iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** XII **Month of publication:** December 2024

DOI: https://doi.org/10.22214/ijraset.2024.65935

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue XII Dec 2024- Available at www.ijraset.com

The ProjectValt: A Repository for the Student's Projects Reports

Abhishek Pandey[1], Aruna Mrutyunjay Badiger[2], Atul Rathore[3], Dr. Bama S[4]
[1, 2, 3]*Student, Dept. of Computer Science and Engineering, Sri Venkateshwara College of Engineering, Bengaluru – 562157*
[4]*Associate Professor, Dept. Of Computer Science and Engineering, Sri Venkateshwara College of Engineering, Bengaluru – 562157*

Abstract: ***Academic institutions amass a wealth of knowledge through student projects, yet this valuable resource often remains untapped. ProjectValt addresses this by creating a centralized platform that empowers users to efficiently discover, understand, and leverage past student work.***
At the core of ProjectValt lies an intelligent summarization tool. Utilizing advanced natural language processing, this tool automatically generates concise summaries of project reports, enabling users to quickly grasp key findings and insights. Furthermore, ProjectValt incorporates sophisticated search capabilities, including keyword matching and semantic similarity analysis. This allows users to effectively search for projects based on specific criteria, fostering a culture of knowledge sharing and reuse within the academic community.
By providing a user-friendly interface and powerful tools, ProjectValt unlocks the potential of past student projects. This platform not only preserves valuable research but also inspires future innovation by enabling students, faculty, and researchers to build upon the work of their predecessors.
Keywords: ProjectValt, Academic Repository, Summarization, Chatbot, Knowledge Sharing, Digital Archiving, Project Management, Research Access, Web Platform, Machine learning, Natural languages processing (NLP), User-friendly interface. Transformers, LLM

I. INTRODUCTION

In educational institutions, managing the wealth of projects completed each year can be overwhelming. These projects are often packed with valuable insights, innovative ideas, and research that can guide future work. However, finding a way to organize and easily access these projects is a challenge many academic institutes face. That's where "ProjectValt" comes in—a platform specifically designed to simplify the collection, organization, and retrieval of past academic projects.

ProjectValt's primary goal is to make these past projects more accessible and usable for students, faculty, and researchers. One standout feature is its summarizer, which takes lengthy project reports and boils them down to brief, clear summaries. This means that instead of wading through pages of content, users can quickly get to the heart of a project—understanding its main goals, methods, and results in just a few lines. It's perfect for students who need a quick overview for inspiration, literature reviews, or to understand the scope of previous work in their field.

Adding to this, ProjectValt also has a chatbot that acts like a virtual guide. Users can ask the chatbot specific questions about the projects in the database, and it will help them find the right information instantly. This makes navigating through a vast collection of academic projects easier and more engaging, turning what could be a time-consuming search into a simple conversation.

ProjectValt is all about making past academic projects a living resource rather than letting them gather dust. It encourages a culture of shared knowledge within academic institutes, helping students and educators tap into a treasure trove of ideas and research. By focusing on academic needs, ProjectValt transforms how institutes preserve and use past projects, making it a valuable tool for learning, inspiration, and continuous academic growth.

.

II. PROPOSED SYSTEM

The proposed system for ProjectValt is a specialized project management system with a user-friendly web interface platform designed to upload all the project reports into the website under the projects section to help academic institutions effectively manage their past student projects. At the heart of the system is a centralized database that archive completed projects, capturing essential details such as project titles, domain, team member details, and the full project documentation. This organized repository with the specific domain section aims to make the retrieval of project information straightforward and easy for students and faculty, allowing them to search according to their interests and quickly access relevant resources according to their needs.

©IJRASET: All Rights are Reserved

One of the standout features of ProjectValt is its summarizer.one is with transformers and another by Using the Map reduce technique, the summarizer can condense lengthy project reports into short, easy-to-read summaries. This feature allows users to quickly understand the main summary of a project without having to sift through the entire document. It's a practical solution for students and researchers who need to gather insights quickly.

In addition to the summarizer, ProjectValt includes an interactive chatbot that serves as a virtual assistant. The chatbot is designed to guide users about the project. This makes exploring the database feel more like a natural conversation, providing a more intuitive way to navigate and discover information. This comprehensive approach makes ProjectValt a valuable resource for academic institutions, encouraging easy access to a wealth of project knowledge and fostering an environment of continuous learning and collaboration.

III.METHODOLOGY

This project is implemented by using various algorithms and techniques, here we will see a brief explanation.

First, we will see the overall project architecture diagram:

A. Diagram

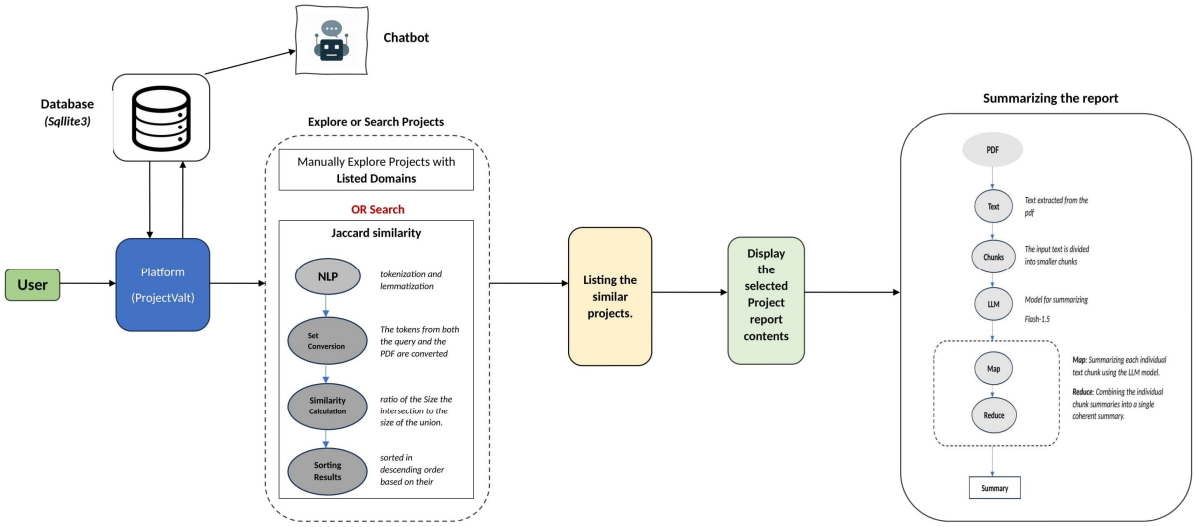


Figure.1 Overall architecture diagram

Hera, the above diagram shows the overall workflow of the ProjectValt.

1) Data source: The data source is a database stored in SQLite. This database contains information about various research projects.

2) Project Exploration: The user can manually explore the projects based on listed domains. This allows for a focused search based on specific research areas.

3) OR Search Functionality: If the user wants to find similar projects based on a specific query, they can use the "OR Search" functionality. This functionality uses Jaccard Similarity to compare the query with the projects in the database.

4) Jaccard Similarity Calculation: As you can see the **first step** is NLP (i.e., natural language processing) which works is to pre-process the query and pdf content by tokenization and lemmatization.

The second step is to make the set conversion of the pdf tokens and the query tokens.

The query and each project's report are turned into sets of unique words.

Example: - Your query set: *{"project," "management," "system," "college"}*

A stored project's set: *{"project," "management," "system," "university"}*

The third step Jaccard Similarity checks how similar the sets are by comparing common words. It calculates the ratio of shared words (intersection) to the total unique words (union).

Intersection (common words): *{"project," "management," "system"}*

Union (all unique words): *{"project," "management," "system," "college," "university"}*

Jaccard Similarity = *(Size of Intersection) / (Size of Union)*

Example: *3 (shared words) / 5 (total words) =0.6.*
The fourth step is to Sorting Results: All projects are sorted based on their similarity score (how close they are to your query). Projects with higher similarity scores (*closer to 1*) appear at the top of the results list.

5) Listing Similar Projects: After calculating similarity for all projects, ProjectValt will show a list of projects that are most similar to your search, sorted from highest to lowest similarity. If a project report has 0.8 similarity, it means it's highly related to your query, and it will appear before projects with lower scores like 0.4.
6) Displaying Project Report Contents: When you select a project from this sorted list, ProjectValt shows you the detailed content of that report.
7) Summarizing the Report: Now, suppose the report is long and you want a quick summary,
The Django view designed to enable users to access pdf document, from which it extracts text and generates a summary using a pre-trained transformer model called BART from Hugging Face's Transformers library. Initially, the necessary modules are imported, including Django's rendering tools and the BART model and tokenizer, which are loaded using the model's name ***facebook/bart-large-cnn***, specifically fine-tuned for summarization tasks.
The function e*xtract_text_from_pdf*(pdffile) utilizes the *pdfplumber* library to read the content of the accessed PDF, extracting text from each page and concatenating it into a single string. In the main function, *summarize_pdf(*request), the application checks for POST requests containing a PDF file. If a file is uploaded, the extracted text is then tokenized using the BART tokenizer, ensuring it adheres to the model's input requirements, and the model generates a summary based on the tokenized input.
Several parameters guide this generation process, such as maximum and minimum lengths for the summary and settings for beam search. The resulting summary is decoded into a readable format, with each sentence presented as a bullet point for clarity. Finally, the formatted summary is rendered in the 'uploads.html' template, allowing users to view the summarized content of their PDF. If no PDF file is uploaded or the request method is not POST, the function simply renders the template without a summary, ensuring the application remains user-friendly and responsive. This implementation effectively combines file handling, text extraction, and advanced natural language processing to provide a seamless experience for summarizing PDF documents.

B. Additional Summarizer

Flow Diagram

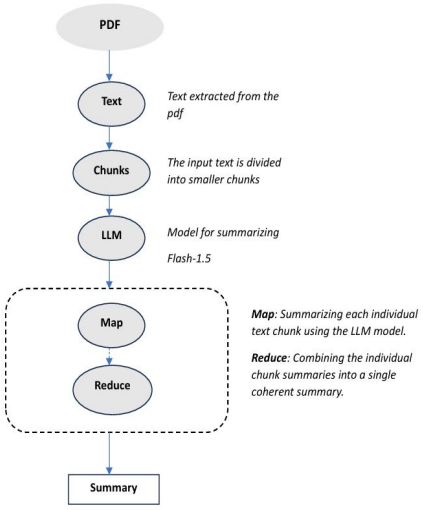


Figure.2 Summarizer Flow diagram

The first step is to Extracting Text from PDF: The selected project report is in PDF format. ProjectValt extracts the text from it.
And the second step is to Chunking: If the report is too long, the text is broken into smaller, manageable pieces (chunks) to make summarizing easier.
The next step is Summarizing with LLM (Large Language Model): ProjectValt uses a model like *Flash-1.5* to summarize each chunk separately. This gives a concise summary for each part.

C. We use MapReduce Method

Map: Each chunk gets summarized individually.

Reduce: The platform combines these small summaries into one comprehensive summary for the entire project report.

At the end of this process, you have a concise, readable summary of a lengthy project report. This allows you to quickly understand what the project is about without reading the entire report.

IV. IMPLEMENTATION

An SQLite database is created to store information about research projects. This might include project titles, keywords, abstracts, and potentially full reports. A user interface element, such as a dropdown menu or search bar, allows users to explore projects based on pre-defined domains. Here's how it could work:

The UI retrieves a list of available domains from the database. Upon selecting a domain, the UI queries the database for projects tagged with that domain and displays them for the user.

Integrate an NLP library like NLTK or spaCy to handle pre-processing tasks. It performs:

1) Tokenization: Breaking down the user query and project report text into individual words (tokens).

2) Lemmatization: Converting words to their base form (e.g., "running" becomes "run").

Convert both the pre-processed user query and project report content into sets of unique words. Python libraries like set can be used. Implement a function using the Jaccard Similarity formula. It takes the intersection (common words) and union (total unique words) of the query and project sets as inputs and calculates the similarity score. Libraries like SciPy can be helpful for calculations.

Utilize a sorting algorithm or library (e.g., Python's sorted function) to arrange all projects based on their Jaccard similarity score, with the most similar projects appearing first. Consider setting a minimum similarity score *(e.g., 0.7)* to filter out projects with low similarity and only display highly relevant ones. Utilize a PDF text extraction library like PyPDF2 to convert the selected project report (initially in PDF format) into plain text. Chunking**:** If the report is lengthy, implement a chunking algorithm or library to divide the extracted text into smaller, manageable pieces for easier summarization. Python's textwrap module can be used for chunking. Integrated an transformers model or API for a Large Language Model (LLM) like Flash-1.5. Send each text chunk to the LLM for summarization, resulting in a concise summary for each portion. Utilize a MapReduce framework like Apache Spark or a simpler implementation in Python. The final summary, should be presented to the user in a clear and readable format within the UI.

This implementation section provides a detailed breakdown of how ProjectValt's functionalities were translated into a working system using various tools and libraries.

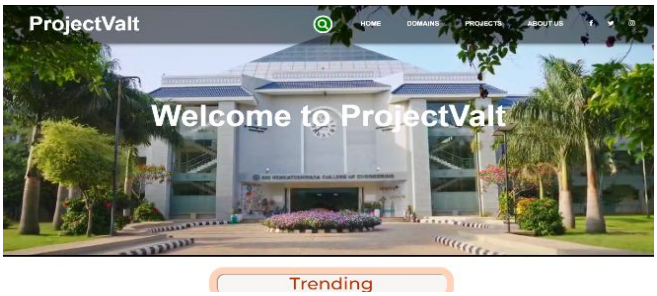
V. ADDITIONAL FEATURE

A. Chatbot

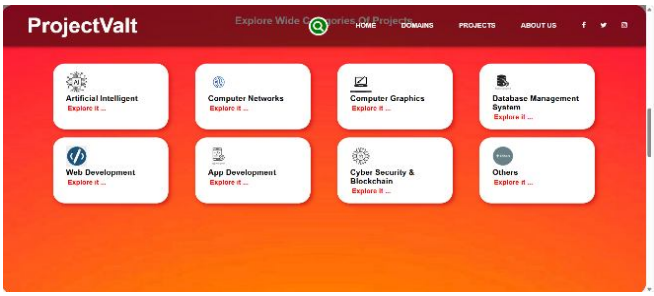
We have also integrated a chatbot that help very useful for user interaction and doubts. The is designed in a way that the user can ask the question regarding the projects available in the database, then the chatbot can efficiently understand the user query and based in the pdf reports it answers', if the user query cannot process through the database content, the chatbot gives the answer using the llm or transformers.

VI. RESULTS

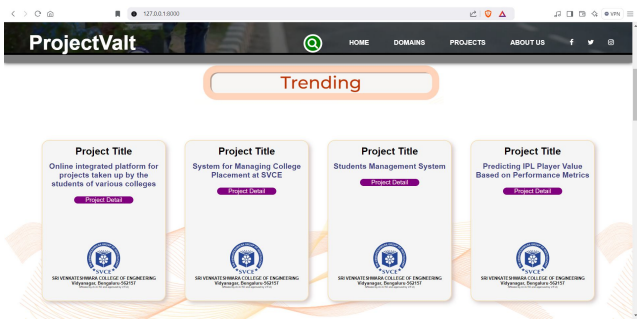
The ProjectValt system successfully implemented the proposed methodology, demonstrating its effectiveness in retrieving relevant existing projects details. The NLP techniques accurately processed and compared text data, leading to precise Jaccard Similarity calculations. The user interface provided an intuitive way to explore and search projects. The LLM-based summarization effectively condensed lengthy reports into concise summaries. While the system showed promising results, further improvements could be made in terms of expanding the database, refining NLP techniques, and exploring advanced summarization methods.



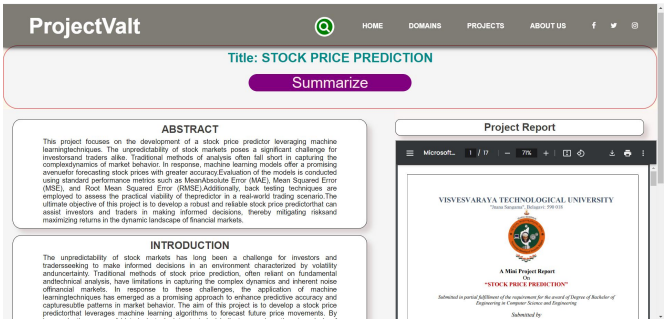
home page



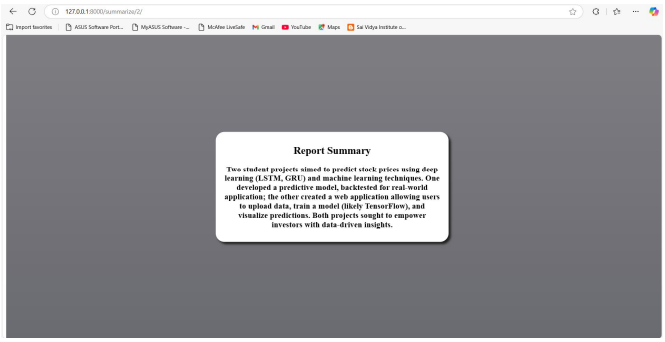
Available domains



Projects page



Detailed view page



Summary page

Figure.3 Result Screen Shots

VII. CONCLUSION

ProjectValt emerges as a powerful tool for academic institutions, addressing the challenge of managing and accessing past projects details. By implementing innovative features like the summarizer and chatbot, the platform significantly enhances the discoverability and usability of this valuable knowledge repository. The summarizer effectively reduces information overload, enabling users to quickly grasp the essence of lengthy reports. The chatbot, acting as a virtual guide, streamlines the search process, making it more efficient and interactive.

By empowering students, faculty, and researchers to easily explore and leverage past projects, ProjectValt fosters a culture of knowledge sharing. This platform not only preserves valuable insights but also inspires future students. As academic institutions continue to generate a wealth of knowledge, ProjectValt stands as a vital tool to unlock its potential and drive innovation.

REFERENCES

[1] Niwattanakul, Suphakit & Singthongchai, Jatsada & Naenudorn, Ekkachai & Wanapu, Supachanun. (2013). Using of Jaccard Coefficient for Keywords Similarity.

[2] Dhanavandan, S., & Tamizhchelvan, M. (2013). Development of Shodhganga repository for electronic theses and dissertations in Tamil Nadu: A study. International Research: Journal of Library and Information Science, 3(4).

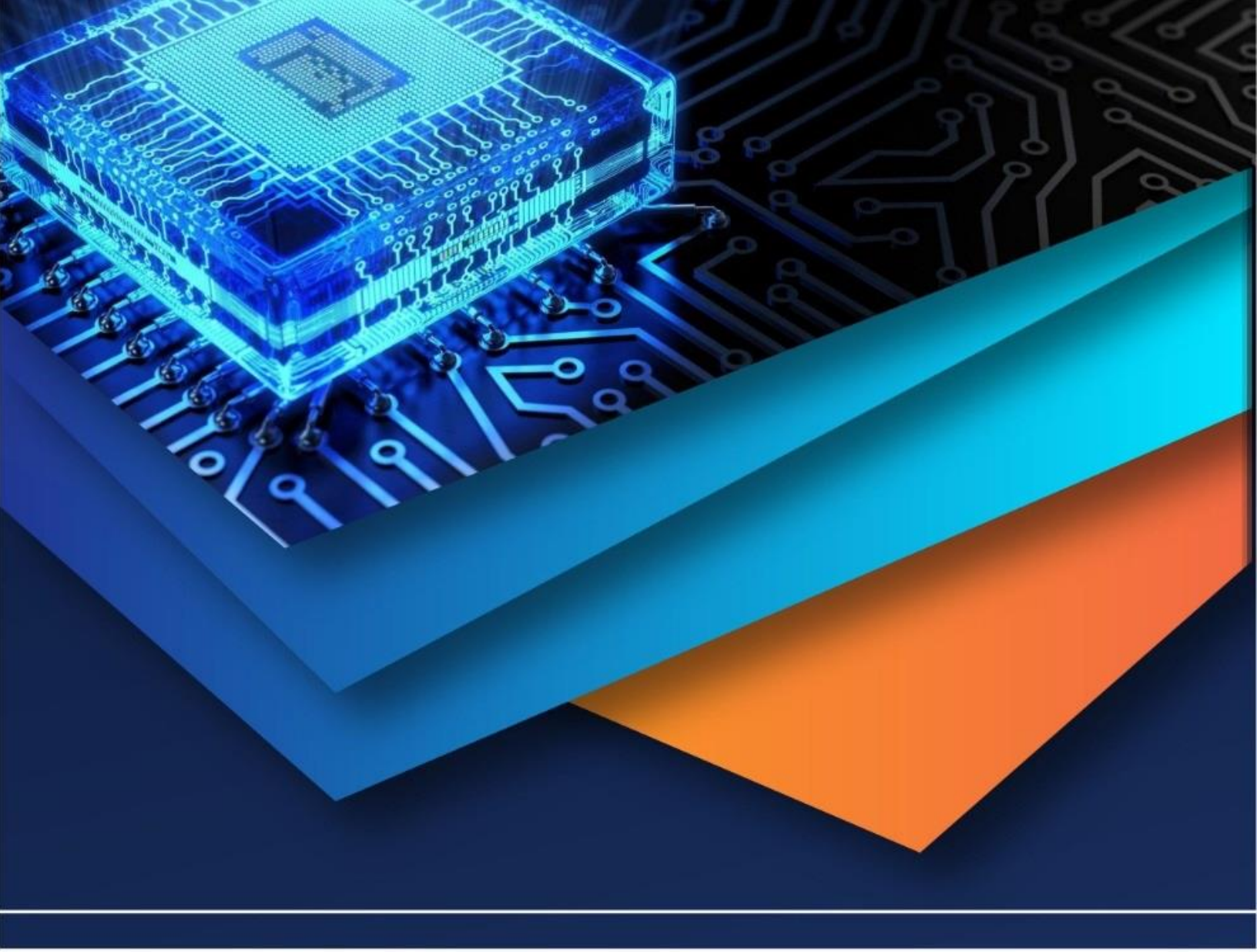
[3] Ramakrishnan, C., Patnia, A., Hovy, E., & Burns, G. A. (2012). Layout-aware text extraction from full-text PDF of scientific articles. Source code for biology and medicine, 7, 1-10.

[4]

[5] Bento, S., Pereira, L., Gonçalves, R., Dias, Á., & Costa, R. L. D. (2022). Artificial intelligence in project management: systematic literature review. International Journal of Technology Intelligence and Planning, 13(2), 143-163.

[6] Arbel, I., Refael, Y., & Lindenbaum, O. (2024). TransformLLM: Adapting Large Language Models via LLM-Transformed Reading Comprehension Text. arXiv preprint arXiv:2410.21479.

[7] Forcier, J., Bissex, P., & Chun, W. J. (2008). Python web development with Django. Addison-Wesley Professional.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)