# CS 25-308 SBSD Web Prototype & GPT

# Preliminary Design Report

Prepared for

willis.morris@sbsd.virginia.gov/VA Dept Small Business and Supplied Diversity

By

Zachariah Dellimore, Jacobo Ceballos, Nate Swetlow, and Victor Olivar

Under the supervision of

John Leonard

Date

October 11, 2024

# Executive Summary

The large volume of user inquiries on the Virginia Department of Small Business & Supplier Diversity's (SBSD) website is a substantial difficulty that frequently causes confusion and delays in the retrieval of information. The department has used the ChatGPT API to create a chatbot in the past, but it had accuracy problems because the responses were often taken from outside sources and did not match the department's official records.In order to meet the requirement for a more efficient solution, this project builds a unique large language model (LLM) that is suited to the internal data of SBSD. The approach we selected is based on the Retrieval-Augmented Generation (RAG) paradigm, which combines real-time generative replies with document retrieval. This method guarantees accurate and flexible responses, giving users dependable, context-sensitive information. By implementing this methodology, SBSD will drastically cut down on the amount of routine questions that staff members get, enabling users to swiftly and effectively use self-service choices. The system will also be expandable, allowing it to accommodate future content expansions and more sophisticated user inquiries.

This project complies fully with Executive Order 30, which sets moral and responsible guidelines for the use of AI in all state agencies in Virginia. In accordance with the Virginia Information Technologies Agency's (VITA) principles for artificial intelligence in public service, the directive guarantees that AI systems are transparent, dependable, and safe. The chatbot solution is compatible with SBSD's current IT infrastructure and complies with Enterprise Architecture (EA) requirements, guaranteeing strong security, data privacy, and operational effectiveness.

Following these state-level regulations and making use of cutting-edge AI technologies, this project not only solves an urgent operational requirement but also acts as a prototype for upcoming AI-driven public sector service solutions. It establishes a new benchmark for raising citizen-government relations, expediting service delivery, and cutting expenses for SBSD and other state agencies.

## Table of Contents

## Section A. Problem Statement

The Virginia Department of Small Business & Supplier Diversity (SBSD) has identified a significant gap in how users access information on their website, which results in confusion and frustration for users trying to find the resources they need. This issue leads to an increased number of calls and emails to the department for basic inquiries that could otherwise be resolved through self-service, such as eligibility requirements for certifications, application procedures, and contact information for different divisions. As a result, small businesses seeking support experience delays in receiving guidance, and department staff face inefficiencies in managing these inquiries. In response to this problem, the department previously implemented a chatbot using the ChatGPT API. While the solution demonstrated potential, it faced several shortcomings, primarily due to its reliance on external information sources from the internet, which often led to inaccurate or misleading answers that did not align with the department's official resources. This issue, known as "hallucination" in natural language processing, diminished the chatbot's credibility as a reliable source of information.

To address this problem, our project aims to develop a custom large language model (LLM) tailored specifically to the Virginia SBSD's needs. Rather than relying on third-party APIs or external data, we will create and fine-tune an LLM using the department's own internal data and website content, ensuring that the chatbot provides accurate, up-to-date, and contextually relevant responses to user queries. This project falls under the fields of conversational AI and business intelligence, contributing to advancements in how government agencies can leverage artificial intelligence to improve public service accessibility. By creating a custom LLM trained on domain-specific data, we push the boundaries of existing chatbot solutions and set a precedent for other agencies facing similar challenges in information dissemination.

Historically, various solutions have been employed to improve information accessibility, ranging from static FAQ pages to traditional rule-based chatbots. However, these approaches often lacked the ability to understand context or provide nuanced responses, leading to poor user satisfaction. Commercially available chatbots, while more sophisticated, also face limitations when it comes to providing information highly specific to an organization's needs. Our project builds on these previous attempts by integrating cutting-edge natural language processing techniques with tailored training data, allowing the chatbot to serve as a more effective digital assistant for SBSD. In addition to addressing an unmet engineering need for the department, this project has the potential to reduce operational costs associated with handling basic inquiries, improve public perception of the department's digital resources, and create a blueprint for other government entities looking to implement AI-driven solutions. The primary stakeholders in this project include the Virginia SBSD, who will benefit from the reduced volume of basic inquiries and the enhanced user experience on their website, as well as the end users, which include small business owners, aspiring entrepreneurs, and other stakeholders seeking information about certification processes and services offered by the department. By the end of this project, we aim to deliver a chatbot solution that not only addresses the current limitations but also elevates the standard for public sector digital communication tools, ultimately enhancing the accessibility and efficiency of information dissemination for the Virginia SBSD and providing a positive impact on the broader community it serves.

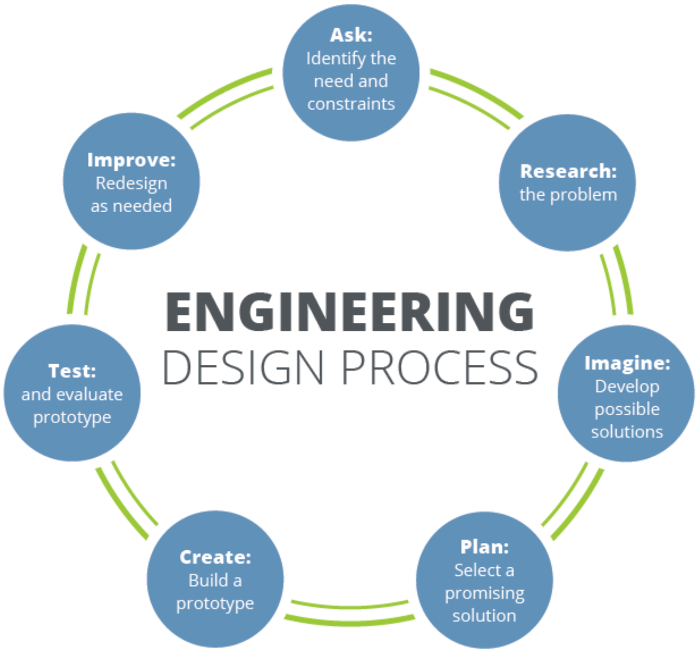


**Figure 1. The iterative nature of the engineering design process [2].**

## Section B. Engineering Design Requirements

This section describes the goals and objectives of the project, as well as all **realistic constraints** to which the design is bound. Our goals are to create a chat-bot to help the Department of Small Business and Supplier Diversity site users navigate the site to find what they need. The chat-bot also has to follow Executive Order 30's rules for Commonwealth of Virginia AI systems. We will also have to develop a web interface for SBSD to test the chat-bot and ensure it meets their requirements.

### B.1 Project Goals (i.e. Client Needs)

There are a number of goals that this project seeks to achieve when improving the chat-bot implementation from last year:

- To create a chat-bot that helps SBSD site users find the necessary information on the website.
- To create a chat-bot that follows Executive Order 30's rules on AI.
- To create a way of checking that the chat-bot is not hallucinating or misleading users.
- To implement a web interface so the sponsor's can test the chat-bot implementation.

### B.2 Design Objectives

The following are a list of key objectives that we hope to achieve by the end of the spring semester:

- The chat-bot will give the correct information to questions users ask about the site.
- The chat-bot will not answer questions it does not have data on.
- The chat-bot will not break any of Executive Order 30's rules on AI.
- The chat-bot will have a way to be tested to ensure it gives correct responses.
- The design will include a web interface for users to test the chat-bot online.

### B.3 Design Specifications and Constraints

The following are a list of constraints for the chat-bot design to ensure it can be used on the SBSD website without issue:

- The chat-bot must be fully compliant with Executive Order 30's rules on AI which can hamper the AI's we use and whether or not we use a local AI or not.
- The chat-bot must be able to be tested to ensure it gives the correct responses.
- The chat-bot must be low cost enough while also being able to give high-quality, fast responses.

## B.4 Codes and Standards

Creating the chat-bot for SBSD requires us to follow several Codes and Standards for Commonwealth of Virginia AI systems and web systems to ensure security. A list of Codes and Standards we need to follow are shown below:

- Executive Order 30 – Chat-bot must follow a strict set of rules to ensure it gives high-quality, correct responses and doesn’t endanger user data.
- EA-225 – “Establishes direction and technical requirements which govern the acquisition, use and management of information technology resources by executive branch agencies.” (ITRM, 2020, p.7)
- WEB – The goal of this standard is to guide the use of web system resources within the Commonwealth of Virginia (VITA, 2024, p.4)

## Section C. Scope of Work

The project scope defines the boundaries of the project encompassing the key objectives, timeline, milestones and deliverables. It clearly defines the responsibility of the team and the process by which the proposed work will be verified and approved. A clear scope helps to facilitate understanding of the project, reduce ambiguities and risk, and manage expectations. In addition to stating the responsibilities of the team, it should also explicitly state those tasks which fall *outside* of the team's responsibilities. *Explicit bounds* on the project timeline, available funds, and promised deliverables should be clearly stated. These boundaries help to avoid *scope creep*, or changes to the scope of the project without any control. This section also defines the project approach, the development methodology used in developing the solution, such as waterfall or agile (shall be chosen in concert with the faculty advisor and/or project sponsor). Good communication with the project sponsor and faculty advisor is the most effective way to stay within scope and make sure all objectives and deliverables are met on time and on budget.

The scope of this project is to design and implement an improved chatbot for the Virginia Department of Small Business & Supplier Diversity (SBSD). The chatbot will be built using a custom large language model (LLM) trained on the department's internal data and website content. The following outlines the project resources, milestones, and deliverables.

The primary goal is to develop a chatbot that accurately answers user queries using information from the SBSD's website, improving access to resources without over-reliance on department staff for simple inquiries. The solution will be compliant with Executive Order 30's guidelines on AI, ensuring transparency, reliability, and data privacy.

### C.1 Deliverables

The following deliverables will be produced throughout the project:

- A fully functional chatbot that can be tested via a web interface.
- Documentation detailing the system's design, implementation, and adherence to Executive Order 30.
- A testing framework to validate that the chatbot provides accurate, compliant answers.
- Capstone deliverables such as the team contract, project proposal, preliminary design report, fall poster, final design report, and Capstone EXPO presentation.

- What deliverables require access to campus? Which/how many students regularly access campus and are physically available to complete tasks?
    - o The main deliverable that may require campus access is obtaining files or code for the old front-end of the chatbot. This may involve accessing specific servers or campus networks where these resources are stored. However, the majority of

    - work can be completed remotely, minimizing the need for physical campus presence.
- What work can be done remotely? What resources might be needed in order to ensure that remote work can be completed effectively (e.g. software licenses, shared drives/folders, etc.)?
    - All core work can be done remotely. Team members can collaborate using GitHub and other online platforms.
    - The only resource we may potentially require is faster hardware for model training and chatbot execution. This can be mitigated by using cloud computing platforms if necessary, although the current infrastructure may suffice.
- What deliverables require ordering from third-party vendors? Will any components potentially required extended lead times? What can the team do in order to mitigate potential supply chain disruptions?
    - There are no expected deliverables requiring orders from third-party vendors. Since the project involves software development and the use of internal SBSD data, no physical components are needed. The project will leverage in-house data and existing frameworks, such as the Retrieval-Augmented Generation (RAG) model, to build the chatbot without reliance on external sources.

### C.2 Milestones

The following key milestones are identified for the project:

**First Quarter:**

- Initial setup, including material purchases for presentation, beginning chatbot development, and selecting a development model (e.g., Retrieval-Augmented Generation (RAG)).

**Second Quarter:**

- Complete core functionality of the chatbot, integrate testing, and build the web interface.

**Third Quarter:**

- Complete testing, address any performance issues, and prepare for the final presentation and report.

All milestones will be revisited regularly to ensure the project stays on track. The team will review progress in biweekly meetings with the sponsor and adjust the timeline as needed.

### C.3 Resources

The following resources are needed to complete the project:

- **Development Tools**: Python, natural language processing (NLP) libraries (such as Hugging Face Transformers), IDEs like Visual Studio Code or PyCharm, and cloud computing resources if required.
- **Testing Resources**: Access to internal SBSD data and any operational databases for validation.
- **Collaborative Tools**: Version control (GitHub), cloud storage, and online collaboration platforms for remote work.

All resources will be utilized to ensure effective remote work, with no expected reliance on third-party vendors or additional purchases.

## Section D. Concept Generation

Creating design ideas for the SBSD chatbot involved using various approaches. A blend of brainstorming was used, and reverse thinking techniques were applied, and existing solutions were examined for inspiration. Many different methods were reviewed for the chatbot's implementation, which led to the decision that was made that the Retrieval-Augmented Generation (RAG) model was most appropriate. The three most important design concepts were described, and evaluated by us for pros, cons, and any risks involved.

At first they thought of a rule-based chatbot that gave responses matching certain keywords or intentions from the user's questions. SBSD's resources were to be examined to identify frequently asked questions (FAQs), which would then be mapped to particular static responses. This design was seen as straightforward and relatively easy to implement. It operated on a fixed logic that made sure predictability, and easy validation. Yet in spite of allowing for quick, and low-cost deployment, large limitations were present. Its main shortcoming was a lack of flexibility. Handling variations in user phrasing or complicated queries beyond predefined patterns would be difficult for a rule-based chatbot. As a result, it would be prone to failure in many real-world scenarios, leading to user frustration. Additionally, updating the system with new FAQs would require manual changes, leading to increased maintenance over time.

For a second idea we thought about creating a chatbot using machine learning as part of a study focused on FAQs. This was designed with text similarity models, user queries could be matched to relevant FAQ entries. Embedded questions and answers from the FAQ database allowed the chatbot to handle user inputs, and identify the closest match based on semantic similarity. Therefore relying not just on keyword matching. This technique handles different phrasings, and more complicated queries better than rule-based models. Thus, more flexibility is offered, and it is more adjustable to users. Yet, increased complication in development and infrastructure was introduced. Machine learning models were implemented, and the FAQ database needed maintenance, with regular updates, and potential retraining of the model as FAQ content evolved. Accuracy was improved over rule-based methods. However, risks exist when irrelevant FAQs are retrieved due to ambiguous queries. More resources and development time were required for this model which led to higher costs.

The model we ended up selecting was the Retrieval-Augmented Generation (RAG) Model because it mixes finding documents with making new text. A query submitted by a user prompts the model to first retrieve relevant documents or FAQ entries. Then a generative model constructs a coherent and contextually appropriate response based on this information. This allows tailored responses that can be accurate and flexible. An advantage is that the model can handle complicated queries even when these queries have no exact match in the FAQ database. Responses are dynamically generated from real-time retrieval and thus nuanced, and contextually relevant answers can be offered. Because of this, the user experience is improved. Efficient scaling as the FAQ data volume increases is also aided by this design, reducing manual update needs to predefined rules.

The best solution for SBSD was the RAG model after looking at each design idea. While approaches like rule-based, and machine-learning-based retrieval chatbots offered lower costs, and could be implemented more quickly, adaptability, and scalability were missing to handle SBSD's complicated inquiries. The RAG model, with its blend of real-time document retrieval and generative responses, provides the flexibility, accuracy, and scalability needed to effectively manage diverse user queries and future expansions of the FAQ content.

## Section E. Concept Evaluation and Selection

Using a systematic decision-making process, evaluate each of the design concepts and choose the one that is most likely to succeed in meeting the design objectives and constraints. A Decision Matrix, or Pugh Matrix, helps to analyze alternatives, eliminate biases, and make rational decisions through thought and structure. First, work to develop a set of selection criteria for which to evaluate the previously generated design concepts. Selection criteria often include concepts of performance, cost, safety, reliability, risk, etc. Note that the selection criteria developed here will likely be more general than the project design objectives. As with the design objectives, conversations with the client help define appropriate selection criteria.

In many cases, the client may value the selection criteria differently, preferring that more emphasis be placed on some than others. In this case, weighting factors may be used to place more or less importance on the various criteria in the decision making process. Again, conversations with the client can be used to define criteria weighting factors. Often times, these conversations must be analyzed and interpreted by the team to determine which criteria are more important to the client and by how much. Feel free to discuss the assigned weighting factors with the client to see if they seem accurate.

Next, define an associated metric to represent each criteria. Metrics should be specific and quantifiable, providing numerical values that quantify the often vague concepts of the selection criteria. Metrics can be obtained, generated, or estimated through a number of methods including simple background research, preliminary design calculations, or basic analyses. Note that these metrics do not need to specifically align with the design specifications although there may be some commonality between the two. Provide a brief discussion of the rationale for selecting each of the assigned metrics.

Using the defined metrics, evaluated each design concept against all selection criteria by filling out a Decision Matrix. Design concepts can be compared by using simple rank scoring, raw scoring, or weighted scoring techniques and design concept with which to move forward can be selected. This type of process provides a meaningful, unbiased means for choosing a preliminary design concept prior to moving forward with more comprehensive, detailed analyses as provided in the design methodology section below. The results of this process should be discussed with the project client prior to moving forward with the selected design. Table 1 provides an example of a simple decision matrix.

**Table 1. Example of a Decision Matrix.**

| | Design Concept A | Design Concept B | Design Concept C | Design Concept D |
|---|---|---|---|---|
| Criteria 1 | | | | |
| Criteria 2 | | | | |
| Criteria 3 | | | | |
| Criteria 4 | | | | |
| Criteria 5 | | | | |
| | | | | |
| Total Score | | | | |

***Note:*** Weights can be assigned to each criterion if desired.

### Criteria

**Accuracy**: How well does the design concept provide accurate information using data from the department’s website? This addresses the main issue of hallucination in the current chatbot.

**Usability**: How easy is it for users to interact with the chatbot? This criterion ensures that the solution is user-friendly and accessible.

**Integration**: How well can the design integrate with the department’s existing systems and website infrastructure? The goal is to ensure seamless data retrieval and user experience.

**Cost**: What are the estimated development, deployment, and maintenance costs associated with each design concept?

**Scalability**: How well does the design concept handle increased traffic or additional features in the future?

**Security and Compliance**: Does the design comply with government regulations and data privacy requirements?

## Viable Design Concepts:

- **Concept A: Traditional FAQ Chatbot with Keyword Matching**
  - Simple chatbot using keyword matching to respond based on a predefined FAQ database.
  - **Pros**: Low cost, easy to implement and maintain.
  - **Cons**: Limited to basic queries, lacks flexibility.
  - **Use Case**: Suitable for handling straightforward, repetitive questions.
- **Concept B: Search-Based Chatbot Using Document Retrieval**
  - Chatbot performs semantic searches through a structured knowledge base of internal documents and provides relevant information snippets.
  - **Pros**: Able to handle a wide range of queries with high accuracy.
  - **Cons**: Slightly more complex implementation, dependent on the quality of the document base.
  - **Use Case**: Useful for answering detailed questions that span multiple documents.
- **Concept C: Hybrid Chatbot with Rule-Based Responses and External API Integration**
  - Combines rule-based responses for common inquiries with external API integration for more complex questions.
  - **Pros**: High flexibility and scalability, can handle a range of queries.
  - **Cons**: More complex due to integration with external systems.
  - **Use Case**: Effective for organizations that need a combination of internal and external data access.
- **Concept D: RAG (Retrieval-Augmented Generation) Chatbot for Accurate Information**
  - Combines document retrieval from an internal knowledge base with answer generation using an LLM.
  - **Pros**: Provides highly accurate, context-based responses; reduces hallucinations.
  - **Cons**: Complex to implement, requires regular updates to the knowledge base.
  - **Use Case**: Ideal for addressing complex, context-specific queries using verified, internal information.

| ?/6<br>6/6 = best | **Concept A: Traditional FAQ Chatbot with Keyword Matching** | **Concept B: Search-Based Chatbot Using Document Retrieval** | **Concept C: Hybrid Chatbot with Rule-Based Responses and External API Integration** | **Concept D: RAG (Retrieval-Augmented Generation) Chatbot for Accurate Information** |
|---|---|---|---|---|
| Accuracy | 0 | 0 | 1 | 1 |
| Usability | 1 | 0 | 1 | 1 |
| Integration | 1 | 0 | 1 | 1 |
| Cost | 0 | 1 | 0 | 1 |
| Scalability | 0 | 0 | 1 | 1 |
| Security & Compliance | 0 | 1 | 0 | 1 |
| Total Score | 2/6 | 2/6 | 4/6 | 6/6 |

## Section F. Design Methodology

Provide a detailed explanation of the methods that will be used to help evaluate, improve, and evolve the design through the iterative engineering design process. Consider that ultimately, the final design must be verified and validated to ensure that it meets all of the previously developed and listed design objectives and specifications. Verification ensures that the design meets all specifications, while validation confirms that the design functions as intended such to meet the client's needs. While it is common for initial design concepts to first be evaluated using simplified design criteria and metrics, the chosen design should be advanced, and later verified, using engineering calculations, computational models, experimental data, and/or testing procedures.

Use this section to describe any underlying physical principles and mathematical equations that govern the design. Provide details of any computer-aided modeling techniques used to evaluate the design including the software used, prescribed boundary conditions, and assumptions. Include a detailed description of any experimental testing methods including required testing equipment, test set-up layout, data acquisition and instrumentation, and testing procedures. If one or more prototypes is to be produced and tested, provide a detailed description of how each will be evaluated.

**Note:** The contents of this section are expected to vary from project to project. Subsections may be appropriate for providing details of analytical, computational, experimental, and/or testing methods. Some potential subsections that may be included in this section are provided. While critical design equations may be provided here, lengthy mathematical derivations may be included in an appendix. Validation procedures are critical and all projects should address such topics.

### F.1 Computational Methods (e.g. FEA or CFD Modeling, example sub-section)

The design incorporates Retrieval-Augmented Generation (RAG), supported by computational methods such as language model fine-tuning and semantic data storage using ChromaDB. The ChromaDB database supports fast semantic search capabilities by embedding SBSD data using vector representations. For computational modeling:

- Language Models: Ollama and Qwen 2.5 are fine-tuned using representative SBSD data to ensure that generated responses are accurate, relevant, and context-specific.
- Boundary Conditions and Assumptions: The database is assumed to contain clean, structured SBSD data, and the queries tested reflect common customer inquiries. The models are validated against these predefined queries.
- Software Used: Python libraries such as Flask for API integration, along with AI-specific tools like PyTorch or TensorFlow for training and testing the models.
- Evaluation Metrics: Metrics like BLEU score, response accuracy, and latency are used to measure the performance of the AI-generated responses.

### F.2 Experimental Methods (example subsection)

Experimental testing involves validating the end-to-end system performance through mock user interactions with the chatbot:

1. Test Setup Layout:

- A test environment replicating the production architecture, including the Flask backend, ChromaDB, and the Next.js frontend.
- Docker containers are used to emulate deployment conditions.

2. Data Acquisition and Instrumentation:

- SBSD-specific datasets are utilized for testing.
- Logs are captured to analyze query accuracy, response times, and potential errors.

3. Testing Procedures:

- Functional testing ensures that the chatbot produces accurate, relevant responses.
- Stress testing evaluates performance under heavy load.
- Usability testing involves mock users from SBSD simulating real-world scenarios to ensure intuitive navigation and satisfactory responses.

### F.3 Architecture/High-level Design (example subsection)

The architecture follows a modular design:

1. Frontend (Next.js): Ensures responsive and user-friendly interfaces for SBSD visitors.
2. Backend (Flask): Integrates ChromaDB for efficient query handling and language models for generating responses.
3. RAG System: Combines language generation and semantic retrieval.

- Query flows: User -> Frontend -> Flask API -> ChromaDB -> Language Model -> Response.
- Key assumption: Frontend and backend are fully synchronized for seamless operations.

4. Compliance: Designed to comply with Executive Order 30 for ethical and secure AI use.

### F.5 Validation Procedure

Describe how the design team will validate that the final design meets the client's needs. This section should include a plan to meet with the client towards the end of the project to discuss final design details and demonstrate a prototype, experimental test, and/or simulation results. Provide a relative time frame for this validation to occur (e.g. "mid-March" or "early-April"). Include a brief discussion on how client feedback will be captured, such as a formal survey, interview, or observation notes of the client using the prototype. It may also include plans to solicit feedback from other stakeholders and/or potential users.

The design validation process ensures that the final product aligns with SBSD's needs and fulfills all project objectives:

1. Client Validation Plan:

- Conduct a client demonstration of the prototype in mid-March, showcasing:
- Real-time interaction with the chatbot.
- Performance metrics like response accuracy and speed.
- Compliance with accessibility and ethical guidelines.

2. Validation Metrics:

- Functional metrics include query response accuracy, retrieval efficiency, and user satisfaction scores.
- Performance metrics include system uptime, latency, and scalability under stress testing.

3. Feedback Integration:

- Feedback from the SBSD and stakeholders will guide the refinement of the chatbot, particularly in fine-tuning responses and improving deployment pipelines.

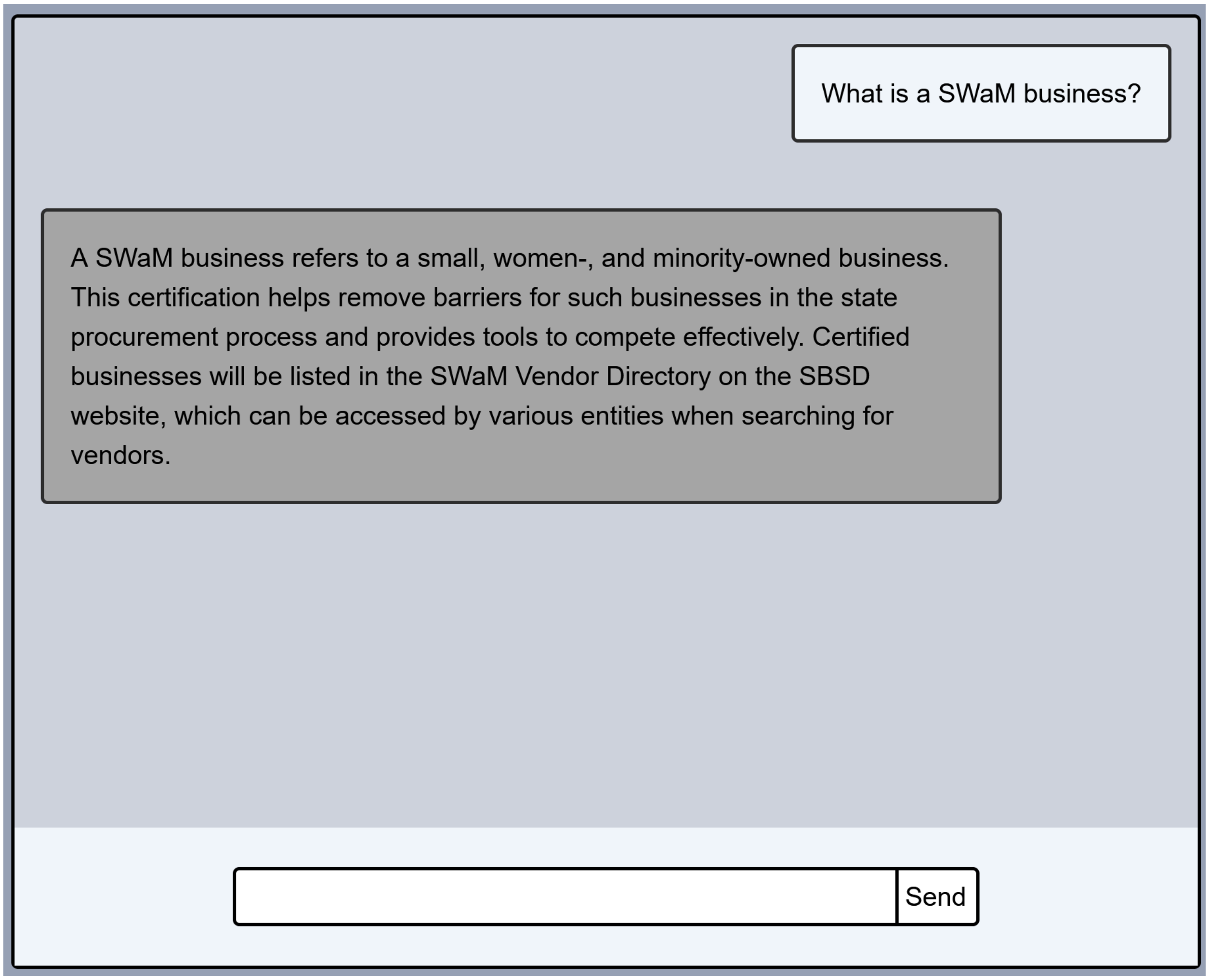
4. Prototypes:

- Initial prototypes will focus on backend and AI integration, validated via unit and system-level testing.
- The final prototype will include the fully integrated frontend-backend system.

## Section G. Results and Design Details

During our first semester we have created a frontend and backend for the SBSD chatbot using Qwen2.5 and Ollama. Utilizing Retrieval Augmented Generation, Qwen2.5 is able to respond to user questions with high quality, helpful answers without the risk of hallucinations which is necessary when following the rules set in place by Executive Order 30. The frontend was made using NextJS and React which gives it a lot of flexibility and portability which is important as our work could be a template for other Commonwealth of Virginia websites.

### G.1 Frontend Results

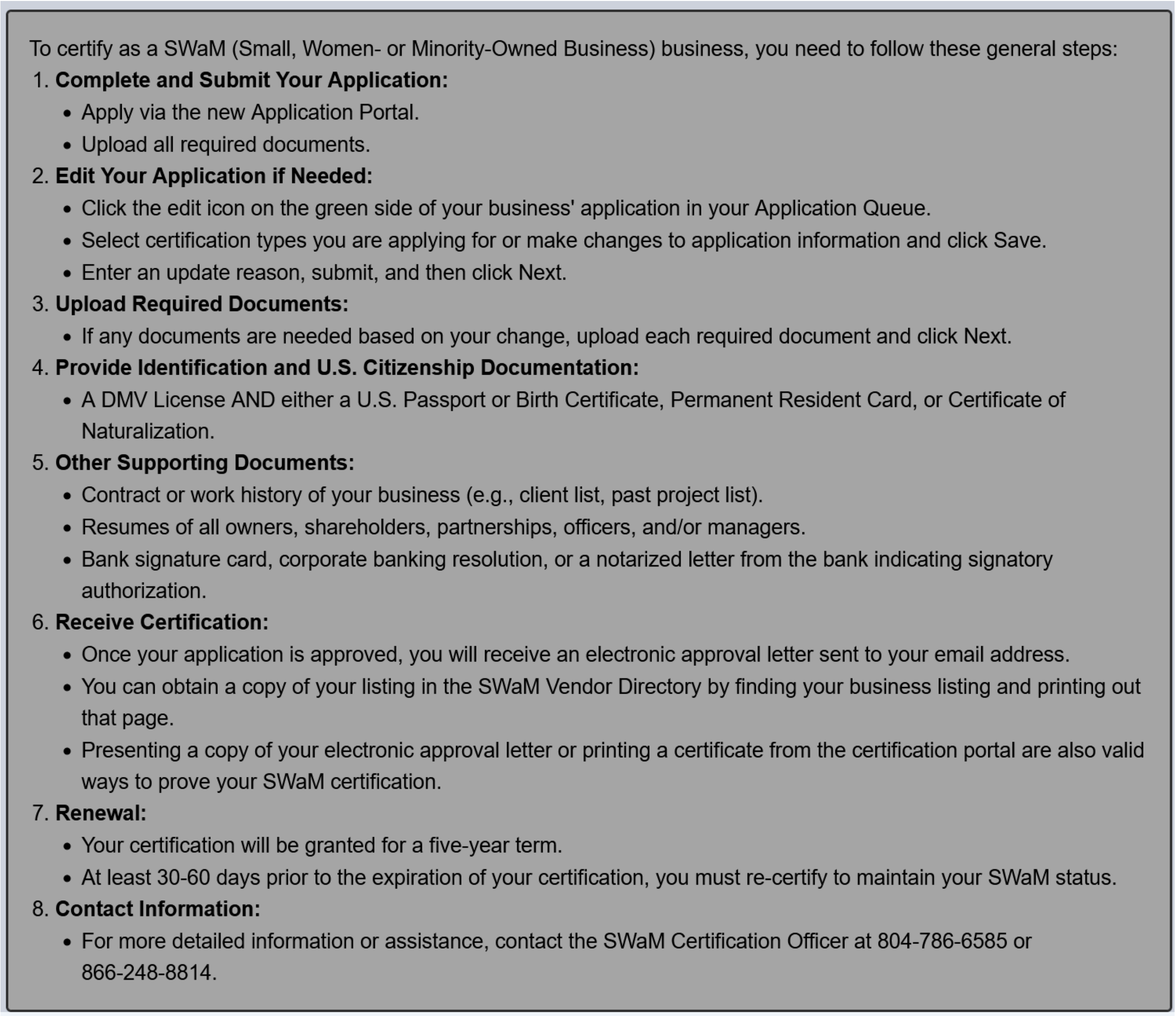


What is a SWaM business?

A SWaM business refers to a small, women-, and minority-owned business. This certification helps remove barriers for such businesses in the state procurement process and provides tools to compete effectively. Certified businesses will be listed in the SWaM Vendor Directory on the SBSD website, which can be accessed by various entities when searching for vendors.

This image shows the state of the frontend at the end of the first semester. We have a chat bar that the user can use to send messages to the chatbot and a message history so the user can see their chat history with the chatbot. The chatbot's messages are in dark grey while the user's messages are in light blue.

## G.2 Chatbot Results

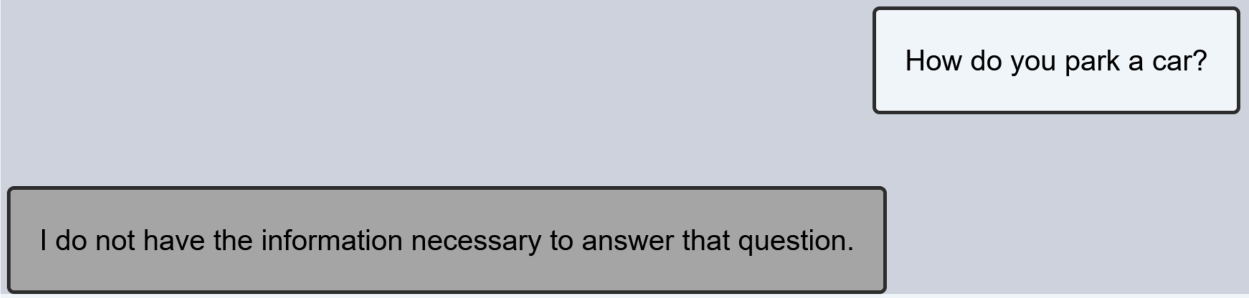


To certify as a SWaM (Small, Women- or Minority-Owned Business) business, you need to follow these general steps:

1. **Complete and Submit Your Application:**
   - Apply via the new Application Portal.
   - Upload all required documents.
2. **Edit Your Application if Needed:**
   - Click the edit icon on the green side of your business' application in your Application Queue.
   - Select certification types you are applying for or make changes to application information and click Save.
   - Enter an update reason, submit, and then click Next.
3. **Upload Required Documents:**
   - If any documents are needed based on your change, upload each required document and click Next.
4. **Provide Identification and U.S. Citizenship Documentation:**
   - A DMV License AND either a U.S. Passport or Birth Certificate, Permanent Resident Card, or Certificate of Naturalization.
5. **Other Supporting Documents:**
   - Contract or work history of your business (e.g., client list, past project list).
   - Resumes of all owners, shareholders, partnerships, officers, and/or managers.
   - Bank signature card, corporate banking resolution, or a notarized letter from the bank indicating signatory authorization.
6. **Receive Certification:**
   - Once your application is approved, you will receive an electronic approval letter sent to your email address.
   - You can obtain a copy of your listing in the SWaM Vendor Directory by finding your business listing and printing out that page.
   - Presenting a copy of your electronic approval letter or printing a certificate from the certification portal are also valid ways to prove your SWaM certification.
7. **Renewal:**
   - Your certification will be granted for a five-year term.
   - At least 30-60 days prior to the expiration of your certification, you must re-certify to maintain your SWaM status.
8. **Contact Information:**
   - For more detailed information or assistance, contact the SWaM Certification Officer at 804-786-6585 or 866-248-8814.

This image shows the level of detail the chatbot's responses are able to go into. The chatbot retrieves all of this information from the documents that it was provided to reduce hallucinations and then responds with formatted text to ensure the user can easily read the chatbot's output.

### G.3 Hallucination Protection

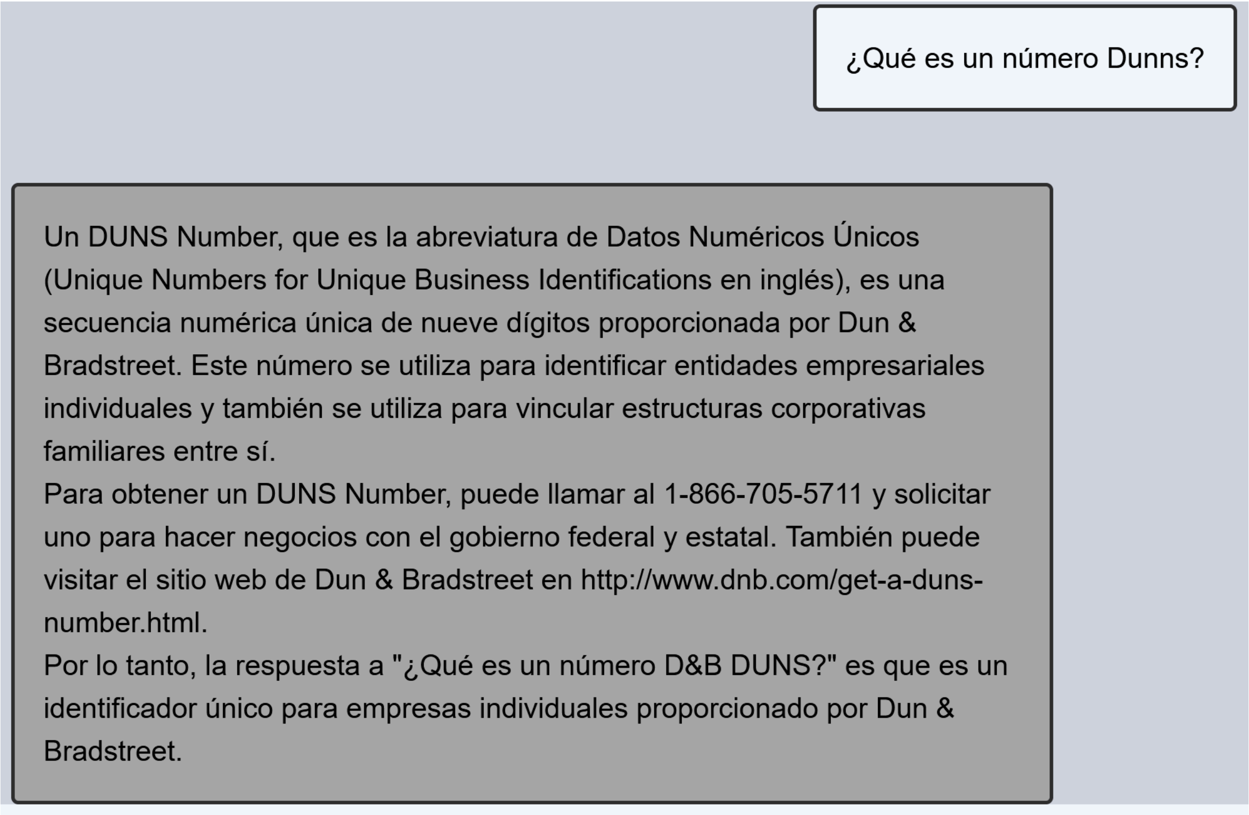


How do you park a car?

I do not have the information necessary to answer that question.

This image shows how the chatbot responds to a question that it does not have the information to answer. This is done by the chatbot to reduce hallucinations and ensure it only responds to questions where it can give information it knows is true.

### G.4. Multilingual support



¿Qué es un número Dunns?

Un DUNS Number, que es la abreviatura de Datos Numéricos Únicos (Unique Numbers for Unique Business Identifications en inglés), es una secuencia numérica única de nueve dígitos proporcionada por Dun & Bradstreet. Este número se utiliza para identificar entidades empresariales individuales y también se utiliza para vincular estructuras corporativas familiares entre sí.
Para obtener un DUNS Number, puede llamar al 1-866-705-5711 y solicitar uno para hacer negocios con el gobierno federal y estatal. También puede visitar el sitio web de Dun & Bradstreet en http://www.dnb.com/get-a-duns-number.html.
Por lo tanto, la respuesta a "¿Qué es un número D&B DUNS?" es que es un identificador único para empresas individuales proporcionado por Dun & Bradstreet.

This image shows how our chatbot is able to answer questions accurately in different languages. One of the reasons we chose Qwen2.5 as our chatbot was the fact that it is able to answer questions in over 29 languages which is very important as many people using this chatbot may not speak english as their first language so using Qwen2.5 would allow many people the ability to use the SBSD website easier.

## Section H. Societal Impacts of Design

In addition to technical design considerations, contemporary engineers must consider the broader impacts that their design choices have on the world around them. These impacts include the consideration of public health, safety, and welfare as well as the potential societal, political/regulatory, economic, environmental, global, and ethical impacts of the design. As appropriate for the project design, discuss how each of these considerations influenced design choices in separate subsections. How will the design change the way people interact with each other? What are the political implications of the design? Does the technology have the potential to impact or shift markets? Does the design have any positive or negative effects on the environment? Don't forget to consider unintended consequences such as process or manufacturing byproducts. What impacts might the design have on global markets and trade? Are there any ethical questions related to the design?

While it is hard to forecast the various impacts of a technology, it is important to consider these potential impacts throughout the engineering design process. When considered during the early stages of the design phase, consideration of these impacts can help determine design objectives, constraints, and specifications and help drive design choices that may mitigate any potential negative impacts or unintended consequences.

***Note:*** A minimum of 4 of these design considerations, including the consideration of public health, safety, and welfare, are required for the Preliminary Design Report while a section for all considerations must be included in the final design report.

### H.1 Public Health, Safety, and Welfare

Provide a list of all design safety features and provide a brief description of each. Discuss the potential effects the design may have on public health, safety, and welfare. References to the codes and standards previous provided and the organizations that produced them may be summarized or referenced here.

### H.2 Societal Impacts

The societal impacts of our RAG-based chatbot are significant, as the project addresses accessibility and inclusivity challenges in accessing government resources. By streamlining how small business owners interact with the Virginia Department of Small Business and Supplier Diversity (VDSBSD), the chatbot fosters a more equitable environment for business growth and community engagement.

One of the key societal benefits is improved access to critical information for underserved and marginalized communities. Small business owners who may face barriers such as limited time, mobility, or digital literacy can now engage with government resources more efficiently through an intuitive and user-friendly platform. The chatbot's 24/7 availability ensures that individuals can seek assistance at their convenience, reducing dependency on office hours and human staff availability.

The design also encourages more dynamic interaction between government entities and the public. By providing a direct, real-time communication channel, the chatbot helps bridge the gap between business owners and the services available to them, fostering trust and transparency. This has the potential to strengthen community ties and encourage more participation in state-led initiatives.

Moreover, the chatbot aligns with societal trends toward digital transformation and innovation. Its implementation showcases how AI can be used responsibly to enhance service delivery while maintaining ethical and secure practices. This demonstration of AI's potential could inspire similar advancements in other public sectors, contributing to societal progress.

Unintended societal consequences, such as over-reliance on AI or exclusion of those without digital access, were also considered in the design phase. To mitigate these risks, the chatbot was designed to complement existing support systems rather than replace them, ensuring that traditional avenues of support remain available. In summary, our chatbot promotes inclusivity, enhances government-public interactions, and contributes to societal advancement by leveraging AI to better serve the needs of diverse communities.

### H.3 Political/Regulatory Impacts

Our chatbot has the potential to serve as a prototype for future AI implementations within the Virginia government. By demonstrating how AI can enhance public service delivery while maintaining compliance with regulatory standards, this project sets a precedent for broader adoption of AI across state agencies.

From a political standpoint, the successful deployment of the chatbot could help build trust in AI technologies among policymakers and stakeholders. By showcasing the chatbot's ability to provide accurate, reliable, and secure responses, it addresses concerns about misinformation, bias, and data privacy, issues that often hinder AI adoption in government. This trust-building may encourage state officials to explore further integration of AI in other public-facing systems, fostering innovation in public administration.

On the regulatory side, our project adheres to existing state and federal guidelines on data security and accessibility, particularly in alignment with the Governor's Executive Order 30. By prioritizing data confidentiality and accessibility for all users, including those with disabilities, our chatbot sets a benchmark for regulatory compliance in AI-driven systems. This proactive approach could influence future regulatory frameworks, ensuring that AI technologies are implemented responsibly and equitably.

Additionally, the project underscores the importance of scalable and adaptable AI solutions within government systems. By demonstrating that AI can operate within strict regulatory constraints without sacrificing efficiency, the chatbot provides a compelling case for policymakers to modernize outdated processes and invest in similar technologies.

In summary, our chatbot not only fulfills immediate objectives but also serves as a foundational step toward the broader integration of AI in Virginia's government, with potential implications for regulatory standards and public policy at both the state and national levels.

### H.4. Economic Impacts

The implementation of our RAG-based chatbot for the Virginia Department of Small Business and Supplier Diversity (VDSBSD) carries significant economic implications, both immediate and long-term. By streamlining access to resources and guidance for small business owners, the chatbot supports the growth and sustainability of small businesses, which are vital contributors to Virginia’s economy. This enhanced support can foster entrepreneurship, job creation, and local economic development.

On an operational level, the chatbot reduces the department's reliance on human staff for routine inquiries, leading to cost savings. Resources previously allocated to handling basic queries can now be redirected to more strategic initiatives, maximizing the department's efficiency.

Additionally, the 24/7 availability of the chatbot minimizes downtime, ensuring users have uninterrupted access to critical information, which is especially beneficial for small business owners operating outside regular business hours. The technology also has the potential to shift markets by democratizing access to business resources. Historically underserved communities may benefit from improved access to funding opportunities, regulatory guidance, and procurement programs, which could reduce economic disparities across the state.

By empowering more small businesses, the chatbot may contribute to increased competition and innovation within local markets. Furthermore, the use of open-source and cost-effective technologies in the development process ensures the solution is economically sustainable for the department. By avoiding expensive proprietary tools and leveraging scalable AI frameworks, the design minimizes initial investment and ongoing maintenance costs.

### H.5 Environmental Impacts

### H.6 Global Impacts

### H.7. Ethical Considerations

Ethical concerns regarding the implementation of AI in government systems were central to our Capstone project. Government officials often express caution about the risks associated with emerging technologies, particularly those involving AI, such as biases, misinformation, and data privacy issues. To address these concerns, our design prioritized transparency, accuracy, and data security. We utilized a Retrieval-Augmented Generation (RAG) approach to significantly reduce hallucinations, a common issue where AI generates inaccurate or fabricated information.

By retrieving responses from a curated and secure vectorized database, we ensured that the chatbot's outputs were both reliable and contextually grounded. This approach not only enhanced the accuracy of the responses but also safeguarded sensitive information, maintaining the integrity and confidentiality of the data. Furthermore, our design explicitly avoids external data dependency, thereby minimizing exposure to vulnerabilities or unintended biases. These measures align with ethical AI principles, ensuring the technology serves its purpose without compromising public trust or safety. By addressing potential ethical challenges early in the design process, our project sets a precedent for the responsible integration of AI in government operations.

## Section I. Cost Analysis

Provide a simple cost analysis of the project that includes a list of all expenditures related to the project. If an experimental test set-up or prototype was developed, provide a Bill of Materials that includes part numbers, vendor names, unit costs, quantity, total costs, delivery times, dates received, etc. Do not forget to include all manufacturing costs incurred throughout the completion of the project. If the design is expected to become a commercial product, provide a production cost estimate including fixed capital, raw materials, manufacturing (including tooling and/or casting), and labor costs to produce and package the device. Note that this type of detailed cost analysis may be listed as a project deliverable.

**Note:** The Preliminary Design Report should include all costs incurred to date. It is expected that this section will be expanded and updated between the preliminary and final design reports.

Cost Analysis

The project utilized open-source and free tools to minimize development costs, focusing on affordability and sustainability.

Below is a breakdown of the project's expenditures and a projection of hosting costs for deployment.

1. Development Costs Software Frameworks and Tools:

Next.js: Open-source and free.

Flask: Open-source and free.

LangChain: Open-source and free.

Chroma DB: Open-source and free.

Model (Qwen by Open Llama): Open-source and free.

Labor Costs: Development, design, and testing were carried out by the project team as part of the capstone deliverable, incurring no additional labor costs outside academic commitments.

2. Infrastructure Costs (Projected) Hosting Services: We plan to use a cloud platform such as AWS or Azure.

The exact pricing depends on the final configuration, but preliminary estimates for hosting a chatbot include:

Compute Resources: Estimated $20-$50 per month for basic virtual machine or serverless setup.

Database Hosting: Approximately $15-$25 per month for managed database services like Azure Cosmos DB or AWS DynamoDB.

Storage Costs: $5-$10 per month for storing embeddings and other data.

Bandwidth Costs: Variable, depending on usage, estimated $10-$20 per month initially.

Total Estimated Hosting Costs: $50-$100 per month.

3. Miscellaneous Costs Testing Infrastructure: Conducted locally and through free-tier services provided by AWS and Azure.

No additional costs incurred. Development Tools: The team utilized freely available development tools and personal devices for coding and testing.

4. Bill of Materials (Prototyping Costs) This project did not involve hardware or physical prototypes; therefore, no Bill of Materials was required.

5. Future Production Cost Estimate (If Commercialized): If this design were to become a commercial product, the following production costs would apply:

Fixed Capital: Setting up scalable cloud hosting with appropriate redundancy: $500-$1,000 one-time setup fee.

Raw Materials (Cloud Resources): Compute, database, and storage services as listed above, scalable with user demand.

Labor Costs: Estimated $5,000-$10,000 for further refinement, marketing, and customer support setup.

Manufacturing: Not applicable, as this is a software-based product. Packaging and Distribution: Not applicable, as the product will be delivered digitally via the VDSBSD website.

Summary of Costs: Development Costs: $0 (open-source tools used).

Hosting (Projected): $50-$100 per month. Future Commercial Costs: Approximately $5,500-$11,000 upfront, with ongoing hosting and scaling costs depending on user demand.

This cost-effective approach ensures the project remains within budget while demonstrating scalability and sustainability for future implementation.

## Section J. Conclusions and Recommendations

### Summary:

The SBSD chatbot's design evolved through refinements to resolve response inaccuracies, transitioning from simpler alternatives to the sophisticated RAG model. Each pase of the design process involved evaluating existing systems, brainstorming solution and addressing challenges such as compliance to EO30 and integrating with SBSD's IT infrastructure. This approach ensured that the final product meet the needs of SBSD and its users.

### Design:

The chatbot delivers accurate response by leveraging RAG and it also features user-friendly web interfaces. It also supports multilingual access to help other users. It supports scalability to handle future growth, ensures strict regulatory compliance to protect user data, and employs a modular design to streamline updates and maintenance. These features collectively establish the chatbot as a robust and efficient tool for improving public service delivery.

**Achievement:**

This experiment successfully lowered hallucination rates in chatbot responses, resulting in improved accuracy and trustworthiness. It produced a scalable prototype that serves as a model for government AI applications, improved small company resource access, and created a rigorous testing framework to certify the system's performance, security, and adherence to ethical AI standards.

**Recommendations for Future Work:**

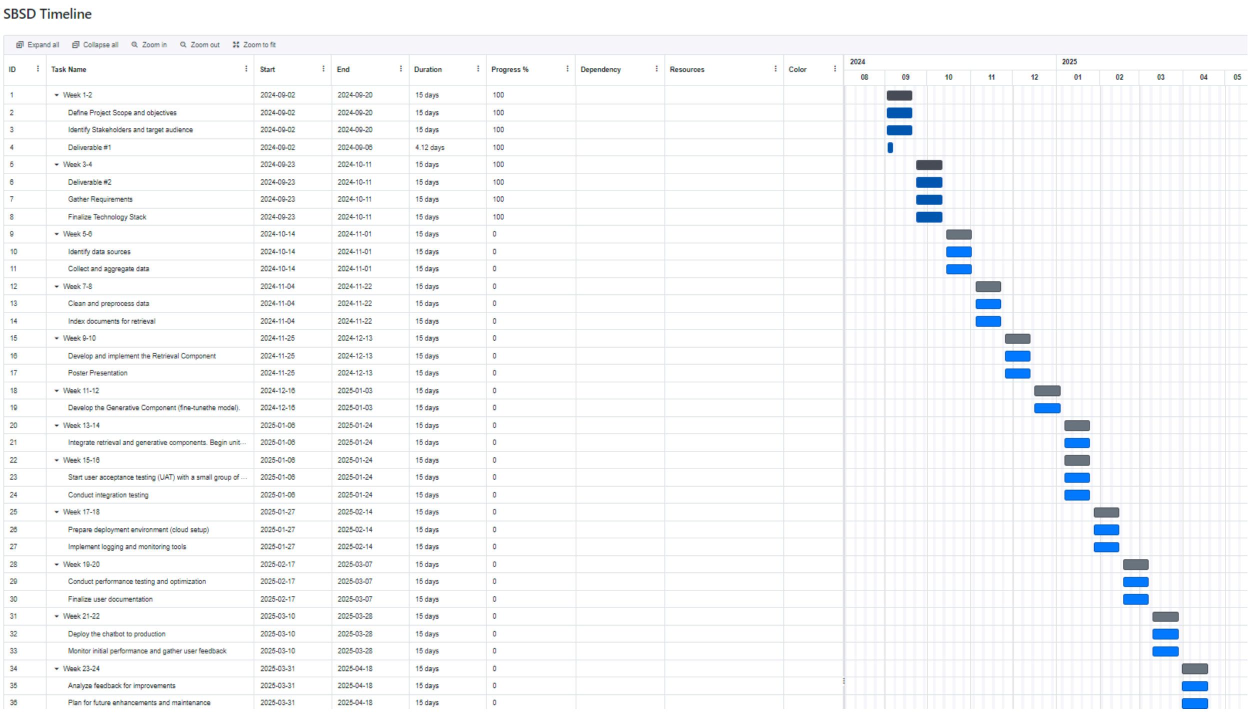
To remain relevant, future development should prioritize regular updates to the training data with fresh SBSD resources. Implementing a feedback mechanism will enable users to suggest improvements and report problems. Expanding language support will improve accessibility, whereas voice interaction will improve usability for those with disabilities. Advanced analytics could provide more detailed insights into user interactions and system performance, allowing for future refinements.

**Closing Statement**

This research demonstrates AI's potential to transform public services by addressing SBSD-specific operational difficulties while complying to ethical and regulatory requirements. Using the RAG model, the chatbot demonstrates how technology may improve government-citizen interactions, streamline resource access, and create the groundwork for future innovations. The team's work establishes a standard for responsible AI use, promoting societal growth and increasing public sector efficiency.

# Appendix 1: Project Timeline

Here is our anticipated timeline for the SBSD Capstone project. This link will take you to a website to view the timeline in greater detail but we attached the image below for easier viewing.



SBSD Timeline

| ID | Task Name | Start | End | Duration | Progress % | Dependency | Resources | Color |
|---|---|---|---|---|---|---|---|---|
| 1 | Week 1-2 | 2024-09-02 | 2024-09-20 | 15 days | 100 | | | |
| 2 | Define Project Scope and objectives | 2024-09-02 | 2024-09-20 | 15 days | 100 | | | |
| 3 | Identify Stakeholders and target audience | 2024-09-02 | 2024-09-20 | 15 days | 100 | | | |
| 4 | Deliverable #1 | 2024-09-02 | 2024-09-06 | 4.12 days | 100 | | | |
| 5 | Week 3-4 | 2024-09-23 | 2024-10-11 | 15 days | 100 | | | |
| 6 | Deliverable #2 | 2024-09-23 | 2024-10-11 | 15 days | 100 | | | |
| 7 | Gather Requirements | 2024-09-23 | 2024-10-11 | 15 days | 100 | | | |
| 8 | Finalize Technology Stack | 2024-09-23 | 2024-10-11 | 15 days | 100 | | | |
| 9 | Week 5-6 | 2024-10-14 | 2024-11-01 | 15 days | 0 | | | |
| 10 | Identify data sources | 2024-10-14 | 2024-11-01 | 15 days | 0 | | | |
| 11 | Collect and aggregate data | 2024-10-14 | 2024-11-01 | 15 days | 0 | | | |
| 12 | Week 7-8 | 2024-11-04 | 2024-11-22 | 15 days | 0 | | | |
| 13 | Clean and preprocess data | 2024-11-04 | 2024-11-22 | 15 days | 0 | | | |
| 14 | Index documents for retrieval | 2024-11-04 | 2024-11-22 | 15 days | 0 | | | |
| 15 | Week 9-10 | 2024-11-25 | 2024-12-13 | 15 days | 0 | | | |
| 16 | Develop and implement the Retrieval Component | 2024-11-25 | 2024-12-13 | 15 days | 0 | | | |
| 17 | Poster Presentation | 2024-11-25 | 2024-12-13 | 15 days | 0 | | | |
| 18 | Week 11-12 | 2024-12-16 | 2025-01-03 | 15 days | 0 | | | |
| 19 | Develop the Generative Component (fine-tunethe model). | 2024-12-16 | 2025-01-03 | 15 days | 0 | | | |
| 20 | Week 13-14 | 2025-01-06 | 2025-01-24 | 15 days | 0 | | | |
| 21 | Integrate retrieval and generative components. Begin unit... | 2025-01-06 | 2025-01-24 | 15 days | 0 | | | |
| 22 | Week 15-16 | 2025-01-06 | 2025-01-24 | 15 days | 0 | | | |
| 23 | Start user acceptance testing (UAT) with a small group of ... | 2025-01-06 | 2025-01-24 | 15 days | 0 | | | |
| 24 | Conduct integration testing | 2025-01-06 | 2025-01-24 | 15 days | 0 | | | |
| 25 | Week 17-18 | 2025-01-27 | 2025-02-14 | 15 days | 0 | | | |
| 26 | Prepare deployment environment (cloud setup) | 2025-01-27 | 2025-02-14 | 15 days | 0 | | | |
| 27 | Implement logging and monitoring tools | 2025-01-27 | 2025-02-14 | 15 days | 0 | | | |
| 28 | Week 19-20 | 2025-02-17 | 2025-03-07 | 15 days | 0 | | | |
| 29 | Conduct performance testing and optimization | 2025-02-17 | 2025-03-07 | 15 days | 0 | | | |
| 30 | Finalize user documentation | 2025-02-17 | 2025-03-07 | 15 days | 0 | | | |
| 31 | Week 21-22 | 2025-03-10 | 2025-03-28 | 15 days | 0 | | | |
| 32 | Deploy the chatbot to production | 2025-03-10 | 2025-03-28 | 15 days | 0 | | | |
| 33 | Monitor initial performance and gather user feedback | 2025-03-10 | 2025-03-28 | 15 days | 0 | | | |
| 34 | Week 23-24 | 2025-03-31 | 2025-04-18 | 15 days | 0 | | | |
| 35 | Analyze feedback for improvements | 2025-03-31 | 2025-04-18 | 15 days | 0 | | | |
| 36 | Plan for future enhancements and maintenance | 2025-03-31 | 2025-04-18 | 15 days | 0 | | | |

## Appendix 2: Team Contract (i.e. Team Organization)

### Step 1: Get to Know One Another. Gather Basic Information.

**Task:** This initial time together is important to form a strong team dynamic and get to know each other more as people outside of class time. Consider ways to develop positive working relationships with others, while remaining open and personal. Learn each other's strengths and discuss good/bad team experiences. This is also a good opportunity to start to better understand each other's communication and working styles.

| ***Team Member Name*** | ***Strengths each member bring to the group*** | ***Other Info*** | ***Contact Info*** |
|---|---|---|---|
| *Zach Dellimore* | *Web development experience, AI development experience, Industry experience* | *I enjoy coding and make lots of personal projects for fun. AI is something I've been wanting more experience with.* | *dellimorez@vcu.edu*<br>*540-394-8593* |
| Victor Olivar | I have experience in AI making machine learning about stocks and data analysis. | I enjoy coding to create projects that will help me with my everyday necessities. This project is really good for me because I've been looking forward to experiencing more work related to AI. | olivarvf@vcu.edu<br>804-647-5826 |
| *Jacobo Ceballos* | I keep people on track and will make sure we are staying on task, not leaving things to the last minute. Discipline, web development experience, I have experience building and deploying AI models. Full stack programming experience | I'm looking forward to working on this project. Over the summer I worked on both web development and AI projects so I hope to bring some of that experience into the table. | ceballosj@vcu.edu<br>804-418-1199 |
| Nate Swetlow | Good with many different coding languages as well as a background in data analytics. | I like to work on coding projects related to finance and data. I have experience with Tableau too. | Swetlownt@vcu.edu<br>703-582-2759 |

|  |  |  |  |
|---|---|---|---|

| ***Other Stakeholders*** | ***Notes*** | ***Contact Info*** |
|---|---|---|
| *John Leonard* | *Met with Prof Leonard on 9/5 did a brief overview of the project and the path we are going down.* | *jdleonard@vcu.edu* |
| *Willis Morris* | *Sent out an email to set up a meeting with Mr. Morris and go over the project.* | Willis.Morris@sbsd.virginia.gov |

## Step 2: Team Culture. Clarify the Group’s Purpose and Culture Goals.

**Task:** Discuss how each team member wants to be treated to encourage them to make valuable contributions to the group and how each team member would like to feel recognized for their efforts. Discuss how the team will foster an environment where each team member feels they are accountable for their actions and the way they contribute to the project. These are your Culture Goals (left column). How do the students demonstrate these culture goals? These are your Actions (middle column). Finally, how do students deviate from the team’s culture goals? What are ways that other team members can notice when that culture goal is no longer being honored in team dynamics? These are your Warning Signs (right column).

**Resources:** More information and an example Team Culture can be found in the Biodesign Student Guide “Intentional Teamwork” page (webpage | PDF)

| ***Culture Goals*** | ***Actions*** | ***Warning Signs*** |
|---|---|---|
| *Teamwork* | - *Help each other with work*<br>- *provide assistance to work you may know*<br>- *Step up for each other.* | - *Student need to contribute or else a warning*<br>- *If student fail to contribute at least once a week, it’ll be a warning.* |
| *Open Communication* | - *Keep team informed on your tasks status*<br>- *Ask for help if you need it* | - *Student shows up for weekly meeting with no considerable work done*<br>- Team members feel out of the loop consistently on the status of other team members task |
| NO PROCRASTINATION | - stay on top of all due lates<br>- absolutely no late turn ins<br>- work ahead of schedule and not leave things for the last minute | - if we miss a due late, there will be a team meeting to discuss time management<br>- if we somehow miss two due dates, meeting with advisor to seek guidance |

## Step 3: Time Commitments, Meeting Structure, and Communication

**Task:** Discuss the anticipated time commitments for the group project. Consider the following questions (don't answer these questions in the box below):

- What are reasonable time commitments for everyone to invest in this project?
- What other activities and commitments do group members have in their lives?
- How will we communicate with each other?
- When will we meet as a team? Where will we meet? How Often?
- Who will run the meetings? Will there be an assigned team leader or scribe? Does that position rotate or will same person take on that role for the duration of the project?

**Required:** How often you will meet with your faculty advisor, where you will meet, and how the meetings will be conducted. Who arranges these meetings?
See examples below.

| ***Meeting Participants*** | ***Frequency***<br>***Dates and Times / Locations*** | ***Meeting Goals***<br>***Responsible Party*** |
|---|---|---|
| *Students Only* | *As Needed, On Discord Voice Channel* | *Update group on day-to-day challenges and accomplishments (Zach will summarize these for the weekly progress reports and meetings with advisor)* |
| *Students Only* | *Every Thursday, in library or discord if no in person spaces available* | *Actively work on the project and update the team on what we accomplished that week. (Zach will document these meetings by taking photos of whiteboards, documents, etc, then post on the Discord channel and update Capstone Report)* |
| *Students + Faculty advisor* | *Once or twice a month on Thursdays at, or around, 6:00pm via Discord* | *Update faculty advisor and get answers to our questions (Zach will scribe; TODO will create meeting agenda and lead meeting)* |
| *Project Sponsor* | *Once or twice a month, via Zoom/Google Meet. If we need to update meeting times we will update the sponsor via Email* | *Update project sponsor and make sure we are on the right track (Zach will scribe; TODO will create meeting agenda and lead meeting; Team will present project developments)* |

## Step 4: Determine Individual Roles and Responsibilities

**Task:** As part of the Capstone Team experience, each member will take on a leadership role, ***in addition to*** contributing to the overall weekly action items for the project. Some common leadership roles for Capstone projects are listed below. Other roles may be assigned with approval of your faculty advisor as deemed fit for the project. For the entirety of the project, you should communicate progress to your advisor specifically with regard to your role.

- **Before meeting with your team**, take some time to ask yourself: what is my "natural" role in this group (strengths)? How can I use this experience to help me grow and develop more?
- **As a group,** discuss the various tasks needed for the project and role preferences. Then assign roles in the table on the next page. Try to create a team dynamic that is fair and equitable, while promoting the strengths of each member.

### Communication Leaders

**TODO:** Assign a team member to be the primary contact for the client/sponsor. This person will schedule meetings, send updates, and ensure deliverables are met.

**TODO:** Assign a team member to be the primary contact for faculty advisor. This person will schedule meetings, send updates, and ensure deliverables are met.

### Common Leadership Roles for Capstone

1. **Project Manager:** Manages all tasks; develops overall schedule for project; writes agendas and runs meetings; reviews and monitors individual action items; creates an environment where team members are respected, take risks and feel safe expressing their ideas.
   **Required:** On Edusourced, under the Team tab, make sure that this student is assigned the Project Manager role. This is required so that Capstone program staff can easily identify a single contact person, especially for items like Purchasing and Receiving project supplies.
2. **Logistics Manager:** coordinates all internal and external interactions; lead in establishing contact within and outside of organization, following up on communication of commitments, obtaining information for the team; documents meeting minutes; manages facility and resource usage.
3. **Financial Manager:** researches/benchmarks technical purchases and acquisitions; conducts pricing analysis and budget justifications on proposed purchases; carries out team purchase requests; monitors team budget.
4. **Systems Engineer:** analyzes Client initial design specification and leads establishment of product specifications; monitors, coordinates and manages integration of sub-systems in the prototype; develops and recommends system architecture and manages product interfaces.
5. **Test Engineer:** oversees experimental design, test plan, procedures and data analysis; acquires data acquisition equipment and any necessary software; establishes test protocols and schedules; oversees statistical analysis of results; leads presentation of experimental finding and resulting recommendations.

6. **Manufacturing Engineer:** coordinates all fabrication required to meet final prototype requirements; oversees that all engineering drawings meet the requirements of machine shop or vendor; reviews designs to ensure design for manufacturing; determines realistic timing for fabrication and quality; develops schedule for all manufacturing.

| ***Team Member*** | ***Role(s)*** | ***Responsibilities*** |
|---|---|---|
| Nate | Researcher<br>Design | ● Coordinates all tasks and ensures deadlines are met.<br>● Breaks down the project into manageable tasks.<br>● Analyzes information and presents findings to the group. |
| Victor | Test Engineer | ● Main responsibility is to make sure the final project is stable, reliable, and of high quality.<br>● Test planning to analyze what the project's requirement for testing<br>● Test execution to find bugs and issues and report it to the group and also to ensure that the changes don't affect the previous working functions. |
| Zach | Systems Engineer | ● Takes notes on client design specifications and will create presentations/notes on our product specification.<br>● Plan out system architecture and how each system will interact with each other in compliance with the EO 30. |
| Jacob | Project Manager | ● Keeping the group on task, coordinated and up to date with all the information from advisor and sponsor<br>● Making sure that all the deliverables are on time<br>● Ensuring that everyone is doing their fair share<br>● time management, financial budget, planning and defining scope |

## Step 5: Agree to the above team contract

*Team Member: Zach Dellimore* *Signature: Zachariah Dellimore*

*Team Member: Jacobo Ceballos* *Signature: Jacobo Ceballos*

*Team Member: Nate Swetlow* *Signature: Nathan Swetlow*

*Team Member: Victor Olivar* *Signature: Victor Olivar*

## References

[1] VITA. *Commonwealth of Virginia Enterprise Architecture Standard (EA-225) Enterprise Solutions Architecture [ESA] Web Systems*. Commonwealth of Virginia, 28 May 2024, https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiv8qfv8YSJAxXK6skDHU_MKAMQFnoECBMQAQ&url=https%3A%2F%2Fwww.vita.virginia.gov%2Fmedia%2Fvitavirginiagov%2Fit-governance%2Fea%2Fdocs%2FEA-Solutions-Web-Systems-Standard.

[2] VITA. (n.d.). Commonwealth of Virginia Enterprise Architecture Standard (EA-225) Enterprise Solutions Architecture [ESA] Artificial Intelligence. Commonwealth of Virginia. Retrieved from https://www.vita.virginia.gov/media/vitavirginiagov/it-governance/ea/pdf/EA-Solutions-Artificial-Intelligence-Standard.pdf

[3] ITRM. (N.d.). Retrieved October 10, 2024. Commonwealth of Virginia Retrieved from https://www.vita.virginia.gov/media/vitavirginiagov/it-governance/ea/pdf/EA_StandardEA225-15-2.pdf