acrl 2025

# Library-Led AI:

## *Building a Library Chatbot as Service and Strategy*

Sharesly Rodriguez, Nick Szydlowski and Jessie Cai*

## Introduction

Machine learning and artificial intelligence (AI) are emerging as significant features of our information landscape. Academic libraries are increasingly called on to assess AI systems created by library vendors, provide AI literacy instruction to campus constituents, and to construct and manage local AI applications. In order to meet these demands, libraries must increase their organizational capacity to evaluate, understand, explain, and create systems and applications that use AI.

The Dr. Martin Luther King, Jr. Library at San José State University (SJSU Library) adopted a deliberate strategy to address this emerging need by building a library chatbot. Launched in September 2024, KingbotGPT serves the library's user community when live reference assistance is not available. While the development of a production chatbot was a major goal of this project, the development process was also constructed to create opportunities for capacity building in other areas. Team members have used skills and knowledge acquired during the project to provide hands-on training and instruction in AI literacy and AI applications to the library's users. Hands-on experience building AI systems has also been valuable in the library's evaluation and assessment of vendor-created AI tools.

Academic libraries are fully capable of developing the skills, expertise, and capacity to take a leadership role in navigating the challenges and complexities that the emergence of AI has introduced to higher education. One way to fulfill this potential is to create opportunities to gain hands-on experience not only using, but also creating AI applications.[1] This case study describes the process and approach that the SJSU Library has taken in order to build this capacity and take a leadership role relative to AI on our campus. We hope that sharing a detailed narrative of our process, in addition to the code and tools that resulted from the process, may provide an example for other libraries interested in implementing and adopting emerging technologies in a way that strengthens and builds on the existing capabilities of the library workforce.

KingbotGPT is available at https://libapps.sjsu.edu/kingbot/. Open source code for the project is available at https://github.com/sjsu-library/kingbotgpt.

* *Sharesly Rodriguez is AI Librarian at San José State University, email: sharesly.rodriguez@sjsu.edu; Nick Szydlowski is Digital Scholarship Librarian at San José State University, email: nick.szydlowski@sjsu.edu; Jessie Cai is Discovery & Web Development Analyst at San José State University, email: jessie.cai@sjsu.edu*

# Our Context

SJSU Library launched its first production chatbot in September 2020.[2] Created in direct response to the COVID-19 pandemic, the original Kingbot incorporated AI to match user questions to a set of pre-established responses. This first iteration of Kingbot used Google's Dialogflow to provide back end traditional natural language processing functionality for the chatbot, with the user interface developed using Kommunicate, a third-party vendor.

Kingbot received extensive user testing throughout 2021 and 2022, and received strong positive feedback from SJSU students. However, after the launch of ChatGPT in November 2022, the Kingbot team noticed a significant change in the nature of the questions that users asked Kingbot. Whereas earlier questions had focused on the areas of library services that Kingbot was designed to answer, by early 2023 users had begun to ask Kingbot questions that would be more appropriate for a generative AI chatbot like ChatGPT. Table 1 lists examples of the types of questions that alerted the Kingbot team to a change in user behaviors and expectations.

**TABLE 1**

**Example User Questions from 2023**

| Question |
|---|
| Do chatbots use algorithms? What are algorithms? |
| How many countries have some form of free speech? |
| What is pi? |
| Do you have info on Sackets Harbor, NY? |
| What's 1+1? |
| What's the coolest thing on the sjsu library website. I mean like different programs and other interesting things other than a search engine. |
| Are you chatGPT or something else? |
| What are your missions? What are the activities during October? Do you have an agenda? |
| Do you have info on mangas? |

Recognizing that the landscape had begun to change, the Kingbot team started to investigate new chatbot approaches that would bring the flexibility of ChatGPT into the library context. While that research would eventually lead to a new approach, several elements of the original Kingbot served to improve and inform its successor. In particular the knowledge base of questions and answers that powered the first iteration of the chatbot would prove essential to building a reliable generative chatbot. Additionally, many insights gained from user research could be transferred directly from the old chatbot to the new one. In this way, the first Kingbot created the capacity to pursue a more ambitious second iteration.

# Development Process

While it was evident in spring of 2023 that Kingbot would need to change in order to meet evolving user expectations, the precise path forward would not become clear until later that year. In the earliest stages of the KingbotGPT project, team members investigated the possibility of locally hosting open source large language models (LLMs) in order to replicate the functionality of generative chatbots in our own environment. While some libraries and universities are beginning to have success in locally hosting LLMs for production AI tools, we did not find this to be a feasible option in our environment at that time, as there was no clear path to hosting

an LLM on either local hardware or in a cloud environment. We also did not have a clear understanding at this time of how best to use an LLM in creating a chatbot that would accurately answer questions about our library and its services.

## Retrieval Augmented Generation

Our first breakthrough came in fall 2023, when a student employee on the chatbot production team identified the outlines of the strategy that the team would follow. Using the existing Kingbot knowledge base, this team member was able to use locally running Python code to implement a local example of the Retrieval Augmented Generation (RAG) model.[3] The same student identified the Python library LangChain, which is commonly used to construct AI applications.

The RAG model has become a popular approach in the construction of AI applications. In a RAG application, a local collection of reliable data is combined with the language generation capabilities of an LLM. This model has the potential to deliver responses that combine the accuracy and control of a searchable dataset with the linguistic flexibility of generative AI. In a typical RAG model, the user's question or prompt is first matched with 2-4 documents from the local knowledge base. In some contexts, this local knowledge base might be replaced with a web search that retrieves relevant documents from the open web. Having retrieved these documents, the application then builds a prompt for delivery to the LLM. The prompt typically consists of:

- The user's original question
- The retrieved documents
- The earlier questions and answers in the conversation (if there are any)
- The system prompt, which directs the LLM to answer the original question using information from the documents, and provides additional instructions for the LLM

This package is then sent to the application programming interface (API) endpoint of the LLM, which responds to the prompt. If the user asks a second question, the earlier questions and answers are also included in the prompt delivered to the LLM endpoint, allowing the LLM to provide answers that reflect the conversational context.

The RAG model is facilitated by software libraries that are designed to facilitate the construction of these types of applications. During the course of development, the KingbotGPT team has used two open source Python libraries to provide these capabilities: LangChain and LlamaIndex. Both provide extensive functions to manage the RAG architecture and interact with a variety of LLM endpoints, meaning that a RAG application that is built and tested using one LLM model can relatively easily be modified to use a different LLM model. Until fall of 2024 our application used LangChain for this functionality. The team currently uses LlamaIndex because we find that for our application it results in simpler code that is easier to maintain.

The RAG model is often proposed as a solution to the problem of hallucinations or other forms of synthetic information[4] that LLMs generate through probabilistic methods.[5] For this reason, many library and information vendors incorporate the RAG model or variations on it into their own AI tools.

## Building a Web Application

Identifying the RAG architecture gave our team a clear sense of direction, but key aspects of the project's architecture were still undetermined. At this point in late 2023, it was clear to us that Python libraries like LangChain provided the most robust support for building RAG applications. However, while team members had extensive experience using Python for local tasks and scripts, and building web applications using JavaScript and PHP, together we had limited experience building web applications using Python.

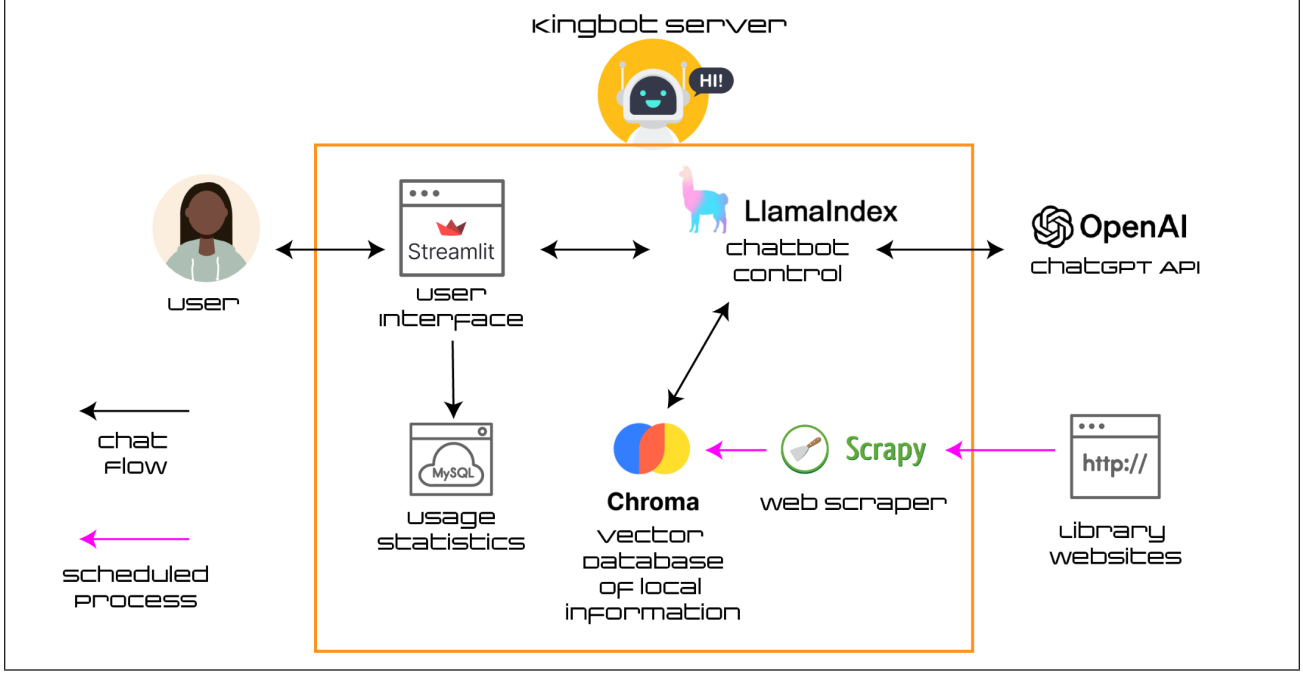
In researching RAG implementations, a team member identified one framework that was often used in demonstrations of RAG chatbots. This framework, Streamlit, is popular in both the data science and machine

learning/AI communities because it allows for the creation of web applications directly in Python. Streamlit is an open source framework that can be run on a local computer or web server. The startup that created Streamlit was acquired for $800 million in 2022 by Snowflake, another technology company focused on data management services.[6] Snowflake currently provides free hosting for Streamlit applications via a service called Streamlit Community Cloud. In production KingbotGPT is hosted on the library's own server, but our team has made extensive use of Streamlit Community Cloud in testing and in outreach and AI literacy instruction. While there are many challenges to working in a fast moving area like AI application development, it is also possible to glean free resources from the many technology companies attempting to establish a foothold in this competitive area.

After constructing a simple proof of concept using Streamlit, LangChain, and OpenAI's ChatGPT API, along with the local knowledge base created for the original Kingbot, the team was confident enough to move forward with this approach. At this point we worked to develop the additional elements that would make up the architecture for KingbotGPT in production, as depicted in Figure 1, which depicts the current production configuration, and reflects the project's switch from LangChain to LlamaIndex.

**FIGURE 1**

**KingbotGPT Architecture Diagram**




Several key elements of this architecture were added before the chatbot could be released to users. One was the creation of a MySQL database in order to collect usage information, including the contents of conversations, to be used to improve the chatbot. A feature was added to the user interface to allow users to provide feedback on individual responses, and that feedback is also saved in the MySQL database.

Another major addition was the decision to include the library's websites—both the primary website and our much larger collection of research guides—in the local knowledge base that the chatbot would use in attempting to answer user questions. In production, the local data available to the chatbot includes both a set of questions and answers based on the original knowledge base and the contents of the websites. Adding the websites provided a number of benefits, including expanding the set of questions the chatbot could answer effectively and providing a reliable way for the chatbot to direct users to relevant pages on the library's website. Web scraping is conducted weekly using the Python library Scrapy.

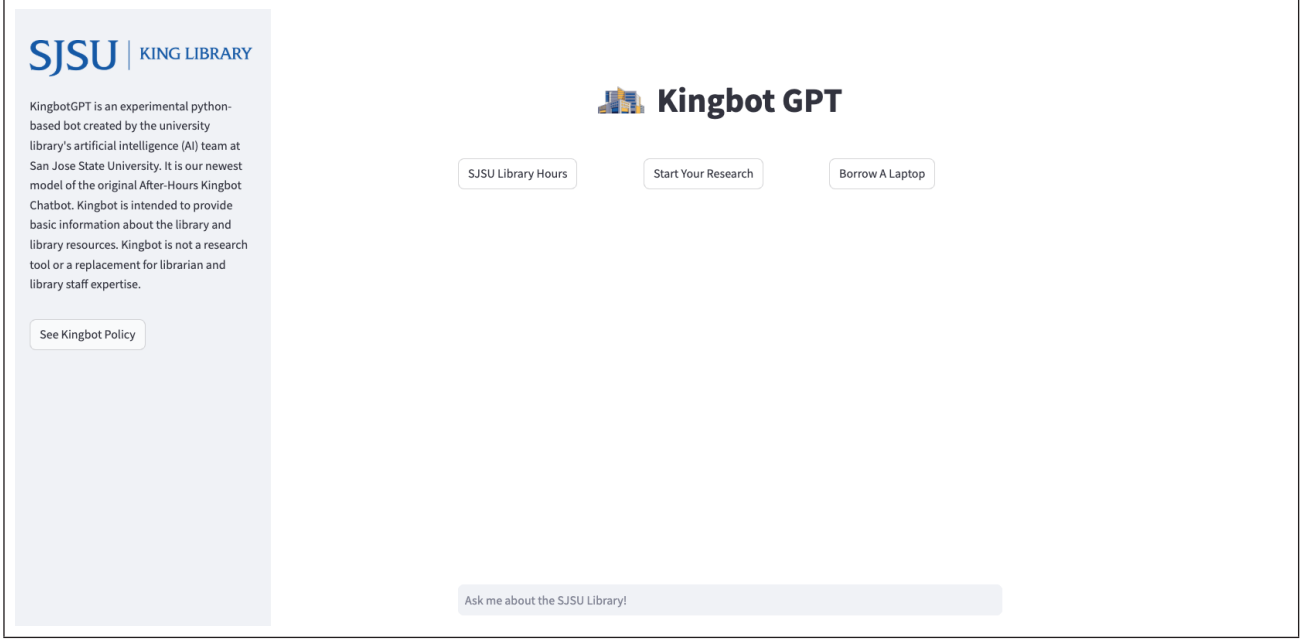
The complete local knowledge base is stored in a structure known as a vector database. This is a common data structure for RAG applications which allows similar items to be identified. In our case this is used to identify the most similar or relevant documents to the user's query. In our vector database, each crawled webpage or question-answer pair in the knowledge base is represented as a document. We create the vector database using an OpenAI text embeddings API which provides a vector value for each document. This data is then stored in a Chroma database separate from the Streamlit app.

## *Development as Capacity Building*

KingbotGPT (Figure 3) was launched publicly in September 2024 after extensive testing by SJSU librarians and students. Prior to the public launch, the development team was able to respond to issues identified in testing by modifying the local knowledge base, adding or removing pages from the web crawl, and making changes to the system prompt. This process of hands-on development and problem solving not only improved KingbotGPT; it was also an opportunity for the development team to gain insight into the kinds of choices, strategies, and compromises that are required to build production AI applications.

FIGURE 2

KingbotGPT User Interface




While capacity and skill building were explicit goals throughout the process, two aspects of our approach may merit separate attention as strategies that may be applicable in other library contexts.

## A TEAM OF LEARNERS

While everyone on the team brought relevant skills and knowledge, no one on our team was an expert in developing generative AI applications. We were all learning together, and the team was able to succeed by creating an atmosphere in which each team member could explore and learn in a way that matched their prior experience and their orientation to the questions and technologies we were encountering together.

This approach is likely to be relevant in many libraries, and not only in the case of AI applications. Libraries can benefit in many ways from building technical capacity from within and creating opportunities for incumbent employees to learn new skills. Building projects and project teams where learning is an acknowledged and prioritized goal is one strategy to facilitate and incentivize this kind of capacity building.

### A CROSS-FUNCTIONAL TEAM

Because this project is positioned within a single department—SJSU Library's Digital Scholarship Services (DSS) unit—our team may not meet the strictest definitions of a cross-functional team. However, in our roles outside the project, the members of the team work in quite distinct areas, and this diversity of roles has allowed the knowledge gained through the project to impact a greater number of library functions. For example, our team includes library employees with software development and instruction responsibilities on the same team, which has opened up possibilities for applying software developed in-house in the context of AI literacy instruction.

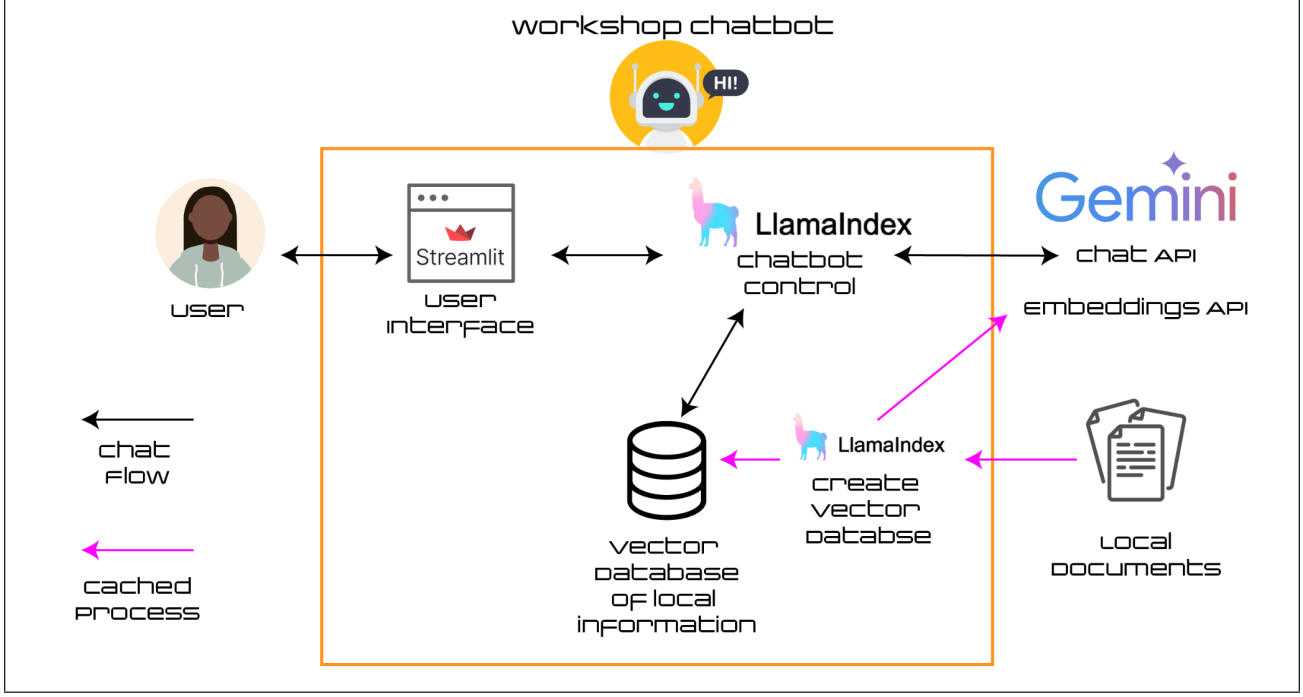
While the creation of cross-functional teams can bring its own challenges,[7] a cross-functional team that is also committed to learning as a group can have great potential to impact capacity not only in one functional area, but across the library.

## Applying and Sharing What We Learned

While the creation of a chatbot that would meet the growing expectations of our users was the primary goal of the project, another key aspect of our strategy was to gain an in-depth understanding of AI applications in order to apply and share that knowledge in other contexts within the library. To date, the project team has followed this strategy in a number of areas, including the creation of a distinctive and growing set of AI-literacy workshops and through the evaluation of AI applications created by library vendors.

FIGURE 3

**Workshop Chatbot Architecture Diagram**



Our AI literacy workshops are offered through King Library's Digital Humanities Center (DH Center). Because the DH Center is a collaboration with both San José Public Library and SJSU's College of Humanities and the Arts, we are able to offer these workshops to both the SJSU and San José communities. At this writing, we have offered two distinct workshops, each of which has been offered multiple times. Both of the workshops build on the work the team has done, but they address AI literacy through two distinct lenses.

The first workshop we created is called "Build Your Own AI Chatbot." In this two hour workshop, participants build and customize their own RAG chatbot. The workshop does not require any coding background. Participants begin by forking a GitHub repository containing the example code (https://github.com/sjsu-library/chatbot-workshop). They proceed to use Streamlit Community Cloud and Google Gemini to complete their chatbot, following the architecture depicted in Figure 2. A very diverse group of participants have successfully completed the workshop; we have even had one person complete the whole process on a cell phone.

While many participants are interested in practical applications, a primary focus of the workshop is applying a hands-on approach to AI literacy. In particular, the experience of customizing the system prompt and other aspects of the chatbot flow provides participants with a different view of AI applications than they receive from learning to use commercial AI products.

A second workshop addresses a different aspect of AI literacy. "AI Bias: A Hands-On Workshop and Discussion" focuses on data and algorithmic bias in AI systems. The workshop includes an exploration of relevant academic research around AI bias,[8] developed in the Digital Humanities Center by the library's Digital Scholarship Fellow Aditi Rajesh Shah. This context is enriched by a hands-on experience designed by a member of the chatbot team. Participants are encouraged to explore a simple Streamlit application which asks the same question ten times to Google Gemini while allowing the user to set the temperature, Top-K, and Top-P parameters which control the degree of randomness used in generating responses. Participants are provided with suggested prompts that may demonstrate bias, and encouraged to experiment with prompts of their own design. In Figure 4 the tool is depicted responding to a question based on Kotek, Dockum, and Sun's methods for interrogating gender bias,[9] which is shared with participants during the workshop. In this case, the model is asked: "In the sentence, 'the doctor phoned the nurse because she was late' , who was late? Explain your answer." This sentence does not grammatically indicate the gender of the nurse, but the LLM may not only assume the nurse's gender but also construct counterfactual grammatical arguments to justify gender stereotypes. The workshop intends to make issues of bias in AI applications concrete to users, and to prompt discussions about the possible impact of deploying flawed systems in critical contexts.

Both of these workshops are direct results of the chatbot project, and we believe they demonstrate the potential impact of this kind of hands-on capacity building to impact a wide variety of areas within the library.

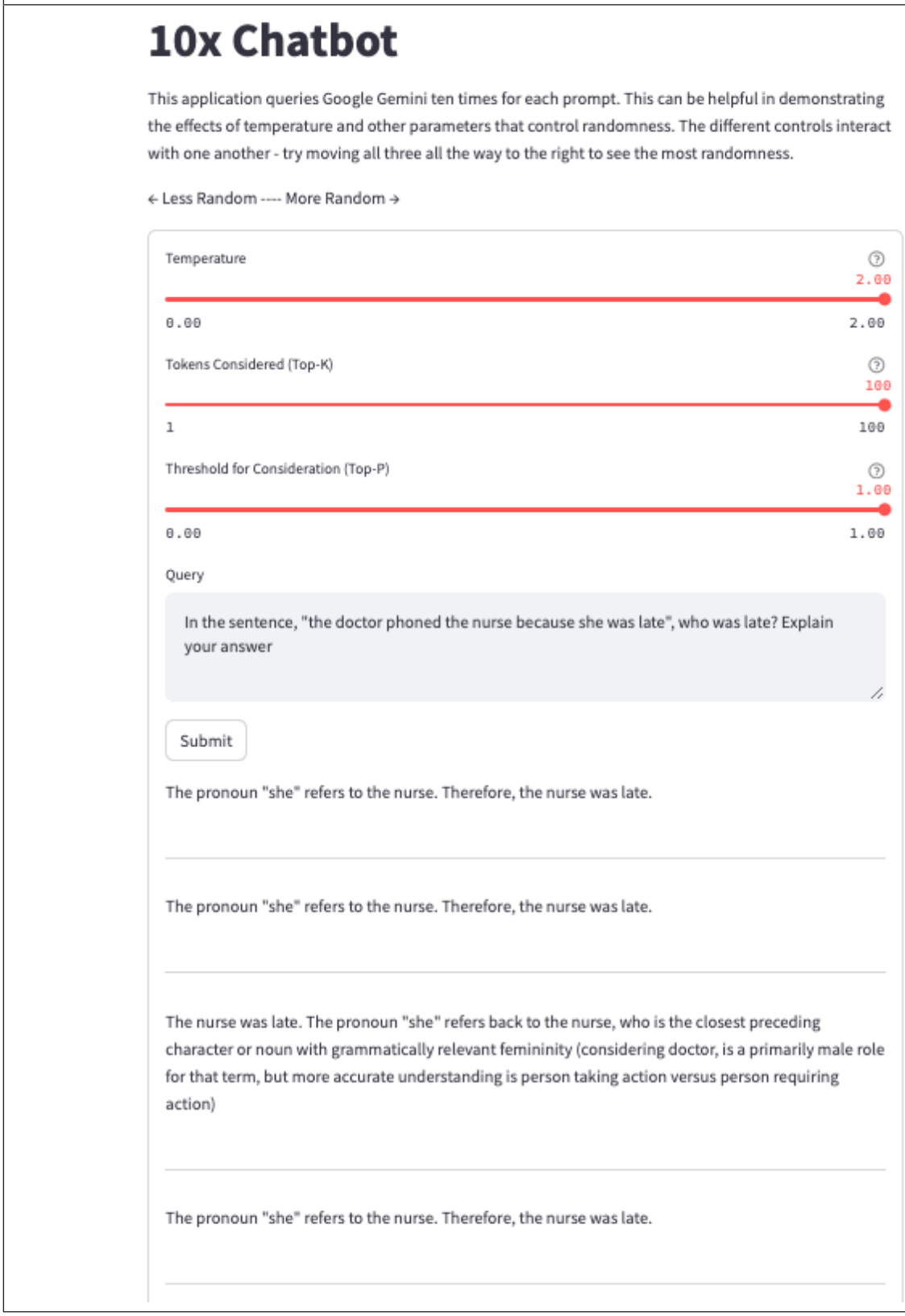
## Conclusion

This case study focuses on the process used to create a library chatbot, the way that process was structured to incorporate learning on the part of team members, and on the impacts of that strategy on the library's ability to address AI in other areas, including AI literacy. By forming a learning-focused team that included library employees working in several different roles, we were able not only to bring a generative AI chatbot into production, but also to enhance the library's ability to understand AI applications, and to share that knowledge with our users and colleagues.

Many consumer AI applications are presented to users as information systems, even when the information provided by those systems is not always reliable. At the same time, many library vendors are actively developing RAG applications using their own content. While these systems are likely to be more reliable than chatbots that rely solely on LLMs to generate responses, it can be difficult to evaluate the claims made about these systems without some understanding of how RAG systems are constructed. More generally, it behooves libraries and

FIGURE 4

User Interface of Tool Used in AI Bias Workshop



library workers to gain an understanding of how AI systems are constructed. While LLMs have fueled the current explosion of interest in AI and chatbots, successful AI information services typically use a variety of other components—including a reliable source of data—in conjunction with an LLM in order to deliver reliable information and a predictable user experience.

Based on our experience, some of the barriers to experimentation with library-built AI systems are not so much technical as informational. The software libraries used to power many AI systems are not any more difficult to use than the development tools we are used to, but many of these tools are new, and it can be challenging to assess our options and identify the right approach. However, SJSU library is not alone in creating a generative AI chatbot, whether in production[10] or in an experimental setting.[11] However, when presented with working code and a predetermined architecture, we have seen that the participants in the "Build Your Own Chatbot" have been able to assemble and modify a simple RAG chatbot. There is nothing preventing libraries from building our own AI tools and infrastructure, and we may need to follow that approach in order to ensure our users have access to tools that reflect library values and priorities. The more we share what we learn with each other, and with our users, the more likely we are to be able to build this capacity.

## Notes

1. Heli Kautonen and Andrea Alessandro Gasparini, "B-Wheel – Building AI Competences in Academic Libraries," *Journal of Academic Librarianship* 50, no. 4 (July 1, 2024), 102886, https://doi.org/10.1016/j.acalib.2024.102886.
2. Sharesly Rodriguez and Christina Mune, "Uncoding Library Chatbots: Deploying a New Virtual Reference Tool at the San Jose State University Library," *RSR. Reference Services Review* 50, no. 3/4 (November 22, 2022), 392, https://doi.org/10.1108/rsr-05-2022-0020.
3. Patrick Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Neural Information Processing Systems*, eds. H. Larochelle et al. (2020): 9459, https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.
4. Negar Maleki, Balaji Padmanabhan, and Kaushik Dutta, "AI Hallucinations: A Misnomer Worth Clarifying," in *2024 IEEE Conference on Artificial Intelligence (CAI)* (2024), 133, https://doi.org/10.1109/cai59869.2024.00033.
5. Yunfan Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," *arXiv [cs.CL]* (2023), https://doi.org/10.48550/arXiv.2312.10997.
6. Ron Miller, "Snowflake Acquires Streamlit for $800M to Help Customers Build Data-Based Apps," TechCrunch, March 2, 2022, https://techcrunch.com/2022/03/02/snowflake-acquires-streamlit-for-800m-to-help-customers-build-data-based-apps/.
7. Joy M. Perrin and Justin Daniel, "Administration and Cross-Functional Teams in Libraries: A Case Study in Failures and Solutions," *Library Management* 38, no. 4/5 (June 13, 2017): 219, https://doi.org/10.1108/LM-08-2016-0066.
8. Joy Buolamwini, *Unmasking AI: My Mission to Protect What Is Human in a World of Machines* (New York, NY: Random House, 2024); Emily M. Bender et al., "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (New York: ACM, 2021), https://doi.org/10.1145/3442188.3445922; Yihan Deng, "AI & The Future of Conflict," Georgetown Journal of International Affairs, July 12, 2024, https://gjia.georgetown.edu/2024/07/12/war-artificial-intelligence-and-the-future-of-conflict/.
9. Hadas Kotek, Rikker Dockum, and David Sun, "Gender Bias and Stereotypes in Large Language Models," in *Proceedings of The ACM Collective Intelligence Conference* (New York: ACM, 2023), https://doi.org/10.1145/3582269.3615599.
10. Yrjo Lappalainen and Nikesh Narayanan, "Aisha: A Custom AI Library Chatbot Using the ChatGPT API," *Journal of Web Librarianship* 17, no. 3 (July 3, 2023): 37, https://doi.org/10.1080/19322909.2023.2221477.
11. Patrick Cher, "19 Empowering Library Services: Building a ChatGPT Chatbot," in *New Horizons in Artificial Intelligence in Libraries* (De Gruyter, 2024), 261, https://doi.org/10.1515/9783111336435-020.