Sprint 3 Report

🥐Team Scrum-ptious🥐

What functionality does the system have at the end of this sprint? List user stories that you successfully implemented during this sprint (5 pts)

5. As a back-end developer, I want to ensure that APIs align with the front-end team's requirements so that the data delivered is in a format that can be seamlessly integrated into the UI, ensuring a consistent and efficient user experience.

- Size: 5
- Priority: High
- Precondition: HTTP API request endpoints are already defined
- Postcondition: API requests are functional for the front end

6. As a front-end developer, I want an API that retrieves both the original and translated versions of an article so that the front end can render and display this content.

- Size: 5
- Priority: High
- Precondition: Both original article content from the source language and the target language are available from the back end
- Postcondition: Both the original and target articles show comparisons in the front end.

7. As a back-end developer, I want consistent response structures across all endpoints so that consuming services can reliably use the data.

- Size: 5
- Priority: High
- Precondition: All API endpoints have been implemented.
- Postcondition: Every API endpoint returns responses in a standardized JSON format as defined in the API specifications.

9. As a back-end developer, I want logs of incoming requests and outgoing responses so that I can debug issues and monitor API behavior.

- Size: 3
- Priority: Low
- Precondition: API endpoints are live and handling traffic

- Postcondition: API requests and responses are recorded in a centralized logging system.

10. As a back-end developer, I want to parse and validate incoming URLs so that I can reduce invalid requests and decrease server load.

- Size: 5
- Priority: Low
- Precondition: API makes a GET request from url; crashes upon failure
- Postcondition: API makes a GET request after validating URL or gracefully fails with error handling (if URL is formatted “xx.wikipedia.org/wiki/xxx”, else, the request fails instantly)

Key functionalities Implemented:

User Story 5: The comparison endpoint is implemented to deliver a JSON response schema utilizing the LABSE model from the LLM, including missing info and extra info fields. The schema aligns with the front end team’s requirement, ensuring that the comparison lists could be seamlessly integrated into the UI.

User Story 7: Consolidated translate and get article endpoint into one “wiki/articles” endpoint, better adhering to DRY principles given the overlap between previous endpoints.

User Story 9: Error handling is now standardized: specific HTTP error codes (like 400 and 404) are returned with informative messages for client-side issues, and when the application is in debug mode, detailed stack traces are included to aid developers in identifying the root cause of problems.

User story 10: Security and server load management are optimized by parsing the URL input and validating it in pieces, as well as handling invalid inputs by providing error messages for detected problems. A generic HTTP 500 error is thrown for complete failure to fetch an article.

Did you end up making any changes to any of these user stories? Did you break down any further user stories? Did you identify any new user stories during this sprint, and if so, did you add them to the product backlog or decide to implement them right away? Explain (5 pts)

User Story 10 *(As a back-end developer, I want our configured middleware to be exceptionally secure and have selective origin parameters.)*: This story became obsolete after implementation. Since the app runs under one domain, CORSmiddleware isn't needed for now. Security and selectiveness shifted toward input validation instead.

NEW STORIES:

16. As a developer, I want to document my knowledge so that people who inherit my code can understand the code structure and my decisions.

- Size: 13
- Priority: High
- Precondition: We have implemented all planned features and have access to prior research
- Postcondition: Future developers will be able to interpret our work and not repeat the research/implementation we have accomplished

17. As a developer, I want my endpoints to accommodate the most current schema design to ensure data consistency.

- Size: 5
- Priority: High
- Precondition: We have implemented an endpoint which does follow updated schema requirements
- Postcondition: All endpoints return responses which are expected by agreed upon schemas

18. As a back-end, I want to consolidate my work along with my teammates in a single branch and make a pull request to main

- Size: 5
- Priority: VERY High
- Precondition: Team has all of their changes in separate branches
- Postcondition: Team has submitted all their changes in a pull request under one branch

CHANGED STORIES:

10. As a back-end developer, I want our configured middleware to be exceptionally secure and have selective origin parameters.

- Size: 2
- Priority: Medium
- Precondition: Rules exist for middleware specifications (CORS handling, API key management, etc.).
- Postcondition: The APIs are secured to only allow certain domains, types of requests, and authorized users.

CHANGED TO

10. As a back-end developer, I want to parse and validate incoming URLs so that I can reduce invalid requests and decrease server load.

- Size: 5
- Priority: Low
- Precondition: API makes a GET request from URL; crashes upon failure
- Postcondition:

Reason: CORS handles resource sharing across domains, like our separate front-end and back-end. It's mainly needed in production once domains are set. While CORS has some security traits, input validation and URL parsing were more critical. CORS is implemented but unused. This story extends User Story 8 by selectively validating URL syntax before creating a Wikipedia instance.

9. As a back-end developer, I want logs of incoming requests and outgoing responses so that I can debug issues and monitor API behavior.

- Size: 3
- Priority: Low
- Precondition: API endpoints are live and handling traffic
- Postcondition: API requests and responses are recorded in a centralized logging system.

CHANGED TO

9. As a back-end developer, I want to be able to see the stack of function calls for each error message in order to make endpoint development more efficient

- Size: 3
- Priority: Medium
- Precondition: API endpoints are being developed
- Postcondition: API endpoints are easier to develop with detailed error messages for developers

DELETED STORIES

This user story is not completable within our volunteering, so we decided to change this to documentation

4. As a front-end developer, I want the target language article to be returned in the same language as the source article so that the user can understand the differences

- Size: 13
- Priority: Medium
- Precondition: MVP is fully functional; comparisons can be made between two articles from Wikipedia in different languages
- Postcondition: The ML can translate Wikipedia articles to the user's desired language, and the comparisons will still be accurate and displayed to the front end.

What are the "lessons learned" at the end of this sprint? What would you do differently next time? Explain (5 pts)

- Documentation and misunderstanding
  - The usage of CORS as a security measure was a monstrously stupid failure point. Understanding how it truly works and what makes it useful, despite it being obsolete in this current implementation, made all the difference.
  - What to do next time: Zoom out a bit to try and put more time and effort into other security measures. Staying stuck to one aspect at a time has its benefits, but at the sacrifice of forward progress in other departments.

- Development standards
    - After witnessing another team merge some beginner mistakes on the main branch of the project, we notified our tech lead about these merges, and if we could revert the main branch to a previous state to clean up the commit history. Upon inquiring, our tech lead told us that NO ONE should be resolving their own pull requests (which we have been doing), which revealed to us important industry standards in terms of version control, and dangers when they are not adhered to.
- Version Control
    - This was the first sprint where we had clearly defined separate branches, and were all contributing to the project in different remote workspaces. Learning how to view each other's work, give opinions, and test out each other's code is extremely valuable for the productivity of not only our team, but other teams' work that we had to glance at to ensure our implementation were up to date.

Provide an updated numbered list of all user stories yet to be implemented; indicate pre- and post-conditions (5 pts)
Remaining User Stories:

16. As a developer, I want to document my knowledge so that people who inherit my code can understand the code structure and my decisions.

- Size: 13
- Priority: High
- Precondition: We have implemented all planned features and have access to prior research
- Postcondition: Future developers will be able to interpret our work and not repeat the research/implementation we have accomplished

17. As a developer, I want my endpoints to accommodate the most current schema design to ensure data consistency.

- Size: 5
- Priority: High
- Precondition: We have implemented an endpoint which does follow updated schema requirements

- Postcondition: All endpoints return responses which are expected by agreed upon schemas

18. As a back-end, I want to consolidate my work along with my teammates in a single branch and make a pull request to main

- Size: 5
- Priority: VERY High
- Precondition: Team has all of their changes in separate branches
- Postcondition: Team has submitted all their changes in a pull request under one branch

Given the current functionality of the system and taking into account the pre- and post-conditions, identify a subset of user stories to be implemented during the next sprint. Be sure that the cumulative size of the selected user stories is about 1/4 of the size of the full backlog. Describe the functionality that your (partially implemented) system will have at the end of this sprint. (5 pts)

Sprint 4:

16. As a developer, I want to document my knowledge so that people who inherit my code can understand the code structure and my decisions.

- Size: 13
- Priority: High
- Precondition: We have implemented all planned features and have access to prior research
- Postcondition: Future developers will be able to interpret our work and not repeat the research/implementation we have accomplished

17. As a developer, I want my endpoints to accommodate the most current schema design to ensure data consistency.

- Size: 5
- Priority: High
- Precondition: We have implemented an endpoint which does follow updated schema requirements
- Postcondition: All endpoints return responses which are expected by agreed upon schemas

18. As a back-end, I want to consolidate my work along with my teammates in a single branch and make a pull request to main

- Size: 5
- Priority: VERY High
- Precondition: Team has all of their changes in separate branches
- Postcondition: Team has submitted all their changes in a pull request under one branch

Description:

In Sprint 4 we will focus on wrapping up our work and having a clean hand off to future teams. We will focus on documentation, as well as sorting out merge conflicts in all of our work in order to make sure that our work is visible on the main branch. In terms of features we will simply be updating our currently configured endpoint in order to be up to date with our most recent schema design.

Points: 23 / 77