

COSC 4P02: Software Engineering II.

Group 9: Progress Report 2

Scrum Master: Jaden Kuhn ([JK21PF@BROCKU.CA](mailto:JK21PF@BROCKU.CA) - 7249683)

Product Owner: Dea Kukuqani ([DK16QS@BROCKU.CA](mailto:DK16QS@BROCKU.CA) – 6196018)

Developers: Shijie Tong ([ST20AZ@BROCKU.CA](mailto:ST20AZ@BROCKU.CA) - 7081201)

Nicholas Caruso ([YQ20OZ@BROCKU.CA](mailto:YQ20OZ@BROCKU.CA) - 7189749)

Thomas Semenak ([TS19CP@BROCKU.CA](mailto:TS19CP@BROCKU.CA) - 6745038)

Dalton Morris ([DM20BQ@BROCKU.CA](mailto:DM20BQ@BROCKU.CA) - 7053184)

Chidera Nwana ([CN20RQ@BROCKU.CA](mailto:CN20RQ@BROCKU.CA) - 7078124)

Instructor: Naser Ezzati-Jivan

Teaching Assistant: Madeline Janecek

March 23<sup>rd</sup>, 2025

## Table of Contents.

Table of Contents.....	2
1. Introduction .....	4
2. Sprint Overviews.....	4
2.1. Sprint Durations .....	4
2.2. Sprint Objectives .....	4
3. Contributions and Key Decisions .....	5
3.1. Web Aggregator Improvements.....	5
3.2. Azure Integration of the Web Aggregator, Data Scraper, and LLM .....	5
3.3. Azure Configuration of the Database .....	6
3.4. Additional Azure Subscriptions .....	7
3.5. Content Categories Implementation .....	7
3.6. Automatic Scheduling of Content Generation Function .....	8
3.7. Social Media Export Function.....	8
3.8. Automated Component Testing .....	9
4. Challenges.....	10
4.1. Web Aggregator Improvements.....	10
4.2. Azure Integration of the Web Aggregator, Data Scraper, and LLM .....	10
4.3. Azure Configuration of the Database .....	10
4.4. Automatic Scheduling of Content Generation Function .....	11
4.5. Social Media Export Function.....	11
5. Jira Board Updates (Sprint 5 + Remaining Backlog) .....	11
6. Overall Sprint Assessment and Next Steps .....	12
6.1. Summary of Sprint 4 Achievements.....	12
6.2. Summary of Sprint 5 Achievements.....	12
6.3. Remaining Priorities for Next Sprints .....	12
7. Conclusion .....	13
Appendix A. Website and GitHub Repository Links .....	13
Appendix B. Meeting Minutes .....	13

Appendix C. Complete Contributions Table.....	14
Appendix D. Jira Sprint Burndown Charts and Backlog.....	17
Appendix E. GitHub Screenshots and Contributions .....	18
Appendix F. Website Screenshots .....	20

# 1. Introduction

The AI Powered Newsletter and Social Media Content Generator project aims to make content creation easier and more efficient for newsletters and social media. The goal is to automate tasks like gathering relevant content, summarizing it, and formatting it in a way that is useful for users. By using machine learning, data scraping, and a user-friendly interface, the platform will allow users to generate high-quality content with minimal effort.

This report outlines the progress made in the four and fifth sprints, covering key developments in the improvement of the web aggregator, integration of the aggregator and data scraper with the LLM, implementation of categorical tagging, automatic scheduling, social media exports, integration of backend components on Azure, and automated testing. It also details the challenges faced, the decisions made, and how the team is staying on track to meet deadlines over the final month of the project.

## 2. Sprint Overviews

### 2.1. Sprint Durations

Sprint 4: February 24<sup>th</sup>, 2025 – March 17<sup>th</sup>, 2025

Sprint 5: March 18<sup>th</sup>, 2025 – March 23<sup>rd</sup>, 2025

### 2.2. Sprint Objectives

The primary objectives for Sprint 4 were:

- Improve the web aggregator to yield better results and additional features,
- Integrate the LLM on Azure,
- Implement a category feature to group similar articles and generated content,
- Implement an automatic scheduler to ease content generation burdens,
- Allow easy exportation of content to common social media platforms,
- Test individual components before integration to confirm functionality.

The primary objectives for Sprint 5 were:

- Integrate additional backend components on Azure, including the web aggregator and data scraper,
- Configure the database on Azure for registration, login, categories, and scheduling,
- Test additional individual components.

## 3. Contributions and Key Decisions

### 3.1. Web Aggregator Improvements

**Team Members Involved:** Thomas, Shijie.

**Objectives:**

- Add more features to the aggregator,
- Improve for future software,
- Make sure that the basic functionality is not lost during improvements.

**Decisions and Progress:**

- **Build the Aggregator System:** Add some additional functions to the system and ensure the operation of the basic functions.
- **Improve the Solution of Flask App:** Add a compatible multithreaded library for flask app and get compatibility with Azure and all web pages. Presently, this scheme has been preliminarily completed but has yet to be tested by multiple parties. It has not been tested with Azure yet.
- **Split the Original Flask App into Smaller Scripts:** Most of the code has been written, but it is still incomplete. Will complete the code as soon as possible in a relatively short time, of course, this solution is only used in the case of incompatible flask app.

**Impact on Project Timeline:**

- This aspect of improvement took us a lot of time, and the time span even reached 2 sprints. The reasons for this are complex. A big part is because of problems between SQL links. As a result, testing cannot be carried out. This may remain an issue until we have finish testing Azure's interfacing with various systems.

### 3.2. Azure Integration of the Web Aggregator, Data Scraper, and LLM

**Team Members Involved:** Jaden.

**Objectives:**

- Implement Deepseek-R1 on Azure,
- Implement the azure function to call the LLM and the other azure functions to handle the web aggregator and html scraper.

### Decisions and Progress:

- **Implementation:** Using AI Foundry, it was possible to implement a pretrained LLM and luckily our chosen model Deepseek-R1 was there to use. I was able to get this working after making azure use less computational resources for the model as this was a limit for our free tier subscription.
- **Azure Functions:** All functions were also integrated into azure correctly and are working as intended. This is fully connected with the website as of now. Only minor tweaks to the response message are needed still as the response JSON object contains JSON like text.

### Impact on Project Timeline:

- This integration/implementation will help us move along at a better pace. With this integration we can start finalizing how the generated posts work on the website and connected the generated posts with the database and dashboard.

## 3.3. Azure Configuration of the Database

**Team Members Involved:** Thomas.

### Objectives:

- The task was to set up an Azure database and link it to our Azure website to support key functions such as a login, registration, and eventually personalized user dashboards that store historical posts,
- Lay the groundwork for capturing user history,
- Enable the development of custom dashboards and additional data driven features.

### Decisions and Progress:

- **Type:** Decided on implementing a PHP- based Azure database. Resolved configuration issues by installing correct php\_sqlsrv.dll file.
- **Issue Resolution:** Worked with Jaden with hopes of fixing the connectivity issues.
- **Result:** Successfully connected the database to the website, verified that user registrations are saved, and created additional tables for articles and categories to support future project tasks.

### Impact on Project Timeline:

- The learning curve associated with Azure databases and resolving configuration issues resulted in delays early in the process.
- With the database now stable and integrated, subsequent project tasks are expected to progress more rapidly.

### 3.4. Additional Azure Subscriptions

**Team Members Involved:** Jaden, Nicholas, Thomas.

**Objectives:**

- Create and configure additional Azure subscriptions to ensure that the application will remain hosted on Azure within the allocated credits for the remainder of the project.

**Decisions and Progress:**

- **Separate Subscription for LLM:** To ensure that the project stays within the budget, it was decided that separate subscriptions would be used. This allows enough resources to be allocated for the LLM on a separate subscription while the main application and database are on a single subscription.
- **Subscription Period:** Each subscription lasts for 30 days with enough credits to host the components for this period. Just before the subscription ends, an additional subscription is configured by another group member. This has been repeated twice so far and will need to be repeated another time before the end of the project.

**Impact on Project Timeline:**

- While additional subscriptions are a minor inconvenience, they are required to stay within the budget for the project and ensure that the application remains hosted until the end of the project.

### 3.5. Content Categories Implementation

**Team Members Involved:** Dea.

**Objectives:**

- Add categories from a predefined list to all articles shows,
- Allow users to save the categories for future references.

**Decisions and Progress:**

- **Implementation:** Make changes to the generate script to adjust for categories and saving them, as well as adjusting the database script to set the categories for the articles with help from Thomas.

**Impact on Project Timeline:**

- Users can filter and save their favourite categories to add to the experience of using our site.

### 3.6. Automatic Scheduling of Content Generation Function

**Team Members Involved:** Dalton.

**Objectives:**

- To create a way to have content be automatically generated at a certain interval set by the user (i.e. if the user wants a new newsletter generated every week, it will create a new newsletter with more recent articles at the beginning of every week). The user would be notified about a new newsletter that is drafted, and the content would be based on the topics that the user has already shown interest in based on their profile and activity previously.

**Decisions and Progress:**

- **Restrictions/Changes:** Started with having the automated schedule options being displayed on the dashboard, but it didn't make sense to have it there. Switched focus to creating a profile section where the user could edit how frequently they wanted the newsletter to be generated, as well as edit other profile information.

**Impact on Project Timeline:**

- Obviously switching focus to a different area (profile section) made it so not as much progress could be made on the automated scheduler until the profile was editable.

### 3.7. Social Media Export Function

**Team Members Involved:** Nicholas.

**Objectives:**

- Allow the users of the platform to automatically export the generated social media content to various platforms,



- Prepopulate the content so the user does not have to copy and paste into the window,
- Ensure that the generated content meets the restrictions of each platform.

#### **Decisions and Progress:**

- **Tool Selection:** After researching and experimenting with Meta APIs and Developer Accounts, the process to obtain individual API keys was beyond the scope of the course. Alternative tools were researched, and it was reduced to two similar tools:
  - ShareThis,
  - Sharaholic.
- **Implementation Status:** ShareThis is currently integrated on the website, with the share buttons present and opening a “New Post” window on each platform when clicked. Available applications can be customized via the ShareThis portal, and currently only populates our website’s URL. This will be modified to include the generated content once the dashboard is fully integrated and configured.

#### **Impact on Project Timeline:**

- The base functionality of the social media export function is present on the site, which will save time in later sprints.
- Additional work will be required to prepopulate the content in the “New Post” window for each platform. This can be completed once the dashboard is fully integrated and configured.

### **3.8. Automated Component Testing**

**Team Members Involved:** Nicholas.

#### **Objectives:**

- Write automated test scripts using Pytest to test the functionality of the web aggregator and the data scraper,
- Test components individually to confirm functionality before integrating them together and eventually onto Azure.

#### **Decisions and Progress:**

- **Testing Framework:** Since the web aggregator and data scraper were implemented in Python, Pytest was used as the framework for these tests.

- **Scope of Tests:** Every method of the web aggregator and data scraper were tested in these scripts, confirming functionality of these features. They provide the expected results and are ready for integration.
- **Test Results:** The test scripts and results have been uploaded to GitHub for these components.

#### **Impact on Project Timeline:**

- Component testing before integration will save time in future sprints, as the functionality of each component has been confirmed and tested.
- Additional components will need to be tested once finalized, and unit testing will also need to occur in future sprints once integration is complete.

## 4. Challenges

### 4.1. Web Aggregator Improvements

During testing we encountered an issue where multiple users' historical search information would be in the same file. So, we decided to save the Json file used to record the history information to SQL for each user, through multithreading to ensure that the basic function of multi-user is not affected. At present, the solution we have adopted is to write several alternative solutions and finally choose the best one. These are mainly technical problems with new features.

### 4.2. Azure Integration of the Web Aggregator, Data Scraper, and LLM

The implementation of the LLM posed many difficulties mostly due to our free tier. I first tried to use Azure Machine Learning Studio and spent a good portion of time trying to get that working with Deekseek-R1. Eventually I decided to try another Azure AI service which worked a lot easier and came with much less difficulty. The integration of the functions in Azure Functions also came with a few challenges. The main problem was because of libraries not being recognized on Azure. This was due to the deployment process of the function and eventually I found out that I could only deploy through my IDE (with the way I set up the function app).

### 4.3. Azure Configuration of the Database

Initially, my unfamiliarity with Azure database architecture and configuration presented a steep learning curve. This challenge was further compounded by local development issues, as my computer lacked the essential PHP components required for a seamless connection

to Azure, leading to frustrating delays and numerous troubleshooting sessions with limited success. Ultimately, the experience displayed the value of problem solving and teamwork; with the help of my team member Jaden – whose computer had the necessary files – we were able to add the required PHP components and successfully connected to the Azure database.

#### 4.4. Automatic Scheduling of Content Generation Function

The main issue with the scheduler was making sure that there were appropriate checks done, such as that a user was logged in, to allow for the automatic scheduling to be adjusted. These have since been implemented and will not cause further issues during integration.

#### 4.5. Social Media Export Function

While implementing the buttons to share to social media platforms was not challenging, the challenge will be in the next sprint when the main components are integrated and the focus shifts to prepopulating the “New Post” window with the generated content and the article URL. There are various ways to do this, with some examples including meta tags, manual data overrides, and dynamic text selection. The best option will be confirmed and implemented to ensure the required functionality is met. Local testing has already begun to find the most efficient way to do this.

### 5. Jira Board Updates (Sprint 5 + Remaining Backlog)

<

Figure 1: The backlog for the in-progress Sprint 5. Items remain in progress until all subtasks are completed.

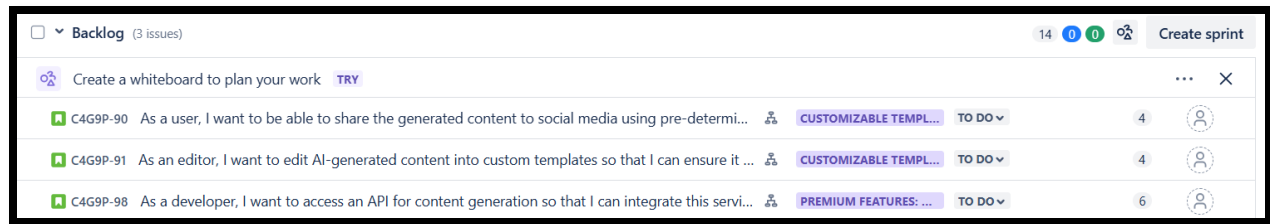


Figure 2: The remaining items in the product backlog for a future sprint. Additional items will be added to this at our next meeting with some housekeeping tasks (for proper documentation).

## 6. Overall Sprint Assessment and Next Steps

### 6.1. Summary of Sprint 4 Achievements

- Improved the web aggregator to yield better results and implemented additional features such as history, filters and different service search,
- Integrated the LLM on Azure,
- Implemented a category feature to group similar articles and generated content,
- Implemented an automatic scheduler to ease content generation burdens,
- Implemented exportation of content to common social media platforms,
- Tested individual components before integration into Azure.

### 6.2. Summary of Sprint 5 Achievements

- Integrated the web aggregator and data scraper with the LLM on Azure,
- Configured the database, allowing for successful login/registration.

### 6.3. Remaining Priorities for Next Sprints

- Integrate remaining components into Azure.
  - Remaining components:
    - Categories,
    - Scheduler,
    - Social Media Exports.
- Implement the ability to save generated content onto the dashboard,
- Improve the look and flow of the website.
- Complete the blank pages within the website.
  - Remaining Pages:
    - About Us,
    - FAQ.
- Further component and unit testing.

## 7. Conclusion

Sprint 4 was completed successfully, with significant progress with the web aggregator, category tagging system, automatic scheduler, Azure integration, social media exports, and component testing. The team has maintained the project timeline by making informed decisions and proactively addressing challenges and began focusing on the next sprint where the features from this sprint were improved and integration/configuration continued within Azure.

Sprint 5 was also completed successfully, with the short sprint focusing on improving the features implemented in the previous sprint, integrating additional components into Azure, configuring the database on Azure, and testing additional components and units. The next sprint will focus on further integration into Azure, dashboard configuration, website improvements, and further testing of the system. Regular updates will continue to ensure the project remains on schedule through the final month of the project.

## Appendix A. Website and GitHub Repository Links

Website Link: <https://group9website-gth5dkhfajb4d4g9.canadaeast-01.azurewebsites.net>

GitHub Repository Link: <https://github.com/JadenKBrock/4P02GroupProject>

## Appendix B. Meeting Minutes

### Meeting 10 – February 25, 2025

**Attendees:** Jaden, Nicholas, Thomas, Chidera, Shijie.

**Absent:** Dea, Dalton.

- Discussed the results of the third sprint, challenges encountered, and improvements for the next sprints.
- Discussed remaining tasks in the product backlog and divided them into the remaining sprints.
- Modified the product backlog to exclude features that are no longer part of the project.

### Meeting 11 – March 4, 2025

**Attendees:** Dea, Dalton, Jaden, Nicholas, Thomas, Chidera.

**Absent:** Shijie.

- Discussed setbacks during this current sprint and decided that additional work is required as tasks are behind schedule for this sprint.
- Discussed what we need to do going forward for this extended sprint and possible tasks for future sprints in general.

### Meeting 12 – March 11, 2025

**Attendees:** Dea, Dalton, Jaden, Nicholas, Thomas, Chidera, Shijie.

- Discussed how the remaining work for the sprint is going and any challenges associated with that.
- Talked about the next steps for the remainder of the sprint and what we wanted to accomplish before the March 18 meeting.

### Meeting 13 – March 18, 2025

**Attendees:** Dea, Dalton, Jaden, Nicholas, Thomas, Chidera, Shijie, Madeline.

- Presented the current state of our project to Madeline for feedback and verification.

## Appendix C. Complete Contributions Table

Team Member	Contribution
Jaden Kuhn	<p>Sprint 4/5:</p> <ul style="list-style-type: none"> <li>– Overhauled our website and connected the base files for the website. I changed all the HTML files to PHP and organized the file structure for our website. I also made it, so all our styling is the same and implemented a common theme among the pages.</li> <li>– Implemented the LLM on Azure using Azure AI Foundry.</li> <li>– Integrated our functions using Azure Functions for the web aggregator as well as the html scraper.</li> <li>– Created another function to call the LLM from an endpoint and get a response.</li> </ul>

	<ul style="list-style-type: none"> <li>– Connected all the functions with each other and integrated with the website so now generating a post works.</li> <li>– Helped Thomas briefly to be able to call the DB on azure</li> </ul>
Chidera Nwana	<p>Sprint 4/5:</p> <ul style="list-style-type: none"> <li>– Designed and created tables for login and registering to save user details for easy and efficient access into the website using saved credentials.</li> <li>– Created queries to be used to access information from the tables ensuring efficient access to any information needed.</li> </ul>
Dalton Morris	<p>Sprint 4/5:</p> <ul style="list-style-type: none"> <li>– Got core functionality sorted out for automated scheduler, very basic for now, allows the user to choose daily, weekly, monthly, or a custom interval to have a newsletter generated.</li> <li>– Created a profile dropdown for the user if they are logged in, if they are not logged in, it shows the login and register section still.</li> <li>– When the user is logged in, it shows the profile icon which is a dropdown where the user can click my profile to edit or logout, which takes the user to the login page.</li> </ul>
Dea Kukuqani	<p>Sprint 4/5:</p> <ul style="list-style-type: none"> <li>– Implemented an automated tagging system to categorize aggregated articles based on predefined keywords.</li> </ul>

	<ul style="list-style-type: none"><li>– Integrated a dashboard feature allowing users to manually assign categories to uncategorized articles.</li><li>– Established backend functionality to update and store category changes, enhancing content management efficiency.</li></ul>
Nicholas Caruso	<p>Sprint 4/5:</p> <ul style="list-style-type: none"><li>– Generated automated test scripts using Pytest to test the web aggregator and data scraper before integration.</li><li>– Hosted the second Azure subscription for the project and configured it with Jaden.</li><li>– Researched and experimented with Meta Developer Accounts and APIs.</li><li>– Implemented the core functionality of the ShareThis tool to allow for exporting to social media.</li></ul>
Thomas Semenak	<p>Sprint 4/5:</p> <ul style="list-style-type: none"><li>- Integrated the login and registration functionality into the app with enhanced styling, ensuring that users are seamlessly redirected to the dashboard homepage upon login.</li><li>– Created SQL database and connected necessary tables to support backend operations.</li><li>– Modified the data aggregator by compiling all links into a list instead of converting them to JSON, which simplifies and allows integration with components such as the web scraper or LLM</li></ul>



Shijie Tong	<p><b>Sprint 4/5:</b></p> <ul style="list-style-type: none"> <li>– all subsequent improvements, including all new features and testing of all related features and links.</li> <li>– Continue to fix the issue where aggregator cannot be used with multiple users.</li> <li>– Basic functional testing of aggregator.</li> <li>– Improvements were researched and several alternate versions were written for backup.</li> <li>– Redesign and improve the database structure.</li> </ul>
-------------	---

## Appendix D. Jira Sprint Burndown Charts and Backlog

### Sprint 4:



Figure 3: Burndown Chart for Sprint 4.

Note: The user stories for this sprint contained many subtasks, and additional user stories were added in the middle of the sprint to better divide them into achievable goals for a

single sprint. Jira does not reflect the closing of subtasks, and therefore those user stories were not closed until the end of the sprint.

### Sprint 5:

Note: This sprint is still in progress, so a burndown chart is not attached. The Jira board status can be found in a [Section 5](#), where tasks remain in progress until the end of the sprint (or upon completion).

## Appendix E. GitHub Screenshots and Contributions

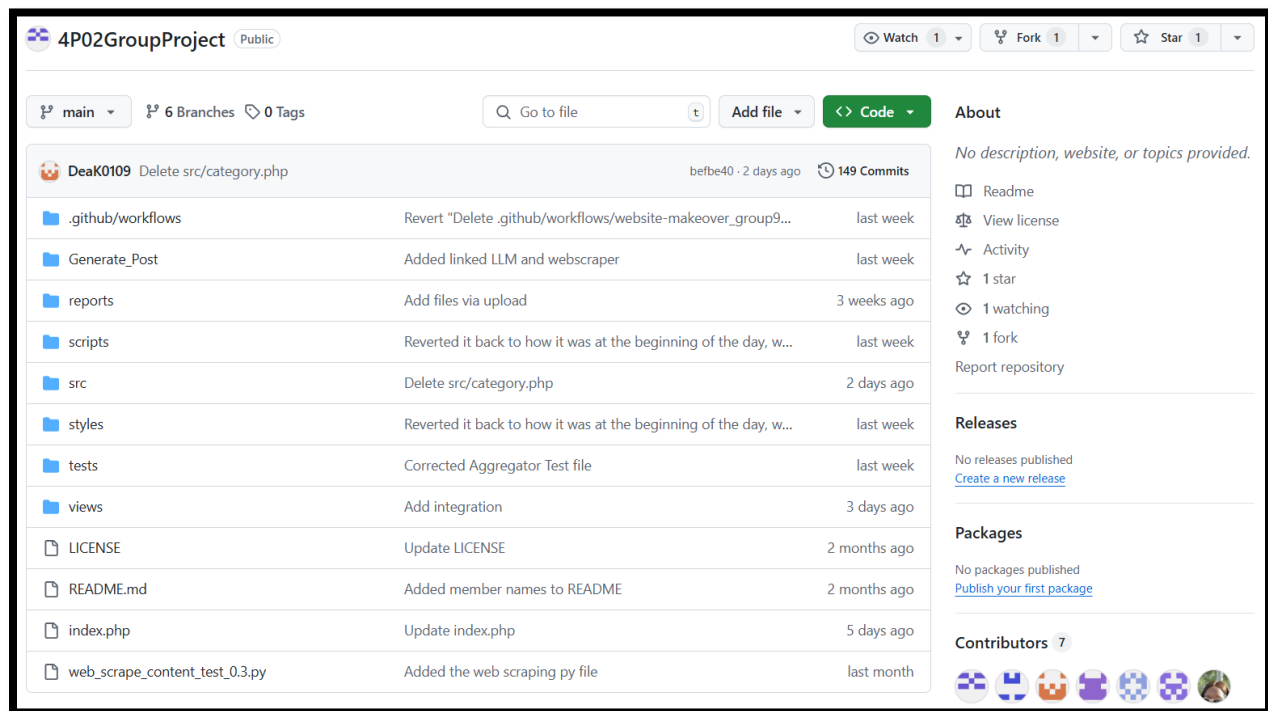


Figure 4: The Main Branch of our GitHub Repository.

4P02GroupProject / src /

DeaK0109 Delete src/category.php befbe40 · 2 days ago [History](#)

Name	Last commit message	Last commit date
..		
Dashboard	Reverted back to what was there at the beginning of the day, just wan...	last week
Generate	Changed generate testing page	2 days ago
Login	Fixed base url error (hopefully)	2 weeks ago
Register	Fixed base url error (hopefully)	2 weeks ago
a2	Merge remote-tracking branch 'origin/main'	3 days ago
aggregator	Updated searching	last month
assets	add background image.(test)	2 months ago
testing	this is a new a2 can work with mutple users. but haven't fully teste...	3 days ago
LLM testing	Update LLM testing	2 weeks ago
Tagging for posts.docx	Explanation + pseudocode for Tagging	2 days ago
main	Create src folder and main doc	2 months ago

Figure 5: The Source Folder of our GitHub Repository.

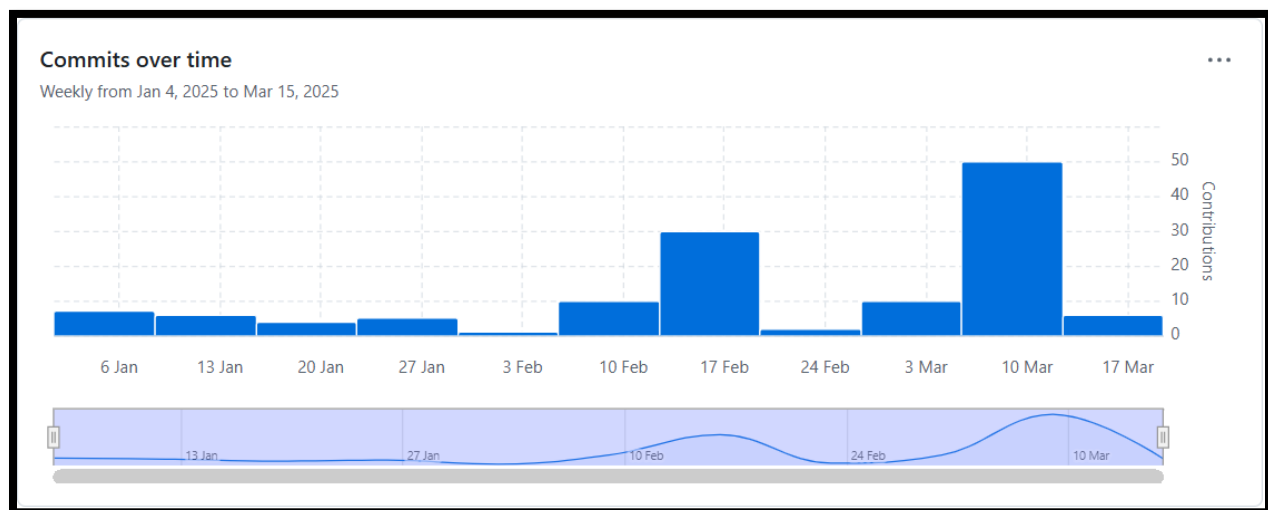


Figure 6: The "Commits Over Time" Graph.

## Appendix F. Website Screenshots

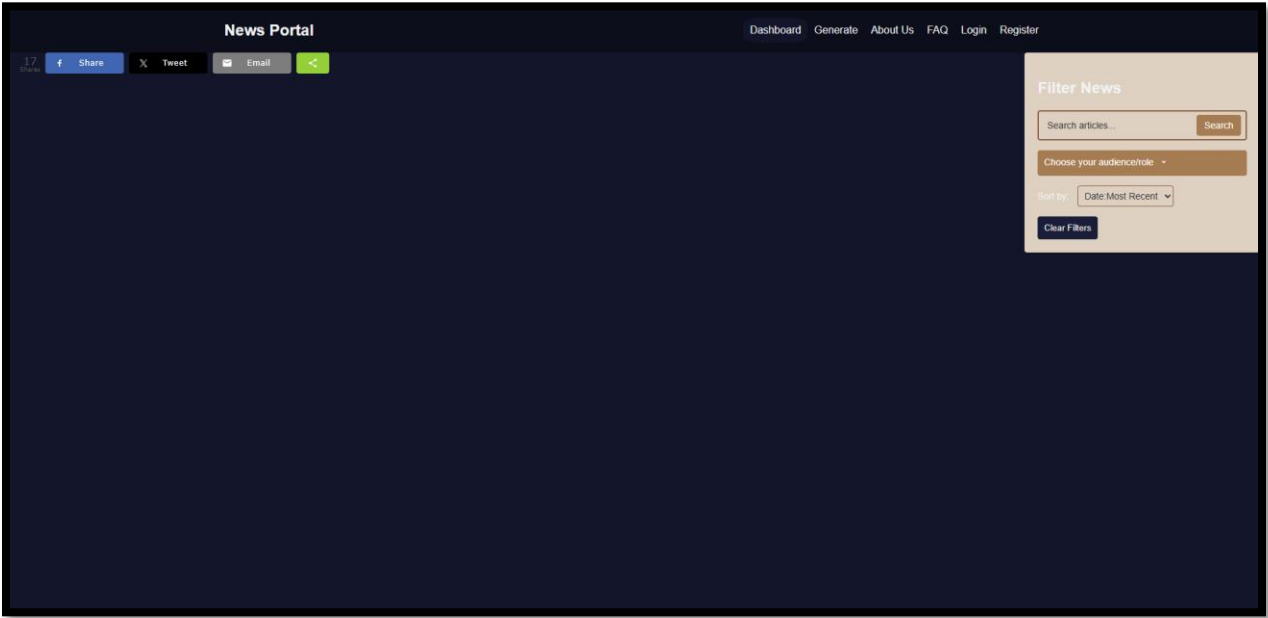


Figure 7: A screenshot of the main dashboard of our application. Social media buttons are present, and the header consists of the pages available within the application. Further work on this page will continue in the next sprint.

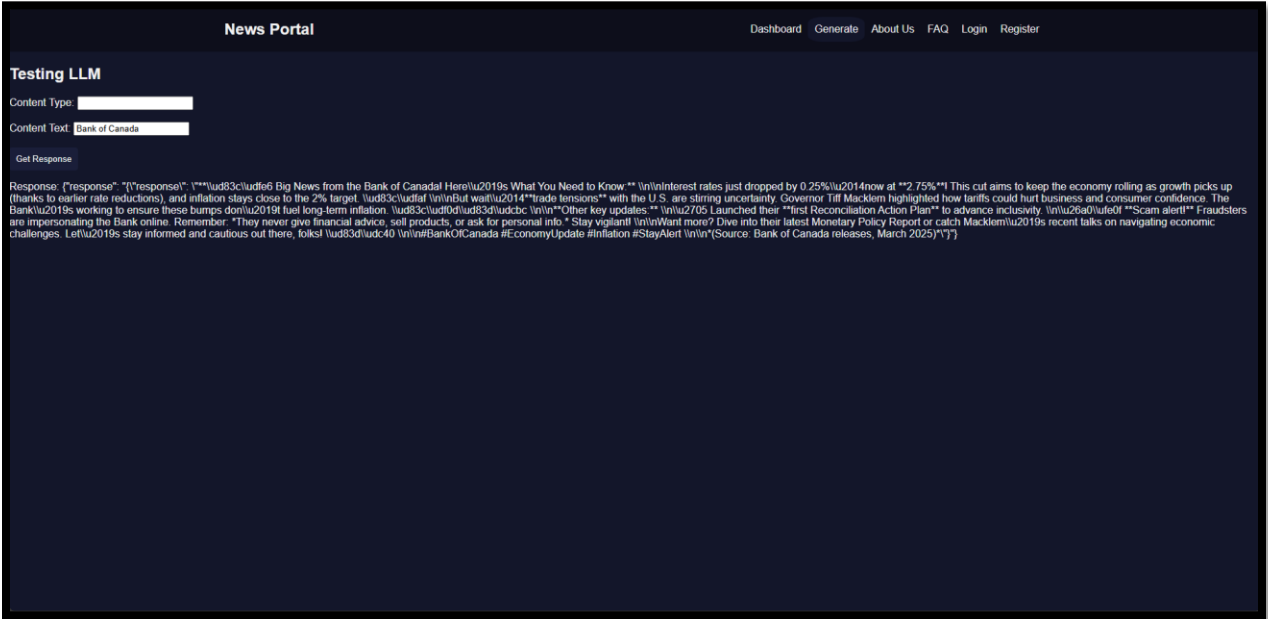
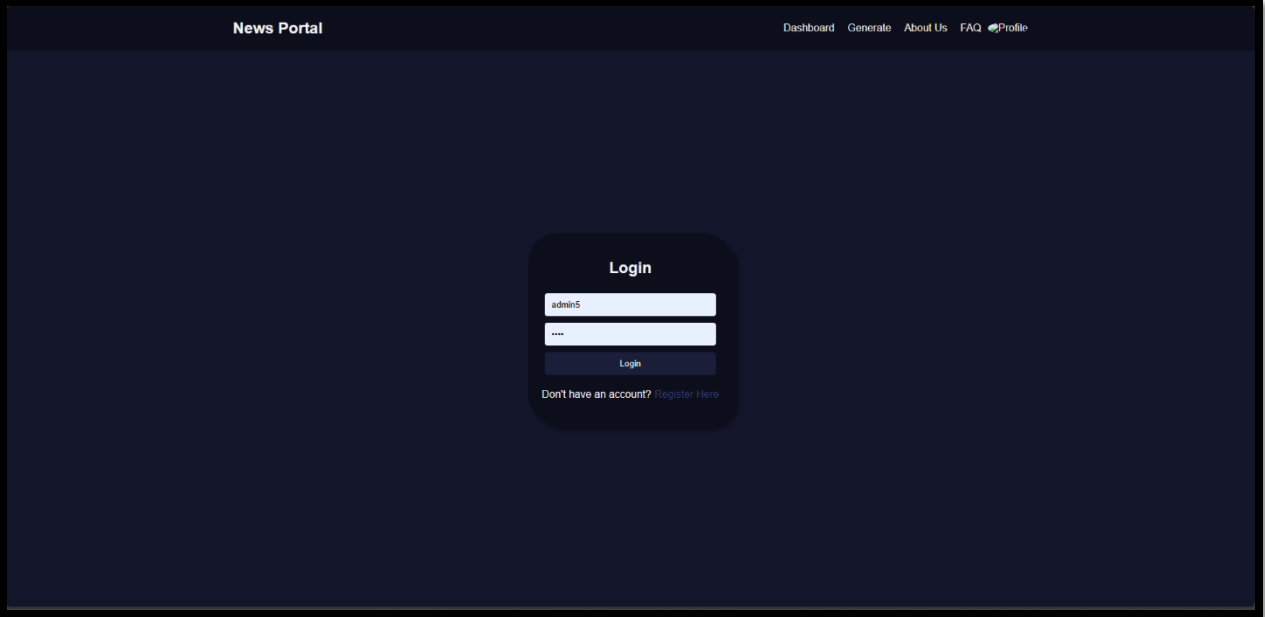
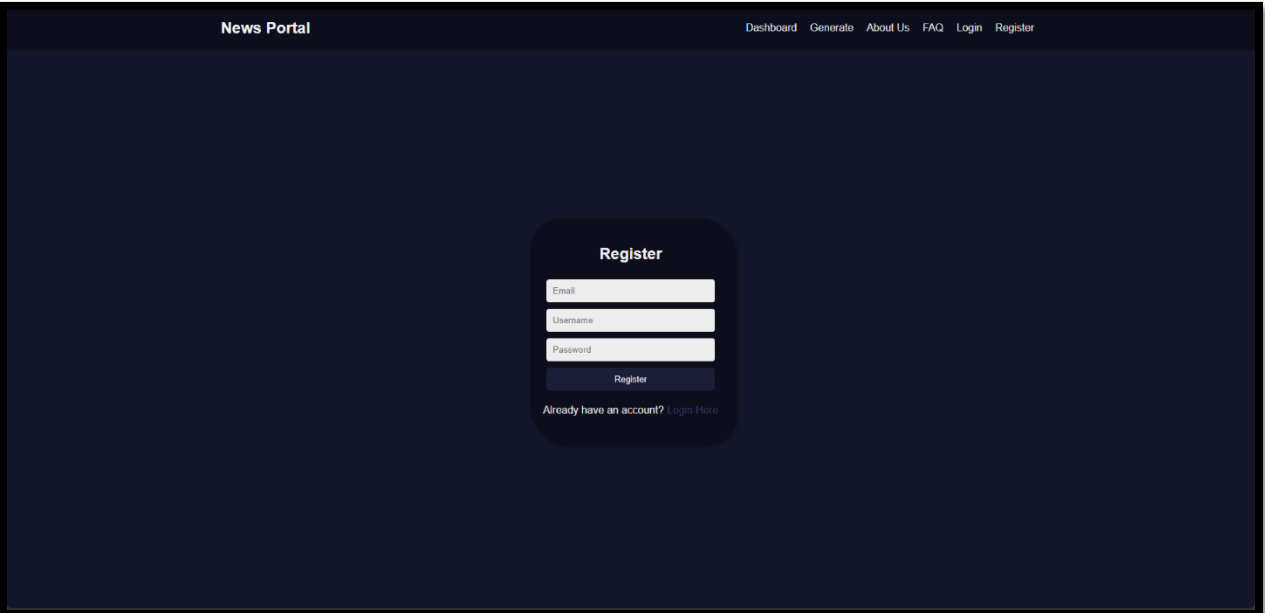


Figure 8: A screenshot of the Generate page, which allows the user to enter their content type and their keyword, and it will generate a social media caption from the LLM. The response will be cleaned up before the end of the current sprint.



The screenshot shows a web application titled "News Portal" in the top left corner. The top right navigation bar contains links for "Dashboard", "Generate", "About Us", "FAQ", and a "Profile" icon. The main content area is dark blue and features a central "Login" form. The form has a title "Login", a username input field containing "admin5", a password input field with masked characters, and a "Login" button. Below the form, there is a link that says "Don't have an account? Register Here".

Figure 9: A screenshot of the login page. This is fully functional.



The screenshot shows the same "News Portal" web application, but with the "Register" form displayed in the center. The navigation bar now includes a "Register" link. The "Register" form has a title "Register", and three input fields for "Email", "Username", and "Password". Below these fields is a "Register" button. At the bottom of the form, there is a link that says "Already have an account? Login Here".

Figure 10: A screenshot of the register page. This is also fully functional.