Speakeasy

Who:

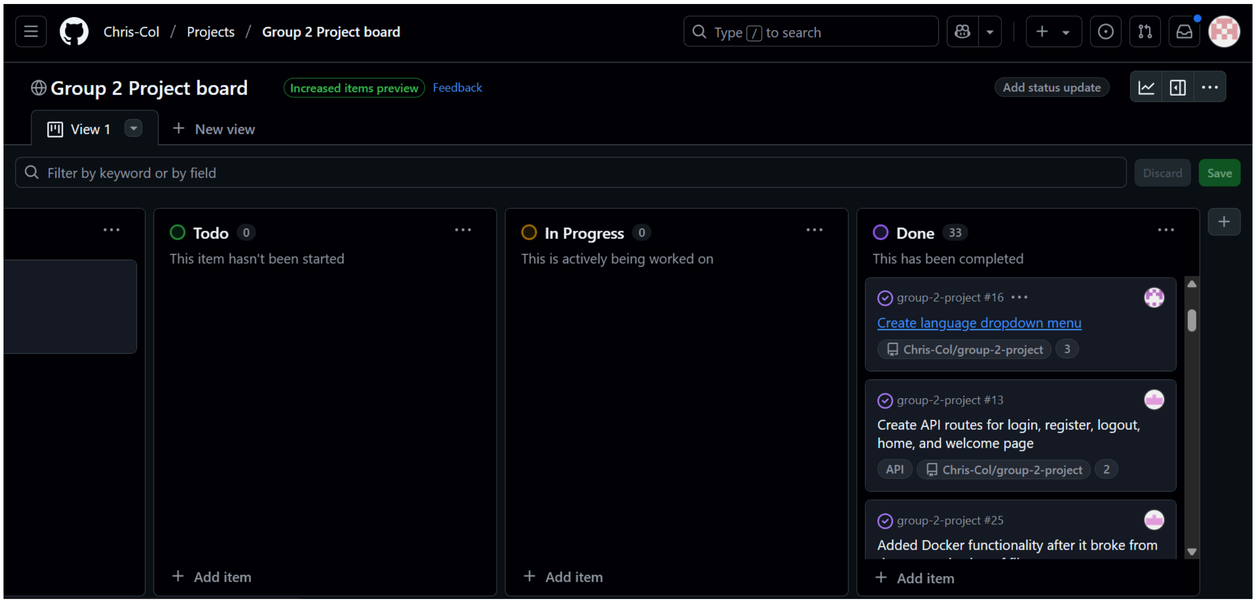
- William Skulic-Jordan/wisk4999/wisk4999@colorado.edu
- Christopher Coleman/Chris-Col/chco2067@colorado.edu
- Hunor Kovacs/hunorkov/huko2764@colorado.edu
- Isaiah Millington/IsaiahMillington/ismi2675@colorado.edu
- Patrick Markey/PatrickMarkey/pama9505@colorado.edu
- Jake Gao/Hao4851/jake.gao@colorado.edu

Project Description: A 200 word summary of the project

We developed an interactive application designed to teach users different languages through the use of simple, engaging minigames. The platform will feature a user-friendly interface that includes essential functionality such as user registration, secure login, and personalized profiles. Once logged in, users will have access to a homepage showcasing a variety of educational minigames. Each game will be designed to focus on a specific language skill, such as vocabulary building, sentence structure, or comprehension. Users will have the ability to select both the language they want to learn and the minigame they wish to play, making the experience flexible and user-driven. To support dynamic language learning across a wide range of languages, we will integrate the Google Translate API into the platform. This will allow us to provide real-time translations and ensure that users receive accurate and contextually appropriate assistance during gameplay. The use of the API will also enable us to easily expand our language offerings in the future without needing extensive manual updates. Our goal is to create an accessible, fun, and motivating environment where users can build language skills through consistent practice and interactive exercises. In future versions, we have ideas to add features such as progress tracking, achievements, and leaderboards to increase user engagement, provide feedback, and create a sense of community among learners.

Project Tracker - GitHub project board:

Project Board link: https://github.com/users/Chris-Col/projects/2



group-2-project #16
Create language dropdown menu
Chris-Col/group-2-project 3

group-2-project #13
Create API routes for login, register, logout, home, and welcome page
API Chris-Col/group-2-project 2

group-2-project #25
Added Docker functionality after it broke from

The card starts in “no status” which serves as a section for potential and future enhancements to our application. They are then promoted to the “Todo” section once we find the timing is correct and someone has the available time to work on it. Team members will drag the card to “In progress” once they have a branch set up and openly working on it. Upon merge and push as a PR, the card will be moved to the “Done” section.

Video:

5 minute or less video demonstrating your project. Your audience is a potential customer or person interested in using your product.

https://cuboulder.zoom.us/rec/share/f-807vBHu3YSzGcSoFVMLTJkQJdk1DohlIX0QqalQFRUYjIBI6HmDpjHRK_52WuI.mbemzJjhpArbdsja?startTime=1745875373000

Passcode: #m8@qq35

VCS:

https://github.com/Chris-Col/group-2-project

Contributions:

William Skulic-Jordan:

I created the handlebars partials and the pages for login, register, displaying the games, and the 4 games. This involved writing HTML for these files and ultimately embedding the games that we have. I built the corresponding API routes for our handlebars pages so that the navigation bar works, and clicking buttons takes you to the correct page. I built and styled the welcome page. I created the leaderboard page. I created the CSS for the blue theme that we applied to our website, which was applied to all pages.

Patrick Markey:

I wrote a lot of the html and css for our login, register, and created the handlebars pages for all of our games. I used Bootstrap for all of the css I wrote. I also worked on lab 13 to get our project deployed using Render.

Jake Gao:

As a game developer for this project, I focused on the architect and polishment of our game that corresponds with translation dynamics. In addition to the game I myself worked on, I also completed the test cases using Mocha/Chai myself. The game I specifically worked on was game 2, drag & drop. While collaborating with Chris, we figured out how to translate words on-the-fly with the Google Translate API. I wanted my game to be a bit more complex, so I added difficulty levels, real-time scoreboard, and feedback system. The UI side unfortunately does not match well with the rest of the games but I myself wanted a different look to each unique game. had to balance the right bootstrap layout with my webpage and handled most merge conflict situations that arise.

Chris Coleman:

I created our multiple choice, flashcards, and finish the sentence games. This included writing the java script code for the games as well as some of my own css and html for my game pages. I also had to embed my work into our website and connect it with features already in the website such as the language drop down selector.

Isaiah Millington:

I was initially responsible for creating the API routes found in our index.js to render the .hbs files and connect them to our server's hosted port. I was in charge of Docker troubleshooting and helped get it back running after we had some issues early on.

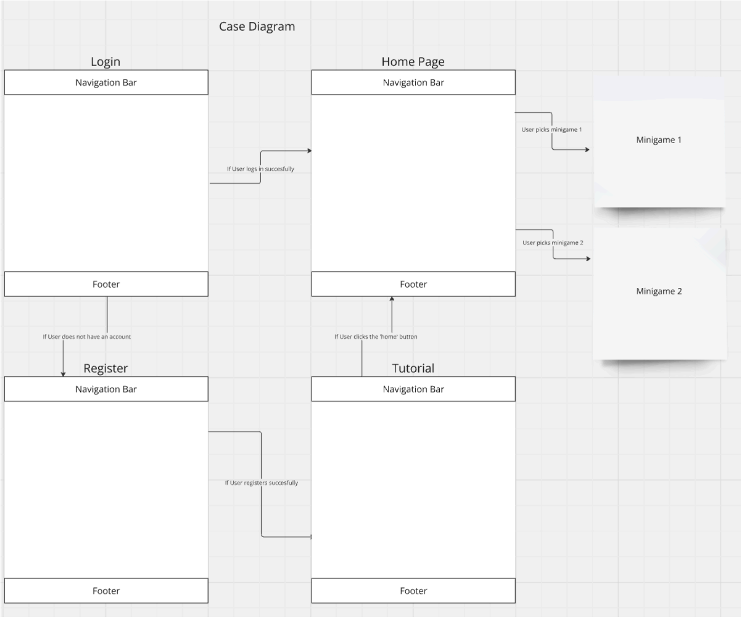
Midway through, I transitioned into something more like a quality checker, by reviewing Git PRs and reorganizing our file structure to better align with the project rubric.

Towards the end, I overhauled the visual design of the website, tweaking the colors to be more engaging, adding clickable buttons, enforcing consistent design across all pages, and adding a dark mode option.

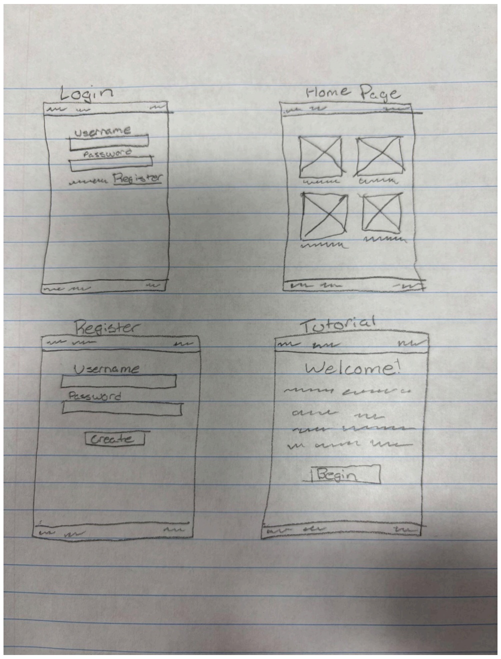
Hunor Kovacs:

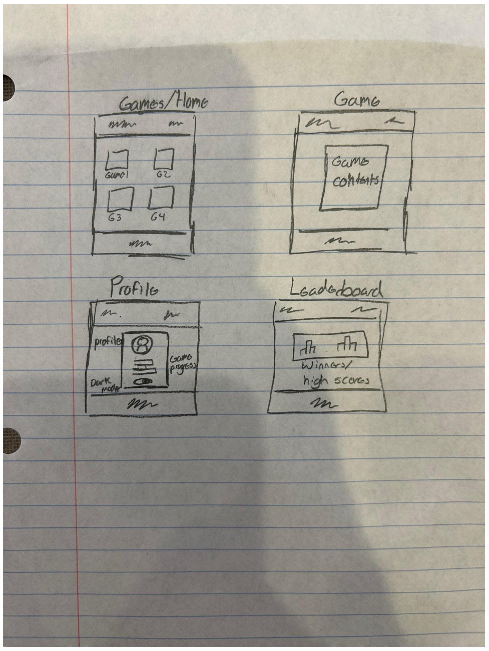
I have worked on the backend, setting up our database. Also created the Google Translate API function found in translate.js and worked some on the API routes. I have added some design details as well (like the float-in animation on the games page).

Use Case Diagram:



Wireframes:





Test results

We tested two major features of our application:

1. User Registration and Login

Goal: To ensure that users can create accounts, handle duplicate usernames, and log in.

Case Tested:

1. Successful registration of a new user.
2. Error on duplicate username registration.
3. Successful login with valid credentials.
4. Proper error handling when an incorrect password is entered.

2. Drag-and-Drop Vocabulary Game (Game 2)

Goal: Ensure the vocabulary game initializes, accepts custom words, and handles correct and incorrect word matches.

Cases Tested:

5. Generate new word/zone pairs after selecting languages.
6. Correct drag-and-drop match increments score.
7. Incorrect match displays error without scoring.
8. Custom word list generates playable content.

Audience:

We tested with two individuals outside our team:

- Ming L., Business student
- Ryan K., Mechanical Engineering student

This helped us gather both technical and non-technical user perspectives.

Observations:

For registering, both Ming and Ryan were able to complete it on their first attempt. So the behavior works as expected. As for login, they felt it was intuitive but behavior also works as expected.

As for game 2, Ming easily understood how drag and drop games worked while Ryan was confused with the custom words system at first. It seems like they were expecting pre-translated words after typing a custom list but didn't realize you would have to click on the button. In a way this part of the behavior passed as well but may not be as user aligned. For the feedback and scoring system of the game, both users liked the implementation of the green/red flash feedback. This immediate scoring system made the game more enjoyable.

Both features passed core functionality testing.

If needed, can refer to UAT_Plan.md under MilestoneSubmissions for more information

Deployment:

https://group-2-project-ql8w.onrender.com