

# COS80029 – Technology Application Project

Final Reflection

Name: Kritika.

## Table of Contents

Introduction .....	2
Background of Project .....	2
Problem Statement.....	2
Project Origin .....	2
Research and Analysis.....	2
Technological Context.....	3
Project Goals and Scopes .....	3
Goals .....	3
In-Scope and Out-of-Scope.....	3
Deliverables .....	4
Timeline .....	5
Self-Assessment .....	6
Roles and Responsibilities .....	6
Achievements and Significance .....	7
List of Contributions .....	8
Contributions .....	8
Charts and Visual Representations.....	15
Learning Outcome Assessment .....	26
ULO 1: .....	26
ULO 2: .....	27
ULO 3: .....	27
ULO 4: .....	27
ULO 5: .....	28
ULO 6: .....	28
Reflection .....	29
The most important things I learnt:.....	29
The things that helped me most were: .....	29
I found the following topics particularly challenging: .....	29
I found the following topics particularly interesting: .....	29
I feel I learnt these topics, concepts, and/or tools really well: .....	30
I still need to work on the following areas: .....	30
My progress in this unit was ...: .....	30
This unit will help me in the future: .....	30
If I did this unit again, I would do the following things differently: .....	30
Other...: .....	31
Conclusion .....	31

## Introduction

This report provides an in-depth account of my individual achievements in the COS80029 - Technology Application Project unit. It is structured to align with my learning objectives and includes a self-assessment, a summarized list of key supporting evidence, an analysis of my activities, a comprehensive demonstration of how the evidence corresponds to the learning outcomes, and a personal reflection on the overall experience and its results.

## Background of Project

### Problem Statement

Decision-makers in the fields of political analysis and campaign strategy are constantly confronted with large and intricate demographic databases. Voter demographics, census data, election statistics, and constituency details are just a few of the many types of information that are included in these datasets. Traditionally, it takes a lot of time, specialist knowledge, and financial resources to extract valuable insights from such vast amounts of data. This time-consuming procedure frequently prevents prompt decision-making and limits the availability of vital data for political analysts, campaign strategists, and community organizations. PoliQ tackles this issue by offering a productive, approachable platform that simplifies data analysis and allows users to quickly get precise insights without requiring in-depth understanding of data science.

### Project Origin

The PoliQ solution was created in response to the political and civic sectors' increasing demand for more user-friendly and effective data analysis tools. Group CH2 aimed to create a solution that uses cutting-edge technology to streamline and speed up the data interpretation process after realizing the shortcomings of conventional data analysis techniques. The project's path was further influenced by the partnership with PoliQ, an organization that provides analytical tools to advocacy groups and legislators. PoliMap, PoliQ's pre-existing platform, offered a strong basis for improving data analysis and accessibility. A significant move to enhance PoliMap's capabilities was the incorporation of a chatbot interface, which made data-driven insights more accessible to a wider user base.

### Research and Analysis

Before developing the PoliQ solution, thorough research and analysis were undertaken to identify the unique requirements of political analysts and campaign strategists. This process involved evaluating existing data analysis tools, pinpointing their limitations, and investigating emerging technologies capable of filling these gaps. The team explored different techniques in natural language processing and data retrieval to determine the most effective methods for the project. Additionally, they evaluated the integration of external knowledge sources to improve the precision and relevance of the generated summaries. A comparative analysis of local versus cloud-based solutions was also conducted to assess factors such as performance, scalability, and cost-effectiveness, ensuring that the chosen technology would align with the project's goals.

## Technological Context

The PoliQ solution is founded on a strong technological framework that incorporates advanced tools and methodologies. The chatbot interface, built with React, ensures a responsive and user-friendly frontend experience. FastAPI serves as the backend's RESTful API framework, facilitating smooth communication between the frontend and backend components. To handle both unstructured and structured data efficiently, the solution uses MongoDB and PostgreSQL, respectively, providing flexibility in data management. This integration of technologies enhances the performance and scalability of the PoliQ solution while ensuring its seamless compatibility within PoliQ's larger ecosystem.

## Project Goals and Scopes

### Goals

The PoliQ solution project is focused on several key objectives to improve data accessibility and enhance analytical capabilities for political analysts, campaign strategists, and civic organizations. The main goals are as follows:

- **Develop an Intuitive Chatbot Interface:** Create a user-friendly chatbot that enables users to interact with large datasets through natural language queries. The aim is to make data retrieval and analysis accessible to non-experts without requiring specialized data science knowledge.
- **Enhance PoliMap's Analytical Features:** Strengthen the existing PoliMap platform with advanced tools for more efficient data-driven decision-making, including demographic analysis, electoral data evaluation, and electorate comparisons.
- **Ensure Effective Data Management:** Leverage MongoDB and PostgreSQL databases to manage unstructured and structured data efficiently, ensuring data integrity, scalability, and accessibility for diverse user needs and large-scale data operations.
- **Optimize Performance and Scalability:** Architect the system to be highly scalable and performant, capable of handling increasing data volumes and user demands while maintaining speed and reliability.
- **Provide Comprehensive Documentation and Support:** Develop extensive user manuals, installation guides, API documentation, and technical manuals to ensure smooth deployment, usage, and maintenance of the PoliQ solution, equipping stakeholders with the necessary tools for effective system management.

### In-Scope and Out-of-Scope

The scope of the PoliMap LLM solution project outlines the boundaries for project activities, clarifying what will be addressed and what is outside the project's scope. The following table distinguishes between in-scope and out-of-scope elements:

Category	In-Scope	Out-of-Scope
<b>Functionalities</b>	<ul style="list-style-type: none"> <li>- Development of a natural language chatbot interface</li> <li>- Implementation of Retrieval-Augmented Generation (RAG) techniques</li> <li>- Integration with PoliMap platform</li> </ul>	<ul style="list-style-type: none"> <li>- Developing additional data visualization tools</li> <li>- Establishing source databases for electoral and census information</li> <li>- Implementing a frontend for the report generation workflow</li> <li>- Customizations that extend beyond standard user queries</li> </ul>
<b>Technologies</b>	<ul style="list-style-type: none"> <li>- Use of React for frontend development</li> <li>- FastAPI for backend services</li> <li>- MongoDB and PostgreSQL for managing structured and unstructured data, along with PGVector for data management.</li> </ul>	<ul style="list-style-type: none"> <li>-Utilize cloud-based database services for data management (e.g., AWS, Azure).</li> </ul>
<b>User Management</b>	<ul style="list-style-type: none"> <li>- Handling user chats, including chat sessions, conversation history, and user-uploaded documents.</li> </ul>	<ul style="list-style-type: none"> <li>- Advanced user profiling and behaviour analytics</li> <li>- User authentication and authorization functionalities</li> </ul>
<b>Data Handling</b>	<ul style="list-style-type: none"> <li>- Handling unstructured chat data</li> <li>- Accessing census and electoral data stored in the client's relational database</li> <li>-Developing a vector database to support the RAG solution</li> </ul>	<ul style="list-style-type: none"> <li>-Developing foundational data sources for electoral and census information</li> </ul>
<b>Documentation</b>	<ul style="list-style-type: none"> <li>- Creation of comprehensive user manuals and installation guides</li> <li>- Development of API and technical documentation</li> </ul>	<ul style="list-style-type: none"> <li>- Marketing materials and user training programs</li> </ul>
<b>Deployment</b>	<ul style="list-style-type: none"> <li>- Cloud hosting setup for scalability and availability</li> <li>- Integration with existing PoliMap infrastructure</li> </ul>	<ul style="list-style-type: none"> <li>- On-premises deployment for non-cloud users</li> </ul>
<b>Performance</b>	<ul style="list-style-type: none"> <li>- Ensuring scalability and reliability of the system.</li> <li>- Optimizing AI agent's performance and ensuring its consistency.</li> </ul>	<ul style="list-style-type: none"> <li>- Guaranteeing zero downtime or absolute system security</li> </ul>

## Deliverables

The PoliMap LLM solution project will result in several critical deliverables, covering both development and documentation aspects. The table below outlines the main deliverables expected by project completion:

Deliverable	Description
<b>Chatbot Interface</b>	A fully functional chatbot integrated with the PoliMap platform, allowing natural language queries and data summarization.
<b>RAG-Enhanced LLM</b>	Implementation of Retrieval-Augmented Generation (RAG) techniques to enhance the relevance and accuracy of chatbot responses.
<b>Report Generation Workflow</b>	Implementing an AI-driven workflow to dynamically generate reports on the PoliMap frontend in response to user's natural language queries.
<b>System Architecture Document</b>	Comprehensive documentation detailing the modular system design, including the frontend, backend, databases, and AI components.
<b>API Documentation</b>	Detailed guides and references for the RESTful APIs developed using FastAPI, ensuring smooth integration and use.
<b>User Manuals</b>	Step-by-step instructions for end-users on interacting with the chatbot, managing data, and utilizing the system's analytical features.
<b>Installation Guides</b>	Technical manuals providing instructions on installation and deployment processes for the PoliMap LLM solution in a cloud environment.
<b>Technical Manuals</b>	In-depth documentation aimed at developers and system administrators, covering system setup, maintenance, and troubleshooting procedures.
<b>Prototype Demonstrations</b>	Interactive prototypes showcasing key features such as chatbot interactions, data summarization, and visualizations.
<b>Final Project Report</b>	A detailed report summarizing the project's goals, development process, results, and recommendations for future improvements.

## Timeline

The project timeline outlines the key milestones and deadlines from initiation to deployment, ensuring that all activities are well-structured and executed efficiently. Below is a week-by-week breakdown of major tasks and accomplishments:

Week	Milestones and Activities
<b>1-2</b>	<ul style="list-style-type: none"> <li>-Team formation and setup of communication tools (Slack, Trello, Notion).</li> <li>-Initial client communication and requirements gathering.</li> <li>-Building first prototype and gathering feedback from client.</li> </ul>
<b>3-4</b>	<ul style="list-style-type: none"> <li>-Completed the initial draft of the Software Requirements Specification (SRS) and the Project Plan</li> <li>-Initiated frontend development, which continued concurrently with other project phases</li> </ul>

5-6	-Implemented the FastAPI backend -Developed backend features for managing chat sessions
7-8	-Integration of FastAPI with backend services. -Development of middleware to link the frontend and backend.
9-10	- Application of RAG techniques and integration with Large Language Models (LLMs) - Implementation of a vector database. - Developing report generation workflow
11-12	-Testing and validation of system functionalities. -Bug fixes and performance improvements. -Integration with the client's PoliMap system.

## Self-Assessment

### Roles and Responsibilities

- **Team Communication and Collaboration:**
  - **Expectation:** I was expected to actively participate in team discussions, stay updated on task progress, and maintain communication with the client and team members.
  - **Responsibility:** I regularly corresponded with the team through Slack and Trello, contributing to task management, coordination, and updates. I also actively committed to GitHub and maintained proper version control for code changes. Additionally, I made notes during our initial client meetings, ensuring that the key takeaways and expectations were documented for the team.
- **UI Development and Features:**
  - **Expectation:** I was expected to enhance the user interface by implementing critical features that improve chat interactions and navigation within the application.
  - **Responsibility:** I developed functionalities like hiding the sidebar, deleting all chats, pinning/unpinning chats, and managing a three-dot menu for options such as delete, archive, and rename. Additionally, I configured the menu to show only on hover for inactive chats while keeping it visible for the active chat session, distinguished by a different colour in the sidebar.
- **Message and Chat Hooks:**
  - **Expectation:** I was responsible for ensuring that hooks functioned efficiently and provided essential chat-related interactions along with message regeneration to enhance the user experience. Moreover, I was expected to debug the features for smooth integration.
  - **Responsibility:** I contributed to developing hooks that allow users to copy messages seamlessly, update chat titles and regenerate messages as required. I collaborated with the backend team to handle API calls and middleware processes.
- **Metadata Preparation and Research:**
  - **Expectation:** My task was to facilitate the team's understanding of the client's repository and to enhance the documentation of crucial components, making the codebase more navigable and clearer for development.

- **Responsibility:** I researched and prepared a detailed document that explained the metadata and key files within the client's repository. This included providing clarity on the structure, flow, and functionality of the codebase. Additionally, I contributed to researching report generation within the client's application, ensuring that the team could effectively integrate and document this feature.
- **Sidebar Enhancement:**
  - **Expectation:** I was tasked with improving the user interface by enhancing the sidebar for better chat organization.
  - **Responsibility:** I designed and implemented a dropdown feature for grouping chats by date, making it easier for users to navigate their chat history and locate previous conversations.
- **Debugging and Testing:**
  - **Expectation:** I was expected to identify and resolve UI issues, ensuring the stability and responsiveness of the application.
  - **Responsibility:** I conducted extensive debugging and testing to refine the codebase and enhance user interactions.
- **Documentation and Report Writing:**
  - **Expectation:** I was expected to contribute to the project's documentation, ensuring that the project's plans, requirements, and overall system architecture were well-articulated and communicated effectively.
  - **Responsibility:** I contributed by reviewing documents, assisting in writing the Project Plan and Software Requirements Specification (SRS). I provided ideas for the project demo submission and helped write the project report. Additionally, I assisted in drafting the technical manual to explain key functionalities and the system's architecture clearly.

## Achievements and Significance

During this project, I was able to contribute significantly to both the development and refinement of key functionalities. My active participation in team discussions via Slack, task management through Trello, and consistent GitHub commits contributed to maintaining strong communication and ensuring that the development process remained streamlined. My work on UI features, such as the sidebar management and chat options, directly enhanced the application's usability and responsiveness, providing users with a more intuitive and efficient interface. Developing hooks for message copying and chat title updates not only refined the user experience but also deepened my understanding of state management and component interaction within React.

Implementing the message regeneration feature was a key learning opportunity that expanded my knowledge of API calls and middleware interactions, which are critical in ensuring smooth backend and frontend communication. My research into metadata and the client's repository files helped the team gain a comprehensive understanding of the codebase, fostering a more collaborative development environment. The addition of the sidebar dropdown chat grouping by date significantly improved user interaction with the chat management system.

Through active participation in debugging and testing, I honed my problem-solving skills and improved the application's overall stability. Additionally, my involvement in the project's documentation and report writing allowed me to articulate the technical details of the project, further solidifying my understanding of system architecture and data flow.

In conclusion, these contributions were not only significant in advancing the project's goals but also instrumental in my personal development as a developer. They equipped me with practical



experience in UI development, API integration, and technical documentation, which are invaluable for my growth in the field.

## List of Contributions

### Contributions

Below is a detailed list of my significant contributions to the project. Where applicable, references to specific deliverables or documentation have been linked, moreover, list of evidence is also given.

Contribution	Description of Contribution	Hours Spent	Path/Deliverable (where applicable)
Correspondence and Team Communication	<p>Regular correspondence through Slack, Trello, and GitHub was essential to my contribution and collaboration with the team. Slack facilitated real-time communication, allowing me to provide frequent updates on my progress, discuss any challenges I encountered, and seek quick feedback or assistance from team members.</p> <p>For each technical Trello task, I created corresponding branches on GitHub to streamline code integration, with regular commits that reflected incremental development progress.</p> <p>This integrated workflow between Slack, Trello, and GitHub fostered effective collaboration and allowed me to contribute to the project in a structured and accountable manner.</p>	50	<p>GitHub repository: <a href="https://github.com/rajnaruka06/PoliQ.git">https://github.com/rajnaruka06/PoliQ.git</a></p> <p>Trello board</p>
SRS (Software Requirements Specification) and Project Plan Document	Contributed to the overall description section and editing of the document, as part of the first submission requirements. This was the foundational draft of the Software Requirements Specification (SRS) and Project Plan,	15	

	<p>where I was delegated to work on the general overview, ensuring it captured the project's objectives, scope, and intended impact. Additionally, I contributed to the organization and structure section, helping to create a logical flow and coherence across the document. My role also involved assisting in editing and refining content to enhance clarity and alignment with the client's expectations, ensuring a professional and cohesive document for submission.</p>		
Frontend Development - Sidebar Functionality	<p>As soon as frontend development began, tasks were created on Trello to implement specific features such as hiding the sidebar, deleting all chats, and pinning/unpinning chats. I took ownership of these tasks, actively working to develop and test these functionalities. Although I initially held off on making GitHub commits due to an overlap of responsibilities among team members—where others were simultaneously working on the same features—I completed my implementations independently. Despite the initial clash in delegation, the team later acknowledged my efforts and contributions, recognizing the work I had done even though some of my initial implementations were ultimately not integrated into the final codebase.</p>	20	Trello board
Three Dot Menu for Chat Options	<p>Implemented a vertical three-dot menu within the sidebar, providing options for managing each chat with functionalities such as delete, pin/unpin, archive, and rename. This menu was designed to improve user experience by consolidating key actions into a compact, accessible interface, streamlining</p>	8	<p>GitHub repository</p> <p>Trello board</p>

	chat management. To make sure that the feature integrates well in the application, commit was made in the Github repository.		
Sidebar Chat Hover Features	Developed a dynamic functionality within the sidebar to display the three-dot menu only when the user hovers over a chat entry, reducing visual clutter and creating a cleaner interface. Additionally, implemented a distinct color highlight for the currently active chat, ensuring users can easily identify which conversation is open. This functionality enhances the user experience by making the interface more intuitive and visually organized.	8	<p>GitHub repository</p> <p>Trello board</p>
Metadata Research and Preparation	Researched and documented the metadata for electoral data files provided by the client, covering key details such as filename, description, and usage for CSV files from 2004 to 2013, which included information on vote counts, political parties, and regional data across Australia. Analyzed critical backend files like house.py, models.py, and csv_to_db.py to understand their role in Object-Relational Mapping (ORM) and created a document summarizing their functions. Additionally, I developed an overview of ORM principles and the structure of the client's application, including a flow diagram to illustrate data relationships and usage within the backend. This documentation enhanced the team's understanding of the repository and facilitated smoother backend integration.	30	<p>Refer to the chart and visual representations section</p> <p>Trello Board</p> <p>Client's repository:  <a href="https://github.com/PoliMap-Pro/coldjango.git">https://github.com/PoliMap-Pro/coldjango.git</a> </p>
Copy Message Function	Before implementing the "copy message" feature, I needed to understand the middleware	10	GitHub repository

	<p>structure first. The "copy message" hook appears under each chatbot response in the chat window. When a user hovers over a response, the "copy" icon becomes visible. By clicking on it, the chatbot's response is copied to the clipboard, and a success message confirms the action. This feature enhances user convenience, allowing quick copying and sharing of information generated by the chatbot.</p>		Trello board
Hooks - Updating Chat Title	<p>Developed a chat title updating functionality. This involved understanding how API requests for updating specific chat details are processed, specifically the PUT request and are stored in MongoDB. I then implemented a hook that allows users to update the chat title either by double-clicking on the title in the sidebar to enter a new name or by selecting the "Rename" option from the chat's three-dot menu. This functionality provides users with an intuitive way to personalize and organize their chat sessions, improving the overall usability of the sidebar.</p>	10	GitHub repository  Trello board
UI Debugging	<p>During development, I took responsibility for debugging several critical UI issues to ensure a smooth and visually consistent user experience. One of the main tasks involved aligning the "update chat title" functionality, as it initially caused a mismatch in the UI layout. I adjusted the component styling to ensure it stayed perfectly aligned, providing a cleaner look. Additionally, I resolved timing issues with the hover functionality on the three-dot menu, which would close inconsistently when</p>	8	GitHub repository  Trello board

	users clicked outside the menu. By fine-tuning the timing and responsiveness, I improved the interaction flow, ensuring that the menu would close reliably upon outside clicks. These debugging efforts contributed significantly to a polished and user-friendly interface.		
System Testing	I conducted thorough testing on the system by designing up to ten questions specifically targeting the client's ElecData database. For each question, I crafted SQL queries to retrieve the correct responses manually from the database, allowing for an accurate comparison against the responses generated by the LLM chatbot. This approach ensured a direct evaluation of the LLM's accuracy and effectiveness in interpreting user queries. For each test, I documented the chatbot's response, the manual SQL query result, and noted any discrepancies, categorizing them by status to identify areas where the LLM's performance could be further refined. This testing process was instrumental in validating the chatbot's functionality and identifying improvement areas to enhance response accuracy. As a result, also got appreciation from the project manager.	15	Refer to the chart and visual representations section  Trello board
Regenerate Messages Hook	This functionality enables users to request a refreshed response from the chatbot. Positioned next to the "copy message" icon, this option becomes visible when the user hovers over the chatbot's response. Upon clicking, the message is regenerated, and a success notification confirms the updated	15	GitHub repository  Trello board

	<p>response. This task involved a deep dive into the middleware structure, specifically focusing on handling PUT requests via the API to trigger a new response generation. I also examined how these updates are processed in MongoDB to ensure the original message is overwritten with the new content without data inconsistency. This functionality provided users with a seamless way to refine responses, improving the chatbot's adaptability and user experience.</p>		
Report Generation Research	<p>Researched how report generation is implemented in the client's application by thoroughly examining key files within the client's repository, including Graph.tsx, Map.tsx, and MapSection.tsx. These files were essential in understanding how report visuals are structured, particularly using Leaflet for interactive maps and GeoJSON for geographical data representation. Leveraging these libraries was crucial for generating dynamic and interactive reports. I documented the process, highlighting specific components and configurations needed to replicate or extend these reporting capabilities within our project. This documentation was shared with the team, serving as a reference to implement similar report generation features.</p>	8	<p>Refer to charts and visual representations section</p> <p>Client's repository:  <a href="https://github.com/PoliMap-Pro/coldjango.git">https://github.com/PoliMap-Pro/coldjango.git</a></p>
Dropdown Date Menu for Chats	<p>Developed a dropdown menu feature in the sidebar to group chats by date, enhancing the organization and accessibility of chat sessions. This functionality enables users to view dates as collapsible headers, reducing sidebar clutter. When expanded, the chats corresponding to a</p>	5	<p>GitHub repository</p> <p>Trello board</p>

	specific date become visible, allowing users to navigate through chat history more efficiently. This feature required careful planning to integrate seamlessly into the sidebar structure and ensures that users can quickly locate conversations based on their dates.		
Project Demonstration	For the project demonstration, video was created for which script and storyboard were made. Helped develop the concept and script for the project demonstration.	3	
Project Report - Requirements Analysis	Contributed to the "Requirements Analysis" section in the final project report. My work included detailing both functional and non-functional requirements to clearly define the system's capabilities and quality attributes. I also documented stakeholder needs, mapping out primary and secondary users and their expectations from the system. Additionally, I assisted in outlining requirement changes, especially those influenced by client feedback. This comprehensive analysis provided a structured foundation for development, ensuring alignment between the project's objectives and the stakeholders' expectations.	10	
Technical Manual	Contributed to writing and editing the technical manual as part of the final project report submission. This manual provides comprehensive documentation of the system's architecture, detailing both frontend and backend functionality. For the frontend, I documented the key components, including the implementation of hooks, UI features, and API integration points. On the backend, I described	20	

	the API endpoints, database interactions, middleware structure, and security protocols. This technical manual is designed to serve as a resource for developers and system administrators, ensuring clarity on system setup, functionality, and maintenance procedures.		
--	---	--	--

## Charts and Visual Representations

Here are the charts or visual representations of the contributions made:

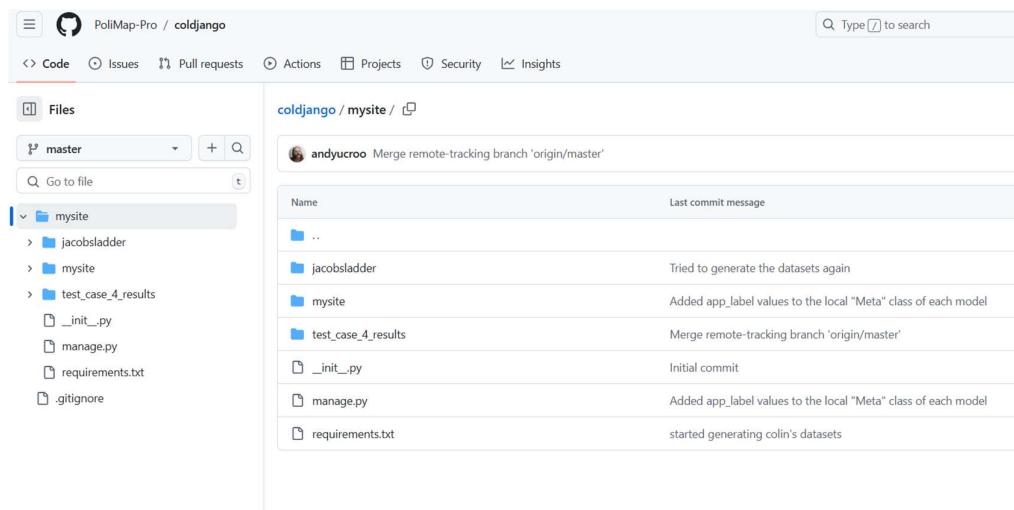


Figure 1: Client's repository structure, crucial for data preparation and report generation features

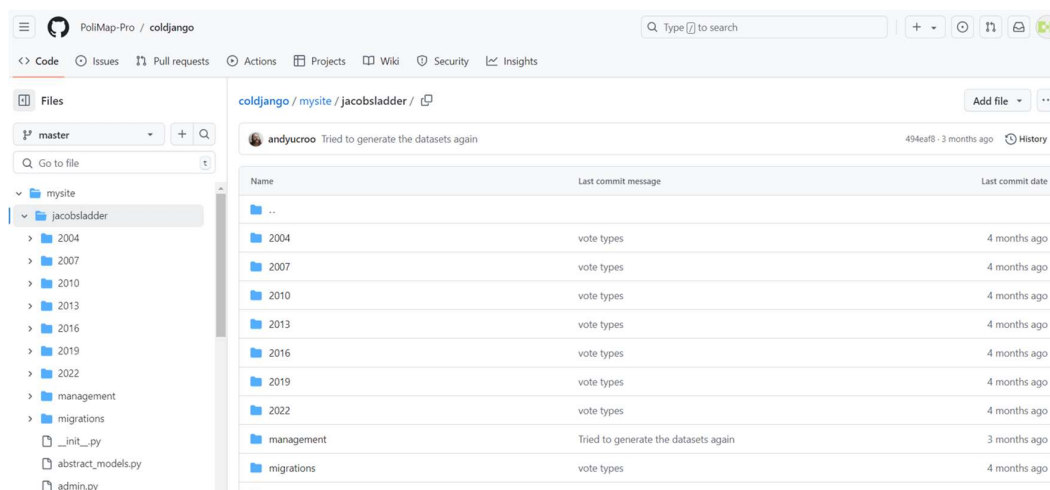


Figure 2: Client's repository structure indicating the files integral for meta data preparation



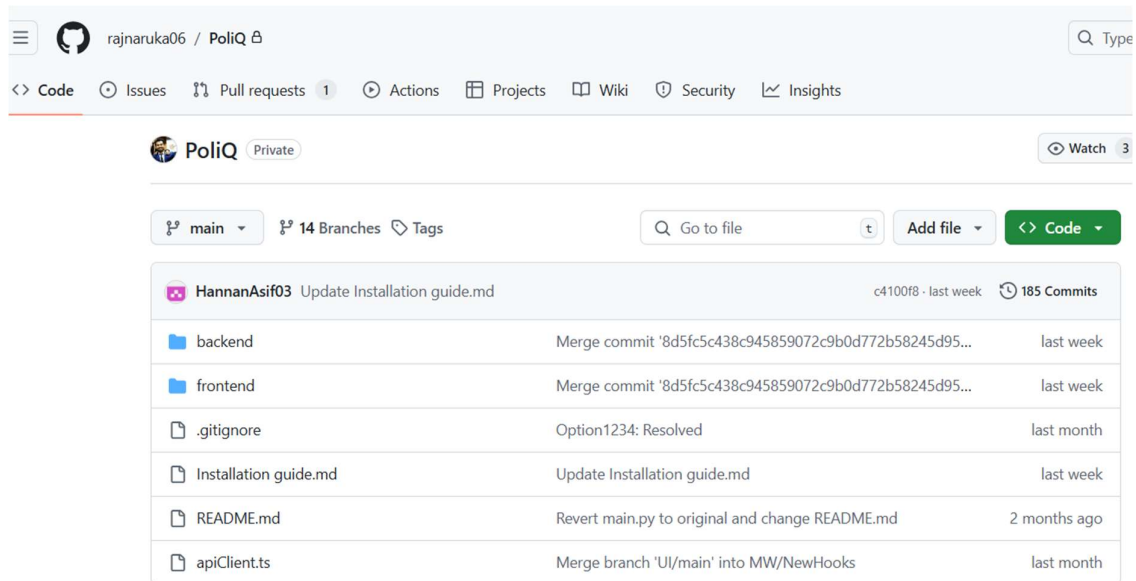


Figure 3: Our team's GitHub repository, to maintain even contributions and for seamless integration

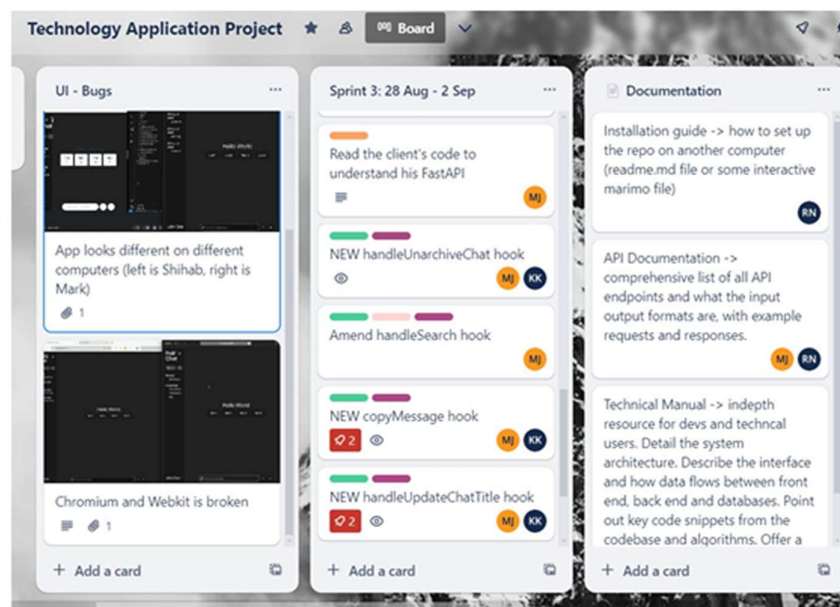


Figure 4: Trello Board

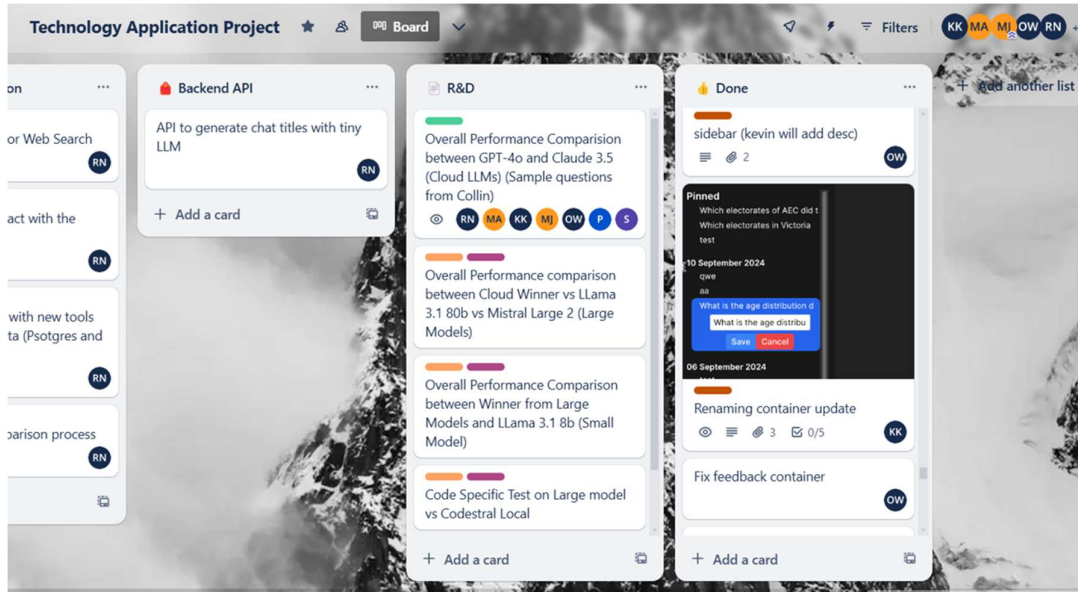


Figure 5: Effective delegation of tasks through Trello Board

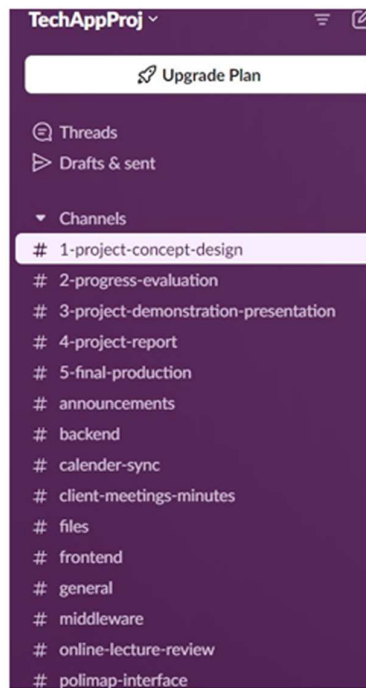


Figure 6: Slack Channel

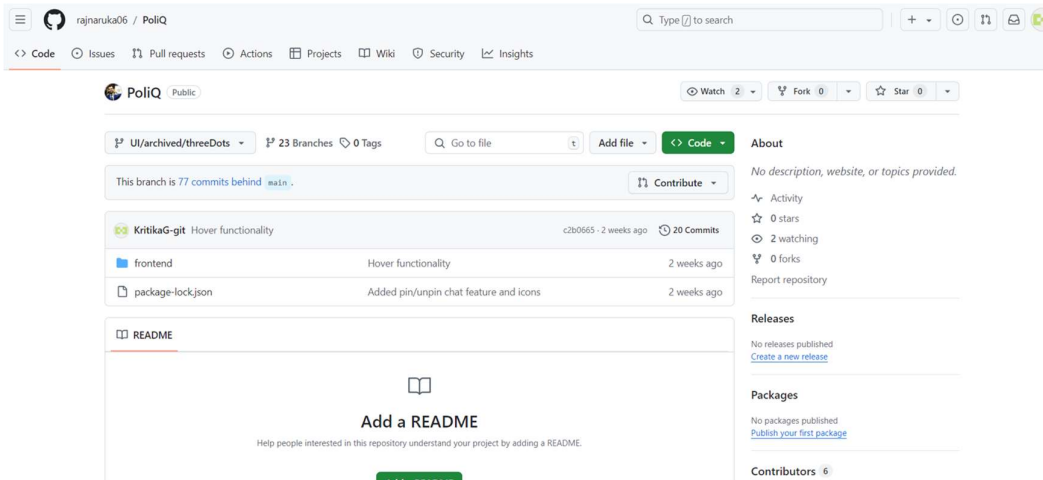


Figure 7: Example of a branch I created along with the commit made

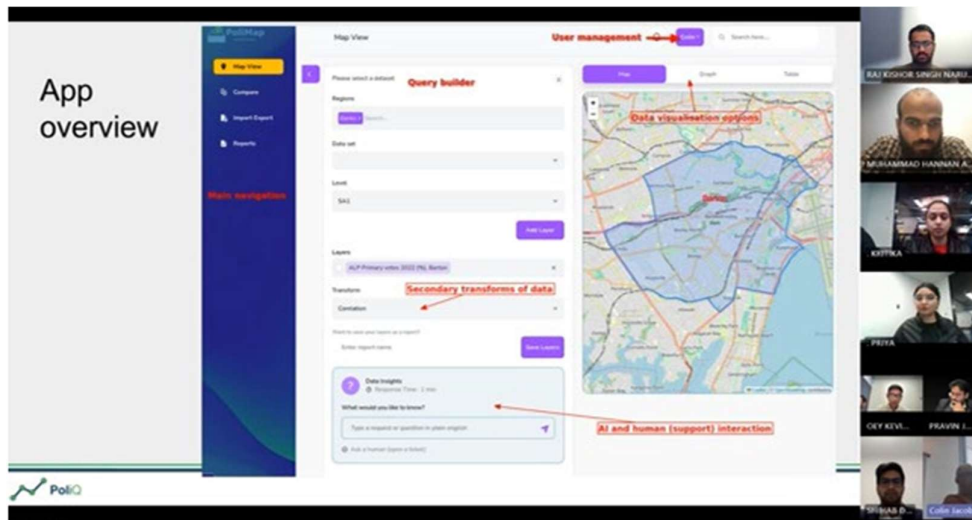


Figure 8: Meeting with the client in the initial weeks, which helped us to set up the project, during the meeting I also made notes and shared them with the team

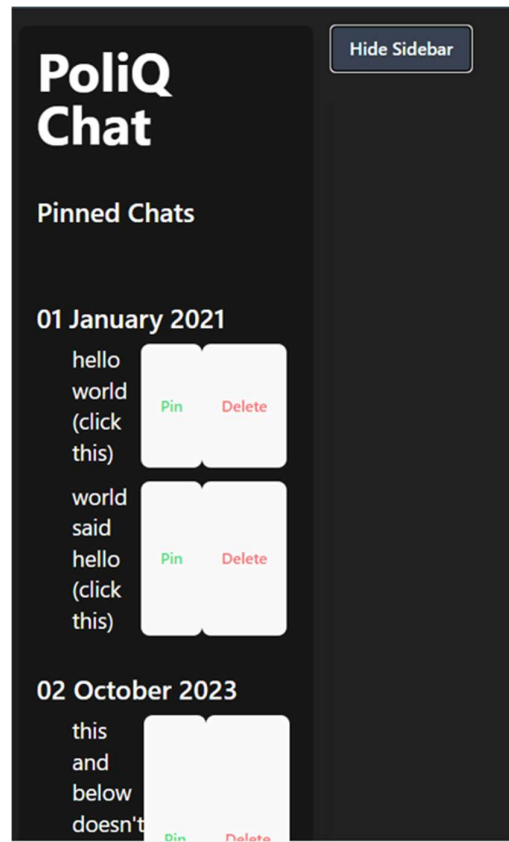


Figure 9: Initial UI task to implement hiding sidebar, pin/unpin a chat and delete the chat

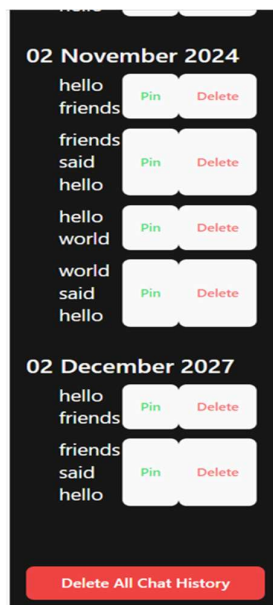


Figure 10: Initial UI task to delete all chats

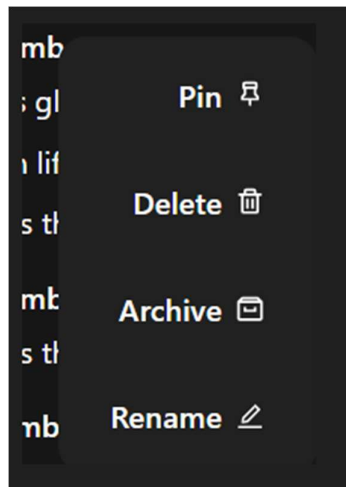


Figure 11: Menu for the different chat management operations

HTTP Method	Endpoint	Description	Parameters
GET	/api/chats/all	Fetch all chat history	Query: userid
GET	/api/chats/{chatid}/messages	Fetch messages for a specific chat	Path: chatid, Query: userid
POST	/api/messages/send	Send a message and process it	Body: MessageRequest (chatid, content), Query: userid
GET	/api/chats/{chatid}/latest	Get latest messages from a chat	Path: chatid, Query: userid, limit (default: 2)
PUT	/api/chats/{chatid}/pin	Pin a chat	Path: chatid, Query: userid
PUT	/api/chats/{chatid}/unpin	Unpin a chat	Path: chatid, Query: userid
DELETE	/api/chats/{chatid}/delete	Delete a chat	Path: chatid, Query: userid
PUT	/api/chats/{chatid}/archive	Archive a chat	Path: chatid, Query: userid
PUT	/api/chats/{chatid}/unarchive	Unarchive a chat	Path: chatid, Query: userid
GET	/api/chats/search	Search chats	Query: term, userid
PUT	/api/chats/{chatid}/title	Update chat title	Path: chatid, Query: userid, newTitle
GET	/api/chats/{chatid}/messages/{messageid}	Copy a specific message	Path: chatid, messageid, Query: userid
PUT	/api/chats/{chatid}/messages/{messageid}	Update a message and generate new response	Path: chatid, messageid, Query: newContent, userid

Figure 12: Before diving into the tasks, I understood API calls and middleware structure

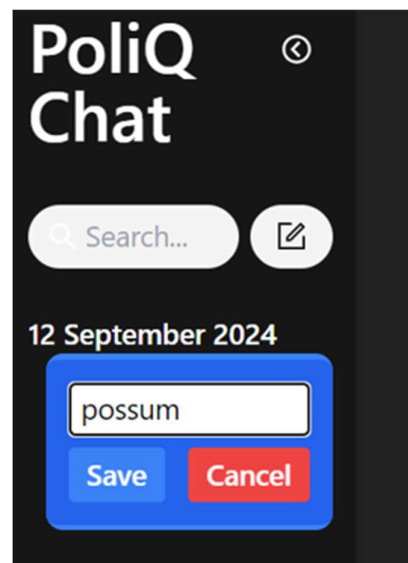


Figure 13: Update chat title feature

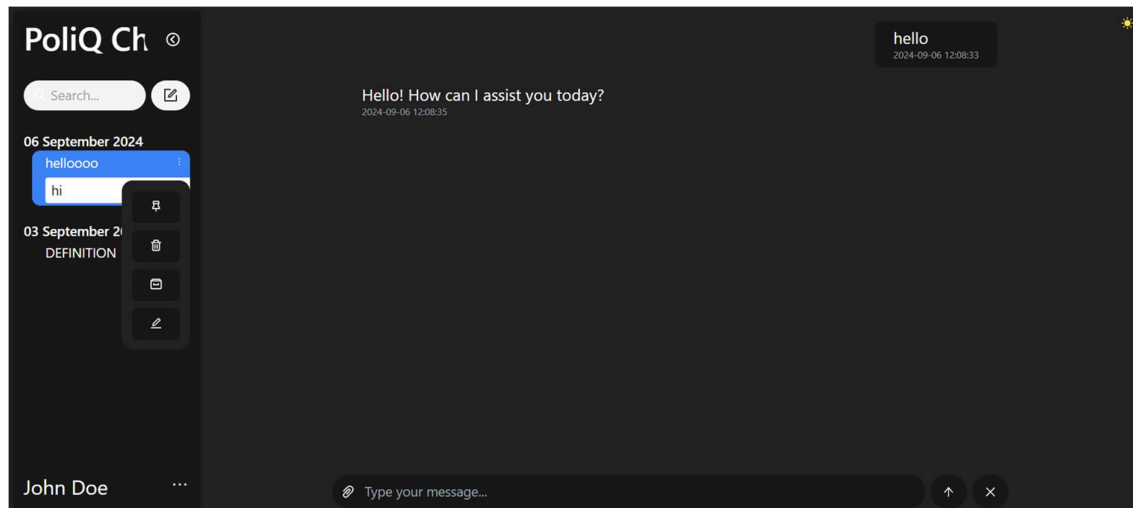


Figure 14: Update chat title feature along with the final version of the menu for chat management functions.

2004

**Document Name:** 2004\_Federal\_Election\_Distribution\_of\_Preferences\_By\_Division.csv

**Description:** This document contains data on the distribution of preferences in the House of Representatives during the 2004 Federal Election in Australia. It provides a detailed breakdown of how voter preferences were allocated to different candidates within each electoral division, capturing the flow of preferences as votes were redistributed according to the Australian electoral system.

**Key Information:**

- **Geographical Data:**
  - StateAb: State abbreviation (e.g., ACT for Australian Capital Territory).
  - DivisionID: Unique identifier for each electoral division.
  - DivisionNm: Name of the electoral division (e.g., Canberra).
- **Candidate Details:**
  - CandidateID: Unique identifier for each candidate.
  - Surname: Candidate's last name.
  - GivenNm: Candidate's first name.
  - PartyAb: Abbreviation of the candidate's party (e.g., ALP for Australian Labor Party).
  - PartyNm: Full name of the candidate's party.
  - SittingMemberFl: Indicates if the candidate was the sitting member.
- **Vote Data:**
  - CountNumber: The count round number in the distribution of preferences.

Figure 15: Comprehensive document created for metadata preparation

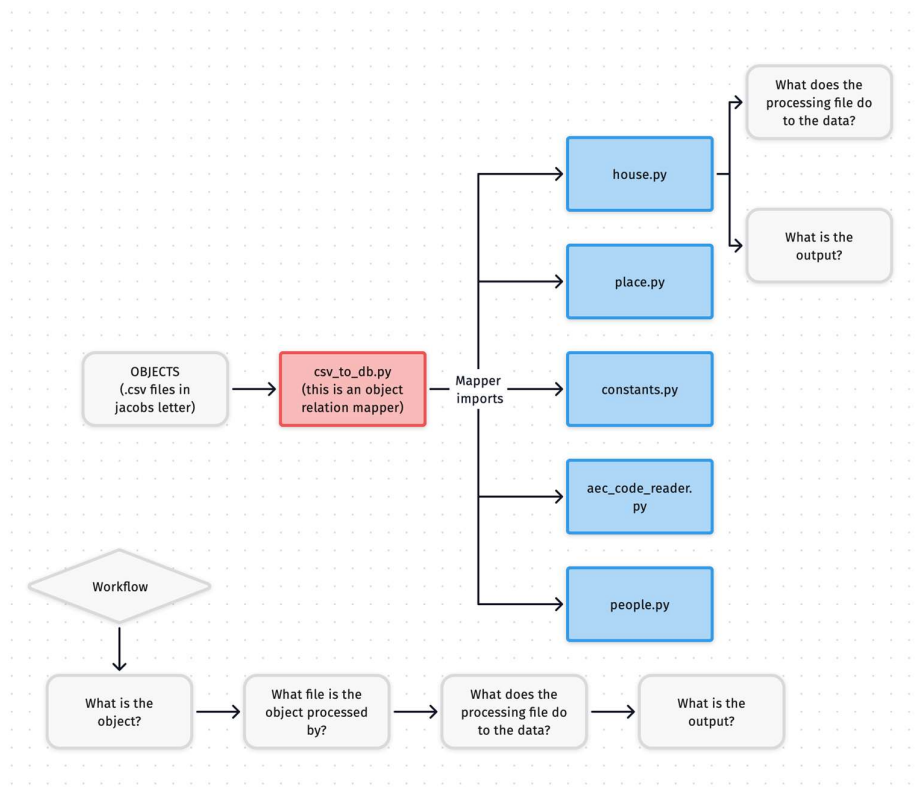


Figure 16: Workflow created to understand Object Relational Mapping (ORM) to understand Client application's data structure

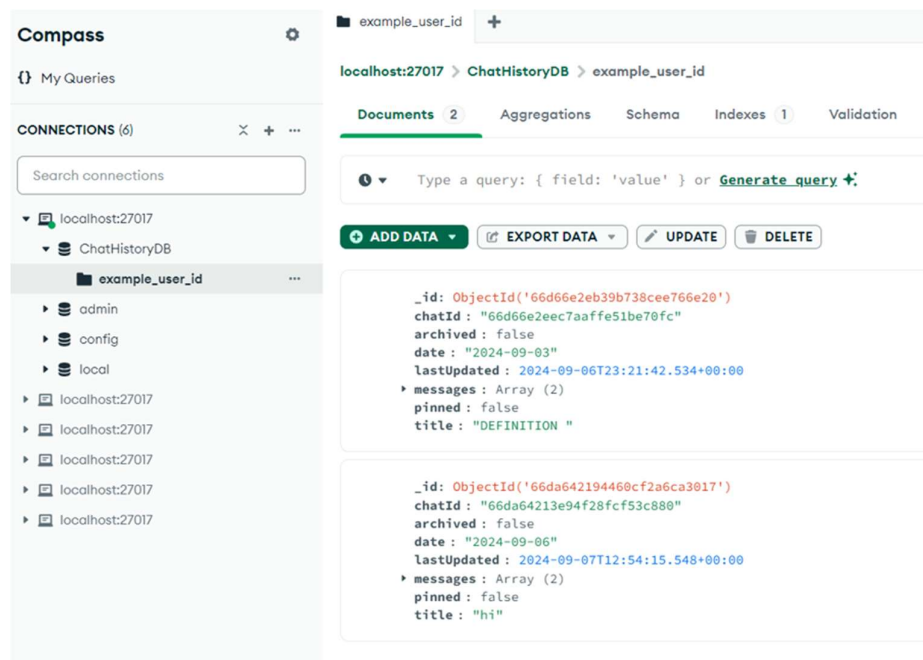


Figure 17: MongoDB structure, showcasing how chat is managed in the backend



### Graph.tsx:

The provided `graph.tsx` file generates graphs using `Highcharts` based on the data from Redux state (layers) or a comparison context (`CompareContext`). It involves a few key elements:

#### 1. Data Preparation:

- `formatSeriesData` function formats data series for plotting.
- Combines data series from multiple datasets for each layer.

#### 2. Dynamic Chart Configuration:

- Multiple chart configurations are dynamically created (`chartOptions1`, `chartOptions2`, `chartOptionsLayer`) based on the comparison context or available layers.
- Uses `Highcharts` options to configure different aspects of the chart like title, `yAxis`, `xAxis`, tooltip, etc.

#### 3. Interactive Controls:

- `ChartControls` is a component that allows the user to change the chart type dynamically.

#### 4. Rendering:

- Based on the conditions (Compare context or layers), different charts are rendered using `HighchartsReact`.

### Insights for Report Generation

From the `graph.tsx` file, we can derive the following points for automating the report generation task:

---

Figure 18: Comprehensive document to understand report generation in client's application

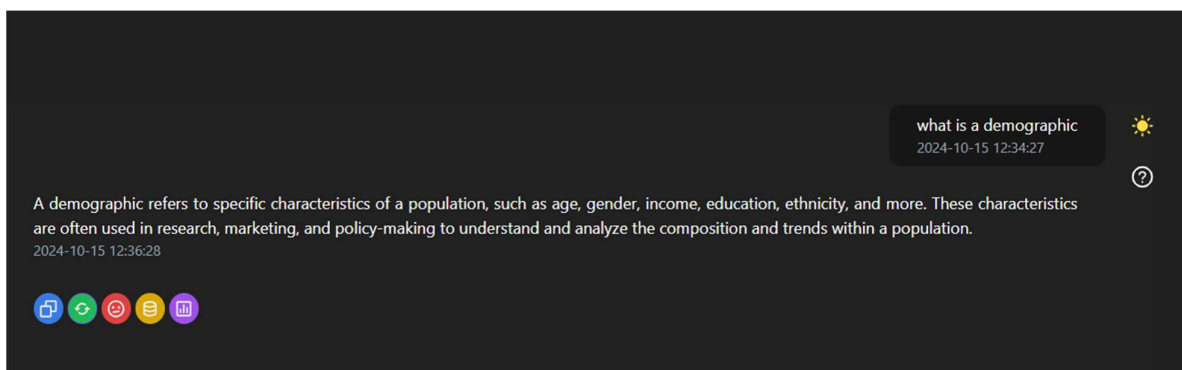


Figure 19: Copy and regenerate buttons along with other options below a response created by the chatbot.



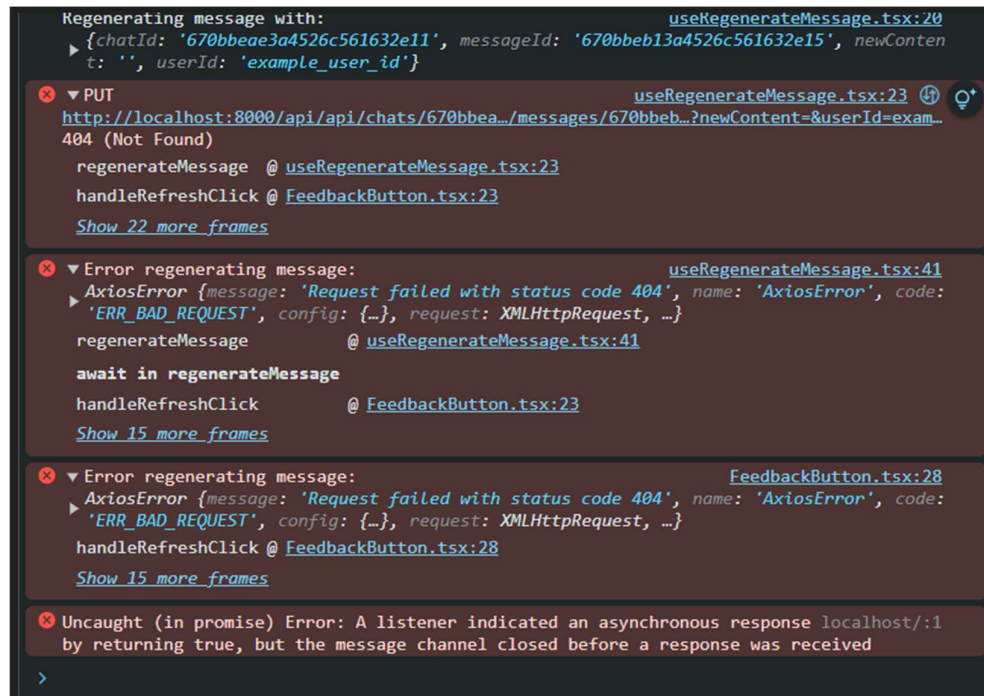


Figure 20: Errors encountered while implementing regeneration of response by the chatbot, which helped me in understanding the middleware structure more effectively

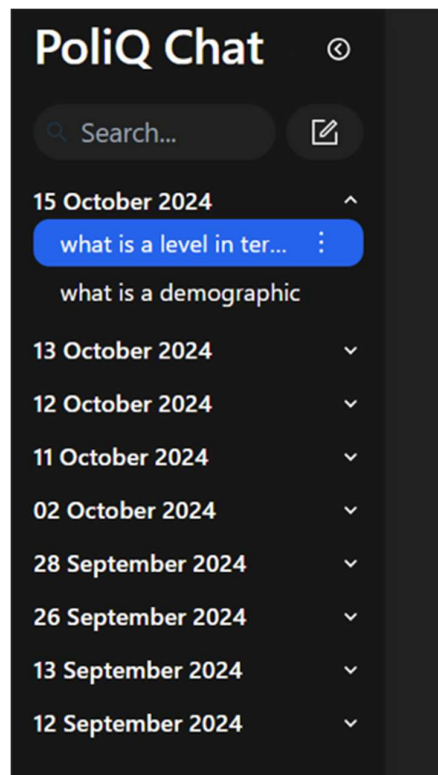


Figure 21: Grouping chats by date functionality

<div> <div>Testing</div> <div>File Edit View Insert Format Data Tools Extensions Help</div> <div> <div>90%</div> <div>\$ % .0 .00 123</div> <div>12</div> <div>B I A</div> </div> </div>			
CONTEXT: Elecdata Database in Postgres			
	A	B	C
1	CONTEXT: Elecdata Database in Postgres		
2			
3			
4	Prompt	LLM Response	Manual Query(Kritika)
5	Where do the renters live in Wills?	No data fetched from database	<pre>SELECT   geo.name AS suburb,   COUNT(elecdata_enrollment.id) AS number_of_renters FROM   elecdata_geography AS geo JOIN   elecdata_enrollment ON geo.id = elecdata_enrollment.seat_id WHERE   geo.name = 'Wills' GROUP BY   geo.name;</pre>
6	Is the Greens vote correlated to the number of renters?	No data fetched from database	<pre>SELECT   vt.election_id,   vt.candidate_id,   vt.primary_votes AS greens_votes,   en.number_enrolled AS number_of_renters FROM   elecdata_votetally AS vt JOIN   elecdata_enrollment AS en ON vt.election_id = en.election_id JOIN   elecdata_party AS party ON vt.candidate_id = party.id WHERE   party.name = 'Greens';</pre>
7	What is the percentage of renters in the Wills electorate?	Invalid SQL query	<pre>SELECT SUM(en.number_enrolled) * 100.0 / SUM(g.pop2005) AS   renter_percentage FROM elecdata_geography AS g JOIN elecdata_enrollment AS   en ON g.id = en.seat_id WHERE g.name = 'Wills';</pre>

Figure 22: System Testing documentation

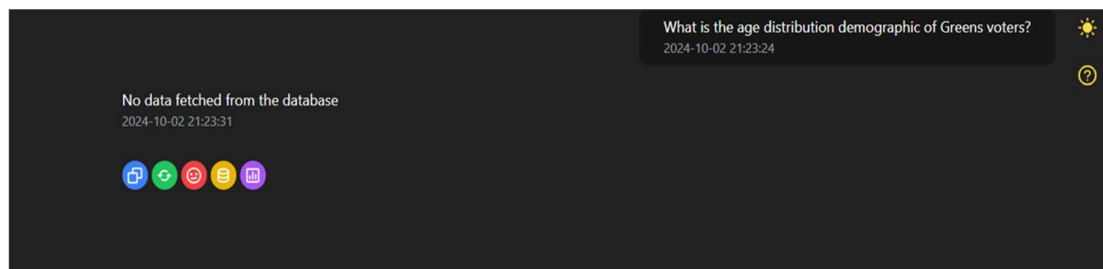


Figure 23: Testing questions to get the LLM response which well help in understanding backend and troubleshooting where necessary

Data Output	Messages	Notifications
<div> <div>+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>		
suburb character varying (50)	number_of_renters bigint	green_party_votes bigint

Figure 24: Documented the responses given by writing the SQL query (PgAdmin)

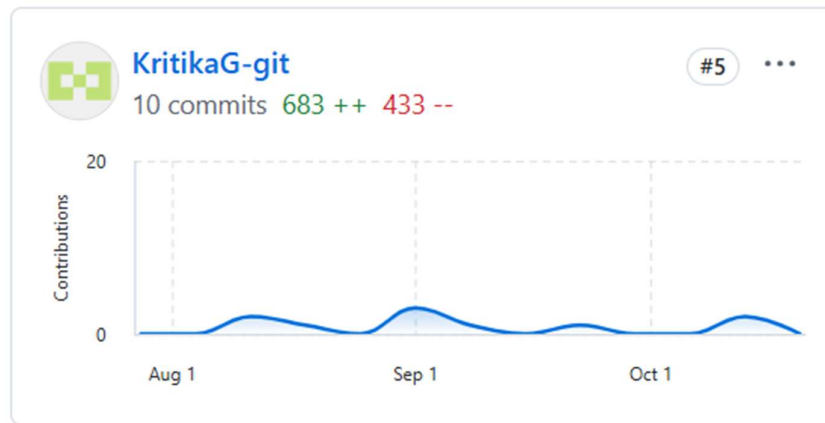


Figure 25: Commits made to the GitHub repository

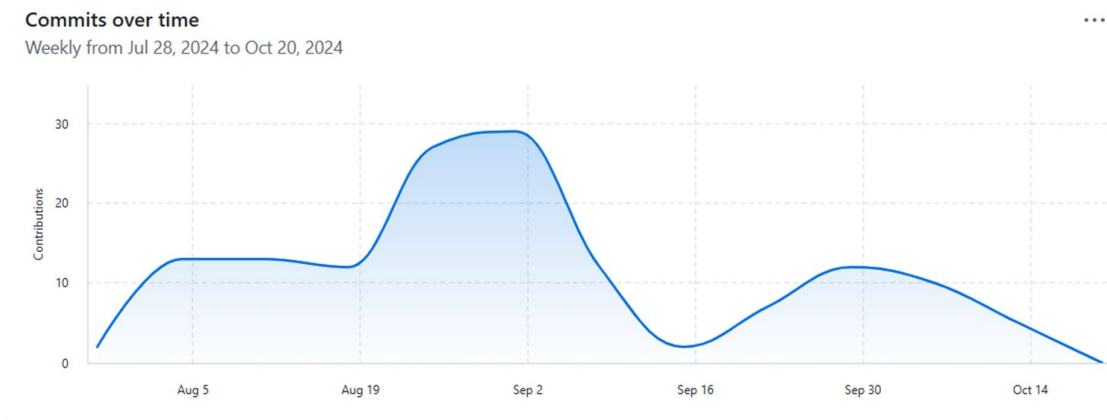


Figure 26: GitHub commits over time for the entire team

## Learning Outcome Assessment

### ULO 1:

*Select and apply appropriate technologies and sources of information to provide cost-effective solutions to real-life problems in the context of a technology application project.*

- **Frontend Development:** I applied React to implement critical features, such as chat management (pin/unpin chats, delete chats), a "copy message" functionality for ease of use, and a sidebar drop-down menu for grouping chats by date, which enhanced the organization of chat sessions. Additionally, I contributed to UI improvements like dynamically displaying the three-dot menu on hover for chat options and ensuring that the current chat was visually highlighted. These features were successfully integrated into the final product, enriching the user experience and helping shape the frontend's overall direction and functionality.
- **Hooks and State Management:** I developed frontend hooks for updating chat titles and regenerating chatbot responses, ensuring efficient interaction between the user and the system. The "update chat title" feature allowed users to seamlessly rename chat sessions directly from the sidebar, while the "regenerate messages" hook enabled users to refresh

chatbot responses with updated information. These functionalities streamlined the user experience, making it easier to manage and organize chat sessions effectively.

- **API Integration and Testing:** Through debugging the UI and testing the system with real-time questions from the database of client's application, I validated the system's LLM responses against SQL query outputs. This ensured that the solution was not only cost-effective but also accurate in responding to real-world queries.

## ULO 2:

*Reflect on the technology challenges of performing scholarly and applied research.*

- **Metadata Research and Preparation:** I researched and prepared a document that explained the client's repository files, which clarified the project's structure for the team. This involved understanding object-relational mapping (ORM) and recognizing how RAG (Retrieval-Augmented Generation) could enhance the system's data retrieval capabilities. My research highlighted challenges in dealing with large datasets and integrating machine learning models into a dynamic system.
- **Report Generation Research:** I conducted research on potential solutions for generating reports in the client's application, which contributed to our discussions on expanding system functionality. This experience revealed the complexity of balancing system capabilities with client expectations while addressing the technical limitations of report generation.

## ULO 3:

*Select and apply proper research methods and practices to interpret and critically appraise data acquired during technology application.*

- **Testing with ElecData database (data provided by the client):** I contributed to testing by comparing SQL queries with LLM responses, ensuring that the results were aligned with the manual responses. This required a systematic approach to designing test cases and analysing discrepancies, which allowed us to refine the system for better accuracy.
- **Researching and Implementing Hooks:** I researched and applied state management practices while creating hooks for the frontend. These hooks allowed users to efficiently interact with chats by copying messages or updating chat titles. The design of these features was based on user feedback and testing, ensuring that they met the real needs of the application.

## ULO 4:

*Plan, design, develop and present solutions for a real-life project in a team environment that demonstrates scholarship in one field relevant to technology application.*

- **SRS and Project Plan:** I contributed to the Software Requirements Specification (SRS) by drafting the overall description section and assisting in the document's editing. Additionally, I contributed to the organization and structure section of the project plan, ensuring both documents clearly conveyed our system's objectives, functionalities, and development timeline.
- **Frontend Development** I planned and developed key features such as the chat management system and sidebar functionalities, including the drop-down for grouping

chats by date and the three-dot menu for chat management options (e.g., delete, pin/unpin, archive). These features significantly enhanced system usability and organization.

- **Collaboration and Communication:** I actively participated in team meetings and discussions across Slack, Trello, and GitHub, contributing to planning and issue resolution. On GitHub, I made regular commits to contribute to the application's development. I also documented initial client meetings to ensure the team remained aligned with client expectations and project goals.

## ULO 5:

*Demonstrate an understanding of the relationships between the knowledge studied in technical units and its practical application in a real-life organisational setting, through reflective practice, and self and peer evaluation.*

- **Frontend and Backend Integration:** I leveraged my technical knowledge from coursework in applying React for frontend development and FastAPI for backend services, ensuring smooth communication and functionality between the two. This integration required a practical understanding of REST APIs, asynchronous data handling, and component-based architecture, aligning with principles learned in my technical units.
- **Documentation and Report Writing:** I contributed to the requirements analysis section of the project report, where I documented key functional and non-functional requirements based on our system's architecture. Additionally, I helped draft the technical manual, providing detailed descriptions of both frontend and backend operations. These tasks involved reflecting on how different components interacted within the system, which helped solidify my understanding of architecture design and integration. I also actively participated in peer evaluations during team discussions, where I provided feedback to refine our project's direction and improve overall quality.

## ULO 6:

*Demonstrate an awareness and understanding of ethical and professional behaviours as well as social and cultural perspectives in an organisational context.*

- **Collaboration via Communication Channels:** I maintained open and respectful communication with my team through Slack, Trello, and GitHub, consistently fostering a professional environment. During discussions, especially in early phases when delegation conflicts arose, I worked to ensure that all team members' contributions were acknowledged and valued. This approach helped create a positive and inclusive working atmosphere, promoting mutual respect and constructive collaboration.
- **Stakeholder Considerations in Documentation:** While developing key documentation such as the project report and technical manual, I focused on making the materials accessible and clear for both technical and non-technical stakeholders. By ensuring that the language was inclusive and understandable, I facilitated effective collaboration and alignment among team members and improved client engagement with the system's functionalities.

## Reflection

### The most important things I learnt:

Over the course of this project, I gained a deep understanding of full-stack development, project documentation, and collaborative work in a real-world setting. Specifically, I learned about integrating frontend and backend components, managing APIs, and working with databases like MongoDB and PostgreSQL. Additionally, the importance of effective communication in team settings became clear to me, as it was essential for coordinating tasks, resolving conflicts, and ensuring alignment on goals. I also discovered the value of user-centred design and accessibility in technology applications, which was integral to our project's success.

### The things that helped me most were:

- **Team Meetings and Communication Tools:** Regular team meetings and the use of Slack and Trello allowed us to stay updated on project status, resolve any issues, and assign tasks efficiently. This structured communication helped us avoid misunderstandings and allowed me to clarify tasks or seek assistance when needed.
- **Hands-On Learning in GitHub:** Working directly in GitHub, including creating branches for specific tasks and making commits, was instrumental in improving my version control skills. This experience also taught me how to handle merge conflicts and collaborate on a shared codebase.
- **Research and Documentation:** Conducting metadata research and preparing technical documentation helped solidify my understanding of the system's architecture and the nuances of client requirements.

### I found the following topics particularly challenging:

- **Object-Relational Mapping (ORM):** Understanding and implementing ORM in the backend was initially difficult as it involved bridging the gap between object-oriented programming and relational databases. Familiarizing myself with this concept took time and required substantial research.
- **Middleware and API Integration:** Working with middleware and ensuring smooth API integration between frontend and backend components was challenging, particularly when ensuring data consistency and error handling. This task required a detailed understanding of request/response handling and how changes in one part of the application could affect others.

### I found the following topics particularly interesting:

- **Retrieval-Augmented Generation (RAG):** The concept of RAG was fascinating as it involved integrating natural language processing with database queries, which made the chatbot's responses more relevant. Learning about this technique expanded my knowledge of AI applications in data retrieval.
- **User-Centered Design and UI/UX Principles:** Designing features that prioritized user experience, like the three-dot menu in the sidebar or the copy message feature, was particularly engaging. I enjoyed the challenge of making the application intuitive and visually appealing.

**I feel I learnt these topics, concepts, and/or tools really well:**

- Frontend Development with React: Working extensively with React for UI components, hooks, and state management helped me become proficient in this framework. I now feel confident in using React to build interactive, responsive interfaces.
- GitHub for Version Control: My GitHub skills improved significantly, particularly in managing branches, resolving merge conflicts, and coordinating code reviews. This will be beneficial in future collaborative projects.

**I still need to work on the following areas:**

- Implementing Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) Techniques: Although I understand the conceptual benefits of using LLMs and RAG for enhancing data retrieval and summarization, I have limited experience with their practical implementation. I aim to deepen my knowledge of how to integrate LLMs with data sources effectively, fine-tune models for specific use cases, and apply RAG techniques to optimize response accuracy and relevance.
- API Error Handling and Logging: I encountered difficulties when implementing error handling in API calls. In future projects, I aim to improve my skills in structuring effective logging mechanisms and handling different types of errors in API responses, ensuring a more robust and maintainable system.

**My progress in this unit was ...:**

From my perspective, my progress in this unit was steady and consistent. I encountered various challenges, particularly with backend integration and certain frontend features, but with regular updates and feedback, I was able to address these issues. Using GitHub, I tracked my contributions and could visually see my progress over time. Reviewing my commits helped me identify areas for improvement and better understand the flow of the project.

**This unit will help me in the future:**

The skills I learned in this unit—such as full-stack development, version control, team collaboration, and API integration—will be invaluable in both my studies and career. In particular, the practical experience with FastAPI, React, and database management provides a strong foundation for future technology projects. I also gained insight into how real-world applications are designed, tested, and documented, which will be beneficial in any technical role.

**If I did this unit again, I would do the following things differently:**

- Improve Initial Planning and Task Delegation: Early on, our team faced some tasks overlap and delegation issues. If I were to repeat this experience, I would suggest clearer initial planning and task allocation to prevent duplicate efforts.
- Prioritize Documentation Throughout Development: Instead of waiting until the end to complete documentation, I would prioritize updating the technical manual and user documentation as we progress. This would provide more clarity and make it easier to reference information later.



### Other...:

This project taught me the importance of balancing technical and interpersonal skills in a team environment. I learned that while technical knowledge is crucial, clear communication, understanding different perspectives, and actively seeking feedback can significantly enhance the quality and cohesion of the final product. Overall, this experience has equipped me with both the skills and the confidence to tackle more complex projects in the future.

## Conclusion

In summary, I believe that I have clearly demonstrated my contributions to the project in several key areas. I actively participated in frontend development tasks, including the implementation of features such as hiding the sidebar, managing chat functionalities, and adding a drop-down menu for grouping chats by date. My contributions to metadata research and preparation provided essential insights into the client's repository and technical requirements. I also played a significant role in the documentation process, including the project report, technical manual, and requirements analysis. Furthermore, I assisted in testing and validating the system, ensuring its alignment with the project's functional requirements and user expectations.

I have made significant contributions in the following aspects: the design and implementation of key frontend features, the integration of hooks for improving user interaction, and the preparation of detailed metadata research that supported team understanding. Additionally, my involvement in testing and debugging contributed to enhancing the overall performance of the application. Finally, my participation in team meetings and coordination across various communication channels helped ensure seamless collaboration and progress throughout the project.