Product Backlog

🥐Team Scrum-ptious🥐

Updated User Stories

1. As a front-end developer, I want to be able to retrieve articles from multiple Wikipedia languages via API, so that I can obtain data for translation comparisons.
    - Size: 3
    - Priority: High
    - Precondition: The requested language is supported by the Wikipedia API
    - Post condition: The article content is visible to user in a specific language
2. As a front-end developer, I want to request translations for the page I am displaying to the end users, so that they can view alternate versions of a page that they can understand.
    - Size: 3
    - Priority: Medium
    - Precondition: Articles in other languages have been fetched from Wikipedia
    - Postcondition: The translated article is available

Epic: As a front-end developer, I want to request content suggestions and semantic comparisons via the API from the LM so that I can present users with a comparison view of the original and translated content.

3. As a back-end developer, I want to configure middleware for API access so that I can ensure secure access to data.
    - Size: 8
    - 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.
4. As a back-end developer, I want to store content from both Wikipedia and the LM, so that I can perform comparisons between the original article and the LM-generated content to identify semantic differences.
    - Size: 5
    - Priority: Medium
    - Precondition: Wikipedia articles and LM translations have been successfully retrieved
    - Postcondition: The semantic comparison results indicate whether the translation is valid.
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: 3
- Priority: High
- Precondition: HTTP API request endpoints are already defined
- Postcondition: API requests are functional for the front-end

Epic: As a backend developer, I want an API so that I can communicate with other components of the system consistently and reliably.

6. As a back-end developer, I want an API that retrieves both the original and LM-translated versions of an article so that other developers can request content for comparison.
    - Size: 5
    - Priority: Medium
    - Precondition: Both original article content from Wikipedia and the translated content from the LM are available
    - Postcondition: Both the original and LM-translated articles are usable by front-end developers.
7. As a back-end developer, I want consistent response structures across all endpoints so that consuming services can reliably use the data.
    - Size: 3
    - 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.
8. As a back-end developer, I want error-handling mechanisms that return meaningful error responses so that consuming services can gracefully handle failures.
    - Size: 5
    - Priority: High
    - Precondition: API endpoints are in place and operational.
    - Postcondition: Each endpoint returns standardized error responses (with appropriate HTTP status codes and descriptive error messages) when failures occur.
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: Medium
    - Precondition: API endpoints are live and handling traffic
    - Postcondition: API requests and responses are recorded in a centralized logging system.

Total size: 38

Updated Glossary

- **Wikipedia:** A free online encyclopedia where people can read and edit articles in many languages. In this project, Wikipedia is used as the main source for retrieving articles for translation and comparison.
- **UI – User Interface:** An interactive graphical representation of the program and data that allows users to interact with the system.
- **LM – Language Model:** A machine-learning model using algorithms to process human language to generate translations, analyze content, and provide semantic comparisons.
- **API – Application Programming Interface:** A collection of method definitions available for usage by modules of the program.
- **API Endpoints:** URLs where the API receives requests to handle.
- **Middleware:** A component that sits between different parts of a software system, controlling how data flows between them.
- **Front-end developer:** A developer working in the UI app of the Symmetry repository responsible for integrating API responses and designing user interactions
- **Backend developer:** A developer working on the Wikipedia APIs and LM APIs, responsible for handling data retrieval, processing, storage, and security.
- **End users:** People who interact with the UI application to request translations, compare content, and access Wikipedia data through the system.
- **CORS - Cross-Origin Resource Sharing:** Rules to ensure that only specific domains can access an API, preventing unauthorized web requests
- **API keys:** Unique credentials used to authenticate and control access to an API, ensuring that only authorized clients can send requests.
- **HTTP requests:** Communication method used in front-end application to communicate with backend APIs
    - **POST:** HTTP request that has a request JSON body for handling complex data that can't be sent in query parameters
    - **GET:** HTTP request for retrieving data using query parameters used for getting data from Wikipedia API
- **Body parameters:** Parameters included in HTTP request, typically in JSON format
- **Query parameters:** Parameters included in HTTP request, typically within the URL

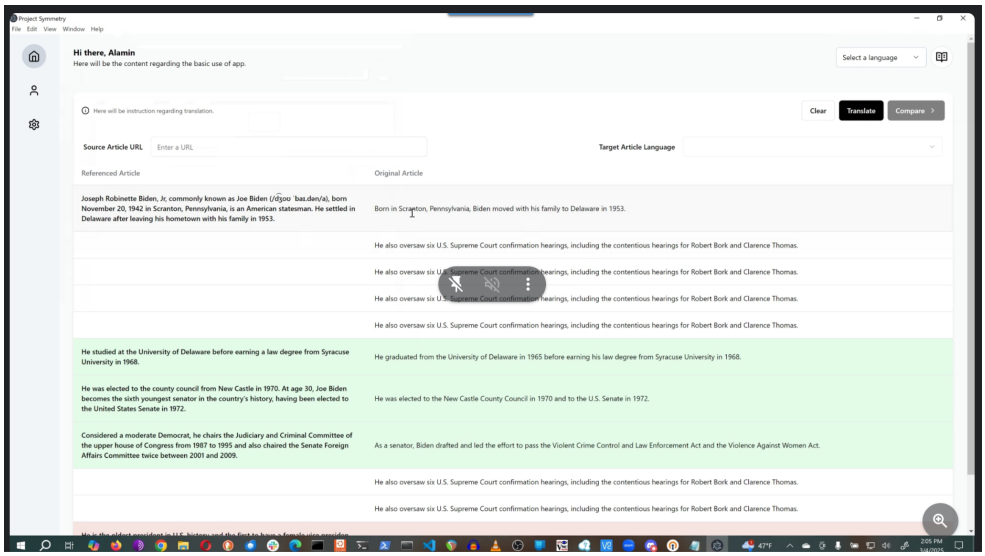
Sprint 1 – Selected User Stories

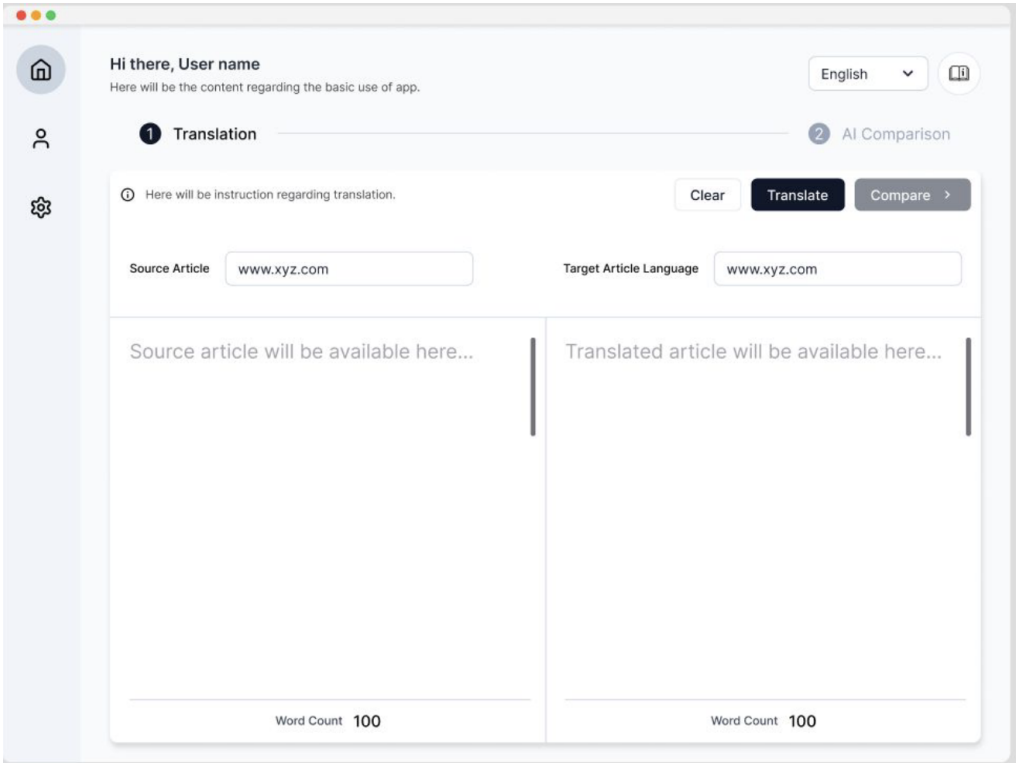
1. User Story 1: As a front-end developer, I want to be able to retrieve articles from multiple Wikipedia languages via API, so that I can obtain data for translation comparisons. - Size: 3

2. User Story 2: As a front end developer, I want to request translations for the page I am displaying to the end users, so that they can view alternate versions of a page that they can understand. - Size: 3
3. User Story 3.1: As a backend developer, I want to configure middleware for API access, so that I can ensure secure access to data. - Size 8

   Sprint 1 total: 14

Provide sketches for your design:




Hi there, User name
Here will be the content regarding the basic use of app.
English
1 Translation
2 AI Comparison
Here will be instruction regarding translation.
Clear
Translate
Compare
Source Article
www.xyz.com
Target Article Language
www.xyz.com
Source article will be available here...
Translated article will be available here...
Word Count 100
Word Count 100