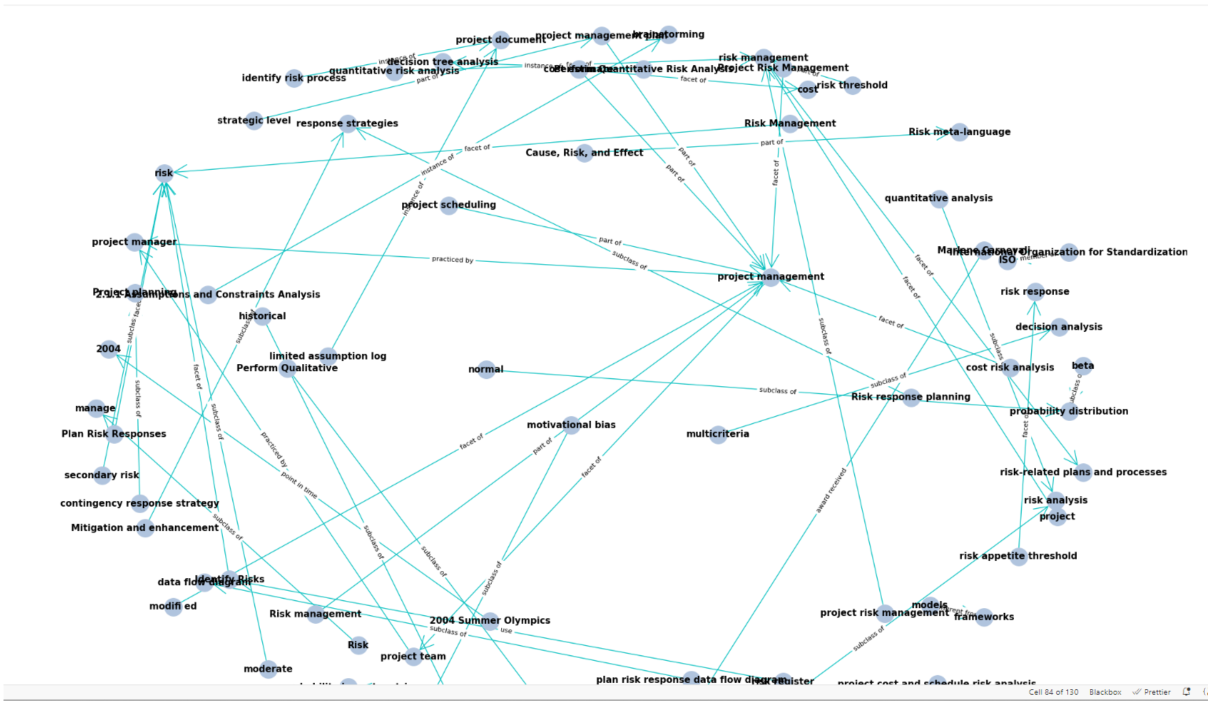


Figure 2: Sample relational graph

4. Feature engineering

Feature engineering in the context of graph-based machine learning involves assigning meaningful features to nodes in a graph.

- Conversion to Sentences:The graph is converted into a list of sentences for Word2Vec, where each sentence is represented by a node and its neighbors.
- Word2Vec Model Training:The Word2Vec model is trained on the list of sentences, generating vector embeddings for each node.
- Node Embeddings:Node embeddings are obtained from the trained Word2Vec model.

5. Graph Neural Network Architecture

Training the Graph Neural Network (GNN) model is a vital step for building our recommendation system . During training, the model acquires the ability to comprehend the underlying patterns and relationships within the graph structure. This involves optimizing the model parameters to make accurate predictions based on node features and edge connections.

5.1. Learning Graph Representations

The GNN model aims to capture intricate relationships and dependencies between nodes in the graph. By processing information through Graph Convolutional Network (GCN) layers, the model generates meaningful representations for each node, iteratively.

5.2. Node Classification

Node classification is often the primary objective, where the model predicts the class or label of each node. This is useful in various scenarios, such as categorizing documents, predicting protein functions, or classifying nodes in a social network.

5.3. Loss Function and Optimization

The choice of the cross-entropy loss function aligns with accurately classifying nodes. It quantifies the difference between predicted and ground truth labels, guiding the optimization process.

5.4. Backpropagation and Parameter Update

Through backpropagation, the model adjusts its internal parameters to minimize the loss. This iterative process refines the model's understanding of the graph structure, enabling it to generalize and make accurate predictions on unseen data.

5.5. Monitoring Training Progress

The training loop includes monitoring the loss during each epoch, providing insights into the model's convergence and learning effectiveness. Recorded loss values serve as diagnostic tools for debugging and hyperparameter tuning.

5.6. Adaptive Learning (Adam Optimizer)

The Adam optimizer adaptively adjusts learning rates for each parameter. This optimization method enhances efficiency, particularly in scenarios with varying gradients and complex loss landscapes.

5.7. Iterative Learning

The model undergoes multiple training epochs, allowing it to iteratively refine its understanding of the graph. This iterative process ensures the model captures both local and global graph patterns, leading to improved generalization performance.

IV. The implementation of the knowledge based Recommendation system

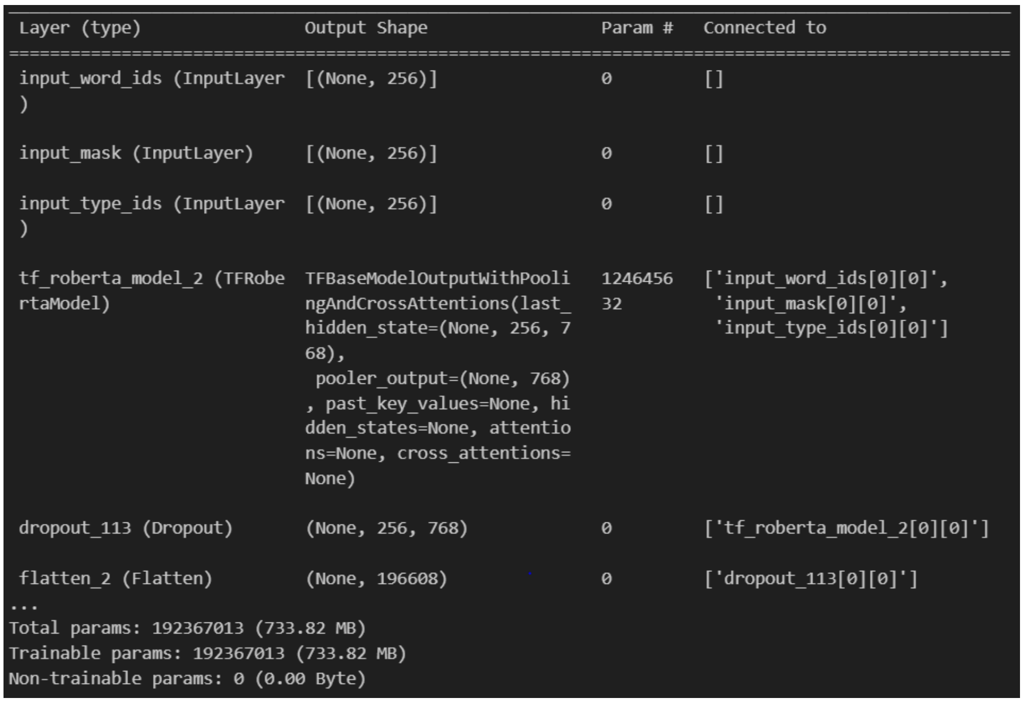
1. Data preprocessing

In this section, we prepared the data for processing before the modeling stage. This section encompassed the following steps:

- dropping unnecessary columns and separating the text data and target labels.
- Tokenization: The RoBERTa tokenizer is used (RobertaTokenizer) to tokenize and encode the text data, converting it into numerical inputs that the model can understand.
- Label Encoding: Target labels (Concepts) are converted into numerical format for model training.
- Train-Test Split using Scikit-learn library.

2. Implementation of the model

Using TensorFlow's Keras API, a neural network architecture is constructed. The base of this architecture is the RoBERTa model (TFRobertaModel). Additional layers are added on top to adapt the RoBERTa model for the specific task of classification. These added layers help in learning patterns and features relevant to the classification problem.



```
 Layer (type)                Output Shape                 Param #   Connected to
==================================================================================================
 input_word_ids (InputLayer  [(None, 256)]                0         []
 )

 input_mask (InputLayer)     [(None, 256)]                0         []

 input_type_ids (InputLayer  [(None, 256)]                0         []
 )

 tf_roberta_model_2 (TFRobe  TFBaseModelOutputWithPooli   1246456   ['input_word_ids[0][0]',
 rtaModel)                   ngAndCrossAttentions(last_   32         'input_mask[0][0]',
                             hidden_state=(None, 256, 7              'input_type_ids[0][0]']
                             68),
                              pooler_output=(None, 768)
                             , past_key_values=None, hi
                             dden_states=None, attentio
                             ns=None, cross_attentions=
                             None)

 dropout_113 (Dropout)       (None, 256, 768)             0         ['tf_roberta_model_2[0][0]']

 flatten_2 (Flatten)         (None, 196608)               0         ['dropout_113[0][0]']
...
Total params: 192367013 (733.82 MB)
Trainable params: 192367013 (733.82 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 3: Added Layers

3. Evaluation

The following figure depicts the accuracy evolution across epochs for both the training and validation sets, providing insights into the model's performance and potential overfitting.

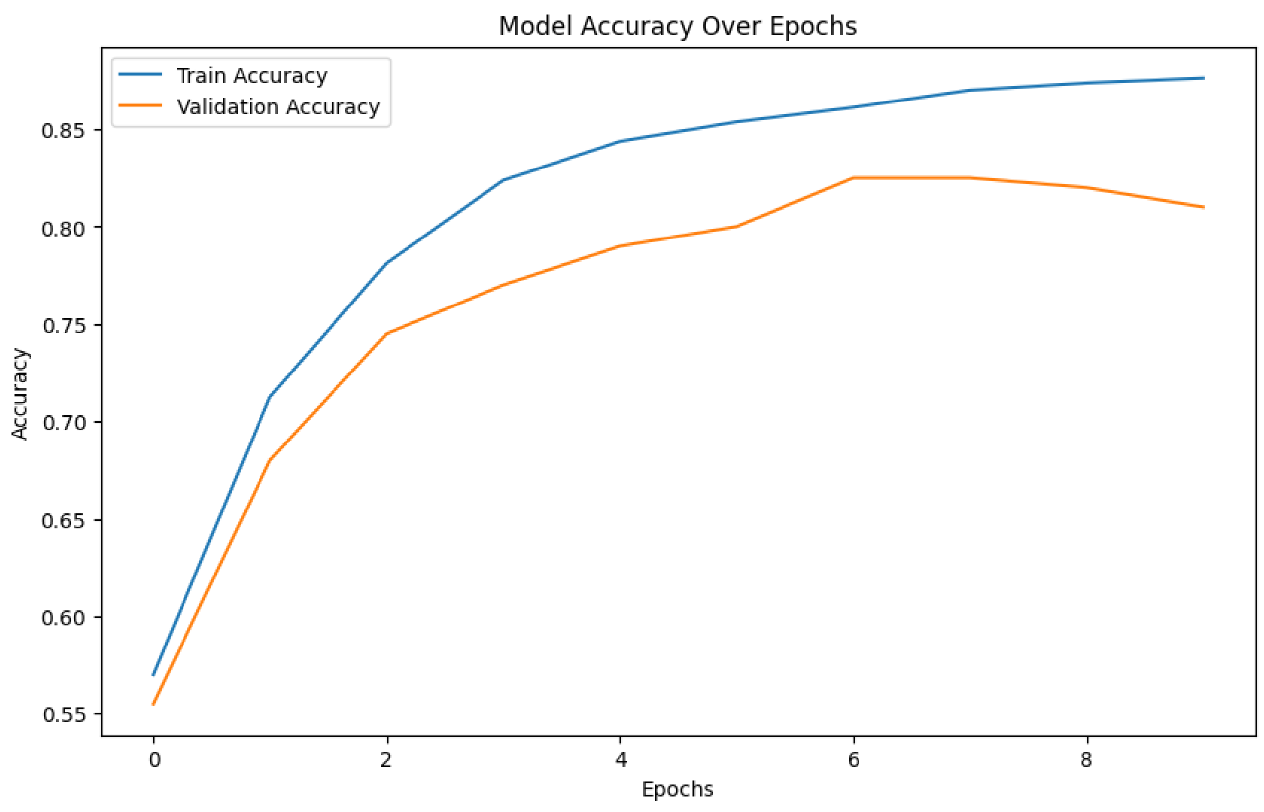


Figure 4: Accuracy Evolution: Training VS Validation Sets

The chosen evaluation metric, accuracy, measures the model's correctness in predicting outcomes. In this context, it reached a commendable level of 81%, signifying the percentage of correct predictions made by the model against the total predictions evaluated.

V. DEPLOYMENT

1. Model Integration using Pickle

1.1. Model Training and Pickling

Prior to deployment, we train our GNN model using appropriate data and feature engineering techniques. Once the model is trained, we serialize it using Python's `pickle` module. Pickling allows us to save the model's state to a file, preserving the trained weights and parameters.

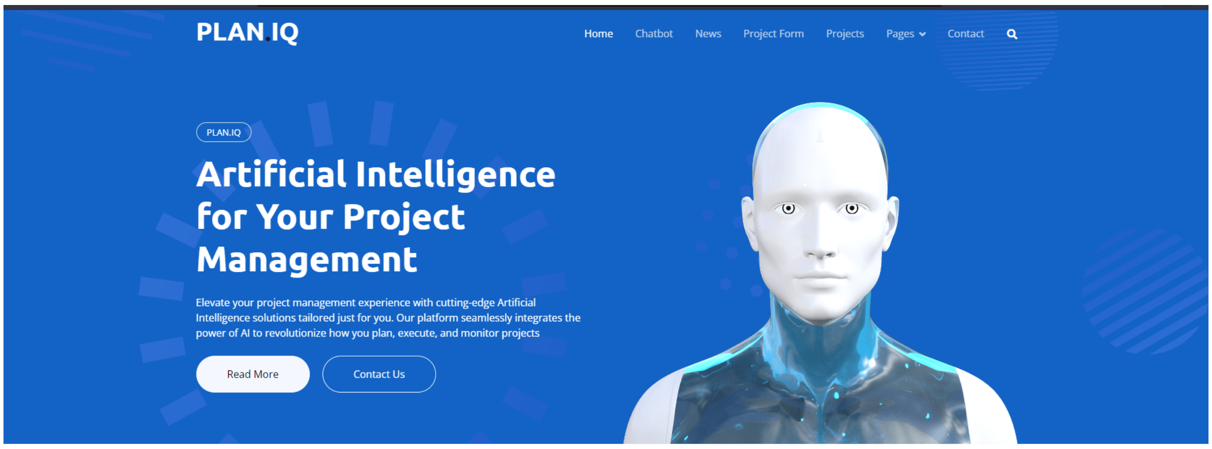
1.2. Loading the Pickled Model

In the Flask application, we load the pickled model during startup to ensure its availability for Our system recommendation . We use the `pickle` module to deserialize the model file and store it in memory.

2. Web Application Visualization

2.1. Flask Web Framework

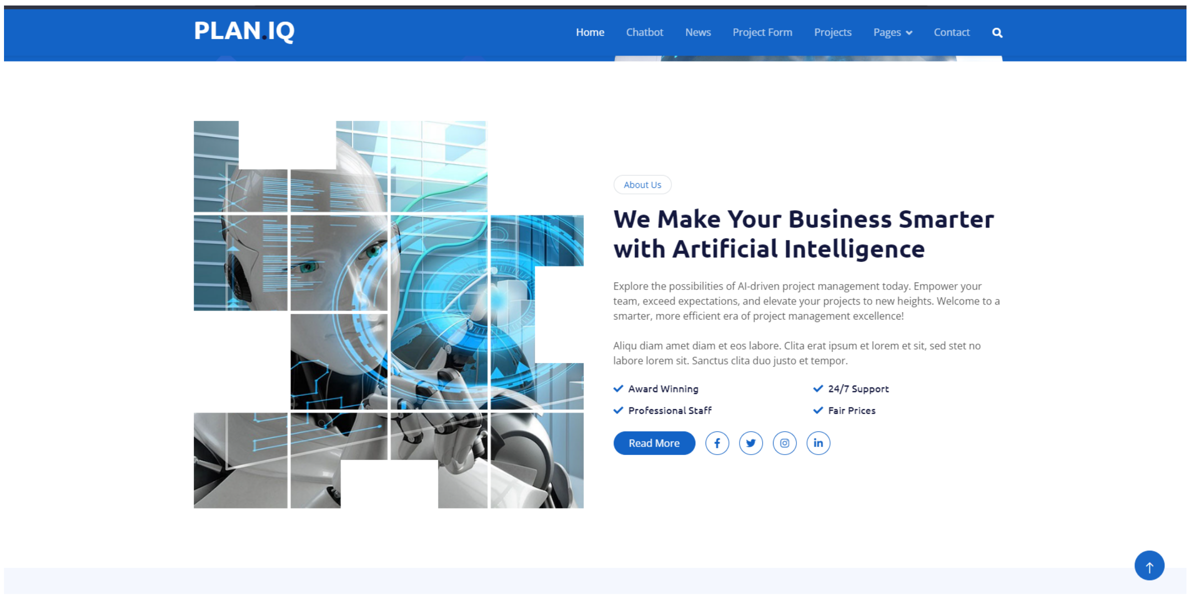
Flask is a lightweight web framework in Python that allows us to build web applications quickly and easily. We use Flask to create routes and define the behavior of our web application. **[3]**



Artificial Intelligence for Your Project Management

Elevate your project management experience with cutting-edge Artificial Intelligence solutions tailored just for you. Our platform seamlessly integrates the power of AI to revolutionize how you plan, execute, and monitor projects

Figure 5: Home Page (Part 1)



We Make Your Business Smarter with Artificial Intelligence

Explore the possibilities of AI-driven project management today. Empower your team, exceed expectations, and elevate your projects to new heights. Welcome to a smarter, more efficient era of project management excellence!

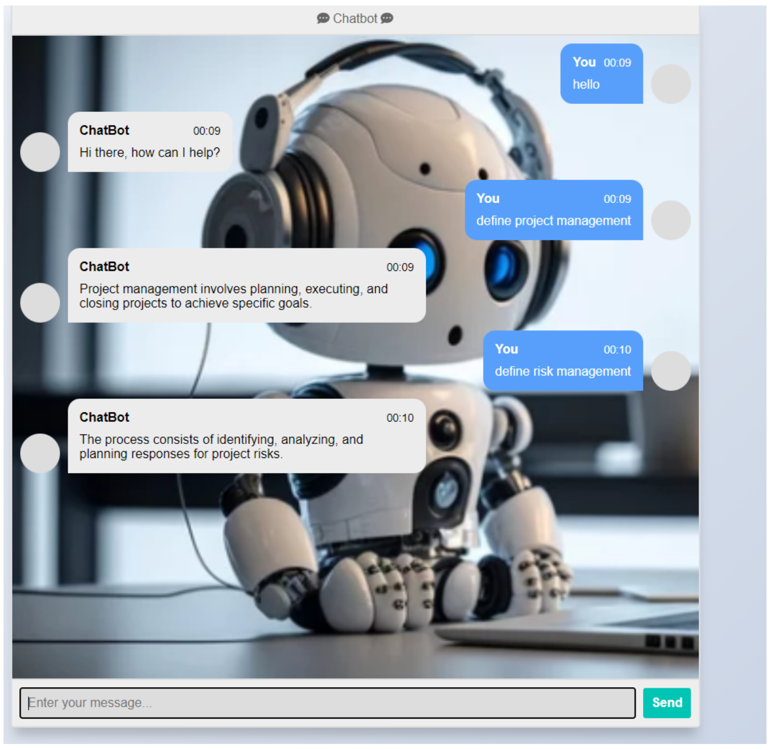
Aliqu diam amet diam et eos labore. Clita erat ipsum et lorem et sit, sed stet no labore lorem sit. Sanctus clita duo justo et tempor.

✓ Award Winning
✓ 24/7 Support
✓ Professional Staff
✓ Fair Prices

Figure 6: Home Page (Part 2)

The home page offers a snapshot of the site's purpose through navigation, introductory content, and visuals to engage and direct visitors.

2.2. Chatbot



You 00:09
hello

ChatBot 00:09
Hi there, how can I help?

You 00:09
define project management

ChatBot 00:09
Project management involves planning, executing, and closing projects to achieve specific goals.

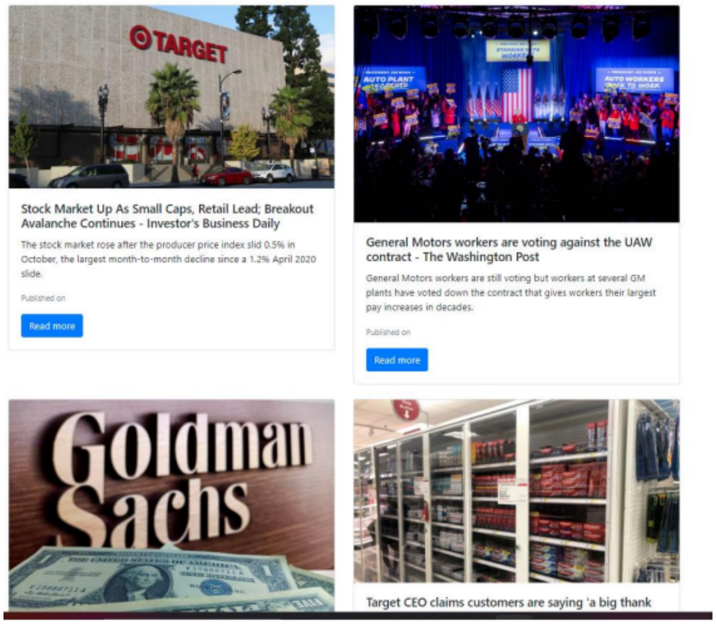
You 00:10
define risk management

ChatBot 00:10
The process consists of identifying, analyzing, and planning responses for project risks.

Figure 7: Chatbot Page

The chatbot page is designed to respond to user inquiries specifically related to project risk management, offering guidance and information on the subject.

2.3. Breaking News


Stock Market Up As Small Caps, Retail Lead; Breakout Avalanche Continues - Investor's Business Daily

The stock market rose after the producer price index slid 0.5% in October, the largest month-to-month decline since a 1.2% April 2020 slide.

Published on

Read more

General Motors workers are voting against the UAW contract - The Washington Post

General Motors workers are still voting but workers at several GM plants have voted down the contract that gives workers their largest pay increases in decades.

Published on

Read more

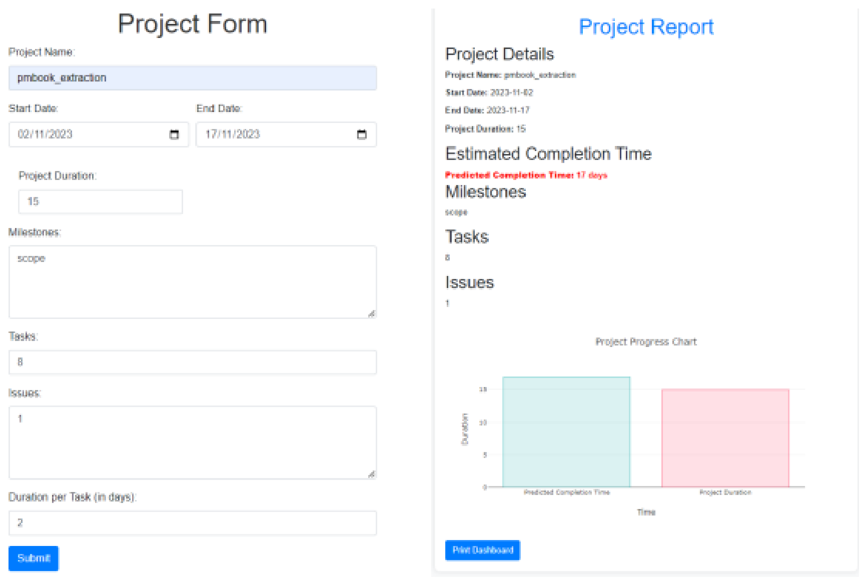
Target CEO claims customers are saying 'a big thank

Figure 8: News Page

The news page is a dynamic section that swiftly delivers the latest and most significant updates across various topics, keeping users informed in real-time.

2.4. Project report

2.4.1. Delay


Project Form

Project Name: pmbook_extraction

Start Date: 02/11/2023

End Date: 17/11/2023

Project Duration: 15

Milestones: scope

Tasks: 8

Issues: 1

Duration per Task (in days): 2

Submit

Project Report

Project Details

Project Name: pmbook_extraction

Start Date: 2023-11-02

End Date: 2023-11-17

Project Duration: 15

Estimated Completion Time

Predicted Completion Time: 17 days

Milestones

scope

Tasks

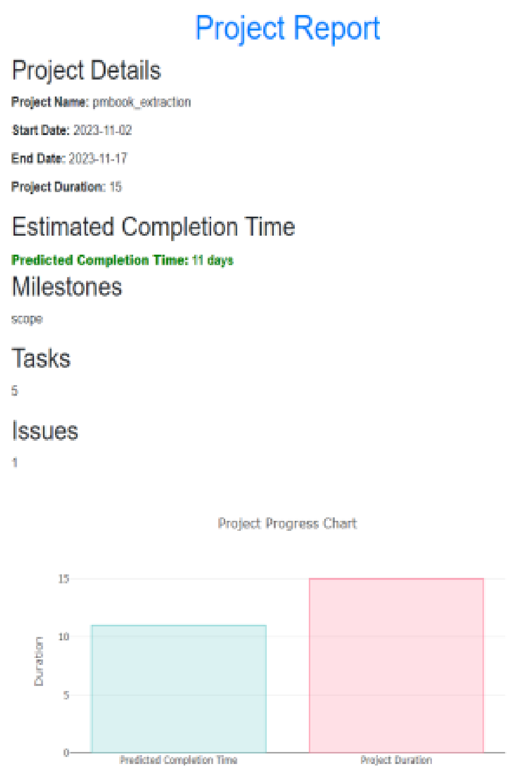
8

Issues

1

Figure 9: Project Report (Case 1)

2.4.2. On Time



Project Report

Project Details

Project Name: pmbook_extraction

Start Date: 2023-11-02

End Date: 2023-11-17

Project Duration: 15

Estimated Completion Time

Predicted Completion Time: 11 days

Milestones

scope

Tasks

5

Issues

1

Figure 10: Project Management Report (Case 2)

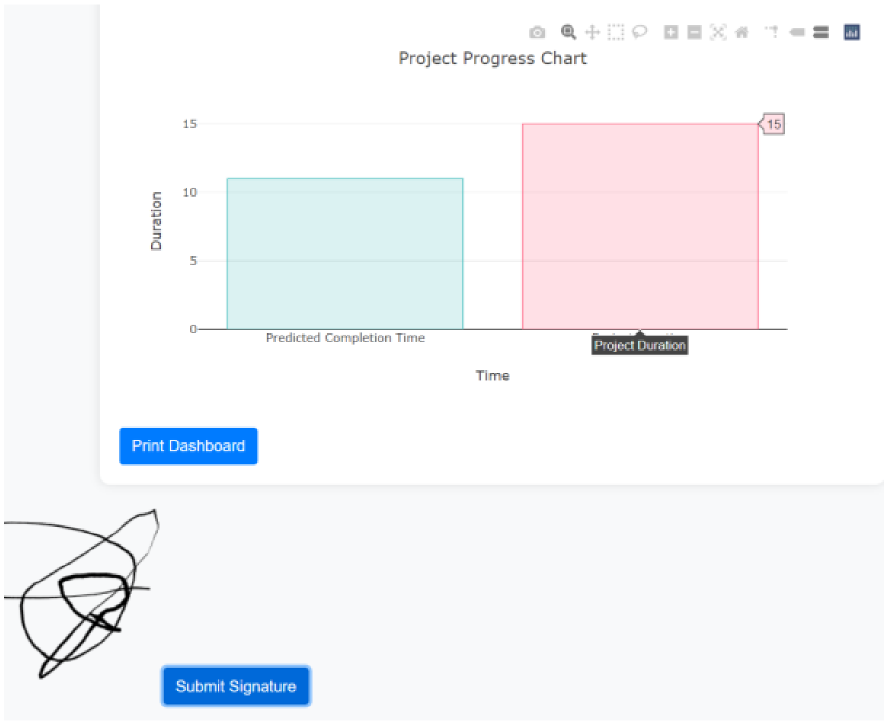
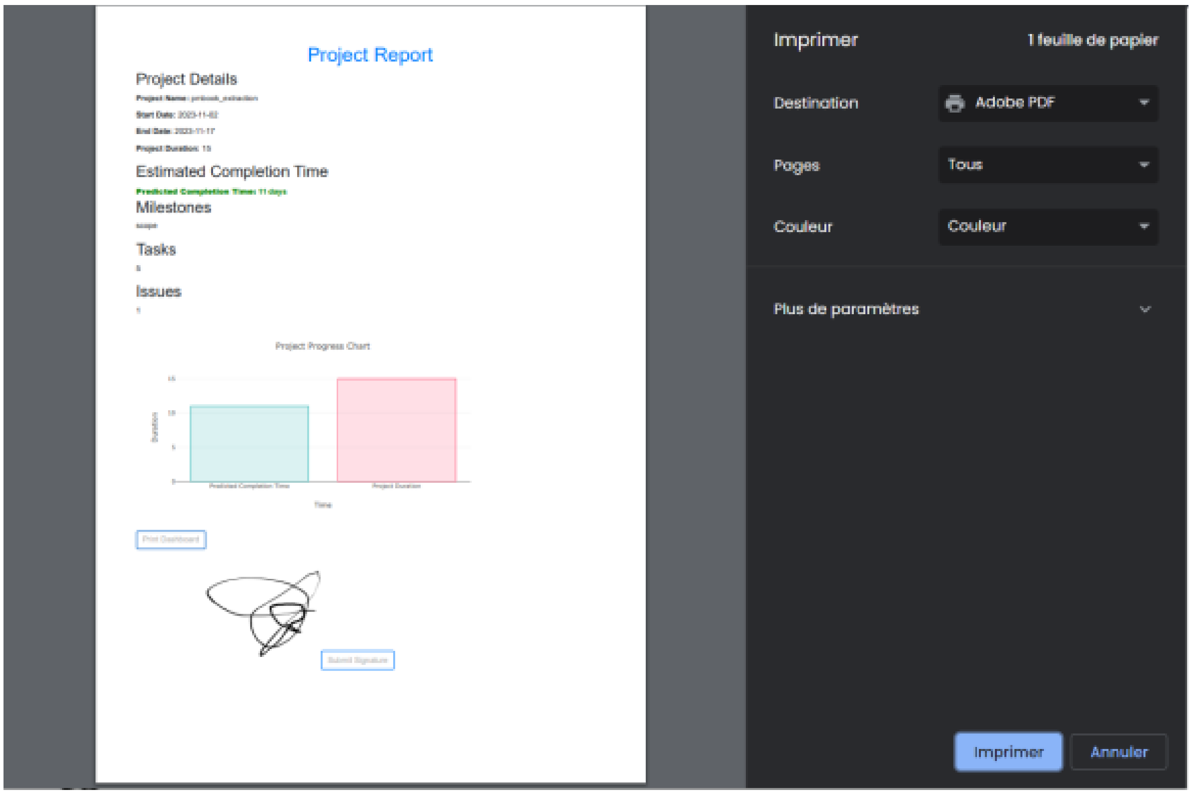


Figure 11: Project Progress Chart


Project Report

Project Details

Project Name: pmbook_extraction

Start Date: 2023-11-02

End Date: 2023-11-17

Project Duration: 15

Estimated Completion Time

Predicted Completion Time: 11 days

Milestones

scope

Tasks

5

Issues

1

Imprimer 1 feuille de papier

Destination: Adobe PDF

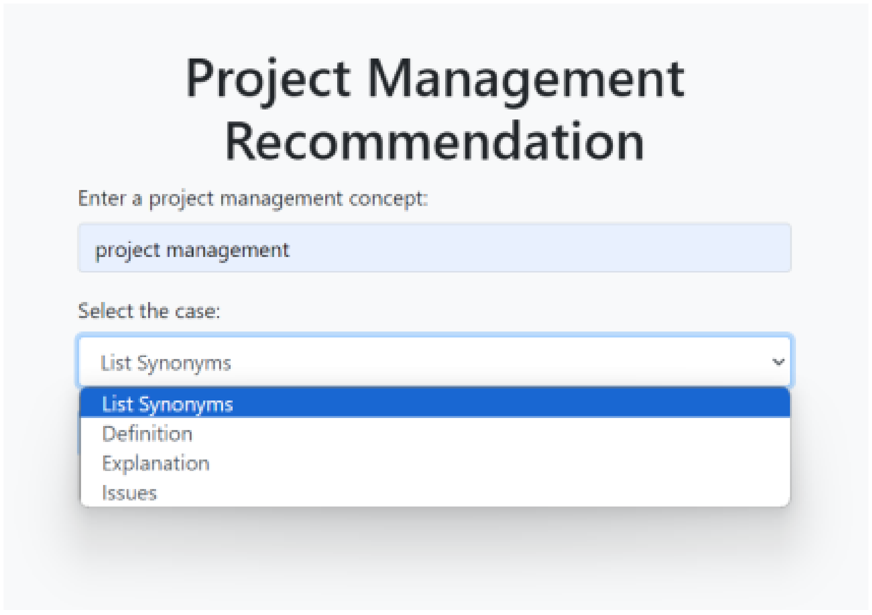
Pages: Tous

Couleur: Couleur

Figure 12: PRM Report Print

The application includes a function to generate project reports, offering two outcomes—indicating if the project is on schedule or issuing a delay warning—alongside the capability to print the report as necessary.

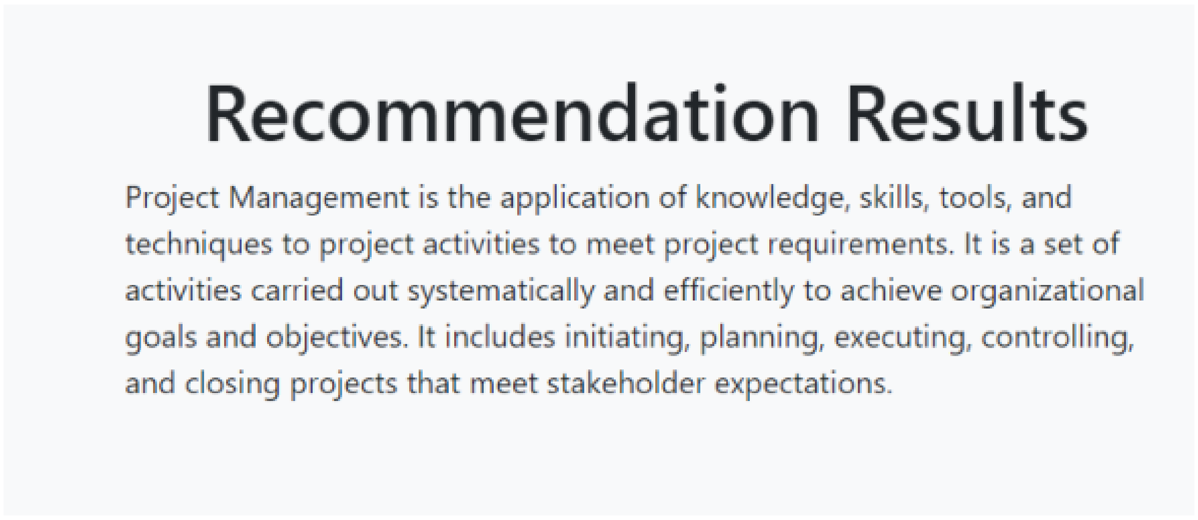
2.4. Recommendation System


Project Management Recommendation

Enter a project management concept: project management

Select the case: List Synonyms

Figure 13: Recommendation System Request

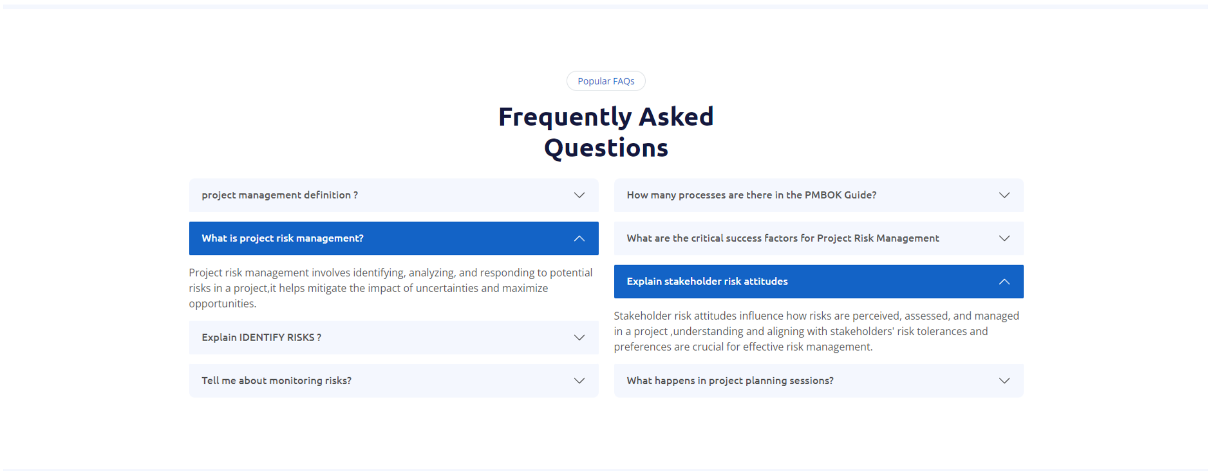


Recommendation Results

Project Management is the application of knowledge, skills, tools, and techniques to project activities to meet project requirements. It is a set of activities carried out systematically and efficiently to achieve organizational goals and objectives. It includes initiating, planning, executing, controlling, and closing projects that meet stakeholder expectations.

Figure 14: Recommendation System Output

The recommendation system page allows users to input a concept and select their preferred type of information—synonyms, definitions, explanations, or related issues—prompting the system to provide tailored answers catering to their specific request.



Popular FAQs

Frequently Asked Questions

project management definition ?

What is project risk management?

Project risk management involves identifying, analyzing, and responding to potential risks in a project,it helps mitigate the impact of uncertainties and maximize opportunities.

Explain IDENTIFY RISKS ?

Tell me about monitoring risks?

How many processes are there in the PMBOK Guide?

What are the critical success factors for Project Risk Management

Explain stakeholder risk attitudes

Stakeholder risk attitudes influence how risks are perceived, assessed, and managed in a project ,understanding and aligning with stakeholders' risk tolerances and preferences are crucial for effective risk management.

What happens in project planning sessions?

Figure 15: Frequently Asked Questions (FAQs)

On the FAQs page, users can find the most commonly asked questions along with their respective answers.

VI. Conclusion

This project's completion represents an important achievement in the field of project risk management. Through meticulous extraction and structuring of information from industry standards, we have developed a full framework implementing a recommendation system and a chatbot. Our approach, detailed within this report, delineates the different steps from data analysis to the creation of a cohesive conceptual graph, highlighting its essential function in enabling the efficient extraction of crucial insights.

The web interface showcases various facets of our structured data framework. From the Home Page's multifaceted presentation to interfaces like the Chatbot Page, News Page, and diverse project report displays, each element represents the result of our efforts. Additionally, the Recommendation System, Project Progress Chart, and FAQ section illustrate the versatility of our structured data in helping professionals in their project and risk management inquiries.

The project is a great example of advancement, offering a practical, user-friendly platform for project and risk management professionals to make use of the knowledge derived from industry standards. Its deployment signifies not just the completion of a comprehensive framework but also a significant stride towards empowering professionals with structured and actionable information for informed decision-making in project risk management.

VII. References

[1] PMBOK
https://www.pmi.org/pmbok-guide-standards/foundational/pmbok

[2] Practice Standard for Project Risk Management
https://www.pmi.org/pmbok-guide-standards/foundational/risk-management

[3] Flask web Framework
https://flask.palletsprojects.com/en/3.0.x/