

Product/Team Name: ULearn/CampusConnect

Iteration 02 - Review & Retrospect

- When: July 6, 2024
- Where: Online (Discord)

Reflection

Decisions that turned out well

- Better allocation of roles and responsibilities: we improved our project management by allocating roles and responsibilities more effectively. Each team member was assigned specific tasks that leveraged their strengths and accommodated their schedules based on other course deadlines, ensuring that work was distributed evenly and everyone had clear, manageable objectives. For example, the filter and search features depended on the homepage being completed first, so we assigned this task to a team member with less coursework at the beginning of the sprint. This approach allowed us to complete our tasks more smoothly and maintain a high level of productivity throughout the sprint.
- Better time management: we significantly enhanced our time management skills during this sprint. By setting realistic deadlines and regularly reviewing our progress, we were able to stay on track and avoid last-minute rushes.
- Better documentation: we made a concerted effort to include in-line comments within our code, which greatly enhanced readability and made it easier for team members to understand each other's work. Main module and function was accompanied by comprehensive code comments that explained its purpose, inputs, outputs, and dependencies.

Decisions that did not turn out as well as we hoped

- Not having our review meeting prior to demo: one decision that did not turn out as well as we hoped was not scheduling a review meeting prior to our demo. This oversight led to several minor issues during the presentation, as some features were not thoroughly tested, and there were minor bugs that could have been addressed beforehand.
- Lack of Pull Request code review: another decision that did not meet our expectations was the insufficient emphasis on pull request (PR) code reviews. Without thorough PR reviews, several issues and bugs slipped through, which could have been caught early in the development process. Also, some files got overwritten during merges which could have been avoided if additional peer code review was performed.

Planned changes

- Having additional thorough pr code review: We plan to implement more thorough peer code reviews before accepting pull requests. Recognizing the issues caused by insufficient code reviews in the sprint, we will establish stricter guidelines for PRs and ensure that every piece of code undergoes a detailed peer review process. This includes verifying code quality, adherence to coding standards, and identifying potential bugs or vulnerabilities, ensuring that the features do not affect existing features or overwrite functions.

- Modularizing code for better cohesion: To enhance the maintainability and cohesion of our project, we plan to modularize our code more effectively. By breaking down our codebase into smaller, self-contained modules, we can ensure that each module has a clear purpose and can be developed, tested, and debugged independently. This approach will not only make our code more readable and manageable but also facilitate easier updates and modifications in the future.

Product - Review

Goals and/or tasks that were met/completed:

1. Home page
 - a. Tutors can be previewed with their profile images, name, at most two courses that they teach, and rate, in a grid style
 - b. Only tutors that teach in the same language as the user are shown in the homepage
 - c. Tutor profiles can be selected for further functionalities
2. Search Feature
 - a. A search bar will be placed at the top of the homepage, centered right below the navbar.
 - b. Search Bar Details
 - i. The search bar should allow users to type course codes to search for tutors.
 - ii. The search functionality should be case-insensitive, meaning the casing of letters doesn't matter.
3. Chat Features
 - a. Users should be able to upload images and files in the chat to send to the other user.
 - b. The chat will be developed using the Getstream API, which includes comprehensive chat features:
 - i. Read receipts
 - ii. Emoji reactions
 - iii. Editing and deleting messages
 - iv. File and image uploads
 - c. Users should be able to scroll through a chat and view entire message history
4. Video Call Feature
 - a. Initiating a Video Call
 - i. At the top of every chat channel, there will be an option for tutors to call students.
 - ii. Students will not have a call button and will not be able to call tutors. This decision was made to prevent students from disturbing tutors or spamming them with calls if a session has not been booked.
 - b. Sending Meeting ID
 - i. When a tutor initiates a call, a unique meeting ID is generated and sent in the chat to the student.
 - ii. Students can click on the link containing the meeting ID, which opens a separate page for the video call.
 - c. Pre-Meeting Setup
 - i. Before entering the meeting, users will have the option to turn their camera and mic on or off.

- d. Using Stream IO API
 - i. The video calling feature leverages the Stream IO API for easy access to video calling functionalities.
 - ii. The application uses the API key and secret key provided by Stream IO to authenticate all users on the platform with a token.
 - iii. This token allows users to access, join, and create calls, all handled by the API.
- e. Stream IO Features
 - i. Stream IO comes with a variety of useful video calling functionalities, including:
 - 1. Built-in call generators
 - 2. Video call layouts
 - 3. Mic and camera settings
 - 4. Screen sharing
 - 5. Recording
- f. During the Call
 - i. Once both users have their mic and camera set up, they are free to communicate for up to an hour.
 - ii. The call interface should allow users to easily manage their mic and camera settings, share their screen, and record the call if necessary.
- g. Ending the Call
 - i. Once the session is done, the tutor or the student have the option to leave then call and the tutor has the option to end the call for everyone.
 - ii. After ending the call, both users will be returned to the application's home page.

Goals and/or tasks that were planned but not met/completed:

- 1. Filtering tutors based on hourly rate, and tutor rating goal was not met due personal reasons of a team member (discussed with the prof).

Meeting Highlights

- Enhance code cohesion: In this sprint, we identified the need to enhance the cohesion of our code. By focusing on making our code more logically organized and ensuring that each part of our codebase is highly related to its function, we aim to improve both maintainability and ease of understanding. This will help us reduce dependencies and make our codebase more robust.
- Focus on modularizing code and improve code usability: To improve the overall structure and manageability of our project, we will focus on modularizing our code. Breaking down our code into smaller, self-contained modules will make it easier to develop, test, and debug. These changes will make our codebase more accessible and easier to navigate for all team members.
- Maintain a more thorough pr code review: We plan to implement a more thorough peer code review process before accepting pull requests. This includes verifying code quality, checking for bugs, checking that the new changes affect other features, etc. We will ensure that new code does not interfere with existing functionalities by incorporating rigorous regression testing. These measures will help us catch issues early, reduce the need for rework, and improve the overall quality of our codebase, fostering a culture of collaboration and accountability within the team.