HICON AI: HIGHER EDUCATION COUNSELING BOT

A MAJOR PROJECT REPORT

Submitted By

ARJUN DEV SINGLA (RA2011031010074)

SHASHANK TRIPATHI (RA2011031010096)

Under the guidance of

Dr. Helen Victoria A.

Assistant Professor, Dept. of Networking and Communications

In partial fulfilment of the requirements for the degree of
BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING
with a specialization in INFORMATION TECHNOLOGY

FACULTY OF ENGINEERING AND

TECHNOLOGY SRM INSTITUTE OF SCIENCE

AND TECHNOLOGY

Kattankulathur, Chengalpattu District

MAY 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR–603 203

BONAFIDE CERTIFICATE

Certified that 18CSP109L major project report titled "**HICON AI: Higher Education Counseling Bot**" is the bonafide work of **ARJUN DEV SINGLA [RegNo:RA2011031010074]** and **SHASHANK TRIPATHI [RegNo:RA2011031010096]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other thesis or dissertation based on which a degree or award was conferred on an earlier occasion for this or any other candidate.

Dr. Helen Victoria A.
SUPERVISOR
Assistant Professor
Dept. of Networking and Communications

Dr. Annnapurani Panaiyappan
HEAD OF THE DEPARTMENT
Professor and Head
Dept. of Networking and Communications

Signature of Examiner I

Signature of Examiner II

Department of Networking and Communications SRM

Institute of Science and Technology

Own Work Declaration Form

Degree/ Course	: B.Tech. in Computer Science and Engineering with Specialization in Information Technology
Student Names	: Arjun Dev Singla, Shashank Tripathi
Registration Numbers	: RA2011031010074, RA2011031010096
Title of Work	: HICON AI: Higher Education Counseling Bot

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

Arjun Dev Singla RA2011031010074	Shashank Tripathi RA2011031010096

Date:

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We would like to express our warmth of gratitude to our Registrar **Dr. S. Ponnusamy**, for his encouragement.

We wish to thank **Dr. T.V. Gopal**, Professor & Dean, College of Engineering and Technology, SRM Institute of Science and Technology, for his valuable support and encouragement.

We would like to express our heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project. We extend my gratitude to **Dr. Annapurani Panaiyappan,** Professor and Head, Department of Network and Communications.

We wish to express our sincere thanks to Panel Head **Dr. Supraja P**, Associate Professor, Department of Networking and Communications, and Panel Member **Dr. Saranya G,** Assistant Professor, Department of Networking and Communications.

We register our immeasurable thanks to our Faculty Advisor, **Dr. M. Safa,** Assistant Professor, Department of Networking and Communications, School of Computing, for leading and helping us to complete our course.

We are highly thankful to our Project Guide, **Dr. Helen Victoria A.,** Assistant Professor, Department of Networking and Communications for his assistance, timely suggestion, and for providing us an opportunity to pursue our project under his mentorship. She provided us the freedom and support to explore the research topics of our interest. Her passion for solving real problems and making a difference in the world has always been inspiring.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, we thank the almighty for showering his blessings on us to complete our Course project.

Arjun Dev Singla (RA2011031010074)

Shashank Tripathi (RA2011031010096)

ABSTRACT

As more and more students are seeking counseling services to help them pursue higher education opportunities abroad, it has become apparent that there is a lack of automation and a monopolization of counseling agencies, which presents a challenge for these students. To address this issue, we have developed **HICON AI: Higher Education Counselor**. This innovative application offers personalized college selection and preparation recommendations to students. By asking a set of defined questions, the bot gets to know the user and provides tailored guidance based on the information provided, utilizing refined Machine Learning models and Retrieval Augmented Generation. We have specifically used Llama 2 LLM by meta for our use case, because of its excellent performance and financial viability. Our new product builds on the success of our previous work, "**COUNSELOR AI: Revolutionizing Counseling System**" and ensures the highest level of accuracy and reliability.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

NLP	Natural Language Processing
LLM	Large Language Models
AI	Artificial Intelligence
ML	Machine Learning
AISHE	All India Survey on Higher Education
GER	Gross Enrollment Ratio
GPT	Generative pretrained Transformer
RAG	Retrieval Augmented Generation
DF	Data Frame
PD	Pandas
SNS	Seaborn
OCR	Optical Character Recognition

CHAPTER-1

INTRODUCTION ON HICON AI CHATBOT

1.1 ABOUT PROJECT PROBLEM

In the rapidly evolving landscape of contemporary education, where higher education options are plentiful and diverse, students embarking on their educational journey encounter a multitude of inquiries. They ponder over which academic discipline aligns most closely with their passions and aspirations, which university offers the most comprehensive courses and avenues for personal growth, and how to navigate the challenging application and admissions procedures. Above all, they seek assurance that their investment in a college education will pave the way for a fulfilling and prosperous career. The role of a higher education counselor is to address these questions and provide tailored guidance to students. According to recent statistics from the All India Survey on Higher Education (AISHE), the Gross Enrollment Ratio (GER) for higher education in India recorded a figure of 27.1 percent in the academic year 2020-21. To address each student's queries and to provide easy, accessible, and personalized guidance sessions for higher studies, the idea of 'HICON AI' is vital.

1.1.1 HIGHER EDUCTION COUNSELING SYSTEM

Counseling is essential for assisting students in making well-informed decisions regarding their academic and professional paths in the areas of education and career development. The counseling system successfully attends to the various needs of students by utilizing both human interventions and AI-powered solutions. Because of its sophisticated algorithms and machine learning powers, artificial intelligence has completely changed counseling in higher education. Artificial intelligence (AI)-powered platforms examine enormous volumes of data to provide students with tailored recommendations that take into consideration their interests, career goals, and academic standing. These tools enable students to make well-informed decisions by recommending appropriate courses, colleges, and career options. Additionally, chatbots driven by AI provide immediate support by directly addressing issues and responding to inquiries, improving students' counseling experiences all around. Conversely, conventional counseling continues to be an essential part of the university system. Human counselors, like the one shown in Figure 1, have the emotional intelligence and sympathetic comprehension required to establish a deeper connection with students. They can provide individualized advice based on each person's goals and difficulties. In-depth conversations regarding a student's goals, shortcomings, and strengths are made possible by manual counseling sessions, which empower counselors to offer mentorship and advice that is more nuanced. Human counselors can also provide emotional support, which helps to reduce the fears and uncertainties that are frequently associated with making the move to a higher education. A comprehensive and all-encompassing higher education counseling system is created by combining AI-driven recommendations with the human touch of manual counseling. This guarantees that students receive individualized guidance that successfully meets both their academic and emotional needs.

Fig1. Manual Counseling System

1.1.2 INTRODUCTION TO HICON AI

Introducing **'HICON AI'** a dedicated chatbot leveraging **Large-scale Language Models (LLM)** to assist students in navigating higher education. Harnessing **Natural Language Processing (NLP)**, it serves as a virtual counselor, analyzing student data and inquiries to deliver personalized advice efficiently. With features like **audio prompting** and **effective class categorization**, 'HICON AI' enhances user interaction and tailors' guidance according to individual needs, thanks to its comprehensive training datasets. Additionally, the implementation of **Retrieval Augmented Generation (RAG)** ensures the relevance and quality of generated data, while the **Resume-Screener** feature streamlines the input process for users.

Designed with a **user-friendly interface**, 'HICON AI' stands as a reliable solution for personalized guidance in higher education, fostering informed decision-making and empowerment among students. In conclusion, 'HICON AI' emerges as an asset, facilitating accessible and individualized support for students, thereby contributing to a more capable and informed student body.

1.1.3 LARGE LANGUAGE MODELS

LLMs, an Artificial Intelligence (AI) program undergo extensive training on diverse datasets enabling it to interpret inputs and produce textual outputs, as illustrated in figure 2. A pivotal aspect of the chatbot model is its capability for student-oriented guidance, characterized by a category-wise segmentation of students. This segmentation ensures that assistance provided is tailored to individual needs, avoiding a one-size-fits-all approach. The creation of comprehensive datasets facilitated class categorization, resulting in prompts that are both generic and relevant. Students are classified into **four** distinct categories:

Category 1 comprises individuals with minimal knowledge of the process; **Category 2** includes those who are just beginning to explore the field and may possess some relevant experiences like projects or internships; **Category 3** encompasses individuals with the requisite knowledge but requiring additional enhancements in their profiles; **Category 4** represents students with complete profiles seeking further insights into the process.

Following categorization, LLMs generate responses based on **10-15 prompts** specific to each category, ensuring structured and contextually appropriate guidance tailored to the individual's classification.

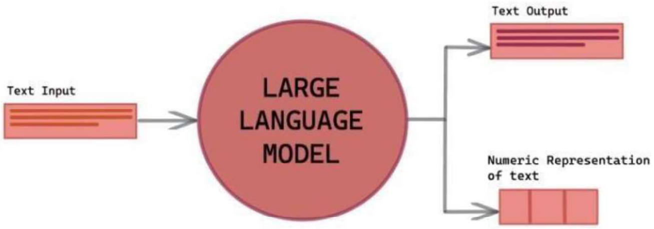


Fig2. Large Language Model

1.1.4 RETRIEVAL AUGMENTED GENERATION

Retrieval-Augmented Generation (RAG) is an approach in **Natural Language Processing** (NLP) that combines retrieval-based methods with generative models to generate coherent and contextually relevant responses. In simpler terms, RAG uses a large database of information to find relevant facts and then uses that information to create new, meaningful responses to user queries, as illustrated in figure 3. This integration allows for more accurate and diverse content generation, improving the quality of AI-driven interactions in various applications such as question answering systems and conversational agents.

Retrieval-Augmented Generation represents a paradigm shift in how counseling content is generated and disseminated. At its essence, this approach integrates two key methodologies: retrieval-based techniques and generative models. By fusing these methodologies, **HICON AI** transcends traditional counseling systems, offering students a dynamic and comprehensive resource that is both informative and original.

Retrieval-based methods form the foundational pillar of HICON AI's approach. Through an extensive repository of scholarly articles, educational resources, and validated information sources, the system ensures access to a wealth of relevant and credible material. This reservoir of knowledge serves as the backbone for addressing student inquiries, providing a solid grounding for the guidance offered. Leveraging retrieval techniques, HICON AI navigates this repository with precision, retrieving pertinent information in response to student queries.

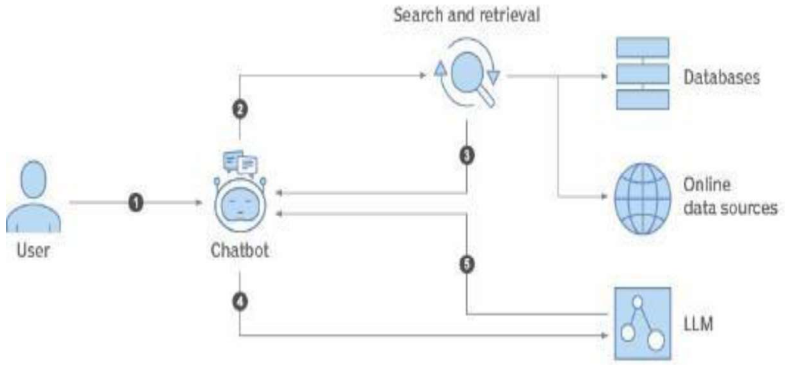


Fig3. Retrieval Augmented Generation

1.2 OBJECTIVE

"HICON AI" is crucial to the education sector because it helps many students through a critical point in their education, especially those who are pursuing higher education. This AI-based counselor is intended to provide fair, individualized, and easily accessible counseling sessions to students 24/7, answering their questions and dispelling any doubts they may have about pursuing higher education. The principal aim of 'HICON AI' is to offer customized counsel and dispel uncertainties for students during this critical juncture in their professional lives.

'HICON AI' places a lot of emphasis on classifying students according to the data they submit. This classification guarantees that the support given is unique to every student, in contrast to the general approach of most AI-based counselors. The four classifications created by 'Counselor AI' serve people at different phases of their pursuit of higher education. Those in Category 1 are those who have little experience and are just learning about the field, whereas those in Category 2 are those who are just starting out and may have completed some projects or internships. Group 3 consists of people who know enough but could still do better in some areas, while Group 4 is made up of people who know enough but are looking for new information to keep current.

The training model for student categorization makes use of a self-created dataset to guarantee that the prompts are brief and encompass all the higher studies material that the students require. This method improves "HICON AI's" ability to provide accurate and pertinent advice that is suited to each student's needs.

Integrating cutting-edge technologies like Retrieval Augmented Generation (RAG) is crucial to HICON AI's functionality. RAG combines natural language generation with information retrieval to provide quick data retrieval and responses that are appropriate for the given context. This guarantees that, in the ever-changing context of higher education, the counsel given to students stays relevant and accurate. HICON AI's overall efficiency is further enhanced using Large-scale Language Models (LLMs) with supercomputer capabilities that have been trained on vast datasets to comprehend user inputs and produce complex, domain-specific responses.

Additionally, HICON AI has a resume screener feature, which takes it beyond academic guidance. This application expedites the resume evaluation process by applying sophisticated algorithms to pinpoint the most important experiences, education, and training that relate to the user's desired career path. HICON AI's usefulness is increased by the resume screener, which helps students create compelling resumes that are relevant to their chosen careers or fields of study.

Using a Large-scale Language Model (LLM), which is well-known in the field of higher education and imitates communication between a real human counselor and students seeking higher education, is another important goal of the "HICON AI." It processes and generates automated responses by analyzing inputs in the form of student data and queries using Natural Language Processing (NLP). Following classification, LLMs will produce answers for each category based on eight to ten prompts. These answers will be well-organized and fully contextualized to the questioner and the class.

Lastly, one of the other main goals of the "HICON AI" is the voice-based dialogue between the model and the students, which makes it appealing to many students, especially those with disabilities. The speech inputs are transformed into text and fed to the model for processing. Once the model has finished processing the text responses, the text responses are transformed back into speech, improving the seamless experience for the students.

1.3 MOTIVATION

The increase of popularity for higher studies among students has been evident. To cater the huge number of students having queries regarding the higher education and those who are looking for a counseling advice, HICON AI would be of great help rather than going for a manual higher education counselor.

HICON AI, designed specifically for higher education, addresses the need for a dedicated chatbot in this domain, offering students personalized assistance. In contrast to conventional chatbots that often provide generic responses, HICON AI categorizes students based on their input, generating specific prompts for tailored advice on course selection, university choices, and other relevant aspects. To enhance the user experience, it incorporates features such as speech-to-text and text-to-speech, accommodating students with disabilities.

An essential component of HICON AI's functionality is the integration of advanced technologies like Retrieval Augmented Generation (RAG). RAG combines information retrieval with natural language generation, enabling swift data retrieval and the generation of contextually appropriate responses. This ensures that the advice provided to students remains current and accurate in the dynamic landscape of higher education. Additionally, HICON AI leverages Large-scale Language Models (LLMs) with supercomputer capabilities, trained on extensive datasets, to understand user inputs and generate nuanced, domain-specific responses, contributing to its overall efficiency.

Moreover, HICON AI goes beyond academic guidance by incorporating a resume screener feature. This tool streamlines the process of evaluating resumes, utilizing advanced algorithms to identify key qualifications, skills, and experiences relevant to the user's career goals. The resume screener enhances HICON AI's utility by assisting students in preparing impactful resumes tailored to their chosen field of study or profession.

1.4 PROBLEM STATEMENT

In the field of higher education counseling, students often encounter challenges in accessing personalized, timely, and plagiarism-free guidance to navigate their academic journey effectively. Existing counseling systems often lack the sophistication to provide tailored support while maintaining academic integrity. Additionally, the sheer volume of information available can overwhelm students, making it difficult to discern reliable sources from less credible ones. This leads to a gap in the provision of comprehensive and trustworthy assistance for students seeking academic advice. Therefore, there is a pressing need for a solution that leverages advanced technologies to deliver personalized counseling, grounded in authoritative sources, while ensuring the originality and authenticity of the content provided. The HICON AI project aims to address these challenges by developing a state-of-the-art platform that utilizes Retrieval-Augmented Generation techniques to offer students dynamic, plagiarism-free counseling tailored to their individual needs and inquiries.

1.5 EXISTING CHALLENGES

There are several existing challenges present in the field of higher education counseling system that are addressed by HICON AI, these are:

- **Credibility of Output Generated:** Fairness and credibility of the output generated pose significant challenges. While Retrieval-Augmented Generation (RAG) enhances response accuracy, ensuring fairness in the selection and presentation of information is crucial to avoid biases or discrimination. Additionally, maintaining the credibility of the generated content necessitates continuous verification and updating of sources to uphold the integrity of the counseling provided.

- **Lack of Structured Dataset:** A significant obstacle in counseling and support services lies in the absence of structured datasets. While vast amounts of unstructured data related to mental health and counseling exist online, the lack of organized and cohesive data impedes the development of effective AI solutions. "HICON AI" aims to rectify this issue by constructing a structured dataset as the cornerstone of its AI-driven services. This dataset is essential for enhancing the accuracy and efficacy of its responses, providing a robust foundation for training and refining the AI model.

- **Generalized Approach:** Many current chatbots and AI-driven counseling platforms employ a one-size-fits-all approach, utilizing a standardized set of prompts and responses for all users. However, this generalized approach lacks personalization and fails to address the individual needs and nuances of users. "HICON AI" recognizes the importance of personalization in its interactions. Leveraging advanced natural language processing techniques, it adapts its responses to each user's characteristics and concerns, enhancing user experience and improving the effectiveness of the counseling and support provided.

- **Not Specific Domain:** Existing counseling platforms often lack specificity when it comes to domains, offering general advice or connecting users with counselors across various areas. However, there is a notable absence of AI-driven solutions tailored to specific domains, such as the one targeted by "HICON AI." This specialized focus is crucial as it allows the AI to develop a deep understanding of the unique challenges, terminology, and context associated with the domain it aims to serve. By concentrating on a specific area, "HICON AI" can deliver more relevant and targeted assistance, establishing itself as a valuable resource for individuals seeking help in this field.

CHAPTER-2

LITERATURE SURVEY ON CHATBOTS

This research paper withholds the amount of ease it can provide to the students by building 'HICON AI.' Our model uses LLMs as a critical factor for the personalized response and gives responses as a chatbot. Here is the prior research from where we took inspiration for our paper.

2.1 RAGged Edges: The Double-Edged Sword of Retrieval-Augmented Chatbots [1]

This paper explores the effectiveness of using retrieval-augmented generation (RAG) to reduce hallucinations in large language models (LLMs) like ChatGPT. RAG integrates external knowledge by retrieving relevant information to include in the prompt given to the LLM. The authors conducted an empirical study where human participants evaluated the accuracy of LLM responses to prompts about their CV, both with and without including context from their actual CV.

The results showed that including context dramatically increased accuracy, with responses being about 18 times more likely to be fully accurate compared to no context. However, even with accurate context provided, around 6% of responses still contained inaccuracies. The authors categorized these errors into five types: noisy context, mismatch between instructions and context, context-based synthesis of plausible but incorrect information, unusual formatting issues, and incomplete context. Particularly concerning were cases where the LLM hallucinated believable synthetic content by interpolating between factual context.

The authors conclude that while RAG is highly effective at improving accuracy, the remaining errors reveal nuanced challenges in prompting and context utilization. They suggest further research is needed on managing user expectations, prompt engineering, and preventing synthetic hallucinations when context is incomplete or contradictory. Overall, the findings highlight the complex nature of hallucinations and the need for continued work to develop more reliable, context-aware AI systems.

2.2 Prompt-RAG: Pioneering Vector Embedding-Free Retrieval-Augmented Generation in Niche Domains, Exemplified by Korean Medicine [2]

This paper introduces Prompt-RAG, a novel methodology for retrieval-augmented generation (RAG) that does not rely on vector embeddings. The authors first demonstrate that large language model (LLM) based vector embeddings may not adequately capture human-perceived conceptual meanings in niche domains like Korean medicine, suggesting conventional RAG could perform suboptimally in such cases.

As an alternative, Prompt-RAG utilizes the language understanding capabilities of LLMs to directly retrieve relevant information from a document's table of contents based on natural language prompts.

The authors develop a Question-Answering chatbot for the Korean medicine domain using Prompt-RAG and evaluate its performance against baselines like ChatGPT and conventional vector embedding RAG models. Their results show that Prompt-RAG outperforms the other models in terms of relevance and informativeness of responses, while being comparable for readability. It excelled across different question types requiring direct retrieval, comprehensive understanding, and functional robustness. Though slower than vector embedding methods, Prompt-RAG overcomes their limitations in specialized domains.

The paper concludes that Prompt-RAG provides a promising approach for developing high-quality RAG systems in niche fields as it takes advantage of rapidly advancing LLM capabilities. The authors discuss potential wider applications, highlight current limitations like reliance on tables of contents, and suggest improvements can be expected as language models continue advancing.

2.3 From RAG to QA-RAG [3]

This paper introduces a new chatbot model called QA-RAG (Question and Answer Retrieval Augmented Generation) that is designed to navigate complex pharmaceutical regulatory guidelines and provide accurate answers to user queries. The QA-RAG model leverages generative AI and the Retrieval Augmented Generation (RAG) method, but improves upon conventional RAG by incorporating a fine-tuned large language model to enhance document retrieval relevance and accuracy.

The key innovations of QA-RAG include using a dual-track approach that retrieves documents based on both the user's question and a hypothetical answer from the fine-tuned LLM, followed by reranking to prioritize the most relevant documents. Experimental results showed QA-RAG outperformed baselines like standard RAG and other retrieval methods in terms of context precision, recall, and final answer generation quality when evaluated on FDA pharmaceutical guideline questions. The researchers position QA-RAG as a promising tool to streamline regulatory compliance in the pharmaceutical industry by automatically providing reliable guidance from complex documentation.

2.4 EduChat: A Large-Scale Language Model-based Chatbot System for Intelligent Education [4]

The terrain of intelligent education is set to change with the deployment of EduChat, a state-of-the-art chatbot platform. Leveraging the capabilities of comprehensive language models, it offers personalized, impartial, and empathetic guidance to teachers, students, and parents. EduChat enhances various aspects of education like Socratic instruction, subjective assessment, emotional aid, and open query handling, utilizing principles from education and psychology. Incorporating LLMs into the world of education isn't without its difficulties, such as inadequate prior pedagogical expertise and the challenge of keeping up with an ever-growing field. EduChat tackles these barriers with a diverse plan of action, which involves pre-training via a wide range of educational reading materials and varied teaching techniques, refined through the guidance of experts and feedback loops, and the use of retrieval-augmented technology for immediate access to online information. In detailed testing using the C-Eval benchmark, EduChat shows impeccable performance across all four tasks, particularly in education-specific features while maintaining equally strong core capabilities with other large-scale models.

Simply put, EduChat is a revolutionary technology that has the power to transform the face of intelligent education. Its mission is to provide personalized and just support, and it aims to broaden its reach in the years to come by incorporating course recommendations, career guidance, and question formulation. In comprehensive evaluations conducted using the C-Eval benchmark, EduChat consistently demonstrates exceptional performance across all four tasks, particularly excelling in the domain of education-specific features, all the while maintaining core capabilities that are on par with other prominent large-scale language models.

2.5 Adoption of AI-Chatbots to Enhance Student Learning Experience in Higher Education in India [5]

This examination paper presents an exhaustive investigation of the groundbreaking capability of Chatbot innovation in the Indian advanced education area, especially with regards to "Training 4.0." This arising instructive scene underscores the joining of Man-made reasoning (computer-based intelligence) inside student focused frameworks, because of the developing necessities of understudies and the difficulties presented by an impacting world. The paper highlights the fundamental idea of instructive headway to adjust to present day ways of life, monetary elements, mechanical advances, and the developing hole in educator accessibility. It sets that Chatbots can assume a crucial part in tending to these difficulties, giving proficient, open, and customized instructive help. The focal examination question tests the manners by which Chatbot advances can help the Indian training area in upgrading the understudy opportunity for growth. The review forms speculations that challenge the tried-and-true way of thinking, declaring that elements like orientation, level of schooling, and age are probably not going to altogether affect

the reception of Chatbot innovation inside the Indian advanced education scene. The paper takes on a thorough quantitative examination system, gathering information through overviews directed across probably the most conspicuous advanced education foundations utilizing Chatbot innovation in India. The outcomes gathered from these reviews uncover that understudies are progressively disposed to use Chatbots for settling instructive inquiries and favor this method of correspondence over conventional channels. Prominently, almost 50% of the reviewed understudies have drawn in with Chatbots to speak with their instructive establishments throughout the last year, with Chatbots arising as the favored method for quick correspondence. In any case, concerns with respect to likely errors in counsel and security issues loom not too far off, which require consideration. The paper eventually reasons that there is no measurably critical connection between respondents' orientation, age, or level of schooling and the reception of Chatbot innovation in the Indian advanced education area. The reconciliation of artificial intelligence Chatbots holds the commitment of propelling understudy focused learning, portrayed by upgraded correspondence, research help, and smoothed out internet based test evaluating. Besides, the paper recommends roads for future exploration, perceiving the basic requirement for a comprehensive comprehension of the developing scene of understudy cooperation with innovation.

2.6 Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects [6]

In the reliably developing space of modernized language dealing with, Enormous Language Models (LLMs) have emerged as moderate components, having the astonishing expertise to understand complex semantic nuances and produce sensibly huge responses. LLMs, a subset of Modernized thinking (mimicked insight), have become critical gadgets across different purposes, including standard language dealing with, machine understanding, vision-based tasks, and question-answering. This comprehensive outline gives a start to finish examination of LLMs, including their genuine new development, designing foundations, getting ready frameworks, different applications, and the pile hardships they face. Starting with a clarification of generative PC based knowledge's critical thoughts and an appraisal of the Generative Pre-arranged Transformers (GPT) plan, the paper investigates through the extraordinary journey of LLMs and the different readiness techniques used. It dives into the broad scene of LLM applications, spreading over clinical, preparing, finance, planning, media, entertainment, legislative issues, and guideline, and examines how LLMs are reshaping the man-made consciousness scene and their pressing position in consistent disclosures, authentic decisive reasoning, and that is only the start. Very, the paper keeps an eye on the confounded challenges related with sending LLMs in authentic circumstances, including moral examinations, model inclinations, interpretability issues, and the critical computational resources they demand. It moreover includes frameworks for working on the generosity, controllability, sensibility, and nature of LLMs inside Generative computerized reasoning. This thorough audit closes by featuring the inevitable destiny of LLM research and the essential to beat troubles, ensuring the trustworthiness and utility of this development. Also, the paper plunges into the space of Chatbots, which have developed positive progress in client care applications, discussing their relevance in further developing client experience and business improvement. It on a very basic level ganders at driving Chatbots, including ChatGPT, Google Versifier, and Meta's Llama-2, considering factors like accuracy, comfort, cost, and blend, offering a significant perspective on their specific credits. The story keeps on giving a close to assessment among ChatGPT and Google Versifier, making sense of the fundamental models, tweaking techniques, and relative characteristics of these notable LLMs. It portrays the unquestionable characteristics of LaMDA and GPT-3-based models.

The discussion further loosens up to the field of "Brief Planning," underlining the fundamental occupation of effectively made prompts in outfitting LLMs' capacities. It plunges into different brief arrangement methodology, including unequivocal headings, system express rules, planning with models, control tokens, and association procedures, featuring their importance in further developing LLM execution.

Finally, the paper considers the troubles and cutoff points natural in LLMs, wrapping a wide bunch of stresses, including uneven data, limited savvy instinct, computational resource demands, moral considerations, and that is only the start. The discussion fills in as an indication of the necessity for subject matter experts and experts to perceive and determine these issues in their excursion to ensure the moral and suitable utilization of LLMs, developing the improvement of novel models that can overcome these challenges and cutoff points. This total outline fills in as a significant resource for researchers, specialists, and fans, offering a sweeping perception of LLMs, their new development, applications, and the different troubles they experience, with the assumption for trim the possible destiny of mimicked knowledge in genuine applications.

2.7 Benchmarking LLM powered Chatbots: Methods and Metrics [7]

The augmentation of free conversational trained professionals, or chatbots, powered by state-of-the-art Generative man-made knowledge gadgets like Tremendous Language Models (LLMs) has become continuously normal in the endeavor space for client and associate help. Evaluating these chatbots, particularly those relying upon LLMs, requires exact examination of their show. Considering this need, this paper familiarizes a unique benchmark insinuated with as the E2E (Beginning to end) benchmark, expected to gauge the precision and utility of chatbot responses. The paper shows the use of the E2E benchmark in surveying a model chatbot across different levels of intricacy, standing apart it from existing state of the art estimations and showing the unmatched show of the proposed benchmark. The show contextualizes the significance of chatbots in the area of man-made awareness, highlighting their work in client help, clinical benefits, preparing, and that is just a glimpse of something larger, especially in giving endeavor thing support. These chatbots much of the time rely upon Colossal Language Models, which are tended to with respect to their abilities and the ascent of an enormous test, implied as representation. Considering this, the paper focuses on the meaning of using a blend of evaluation methodology to study immense language models, particularly concerning chatbots.

It features the essential components of precision and accommodation in benchmarking chatbot execution, keeping an eye on the propelling hardships related with present day LLM-filled chatbots. The paper outlines different kinds of chatbot benchmarks and their different advantages and downsides, proposing the E2E benchmark as a client driven approach. The E2E benchmark, which bases on semantic comparability estimations, is discussed thoroughly, highlighting its ability to update the evaluation and appraisal of chatbots, particularly when diverged from the extensively used Radical scores. Eventually, the paper includes the speedy improvement of chatbots and the essential necessity for suitable execution assessment through benchmarking methods.

It features the introduction of the E2E benchmark, counting the usage of cosine similarity for appraisal and checking out at changed sentence embeddings. The paper looks at the show of chatbots concerning various libraries and methods, finally thinking that the E2E benchmark, particularly when used with the ST library, is the best choice for benchmarking chatbot execution. On the other hand, the Dissident scores are seen as erratic, inciting the idea that they may not be a reliable benchmarking strategy. By and large, paper focuses on the meaning of advancing exploration in chatbot benchmarking and assumes that the E2E benchmark is a strong procedure.

CHAPTER-3

SYSTEM ARCHITECTURE OF HICON AI

3.1 SYSTEM ANALYSIS

System planning is the process of creating the factors, components, interfaces, modules, and data flows that are necessary for a system to function. The system architecture process seeks to offer thorough information about the structure and its main components, ensuring that the application is consistent with the architectural framework portrayed in system architecture replicas and views. It is critical to understand that the system design process is not a one-time event; rather, it is a continuous, evolving process that adapts as the system evolves. This technique serves as a blueprint for creating a system that ensures the smooth operation of all of its components.

3.1.1 FUNCTIONAL REQUIREMENTS

Functional requirements are specifications that outline the features or functionalities that a software system, product, or application needs to satisfy user needs and achieve its intended goals. These specifications spell out the functions and responses that the system must have for different inputs or circumstances. Functional requirements form the basis for design, development, testing, and quality assurance and are an essential part of software and system development. They aid in making sure that the finished product satisfies both business and user goals. Functional necessities for a large-scale language model-based Counselor AI chatbot, specially designed for higher observe counseling within the domain, include the subsequent points:

- **Domain Expertise:** The chatbot need to possess in-depth expertise of higher education systems, universities, faculties, courses, and admission approaches to offer accurate and applicable guidance.

- **Personalized Counseling:** The chatbot must provide customized counseling periods, and other functionalities.

- **Course Recommendations:** It must be capable of suggesting precise publications, majors, and packages based at the user's profile and preferences.

- **Admission Guidance:** The chatbot must guide users through the application and admission tactics, which include necessities, time limits, and documentation.

- **Financial Aid Information:** Providing facts on scholarships, offers, and economic aid options for higher education is important to assist customers access funding opportunities.

- **Test Preparation Assistance:** The chatbot can provide steering on standardized tests (e.g., SAT, GRE) and offer have a look at resources and techniques.

- **University Selection:** It must help customers in selecting universities or schools that align with their academic and career goals, considering factors like vicinity, ranking, and specialization.

3.1.2 NON-FUNCTIONAL REQUIREMENTS

Specifications that define the overall traits and behavior of a software program system or software are referred to as non-functional requirements, on occasion known as high-quality attributes or machine attributes. Non-functional requirements give attention to the traits and techniques by which the gadget contains out its distinct tasks, in comparison to purposeful necessities, which specify unique functions or functions the machine must have. For evaluating a gadget's ordinary high-quality, overall performance, security, and value, non-functional requirements are essential.

- **System Security:** Ensuring system security to prevent unauthorized access and sessions. Data must also be stored in a secure environment and location. It requires a safe channel for data transfer.

- **Concurrency and Capacity Multiple:** Computations should be compatible with the system's concurrency and capacity. Coincidentally and maybe in trade with one another.

- **Performance:** Most individuals link performance with a timeframe. These are some of the most important considerations, especially while the design is at the armature phase.

- **Trust Ability:** It is vital to ensure and alert about system transactions and processing, and something as basic as keeping a system log will increase the time and effort required to do it from the very first morning. Data should be exchanged in a reliable manner utilizing trusted protocols.

- **Maintainability:** A well-designed system is meant to work for a long period. As a result, it will always be necessary to practice preventive and remedial conservation. Conservation might indicate that the system will eventually develop and update its capabilities and operations.

- **Usability:** One of the major pillars that supports a product is end-user satisfaction with design accomplishment. The requirements for stoner experience should be considered from the start of the design process.

3.2 SYSTEM SPECIFICATION REQUIREMENTS

3.2.1 SOFTWARE REQUIREMENTS

Minimal software specifications of the proposed system are:

- **Operating System:** Windows 11

- **Technology:** Natural Language processing

- **Tools / Model:** pycharm, pandas, socket, numpy, whisper

- **Dataset:** Self-created students' dataset

3.2.2 HARDWARE REQUIREMENTS

The minimal hardware specifications of the proposed system are:

- **CPU:** Intel i3
- **Memory:** 4 GB
- **Storage:** 32 GB

3.3 DESIGN OF MODULES

System design entails the development of essential system components such as the framework, modules, and elements, as well as the different interactions between these components and the flow of data inside the system. The major purpose component pieces allow for smooth alignment with the architectural criteria stated in the system's models and views.

System Components:

- **Layout:** This conceptual model specifies the system's structure, activity, and numerous views, which are frequently depicted using flowcharts.
- **Modules:** These are discrete components that perform specialized duties within the system. These components work together to make up the system.
- **Components:** These include particular functions or groups of related functions built from modules.
- **UI:** These are the interaction boundaries that allow system components to share information and communicate.
- **Data:** Controls the processing and movement of information and data across the system.

3.3.1 INITIALIZE DESIGN DEFINITIONS

- Create a strategic strategy for identifying, choosing, and defining the system's fundamental technologies and physical interfaces.

- Establish contingency plans to manage anticipated obsolescence or evolution of system technology and components during operation.

- Develop a comprehensive design description approach, including necessary prerequisites for supporting structures, goods, and facilities for successful design execution.

3.3.2 ESTABLISH DESIGN FEATURES

- Specify design characteristics related to architectural features and assess their feasibility.

- Identify and establish interfaces that may not have been previously defined as design details evolve.

- Define and validate the design characteristics for each element within the system.

3.3.3 MEASURE ALTERNATIVES FOR OBTAINING SYSTEM ELEMENTS

- Evaluate available design options.

- Select the most appropriate choice.

- If the decision is to develop the system element, proceed with the design description process and implementation.

- If the decision is to purchase or utilize a system element, employ the acquisition process to acquire the system element.

3.3.4 MANAGE THE DESIGN

- Document and manage the rationale behind all choices among alternatives and design features.

- Oversee and regulate the elaboration of design characteristics.

3.4 SYSTEM DIAGRAM

A system diagram is a visual representation of a system's components and their interconnections. It helps in understanding the system's structure and functionality, facilitating effective planning, design, and troubleshooting.
Description also indicates how non-functional conditions will be satisfied. For Example:

- Safety integrity rudiments of the design that reduce the threat that the system will beget (or allow occasion of detriment to property and mortal beings.

- System vacuity for illustration, rudiments of the design that enable a system to operate24/7.

- Fault forbearance rudiments of the design that allow the system.

3.4.1 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a visual representation illustrating the flow of data within a process or system. It utilizes standardized symbols like blocks, circles, arrows, and concise text labels to portray data inputs, processes, storage points, and the connections between these elements. DFD can vary from simple, hand-drawn process overviews to intricate, multi-level diagrams that provide in-depth insights into data handling.

There are few levels of Data flow Diagram:

- DFD Level 0
- DFD Level 1

DFD Level 0 offers an initial overview of the entire system or process being analyzed or modeled. It is designed to provide a quick and comprehensive perspective, representing the system as a single high-level process while showcasing its interaction with external entities.

The fig. 4 DFD Level 0 is shown below depicts the initial overview of the system.

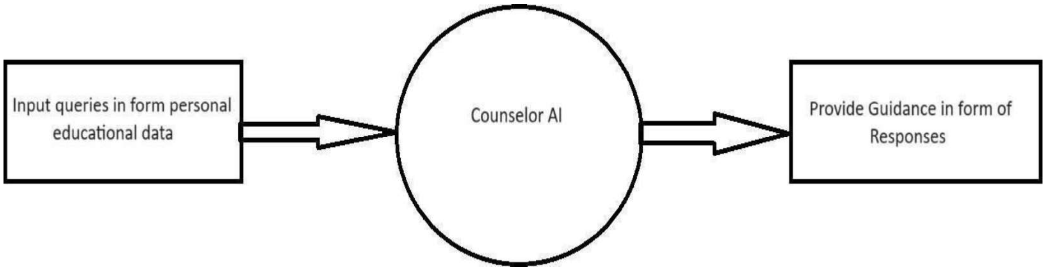

Input queries in form personal educational data
Counselor AI
Provide Guidance in form of Responses

Fig 4. DFD Level 0 Diagram of HICON AI

DFD Level 1 of the system shown in fig. 4 serves to dissect the primary process into sub-processes, which can be further explored at a more detailed level. Additionally, it includes data stores that are utilized by the main process. It delineates each of the primary sub-processes, collectively forming the comprehensive system.

The fig. 5 DFD Level 1 on the next page depicts the primary processes and its sub-processes overview of the system.

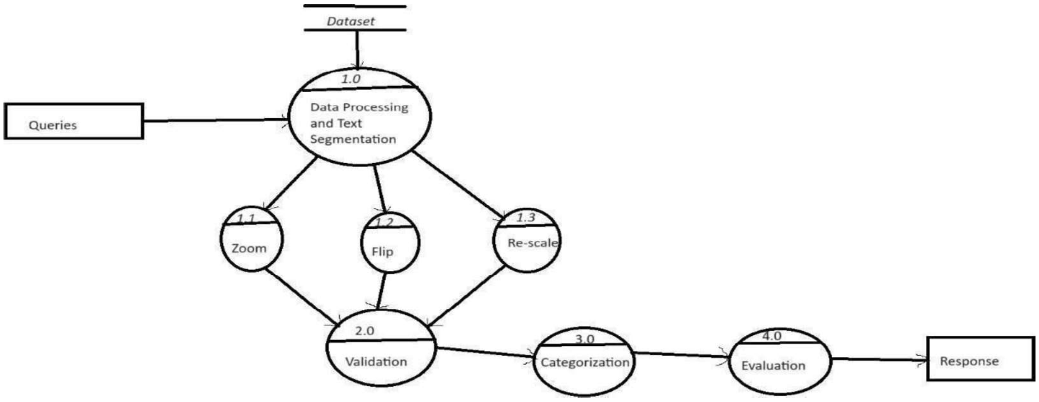


Fig 5. DFD Level 1 Diagram of HICON AI

3.4.2 SEQUENCE DIAGRAM

It is a part of the Unified Modeling Diagram (UML), which is a type of interaction diagram that visualizes how messages and tasks are passed between objects within a system over a time. Its shows the sequence of those interaction, hence the name. The below fig. 5 shows the Sequence Diagram of the system.

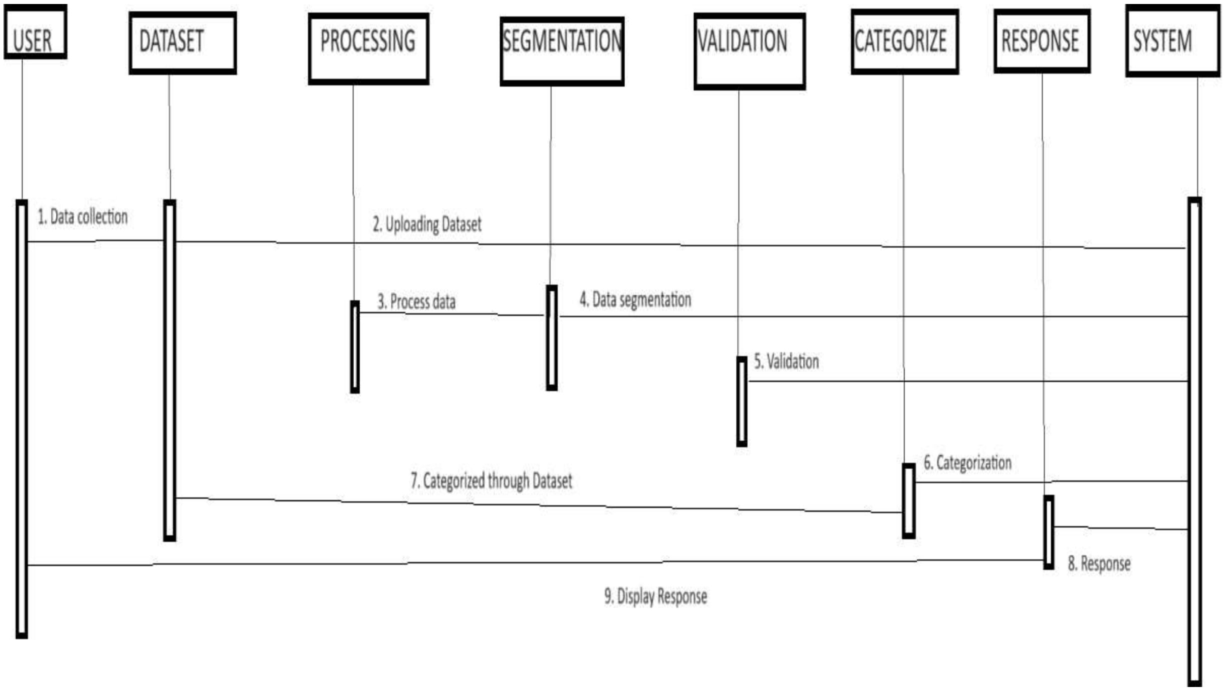


Fig6. Sequence Diagram of HICON AI

3.4.3 ARCHITECTURE DIAGRAM

An architecture diagram is a visual representation that shows the components, interactions, relationships, and structure of a system or software application. It is also referred to as a system architecture diagram or software architecture diagram. It offers a high-level summary of how various components of the software or system cooperate to fulfill its intended function.

The below fig. 7 shows the Architecture Diagram of the system.

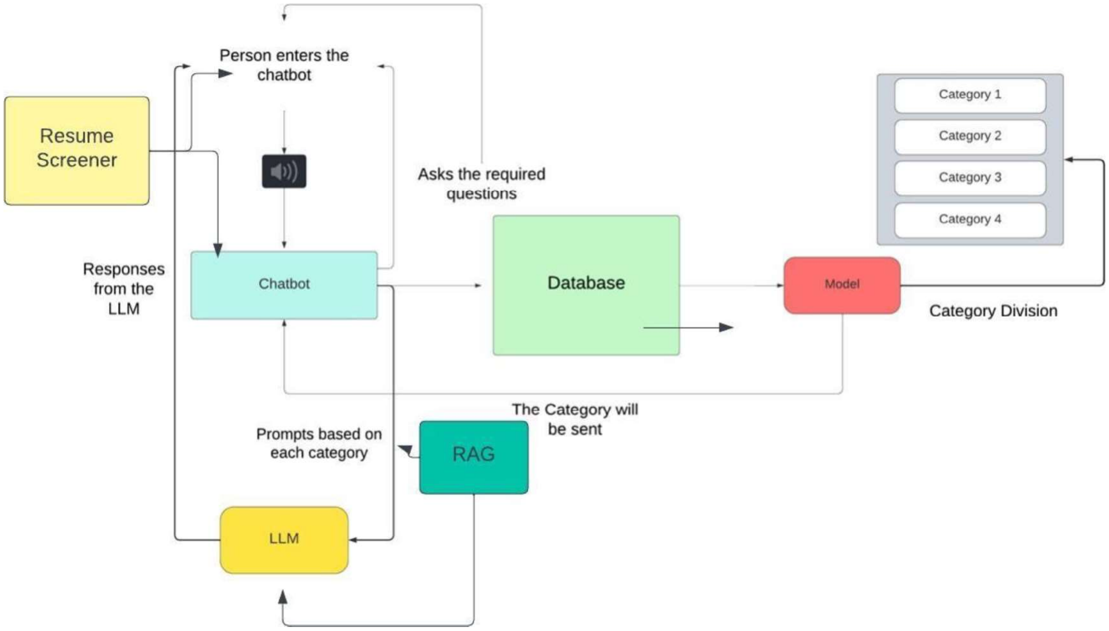


Fig. 7 Architecture Diagram of HICON AI

CHAPTER-4

METHODOLOGY INVOLVED IN HICON AI

4.1 ABOUT SYSTEM IMPLEMENTATION

The stage of the software development life cycle (SDLC) known as "system implementation" is when the planned system is constructed and put into use. It is a crucial phase that comes before system testing, integration, and maintenance and comes after system design. Converting the conceptual design into a functional software system is known as system implementation.

4.1.1 IMPLEMENTATION DESCRIPTION

For the system implementation of the assistant, we opted for a form-based, text-input approach instead of socket programming. This approach allows the user, who is the student, to input data directly from their resume using a Resume Screener. This decision was made to streamline the process and simplify communication between the user and our Counselor AI model. Within the application, there are two main phases: one where data is gathered from the user via the form, and another where responses are provided based on the input received.

Before Categorization, a significant amount of data pre-processing is performed, during which all data items are assigned to their corresponding parameter classes and binary parameters are mapped. Data dictionary converted to pandas. Data frames will be used by the model to classify the data before being returned to the main dataset for further usage.

To create the categorizer, a random forest model was developed for category division. Due to the unique dataset created specifically for the model, there was limited data available. Random forest was chosen for its efficiency with smaller datasets. While both decision trees and random forest algorithms could have been utilized for categorization, random forest was preferred due to its ability to mitigate overfitting and its robustness to noise. Random forest is essentially a combination of multiple decision trees, which helps reduce overfitting by allowing each tree to overfit differently \cite{randomforest}. Overfitting occurs when a model accurately predicts training data but fails to generalize to new data. This phenomenon arises when variance increases while bias decreases.

After creating the categorizer, we integrated the prompts for each category directly into the question-answer category interface. This eliminated the need for socket programming to deliver responses. Additionally, for personalized responses, we included data obtained from the user within the prompts themselves.

To finalize our project, we're integrating an LLM to provide responses. We opted for Llama 2 developed by Meta-AI. Llama 2 was chosen over other LLMs primarily due to its novelty and accessibility. Moreover, Llama 2 boasts extensive training on billions of parameters, ranging from 7B to 70B parameters, making its responses exceptionally efficient. Additionally, we're augmenting the model with Retrieval-Augmented Generation (RAG) for further fine-tuning and enhanced efficiency in response generation.

4.1.2 SYSTEM ARCHITECTURE DESCRIPTION

When a user inputs 'HICON AI,' the assistant greets them warmly. The assistant will then prompt the user to answer specific questions about their unique use case. These questions will cover essential information like the user's name, age, year of study, as well as professional details such as internships completed and training specifics. The responses to these questions will be immediately populated in the resume screener. Additionally, there will be behavioral inquiries, such as whether the user has joined any social groups, with further details on these groups being saved in a separate database. These questions aid in categorizing users into distinct groups. Furthermore, language models (LLMs) will generate personalized questions based on these categories and tailored to everyone, such as asking about their desired university and courses. Since each person will have a unique perspective, responses will be customized to their interests. This process helps construct the final responses of HICON AI.

4.2 MODULES DESCRIPTION

4.2.1 DATA GATHERING

The HICON AI chatbot's first communication with the user takes place in the "Collection of

Data" module. Its main goal is to compile pertinent student data, which is necessary for individualized counseling and guidance. This module can be divided into multiple essential parts:

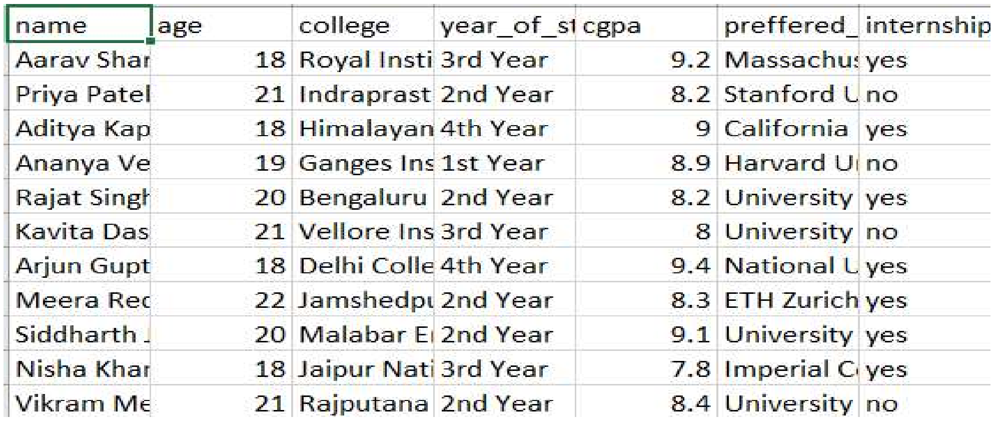
- **User Profiling:** The first step of the process involves asking the user a series of questions about their educational background and extracurricular activities. These inquiries are meant to record specifics like educational background, professional experiences, research endeavors, volunteer work, social activities, and roles of authority held. The chatbot obtains a comprehensive understanding of the user's profile by gathering this extensive data.

- **Data Analysis:** After gathering user responses, the system uses data analysis methods to extract pertinent data and parse it. Finding significant keywords, trends, and patterns in the user's input is the goal of this analysis.

- **Personalization:** A user's profile is made specifically for them using the information that has been gathered. The chatbot uses this profile as the starting point for advice and counseling during the conversation. It guarantees that the suggestions and guidance given are specific to the user's goals and background.

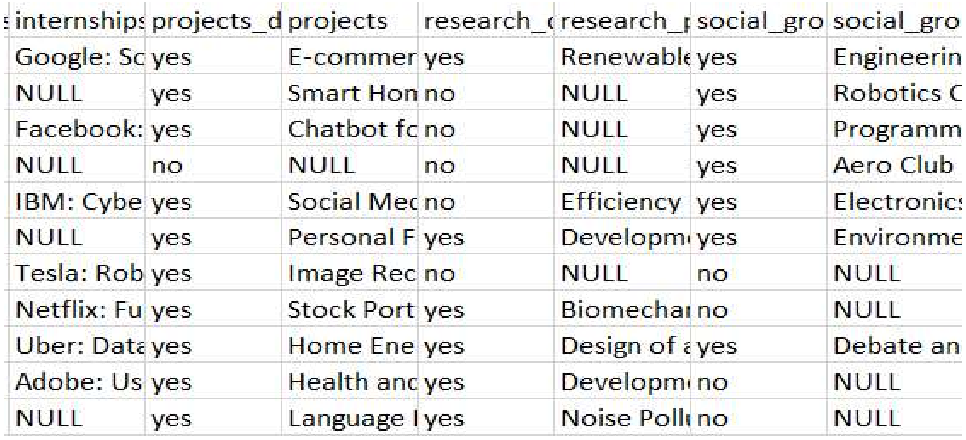
- **Privacy and Data Security:** Ensuring user privacy and data security is a key component of this module. To protect the user's sensitive information, the chatbot should abide by stringent data protection laws and guidelines.

The Table 1 User Data1 and Table 2 User Data2 (on the next page) shows the self-created dataset used to trained the LLM Model.



name	age	college	year_of_st	cgpa	preffered_	internship
Aarav Shar	18	Royal Insti	3rd Year	9.2	Massachus	yes
Priya Patel	21	Indraprast	2nd Year	8.2	Stanford U	no
Aditya Kap	18	Himalayan	4th Year	9	California	yes
Ananya Ve	19	Ganges Ins	1st Year	8.9	Harvard U	no
Rajat Singh	20	Bengaluru	2nd Year	8.2	University	yes
Kavita Das	21	Vellore Ins	3rd Year	8	University	no
Arjun Gupt	18	Delhi Colle	4th Year	9.4	National U	yes
Meera Rec	22	Jamshedpu	2nd Year	8.3	ETH Zurich	yes
Siddharth J	20	Malabar E	2nd Year	9.1	University	yes
Nisha Khar	18	Jaipur Nati	3rd Year	7.8	Imperial C	yes
Vikram Me	21	Rajputana	2nd Year	8.4	University	no

Table1. User Dataset1



internships	projects_d	projects	research_	research_	social_gro	social_gro
Google: Sc	yes	E-commer	yes	Renewable	yes	Engineerin
NULL	yes	Smart Hon	no	NULL	yes	Robotics C
Facebook:	yes	Chatbot fc	no	NULL	yes	Programm
NULL	no	NULL	no	NULL	yes	Aero Club
IBM: Cybe	yes	Social Mec	no	Efficiency	yes	Electronics
NULL	yes	Personal F	yes	Developm	yes	Environme
Tesla: Rob	yes	Image Rec	no	NULL	no	NULL
Netflix: Fu	yes	Stock Port	yes	Biomechar	no	NULL
Uber: Data	yes	Home Ene	yes	Design of a	yes	Debate an
Adobe: Us	yes	Health anc	yes	Developm	no	NULL
NULL	yes	Language I	yes	Noise Poll	no	NULL

Table2. User Dataset2

4.2.2 STUDENT DIVISION

The "Category Division" module is essential for classifying users into different groups according to their objectives and plans in the higher education field. This classification makes it possible for the chatbot to provide tailored advice and resources. The following parts make up the module:

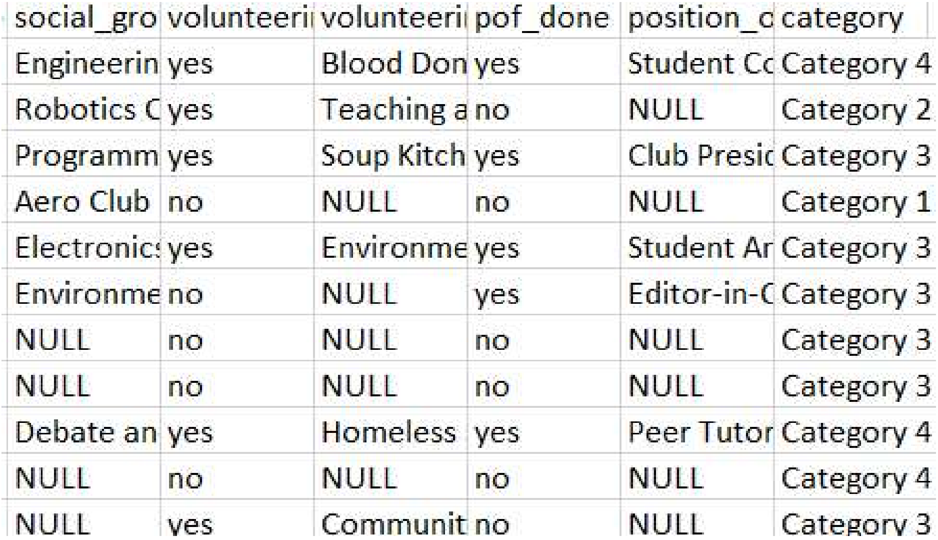
- **Machine Learning Classification:** A machine learning model is used to classify users into one of the predefined categories after gathering user data. These groups function as a system of classification that aids in determining the main goals of the user.

- **Random Forest Algorithm:** Random Forest is a robust machine learning technique that falls under the umbrella of ensemble learning. This adaptable method is commonly used for classification and regression tasks. Random Forest, created by Leo Breiman and Adele Cutler, works by building several decision trees and mixing their predictions to improve accuracy and generalization.

Random Forest's underlying notion entails numerous critical steps. To begin, it involves bootstrapping, which is the process of constructing several subsets (bootstrapped samples) from the original dataset using random sampling with replacement. After then, each of these subgroups is utilized to train a different decision tree. The incorporation of feature unpredictability is one of Random Forest's unique features. Random Forest considers only a random selection of features at each split in a decision tree, rather than all available features. This method increases tree diversity while reducing the risk of overfitting.

Random Forest then constructs numerous decision trees using these subsets of data and feature randomness. Based on the data and labels provided, these trees attempt to predict the target variable. They are developed to a predetermined depth, resulting in various structures and decision rules. Random Forest also minimizes variance. Decision trees can be sensitive to changes in training data, resulting in slightly different decision tree architectures with minor differences in predictions. The ensemble approach of Random Forest reduces these variances, resulting in more stable and dependable forecasts. Random Forest excels on small datasets with noisy or irrelevant attributes. Decision trees frequently overfit noisy variables. Random Forest includes feature randomness, which ensures that only a fraction of the features is used at each node, limiting the impact of noisy variables. Furthermore, Random Forest is known to be more accurate. Even with minimal data, it builds a more robust and accurate model by pooling the wisdom of several decision trees.

- **Four Distinct Categories:** There are four primary categories into which the categorization is usually split.
  - Category 1: Uninformed users who are examining the domain's features.
  - Category 2: People who want to advance their professional and academic abilities.
  - Category 3: Individuals looking to fill in the gaps in their profiles.
  - Category 4: People who want to stay abreast of current events and improve their chance of getting into the universities of their choice.
- **Dynamic Classification:** It is crucial to remember that a user's category may change as their objectives and interactions with the chatbot alter over time. The chatbot's ability to modify its responses in response to changes in the user's needs and intentions is ensured by this dynamic classification.



social_gro	volunteeri	volunteeri	pof_done	position_c	category
Engineerin	yes	Blood Don	yes	Student Cc	Category 4
Robotics C	yes	Teaching a	no	NULL	Category 2
Programm	yes	Soup Kitch	yes	Club Presic	Category 3
Aero Club	no	NULL	no	NULL	Category 1
Electronic:	yes	Environme	yes	Student Ar	Category 3
Environme	no	NULL	yes	Editor-in-C	Category 3
NULL	no	NULL	no	NULL	Category 3
NULL	no	NULL	no	NULL	Category 3
Debate an	yes	Homeless	yes	Peer Tutor	Category 4
NULL	no	NULL	no	NULL	Category 4
NULL	yes	Communit	no	NULL	Category 3

Table3. User Dataset3

The above data of Table3. User Dataset 3 is used to trained the LLM.

4.2.3 PROMPTING

The HICON AI chatbot's ability to customize its dialogue and offer advice is greatly enhanced by the "Prompts for Each Category" module. It involves the subsequent elements:

- **Category-Specific Prompts:** The chatbot creates a series of prompts that are unique to each of the predefined categories. These questions are intended to direct the discussion and cover the particular use cases and objectives related to each category.

- **Use Case Alignment:** The prompts are in line with the specific use cases, difficulties, and goals that users in each category frequently face. For instance, prompts for Category 1 might expose users to fundamental ideas and materials, whereas prompts for Category 4 might concentrate on sophisticated approaches and chances.

- **Personalization Through Data:** The chatbot uses the user's personal information to fine-tune the prompts to improve personalization. This guarantees that the discussion stays extremely pertinent and speaks to the user's unique situation and goals.

- **User Engagement:** Sustaining user engagement requires the use of effective prompts. They initiate conversations and provide users with organized, educational guidance as they move through counseling process.

4.2.4 RESPONSES

The core of the HICON AI chatbot's capacity to offer insightful advice and assistance is the "Response Generation" module. It includes the subsequent elements:

- **Language Models (LLMs):** Responses to the user's questions and prompts are generated by language models, such as GPT-3 or other sophisticated AI models. To create responses that are both logical and sensitive to context, these models make use of natural language processing (NLP) techniques.

- **Recommendations for Resources:** The chatbot makes recommendations for a variety of resources, such as articles, videos, courses, and university programs, by leveraging LLMs. These suggestions are customized based on the user's needs and category.

- **Advisory Responses:** The chatbot is programmed to provide users who are looking for direction or advice with practical suggestions and methods. These answers are based on a wealth of information and experience in the higher study domain.

4.2.5 RETRIEVAL AUGMENTED GENERATION (RAG)

The LLM-based project by HICON AI uses cutting-edge technology called "Retrieval Augmented Generation" (RAG), which is a paradigm shift in the processing of legal information. Fundamentally, retrieval and generation processes interact intricately using large-scale pre-trained language models like BERT or GPT, which RAG leverages. The information retrieval part of the system makes use of sophisticated search algorithms and a sizable legal database to quickly find and extract pertinent statutes, case law, and legal precedents.

To improve the accuracy of legal research, RAG uses cutting edge Natural Language Understanding (NLU) methods to understand intricate legal questions contextually and with subtlety. This guarantees that the information that is retrieved closely matches the user's intent, giving legal professionals a thorough and precise starting point for their work on the LLM-based project. Then, the system smoothly moves into the natural language generation stage, using sophisticated language models to generate comprehensible and contextually relevant responses, which speeds up the document drafting process.

- **Transformer-based Architecture:** HICON AI employs transformer-based language models (e.g., BERT, GPT) for both information retrieval and natural language generation, forming a robust technical foundation.

- **Vectorized Retrieval:** Advanced search algorithms with dense vector representations enable rapid and precise retrieval of relevant legal documents, such as statutes and case law, enhancing the efficiency of information retrieval.

- **Fine-tuned NLU and Domain-specific Generation:** The system utilizes fine-tuned domain-specific embeddings for Natural Language Understanding (NLU), enabling nuanced comprehension of complex legal queries. In the generation phase, transformer-based models fine-tuned for legal language ensure precise and contextually appropriate content generation.

- **Adaptive Learning and Real-time Updates:** Continuous learning mechanisms, including reinforcement learning, enable HICON AI's RAG to adapt to evolving legal landscapes, providing real-time insights into the latest legal precedents, regulatory changes, and judicial interpretations.

4.2.6 RESUME SCREENER

HICON AI's resume screener is a specialized tool designed for our LLM-based project. This advanced technology uses smart algorithms to quickly sift through legal resumes and pinpoint the most suitable information required to give the best possible counselling advice. Unlike traditional methods, our screener goes beyond just matching keywords; it understands the intricacies of legal language and context, ensuring a more accurate assessment of candidates' qualifications.

With a focus on key credentials like legal education, relevant experience, and specific skills, the HICON AI resume screener simplifies the hiring process for law firms and legal departments. It keeps up-to-date with the latest legal trends, making it an asset for any project requiring specialized legal expertise. By making candidate selection more efficient and precise, our resume screener contributes to the overall success of LLM-based projects.

In the resume screening process, HICON AI leverages Optical Character Recognition (OCR) technology, particularly Tesseract, to extract textual information from resumes efficiently. Tesseract, an open-source OCR engine developed by Google, serves as a cornerstone in accurately converting scanned documents, such as resumes, into editable text. This functionality enables HICON AI to rapidly analyze vast quantities of resumes, extracting crucial details like education, work experience, and skills. By employing Tesseract, HICON AI streamlines the initial screening phase, allowing recruiters to swiftly identify qualified candidates based on predefined criteria. Additionally, the integration of Tesseract enhances the accuracy and speed of data extraction, contributing to a more efficient and effective recruitment process.

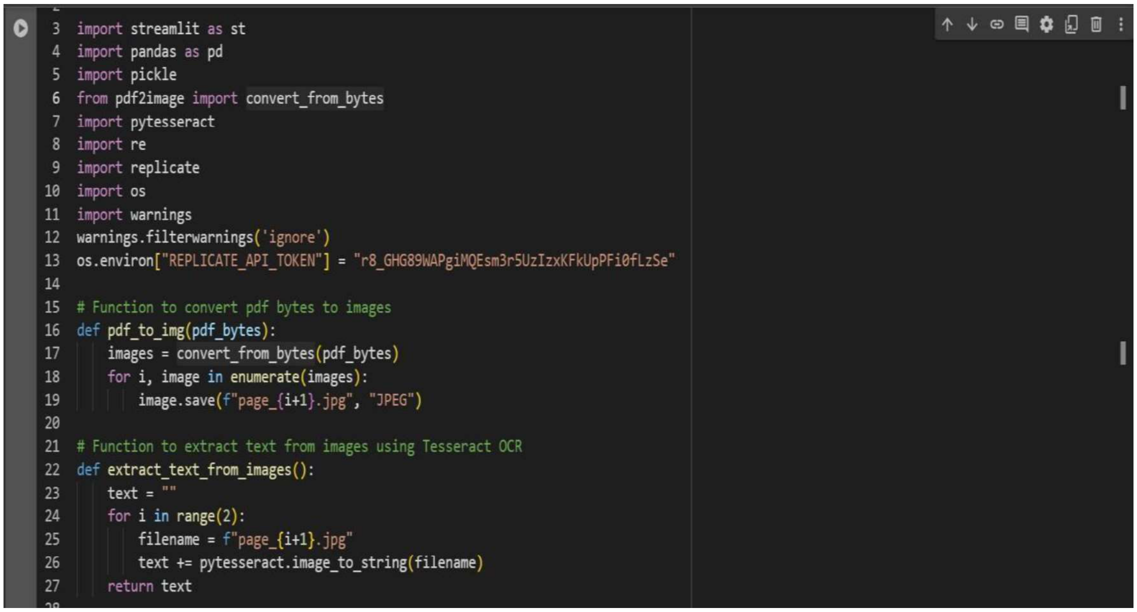
CHAPTER-5

CODING AND TESTING OF HICON AI

Coding and testing of the system involve the development and verification of the source code which is the primary component of the building the system, as well as the validation of the system's functionalities, performance, and security.

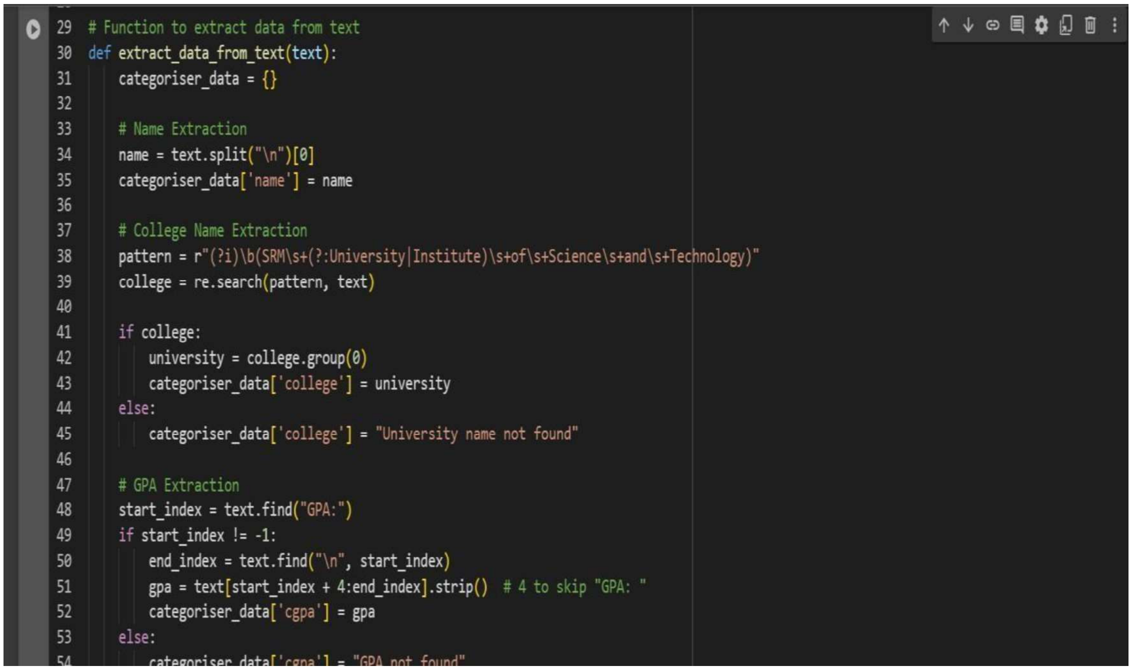
5.1 CODE SNAPSHOTS

Coding Snapshots of the source code are being captured and presented in below Figure 8 to Figure 13.



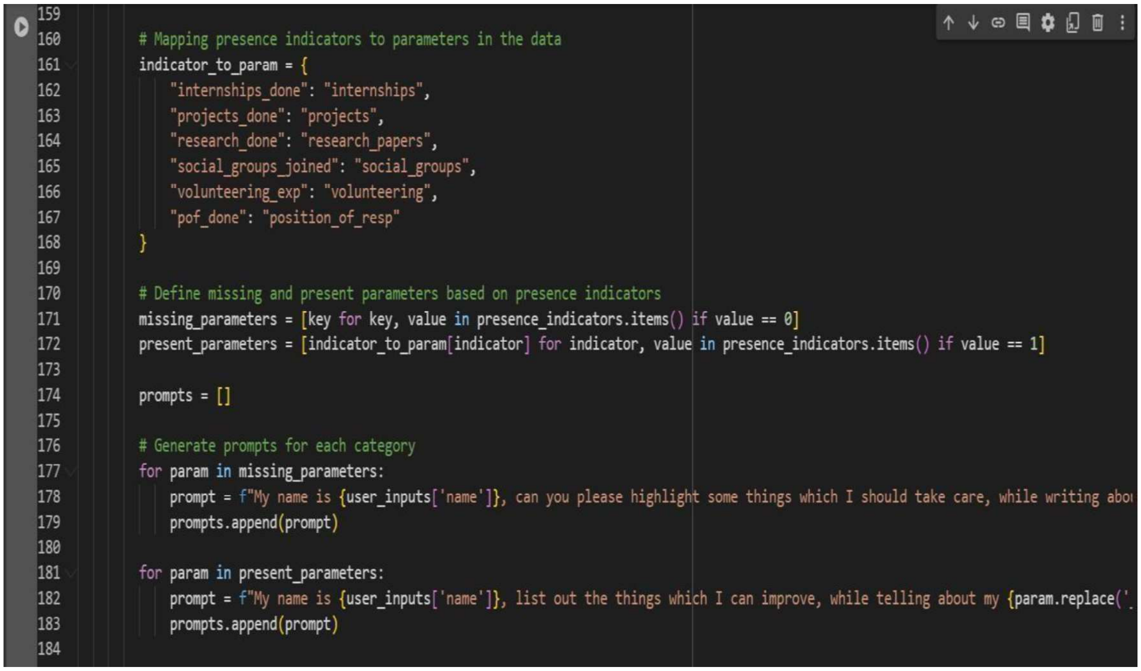
```python
import streamlit as st
import pandas as pd
import pickle
from pdf2image import convert_from_bytes
import pytesseract
import re
import replicate
import os
import warnings
warnings.filterwarnings('ignore')
os.environ["REPLICATE_API_TOKEN"] = "r8_GHG89WAPgiMQEsm3r5UzIzxKFkUpPFi0fLzSe"

# Function to convert pdf bytes to images
def pdf_to_img(pdf_bytes):
    images = convert_from_bytes(pdf_bytes)
    for i, image in enumerate(images):
        image.save(f"page_{i+1}.jpg", "JPEG")

# Function to extract text from images using Tesseract OCR
def extract_text_from_images():
    text = ""
    for i in range(2):
        filename = f"page_{i+1}.jpg"
        text += pytesseract.image_to_string(filename)
    return text
```

Fig8. Code Snapshot1



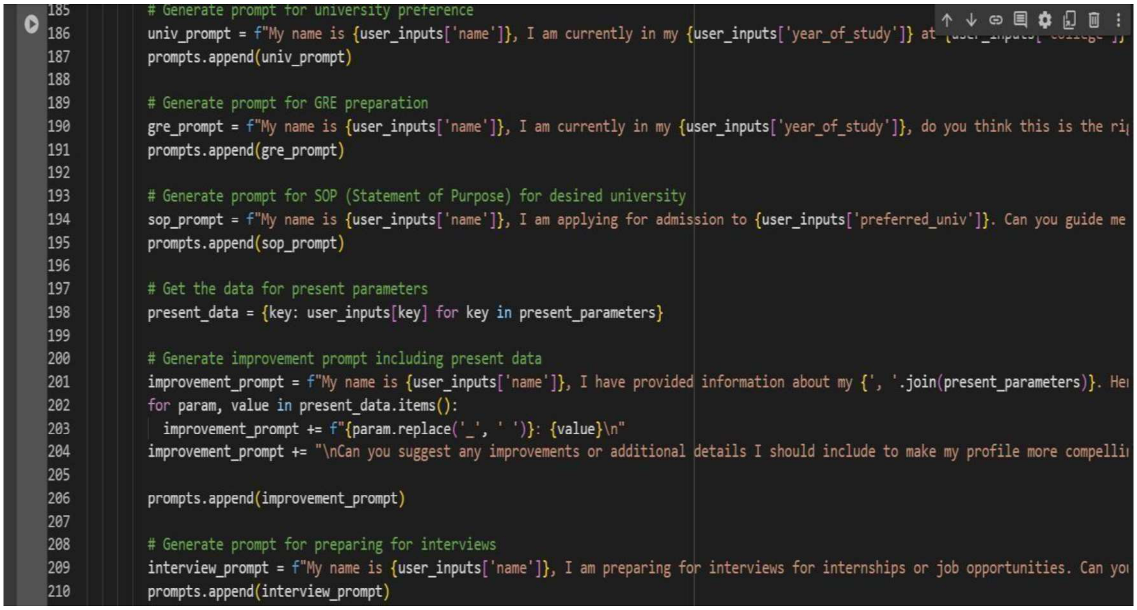
```python
# Function to extract data from text
def extract_data_from_text(text):
    categoriser_data = {}

    # Name Extraction
    name = text.split("\n")[0]
    categoriser_data['name'] = name

    # College Name Extraction
    pattern = r"(?i)\b(SRM\s+(?:University|Institute)\s+of\s+Science\s+and\s+Technology)"
    college = re.search(pattern, text)

    if college:
        university = college.group(0)
        categoriser_data['college'] = university
    else:
        categoriser_data['college'] = "University name not found"

    # GPA Extraction
    start_index = text.find("GPA:")
    if start_index != -1:
        end_index = text.find("\n", start_index)
        gpa = text[start_index + 4:end_index].strip()  # 4 to skip "GPA: "
        categoriser_data['cgpa'] = gpa
    else:
        categoriser_data['cgpa'] = "GPA not found"
```

Fig9. Code Snapshot2



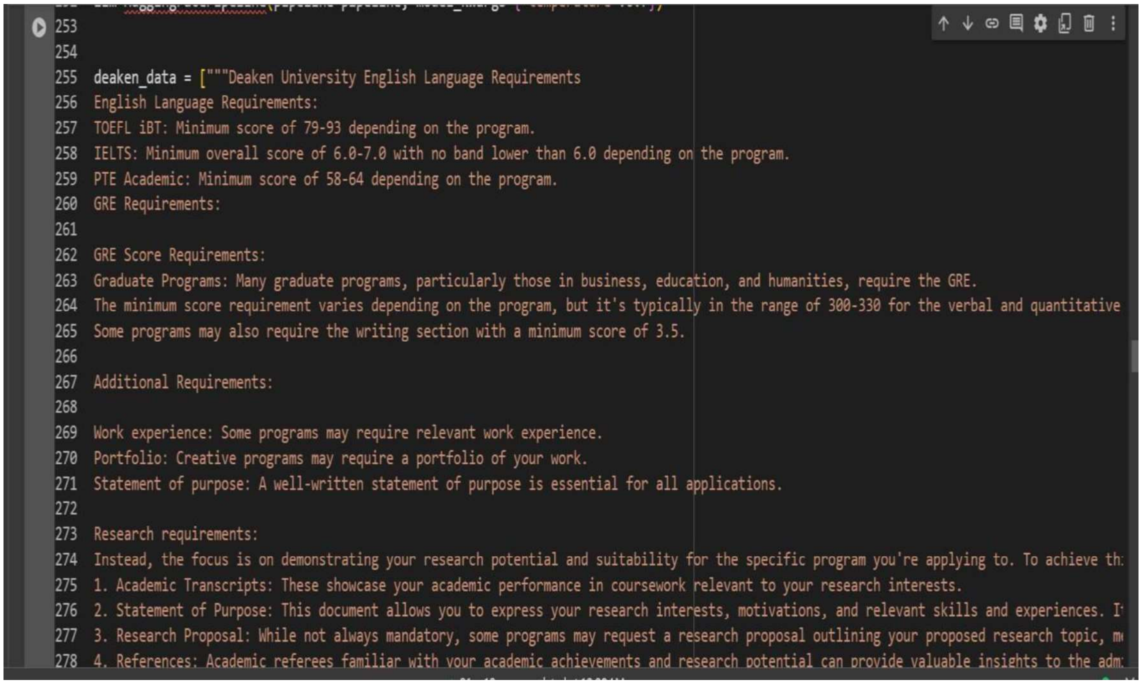
```python
        # Mapping presence indicators to parameters in the data
        indicator_to_param = {
            "internships_done": "internships",
            "projects_done": "projects",
            "research_done": "research_papers",
            "social_groups_joined": "social_groups",
            "volunteering_exp": "volunteering",
            "pof_done": "position_of_resp"
        }

        # Define missing and present parameters based on presence indicators
        missing_parameters = [key for key, value in presence_indicators.items() if value == 0]
        present_parameters = [indicator_to_param[indicator] for indicator, value in presence_indicators.items() if value == 1]

        prompts = []

        # Generate prompts for each category
        for param in missing_parameters:
            prompt = f"My name is {user_inputs['name']}, can you please highlight some things which I should take care, while writing abo
            prompts.append(prompt)

        for param in present_parameters:
            prompt = f"My name is {user_inputs['name']}, list out the things which I can improve, while telling about my {param.replace('_
            prompts.append(prompt)
```

Fig9. Code Snapshot3



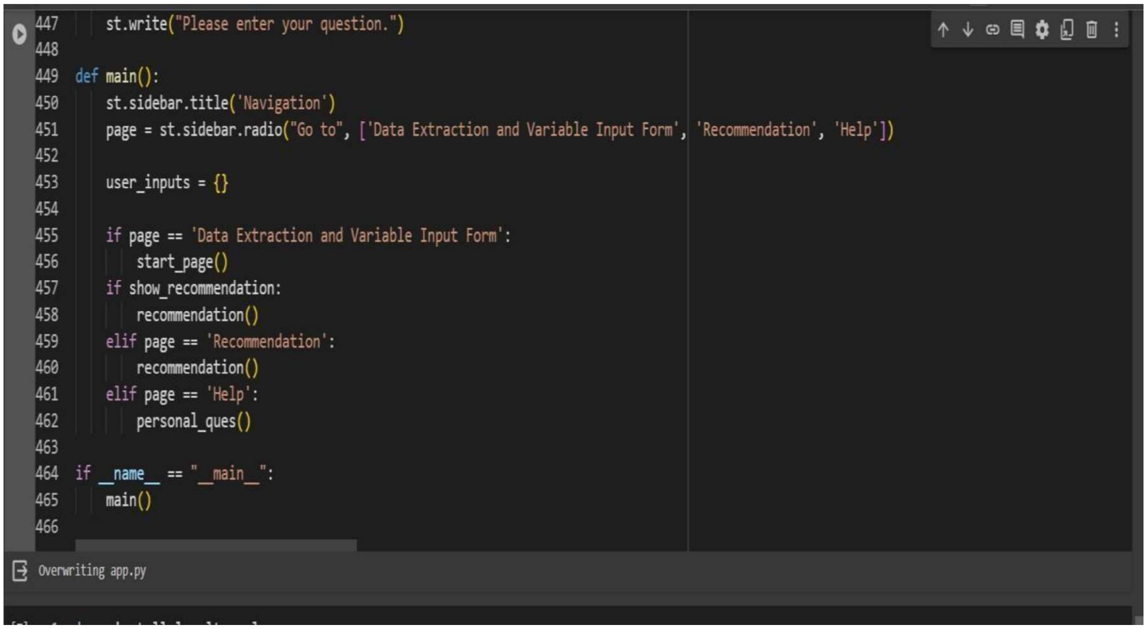
```
        # Generate prompt for university preference
        univ_prompt = f"My name is {user_inputs['name']}, I am currently in my {user_inputs['year_of_study']} at {user_inputs['college']}
        prompts.append(univ_prompt)

        # Generate prompt for GRE preparation
        gre_prompt = f"My name is {user_inputs['name']}, I am currently in my {user_inputs['year_of_study']}, do you think this is the ri
        prompts.append(gre_prompt)

        # Generate prompt for SOP (Statement of Purpose) for desired university
        sop_prompt = f"My name is {user_inputs['name']}, I am applying for admission to {user_inputs['preferred_univ']}. Can you guide me
        prompts.append(sop_prompt)

        # Get the data for present parameters
        present_data = {key: user_inputs[key] for key in present_parameters}

        # Generate improvement prompt including present data
        improvement_prompt = f"My name is {user_inputs['name']}, I have provided information about my {', '.join(present_parameters)}. He
        for param, value in present_data.items():
            improvement_prompt += f"{param.replace('_', ' ')}: {value}\n"
        improvement_prompt += "\nCan you suggest any improvements or additional details I should include to make my profile more compelli

        prompts.append(improvement_prompt)

        # Generate prompt for preparing for interviews
        interview_prompt = f"My name is {user_inputs['name']}, I am preparing for interviews for internships or job opportunities. Can yo
        prompts.append(interview_prompt)
```

Fig11. Code Snapshot4



```


deaken_data = ["""Deaken University English Language Requirements
English Language Requirements:
TOEFL iBT: Minimum score of 79-93 depending on the program.
IELTS: Minimum overall score of 6.0-7.0 with no band lower than 6.0 depending on the program.
PTE Academic: Minimum score of 58-64 depending on the program.
GRE Requirements:

GRE Score Requirements:
Graduate Programs: Many graduate programs, particularly those in business, education, and humanities, require the GRE.
The minimum score requirement varies depending on the program, but it's typically in the range of 300-330 for the verbal and quantitative
Some programs may also require the writing section with a minimum score of 3.5.

Additional Requirements:

Work experience: Some programs may require relevant work experience.
Portfolio: Creative programs may require a portfolio of your work.
Statement of purpose: A well-written statement of purpose is essential for all applications.

Research requirements:
Instead, the focus is on demonstrating your research potential and suitability for the specific program you're applying to. To achieve th
1. Academic Transcripts: These showcase your academic performance in coursework relevant to your research interests.
2. Statement of Purpose: This document allows you to express your research interests, motivations, and relevant skills and experiences. I
3. Research Proposal: While not always mandatory, some programs may request a research proposal outlining your proposed research topic, m
4. References: Academic referees familiar with your academic achievements and research potential can provide valuable insights to the adm
```

Fig12. Code Snapshot5



```
    st.write("Please enter your question.")

def main():
    st.sidebar.title('Navigation')
    page = st.sidebar.radio("Go to", ['Data Extraction and Variable Input Form', 'Recommendation', 'Help'])

    user_inputs = {}

    if page == 'Data Extraction and Variable Input Form':
        start_page()
    if show_recommendation:
        recommendation()
    elif page == 'Recommendation':
        recommendation()
    elif page == 'Help':
        personal_ques()

if __name__ == "__main__":
    main()

```

```
Overwriting app.py
```

Fig13. Code Snapshot6

5.2 SOURCE CODE

The source code of the system is being provided below and is carefully tested and verified.

```
%%writefile app.py

import streamlit as st
import pandas as pd
import pickle
from pdf2image import convert_from_bytes
import pytesseract
import re
import replicate
import os
import warnings
warnings.filterwarnings('ignore')
os.environ["REPLICATE_API_TOKEN"] =
"r8_RkEnvTZV8wSFTVoivgZzDEDDBp01GwO482iNK"
```

```
# Function to convert pdf bytes to images
def pdf_to_img(pdf_bytes):
    images = convert_from_bytes(pdf_bytes)
    for i, image in enumerate(images):
        image.save(f"page_{i+1}.jpg", "JPEG")

# Function to extract text from images using Tesseract OCR
def extract_text_from_images():
    text = ""
    for i in range(2):
        filename = f"page_{i+1}.jpg"
        text += pytesseract.image_to_string(filename)
    return text

# Function to extract data from text
def extract_data_from_text(text):
    categoriser_data = {}

    # Name Extraction
    name = text.split("\n")[0]
    categoriser_data['name'] = name

    # College Name Extraction
    pattern = r"(?i)\b(SRM\s+(?:University|Institute)\s+of\s+Science\s+and\s+Technology)"
    college = re.search(pattern, text)

    if college:
        university = college.group(0)
        categoriser_data['college'] = university
    else:
        categoriser_data['college'] = "University name not found"

    # GPA Extraction
```

```
start_index = text.find("GPA:")
if start_index != -1:
    end_index = text.find("\n", start_index)
    gpa = text[start_index + 4:end_index].strip()  # 4 to skip "GPA: "
    categoriser_data['cgpa'] = gpa
else:
    categoriser_data['cgpa'] = "GPA not found"

# Internships Section
start_index = text.find("LEADERSHIP & EXPERIENCE")
if start_index == -1:
    raise ValueError("Leadership & Experience section not found")

end_index = text.find("PROJECTS", start_index)
if end_index == -1:
    raise ValueError("Projects section not found")

section = text[start_index:end_index].strip()
categoriser_data['internships'] = section

# Projects Section
project_start_index = text.find("PROJECTS")
if project_start_index == -1:
    raise ValueError("projects section not found")

project_end_index = text.find("TECH SOCIETIES", project_start_index)
if project_end_index == -1:
    raise ValueError("tech societies section not found")

projects_section = text[project_start_index:project_end_index].strip()
categoriser_data['projects'] = projects_section

return categoriser_data
```

```
# Load the model
with open('model.pkl', 'rb') as file:
    loaded_model = pickle.load(file)



show_recommendation = False

def start_page():

    # Initialize variables
    category_person = None
    question_arr = None



    # Display file uploader for document file
    document_file = st.file_uploader("Upload document file", type=['pdf'])

    if document_file:
        # Convert pdf bytes to images
        pdf_to_img(document_file.getvalue())
        st.write("Document file processed and converted to images")

        # Extract text from images
        text = extract_text_from_images()

        # Extract data from text
        categoriser_data = extract_data_from_text(text)

        # Display the extracted data
        st.write("Extracted data:", categoriser_data)

        # Create a placeholder for the user inputs
        user_inputs = {}
        # Create a dictionary to store presence indicators
        presence_indicators = {
            'internships_done': 0,
            'projects_done': 0,
            'research_done': 0,
            'social_groups_joined': 0,
            'volunteering_exp': 0,
            'pof_done': 0
        }

        # Display the form for each variable
        st.title("Variable Input Form")
        input_variables = ['name', 'age', 'college', 'year_of_study', 'cgpa', 'preferred_univ',
'internships', 'projects',
                    'research_papers', 'social_groups', 'volunteering', 'position_of_resp']
        for variable in input_variables:
            # Check if the variable is present in the categoriser_data dictionary
            if variable in categoriser_data:
                default_value = categoriser_data[variable]
            else:
                default_value = ""

            # Create input field for the variable
            user_input = st.text_input(f"Enter value for {variable}:", value=default_value)

            # If input is empty, mark it as None
            if not user_input:
                user_inputs[variable] = ""
            else:
                user_inputs[variable] = user_input

            # Update presence indicators
            if variable == 'internships' and user_input != "":
                presence_indicators['internships_done'] = 1
            if variable == 'projects' and user_input != "":
                presence_indicators['projects_done'] = 1
            if variable == 'research_papers' and user_input != "":
                presence_indicators['research_done'] = 1
            if variable == 'social_groups' and user_input != "":
                presence_indicators['social_groups_joined'] = 1
            if variable == 'volunteering' and user_input != "":
                presence_indicators['volunteering_exp'] = 1
            if variable == 'position_of_resp' and user_input != "":
                presence_indicators['pof_done'] = 1

        # Mapping presence indicators to parameters in the data
        indicator_to_param = {
            "internships_done": "internships",
            "projects_done": "projects",
            "research_done": "research_papers",
            "social_groups_joined": "social_groups",
            "volunteering_exp": "volunteering",
            "pof_done": "position_of_resp"
        }

        # Define missing and present parameters based on presence indicators
        missing_parameters = [key for key, value in presence_indicators.items() if value == 0]
        present_parameters = [indicator_to_param[indicator] for indicator, value in presence_indicators.items() if value == 1]

        prompts = []

        # Generate prompts for each category
        for param in missing_parameters:
            prompt = f"My name is {user_inputs['name']}, can you please highlight some things which I should take care, while writing about my {param.replace('_', ' ')}, please be specific and just write everything in points. Also, give responses like you are talking to a real person and be polite while calling names and everything"
            prompts.append(prompt)

        for param in present_parameters:
            prompt = f"My name is {user_inputs['name']}, list out the things which I can improve, while telling about my {param.replace('_', ' ')}, please be specific and just write everything in points. Also, give responses like you are talking to a real person and be polite while calling names and everything"
            prompts.append(prompt)

        # Generate prompt for university preference
         univ_prompt = f"My name is {user_inputs['name']}, I am currently in my  {user_inputs['year_of_study']} at {user_inputs['college']}. I want to go in these {user_inputs['preferred_univ']} universities. Can you list a tailored structure of how to get into these universities, please be specific and just write everything in points. Also, give responses like you are talking to a real person and be polite while calling names and everything"
        prompts.append(univ_prompt)

        # Generate prompt for GRE preparation
        gre_prompt = f"My name is {user_inputs['name']}, I am currently in my {user_inputs['year_of_study']}, do you think this is the right time for me to prepare for GRE, and tell me is GRE necessary for {user_inputs['preferred_univ']}. If you think this is the right time to start, how should I prepare for it?, please be specific and just write everything in points. Also, give responses like you are talking to a real person and be polite while calling names and everything"
        prompts.append(gre_prompt)

        # Generate prompt for SOP (Statement of Purpose) for desired university
        sop_prompt = f"My name is {user_inputs['name']}, I am applying for admission to {user_inputs['preferred_univ']}. Can you guide me on how to write a compelling Statement of Purpose (SOP) specifically tailored for this university? Please include key points to cover and any specific requirements or tips from your experience. Also, give responses like you are talking to a real person and be polite while calling names and everything."
        prompts.append(sop_prompt)
        # Get the data for present parameters
        present_data = {key: user_inputs[key] for key in present_parameters}

        # Generate improvement prompt including present data
        improvement_prompt = f"My name is {user_inputs['name']}, I have provided information about my {', '.join(present_parameters)}. Here are the details:\n\n"
        for param, value in present_data.items():
          improvement_prompt += f"{param.replace('_', ' ')}: {value}\n"
        improvement_prompt += "\nCan you suggest any improvements or additional details I should include to make my profile more compelling for university applications or job opportunities? Please provide specific advice and suggestions. Also, give responses like you are talking to a real person and be polite while calling names and everything."

        prompts.append(improvement_prompt)

        # Generate prompt for preparing for interviews
        interview_prompt = f"My name is {user_inputs['name']}, I am preparing for interviews for internships or job opportunities. Can you provide tips and strategies for preparing effectively for technical and behavioral interviews? Please include common questions, how to approach technical assessments, and advice for presenting my experience and skills confidently. Also, give responses like you are talking to a real person and be polite while calling names and everything."
        prompts.append(interview_prompt)

        # Generate prompt for improving communication skills
        communication_prompt = f"My name is {user_inputs['name']}, I want to improve my communication skills, both written and verbal, to enhance my professional profile. Can you suggest resources, courses, or activities that can help me develop effective communication skills? Please include practical tips and exercises for improving clarity, coherence, and persuasiveness in communication. Also, give responses like you are talking to a real person and be polite while calling names and everything."
        prompts.append(communication_prompt)
```

```
        # Add a submit button
        if st.button("Submit"):
          # Create DataFrame from presence_indicators dictionary
          df = pd.DataFrame([presence_indicators])

          show_recommendation = True

          # Make prediction using loaded model
          category_person = loaded_model.predict(df)[0]
          question_arr = prompts

          # Store category_person and question_arr in session state
          st.session_state['category_person'] = category_person
          st.session_state['question_arr'] = question_arr

          return None

    return None

# model="meta-llama/Llama-2-7b-chat-hf"
# tokenizer=AutoTokenizer.from_pretrained(model)
# pipeline=transformers.pipeline(
#     "text-generation",
#     model=model,
#     tokenizer=tokenizer,
#     torch_dtype=torch.bfloat16,
#     trust_remote_code=True,
#     device_map="auto",
#     min_length=200,
#     max_length=1000,
#     do_sample=True,
#     top_k=10,
#     num_return_sequences=1,
#     eos_token_id=tokenizer.eos_token_id
#    )

# llm=HuggingFacePipeline(pipeline=pipeline, model_kwargs={'temperature':0.7})


# deaken_data = ["""Deaken University English Language Requirements
# English Language Requirements:
# TOEFL iBT: Minimum score of 79-93 depending on the program.
# IELTS: Minimum overall score of 6.0-7.0 with no band lower than 6.0 depending on the program.
# PTE Academic: Minimum score of 58-64 depending on the program.
# GRE Requirements:

# GRE Score Requirements:
# Graduate Programs: Many graduate programs, particularly those in business, education, and humanities, require the GRE.
# The minimum score requirement varies depending on the program, but it's typically in the range of 300-330 for the verbal and quantitative sections.
# Some programs may also require the writing section with a minimum score of 3.5.

# Additional Requirements:

# Work experience: Some programs may require relevant work experience.
# Portfolio: Creative programs may require a portfolio of your work.
# Statement of purpose: A well-written statement of purpose is essential for all applications.

# Research requirements:
# Instead, the focus is on demonstrating your research potential and suitability for the specific program you're applying to. To achieve this, they require several key documents:
# 1. Academic Transcripts: These showcase your academic performance in coursework relevant to your research interests.
# 2. Statement of Purpose: This document allows you to express your research interests, motivations, and relevant skills and experiences. It's your chance to highlight your specific contributions to the field and why you're a good fit for the program.
```

# 3. Research Proposal: While not always mandatory, some programs may request a research proposal outlining your proposed research topic, methodology, and expected outcomes. This helps assess your research skills and alignment with the program's focus.

# 4. References: Academic referees familiar with your academic achievements and research potential can provide valuable insights to the admissions committee.

# 5. Additional documents: Depending on the program, you might need to submit work samples, publications (if any), evidence of research experience, or other relevant materials.

# Statement of Purpose:

# Template:

# Introduction:

# Briefly introduce yourself and state your intention to apply to a specific program at Deakin University.

# Mention the semester and year you are applying for.

# Academic Background and Achievements:

# Highlight your relevant academic background, including your degree(s), relevant coursework, and your overall academic performance.

# Mention any academic awards, scholarships, or distinctions you have received.

# Research Experience and Interests:

# If applicable, describe your research experience (e.g., internships, lab work, independent projects).

# Clearly articulate your research interests and align them with the specific program you are applying to.

# Mention specific faculty members whose research aligns with your interests and how you see yourself contributing to their work.

# Motivations and Goals:

# Explain your motivations for pursuing this program at Deakin University. What attracts you to the program and the university?

# Clearly state your short-term and long-term career goals and how this program will help

```
you achieve them.
# Conclusion:

# Briefly summarize your key strengths and qualifications.
# Reiterate your enthusiasm for the program and express your gratitude for the committee's
time and consideration.
# Tips:

# Be specific and relevant: Tailor your statement to the specific program you are applying to.
Highlight skills and experiences directly related to the program's requirements and research
focus.
# Demonstrate your passion: Show your genuine interest in the program and the field of
research. Use specific examples and evidence to support your claims.
# Be concise and clear: Aim for a clear and concise statement within the suggested word
limit. Use strong verbs and avoid unnecessary jargon.
# Proofread carefully: Ensure your statement is free of grammatical errors and typos.
# """]

# vectorstore = FAISS.from_texts(
# deaken_data, embedding=HuggingFaceEmbeddings()
# )

# # Create a retriever from the vectorstore
# retriever = vectorstore.as_retriever()

# # Create a new prompt template using the HuggingFace embeddings
# template = """Answer the question based only on the following context: {context}

# Question: {question}
# """

# min_tokens = 256

# prompt = ChatPromptTemplate.from_template(template, min_tokens=min_tokens)
# chain = (
#     {"context": retriever, "question": RunnablePassthrough()}
#     | prompt
#     | llm
#     | StrOutputParser()
# )


def question_and_answer(question, answer):
    """Create a question and answer card with a dropdown icon.

    Args:
        question: The question to be displayed on the card.
        answer: The answer to be displayed on the card.
    """
    with st.expander(question):
        st.write(answer)


def recommendation():
  st.title("Recommendation")
  st.write("## Question and Answer")

  if 'category_person' in st.session_state and 'question_arr' in st.session_state:
    category_person = st.session_state['category_person']
    question_arr = st.session_state['question_arr']
    st.write("Category: ", category_person)
    st.write("Question list: ", question_arr)

    # Add your questions and answers here
    questions = question_arr  # Assuming question_arr is the list of prompts
    answers = []
    # Track current question index and initialize to 0
    current_question_index = st.session_state.get('current_question_index', 0)

    # Display logic with button handling
    if current_question_index < len(questions):
      question_prompt = questions[current_question_index]

      # Generate answer using large language model
      output = replicate.run(
          "meta/llama-2-13b-
chat:f4e2de70d66816a838a89eeeb621910adffb0dd0baba3976c96980970978018d",
          input={"prompt": question_prompt,
                  "temperature": 0.75,
                  "max_new_tokens": 2000,
                  "min_new_tokens": 256}
      )

      response_text = "".join(item for item in output)
      answers.append(response_text)

      # Display the question and answer
      question_and_answer(question_prompt, response_text)

      # Button layout with Prev and Next
      col1, col2 = st.columns(2)
      if current_question_index > 0:
        if col1.button("Previous"):
          current_question_index -= 1
      if current_question_index < len(questions) - 1:
        if col2.button("Next"):
          current_question_index += 1

      # Update session state with the new index
      st.session_state['current_question_index'] = current_question_index
```

```
  else:
        st.write("Please fill out the form on the previous page.")



def personal_ques():
  st.title("Help Page")

  # User input for question
  question = st.text_area("Ask your question here:")

  # Generate answer using large language model (replace with your specific API call)
  if question:
    output = replicate.run(
        "meta/llama-2-13b-
chat:f4e2de70d66816a838a89eeeb621910adffb0dd0baba3976c96980970978018d",
        input={"prompt": question,
               "temperature": 0.75,
               "max_new_tokens": 2000,
               "min_new_tokens": 256}
    )

    response_text = "".join(item for item in output)
    question_and_answer("Your Question:", question)
    question_and_answer("Answer:", response_text)

  else:
    st.write("Please enter your question.")

def main():
    st.sidebar.title('Navigation')
    page = st.sidebar.radio("Go to", ['Data Extraction and Variable Input Form',
'Recommendation', 'Help'])
```

```
    user_inputs = {}

    if page == 'Data Extraction and Variable Input Form':
        start_page()
    if show_recommendation:
        recommendation()
    elif page == 'Recommendation':
        recommendation()
    elif page == 'Help':
        personal_ques()

if _name_ == "_main_":
    main()
```

5.3 TESTING

Testing is an important step in the software development and quality assurance processes. It entails the systematic examination of software programs or systems to detect and correct flaws, verify that they meet the intended requirements, and provide a high degree of quality. Testing acts as a quality control tool, assisting in the prevention of errors, improving software reliability, and increasing user happiness.

5.3.1 ABOUT TESTING

The primary purposes of testing are as follows:

- **Identify Bugs and faults:** Testing assists in detecting and reporting software faults, errors, or inconsistencies that may jeopardize an application's operation or performance.

- **Ensure Correct Functionality:** Testing ensures that the product works as it should, fulfilling the requirements and user expectations.

- **Improve dependability:** Testing contributes to software's dependability and robustness by ensuring that it functions consistently under a variety of settings and scenarios.

- **Improve Security:** Security testing identifies flaws and weaknesses in software that hostile actors could exploit.

- **Optimize Performance:** Performance testing assesses a software application's speed, responsiveness, and scalability to ensure it works properly under predicted workloads.

- **Validate User Experience:** User interface (UI) testing focuses on the visual characteristics and usability of software to ensure that it provides an outstanding user experience.

- **Achieve Compliance:** Certain sectors and applications demand strict adherence to specific norms and standards, and testing assures that these requirements are met.

Iterative testing should begin early in the software development life cycle and continue throughout the project. It is critical for quickly identifying errors, lowering development costs, and providing a high-quality product to end customers. Effective testing contributes to the overall success of a project by helping to develop trust in software applications.

5.3.2 TEST CASES

Thorough testing is required in the field of AI counseling to ensure that AI-driven counselors can give users with accurate, dependable, and supportive advice. Various test cases are used to validate the functionality and usefulness of these AI counselors. Some of the essential test cases in AI counseling system testing include:

- **Category Predictor Test Case:** The AI counselor's capacity to correctly recognize and categorize a user's data is evaluated in this test case. It determines if the AI system can differentiate between various conditions such as their internships, projects, research paper, social activities, volunteering experience and positions of responsibility.

- **Test Case for Parameter Validation:** It guarantees that the user's data is correctly processed and routed to the proper settings within the AI system.

- **Test Case for a RAG:** It is critical to test and update the knowledge base of RAG. Thus, increases the credibility and fairness of the output generated by the RAG which helps in getting rid of hallucination of the chatbot.

- **Test Case for Prompt-Based Response:** One of an AI counselor's primary functions is to reply effectively to user prompts. This test case assesses the counselor's ability to respond in a relevant, empathic, and supportive manner in response to the user's feedback or cues. It guarantees that the AI counselor can have meaningful interaction.

CHAPTER-6

RESULT AND DISCUSSION

The "Results and Discussion" section of a report presents and interprets the findings of the project. It typically begins by summarizing the key results of the study and then provide a detailed analysis and discussion of those results.

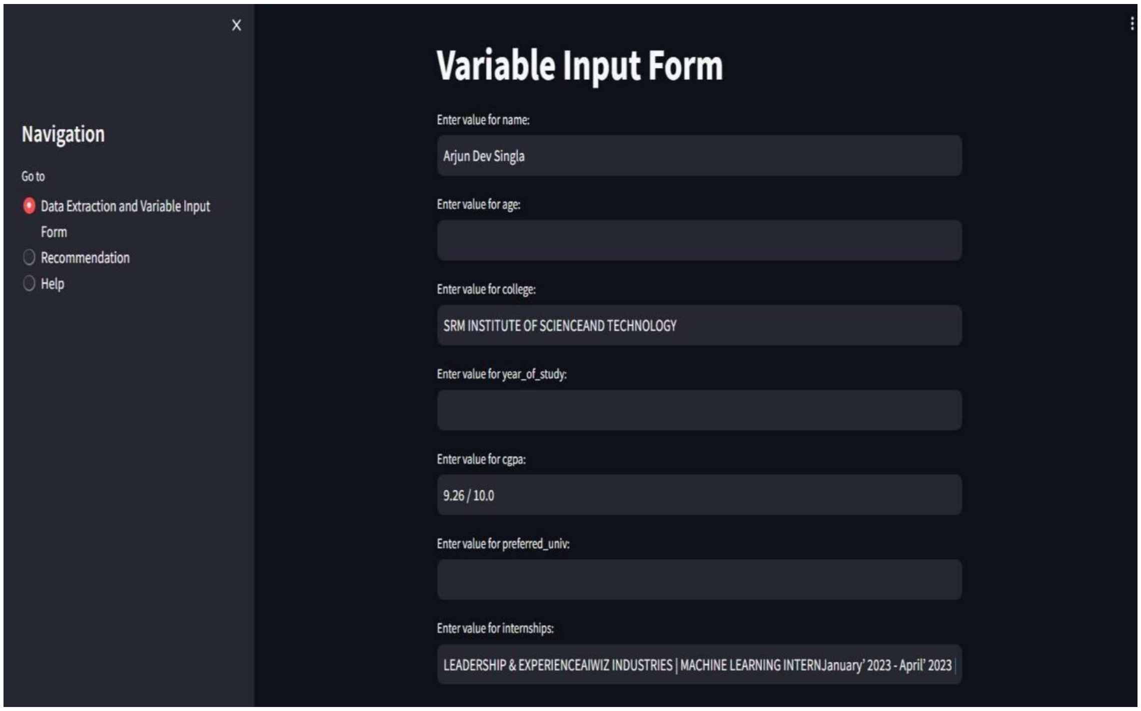
6.1 DATA COLLECTION

In the preliminary stage, HICON AI undertakes the vital task of gathering user information to tailor its counseling services effectively. This process encompasses a diverse range of data points aimed at constructing a holistic understanding of the user's profile. Initially, the system collects essential personal details such as the user's name, age, and their current academic standing. Additionally, it seeks insights into the user's academic journey, including their preferred universities and relevant academic achievements.

Moving beyond the academic realm, HICON AI delves into the user's extracurricular engagements and social activities. This entails compiling data on the user's involvement in various endeavors such as internships, projects, and research papers authored. By understanding the breadth and depth of the user's academic pursuits, the system can provide tailored guidance that aligns with their educational goals and aspirations.

Moreover, HICON AI recognizes the significance of holistic development and hence endeavors to capture details regarding the user's participation in social activities. This includes gathering information on volunteering experiences and any positions of responsibility held by the user. By considering these aspects alongside academic achievements, HICON AI ensures that its counseling services encompass a comprehensive understanding of the user's interests, strengths, and aspirations, thus facilitating personalized and impactful guidance.

The figure 14 below shows the data collection of the students by the HICON AI.



Variable Input Form

Enter value for name:
Arjun Dev Singla

Enter value for age:

Enter value for college:
SRM INSTITUTE OF SCIENCEAND TECHNOLOGY

Enter value for year_of_study:

Enter value for cgpa:
9.26 / 10.0

Enter value for preferred_univ:

Enter value for internships:
LEADERSHIP & EXPERIENCEAIWIZ INDUSTRIES | MACHINE LEARNING INTERNJanuary' 2023 - April' 2023 |

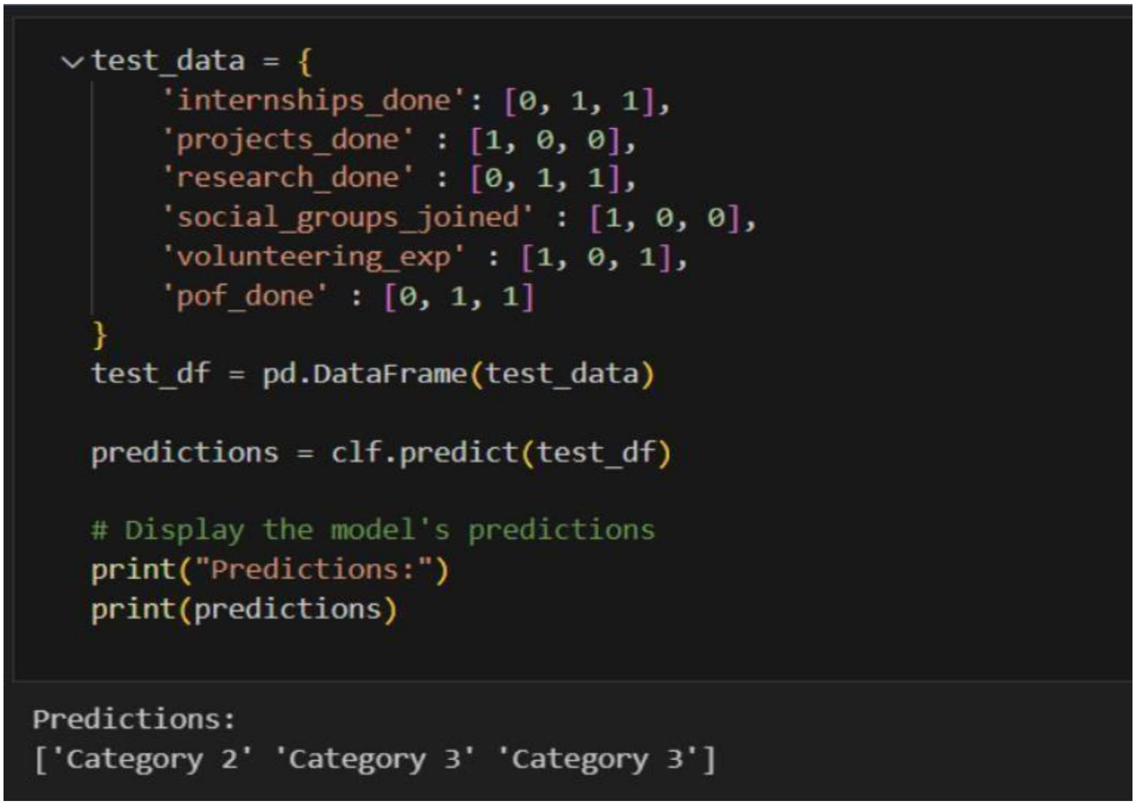
Fig.14 Data Collection

6.2 CATEGORIZING

Once the requisite student data is obtained, our focus shifts to employing our categorizer model. This model is designed to predict the user's category by leveraging the data provided in the data dictionary. By analyzing the collected information in accordance with predetermined criteria, the categorizer model ensures accurate categorization, facilitating efficient and tailored guidance for the student's specific needs and aspirations.

- **Category 1:** Uninformed users who are examining the domain's features.
- **Category 2:** People who want to advance their professional and academic abilities.
- **Category 3:** Individuals looking to fill in the gaps in their profiles.
- **Category 4:** People who want to stay abreast of current events and improve their chance of getting into the universities of their choice.

The below figure 15 shows the categorization of the students.



```
test_data = {
        'internships_done': [0, 1, 1],
        'projects_done' : [1, 0, 0],
        'research_done' : [0, 1, 1],
        'social_groups_joined' : [1, 0, 0],
        'volunteering_exp' : [1, 0, 1],
        'pof_done' : [0, 1, 1]
    }
    test_df = pd.DataFrame(test_data)

    predictions = clf.predict(test_df)

    # Display the model's predictions
    print("Predictions:")
    print(predictions)
```

```
Predictions:
['Category 2' 'Category 3' 'Category 3']
```

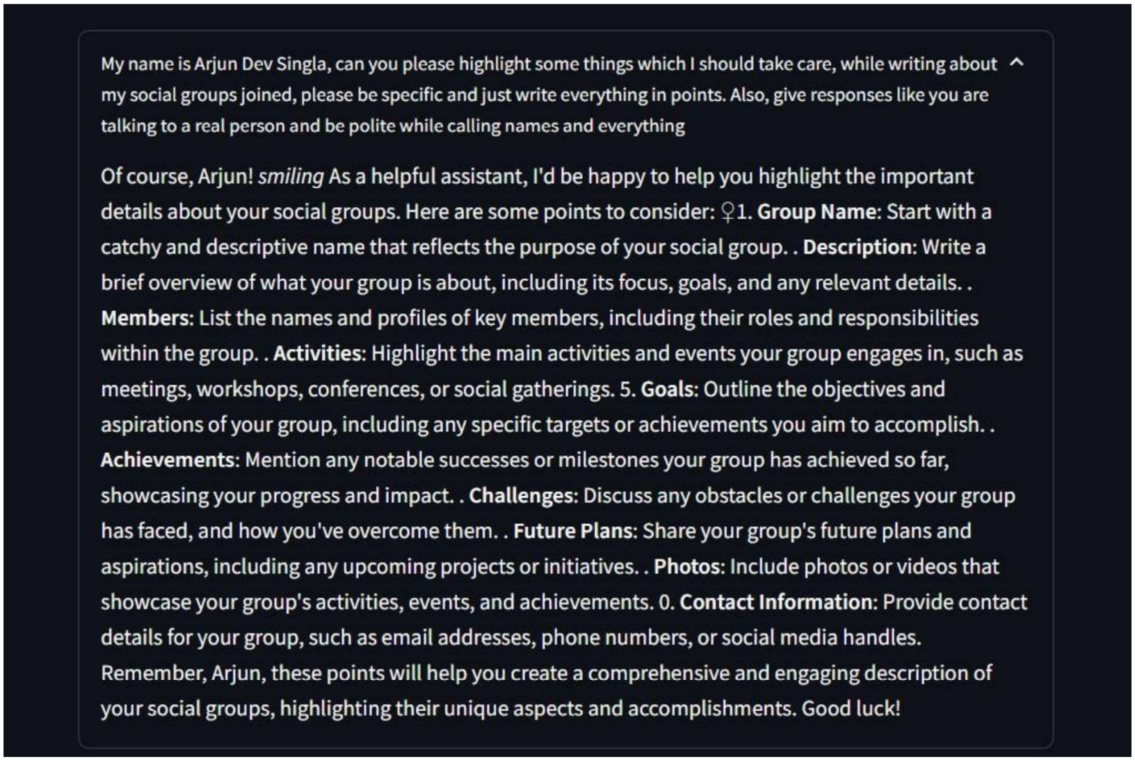
Fig.15 Categorization

6.3 RESPONSES

After gathering the requisite student data, our focus shifts towards leveraging our categorizer model, enriched by Retrieval-Augmented Generation (RAG), to discern the student's category effectively. This advanced model analyzes the collected data against predefined parameters, utilizing RAG's capabilities to extract relevant insights from vast knowledge repositories. Once the student's category is determined, we proceed to tailor prompts specific to each category, incorporating student-specific parameters obtained earlier. This personalized approach ensures that responses are finely tuned to meet the individual needs and aspirations of the student.

Subsequently, with the model in action, we generate responses that are not only personalized but also enriched with accurate and relevant information. RAG enhances this process by synthesizing contextual responses grounded in authoritative sources, ensuring the credibility and depth of the guidance provided. By seamlessly integrating user-specific parameters and category-tailored prompts, our system delivers responses that resonate with the user's academic and career objectives.

Figure 16. shows the detailed response generated by the LLM.



My name is Arjun Dev Singla, can you please highlight some things which I should take care, while writing about my social groups joined, please be specific and just write everything in points. Also, give responses like you are talking to a real person and be polite while calling names and everything

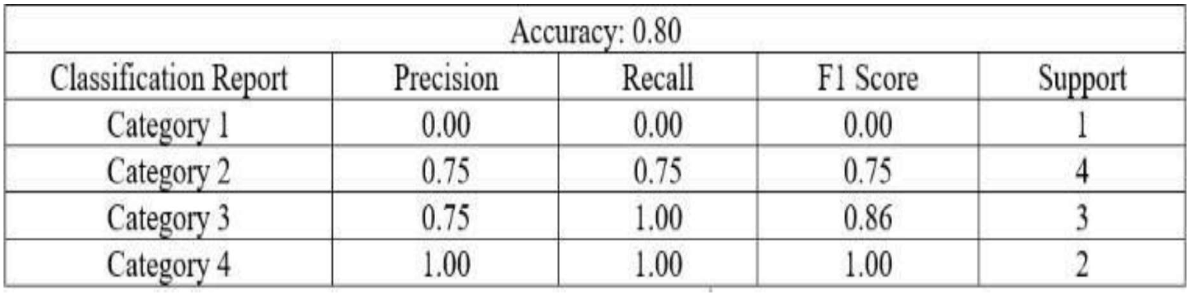
Of course, Arjun! *smiling* As a helpful assistant, I'd be happy to help you highlight the important details about your social groups. Here are some points to consider: ♀1. **Group Name**: Start with a catchy and descriptive name that reflects the purpose of your social group. . **Description**: Write a brief overview of what your group is about, including its focus, goals, and any relevant details. . **Members**: List the names and profiles of key members, including their roles and responsibilities within the group. . **Activities**: Highlight the main activities and events your group engages in, such as meetings, workshops, conferences, or social gatherings. 5. **Goals**: Outline the objectives and aspirations of your group, including any specific targets or achievements you aim to accomplish. . **Achievements**: Mention any notable successes or milestones your group has achieved so far, showcasing your progress and impact. . **Challenges**: Discuss any obstacles or challenges your group has faced, and how you've overcome them. . **Future Plans**: Share your group's future plans and aspirations, including any upcoming projects or initiatives. . **Photos**: Include photos or videos that showcase your group's activities, events, and achievements. 0. **Contact Information**: Provide contact details for your group, such as email addresses, phone numbers, or social media handles. Remember, Arjun, these points will help you create a comprehensive and engaging description of your social groups, highlighting their unique aspects and accomplishments. Good luck!

Fig. 16 Detailed Response

6.4 ACCURACY METRICES

The Accuracy metrics for the categorizer, presented in Table 4, outline the performance across four categories. Utilizing a random forest algorithm, the metrics include Precision, Recall, and F1 Score for each category.

Discussing the evaluation of the LLM, we utilize specific tools and comparisons. One benchmark used to assess the truthfulness of the LLM is TruthfulQA, where the score for LLaMA-2 7B is 45.2%, indicating satisfactory performance. While there are other evaluation benchmarks available, they are more meaningful when the LLM is fine-tuned to a specific dataset. Since our model was not fine-tuned, we adjusted the prompts to optimize the output quality.



Accuracy: 0.80

Classification Report	Precision	Recall	F1 Score	Support
Category 1	0.00	0.00	0.00	1
Category 2	0.75	0.75	0.75	4
Category 3	0.75	1.00	0.86	3
Category 4	1.00	1.00	1.00	2

Table 4. Accuracy Metrices

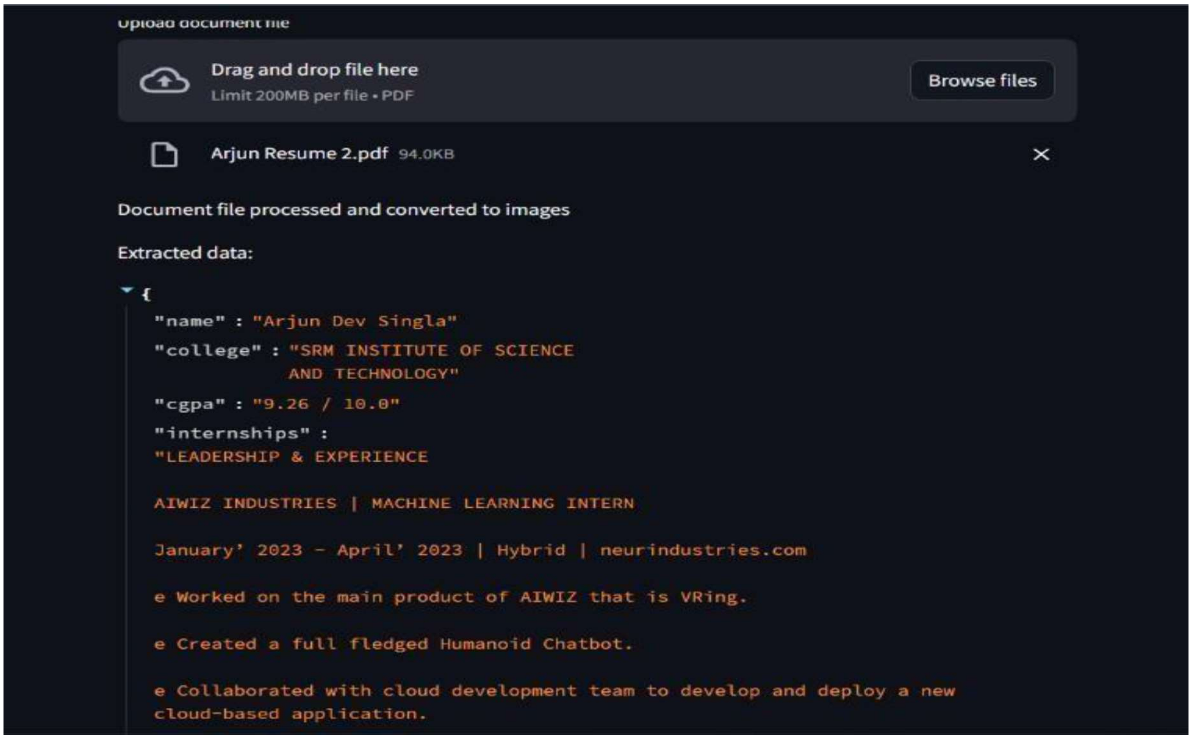
CHAPTER-7

CONCLUSION AND FUTURE SCOPE

"HICON AI" represents a cutting-edge model poised to serve as a valuable resource for students aspiring to study abroad. With its focus on user-centricity, it offers an effective solution for various industries. Leveraging large language models and Natural Language Processing techniques, "HICON AI" delivers personalized guidance to students preparing for higher education. By reducing reliance on costly external counselors through targeted advice and support, it aims to democratize access to higher education, making it more affordable. While "HICON AI" is still in its nascent stages, advancements in technology promise to revolutionize the student experience, reshaping the landscape of higher education counseling.

Moreover, there is untapped potential within the AI counseling domain. Strategies such as fine-tuning the model with college-specific data and implementing a peer review system for LLM-generated responses can enhance efficiency. Additionally, incorporating various score predictors for evaluating components like Statements of Purpose and Letters of Recommendation could further enrich the model's capabilities.

The figure 17 and figure 18 show the various features of the HICON AI.



Upload document file

Drag and drop file here

Limit 200MB per file • PDF

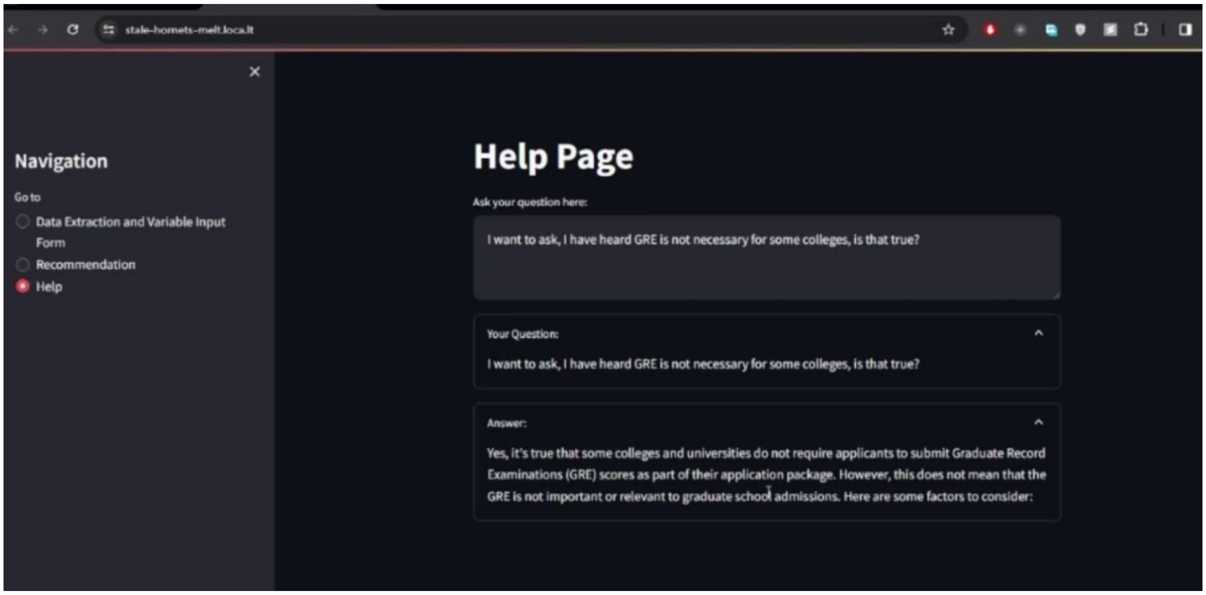
Arjun Resume 2.pdf 94.0KB

Document file processed and converted to images

Extracted data:

```
{
  "name" : "Arjun Dev Singla"
  "college" : "SRM INSTITUTE OF SCIENCE
              AND TECHNOLOGY"
  "cgpa" : "9.26 / 10.0"
  "internships" :
  "LEADERSHIP & EXPERIENCE

  AIWIZ INDUSTRIES | MACHINE LEARNING INTERN

  January’ 2023 - April’ 2023 | Hybrid | neurindustries.com

  e Worked on the main product of AIWIZ that is VRing.

  e Created a full fledged Humanoid Chatbot.

  e Collaborated with cloud development team to develop and deploy a new
  cloud-based application.
```

Fig17. Resume Screener



Navigation

Go to

○ Data Extraction and Variable Input Form

○ Recommendation

● Help

Help Page

Ask your question here:

I want to ask, I have heard GRE is not necessary for some colleges, is that true?

Your Question:

I want to ask, I have heard GRE is not necessary for some colleges, is that true?

Answer:

Yes, it's true that some colleges and universities do not require applicants to submit Graduate Record Examinations (GRE) scores as part of their application package. However, this does not mean that the GRE is not important or relevant to graduate school admissions. Here are some factors to consider:

Fig18. Personalized Advice (Response)

REFERENCES

[1] Philip Feldman, James R. Foulds, and Shimie Pan: RAGged Edges: The Double-Edged Sword of Retrieval-Augmented Chatbots, 2024

[2] Bongsu Kang, Jundong Kim, Tae-Rim Yun, Chang-Eop Kim: Prompt-RAG: Pioneering Vector Embedding-Free Retrieval-Augmented Generation in Niche Domains, Exemplified by Korean Medicine,2024

[3] Jaewoong Kim, Moohong Min: From RAG to QA-RAG: Intergrating Generative AI for Pharmaceutical Regulatory Complaince Process, 2023

[4] Yuhao Dan, Zhikai Lei, Yiyang Gu, Yong Li, Jianghao Yin, Jiaju Lin, Linhao Ye, Zhiyan Tie, Yougen Zhou, Yilei Wang, Aimin Zhou, Ze Zhou, Qin Chen, Jie Zhou, Liang He, and Xipeng Qiu. Educhat: A large-scale language model-based chatbot system for intelligent education, 2023

[5] Nitirajsingh Sandu, Ergun Gide: Adoption of AI-Chatbots to Enhance Student Learning Experience in Higher Education in India,2019

[6] Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects. 2023.

[7] Debarag Banerjee, Pooja Singh, Arjun Avadhanam, and Saksham Srivas-tava. Benchmarking llm powered chatbots: Methods and metrics. arXiv preprint arXiv:2308.04624, 2023.

[8] IBM Documentation. Socket programming. 2021.

[9] Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects. 2023.

[10] Muskan Sharma, Anita Kumari, and Jyotsna. Ai-based deep learning chatbot for career and personal mentorship. In 2023 IEEE 3rd International Conference on Technology, Engineering, Management for Societal impact using Marketing, Entrepreneurship and Talent (TEMSMET), pages 1–6, 2023.

[11] Hao Li Siddharth Misra. Noninvasive fracture characterization based on the classification of sonic wave travel times. 2020.

[12] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne, Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[13] A. Thodge, H. M and C. P. D. Cyril, "Web-based chatbot for basic financial and mortgage services," 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN), Vellore, India, 2023, pp. 1-5, doi: 10.1109/ViTECoN58111.2023.10157424.

[14] Saha, Baibhab and Devi, Dr.G.Renuka and Banerjee, Himanshu, Healthcare Chatbot Using DecisionTree Algorithm (October 14, 2022). Available at SSRN: https://ssrn.com/abstract=4247821 or http://dx.doi.org/10.2139/ssrn.4247821

[15] Gaayathri, R. S., S. Suman Rajest, Varun Kumar Nomula, and R. Regin. "FMDB Transactions on Sustainable Computer Letters." (2023).

[16] Hiran H. Lathabai, Abhirup Nandy, Vivek Kumar Singh, "Institutional collaboration recommendation: An expertise-based framework using NLP and network analysis, Expert Systems with Applications," ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2022.118317 (2022)

APPENDIX A

STREAMLIT

Streamlit is an open-source Python library that simplifies the process of creating web applications for data science and machine learning projects. It allows users to build interactive and customizable web apps with minimal code, making it accessible to both beginners and experienced developers.

Key features of Streamlit include:

- **Simplicity:** Streamlit's syntax is straightforward and intuitive, allowing users to create interactive web apps using familiar Python scripting. Developers can quickly prototype and deploy applications without the need for extensive web development knowledge.

- **Interactivity:** Streamlit enables the creation of dynamic and responsive user interfaces, allowing users to interact with visualizations, widgets, and data inputs in real-time. This fosters exploration and analysis of data through intuitive user interactions.

- **Flexibility:** Streamlit offers a wide range of built-in components and widgets, including sliders, buttons, and text inputs, to customize the appearance and functionality of web apps. Users can also integrate popular Python libraries such as Pandas, Matplotlib, and Plotly to create rich and interactive data visualizations.

- **Deployment:** Streamlit provides seamless deployment options, allowing users to deploy their web apps on various platforms, including Streamlit Sharing, Heroku, and AWS. With just a few commands, developers can share their applications with colleagues, clients, or the broader community.

paper

ORIGINALITY REPORT

8%	5%	1%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to King's College Student Paper	2%
2	arxiv.org Internet Source	1%
3	huggingface.co Internet Source	1%
4	www.coursehero.com Internet Source	<1%
5	medium.com Internet Source	<1%
6	iemtronics.org Internet Source	<1%
7	link.springer.com Internet Source	<1%
8	www.researchgate.net Internet Source	<1%
9	Submitted to Colorado State University, Global Campus Student Paper	<1%

10	blogs.lawrence.edu Internet Source	<1 %
11	digital.library.unt.edu Internet Source	<1 %
12	Submitted to Texas A & M University, Kingville Student Paper	<1 %
13	docslib.org Internet Source	<1 %
14	Submitted to University of North Texas Student Paper	<1 %
15	python.langchain.com Internet Source	<1 %
16	www.grin.com Internet Source	<1 %
17	Submitted to SRM University Student Paper	<1 %
18	aclanthology.org Internet Source	<1 %
19	www.rdv.vic.gov.au Internet Source	<1 %
20	Submitted to Queen's University of Belfast Student Paper	<1 %
21	Submitted to Universiti Teknologi Malaysia Student Paper	<1 %

22	Submitted to Temple University Student Paper	<1 %
23	uwspace.uwaterloo.ca Internet Source	<1 %
24	www.studymode.com Internet Source	<1 %
25	Submitted to City University Student Paper	<1 %
26	text-id.123dok.com Internet Source	<1 %
27	pypi.org Internet Source	<1 %
28	"Computational Sciences and Sustainable Technologies", Springer Science and Business Media LLC, 2024 Publication	<1 %
29	github.com Internet Source	<1 %
30	vdocuments.mx Internet Source	<1 %
31	www.mdpi.com Internet Source	<1 %
32	Severin Rodler, Conner Ganjavi, Pieter De Backer, Vasileios Magoulianitis et al.	<1 %

"Generative artificial intelligence in surgery", Surgery, 2024

Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

APPENDIX B

PLAGIARISM REPORT

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES

(To be attached in the dissertation/ project report)

1	Name of the Candidate **(IN BLOCK LETTERS)**	ARJUN DEV SINGLA SHASHANK TRIPATHI
2	Address of the Candidate	as9301@srmist.edu.in st3158@srmist.edu.in
3	Registration Number	RA2011031010074 RA2011031010096
4	Date of Birth	16-12-2002 25-09-2001
5	Department	Networking and Communications
6	Faculty	Engineering and Technology, School of Computing
7	Title of the Dissertation/Project	HICON AI
8	Whether the above project /dissertationis done by	~~Individual~~ or group : (Strike whichever is not applicable) No. of Group Members: 02 : Name: Arjun Dev Singla (RA2011031010074) Shashank Tripathi (RA2011031010096)
9	Name and address of the Supervisor / Guide	Dr. Helen Victoria A **Mail ID:** helenvia@srmist.edu.in **Mobile Number:** 9790836572
10	Name and address of Co-Supervisor / Co- Guide (if any)	Dr. P. Supraja **Mail ID:** suprajap@srmist.edu.in **Mobile Number:** 9786331333 Dr. G. Saranya **Mail ID:** saranyag3@srmist.edu.in **Mobile Number:** 9176583416

z

11	Software Used			
12	Date of Verification			
13	**Plagiarism Details: (to attach the final report from the software)**			
Chapter	**Title of the Chapter**	**Percentage of similarity index (including self citation)**	**Percentage of similarity index (Excluding self-citation)**	**% of plagiarismsafter excluding Quotes, Bibliography, etc.,**
1	INTRODUCTION			
2	LITERATURE REVIEW			
3	SYSTEM ARCHITECTURE			
4	METHODOLOGY			
5	CODING AND TESTING			
6	RESULT AND DISCUSSION			
7	CONCLUSION AND FUTURE SCOPE			
8	OVERALL			
	Abstract			
	Appendices			

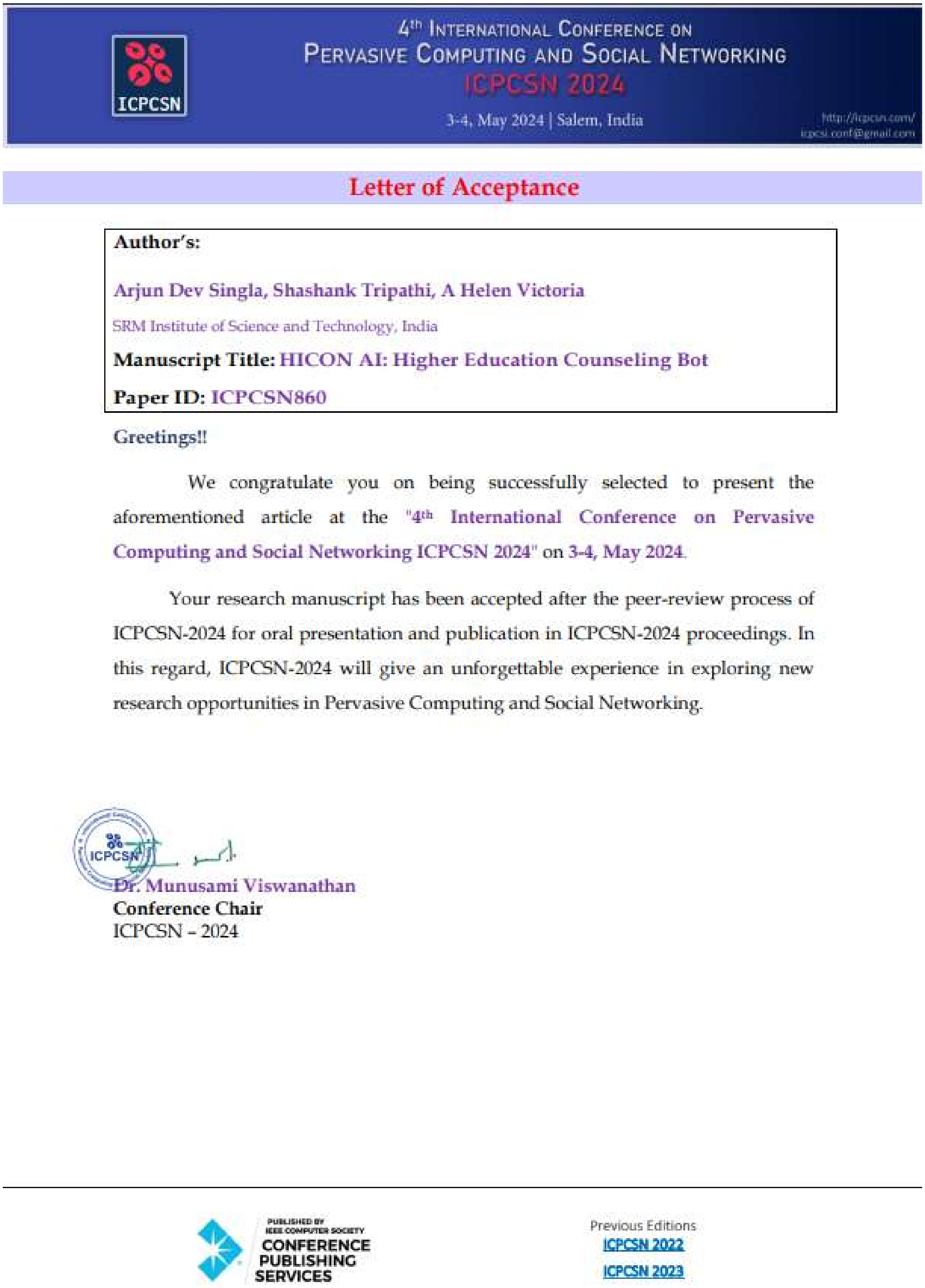
I / We declare that the above information has been verified and found true to the best of my / our knowledge.

Signature of the Candidate	**Name & Signature of the Staff (Who uses the plagiarism check software)**
Name & Signature of the Supervisor/ Guide	**Name & Signature of the Co-Supervisor/Co-Guide**

Name & Signature of the HOD

APPENDIX C

PAPER PUBLICATION



4th INTERNATIONAL CONFERENCE ON
PERVASIVE COMPUTING AND SOCIAL NETWORKING
ICPCSN 2024

3-4, May 2024 | Salem, India

http://icpcsn.com/
icpcsi.conf@gmail.com

Letter of Acceptance

Author's:

Arjun Dev Singla, Shashank Tripathi, A Helen Victoria

SRM Institute of Science and Technology, India

Manuscript Title: HICON AI: Higher Education Counseling Bot

Paper ID: ICPCSN860

Greetings!!

We congratulate you on being successfully selected to present the aforementioned article at the **"4th International Conference on Pervasive Computing and Social Networking ICPCSN 2024"** on **3-4, May 2024**.

Your research manuscript has been accepted after the peer-review process of ICPCSN-2024 for oral presentation and publication in ICPCSN-2024 proceedings. In this regard, ICPCSN-2024 will give an unforgettable experience in exploring new research opportunities in Pervasive Computing and Social Networking.

ICPCSN

Dr. Munusami Viswanathan
Conference Chair
ICPCSN – 2024

PUBLISHED BY
IEEE COMPUTER SOCIETY
CONFERENCE
PUBLISHING
SERVICES

Previous Editions
ICPCSN 2022
ICPCSN 2023