Software Development Folder

Table of Contents

Project Status Sheets

Gruuper

Grouping Web App powered by Machine Learning

Proposal Document

Gruuper

Grouping Web App powered by Machine Learning

In the realm of modern education, fostering collaboration and engagement among students is crucial for effective teaching and learning. Group assignments serve as a valuable means to encourage teamwork and knowledge sharing. However, the process of creating and managing student groups can be a time-consuming and complex task for educators.

Enter "Gruuper" - our senior project's innovative web application tailored for professors and teachers. The primary objective of Gruuper is to simplify the creation and management of student groups, ultimately improving the classroom experience. This web app provides educators with an intuitive platform to generate, save, and adjust student groups as needed. By developing Gruuper, we aim to streamline group management for educators, enhancing the educational experience for both instructors and students.

Currently, the only class-grouping websites out there are all randomly based. There are an endless amount of Random Team/Group generators out there, both as websites and applications. Some bigger applications, like Class Dojo, also have functions to randomize groups as well, but with no complex group matching.

With our questionnaire, we aim to lead the market with specifically-curated group creation, rather than random. In addition, with high customization, the teacher can use their knowledge of the class to hone in the ability of our group selector. We hope to design an intuitive website to capture all of these different aspects in a clean design.

For this project, we will be constructing the web-app with React, and deploying it with Vercel. For the back end, we will primarily be using Google Firebase. Our goal for Semester II is to implement our "group-selector" using Python.

At Gruuper, we're embarking on an exciting journey. In last semester, our focus has been on laying the foundation for a user-friendly platform designed to empower teachers in creating and managing class groups effortlessly. However, we recognize that this is just the beginning. Our primary goal is to introduce the standout feature: intelligent grouping functionality.

In the upcoming months, our roadmap is clear. Firstly, we are dedicated to implementing a distinctive student questionnaire profiling system. This will involve students completing comprehensive profiles that delve into their skills, work-life experiences, and personal interests.

Subsequently, leveraging the power of Machine Learning, we aim to utilize these anonymized personal profiles to intelligently match students into groups that best complement their individual strengths and preferences. This cutting-edge approach will ensure optimized group formations tailored to enhance the learning experience for every student.

As Computer Science majors, we have had plenty of experience with the struggles of group work. For this project, we will have the opportunity to use the skills we have learned in classes such as Web App Development, Interaction Design, and Cognitive Systems Design in order to create our project. However, we also get to explore areas where we have not spent a lot of time, like User Experience and Machine Learning. With our web app already built from last semester, it leaves us the perfect amount of time to focus on our group selection functionality this semester. We feel that this project is something that we are certainly capable of completing, and we have gotten great feedback from instructors like Dr. Forney who would love to use it in the future!

All Project Presentation Slides

Gruuper

Grouping Web App powered by Machine Learning

Software Development Plan

Gruuper

Grouping Web App powered by Machine Learning

4.1 Plan Introduction

This Software Development Plan provides the details of the planned development for the Gruuper Software which is a web application to streamline homework and test grouping for Teachers, Professors, and more.

The Gruuper project aims to revolutionize the way group formation occurs within educational settings by leveraging technology to streamline the process. At its core, Gruuper is a web application designed to facilitate group creation for homework assignments, tests, and other classroom activities. Through Gruuper, teachers can easily create classrooms and generate unique codes for student enrollment. Once enrolled, students provide information about their availability, skills, and interests through a user-friendly interface. Gruuper then utilizes a sophisticated machine learning model to analyze this data and form groups of students with complementary attributes, fostering collaboration and productivity. The rationale behind the development of Gruuper stems from the recognition of the challenges inherent in traditional group formation methods, such as manual sorting and coordination, which are often time-consuming and inefficient. By automating and optimizing this process, Gruuper aims to enhance the educational experience for both teachers and students, ultimately leading to improved learning outcomes.

Master Schedule Milestones:

- ➢ Weekly written status reports
- ➢ Weekly oral reports
- ➢ Software Development Plan Document [Updated]
- ➢ Requirements Specification Document [re-submit]
- ➢ Web app testing via in-classroom use
- ➢ Completed Machine Learning Model (ML)
- ➢ ML model effectiveness testing and tuning
- ➢ Fully furnished and consistent web app user interface (UI)
- ➢ Preliminary Version of Poster
- ➢ FINAL Project Presentation
- ➢ FINAL Product Delivery [Final Report and Code]

4.1.1 Project Deliverables

a. Weekly written status reports
   - Due: Weekly
   - Explanation: Status reports are really important for tracking your own progress. As a project is developed, being able to see which milestones are approaching, which ones have passed, and which ones are late can help you understand where you need to focus more effort for success. Status reports should also include a section for identified problems and possible solutions to those problems – this is the whole idea behind risk mitigation. Status reports are also a handy mechanism for documenting the amount of time you have spent on a project, which is very important when you are employed as a consultant. The more you know about the time required to complete a task, the better you can forecast the next job, so you can bill your clients accurately. See the material on the PSP page for more ideas.

b. Weekly oral reports
    - Due: Weekly
    - Explanation: Same as written reports except they will be presented orally.
c. Software Development Plan Document [Updated]
    - Due: Week 12
    - Explanation: This plan is intended to describe the process that will be used during the semester, leading to the production of all required documents and software for your project. This SDP has sections about all the gory details of the software project, including how the development team is structured, what the configuration management plan is, how the defined coding standards go, and on and on and on.
d. Requirements Specification Document [re-submit]
    - Due: Week 13
    - Explanation: The Software Requirements Specification document (SRS) is intended as a part of the contract between the customer and the solution provider. The document specifies EXACTLY what is being built.
e. Web app testing via in-classroom use
    - Due: Week 12
    - Explanation: Web app testing via in-classroom use involves deploying the Gruuper application within actual classroom settings to assess its performance and usability in real-world scenarios. This method allows testers to observe user interactions, group formation processes, and data management within the application. By gathering feedback from teachers and students, testers can identify any issues or areas for improvement and ensure that the application meets the needs of its intended users effectively. Additionally, in-classroom testing enables the evaluation of system performance under authentic conditions, including network connectivity and user behavior variability, to ensure reliability and scalability.
f. Completed Machine Learning Model (ML)
    - Due: Week 13
    - Explanation: This ML model is an integral component of the Gruuper application, responsible for processing user-provided information and generating optimal group configurations. Once completed, the ML model should demonstrate proficiency in accurately clustering users into cohesive groups, thereby enhancing collaboration and productivity within educational settings. The completion of the ML model signifies that it has undergone rigorous training and validation processes, ensuring its reliability and effectiveness in supporting the group formation functionalities of the Gruuper application.
g. ML model effectiveness testing and tuning
    - Due: Week 15
    - Explanation: In order to ensure the effectiveness of the ML model within the Gruuper application, rigorous testing and tuning

processes are essential. ML model effectiveness testing involves subjecting the model to various datasets representing different user profiles and scenarios to evaluate its performance in accurately grouping individuals based on predetermined criteria. This testing phase aims to assess the model's ability to produce meaningful and coherent groupings while minimizing errors and misclassifications. Additionally, tuning the ML model involves refining its parameters and algorithms to optimize performance and enhance accuracy. Through iterative testing and tuning cycles, potential weaknesses or limitations of the model can be identified and addressed, ultimately improving its reliability and effectiveness in facilitating group formation within the Gruuper application.

h. Fully furnished and consistent web app user interface (UI)
    - Due: Week 16
    - Explanation: The Gruuper project necessitates the development of a fully furnished and consistent web application user interface (UI). This UI serves as the primary point of interaction for users, facilitating intuitive navigation and seamless access to all features and functionalities of the application. The UI design should prioritize user experience, ensuring clarity, consistency, and ease of use across all screens and elements. By achieving a fully furnished and consistent UI, the Gruuper application aims to enhance user engagement, satisfaction, and overall usability.
i. Preliminary Version of Poster
    - Due: Week 14
    - Explanation: The preliminary version of the poster serves as a visual representation of the Gruuper project's objectives, methodology, and anticipated outcomes. It outlines the key components and functionalities of the application, providing stakeholders with a comprehensive overview of the project's progress and direction. The poster serves as a communication tool for presenting initial findings, insights, and challenges encountered during the project's development phase. Feedback obtained from the presentation of the preliminary poster version guides further refinement and enhancement of the Gruuper application in preparation for the final project presentation.
j. FINAL Project Presentation
    - Due: Week 16
    - Explanation: The final project presentation marks the culmination of the Gruuper project, showcasing the completed application and its capabilities to stakeholders, including project sponsors, instructors, and peers. During the presentation, the project team provides a detailed overview of the application's features, functionalities, and implementation process. Additionally, the team highlights key achievements, challenges overcome, and lessons learned throughout the project lifecycle. The final presentation offers an opportunity to demonstrate the Gruuper application's value proposition, relevance, and impact within

educational settings, as well as to receive valuable feedback and insights from the audience.

k. FINAL Product Delivery [Final Report and Code]
    - Due: Week 17
    - Explanation: The final product delivery encompasses the submission of the Gruuper project's deliverables, including the final report and codebase. The final report provides a comprehensive documentation of the project's objectives, methodologies, findings, and conclusions. It includes detailed descriptions of the application's architecture, design decisions, implementation strategies, and testing methodologies. Additionally, the report presents an analysis of the project's outcomes, including user feedback, performance evaluations, and future recommendations. Alongside the final report, the project team delivers the complete codebase of the Gruuper application, ensuring transparency, reproducibility, and accessibility for future iterations or extensions of the project.

4.2 Project Resources

4.2.1 Hardware Resources

Development Machine:

- Device: Apple M1 MacBook Air
- Processor: Apple M1 chip with 8-core CPU
- RAM: 16 GB unified memory
- Storage: 512 GB SSD
- Operating System: macOS Monterey

Application Host (Server-side):

- Service: Amazon Web Services (AWS)
- Instance Type: EC2 (t3.micro)
- Processor: Virtualized
- RAM: 1 GB
- Storage: EBS (20 GB SSD)
- Operating System: Ubuntu Server 20.04 LTS

Database Server:

- Service: Google Firebase Firestore
- Plan: Spark Plan
- Storage: Cloud Firestore (scalable)

Mobile Device Testing:

- Devices: iPhone 12 Pro (iOS)
- Operating System: iOS 15

4.2.2 Software Resources

During the development of the Gruuper project, the following software tools will be required:

Development Environment:

- Text Editor: Visual Studio Code (Version 1.63.2)
- Web App Engine: Chrome (Latest)
- Operating System: macOS Monterey (for development machine)

Frontend Development:

- Framework: React.js (Version 17.0.2)
- Package Manager: npm (Version 8.4.0)
- Version Control: Git (Version 2.35.1)
- Design Tool: Figma

Backend Development:

- Server-side Framework: Node.js (Version 16.14.0)
- Database: Google Firebase Firestore
- Cloud Platform: Amazon Web Services (AWS)

Mobile Development:

- IDE: Visual Studio Code (Version 1.63.2)
- Programming Language: JavaScript (for web app)

Testing:

- Testing Framework: Jest (Version 27.5.0) (Potentially)

Collaboration and Communication:

- Project Management: Gantt Chart
- Communication: Discord

Version Control Hosting:

- Platform: GitHub

These software tools, along with the specified versions, will facilitate the development, testing, and collaboration efforts required for the successful implementation of the Gruuper application.

4.3 Project Organization

The Gruuper project is organized into three major functions, each overseen by a dedicated team member:

1. Development Team (Brady)
    - Responsibilities: Brady is responsible for overseeing the overall development of the Gruuper application, including frontend and backend development,

database integration, and ensuring the alignment of technical solutions with project requirements.
    - Development Plan: Brady will lead the development team in implementing the frontend web application using React.js, designing and deploying the backend infrastructure on AWS, integrating Google Firebase Firestore for database functionality, and collaborating with the ML team to integrate the ML model into the application.
2. Machine Learning Team (CJ)
    - Responsibilities: CJ is responsible for developing the machine learning model for Gruuper, which will be written in Python and exposed as a separate API. This model will analyze user data to form groups based on interests and work styles.
    - Development Plan: CJ will work on developing, training, and fine-tuning the machine learning model using Python libraries such as TensorFlow or PyTorch. They will also create a RESTful API to expose the model's functionality and integrate it seamlessly into the Gruuper application.
3. Quality Assurance Team (Sam)
    - Responsibilities: Sam is responsible for ensuring the quality and reliability of the Gruuper application through comprehensive testing and quality assurance processes. This includes conducting unit tests, integration tests, and user acceptance testing to identify and address any bugs or issues.
    - Development Plan: Sam will develop a testing strategy and execute test plans to validate the functionality and performance of the Gruuper application across different platforms and environments. They will collaborate closely with the development team to address any identified issues and ensure a high-quality end product.

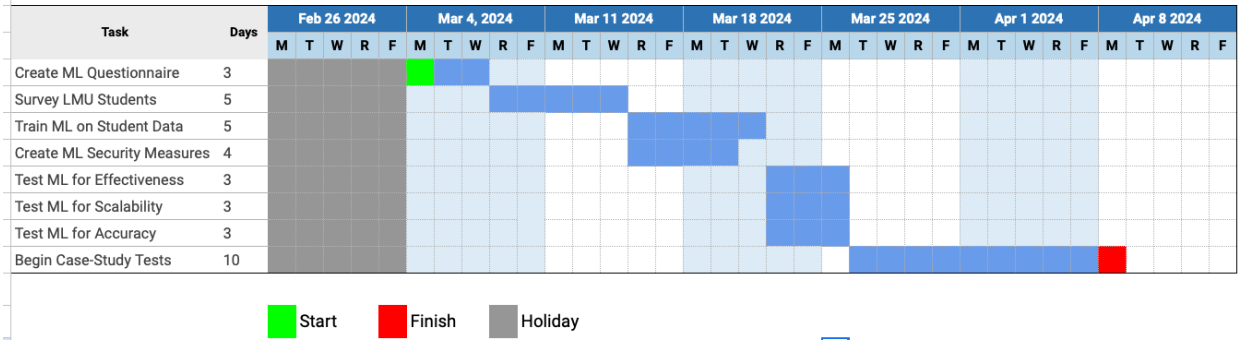
By dividing the project into these major functions and assigning responsibilities to each team member, the Gruuper project is well-organized and equipped to efficiently achieve its goals.

4.4 Project Schedule

This section provides schedule information for Gruuper. After our first semester, we have built up the main infrastructure of the Web App. Here are our remaining tasks for the semester:

1. Create ML Questionnaire
2. Survey LMU Students
3. Train ML on Student Data
4. Create ML Security Measures
5. Test ML for Effectiveness
6. Test ML for Scalability
7. Test ML for Accuracy
8. Begin Case-Study Tests

4.4.1 PERT / GANTT Chart



4.4.2 Task / Resource Table

Task	Resources
Create ML Questionnaire	Surveying, Google Forms, Academic Research
Survey LMU Students	LMU Students & Faculty
Train ML on Student Data	Method TBD
Create ML Security Measures	Manual performance and security testing
Test ML for Effectiveness	Manual testing
Test ML for Scalability	Manual Testing
Test ML for Accuracy	Manual Testing
Begin Case-Study Tests	LMU Students & Faculty

Requirements Specification Document

Gruuper

Grouping Web App powered by Machine Learning

5.1 Introduction

Gruuper maintains three components that work together:

- A **web app** for simple user interaction and group formation
- A **database** for storing user information and classroom members
- An **Machine Learning (ML) model** to group people with similar interests and workstyles

The remainder of this document is structured as follows:

- Section 5.2 contains the functional requirements for the application
- Section 5.3 describes the application's performance requirements
- Section 5.4 includes the environment requirements

5.2 Functional Requirements

This section describes the requirements each component of the application must meet.

5.2.1 Web App (Frontend) Functional Requirements:

5.2.1.1 Users shall be greeted with a sign in page with email and password authentication.

5.2.1.2 The front-end shall utilize the Material UI library for a clean and consistent design across all components.

5.2.1.3 There shall be a screen with the option to create or join a classroom.

5.2.1.4 The website shall feature a profile page for the user to view their information.

5.2.1.5 The profile page shall include a survey to give information to the ML model

5.2.1.6 The classroom page shall have a list of the members of the class, as well as their groupings.

5.2.1.7 When in the classroom view, a professor shall have the ability to randomize groups, select group size, and manually edit groups

5.2.1.8 Students shall not have this ability

5.2.1.9 There shall be a section where users can view a list of their enrolled classes, and be able to view them with a click

5.2.2 Firebase Datastore Functional Requirements:

5.2.2.1 The database will be hosted on Google Firebase using the Realtime database.

5.2.2.2 The database will store user authentication information and process log-in attempts.

5.2.2.3 The Realtime Database will store user data such as name, user ID, and enrolled classrooms

5.2.2.4 The database shall also store classroom information such as the creator, group pairings, and members

5.2.2.5 Users and data can also be deleted from firebase by an administrator.

5.2.2.6 Database administrators will be able to track user activity in Firebase.

5.2.2.7 The authentication shall allow users to update and change their email and password.

5.2.2.8 The authentication shall validate users' emails when signing up for the first time.

5.2.3 Machine Learning Functional Requirements:

5.2.3.1 The model shall group people in the class by their interests and workstyles

5.2.3.2 The model shall get its information from each user’s personality survey answers

5.2.3.3 The model shall be trained on labeled data.

5.2.3.4 The model shall adhere to security best practices, such as encryption of sensitive data and access control mechanisms

5.2.3.5 The model shall allow instructors to override automatic groupings to meet specific pedagogical needs.

5.2.3.6 The training process shall be scalable to large datasets such as large numbers of students in a given classroom.

5.3 Performance Requirements

The performance requirements of the Gruuper application are crucial to ensure a seamless user experience and efficient operation. These requirements dictate the responsiveness, reliability, and scalability of the system under various conditions. In this section, we outline the performance criteria that the application must meet to meet user expectations and support its intended usage scenarios.

5.3.1 Web App (Frontend) Performance Requirements

5.3.1.1 The webpage must load within one second.

5.3.3 Firebase Datastore Performance Requirements

5.3.3.1 The datastore shall respond to queries from the web app within 5 seconds. If the datastore fails to do this, the request should be logged for debugging.

5.3.4 Machine Learning Performance Requirements

5.3.4.1 The model shall be scalable to handle increasing volumes of data and computational resources.

5.3.4.2 The model shall respond to queries from the web app within 10 seconds.

5.4 Environment Requirements

The following sections describe what is required for a developer to run and deploy the application.

5.4.1 Development Environment Requirements

A developer for Gruuper will need a system meeting the following requirements to build the web app:

- Windows, MacOS, or Linux operating system
- A modern web browser, such as Chrome or Firefox
- Node.js v18.17.1 runtime and Python v3.11 or newer

In development, the only external server required is the Firebase datastore, which is automatically kept online and accessible. All other components can be run locally on the developer's system, and source files can be modified with any code editor. The Smart Grouping Machine Learning Model, while in development, will be hosted using Flask. The web app uses node.

5.4.2 Production Environment Requirements

Currently, the web app is deployed on Vercel out of ease. Eventually, both the web app and the Smart Grouping Machine Learning API will be hosted on AWS.

5.4.3 Execution Environment Requirements

- Operating System: Compatible with modern web browsers on Windows, macOS, and Linux.
- Internet Connection: Stable internet connection required for accessing the web application.

Final User Manual

Welcome to Gruuper, an innovative web application designed to simplify classroom management for educators. Gruuper offers a user-friendly interface that helps professors and teachers to create, manage, and modify student groups effortlessly. This guide will walk you through the steps of installing, using, and troubleshooting Gruuper to ensure you get the most out of your experience.

Table of Contents

1. Installation

Requirements

- A modern web browser (Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge)
- Active internet connection

Steps

- Gruuper is a web-based application and does not require traditional installation. Simply open your web browser and navigate to Gruuper's website to get started.

2. Starting the Application

- Open your web browser.

- **Enter the URL:** `https://gruuper.vercel.app`
- **The Gruuper homepage will load, presenting you with options to sign in or sign up.**

3. Stopping the Application

- **To stop using Gruuper, simply close the web browser tab or window where Gruuper is running.**

4. Uninstalling the Application

- **Being a web-based application, Gruuper does not require uninstallation. If you wish to stop using Gruuper, you can simply cease to visit the website.**

5. Using Gruuper

Accessing and Navigating the Interface

Upon visiting Gruuper, you will be greeted by the homepage. Here you can sign up or log in using your credentials.

Creating a Classroom

- **Navigate to the “Classrooms” section after logging in.**
- **Click on “Create Room.”**
- **Enter the classroom details and confirm by clicking “Create.”**
- **Your new classroom will be visible in your dashboard.**

Joining a Classroom

- **On the dashboard, click “Join Room.”**
- **Enter the classroom code provided by your institution.**
- **Click “Join” to be added to the classroom.**

Managing Student Groups

- **Within a classroom, select “Manage Groups.”**

- **Use the interface to add, remove, or shuffle students between groups. You can drag and drop groups**
- **Use the “Shuffle” button to create random groups**
- **Use the “Smart Match” button to create matched groups**
- **Click the “Lock” icon to lock a group**
- **Changes can be saved automatically or by clicking the “Save” button.**

Advanced Features

- **Group Randomizer: Automatically assign students to groups based on pre-set criteria.**
- **SmartMatch System: Optimize group dynamics based on student profiles and past interactions.**

Setting Up User Profiles

To enhance the group formation process, Gruuper employs a unique profile questionnaire that gathers information to facilitate smarter group assignments.

Completing Your Profile Questionnaire

- **Navigate to your profile page by clicking on your avatar in the top right corner and selecting "Profile."**
- **On the profile page, you will find the "Edit Profile" button.**
- **Click on "Edit Profile" to access the questionnaire.**

Questionnaire Fields

- **Major: Helps in grouping students by academic discipline.**
- **Class Year: Useful for matching students in similar stages of their academic career.**
- **Night or Morning Person: This preference helps in aligning group members with similar working hours.**
- **Social Preference: Indicates whether the student prefers to work alone, in pairs, or in larger groups.**
- **Deadline Behavior: Helps understand a student’s approach to deadlines, which is critical for group harmony and effectiveness.**
- **After filling out the questionnaire, click “Save” to update your profile with this new information.**

6. Troubleshooting

- **Problem: Unable to log in.**
    - **Solution: Check your internet connection and ensure you are using the correct credentials. Reset your password if necessary.**
- **Problem: Changes to groups are not saving.**
    - **Solution: Ensure you click the “Save” button after making changes. If the problem persists, check your network connection.**

7. Contact Information

For further assistance, please contact our support team:

- **Email: sricha36@lion.lmu.edu**

8. Glossary

- **Classroom Code: A unique identifier used by students to join a classroom created in Gruuper.**
- **Group Randomizer: A feature in Gruuper that automatically distributes students into groups based on set parameters.**

This user manual is designed to assist you in navigating through the Gruuper platform with ease, ensuring a smooth and efficient classroom management experience. Whether you are a seasoned educator or new to using digital platforms, Gruuper is here to enhance your educational environment.

Status Sheets(Kanban)

Kanban Task Tracking

Gruuper

Sam Richard, Brady Amunsdon, CJ Plantemoli

Queue	Development	Done
[1] Update NavBar UI Due: 2024/04/24 [1] Add more to classrooms page Due: 2024/04/24 [1] Fix Mobile view Due: 2024/04/24 [1] Unmatched Mems to Firebase? Due: 2024/04/24 [1] Randomizer animation Due: 2024/04/24 [1] SmartMatch animation Upgrade Due: 2024/04/24 [1] Bug when deleting student and refreshing Due: 2024/04/24 [1] ^and removing from unmatched members Due: 2024/04/24 [1] Report for smartmatch results Due: 2024/04/24 [1] Class groupchat Due: 2024/04/24 [1] Student submitted classmate preferences Due: 2024/04/24 [1] Professor to access student info Due: 2024/04/24 [1] Multiple assignments within a classroom, with the ability to show the current one and go back and see the old ones Due: 2024/04/24 [1] Admin account to see all groups, edit instructor, etc Due: 2024/04/24 [1] Log group changes Due: 2024/04/24 [1] Student/Faculty dashboard Due: 2024/04/24 [1] lmu registration requirement Due: 2024/04/24 [1] sync profile pic Due: 2024/04/24 [1] edit photo /reset password bug Due: 2024/04/24	[1] Student Classroom View UI Update Due: 2024/04/24	[3] Change Password Due: 2024/04/24 [1] Hide groups in Student View Due: 2024/04 [1] Make SmartMatch work Due: 2024/04/24 [1] Show profile details/Update UI Due: 2024/ [1] Populate existing profile details Due: 2024 [1] Lock groups Due: 2024/04/24 [1] Unmatched Members List Due: 2024/04/24 [1] Update Home UI Due: 2024/04/24 [1] Wrong student view when joining a group [1] If not instructor redirect to student view D [1] Fix New Group Area Due: 2024/04/24 [1] Group size + lock groups Due: 2024/04/24 [1] Firebase profile pic Due: 2024/04/24