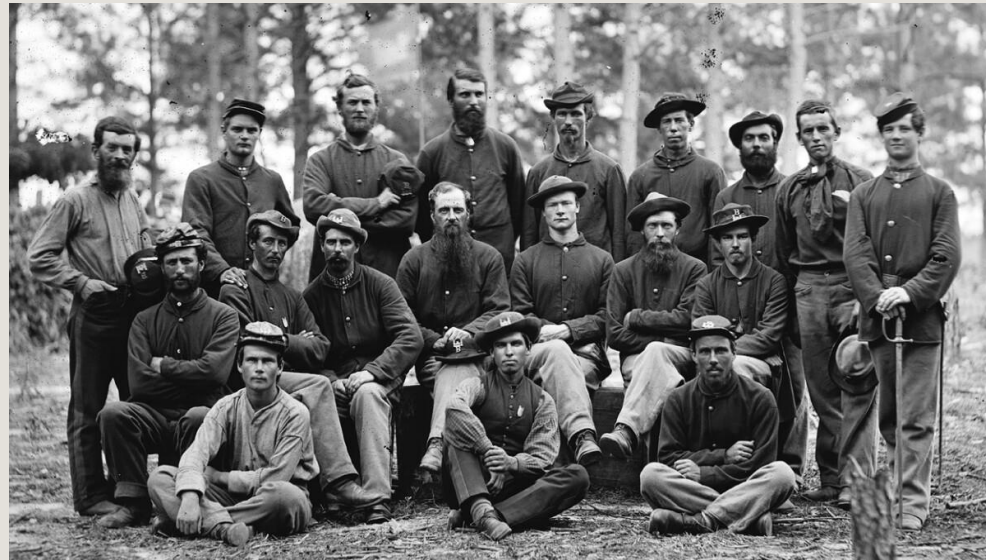


# CIVIL WAR PHOTO SLEUTH

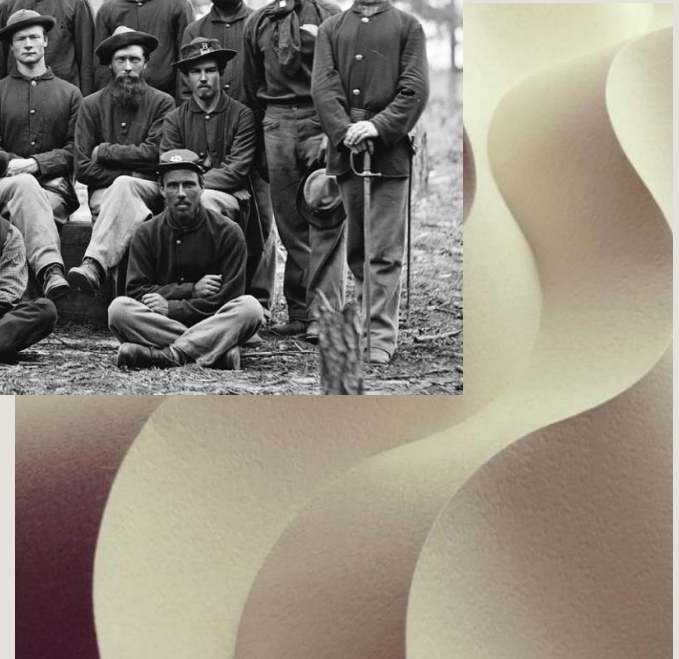
## *CHATBOT*



---

### *Group 2*

Britney Aiken, Kuldeep Dixit, Thaddeus Nongni, Kirubanidhi Ramachandran, Shivangi Sarkar, John Swecker



# Project overview

- **Project description**
  - This project focuses on improving the CWPS website, a platform dedicated to aiding historical research.
  - The primary task of this project is to develop a conversational, interactive chatbot that is able to assist users of the Civil War Photo Sleuth website.
- **Context + background**
  - Historians and researchers face difficulties accessing and analyzing fragmented Civil War archives, slowing the research process.
  - Tools like Civil War Photo Sleuth (CWPS) offer valuable resources, but the absence of a chatbot for creating effective queries and querying multiple sources further impedes efficiency.

# Project goals

## Goal

Develop a conversational and interactive CWPS chatbot

- *Answer queries about specific Civil War soldiers*
- *Respond to broader queries about specific military units*

## Main functionalities

Leverage an LLM to converse in natural human language with the user

Formulate queries based on CWPS database records in tandem with other data sources

Ensure user queries are relevant

Who was Henry Mack?

Who served in the 18th NH Infantry?

Tell me about Oliver Johnson

Give me a list of soldiers in the Tennessee infantry, 5<sup>th</sup> regiment

# Target users



Historians and researchers

Genealogists

Civil War enthusiasts

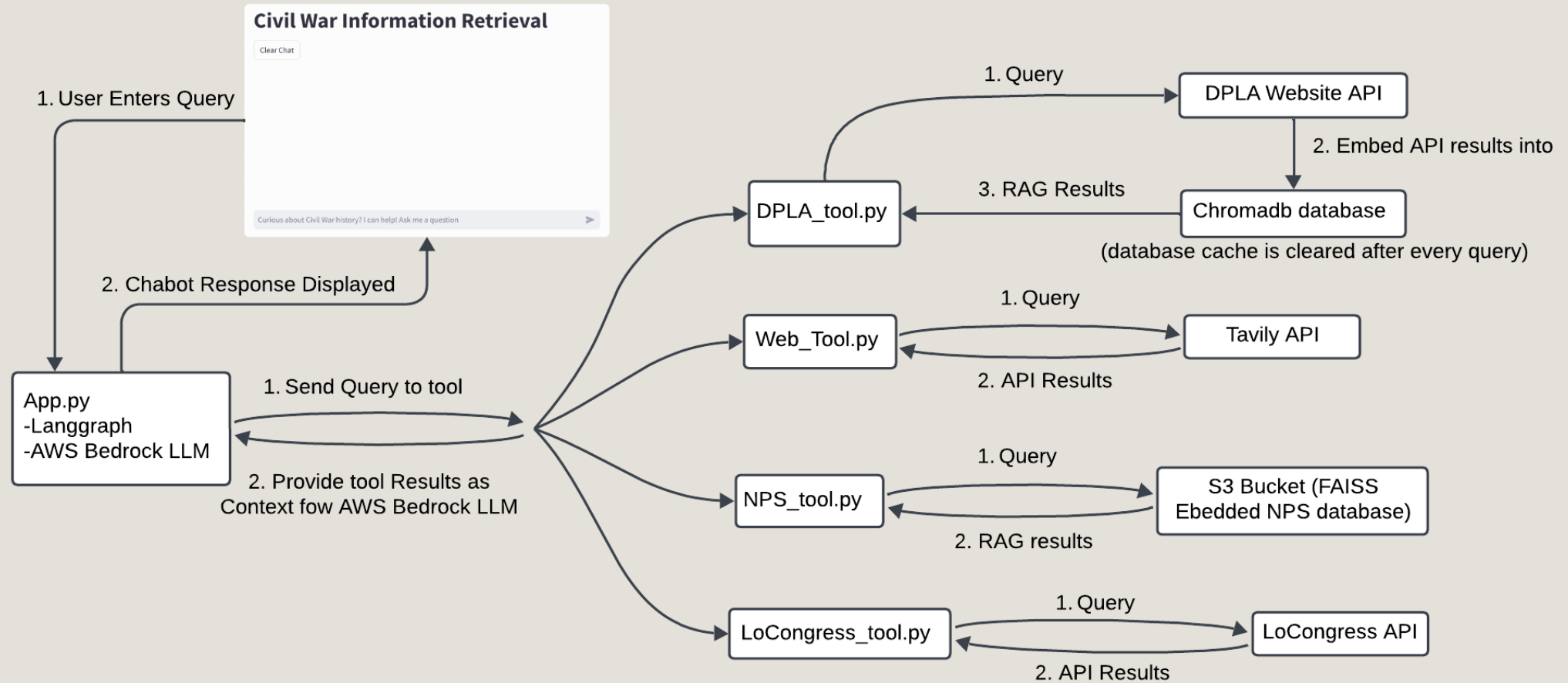
Government departments

Students and instructors

# Tools and IDEs

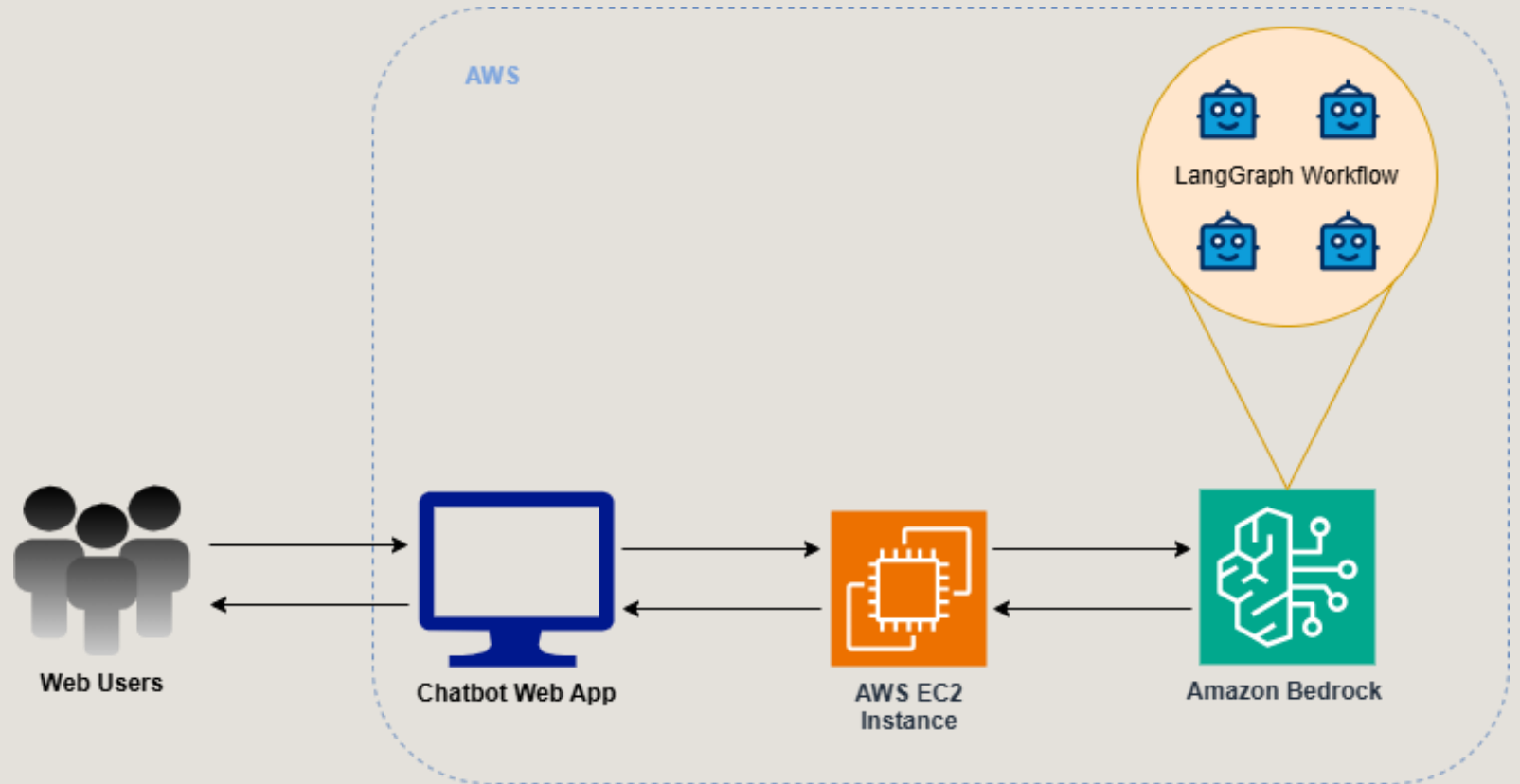
- Jira
  - Project Management and sprint planning
- GitLab
  - Version control system
- VS-Code
  - Integrated Development Environment
- Huggingface LLM's
  - LLM's that were installed and run on our local machines before access AWS bedrock

# Chatbot Diagram



# High-Level Architecture

---



# Cloud Automation & Deployment

- **Cloud Design Considerations**
  - Scalability: Make deployment scalable to accommodate greater workloads and improve performance.
  - Microservice architecture: Each Data source eg. Websearch, nps for example are deployed independent of the other.
  - Cloud Security: All internal services such as Faiss, Chromadb and Bedrock were all shielded from internet access.
- **Deployment Tools & components**
  - Terraform ( Cloud Agnostic infrastructure as code tool) is used deploy VPC and all cloud resources.
  - Bash scripts are used to deploy Docker containers.
  - Both tools above are used to fully automated the deployment of CWPS into the Cloud



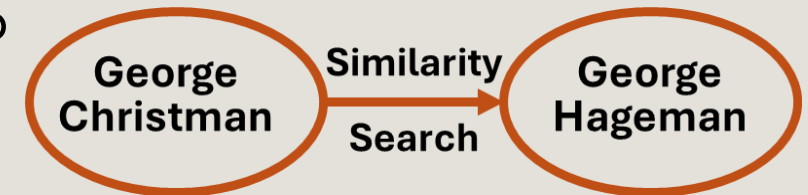
# Cloud Services & Microservices

- AWS Bedrock Service ( LLM ).
- An EC2 instance optimized for compute-intensive tasks.
- Three Chromadb & FAISS database Docker Containers running within the EC2 instance.
- A Docker Container running admin.py dedicated to uploading and processing documents/files.
- A Docker Container running app.py.
- An IAM role that allows access to all cloud resources for this project.

# Demo

# Lessons Learned & Reflections

- Limited ability of LLM to distinguish entities
  - o Example: The last name "Black" was not identified as name and kicked off-topic branch or, searched just for first name (in case of full name).
- Similarity search missing names in database
  - o A randomized 100 name search yielded only 46 exact matches, and the remaining 54 names were either soft matches (either first or, last name matched) or no match (similar sounding names).



# Continued Work

- Possible future direction for this project is there is room for improvement:
  - Memory implementation for follow-up questions
  - CWPS website integration with the chatbot
  - Image display integration in the UI
  - Adding search by picture feature to the chatbot
  - Improving chatbot accuracy by placing reinforcement steps in the LangGraph