Global News Project Report

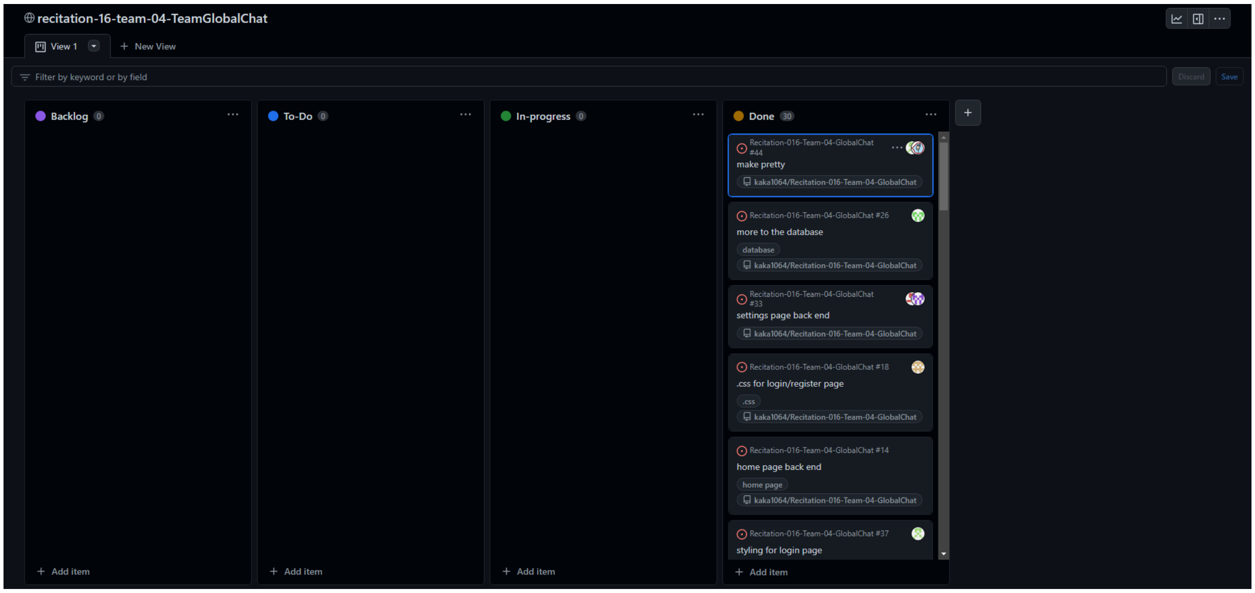
Team members:
Andrew Pleeter, Catrina Smith, Joseph Westley, Kailash Kalaiselvan, Peter Magro, TJ Jablonski

Project description:

People around the world rely on the internet for news, as well as to communicate with others. In this project, we created Global News, a website which removes the language barrier across countries by allowing people to post, view, and update news posts in any language they choose, and translate other people's messages into their preferred language at the click of a button. The completed website contains login, register, home, news, and profile pages. Once a user has selected their preferred language, they can click the "Translate" button next to any post, and a message will be displayed containing the post translated into their preferred language.

Project tracking board:

https://github.com/users/kaka1064/projects/1



Recitation-016-Team-04-GlobalChat #44
make pretty
kaka1064/Recitation-016-Team-04-GlobalChat

Recitation-016-Team-04-GlobalChat #26
more to the database
database
kaka1064/Recitation-016-Team-04-GlobalChat

Recitation-016-Team-04-GlobalChat #33
settings page back end
kaka1064/Recitation-016-Team-04-GlobalChat

Recitation-016-Team-04-GlobalChat #18
.css for login/register page
.css
kaka1064/Recitation-016-Team-04-GlobalChat

Recitation-016-Team-04-GlobalChat #14
home page back end
home page
kaka1064/Recitation-016-Team-04-GlobalChat

Recitation-016-Team-04-GlobalChat #37
styling for login page

Demonstration video:

https://drive.google.com/file/d/1LYE-KTqRA0MfjV9rIGQPuUqylra5zSxR/view?usp=sharing

VCS:

https://github.com/kaka1064/Recitation-016-Team-04-GlobalChat

Contributors:

TJ

I first worked on the front end for the news page, where I implemented the table that displays the author, post, topic and “translate” button. I also coded the GET routine to get the values for author, post, and topic. Next, I fully styled the login/register page to have a more user-friendly interface and clean up the look. Lastly, I implemented the edit button for the profile page, both front and back end, as well as styled the actions column to look better. I also changed the post input to textarea to allow for easier editing when posting/editing.

Kailash

For this application, I developed the login and register page before it got modified and styled to fit our product style overall. I also fixed the issue with the authentication for the pages to have the user logged out when the URL is loaded on a certain page. Then I made a dropdown to nest the settings and logout page which was scraped for our final product. Lastly I developed the profiles page for the project to display the individual users on the profiles page in chronological order to simplify and view posts that a user posted to make it easier to view their own posts and messages.

Peter

For this application, I handled the styling of the website, including the base CSS, homepage layout, and logo. I also performed initial research and testing on the DeepL API, creating a set of Postman requests which were used as the basis for the server API calls. I also added cards to the homepage to display the 5 most recent posts, and implemented the backend necessary to retrieve the posts. Finally, I implemented the “Delete” function on the Profile page.

Andrew

For this application, I handled the settings page, there were a bit of issues regarding this page. However, at the time of presentation, a github push existed but was not applied to the azure cloud. I implemented multiple get and post api’s, such as updating user passwords, changing language preferences, and modifying the first/last name for the current user. In addition to that, I helped implement the skeleton code for the initial project, as it was imperative that we organized each part of the project for each person. The settings page does in fact work but was not applied to main branch at the time of presentation, this was because of github issues with pulling branches.

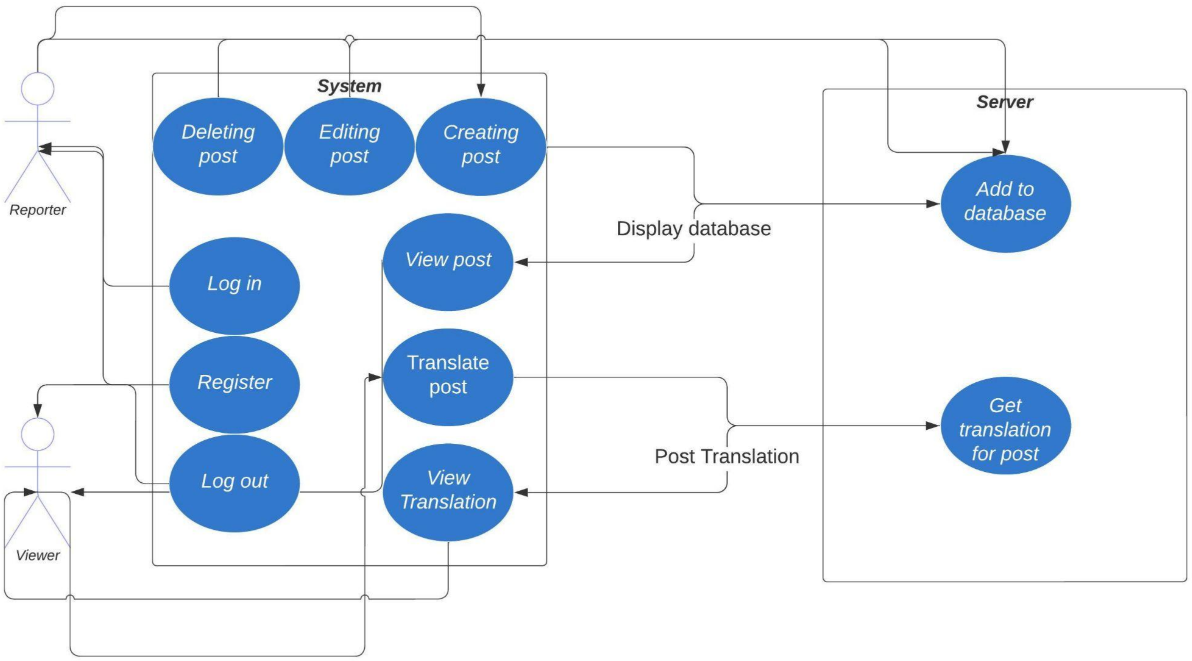
Joseph

For this application my two main responsibilities were handling the front end and back end for the language translations with their use with the DeepL API key and the front end and back end of creating a new post. Furthermore, I aided in the initial set up of the application managing many of the gets that render in pages. I created the database tables as well as the inserts into said tables. I created the test cases that were tested during startup during the lab 11 timeline of the project. Finally, i aided in the creation of the table posted in the front end of the news page by working on the backend and front end for it.

Catrina

カトリーナから何もない．私たちは知りません。

Use Case Diagram:



Test results from lab 11:

Unit Tests:

- “Welcome” test: PASS
- Login test

        - Positive: PASS
        - Negative: PASS
    - Register test
        - Positive: PASS
        - Negative: PASS

Acceptance tests:

- Login feature
    - Log in upon entering a correct user/pass combo: PASS
    - Do not log in upon entering an incorrect user/pass combo: PASS
- Settings feature
    - FAIL: The “Settings” page was not implemented
- Messaging/Posting:
    - Users may create new posts: PASS
    - Posts can be translated into a user’s preferred language: PASS

User Acceptance Testing plans from lab 11:

1. Feature - Login **(Feature finished by project deadline)**
    a. Description: As a user i should be on the login page and see two visible fields, a username and password field. If I enter the correct username and password I should see the home page be displayed to me, and if I enter an incorrect username or password I should see a message letting me know of my errors.
    b. Test Data: the test data that will be used to test the feature is a username and a password.
    c. Test environment: the environment that will be used to test the feature at the moment is the development environment, To where we're going to be using our own local machine.
    d. Test results: there are two results to test the feature, a successful result and a failed result. If the result is successful then the results should be a home page that welcomes the user. And if the results are a failure, then the results should be a message that tells the user that there was an error that either keeps the login page or renders the register page
    e. User acceptance testers: the testers depend on the stage of development, if the feature is in the early stages of development then the testers will be the team members and if the feature is finished the testers are end users.
2. Feature - Settings **(Feature was in final stages by project deadline, still tested)**
    a. Description: As a user I should be able to access settings from the navbar at the top right corner and on the settings page I should be able to view and edit my information and change my language preference.
    b. Test Data: the test data that will be used to test the feature is the preferred language

    c. Test Environment: the environment that will be used to test the feature at the moment is the development environment, To where we're going to be using our own local machine.
    d. Test Results: the test results that will be used to test this feature is the database, if the database gets updated based off of what the user changes then that is a successful result, and if the database does not get updated after the user makes changes then that is a failed result.
    e. User acceptance testers: the testers depend on the stage of development, if the feature is in the early stages of development then the testers will be the team members and if the feature is finished the testers are end users.

3. Feature - Sending messages **(Feature was thrown away, TA's suggestion, did news instead)**
    a. Description: As a user I should be able to go to the chatbox page and send messages in my preferred language.
    b. Test Data: the test data that will be used to test the feature will be messages sent within the chat box
    c. Test Environment: the environment that will be used to test the feature at the moment is the development environment, To where we're going to be using our own local machine.
    d. Test Results: the test results that will be used to test this feature will be the translation of messages. If the user sends a message in their preferred language and the messages get translated then the results are a success, and if the messages sent by the user don't get translated then the results are a failure.
    e. User acceptance testers: the testers depend on the stage of development, if the feature is in the early stages of development then the testers will be the team members and if the feature is finished the testers are end users.

Observations:

1. For test plan 1 of our user acceptance testing plans, we finished this feature and was able to test with possible end users. We wanted to test our login feature but got to test the register feature as well since end users will need to make an account in the first place anyways. All end users were easily able to figure out what they needed to do for the login as they were able to read the labels for each input box. We believe their behavior is within our expectations. But from our findings we found out from one end user testing this feature that we could have displayed a different message for users so that they knew if their password or username was incorrect instead of 'invalid input', but since this was a small issue with the minority we didn't include it in our final project. Looking back at it, If given extra time we would have added this feature.
2. For test plan 2 of our user acceptance testing plans, we did not finish this feature by the project deadline, but we were in the final stages of it by the project deadline and decided to still test the feature with possible end users. During testing, we saw all users be capable of navigating to the settings page from any page with ease. Most user's when getting to the settings page, scanned the entire page before any other action, then only changed their password. The reasoning behind this is because "the other options are to the user's liking since it was made from the register page. This caused a deviation from

the expected actions since we expected users to change more than just their password. Furthermore, something we started to try and implement before the project deadline was a confirm password to prevent errors, because if a user made a typo unknowingly then they wouldn't be able to log back into their account during future attempts once they logout.

3. For test plan 3 of our user acceptance testing plans, we had to scratch this feature since our TA Nikita told us of the troubles that come from live chatting. Therefore, we changed our project to one of viewing and creating posts that could be translated. So instead of that testing plan that was created during lab 11 we tested two features instead. Creating and deleting a post. With these features we wanted to test whether or not an end user can successfully create a post and then delete it. During testing, all end users navigated to the news page and were easily able to find the button to create a new post. I believe this was within our expectations and this behavior was consistent throughout all testing. Unfortunately, the same can't be said for deleting a post as there was a slight hesitation from most users when trying to delete the post as they expected the delete button to be close by, but after that slight hesitation they went to the profile page and saw the big red 'x' next to each post and instantly understood how to delete the post they created. So a deviation did exist but it wasn't a huge deviation from what we expected. So we didn't have to change anything to our application.

Deployment:

http://recitation-016-team-04.eastus.cloudapp.azure.com:3000/