IIT ORSP Assist: Enhancing Faculty Support with an Intelligent Chatbot

Sahil Thorat*
CSP 572 Data Science Practicum
sthorat1@hawk.iit.edu

Albert Mandizha
CSP 572 Data Science Practicum
amandizha@hawk.iit.edu

Anette Volkova
CSP 572 Data Science Practicum
avolkova@hawk.iit.edu

Avinash Sankar
CSP 572 Data Science Practicum
asankar5@hawk.iit.edu

Chandana Poloju
CSP 572 Data Science Practicum
cpoloju@hawk.iit.edu

Priyadarshini Rajendran
CSP 572 Data Science Practicum
prajendran3@hawk.iit.edu

Hemanth K Kamma
CSP 572 Data Science Practicum
hkamma@hawk.iit.edu

Ranjan Mishra
CSP 572 Data Science Practicum
rmishra11@hawk.iit.edu

Vivian Jacob Varghese
CSP 572 Data Science Practicum
vvarghese@hawk.iit.edu

Dr. Yong Zheng
Director of the Center for Decision Making and Optimization (DMO)
Director of the PhD Program in the ITM Department
College of Computing, Illinois Institute of Technology
yzheng@iit.edu

April 26, 2024

ABSTRACT

This report documents the development process and outcomes of a sophisticated chatbot tailored specifically for the Illinois Institute of Technology's Office of Research and Sponsored Programs (ORSP). The chatbot addresses the need for a streamlined communication channel to handle faculty inquiries regarding grant applications, procedural guidelines, and other pertinent information crucial to their research endeavors. Leveraging cutting-edge technologies and innovative functionalities, the chatbot enhances accessibility to information, improves operational efficiency, and contributes to the advancement of research initiatives at the Illinois Institute of Technology. The report outlines the problem statement, research objectives, methodology, system architecture, development phases, evaluation methods, and recommendations for future enhancements. Additionally, it acknowledges the contributions of project advisors, stakeholders, and the institution, underscoring the collaborative effort behind the successful completion of the project

Contents

List of Figures

1 Introduction:

In this final report, we present the culmination of our efforts in developing a sophisticated chatbot tailored specifically for the Illinois Institute of Technology's Office of Research and Sponsored Programs (ORSP). The creation of this chatbot stems from the recognition of the ORSP's need for a streamlined communication channel to address faculty inquiries regarding grant applications, procedural guidelines, and other pertinent information crucial to their research endeavors. Throughout the development process, we aimed to leverage cutting-edge technologies and innovative functionalities to ensure the chatbot's utility and user-friendliness.

This report serves as a documentation of our journey from conceptualization to execution, detailing the methodologies, technologies, challenges, and successes encountered along the way. We believe that our chatbot represents a significant milestone in the evolution of communication and information management within the ORSP, and we are confident that its deployment will contribute to the enhancement of operational efficiency and the advancement of research initiatives at the Illinois Institute of Technology.

1.1 Problem Statement

Faculty members at the Illinois Institute of Technology (IIT) frequently seek information from the IIT Sponsor Office regarding various aspects of grant applications, grant documents, and procedures surrounding grant approvals. While IIT provides Quality Assurance (QA) webpages, these resources may not always address the diverse range of questions faculty members have. To address this issue, there is a need to develop a Chatbot leveraging large language models (LLMs) to efficiently provide answers to faculty inquiries.

1.2 Research Objectives

The Primary Objectives of this project are as follows:

1. Develop a user-friendly Chatbot interface enabling faculty members to pose questions and promptly receive accurate responses, thereby enhancing accessibility to information from the IIT Sponsor Office.

2. Implement a comprehensive QA response system within the Chatbot, including both single-answer responses and responses requiring additional user input to navigate a flowchart, to cater to the diverse range of faculty queries effectively.

3. Incorporate default response mechanisms for questions beyond the Chatbot's scope, directing users to relevant resources or facilitating communication with the IIT Sponsor Office for personalized assistance, thereby optimizing user support.

4. Establish a robust logging mechanism within the Chatbot to capture unanswered questions and forward them to the IIT Sponsor Office for resolution and integration into the knowledge database, ensuring continuous improvement of the Chatbot's capabilities and the availability of comprehensive resources for faculty members.

2 Overall System Architecture

2.1 Client/Server (CS) System

- We used Flask for backend development and adopted MySQL for database storage
- Frontend using CSS and HTML and JScript.
- User authentication via Google API.
- File uploads for managers and Admins.

- QA entry management with multi-turn capabilities.

2.2 ChatBot Design

- Opted for HTML, CSS, and JavaScript for frontend development.
- Adopted Python Flask and its extensions for the backend development of the Chatbot.

2.2.1 Information Retrieval (IR) for QA

- Streamlined IR Process: Transforming queries and database Q&As into vectors for efficient similarity analysis.
- Utilization of Pre-Trained Models: Employing embedding models for effective text vectorization.
- UThese foundational strategies are meant to guide the initial phase of the project, encouraging exploration of alternative solutions and readiness for unforeseen challenges
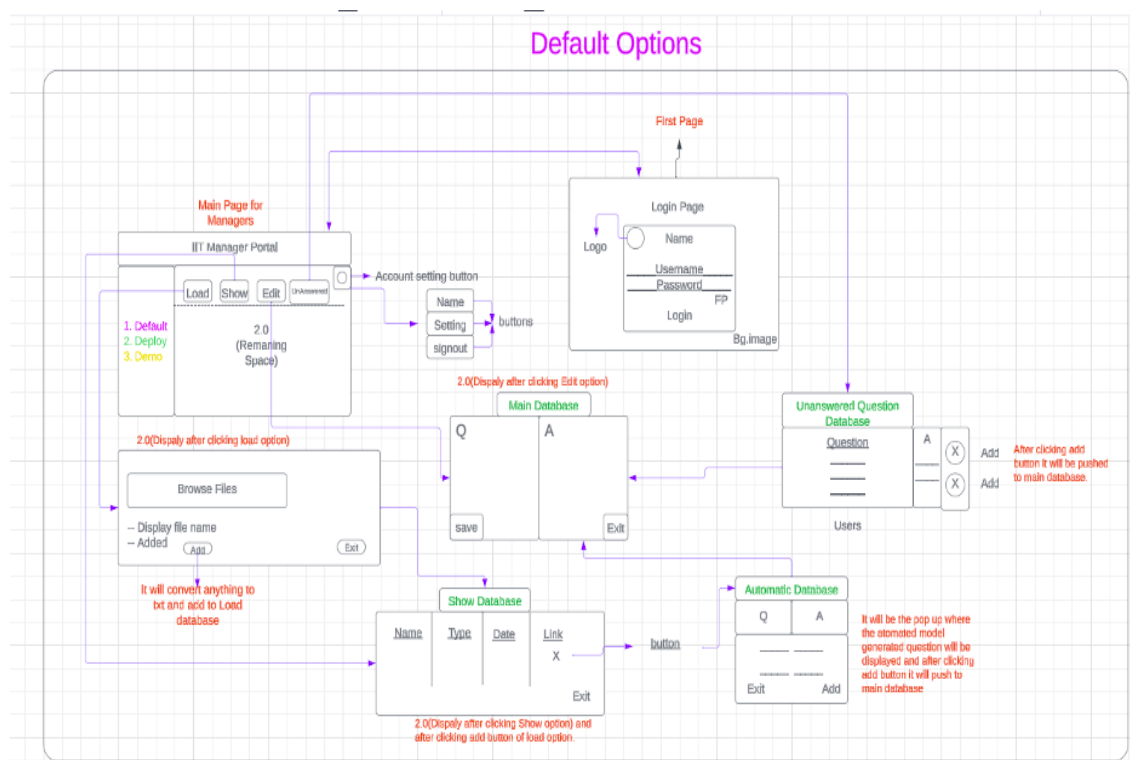
Figure 1: Initial System Architecture.

3 Methodology

3.1 Client/Server (CS) System Design

The client/server system design underwent a phased evolution, progressing through two distinct stages. Initially, the project entered the proof-of-concept phase, during which the team focused on establishing the fundamental workings of the system and testing various hypotheses. Following successful testing and a comprehensive understanding of the system's requirements, the project transitioned into the second design phase.

In this phase, the team revamped the manager and admin console system, refining its appearance and functionality to better align with the project's objectives. Finally, the project advanced to its third phase, which involved the integration

of the entire system with the Information Retrieval (IR) team. This integration marked the culmination of the design process, bringing together the various components into a cohesive and functional client/server system tailored to meet the needs of the client, ORSP.

3.1.1 Implemented Features First Phase

Throughout the initial stages of development, several key enhancements and functionalities were implemented to lay the foundation for the Chatbot project:

1. CRUD Functionality: The addition of Create, Read, Update, and Delete (CRUD) operations for Quality Assurance (QA) entries enabled comprehensive management of QA data within the system. This functionality provided administrators with the ability to efficiently manipulate QA entries as needed, ensuring data accuracy and integrity.

2. Visual Enhancements: Significant improvements were made to the visual presentation of QA management and display pages. These enhancements not only enhanced the aesthetic appeal of the user interface but also contributed to improved user experience and usability.

3. Database Migration: A transition was made from PostgreSQL to MySQL for database management purposes. This migration was undertaken to leverage the benefits offered by MySQL, including enhanced performance, scalability, and ease of management.

The figures below depicts the initial development stages of the Chatbot project, illustrating the utilization of HTML, CSS, and Flask to lay the groundwork for subsequent functionalities and features. These early stages marked the inception of the project, setting the course for further development and refinement as the project progressed.

Figure 2: Manager Console
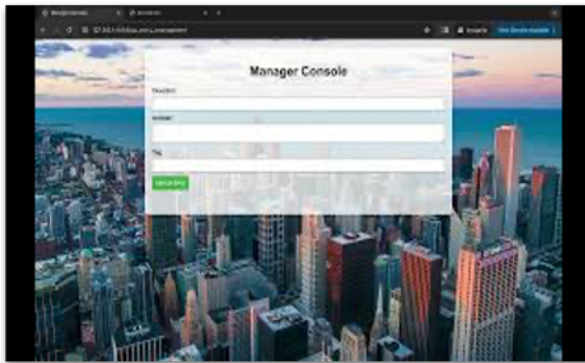

Figure 3: Manager Console

3.1.2 Implemented Features Second Phase

The second phase of our chatbot focused on enhancing the CRUD and developing a new look of our registration interface for better user experience. The team also went on to implement CRUD operations for QA entry and management. The dashboard integrated with a login Page. The figure below shows the enhanced features.
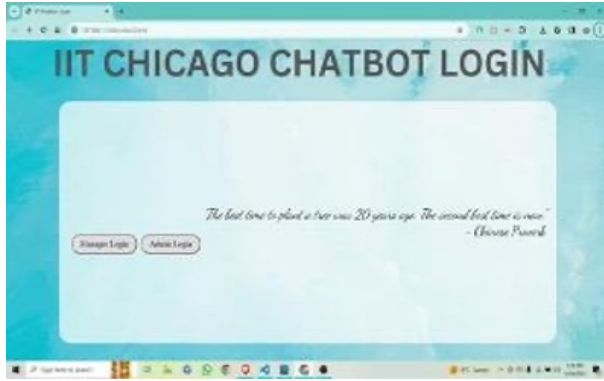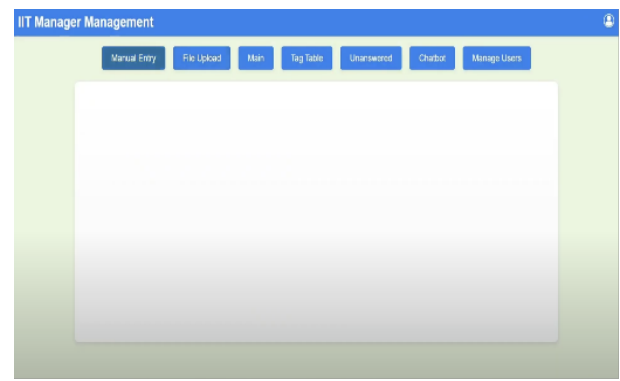
Figure 4: Login Dashboard

Figure 5: Admin Dashboard

3.1.3 Manager Console Dashboard Enhancements

We made significant improvements to our system across various aspects:

- **Manual Entry:**

  - The team was able to add new questions, answers, tags, and options directly within the dashboard interface.

  - This functionality ensured instant updates in both the database and the dashboard table, enabling seamless management of content.

- **Editing Entries:**

  - Users could modify existing questions, answers, tags, and options using the "Edit" button within the dashboard.

  - Changes made were reflected in real-time updates within both the database and the dashboard interface, ensuring accuracy and efficiency in content management.

- **Deleting Entries:**

  - We introduced a feature to remove unwanted entries swiftly by utilizing the "Delete" button within the dashboard.

  - This resulted in instant removal of the selected entries from both the database and the dashboard, maintaining data integrity and decluttering the interface.

- **Dual Dashboards:**

  - We designed and implemented two distinct dashboards—one for administrators and another for managers.

  - This segregation ensured that each user group had access to tailored functionalities and information, optimizing user experience and enhancing administrative control.

Figure 6: Login Dashboard

Figure 7: Admin Dashboard

These enhancements collectively aimed to streamline content management, improve user efficiency, and provide a more intuitive and customizable experience for both administrators and managers alike.

3.2   Chatbot Development:

The chatbot Bot development also went through the two-phase development circle with the initial stages being the User interface design and the team also determined the best development tools that would integrate seamlessly with our Python Flask backend.

3.2.1   User Interface Design First Stage

In the initial stage of our chatbot development, we focused on designing a user-friendly interface using HTML, CSS, and JavaScript. This milestone represents a significant accomplishment in laying the foundation for our project. Despite facing various challenges, our team successfully overcame them and crafted an intuitive UI that greatly enhances the overall user experience. The collaborative efforts of our team members were crucial in achieving this milestone, and we were excited to build upon this solid foundation as we progressed further.

The following figure illustrates the design approach taken during the first stage of our chatbot development. This initial groundwork has provided a strong basis for subsequent phases of development, enabling us to advance towards our project objectives with confidence and purpose.

Figure 8: Chatbot User Interface.

3.2.2   Chatbot Development Continued'

In the second phase the team seamlessly integrated the chatbot interface with MySQL Database and implemented example queries with corresponding responses stored in the database. The team also ensured the functionality for handling automatic reply feature and the unknown question answers were send to. The unanswered table in the Database. The team also ensured that the chatbot is displaying the top three questions using tags. The integration of the all mini model was also done.

3.3   Information Retrieval (IR):

The information retrieval had two distinct stages with the first stage being the one to ensure the format of the data and model selection. This first stage was characterized by tagging of the questions and answers and understanding how the questions and answers will be stored and retrieved from the database by the chatbot.

3.3.1   Data formatting and Database alignment

In the first stage of the Information Retrieval (IR) team's work this week, we focused on solidifying our workflow and laying the groundwork for subsequent tasks. Key achievements include finalizing the format for Quality Assurance (QA) documents to ensure consistency and ease of processing. Additionally, we completed the development of code responsible for converting documents to dataframes, which will be stored in the database and used for training the embedding model.

Furthermore, we made significant progress in defining the initial set of tags to be implemented in our system. These tags, including grants, scholarships, proposal, research, policies, fees, other, and unknown, will serve as crucial identifiers for organizing and categorizing information within the chatbot.

To facilitate model comparison and evaluation, we created a set of similar questions to be used as benchmarks across different models. This will enable us to assess the performance and effectiveness of various approaches as we continue to refine and optimize our chatbot system.

The following figure illustrates the finalized format for Quality Assurance (QA) documents, providing a visual representation of the structure and organization that will be utilized in our system:
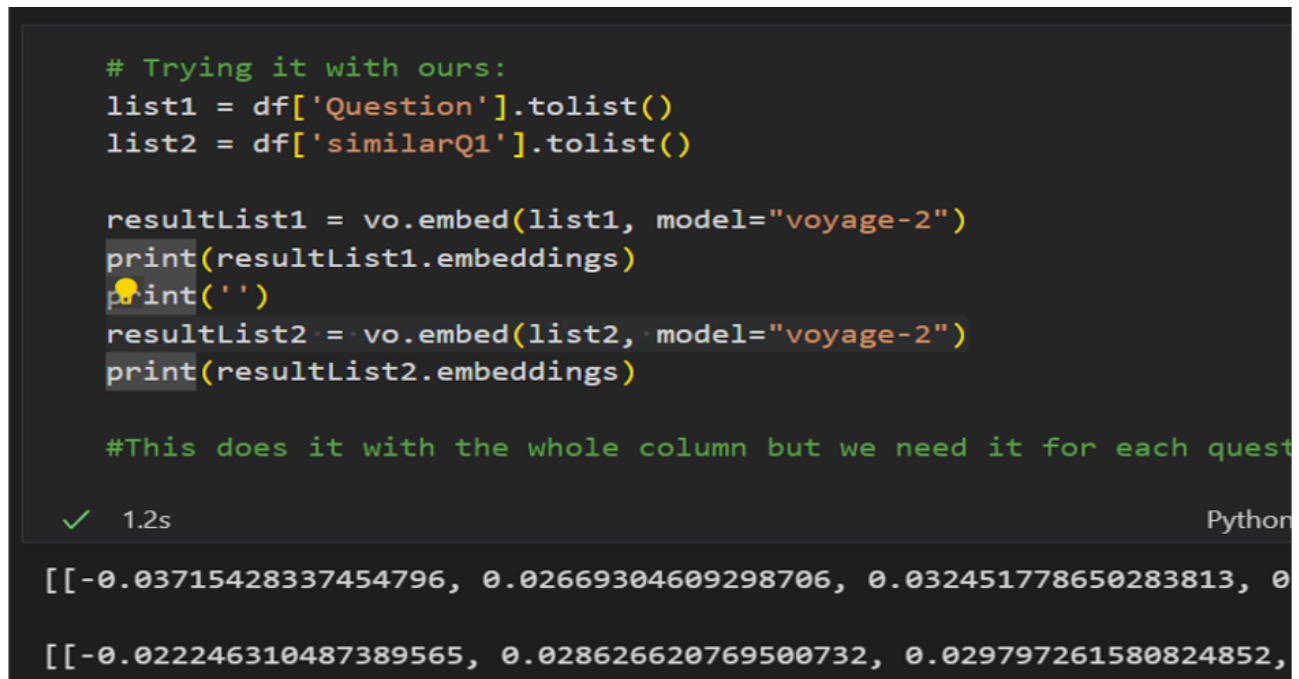
	Question	Answer	Tag
0	I would like to submit a proposal, how do I ge...	Please start a routing sheet (SP proposal) in ...	Proposal
1	How soon should I start my Cayuse routing sheet?	Please start the first page, the General tab, ...	Proposal
2	What are IIT's internal deadlines?	The administrative portions of the proposal (b...	Policies
3	How will I know who my assigned research admin...	After you start your routing sheet, OSRP will ...	Research
4	OSRP only handles federally sponsored projects...	No. OSRP handles all sponsored projects for th...	Other

Figure 9: Question Answer Tag Format.

3.3.2 Model comparison First Phase

At this stage we also started the comparison of the models which we were going to use in Information Retrieval process. Started comparing two models: 1 on the leaderboard - SRF-embedding-mistral and 8 on the leaderboard - BGE-large-en-v1.5

We found out that 1 on the leaderboard takes payment after a while so we moved on to 2 voyage-lite-02-instruct. The figure below shows an image of voyage-lite, we added the original questions and the first set of similar ones and are currently working on the code that compares the vectors

```python
# Trying it with ours:
list1 = df['Question'].tolist()
list2 = df['similarQ1'].tolist()

resultList1 = vo.embed(list1, model="voyage-2")
print(resultList1.embeddings)
print('')
resultList2 = vo.embed(list2, model="voyage-2")
print(resultList2.embeddings)

#This does it with the whole column but we need it for each quest
```

```
✓ 1.2s                                                                Python

[[-0.03715428337454796, 0.02669304609298706, 0.032451778650283813, 0

[[-0.022246310487389565, 0.028626620769500732, 0.029797261580824852,
```

Figure 10: Model Comparison First Phase.

3.3.3 Model comparison Second Phase

The second phase of the IR team involved iterating on all the top seven models to establish the one suitable for the chatbot. The team looked at their Elapsed Time, GPU usage, CPU Usage, Memory Usage, Average accuracy based on how well the model aligns the similar questions to the ones in the Database. The team also worked on figuring out which cosine similarity to use between vectors.

3.3.4 The format for model Comparison

Format for comparing the models:

1. Loaded in data frame Questions, Answer, Tag

2. Created a list of users' questions:
   - Six questions - allowed for iteration of multiple questions to see if model was matching correctly and we made questions of different difficulty to match.

3. Started time.

4. Loaded in model.

5. Embedding Questions from data frame, embedded user question, found cosine similarity give output, stop time.

6. Did this for 7 models: Bge-zh-v1.5, bge-m3-flag, pruneBERT, Voyage-2, GTE-base, All mini-M L6, distill-BERT

7. Recommended: All mini-M L6

The team looked at the model BGE-Large-en-v1.5 [?] and voyagelite: From Hugging Face. The table below gives a detailed comparison of all the models.

Stats	Bge-zh-v1.5	bge-m3-flag	PruneBERT	Voyage-2	GTE-base	All mini-M L6	distilbert
AVG Time elapsed (seconds)	9.3	48.96	1.898	5.73	1.99	0.18	1.72
CPU (final - initial)	2.90%	3.60%	-4.10%	9.9		11.4	67.8
Memory (total usage)	-2147.68 MB	1436.16 MB	165.57 MB	7.68	392.33	58.06	256.06
Size	326M params	568M params	109M params	4000 tokens	102M params	22M params	66M params
Q1.1 cosine similarity	99.99%	100%	99.99%	100	100	100	66.2
Q1.2 cosine similarity	71.14%	73.27%	71.14%	95.37	90.96	82.93	61.08
Q1.3 cosine similarity	62.05%	43%	62.05%	83.21	84.9	42.93	53.42
Q2.1 cosine similarity	75.78%	83.13%	75.78%	93.85	100	88.48	54.91
Q2.2 cosine similarity	69.62%	58.79%	69.62%%	85.8	94.56	71.17	51.18
Q2.3 cosine similarity	60.38%	44.07%	60.38%	84.29	89.6	50.04	50.99
Q3.1 cosine similarity	56.67%	62.05%	56.67%	85.92	100	25.96	39.21
Q3.2 cosine similarity	50.22%	27.15%	50.22%	79.22	93.77	11.23	38.58
Q3.3 cosine similarity	44.67%	23.30%	44.67%	78.61	93.6	10.48	37.4
Q4.1 cosine similarity	76.99%	89.91%	76.99%	97.83	100	92.09	60.97
Q4.2 cosine similarity	60.18%	65.75%	60.18%	94.33	92.25	77.42	56.02
Q4.3 cosine similarity	55.10%	42.56%	55.10%	85.43	88.98	46.74	50.99
Q5.1 cosine similarity	64.40%	50.58%	64.40%	94.45	100	72	42.11
Q5.2 cosine similarity	51.19%	20.96%	51.19%	85.71	88.12	40.53	39.48
Q5.3 cosine similarity	46.10%	16.94%	46.10%	85.52	86.66	36.43	37.12
Q6.1 cosine similarity	77.38%	77.85%	77.38%	95.84	100	90.58	50.42
Q6.2 cosine similarity	50.17%	23.78%	50.17%	79.55	81.03	33.82	50.29
Q6.3 cosine similarity	48.70%	21.40%	48.69%	76.97	80.1	26.89	47.91
correct #1 match	✓	✓	✓	✓	✓	✓	✓
correct #2 match	✓	✓	✓	✓	✓	✓	✓
correct #3 match	✓	✓	✓	✓	✓	✓	✓
correct #4 match	✓	✓	✓	✓	✓	✓	✓
correct #5 match	✓	✓	✓	✓	✓	✓	✓
correct #6 match	✓	✓	✓	✓	✓	✓	✓
AVG Q1	77.73%	72%	77.73%	92.86	91.95333333	75.28666667	60.23333333
AVG Q2	68.59%	62.00%	68.08%	87.98	94.72	69.89666667	52.35333333
AVG Q3	50.52%	37.50%	50.52%	81.25	95.79	15.88333333	38.4
AVG Q4	64.09%	66.07%	64.09%	92.53	94.07666667	72.08333333	55.99333333
AVG Q5	53.90%	29.50%	53.90%	88.56	91.59333333	49.65333333	39.57
AVG Q6	58.75%	41.01%	58.75%	84.12	87.04333333	50.43	49.54
Overall AVG	62.26%	51%	62.18%	87.88333333	92.52944444	55.53888889	49.34833333
Overall AVG based on first match or	75.20%	77%	75.20%	94.64833333	100	78.185	52.30333333

Figure 11: Model Performance Comparison

3.3.5 Decision on Model All Mini-M L6

After model training, we managed to establish the key metrics as below:

- **Best Speed:** All Mini-M L6 models demonstrate superior speed performance compared to other models.

- **Best CPU Usage:** All Mini-M L6 models exhibit the most efficient CPU usage among the models evaluated.

- **Best Size:** All Mini-M L6 models boast optimal size, making them highly suitable for deployment in resource-constrained environments.

- **Best Cosine Similarity (Total AVG):** Voyage-2 model achieves the highest cosine similarity on average across all queries, indicating strong performance in overall response accuracy.

- **Best Cosine Similarity (First Match AVG):** Voyage-2 model demonstrates the highest cosine similarity on average for the first matched query, showcasing its effectiveness in providing accurate initial responses.

As a team, we collectively decided to move forward with the All Mini-M L6 model for our chatbot implementation. This decision was grounded on its exceptional performance metrics, including speed, CPU usage, and size. By leveraging this model, we aimed to optimize the efficiency and effectiveness of our chatbot solution, aligning with our project objectives.

However, in our analysis, we observed that despite the increased pool of options, our models continued to exhibit commendable performance, accurately matching criteria. However, it became evident that certain models displayed superior accuracy compared to others. Notably, the Bge-m3-flag model experienced a decline in similarity scores with the incorporation of additional entries, signaling a potential area for further investigation. Our utilization of sentiment analysis reaffirmed its utility in our workflow. Nonetheless, the exercise prompted us to reevaluate our threshold for decision-making, recognizing the importance of setting it appropriately. Throughout this process, it became clear that the "All mini" option remains our preferred choice, highlighting its continued relevance and effectiveness.

4   Analysis and Results

4.1   Introduction

The results and analysis chapter cover the main functionality of the chatbot and seeks to identify how well we managed to answer our research objectives. The assigned team managed to build a chatbot for the IIT ORSP.

4.2   Operations of the Chatbot

4.2.1   Backend Operations

The following will be procedures on how the chatbot login, file upload, and dashboard interactions work. These operations seek to answer Objective 1 of our research.

Research Question 1
Develop a user-friendly Chatbot interface enabling faculty members to pose questions and promptly receive accurate responses, thereby enhancing accessibility to information from the IIT Sponsor Office.

Results
The chatbot project managed to provide solutions to this research objective with the option of the user to login according to their assigned role, and the Admin has the option to add and remove users, making it easier for the admins to manage the users (managers) of the chatbot. The figures below illustrate the operations of the Manager and Admin as they add and manipulate the questions in the database for a smooth operation and enhanced performance of our Chatbot.
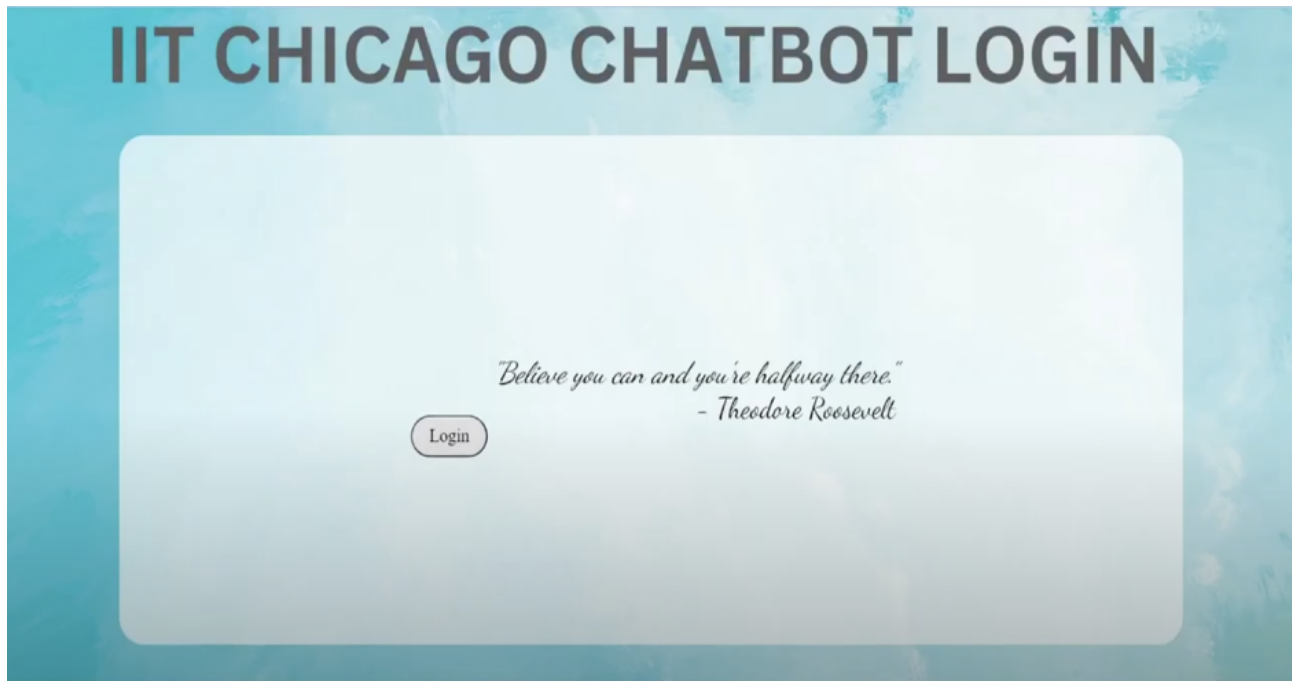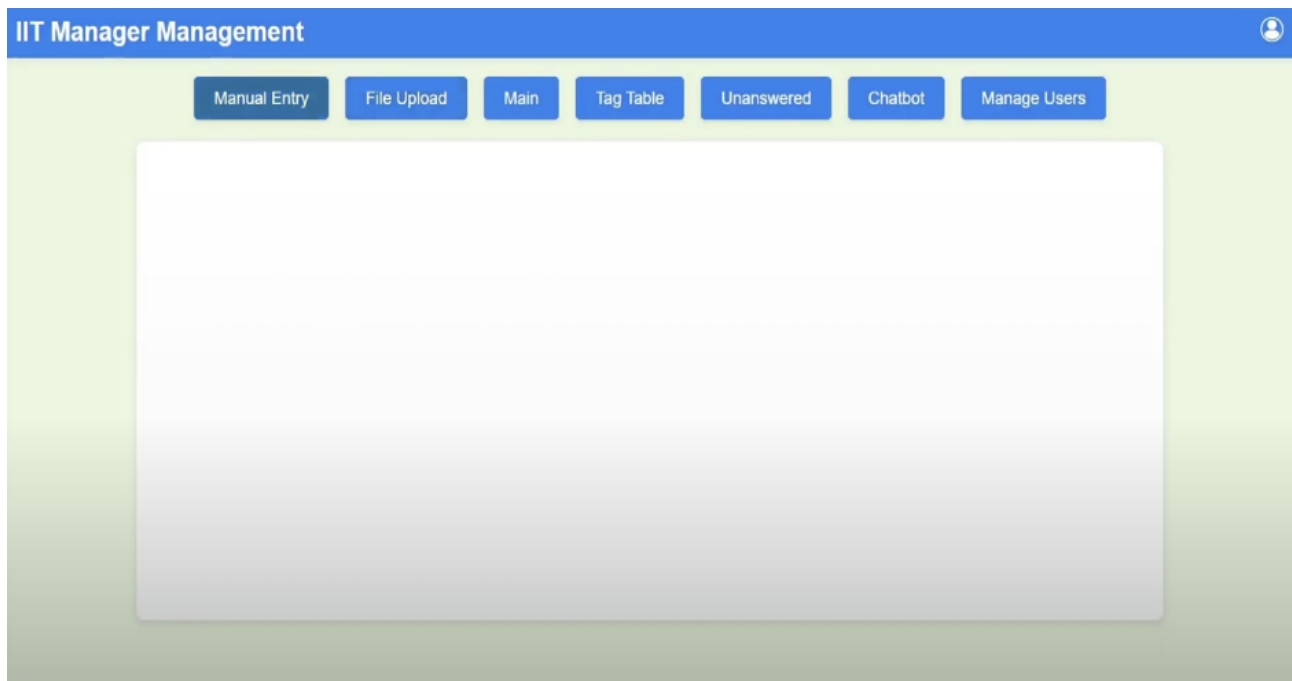
Figure 12: Login Page

Figure 13: Admin Manager Management Console

Figure 14: Admin Manager Manual QA Upload

The Manager and Admin personnel can manually enter single questions into the database and this will also allow them to also provide options as they enter the questions.

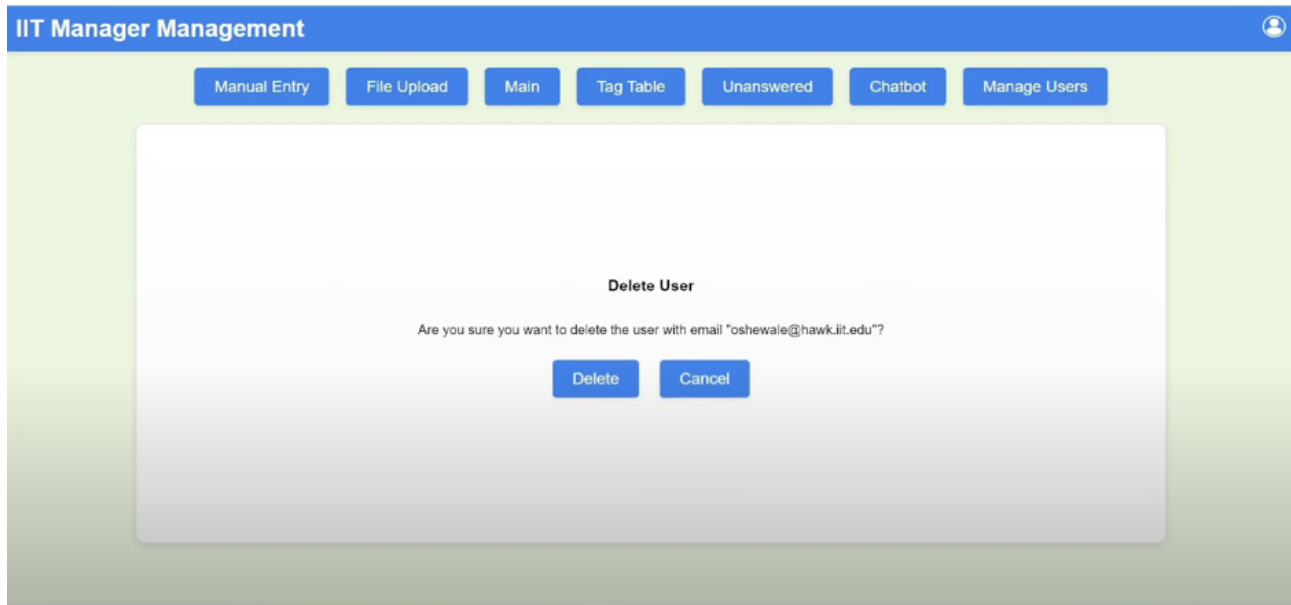Figure 15: Admin Manager Main On QA Details

Figure 16: Admin Manager Delete User

4.2.2 Backend/Frontend Operations

Research Question 2

Implement a comprehensive QA response system within the Chatbot, including both single-answer responses and responses requiring additional user input to navigate a flowchart, to cater to the diverse range of faculty queries effectively thus enhancing multi-response.

Results

Implementation of the Comprehensive QA response within the chatbot was successful, and we managed to accommodate both single answer responses and responses requiring additional user input to allow a flowchart to cater for the diverse queries. The figure below shows how we successfully met our second objective for this research project.

4.2.3 Frontend Operations

Research Question 3

Incorporate default response mechanisms for questions beyond the Chatbot's scope, directing users to relevant resources or facilitating communication with the IIT Sponsor Office for personalized assistance, thereby optimizing user support.

Results

The chatbot can return a similar question from the chatbot if a question is answered that the chatbot cannot answer it will store the unanswered question in the database. On the user frontend side, the chatbot provides some contact information for the users to contact the office for personalized assistance.
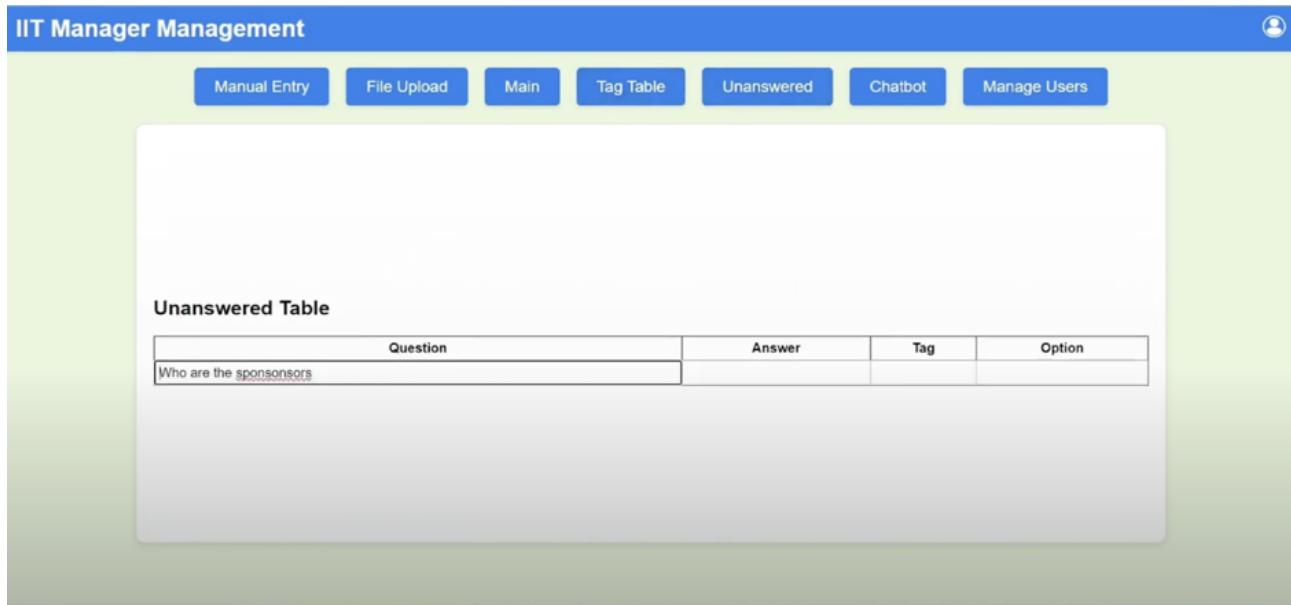
Figure 17: Admin Manager Viewing Unaswered Questions Backend
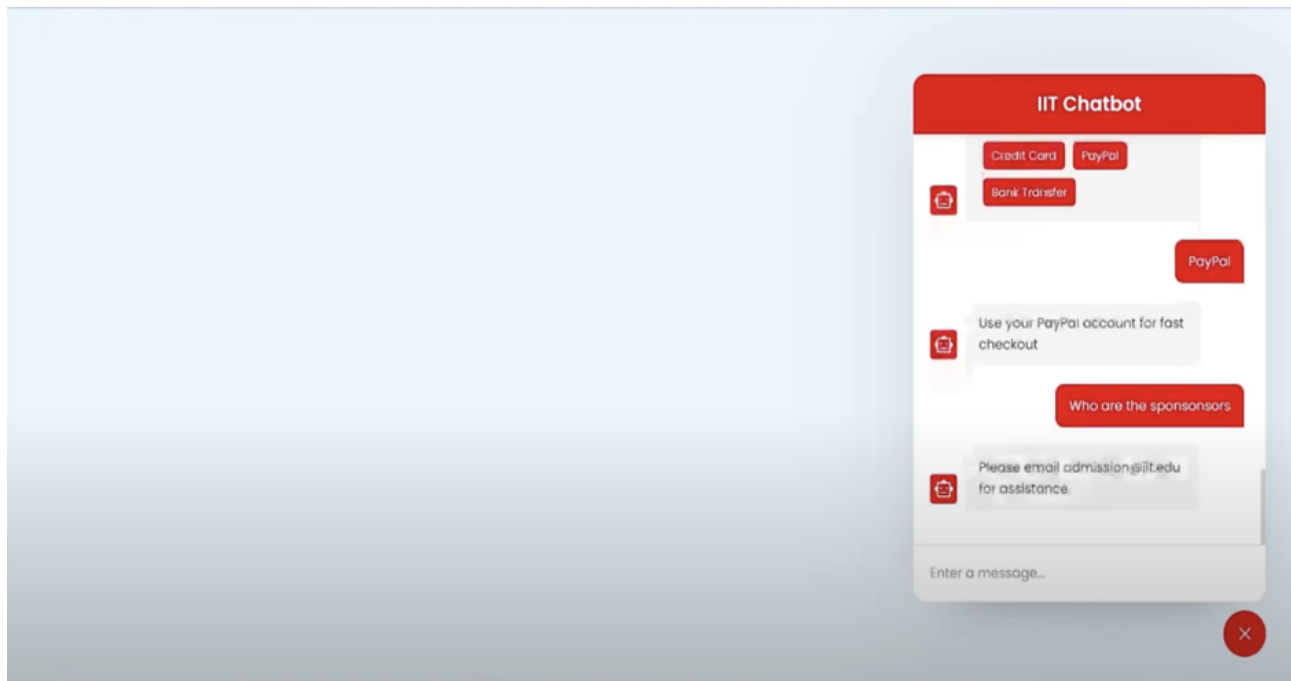

Figure 18: User Interaction with Chatbot Generating unanswered Questions

In this instance a user types a question that has a low similarity score in the Database rather we may say the question that is not present in the database, and it is stored in an unanswered table in our database and can be viewed by the admins and managers at the backend as illustrated below.

The user can click on a tag for instance clicking Bank in this case and the Chabot will pop up the top three frequently asked questions allowing the user to place an option and if the top three are not the ones the user intends to understand about the User has an option to type their question as well.
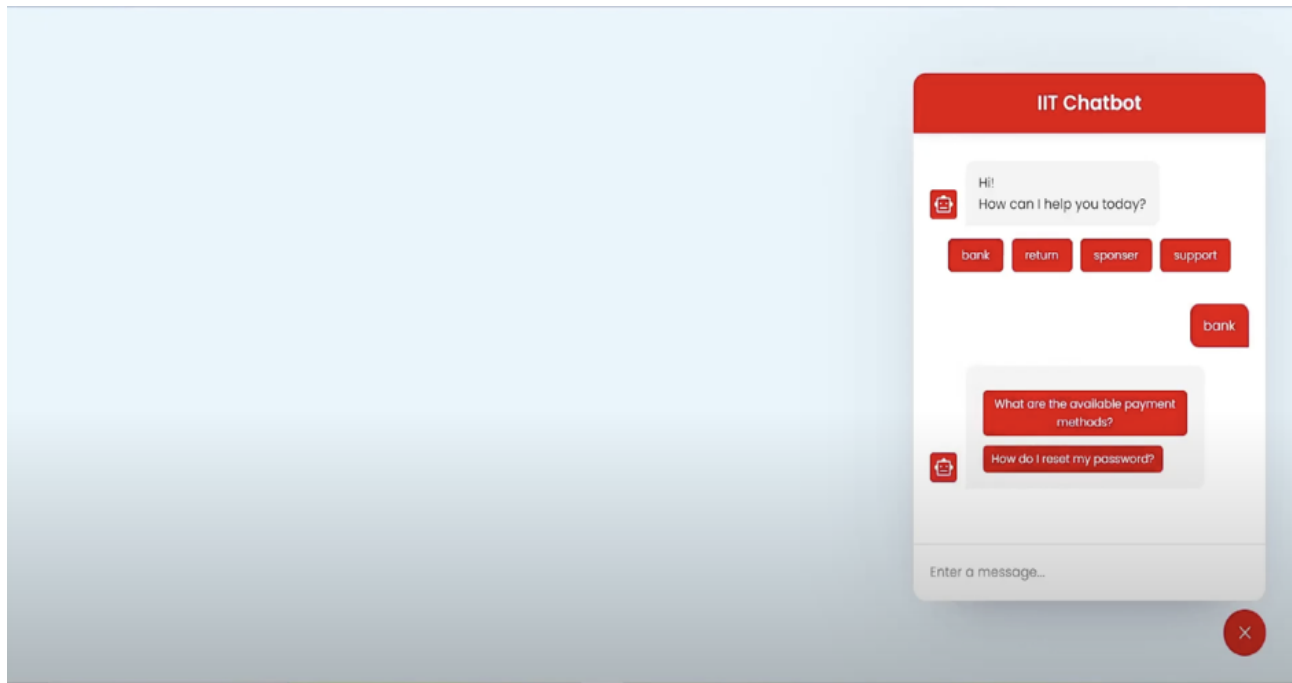

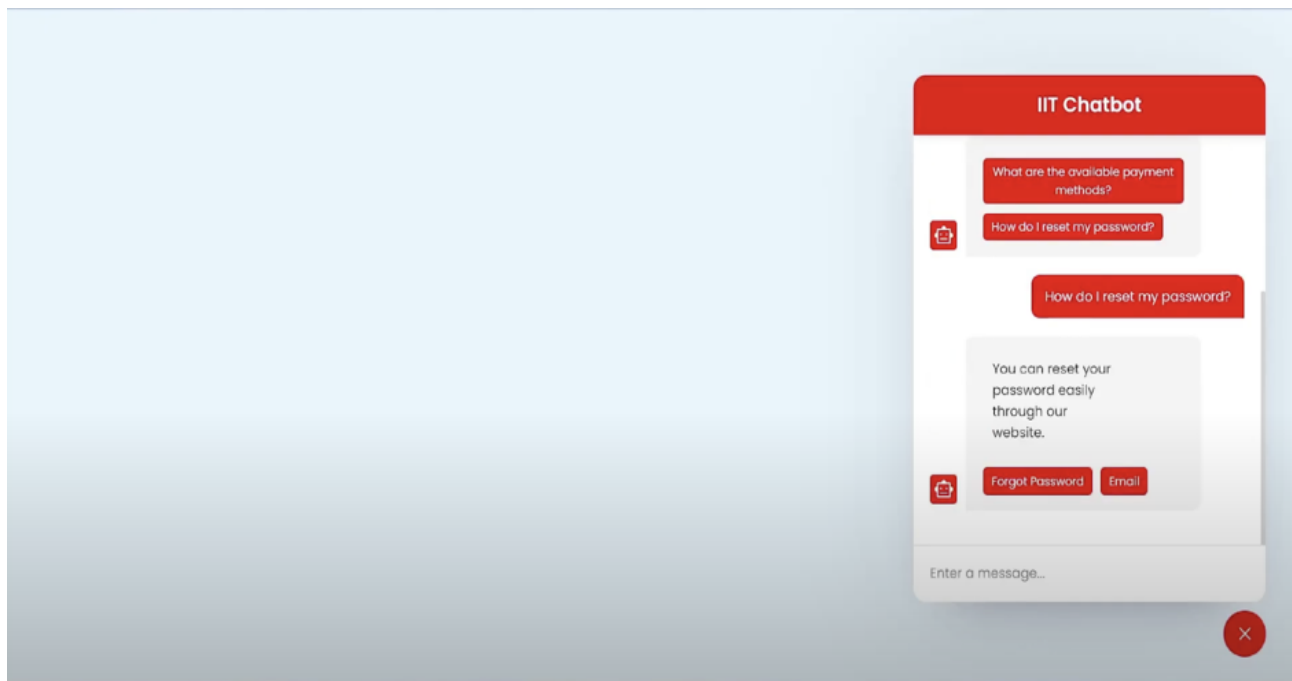Figure 19: Chatbot User Interaction

Figure 20: Chatbot User Interaction

The user can continue interacting with the chatbot to gather all the information they require from IIT ORSP. In cases where the chatbot cannot provide assistance, it provides the user with contact information for faculty members for further assistance.

4.3 Evaluation Methods

We conducted several evaluation methods to help establish the efficiency of the Chatbot and areas for future improvements as discussed below.

4.3.1 Human Evaluation

In the evaluation of the chatbot's performance, a significant majority of human evaluators, constituting 90%, conferred a positive assessment, rating the bot's responses as accurate. This result underscores a commendable level of proficiency in delivering correct answers to users' queries. However, the feedback from evaluators also revealed an area of challenge for the chatbot, particularly in handling nuanced questions.

Despite this difficulty, the chatbot demonstrated a robust capability in addressing straightforward inquiries, consistently delivering accurate responses. These insights from human evaluators highlight both the strengths and areas for improvement, guiding further refinements in the chatbot's functionality to enhance its overall effectiveness and user satisfaction.

4.3.2 User Feedback

User feedback on the chatbot's performance is largely positive, with 85% of users recognizing its helpfulness and accuracy. While this underscores the chatbot's effectiveness in addressing inquiries, users also express a need for more detailed responses. This feedback offers valuable insights into user expectations, guiding improvements for a more satisfying user experience.

4.3.3 Manager and Admin Feedback

The managers were able to upload the CSV and text files of questions and answers with 85% of the managers citing easy use of the dashboard management. The admin was able to add and delete managers as well as manage the questions and answers uploaded by the managers. Also, managers were able to edit questions as well as see unanswered questions in the database. The rating by managers was 82%; however, the sample population was small for us to generalize to the whole population, hence we can only determine the level of satisfaction when we fully deploy the Chatbot.

4.3.4 Unanswered Questions

The low number of unanswered questions in the Database and the low number of emails will be reflective of how the users' questions are being answered by the Chatbot. In a preliminary deployment to 30 students at IIT, the chatbot managed to answer 23 of the questions from the QA already in the database, and three emailed the team for personalized assistance, and the other seven were recorded as unanswered questions in the database. This helped us also add to the FAQs in the database.

4.4 Conclusion

This chapter presents the results and analysis of the chatbot's performance, along with its operations. Evaluation feedback from users and managers provides valuable insights. While the chatbot excelled in accuracy, it struggled with nuanced questions. Users expressed a need for more detailed responses in some cases. Managers found the dashboard easy to use but suggest further deployment for broader insights. The low number of unanswered questions reflects the

chatbot's effectiveness, emphasizing its potential to improve accessibility to information from the IIT Sponsor Office with continued refinement.

5 Recommendations and Conclusion

5.1 Conclusion and Insights

- The development of a sophisticated chatbot tailored for the Illinois Institute of Technology's Office of Research and Sponsored Programs (ORSP) represents a significant milestone in improving communication and information management.

- The chatbot successfully addresses faculty inquiries regarding grant applications, procedural guidelines, and other pertinent information, enhancing operational efficiency within the ORSP.

- The project's methodology, leveraging cutting-edge technologies and innovative functionalities, ensures the chatbot's utility and user-friendliness.

- Human evaluation and user feedback highlight the chatbot's proficiency in delivering accurate responses, although there's room for improvement in handling nuanced questions and providing detailed answers.

- Managers find the dashboard management system easy to use, indicating a positive user experience, but further deployment and feedback collection are necessary for broader insights.

- The low number of unanswered questions reflects the chatbot's effectiveness, demonstrating high question resolution rates during preliminary deployment.

- Overall, the chatbot has the potential to significantly enhance accessibility to information from the IIT Sponsor Office, suggesting continued refinement for optimal user support and satisfaction.

5.2 Future Work and Opportunities

- Continuously refine the chatbot's natural language processing capabilities to improve its ability to handle nuanced questions and provide more detailed responses.

- Expand the chatbot's knowledge base by regularly updating the database with new questions and answers, incorporating feedback from users and managers to ensure relevance and accuracy.

- Explore opportunities to integrate additional features, such as multimedia support for richer user interactions and personalized recommendations based on user preferences and past interactions.

- Conduct further user testing and evaluation to gather comprehensive feedback from a larger user base, allowing for more robust insights into user needs and preferences.

- Collaborate with other departments or institutions to adapt the chatbot for broader applications beyond the ORSP, potentially serving as a model for similar initiatives in academia or other industries.

- Stay informed about advancements in artificial intelligence and chatbot technologies, incorporating new tools and techniques to continually enhance the chatbot's performance and capabilities.

5.3 Conclusion

These recommendations and insights provide a roadmap for future development and improvement of the chatbot, ensuring its continued effectiveness and relevance in meeting the needs of faculty members and enhancing operational efficiency within the Illinois Institute of Technology's Office of Research and Sponsored Programs.

6 Acknowledgment

We would like to express our sincere gratitude to Dr. Yong Zheng for his invaluable guidance and support throughout this project.

7 Bibliography

1. Hugging Face. (2024, February 12). spaces. Retrieved from `https://huggingface.co/spaces/mteb/leaderboard`

2. Chen, D., & Manning, C. D. (2014). A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 740-750). Association for Computational Linguistics.

3. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is All You Need. In *Advances in Neural Information Processing Systems* (pp. 5998-6008).

5. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... others. (2020). Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165*.

6. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. *OpenAI Blog*, 1(8), 9.