School of Informatics, Computing, and Cyber Systems

NORTHERN ARIZONA UNIVERSITY

Topic

By

Candidate: Venkata Lakshmi Ponugoti

Student Number: 6319241

Candidate: Sai Pavan Kumar Rasamsetty

Student Number: 6323688

Candidate: Avinash Reddy Kolagottu

Student Number: 6321044

Candidate: Anusha Devaraj

Student Number: 6321314

A dissertation submitted in fulfilment of the requirement for the Master of Information Technology

School of Informatics, Computing and Cyber Systems

Department of Information Technology

Supervisor: Prof. Troy Lee Adams

2024

Table of Contents **Page No.**

CHAPTER 1: INTRODUCTION AND PROBLEM STATEMENT

1.1 Problem Selection: Using Machine Learning (ML/AI) in Customer Service

Customer service is very important in industries like online shopping, banking, and technology. However, many people have bad experiences because of slow replies, unhelpful assistance, or needing to repeat their problems to multiple agents. These issues waste time, frustrate customers, and hurt the company's image. For example, shoppers often need quick help comparing products, getting details, or solving issues after buying.

While there are AI tools like OpenAI's systems that help, they often cost too much, are hard to customize, and raise privacy concerns. Cloud-based tools might not work well for companies dealing with sensitive data, and smaller businesses may find them too expensive. This project aims to create an affordable AI chatbot that can work locally to protect data, answer product questions, help with buying decisions, and provide post-purchase support. It will be designed to solve the problems of current AI tools, like high costs and lack of privacy, while being easy for businesses to customize.

1.2 Introduction to ML/AI

Machine learning is a type of artificial intelligence (IBM) that helps computers learn from data and make decisions without needing constant instructions. In customer service, ML is used in tools like chatbots, virtual assistants, and support systems to make them more efficient. For example, ML can take care of simple tasks automatically and give useful suggestions to customer service agents to help them work faster. It can also analyze customer information to predict what they need and offer the right help. Using ML in customer service makes it easier for teams to do their jobs, provide better support, keep customers satisfied.

1.3 Problem Justification:

These days, many people often need to contact customer support for things like booking airline tickets, checking online order status, getting train ticket information from companies like Amtrak, or asking about services like Flex Bus. The big issue is that customers usually face long wait times, especially during busy periods. It's not unusual for people to wait on hold for 30 minutes or even over an hour.

While some companies have started using chatbots to help, these bots are mostly limited to answering simple questions or providing basic information. They usually can't handle more complicated or personalized issues that need an understanding of the customer's specific needs. This leaves customers feeling frustrated and makes it hard for businesses to provide good service efficiently.

Our solution to address the above-mentioned real-world problem is "Incorporating Machine Learning (ML/AI) in Customer Service."

With Machine Learning (ML), we can build smarter systems that do more than just give basic answers. They can learn from past interactions to handle more complicated and personalized questions. This can help cut down wait times, make customers happier, and make the whole customer support process more efficient. (Aggarwal, B., & Subhashis, J. (2024, May 7).

Expected Impact:

Using ML/AI to improve customer service has many benefits. It can automate simple tasks, give quick answers to common questions, and learn from past interactions to handle more complex issues. This makes customer service faster and more accurate, keeping customers happy and more likely to stay loyal and recommend the company, which improves its reputation. It also saves money by reducing the need for human agents and helps keep customers by offering better service, which can increase revenue. With these advantages,

ML/AI (Aggarwal, B., & Subhashis, J. (2024, May 7).is a powerful tool for improving customer experience and helping businesses succeed.

CHAPTER TWO: PROJECT OBJECTIVES AND RESEARCH

2.1 Objectives for Solving the Problem

The main goal of this project is to create and use a local AI chatbot to improve customer service by fixing issues like long wait times, unhelpful responses, and privacy concerns.

We are following the SMART goals method to guide our steps:

Clear Goal: Develop a chatbot that can answer product-related questions, compare items, and assist with setup and troubleshooting by using information from product PDFs. *(Kemper, Jan, 2017)*

Measure Success: Aim for the chatbot to give correct answers 90% of the time for product-related queries and ensure at least 80% of users are satisfied with its speed and performance.

Achievable: Build the chatbot using existing language technology, train it with product manuals and guides, and test it thoroughly to ensure accurate responses.

Relevant: This chatbot will address the main problems by reducing the need for human agents, speeding up response times, and keeping user data secure.

Summary of Research and Evaluation Tool

To create the local AI chatbot, we researched the latest tools in natural language processing (NLP), customer service automation, and product comparison. Here's what we focused on:

NLP Models:

We studied popular language models like OpenAI's GPT and smaller open-source options like LLaMA 3.2, Gemma, Granite, and Mistral. We evaluated how well these models can answer questions about products and compare items. Our focus was on models that can run locally to save costs and protect user data. *(OpenAI, 2024; Meta AI. 2024)*

Local vs. Cloud Solutions:

While cloud tools like OpenAI's API are powerful, they come with concerns about data privacy and can become expensive with heavy use. Local models provide better control over data and can be customized more easily. We compared the hardware requirements and setup needs for these local models to ensure they were suitable for our project. *(OpenAI, 2024)*

2.1.a) Cost Comparison Table: OpenAI API vs. On-Premise Solution with Nvidia 4090 GPUs

Cost Component	**OpenAI API (Cloud-Based)**	**On-Premise Solution (2x Nvidia 4090 GPUs)**
Initial Setup	None; pay-per-use model	$5,000 - $8,000 (GPUs, CPU, RAM, Storage) - Estimated
Hardware	N/A	Nvidia 4090 GPUs: ~$1,600 each CPU, RAM, Storage: ~$3,000
API Usage	~$2.5 to $10 per 1M tokens (GPT-4)	N/A
Operational Costs (Monthly)	varies based on the level of usage.	Electricity bill (~$100/month for both GPUs) Maintenance is Variable
Scalability	Easily adjustable for different usage levels.	Limited by hardware; may require more GPUs for scaling
Data Privacy	Concerns about data privacy	Full control over data; no external sharing

Customization	Limited to built-in features.	Can be fully customized to your needs.
Maintenance & Updates	OpenAI handles everything; no effort needed.	You need to handle maintenance and updates.
Access to Latest Models	Always up to date with the newest models.	Updates need to be done manually.
Performance	Depends on OpenAI's servers.	Can be very fast with good hardware.

Working with Product Details

Since the chatbot needs to provide support using product PDFs, we explored tools that can read and process these documents. We reviewed open-source tools like Apache Tika, spaCy, and PyPDF2 to ensure the bot can extract accurate information from PDFs. This will help the bot answer questions about product features, setup instructions, and troubleshooting. *(Apache Tika, 2024 & PyPDF2, 2024)*

Learning from Other Chatbots:

We also studied reports and research on customer service chatbots to understand how they reduce response times and improve user satisfaction.

2.2 Description of the Evaluation Tool

To choose the best technology for the chatbot, we created a tool to evaluate and compare different options. This tool scores each option based on several key criteria that align with our project goals

Performance & Privacy: LLaMA ensures 80% accuracy when answering questions and keeps data secure by running locally, outperforming cloud-based models. *(Llama, 2024)*

Cost & Integration: LLaMA is open-source and affordable, and it integrates smoothly with tools like LangChain and PyPDF2 for handling product PDFs. *(Llama, 2024; LangChain, 2024. & PyPDF2, 2024)*

Scalability & Support: Being open-source, it can be fine-tuned for specific needs, and it has strong industry support for reliability and customization. *(Meta AI, 2024)*

We selected LLaMA 3.2 because it meets all these requirements effectively. *(Llama, 2024)*

Demonstration of the Evaluation Tool

To find the best technology for our local AI chatbot, we created an evaluation tool that compares options based on key project goals like performance, privacy, cost, and integration. *(Meta AI. 2024; Chibber, A. (2024) & OpenAI, 2024)*

Evaluation Criteria and Scoring System

The tool uses a weighted scoring system to rank each solution from 1 to 5, where 1 is poor and 5 is excellent.

Criteria	Weight	Description
Performance and Accuracy	30%	How well the model answers queries and understands product specifications.
Data Privacy	25%	Whether the solution can be run on-premise, ensuring no customer data leaves the local environment.
Cost	20%	Overall cost of setup and maintenance, including hardware and software expenses.
Ease of Integration	15%	How easily the solution integrates with product specification PDFs and other necessary systems.
Scalability and Customizability	10%	The ability of the solution to scale and be customized as the project expands or new product lines emerge.

Each solution was evaluated based on these criteria, and a weighted score was calculated to identify the best option. Below is an example of how this evaluation tool was used to compare three top candidates: OpenAI GPT-4, LLaMA 3.2, and Gemma. *(Meta AI, 2024; Chibber, A. (2024) & OpenAI, 2024)*

Evaluation Demonstration

Performance and Accuracy

OpenAI GPT-4: 5/5 – GPT-4 excels in understanding natural language and provides highly accurate responses for product questions and comparisons. *(OpenAI, 2024)*

LLaMA 3.2: 4/5 – LLaMA is accurate for most queries but struggles slightly with edge cases compared to GPT-4. *(Llama, 2024)*

Gemma: 3/5 – Gemma performs well on basic queries but has difficulty handling complex or technical ones. *(Chibber, A. 2024)*

Data Privacy

OpenAI GPT-4: 2/5 – Requires cloud-based processing, which poses risks for sensitive data.

LLaMA 3.2: 5/5 – Runs entirely on local systems, ensuring full data privacy and security.

Gemma: 5/5 – Similar to LLaMA, Gemma operates locally, providing complete control over data.

Cost

OpenAI GPT-4: 2/5 – Expensive due to cloud usage fees and potential data transfer costs.

LLaMA 3.2: 5/5 – An open-source model with low hardware requirements, making it highly cost-effective.

Gemma: 4/5 – Requires moderate initial hardware investment but has no ongoing costs, making it affordable over time.

Ease of Integration

OpenAI GPT-4: 4/5 – Offers a variety of APIs and tools but faces challenges in processing local documents due to cloud restrictions.

LLaMA 3.2: 4/5 – Works seamlessly with tools like Apache Tika, making it effective for processing product PDFs.

Gemma: 4/5 – Like LLaMA, Gemma integrates easily with open-source tools.

Scalability and Customizability

OpenAI GPT-4: 4/5 – Scalable but limited by cloud infrastructure and high costs, which may restrict flexibility.

LLaMA 3.2: 5/5 – Highly scalable and customizable due to being open-source, allowing for future changes and expansions.

Gemma: 4.5/5 – While scalable, Gemma's larger model requires higher initial hardware investment compared to LLaMA 3.2.

Sample Weighted Scores Calculation

Criteria	**Weight**	**OpenAI GPT-4o**	**LlaMA 3.2**	**gemma**
Performance and Accuracy	30%	5 * 30% = 1.5	4 * 30% = 1.2	3 * 30% = 0.9
Data Privacy	25%	2 * 25% = 0.5	5 * 25% = 1.25	5 * 25% = 1.25
Cost	20%	2 * 20% = 0.4	5 * 20% = 1	4 * 20% = 0.8
Ease of Integration	15%	4 * 15% = 0.6	4 * 15% = 0.6	4 * 15% = 0.6
Scalability and Customizability	10%	4 * 10% = 0.4	5 * 10% = 0.5	5 * 10% = 0.5
Total Score	**100%**	**3.4**	**4.55**	**4.05**

2.3 Sample Accuracy Questionnaire

Appliances Evaluated: Samsung Refrigerator, LG Refrigerator, Whirlpool Washing Machine, Bosch Dishwasher.

Scoring Criteria:

Energy Efficiency: 5 points for Energy Star certification; all appliances qualify.

Capacity: Samsung (28 cu. ft.), LG (26.2 cu. ft.), Whirlpool (4.8 cu. ft.), Bosch (16 place settings). Points assigned based on size.

Features: Advanced features (5 points) to basic (1 point). Examples include Samsung Family Hub, LG SmartThinQ, Whirlpool Load & Go™, and Bosch PrecisionWash™.

Evaluation Results

LLaMA 3.2: Scored 4.35, excelling in privacy, cost, scalability, and integration, making it the best choice.

Gemma: Scored 4.0, strong in cost and privacy but weaker in performance and integration.

GPT-4: Scored 3.4, hindered by cloud dependence, higher costs, and privacy issues.

Outcome

LLaMA 3.2 was the optimal choice, meeting project goals for privacy, cost-effectiveness, and easy integration.

CHAPTER THREE: PROJECT PLAN AND BUDGET DEVELOPMENT

3.1 Objective

Create and set up a local AI chatbot to improve customer service by answering product questions, providing technical support, and protecting user data.

3.2 Task Breakdown

The project is divided into clear phases, each with specific tasks, team members, and timelines. Every phase is essential to achieving the project's goals.

Task	Description	Team Member(s)	Duration (Approx)

Project Initialization	Define project requirements, set objectives, assign team roles, and finalize scope of work	Entire Team	1 month
Infrastructure Setup	Procure necessary hardware (e.g., Nvidia 4090 GPUs), set up the software environment for the project	Everyone in the Team	45 days
AI Model Selection & Training	Fine-tune the LLaMA 3.2 model using product-specific data, ensure it is optimized for customer support	AI Team	2-3 months
Product PDF Integration	Implement PDF parsing tools (e.g., LangChain, Apache Tika, PyPDF2) to enable chatbot to interpret product manuals.	AI Team	45 days
Chatbot Testing & Refinement	Test the chatbot with product manuals and real-world queries, achieve 90% accuracy target for responses	QA Team	2 months
User Feedback Gathering	Gather feedback from initial users; make refinements as needed to improve chatbot performance	Support Team	Continuous
Build & Release Activities (Final Deployment)	Deploy chatbot to the customer service environment, coordinate with build and release teams	Build & Release Team	1 month (And we can deploy, if there are any updates.

Monitoring & Maintenance	Provide post-launch support, monitor performance, fix bugs, and make necessary updates	Support Team	Continuous

3.3 Develop Budget

In this step, we will estimate the costs for developing and implementing the LLaMA-based chatbot. The budget is divided into hardware and software costs (for both the development and production phases) and personnel costs. To keep this academic project affordable, we will use existing resources and assign tasks to team members.

Hardware and Software Costs

The budget is split between the development phase and the production phase. During development, we will use readily available hardware, like laptops with Nvidia 4090 GPUs or Apple M1 Max chips. For production, we will plan for scalable infrastructure using dedicated servers.

Project Development Costs

In the development phase, we will use laptops with built-in GPUs, which helps to significantly lower hardware expenses.

Item	**Quantity**	**Unit Cost**	**Total Cost**	**Justification**
Apple M1 Max or Nvidia 4090 Laptop	1	$3,000 - $5,000	$3,000 - $5,000	Laptop used for model training, testing, and development. Both options fall within this budget.

LLaMA Model (Open-source)	N/A	Free	Free	No licensing fees for using the LLaMA model.
PDF Parsing Tools	N/A	Free	Free	Open-source tools such as LangChain, PyPDF2, and Apache Tika for document processing.

Development Maintenance Costs	Monthly	$50	$600/year	Electricity and minimal upkeep costs for running development setups.

Total Project Development Costs: **$3,600 - $5,600**

Production Deployment Costs: For the production phase, dedicated hardware is required to ensure scalability, better performance, and reliable, continuous operations.

Item	**Quantity**	**Unit Cost**	**Total Cost**	**Justification**
Nvidia 4090 GPUs	2	$1,600	$3,200	High-performance GPUs to ensure the chatbot can handle real-time customer queries.
CPU, RAM, and Storage	1	$3,000	$3,000	Hardware required to support the on-premise production setup.
LLaMA Model (Open-source)	N/A	Free	Free	Open-source model, which eliminates licensing fees.
PDF Parsing and other Tools and Libraries	N/A	Free	Free	Tools like Apache Tika and PyPDF2 for document handling in production are open-source.

Maintenance Costs	Monthly	$100	$1,200/year	Estimated electricity and upkeep costs for running production servers.

Total Production Deployment Costs: **$7,400**

Personnel Costs

For this academic project, personnel costs are $0 since team members handle multiple roles, gaining hands-on experience in AI, DevOps, QA, and project management. In a real-world scenario, personnel costs would be approximately $16,650.

3.4 Project Budget

The total budget ranges from $3,600–$5,600 for development and $7,400 for deployment. By using existing laptops and open-source tools, we've kept costs low, making this a cost-effective and long-term solution.

Financial Analysis

Cost Savings: Setting up on-premise with Nvidia GPUs costs $5,200–$8,500 upfront. In contrast, cloud-based models like GPT-4 can cost thousands of dollars monthly, increasing significantly with more usage.

Operational Costs: On-premise setups cost about $100 per month, offering predictable expenses, while cloud costs can range from $24,000–$48,000 annually.

Customization & Scalability: On-premise solutions like LLaMA provide complete customization at no extra cost, whereas cloud models charge additional fees for advanced features and scalability.

3.5 Return on Investment (ROI) and Payback Period

Initial Investment: $5,200–$8,500 (one-time hardware cost).

Cloud Costs: $24,000–$48,000 annually.

Annual Savings: $15,500–$40,000 after operational costs.

3-Year Savings: $63,000–$114,000, with an ROI of nearly 1000%.

Payback Period: 4–6 months to recover the initial investment, followed by continued savings.

CHAPTER FOUR: RISK ANALYSIS AND MITIGATION PLAN

What is Risk Analysis?

Risk analysis in a project means identifying possible problems, figuring out how likely they are to happen, and understanding how much they could impact the project. It can be explained with a simple formula:

Risk = Probability of Occurrence × Impact of the Event

What is Risk Mitigation?

Risk mitigation involves taking steps to reduce the impact of potential problems, like cyberattacks or natural disasters. This includes actions to lower risks and protect the business. Different businesses use different methods for this, but the main goal is to be prepared and minimize any negative effects. *(Lutkevich, B. 2024, July 5)*

4.1 Technical Risks

4.1.1 Model Performance and Accuracy

Risk: The LLaMA model may struggle with complex or specific questions due to limitations in its pre-trained data. For example, it might fail to provide detailed firmware update instructions for a smart thermostat, frustrating users and reducing trust in the chatbot. This could lead to more reliance on human support and increased costs, with a medium chance of happening. *(Cascella, L. M. 2023, May 1)*

Mitigation Plan: Fine-tune LLaMA 3.2 using AWS SageMaker to improve accuracy and scalability.

Add preprocessing to fix user input errors and run regular accuracy and latency tests.
Use human oversight as a backup for complex queries, ensuring better handling of common questions. *(Kumar, M. 2024, June 15)*

Resources Needed: AWS SageMaker, testing tools, ML engineers, and a training budget.

4.1.2 Hardware Limitations

Risk: Nvidia 4090 GPUs may not handle heavy traffic or more features in the future, causing slow replies, system crashes, and unhappy users during busy times like sales events. This has a medium chance of affecting the system's scalability and user experience.

Mitigation Plan: Keep checking GPU performance, use load balancing to spread the work, and tools like HD Sentinel to find problems early. Manage tasks carefully during busy times and plan to upgrade or add GPUs. Use RAID storage for keeping data safe. Resources needed are monitoring tools, IT staff, and extra GPUs or cloud options. *(EM360 Tech. 2021, May 15)*

4.1.3 Integration with PDF Parsing Tools

Risk: PDF tools like PyPDF2, Apache Tika, and Lang Chain may not work well with complex document formats, causing errors or wrong text extraction. This can confuse users, especially for technical questions, and lower trust in the chatbot. The chance of this happening is medium to high (*Rozy, N. 2022, June 28)*

Mitigation Plan: Use tools with OCR and pattern detection to extract data better. Add systems to check and fix errors automatically and use machine learning to manage complex formats. For important documents, do manual checks. Invest in training and testing to make the chatbot more reliable and scalable over time. *(Valleskey, B. 2024, April 3)*

4.2 People-Related Risks

4.2.1 Skill Gaps in AI and DevOps Roles

Risk: The team's limited experience in AI and DevOps could lead to delays, errors, or poor system performance, especially during high user loads. This may harm reliability, user experience, and increase project costs. *(Landsman, I. 2024, April 10)*

Mitigation Plan: Hire external experts to train the team in AI and DevOps best practices. Provide regular training on cloud systems, model optimization, and deployment pipelines. For complex tasks, collaborate with skilled vendors to ensure quality while managing minor delays. *(Microsoft. 2024)*

4.2.2 Limited QA Resources

Risk: Without a QA team, rare bugs might not be noticed during testing, causing problems like system crashes or slow responses after launch. This can hurt user experience and require costly fixes. The chance of this happening is high and could affect the chatbot's reliability.

Mitigation Plan: Test all important chatbot features thoroughly using detailed plans and automated checks after updates. Focus on critical functions and hire experts for tasks like performance or security testing. Allow small delays to fix less important issues and ensure the system is stable. *(Testlio. 2024, April 19)*

4.3 Process-Related Risks

Risk: Delays in getting hardware or setting up software can slow down the project, especially in important stages like model training and testing. For example, if GPUs arrive late, it can affect performance optimization and deployment, causing missed deadlines and opportunities. Such delays are common and can disrupt the overall schedule. *(Aha! 2024)*

Mitigation Plan:

Keep IT strategies updated, monitor infrastructure needs, and have backup plans for hardware or cloud delays. Use tools like Jira to adjust timelines and focus on important tasks. Hire

third-party vendors to set up infrastructure quickly and reduce the load on the team. *(Planview 2024)*

4.3.1 Tight Project Timeline

Risk: A tight project timeline allows little room for unexpected delays, which can affect important stages like testing and debugging. For example, if LLaMA 3.2 model training is delayed, testing might be rushed, leading to missed errors and poor chatbot performance. This can reduce user satisfaction, increase fixes after launch, and harm stakeholder trust.

Mitigation Plan: Use Agile project management with short, step-by-step development cycles to prioritize tasks and prevent delays. Set aside enough resources, tools, and extra time to manage unexpected issues. Use tools like Jira for task management and keep communication clear to handle risks quickly and keep the project on schedule. *(Eby, K. 2022, November 15)*

4.4 External Factors

4.4.1 Regulatory Compliance

Non-compliance with data privacy laws like GDPR could lead to legal issues, financial penalties, and loss of user trust. For instance, failing to implement proper data protection measures might result in delays, costly adjustments, and harm the chatbot's reputation. Users may abandon the service if they feel their data isn't handled securely.

Mitigation Plan:

Conduct a Data Protection Impact Assessment (DPIA) to identify privacy risks and incorporate Privacy by Design principles from the start. Implement clear consent mechanisms, informing users about data usage and obtaining explicit approval. Hire legal experts to ensure compliance, use secure data tools like encryption, and focus on limited data processing where necessary.

4.4.2. Market Dynamics and Competition

Risk: Market changes or competition could threaten the chatbot's success. A competitor might release a superior product with better features or user experience, reducing the chatbot's user base. Additionally, declining interest in AI customer support could hinder the project's growth and profitability.

Mitigation Plan: Conduct regular market analysis to track trends, competitors, and customer preferences, while investing in R&D to improve features like advanced NLP. Implement flexible pricing and targeted marketing strategies to attract and retain users. Release incremental updates based on user feedback and explore partnerships to expand reach and share risks. *(Investopedia. 2024, August 20)*

CHAPETR 5 : CONCLUSION

The integration of Machine Learning (ML) and Artificial Intelligence (AI) into customer service addresses critical issues like delayed responses, unhelpful interactions, and data privacy concerns. This project focuses on a locally deployed AI chatbot to overcome the cost, customization, and privacy limitations of cloud-based solutions. The decision to use the LLaMA 3.2 model, coupled with tools like Apache Tika and PyPDF2, ensures a secure, cost-effective, and scalable solution. By automating routine tasks and improving response accuracy, the chatbot enhances customer satisfaction and operational efficiency. This topic was chosen for its relevance and transformative potential in redefining customer service. It bridges cutting-edge AI technology with practical applications, making it accessible to businesses of all sizes. Ultimately, this project demonstrates how AI can foster trust, loyalty, and operational excellence, setting a new standard for customer support in the digital age.

5.1 Practical recommendations for implementing the proposed solution.

To build the AI chatbot, start by using laptops with Nvidia 4090 GPUs for training during development and set up stronger servers for when it's ready for use. Train the LLaMA 3.2 model using AWS SageMaker with data from product manuals to make it 90% accurate. Use tools like Apache Tika and PyPDF2 to read product PDFs, and add OCR for difficult files. If the tools fail, have a backup plan for manual checks to fix errors.

Train your team to improve their skills in working with AI models, managing software, and building chatbots. Test the chatbot carefully to make sure it gives correct answers and works well with tools like PDF readers. Simulate busy times to check if it can handle a lot of users without slowing down. Let users give feedback after testing so you can fix issues and make the chatbot better.

Prepare for problems like delays in getting equipment by having backup options for hardware and hiring experts for tough tasks. Make sure all user data is safe by running the chatbot locally and using encryption to protect information. Keep an eye on hardware like GPUs with tools like HD Sentinel and plan for adding more servers or using cloud options as the number of users grows. To make the chatbot successful, watch what competitors are doing and keep improving based on user suggestions. Work with product companies to expand what the chatbot can do. Use free tools and existing hardware to save money, only upgrading when needed. Keep the chatbot updated and running smoothly to give users a reliable experience.

Project Implementation Overview

5.1.1 Chatbot Design:

User Interface: A simple and easy-to-use design where users can chat and upload documents.

Backend System: A smart system that combines information retrieval with generating answers.

Uses a database to quickly find the right information for user queries.

Knowledge Base: Stores important data from product manuals and allows quick updates.

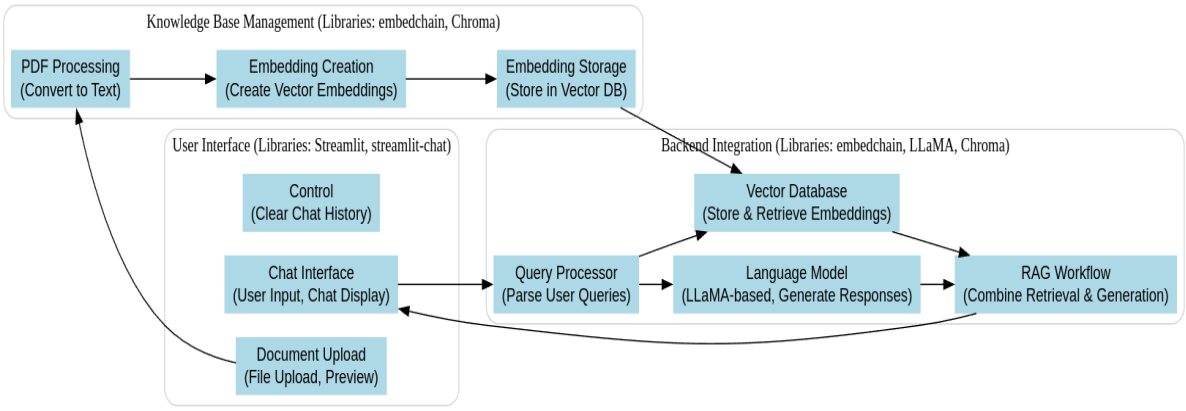

Figure 5.a: Architecture diagram of the sample implementation.

5.1.2. How the Chatbot Works:

Preparing Data: Product manuals are processed into a searchable format (embeddings).

Processing Questions: The chatbot searches the database for relevant information based on user queries.

Creating Responses: A powerful AI model generates accurate and clear replies in real time.

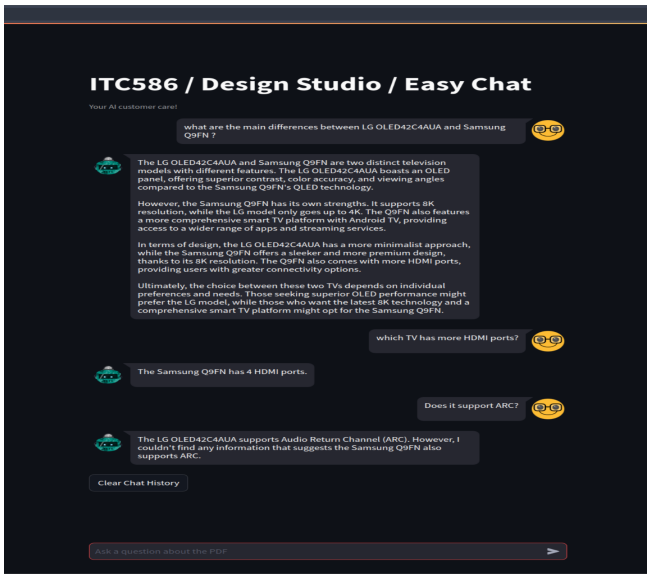

Figure 5.b : Sample Implementation demonstrating queries comparing two TVs.

5.1.3 Key Technologies:

RAG Framework: Combines retrieving information and generating responses. (AWS, n.d)

AI Language Model: Understands user questions and creates answers.

Vector Database: Stores and quickly retrieves data.

Interactive Interface: Simple design for chatting and uploading documents.

5.1.4. Special Features:

Quick Responses: Answers appear in real time without long waits.

Easy Updates: New documents can be added at any time to keep information current.

Chat History: Users can continue conversations without losing context.

5.1.5. Flexible and Scalable:

Can Grow: The system can handle more users and data as needed.

Supports More Files: Currently supports PDFs but can be expanded to include Word and Excel files.

6. Future Improvements:

Multi-Language Support: Let users ask questions in different languages.

Emotion Detection: Understand user feelings to give better responses.

Personalization: Keep track of user-specific queries for better assistance.

Analytics: Provide data to businesses to improve customer care.

REFERENCES

Aha! (2024). IT infrastructure assessment checklist. *Aha!* Retrieved from https://www.aha.io/roadmapping/guide/it-plans/infrastructure-assessment-checklist

Adam, M., Wessel, M., & Benlian, A. (2020). AI-based chatbots in customer service and their effects on user compliance. *Electronic Markets, 30*(1), 41–57. https://doi.org/

Aggarwal, B., & Subhashis, J. (2024, May 7). The role of machine learning (ML) in customer service. *Sprinklr.* Retrieved from https://www.sprinklr.com/blog/machine-learning-in-customerservice/

Amazon Web Services (AWS). (n.d.). *What is retrieval-augmented generation?* Retrieved December 9, 2024, from https://aws.amazon.com/what-is/retrieval-augmented-generation/

Apache. (n.d.). Apache Tika: A content analysis toolkit. *Apache.* Retrieved October 7, 2024, from https://tika.apache.org/

Blanco, P. (2024, January 3). The security issues under LLMs like GPT, LLaMA, and Bard. *Rootstrap.* Retrieved from https://www.rootstrap.com/blog/the-security-issues-under-llms-like-gpt-llama-and-bard

Cascella, L. M. (2023, May 1). Let's chat: Examining top risks associated with generative artificial intelligence chatbots. *MedPro Group.* Retrieved from https://www.medpro.com/risks-associated-with-ai-chatbots

EM360 Tech. (2021, May 15). Understanding computer hardware risk mitigation methods. *EM360.* Retrieved from https://em360tech.com/tech-article/understanding-computer-hardware-risk-mitigation-methods

Intercom. (n.d.). What is the role of machine learning in customer service? Retrieved from https://www.intercom.com/learningcenter/machine-learning-in-customer-service

IBM. (n.d.). AI vs. machine learning vs. deep learning vs. neural networks. *IBM*. Retrieved from https://www.ibm.com/topics/artificial-intelligence

Kemper, J. (2017). The power of online customer reviews in fashion e-commerce: An empirical analysis across categories and brands. *Proceedings of the 25th European Conference on Information Systems (ECIS)*. Retrieved from https://aisel.aisnet.org/ecis2017_rp/29

Kumar, M. (2024, June 15). Llama 3 fine-tuning: Strategies for efficient LLM inference in task execution. *Medium*. Retrieved from https://medium.com/@manojkumar/llama-3-finetuning-strategies-for-efficient-llm-inference-in-task-execution

LangChain. (n.d.). LangChain. Retrieved October 7, 2024, from https://www.langchain.com/

Lutkevich, B. (2024, July 5). What is risk mitigation? Strategies, plan and best practices. TechTarget. Retrieved from https://www.techtarget.com/searchdisasterrecovery/definition/risk-mitigation

Meta AI. (2024). Introducing LLaMA 3.2: Revolutionizing edge AI and vision with open, customizable models. *Meta AI Blog*. Retrieved October 7, 2024, from https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/

Microsoft. (2024). How to choose a cloud service provider. *Microsoft Azure*. Retrieved from https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/choosing-a-cloud-service-provider

PyPDF2. (n.d.). PyPDF2 documentation. *Read the Docs*. Retrieved October 7, 2024, from https://pypdf2.readthedocs.io/en/3.x/

Rebecca Jen-Hui Wang. (2020). Branded mobile application adoption and customer engagement behavior. *Computers in Human Behavior*. Retrieved from https://doi.org/10.1016/j.chb.2020.106245

Rozy, N. (2022, June 28). How to mitigate the risk of cloud downtime. *Fast Company.* Retrieved from https://www.fastcompany.com/90761917/how-to-mitigate-the-risk-of-cloud-downtime

Testlio. (2024, April 19). The ultimate guide to functional testing. *Testlio.* Retrieved from https://testlio.com/blog/ultimate-guide-to-functional-testing

MacBook Pro 16.2" with Liquid Retina XDR Display, M3 Max Chip with 16-Core CPU and 40-Core GPU, 64GB Memory, 2TB SSD, Space Black, Late 2023

https://a.co/d/b5EQMEg

https://a.co/d/gnyXwRw

https://a.co/d/2btNlpE

https://a.co/d/fxK4neE

https://a.co/d/iGNwSBL

https://a.co/d/2kcN515

Youtube Link: https://youtu.be/qEOblxT9CXw