Spring 5-9-2025

Optimized Student Grouping for Enhanced Classroom Performance

Kathryn E. Reardon
University of Mississippi

OPTIMIZED STUDENT GROUPING FOR ENHANCED CLASSROOM PERFORMANCE

by
Kathryn Elizabeth Reardon

A thesis submitted to the faculty of The University of Mississippi in partial fulfillment of the requirements of the Sally McDonnell Barksdale Honors College.

Oxford, MS
May, 2025

Approved By

Advisor: Dr. Yixin Chen

Allan Bellman

Reader: Dr. Allan Bellman

Timothy Holst

Reader: Dr. Timothy Holston

DEDICATION

To my parents, I owe my life successes. You have taught me the power of hard work and never giving up, but more importantly, you have shown me the importance of balancing academics with personal life. Thank you for all your love and support on my academic journey.

To my brother, I could not have completed this project without your unwavering support. You kept me grounded when I was frustrated with coding and helped me think through the difficult challenges. Thank you for your encouragement and wisdom.

To my sister, I always look to your wise words when I am ready to give up. I appreciate you always holding me to a high standard, encouraging me to keep trying, and reminding me of how far I have come.

To Dr. Bellman, you have helped me become a great educator. I would not be writing this capstone today without your support and guidance throughout my four years at this institution. Thank you for always holding me to a high standard and pushing me to be the best I can be.

ACKNOWLEDGEMENTS

To Dr. Yixin Chen, thank you so much for all your help throughout my time at the university and more specifically on my capstone project. Without your guidance, encouragement, and support, I would not be writing this thesis. I appreciate you always having your door open to help me make decisions about courses and support me with my academic studies. Thank you for guiding me and supporting me throughout my time in the computer science department.

ABSTRACT

OPTIMIZED STUDENT GROUPING FOR ENHANCED CLASSROOM PERFORMANCE

Effective grouping methods enhance classroom collaboration and allow for a student-centered teaching approach; however, traditional grouping methods are time-consuming, subjective, and can create inconsistent group dynamics. This project addresses these challenges by employing a data-driven approach to optimize student groups based on academic performance, behavior, attendance, language barriers, and teacher preferences. The minimum viable product is a web application with an algorithm-driven system to group students and a database storage for group results. During the initiation phase, a problem was defined with a proposed solution. During the planning phase, potential design choices and grouping methods were researched and assessed. During the development phase, the web application with the grouping algorithm was developed. The final phase was the closing phase where the product was delivered with a final report and oral presentation. These phases were completed in conjunction with status report updates to develop a successful minimum viable product.

TABLE OF CONTENTS

1. INTRODUCTION

The University of Mississippi Computer Science Department strives to support students in developing fundamental computing skills that can be applied to solve practical problems. Students took many courses that target a focus area such as Java, Python, algorithms, operating systems, databases, etc. In cumulation of these courses, undergraduate students completed senior projects. This course requires students to apply the knowledge and skills they have acquired over their time at the university to develop a marketable product.

1.1 SENIOR PROJECT OVERVIEW

Over one semester course, computer science undergraduate students completed their senior project—a course designed for students to apply their computing skills to address a problem of their choosing. Students learned to engage in professional manner by pitching a project idea to find a sponsor, completing status updates, documenting the development process, and presenting their final products to faculty members. The projects developed in this course were completed independently and challenged students to intertwine all the skills learned over their four years at the university. This course is completed in phases—initiation, planning, development, and closing—with status reports throughout the course of the semester to monitor student progress.

2. INITIATION

The initiation phase involved determining a problem and devising a product to address it, pitching the project to faculty members to find a sponsor for the project, and developing the Prospectus. The Prospectus is a document that details the problem and selection of the problem, the minimum viable product, the users of the product, challenges that may occur in developing the product, and timeline of the product.

2.1 PROJECT PITCH

The project was pitched at the beginning of the Fall 2025 semester during the first two weeks of classes. The project was pitched to some Department of Computer Information and Science faculty members to find a sponsor for the project. Once the sponsor was determined, the capstone survey form and professional conduct statement were signed, and the prospectus with the overview of the project details was developed with the sponsor [Appendix A-B]. These documents were then submitted for approval by the course instructors. They reviewed the documents to determine if the project was challenging enough to demonstrate the abilities gained at the institution to the faculty, other students, and potential employers. Once the project was approved, the capstone project commenced.

2.2 PROSPECTUS

The Prospectus established an overarching outline for the project, explaining the problem it aimed at addressing, the minimum viable product, who the users of the product would be, the challenges that may occur in developing the product, and a breakdown of the tasks needed to successfully develop the product [Appendix C].

2.2.1 PROBLEM

As an education, mathematics, and a computer science major, the problem selected to address was grouping students in a classroom, which is time consuming and subjective. Using the disciplines studied at the institution—mathematics, computer science, and education—would allow a product to be created to streamline the grouping process while reducing subjectivity and reallocating time back to educators to lesson preparation.

2.2.2 MINIMUM VIABLE PRODUCT

The minimum viable product (MVP) was developed based on addressing the problem and outlining the result of the product that would be delivered. This was done prior to the research phase, so methods of grouping students, coding language, and other specifications had not been determined at this point. The minimum viable product for grouping students was a web application that groups students based on student data and teacher preferences and stores generated groupings for reference later.

2.2.3 USER TYPE

The user types outline who would use this product once it is complete for all aspects of the project. Since the minimum viable product entails a web application, there had to be an administrator who oversees the overall upkeep of the website. The problem is aimed towards educators, so the other user type was teachers who will use the site to generate groups and reference past groupings. While school administrators, students, and other school personnel could potentially use the site, the base model of this website does not target these audiences.

2.2.4 CHALLENGES

Before beginning to research the information needed to develop the site and while still considering the project, it was important to note challenges and risk that may occur when developing with web application. The biggest challenge that was a concern was using hypothetical data instead of real student data to develop the website. Real student data would not be possible to use due to permission constraints and the one semester time constraint. Other challenges to consider included assimilating quantitative assessment scores with quantitative data and determining the type of sorting algorithm to use.

2.2.5 TASKS AND TIMELINE

The timeline of the project had to be completed within the semester. The senior project had five status report meetings with requirements of what to have prepared to present at these meetings as well as a oral presentation to faculty members at the end of the semester. To complete the project in this timeframe, the project was divided into tasks. First a hypothetical dataset had to be created for student data. Then a compatibility matrix was developed to determine the compatibility between two students. After determining student compatibility from scores, the majority of the project was focused on developing an algorithm to group the students. Finally, the algorithm had to be made accessible to teachers through a user-friendly teacher interface.

3. PLANNING

Once the project was outlined and approved by the sponsor and course instructors, the planning phase defined the project in more detail to include the methods of grouping and coding design choices. This process consisted of the research portion of developing the product and was completed during the first three weeks of the fall semester.

3.1 PROJECT DEFINITION

A web application was researched and developed to group students in a classroom. The status reports and oral presentations were administered at Weir Hall. The initiation and planning phase were completed during the first three weeks of the course and were completed independently outside of course hours. The design specifications, oral presentation slides, and evaluation plan were submitted alongside a final report at the conclusion of the project. By the end of the semester course, a product that met the minimum viable product was completed, the final oral presentation was presented to the faculty members, and a final report was written.

3.2 MINIMUM VIABLE PROJECT

A web application that groups students based on assessment data, behavior data, language barriers, attendance, and teacher preferences and stores generated groupings for reference later was developed. The design of the web application contained a sorting algorithm, a user-interface, a database, a Laravel framework, and a file type.

3.3 GROUPING RESEARCH

For this project, the project was divided into two main parts, the educational aspect of grouping and the coding aspect to develop the website. Both aspects required research into a productive method to develop the website. For grouping students, research was completed to determine the method of grouping as well as what considerations to target when grouping. The grouping research and choices were not the primary focus for the senior project course, so there was no direct documentation of this grouping research. The grouping research was, however, presented during the final presentation.

3.3.1 GROUPING METHODS

Students can be placed in heterogeneous groups, where they are grouped with students of different ability, interest, or characteristics, or they can be placed in homogenous groups of similar characteristics. Heterogeneous grouping promotes collaboration and critical thinking while exposing students to a variety of perspectives. This grouping method also aids students' development of social-emotional skills by teaching students to value and respect differences among peers [1]. Homogeneous grouping allows students to work alongside peers with similar skills or needs. This allows for targeted intervention for some learners and advanced challenges for others. For this project, heterogeneous grouping was used to develop the website as these groups would optimize a classroom by promoting collaboration skills. Students can also be grouped based on ability, interest, or a random grouping. Ability grouping can target a student's skill level and meet the needs of each student by challenging them and providing extra support when needed. Interest grouping is a useful grouping method for project-based or inquiry-based learning as these groupings pair students with similar passions or hobbies to motivate the learner. Random grouping is valuable in that it promotes inclusivity and diminishes subjectivity. Since this project is fixated on the day-to-day classroom and not as much on projects, ability-based grouping was selected to challenge and grow each learner and to encourage learners to use the support of their peers ability to help each other learn. The final grouping method determined after research was a heterogeneous ability-based grouping method. This would promote collaboration and group learning where group members can support and challenge each other by asking their peers for help and explaining concepts to their peers [2]. For a teacher, this would help reduce the amount of one-on-one support because the students would be able to support each other.

3.3.2 GROUPING CONSIDERATIONS

Other grouping considerations were also considered: ability, behavior, personality, interest, social dynamics, and cultural and linguistic diversity [4]. Ability is vital to the growth of students and plays a role in classroom management. Students need to be challenged from their current ability level to help them learn and grow but also to keep them motivated in the classroom. To address this, an assessment file would be used to upload their current assessment scores and determine a student's current ability level. Behavior plays an important role in the classroom management aspect of a classroom, and behavior patterns should be considered to create a productive group dynamic with minimized disruptions. Cultural and linguistic diversity also plays a role in a classroom to promote inclusivity and learning. Behavior and cultural and linguistic diversity were determined as vital to the grouping dynamics, so to include these considerations when grouping students, a behavior file was created for a teacher to rate aspects of student behavior such as disruptive talking, active participation, and attendance as well as identify any language barriers to be evaluated when grouping students. Other considerations such as personality, interests, social dynamics, and relationships are also a key to grouping students in groups that will optimize classroom productivity. However, these characteristics are much more challenging to measure or rate, so the product would allow for teacher input to determine if two students absolutely cannot or must sit together.

3.4 CODING DESIGN CHOICE

There were a lot of aspects of the programming design that had to be researched. Part of this research took place concurrently with the development process as many unforeseen challenges presented themselves when connecting the algorithm to the user interface. The programming language, website language, database, and framework had to be researched to complete this project. All design choices were documented in the prospectus, and the finalizations of the design choices at the conclusion of the project were recorded in the design specifications [Appendix D].

3.4.1 PROGRAMMING LANGUAGE

The programming languages considered for this project were Python, Java, JavaScript and PHP. These choices were selected due to prior experience working with Python, Java, and JavaScript in other courses and were strong choices when working with algorithms. Python was also a primary contender as it was a strong programming language when working with data. This was the original choice selected for the project but was ultimately changed to PHP due to integration with the frontend development of the project. PHP provides support for server-side scripting and easy deployment [6]. Having no prior experience with PHP was a challenge, so research had to be completed on syntax of the language as well.

3.4.2 WEBSITE LANGUAGE

JavaScript, Python, HTML, and CSS was considered for the frontend website language. JavaScript and Python were strong choices due to their extensive libraries allowing for easy integration; however, due to experience in a concurrent course, HTML and CSS were ultimately chosen to develop the website. These languages were essential structure and styling of web application content. It was important to consider how to handle interactivity and logic with a scripting language. This provided further research which was later handled with a framework.

3.4.3 DATABASE

MySQL, SQLite, and Firebase were databases evaluated and considered during the research aspect of this process. Limited prior experience with any type of database made this process challenging, so peers and faculty aided in determining potential options. Research was carried out from there independently. For the framework that was ultimately chosen, SQLite was the primary database used. This was not an ideal option upon research due to its limited scalability and concurrent write operations. MySQL was chosen over Firebase because it was reliable and scalable. Further research had to be completed to learn how to implement a database due to lack of experience working with databases.

3.4.4 FRAMEWORK

Initially, a framework was not considered. This was considered in the development process after facing challenges with integration of the front and back end of the web application. Peers and faculty aided in providing suggestions on potential frameworks which ultimately ended with the options of Flask, React, or Laravel. Flask was a top choice when working with Python as it offers flexibility in custom applications, but it created challenges with the deployment of the website as well as with integration. At this point, Laravel was selected and researched on implementation. Research also showed that it had built in tools for rapid development [3] This research led to the switch from Python to PHP for programming languages as Laravel supports PHP languages.

4. DEVELOPMENT

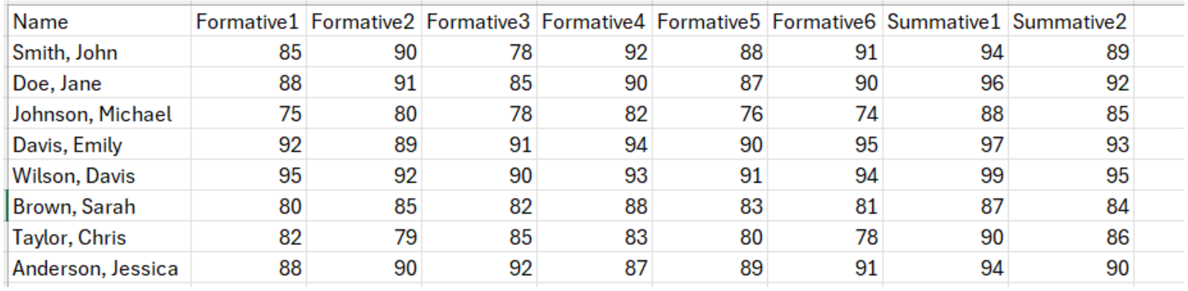
After completing the planning phase of this project, the development of the product took place. This process took place over the course of the semester after the conclusion of the initial research during the third week of the semester and continued until the first week of December. The product was developed using PHPStorm. The development process started with determining how to input the data source, then developing the algorithm to group students, followed by creating a user-interface, and then creating the database to store the groups. Throughout this process, testing of each aspect of the project took place and research continued to be conducted. The week before Thanksgiving, the deployment of the product was commenced.

4.1 DATA SOURCE

Before developing the algorithm, the data source had to be determined. To address all the grouping methods and considerations determined from the planning phase, two CSV files were used to upload assessment data and behavior data. Functionality of the data sources was of key importance when developing these files. After careful consideration of groupings, it was also determined that teacher preferences needed to be considered and input. However, this would be handled outside of a file.

4.1.1 ASSESSMENT FILE

Teachers input assessment scores into an online system at schools, so to make the website practical for a teacher, the assessment file would be downloaded as a CSV file from PowerSchool. PowerSchool is the platform that teachers at many school districts use to input student results, so it was a simple solution to save teachers time to simply download the file. The assessment file was a quantitative source that did not require any adjustments for determining student compatibility through an algorithm.

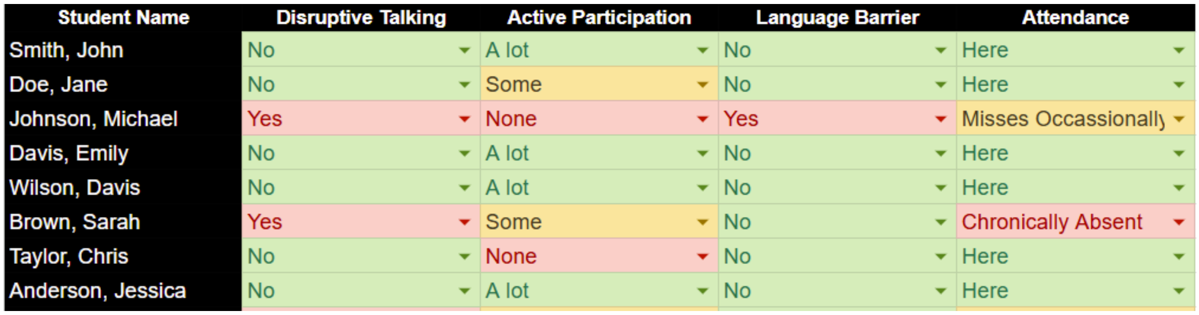


Name	Formative1	Formative2	Formative3	Formative4	Formative5	Formative6	Summative1	Summative2
Smith, John	85	90	78	92	88	91	94	89
Doe, Jane	88	91	85	90	87	90	96	92
Johnson, Michael	75	80	78	82	76	74	88	85
Davis, Emily	92	89	91	94	90	95	97	93
Wilson, Davis	95	92	90	93	91	94	99	95
Brown, Sarah	80	85	82	88	83	81	87	84
Taylor, Chris	82	79	85	83	80	78	90	86
Anderson, Jessica	88	90	92	87	89	91	94	90

Figure 1: This is a sample file of what an assessment file. The first column is the students' names with the format "LastName, FirstName." The other columns are the scores with the top row being the headers.

4.1.2 BEHAVIOR FILE

The behavior file is not something a teacher typically completes in their daily routine and most of this data is qualitative data that is observation based. To address this, a google sheet form was developed with a drop-down option of two to three choices for each category. This was an ideal option as teachers are familiar with google drive and sheets, and dropdowns allow for a quick selection. The file has four categories of information: disruptive talking, active participation, language barrier, and attendance. Each dropdown was preset to what would be expected of the average student, so teachers would not have to go through and select every student's data. They would only have to adjust the student that was above or below average. Having a dropdown option would allow simpler use for teachers but also allow for easier integration when converting from qualitative data to quantitative data during algorithm development. While this method is not quite as user friendly for the teachers, for the purpose of developing a base model of the product, this was the choice for the behavior file. Later development outside of the scope of this project would allow for the behavior file to populate on the user interface for input instead of as an extra file to download, fill out, and upload.



Student Name	Disruptive Talking	Active Participation	Language Barrier	Attendance
Smith, John	No	A lot	No	Here
Doe, Jane	No	Some	No	Here
Johnson, Michael	Yes	None	Yes	Misses Occassionally
Davis, Emily	No	A lot	No	Here
Wilson, Davis	No	A lot	No	Here
Brown, Sarah	Yes	Some	No	Chronically Absent
Taylor, Chris	No	None	No	Here
Anderson, Jessica	No	A lot	No	Here

Figure 2: This is a sample file of a behavior file. It has the students' names on the first column with the format "LastName, FirstName." The top row is the headers, and the drop-down options are color coded for easy reference for a teacher.

4.1.3 TEACHER INPUT

Teacher input was handled on the user interface instead of in a file format for ease of use for a teacher. A teacher can input as many preferences as they would like. For simplicity of coding design, a teacher is given three dropdown menus. The first dropdown menu is used for a teacher to select if they want a student pair to sit together or apart. Then the second two dropdown menus populate the list of students from the assessment file for the teacher to select the two students that they want to be together or apart. Originally this was set to where the teacher would type in the names, but this could lead to errors if the name is misspelled or not formatted the exact same as the assessment file names.

4.2 ALGORITHM

The algorithm was developed after the data sources were determined and developed. The algorithm was divided into two key components: the compatibility formula and the grouping process. The compatibility formula determines how compatible two students are to each other, and the grouping process uses the compatibility scores to place students into groups.

4.2.1 COMPATABILITY FORMULA

The compatibility formula takes in the data files and determines how well two students would be able to work together. This formula was set up using if statements to assign scores for each category of assessment scores, disruptive talking, active participation, language barriers, and attendance. Each category receives a score between zero and one. This is the part of the algorithm where many of the grouping considerations researched were used. For assessment scores, the top 25% of the class are categorized as the high students, the bottom 25% of the class are categorized as the low students, and everyone else is categorized as middle students. Students in the same category are given lower compatibility scores for the assessment portion than students in other categories. Students, where one was in the high and one was in the low category, are also given lower scores. This is to create balanced groups where the students are not so far from each other in understanding that they can still communicate and collaborate effectively with each other. Students are also assigned scores based on disruptive talking, language barriers, attendance, and participation. Assigning scores for behavior is not strictly based on one category but instead based on multiple category considerations. For example, a student who is a disruptive talker would be a good partner with someone with a language barrier, so this pair would receive a higher score than two disruptive talkers or two language barriers.

Once all considerations and category scores are assigned, the scores are multiplied by the weight of the category and added together to get the total compatibility score. Weights given are based on both research and observations from student teaching experience on what impacts grouping the most. Assessment scores received the largest weight for grouping. The methodology behind the if statements and weighted grouping scores aims to replicate the way a teacher would manually group students. The only remaining input for the algorithm to consider is the teacher preferences, but this was handled in the algorithm as these take precedence over compatibility scores.

4.2.2 GROUPING PROCESS

Once the compatibility formula was developed, the grouping process was developed to group students based on the compatibility scores. There were a lot of intricate details to consider that went beyond the scope of simply putting the students into groups. The first detail that had to be handled was group sizes. In the teacher preferences section of the user interface, a teacher would select the group size they wanted. The algorithm had to be able to handle the possibility of the class size not being a multiple of the group size. For example, if a teacher wants groups of four for a class size of twenty-one students. It would not be an ideal grouping outcome to have five groups of four students and one group of one student. The group of one student would not optimize collaboration in the classroom because the student would not be working with a group and would also not be having the maximum amount of interaction with other students. Instead, it would be better for four groups of three students and three groups of three students. This would produce the most full or close to full groups possible. The algorithm determines the number of full and smaller groups that needed to be formed.

The next step of the grouping algorithm was to incorporate preferred pairing first. This was to ensure that these students were placed together and would not require other students to be moved around after being placed in groups to accommodate together preferences. To do this, if neither student in the pairing was placed in a group, then both students would be placed in the first empty group available. If there were no empty groups left, they would be placed with another together pair. Because certain group sizes might prohibit preferred pairings, error handling warns the teacher that the together preferences would not work with the group size constraint. If a student was already placed in a group, the student not placed yet would be added to the group with the student from their together pair preference.

After honoring together preferences, a constraint-based greedy approach was used to group the remaining students. Empty groups were prioritized in the grouping process, so students would be placed in an empty group if there were any available. Once all the empty groups were taken up, then the algorithm determined the best group for a student. To find the most suitable group for a student, the algorithm was developed to check the apart preferences and not consider that group option, and then to check the average compatibility score for a group with the addition of the student. That student is placed in the group for which he/she most boosts the group compatibility score. Once all the students are placed in groups, the results are stored in a JSON file and displayed on the user interface. The JSON file allows for storage in the database and can be used to download later.

4.3 USER INTERFACE

The user interface is the web application. This part was developed after forming the algorithm. The website consists of five main pages: home, why group, generate groups, files, and admin. The home page is a welcome page to let users know about the website and what the other pages on the website are about. The group page displays information about grouping methods and things to consider when grouping. It also includes some resources for groupwork and assignments. Its purpose is to show some of the research done for grouping, but also to serve as a resource to teachers that have not had a lot of experience with groupings. The bulk of the user interface, however, is the generate groups page, files page, and admin pages.

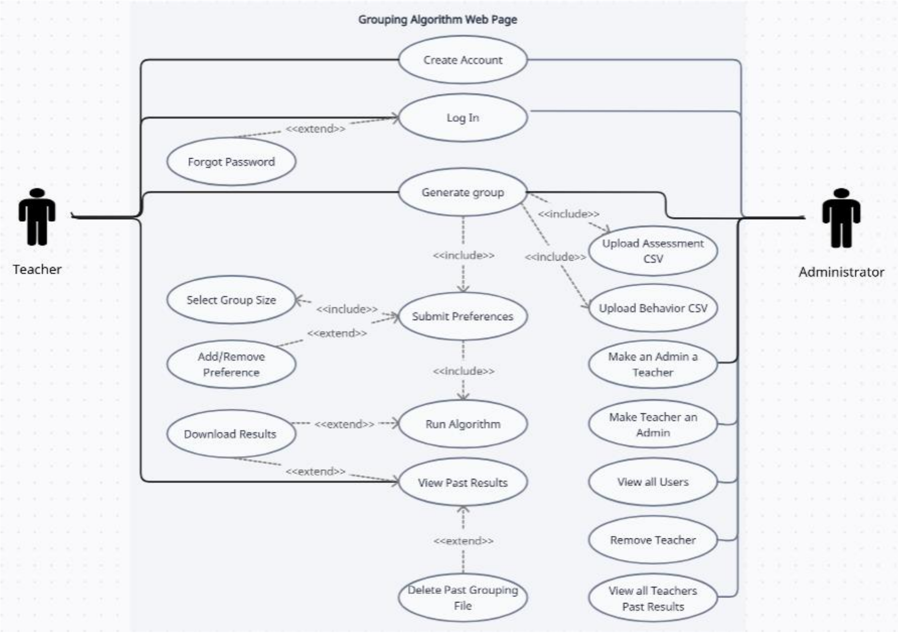


Figure 3: The use case diagram outlines the functionality that the teacher and administrator has on the web application.

The generate groups page walks through uploading the assessment file, behavior file, and teacher preferences to generate groups. Support dropdown options are provided to help new users navigate this page. The first part of this page is the CSV file upload for the assessment and behavior file. It also has a link to download the behavior file to fill out to upload. The next part once submitting those documents is the preferences page where the desired group size is input and any preferences on specific students being together or apart are added. This page is where most of the error handling is covered to prevent the algorithm from not grouping a student. It provides user feedback if the together or apart preference violates the group size or if the same student pair is selected as together and apart. Then once completing all the input that the algorithm needs to run, the final page is created to display the groupings that the algorithm corrected in a table. A download link is also created to allow the teacher to download the groupings with specific information on the compatibility of the students in each group. This is done using the JSON file that was saved to the database when the algorithm was created.

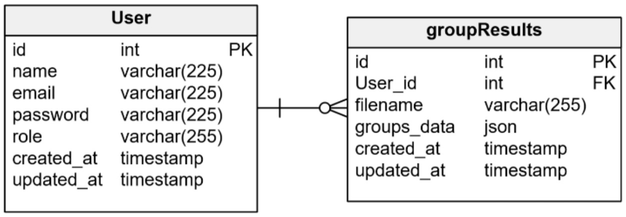
The files page allows teachers to reference past group results. They can download the grouping file with information on each group compatibility, rename the grouping file, or delete a file. This page is created concurrently with the database because it pulls information from the database.

The last set of pages created were the administrative pages. The goal is to give admins control of users' roles like adding a new admin or demoting an admin to a teacher. They can also delete a user if they are no longer using the site to help with maintenance of the website when there are a lot of users. Another admin page gives a view of all groupings created on the website. The role of the admin is to oversee and maintain the upkeep of the website, so these were the only pages added for the admin.

The website was created with a login page, forgot password page, and create account page because it does have a database. This was important to add so the teacher can reference their past groupings and store the information in the database. To do this, the login page and forgot password page send emails to verify emails and reset passwords, which was set up in the deployment phase.

4.4 DATABASE

The database is used to store past groupings and user information such as their emails and password. The database was originally an SQLite database because this was the default database setting for Laravel. In the research phase, it was determined that the MySQL was the better choice for database, so the database had to be set up as a MySQL. The focus for the database is to make sure that there is no sensitive student information being stored. This prevents any issues with security as the only thing stored are the JSON files with compatibility scores and student grouping but without student assessment or behavior scores. Other than files, user information is stored for login purposes of the website.



User		
id	int	PK
name	varchar(225)	
email	varchar(225)	
password	varchar(225)	
role	varchar(255)	
created_at	timestamp	
updated_at	timestamp	

groupResults		
id	int	PK
User_id	int	FK
filename	varchar(255)	
groups_data	json	
created_at	timestamp	
updated_at	timestamp	

Figure 4: The entity relationship diagram (ERD) shows what information is stored in the database and the relationship between the two tables stored in the database. A user has a one to many relationship with groups, meaning a user can have many group results.

4.5 DEPLOYMENT

Deployment of the project took place the week before Thanksgiving through the first week of December. It was important to make sure that all permissions were given to be able to read and write from a file. The project was hosted at the University of Mississippi's Turing system. This required meetings with faculty to get permissions set up. From there, the project had to have the email system set up to verify emails when creating an account and sending a forgotten password link. Also, the database had to be set up and linked to the web application again to be used on Turing. When switching from local hosting to Turing, there were some parts of the project that had to be adjusted, but these adjustments were minor.

5. CLOSING

The closing phase consists of creating a user manual for the web application, writing a final report, and giving an oral presentation. The user manual was created as the project was developed detailing the dependencies for, how to use, and administer the web application [Appendix E]. After the project submission, an oral presentation was given to five computer science faculty members to defend the project for results and a final report was written. The final report detailed project goals, the successes and challenges of the project, opportunities for improvement, lessons learned from the project, and future enhancements [Appendix F]. The oral presentation detailed the computer science aspects of the final product and included a demo of the product [Appendix G]. The demo of the product walked through each page of the website, and then ran the grouping algorithm using sample files [Appendix J-K]. This allowed the faculty to see the website and ask any questions that were not clarified in the presentation or in the documentation.

5.1 USER MANUAL

One of the most important aspects of this project was developing a product that could be implemented by a company. The user manual was a vital aspect to make this product accessible for a company by breaking down the dependencies, use, administration, and maintenance of the web application. The use section of the manual featured visual step by step directions with how to use the functionality components of the web application such as how to upload files, input teacher preferences, and generate groups. It also provided maintenance and administrative descriptions of how to keep the web application up to date as well as enhancements that could further improve the website in the future.

5.2 EVALUATION PLAN

Since this algorithm has not been tested with real classroom data and only sample files, an evaluation plan was formulated for how to evaluate the algorithm and user interface before deployment on a large scale [Appendix H]. The plan would evaluate five target areas: saving, engagement, academic impact, teacher attitude, and student perspective. It would be implemented over one academic semester through three phases: the pre-semester set-up, data collection, and post-semester evaluation. This pre-semester set-up would set a baseline for the evaluation through teacher and student surveys. The post-semester evaluation would be the same as the pre-semester survey but asking how the web application impacted grouping practices. This data would be compared to the baseline to determine how well it optimized classroom performance. The data collection stage would be implemented through classroom observation and activity feedback forms to get data for each groupwork activity completed instead of just an overall end of year recap of grouping, and it would also allow for an outside observer's perspective of the grouping practice.

5.3 FUTURE ENHANCEMENTS

This project is the beginning framework of what could be a powerful tool to impact teaching practices in the classroom. It is currently limited to just creating heterogeneous groupings, and the interface could be streamlined to make the web application more accessible and user friendly. Some of the enhancements for this project are stated in the final report that could be implemented in the near future, and other enhancements that would be more challenging to implement are stated in the user manual.

6. CONCLUSION

The purpose of the senior project was to address a problem using computing skills developed while at the university. For this specific project, the web application with an algorithm to group students aimed to streamline the process of grouping students, optimizing student groups based on academic performance, behavior, attendance, language barriers, and teacher preferences. This project was developed through phases. During the initiation phase, a product was proposed to address a problem. During the planning phase, potential design choices and grouping methods were researched and assessed. During the development phase, a student grouping algorithm was developed and implemented through a web application user interface. The final phase was the closing phase where the product was delivered with a final report and oral presentation. The minimum viable product was successfully completed and an evaluation was developed to determine if the product optimizes student groups.

BIBLIOGRAPHY

[1] Cohen, E., & Lotan, R. (2014). *Designing groupwork: Strategies for the heterogeneous classroom (3rd ed.).* Teachers College Press.

[2] Kagan, S. (1989). *Cooperative learning resources. Educational Leadership, 47(4), 12–19.* Retrieved from https://files.ascd.org/staticfiles/ascd/pdf/journals/ed_lead/el_198912_kagan.pdf.

[3] *Laravel—The PHP Framework For Web Artisans*. (n.d.). Retrieved November 7, 2024, from https://laravel.com/

[4] Ma H, Li J, Zhu H, Tang W, Huang Z, and Tang Y. (2024). *Collaborative Optimization of Learning Team Formation Based on Multidimensional Characteristics and Constraints Modeling: A Team Leader-Centered Approach via E-CARGO*. Retrieved November 7, 2024, from https://doi.org/10.1109/TCSS.2022.3224762.

[5] *MySQL Tutorial—Learn MySQL Fast, Easy and Fun.* (n.d.). MySQL Tutorial. Retrieved November 7, 2024, from https://www.mysqltutorial.org/

[6] *PHP: Introduction - Manual*. (n.d.). Retrieved November 7, 2024, from https://www.php.net/manual/en/introduction.php

APPENDICES

APPENDIX A. CAPSTONE PROJECT FORM

UNIVERSITY OF MISSISSIPPI – UNDERGRADUATE CAPSTONE PROJECT SURVEY

Thank you for your interest in participating in a University of Mississippi Capstone Project Senior Design Course. This non-binding survey is designed to help communicate expectations among the Student(s), Client(s), and UM Faculty. Additionally, this survey will assist the University in identifying and preparing, if needed, necessary agreements. This survey should be completed and reviewed by all involved parties.

Course Information Course Name: Senior Project Department: Computer and Information Science Course number(s): CSCI 487	Faculty Instructors: Joseph Carlisle Timothy Holston Faculty Email: jcarlis1@olemiss.edu tlholsto@olemiss.edu Semester(s): Fall 2024
Student Information Name: Kathryn Reardon Year: Senior Major: B.A. Computer Science, B.S. in Mathematics, B.A.Ed in Math Education	Address: 900 Whirlpool Dr Apt 9202, Oxford, MS 38655 Phone: (501)388-7813 Email: kreardo1@go.olemiss.edu
Client Information Client/Sponsor Name: Dr. Yixin Chen Title: Chair and Professor Project Coordinator (if not Client): n/a Title: n/a	Address: 203 Weir Hall, University, MS 38677 Phone: (662)915-7309 Email: ychen@cs.olemiss.edu
Project Title and Brief Description - attach additional description page as necessary: (***CSCI 487 students**: The Brief Description should be 5 to 7 sentence paragraph – preferable typed below*) The project is titled “Optimized Student Grouping for Enhanced Classroom Performance.” A crucial component of adopting a productive classroom discussion and improving overall classroom performance and functionality is grouping students. There are so many factors to consider when grouping and traditional methods of grouping tend to be time consuming, limited and subjective. This project would aim to provide data driven results to help teachers establish balanced and effective student groups based on mathematical models that address student interaction and behavior as well as academic achievement. This project will employ the stable marriage algorithm to group students using a compatibility matrix determined by a scoring formula that considers criteria such assessments, behavior, and teacher constraints.	
☐ **Team Project** ☒ **Individual Project**	
Project Goals and Deliverables (“Student Project Results”) - include desired format: report, code, prototype, CAD models, etc. – Attach additional pages as necessary: (***CSCI 487 students**: State your MVP – preferably typed below*) The minimum viable product is a web application using the stable marriage algorithm to group students using a compatibility matrix based on student data and teacher input. It will store each week’s grouping to manage for later use. The deliverables will include a report outlining the algorithm’s design and implementation, source code of the program, and an evaluation plan for determining the effectiveness of the algorithm.	
FOR UM CLIENTS ONLY: **Are any sponsored research funds tied to this project?** **If yes, where from and how much?**	

1. **Confidentiality.**

The University prefers that clients make reasonable efforts to avoid the inclusion of confidential information into student academic course projects, which take place in an open and collaborative academic learning environment. However, it is understood that there may be situations where a client wants to share information that is to be held in confidence. If so, the client, the University, and participating students should sign a separate Non-Disclosure Agreement (“NDA”).

Please check the appropriate box below:

☒ Client WILL NOT share any confidential information during the Project and will not need an NDA.

☐ Client WILL share confidential information during the Project and will need an NDA. Please contact Allyson Best, Director of Technology Management in the ORSP – amilhous@olemiss.edu

2. **Student Intellectual Property Rights (if applicable).**

Background:
In the event that commercializable intellectual property is created, student's rights must be considered. While the University encourages clients to permit student participants to retain their intellectual property rights developed under the client projects, it is understood that there may be situations where the client wants to obtain ownership or license rights from the student participants. Students are never obligated to participate in projects or activities that require the assignment of the student's intellectual property to the University or to another entity. In this course, students are presented with two options: (1) to participate in projects or activities that do not require the student to license/assign their intellectual property or (2) to participate in projects or activities that require the student to license/assign their intellectual property.

Please check the preferred potential outcome for Student Project Results:

☐ No expected commercializable IP.

☒ Student Retains Student IP. – Under this option, Client will be provided with the Student Project Results for Client's non-commercial internal review and analysis only. Any and all rights to the Student Project Results, including all intellectual property rights, if any, shall remain with the individual Student Participants under appropriate University policies regarding rights to and ownership of intellectual property. Prior to any commercial use or subsequent transfer of any Student Project Results, Client must obtain the appropriate rights from the respective Student Participant owners.

☐ Student Provides a NERF to Client. – The Student Participants will grant Client a non-exclusive royalty free commercial license ("NERF") from the students to use the intellectual property in the Student Project Results. Please contact Allyson Best, Director of Technology Management in the ORSP at amilhous@olemiss.edu

☐ Client Owns Student IP. – The Student Participants will assign their intellectual property rights in the Student Project Results to Client. Please contact Allyson Best, Director of Technology Management in the ORSP at amilhous@olemiss.edu

Related University Policies

- Copyright (Intellectual Property) – RSP.TM.400.001
- Patents and Inventions – RSP.TM.400.003

UM CLIENTS ONLY

1. Student Academic Research in Potential Publication(s)

☒ No academic research in project is anticipated.

☐ Academic research is possible and student acknowledgment/authorship will be determined accordingly.

Submitted by:

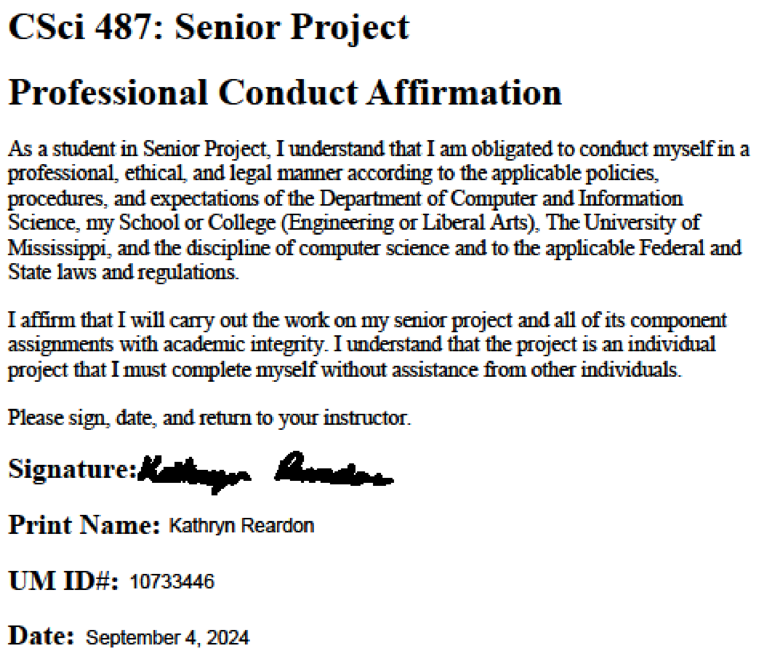
Yixin Chen________________ 9/5/2024___________________
AUTHORIZED SPONSOR/CLIENT NAME Date

Joseph Carlisle 8/26/2024
UM FACULTY INSTRUCTOR NAME Date

Timothy Holston 8/26/2024
UM FACULTY INSTRUCTOR NAME Date

Kathryn Reardon 8/28/2024
STUDENT NAME Date

APPENDIX B. PROFESSIONAL CONDUCT STATEMENT



CSci 487: Senior Project

Professional Conduct Affirmation

As a student in Senior Project, I understand that I am obligated to conduct myself in a professional, ethical, and legal manner according to the applicable policies, procedures, and expectations of the Department of Computer and Information Science, my School or College (Engineering or Liberal Arts), The University of Mississippi, and the discipline of computer science and to the applicable Federal and State laws and regulations.

I affirm that I will carry out the work on my senior project and all of its component assignments with academic integrity. I understand that the project is an individual project that I must complete myself without assistance from other individuals.

Please sign, date, and return to your instructor.

Signature:

Print Name: Kathryn Reardon

UM ID#: 10733446

Date: September 4, 2024

APPENDIX C. PROSPECTUS

Optimized Student Grouping for Enhanced Classroom Performance Prospectus

Kathryn Reardon | kreardo1@go.olemiss.edu

Dr. Yixin Chen (Sponsor) | ychen@cs.olemiss.edu

Problem: Traditional student grouping methods are time consuming and depend on subjective criteria derived from student scores and teacher observation but are inconsistent in producing effective group dynamics. This project aims to address these challenges by employing a data-driven algorithm to produce optimal student groups based on behavior, attendance, and academic performance.

Selection: As a student teacher, I am responsible for grouping students for each daily lesson based on criteria such as assessments, student behavior, language barriers, and classroom observation. In the 90-minute planning block, student teachers spend 30 minutes on average grouping the class, which often results in suboptimal group dynamics and limits time for lesson preparation. I initiated this project to develop a more effective and efficient data-driven algorithm to optimize student grouping, which in turn allocates more time towards planning meaningful instruction.

MVP: The minimum viable product is a web application that groups students based on student data and teacher input, and it will store generated groupings for later reference. The deliverables will include a report outlining the algorithm's design and implementation, source code of the program, and an evaluation plan for future work to determine the effectiveness of the algorithm.

Risks/Challenges: There are numerous risks associated with this project to include the use of hypothetical data due to permission constraints that may require adjustments when integrated to real student data. There are also challenges with assimilating quantitative assessment scores and qualitative behavior into a weighted compatibility formula to accurately depict student compatibility. Further, adapting the stable marriage problem to group sizes larger than pairs, incorporating teacher preferences, and ensuring scalability while maintaining algorithm effectiveness and accuracy will be a complex task to overcome.

User Types: Teachers will use this product to create optimized student groups and enhance classroom performance and collaboration. School administration could adjust groupings to meet educational goals, align with a student's Individualized Education Program, and refine the algorithm. While not direct users, students will benefit from improved groups.

Tasks/Timeline: To complete this project, I will create a hypothetical dataset based on realistic student data. Then, I will develop a student compatibility matrix using a scoring formula with weighted criteria. Next, I will develop a stable marriage problem algorithm to group students based on the matrix. I will incorporate functionality for teacher constraints and input. The program will be tested and refined using the hypothetical dataset. Finally, I will formulate an evaluation plan to assess the algorithm's effectiveness and efficiency with real student data. My aim is to complete one task each week, but the weeks of constructing the algorithm and scoring formulas may require a week and a half. My goal is to keep this pace, but I will also allow flexibility in this timeline to ensure a quality product.

APPENDIX D. DESIGN SPECIFICATION

Addendum to Design Specification Document

Date: December 8, 2024
Project Title: Optimized Student Grouping for Enhanced Classroom Performance

Updates to the ERD (Entity-Relationship Diagram)

Change Description: The ERD now includes an updated groups_data entity that stores JSON files containing group configurations and related data. This update was made to improve the system's data organization and allow for easier retrieval of past grouping results.

Impact on Design: This improved the scalability of storing and retrieving group-related data. The file is saved as a JSON file to eliminate the dependency of local file storage for group data.

Updates to the User Interface and Use Case Diagram

Change Description: The user interface and use case diagram have been updated to reflect the expanded role of administrators. Specifically, administrators can now perform all actions available to teachers. Additionally, administrators have the capability to view past results of teacher groupings but not delete other user’s groupings and promote teachers to admins or demote admins to teachers.

Impact on Design: The updated design consolidates shared functionalities for teachers and administrators while highlighting the exclusive privileges of administrators. This ensures a streamlined interface for both roles and simplifies role-based access control for administrators.

Reason for Changes

These updates were implemented to improve the system’s scalability, flexibility, and alignment with project goals. By eliminating local file storage dependencies for group data and enhancing role-based functionality for administrators, the system is better positioned to support long-term usability and maintainability.

Updated Diagrams

The revised diagrams have been updated in this document:

1. **Updated ERD**: Incorporates the groups_data entity for directly storing JSON data.
2. **Updated Use Case Diagram**: Reflects the shared teacher-admin functionalities and exclusive admin privileges.

This addendum reflects all changes made since the initial design specification document and ensures alignment with the updated system requirements.

Optimized Student Grouping for Enhanced Classroom Performance
Design Specification

Kathryn Reardon | kreardo1@go.olemiss.edu

Dr. Yixin Chen (Sponsor) | ychen@cs.olemiss.edu

Project Overview: Grouping students effectively is time-consuming and often relies on subjective criteria, which can lead to inconsistent classroom dynamics. As a student teacher, I spend valuable planning time on this task, limiting my lesson preparation. This project aims to develop a data-driven algorithm that uses behavior, attendance, and academic performance to create optimal groups, enhancing collaboration and freeing up time for more meaningful instruction.

User Requirements: The minimum viable product (MVP) is a web application that groups students based on student data and teacher input, and it will store generated groupings for reference. Deliverables will include a report outlining the algorithm's design and implementation, the source code of the program, and an evaluation plan for future work to determine the effectiveness of the algorithm. The algorithm will be a derivation of an algorithm from Combinatorics.

The users of this website will be teachers and administrators. User stories for these roles include:

> As a teacher, I want to create a group seating chart so that I can foster a collaborative learning environment for group work.
>
> As a teacher, I want to group students so that I can have well-balanced groups for a group project.
>
> As an administrator, I want to maintain and upkeep the web application so that the website stays up to date.
>
> As an administrator, I want to be able to view all users and their groupings and delete past groupings and users as necessary.

Design Choices: This project needs a programming language, database, framework, and file type that is chosen based on performance, usability, and scalability.

Programming Languages for Algorithm: Python, Java, JavaScript, and PHP were evaluated. Python is ideal for fast development with its extensive data manipulation libraries and readable syntax, making it suitable for quick prototyping and efficient handling of data-centric tasks. Although it has slower execution compared to compiled languages, its flexibility makes it a strong choice. Java offers robust performance, platform independence, and high reliability as a compiled language. However, it comes with more complex syntax and higher memory consumption. JavaScript is dynamic and essential for web-based applications. It supports asynchronous programming but poses challenges in managing it effectively and has slower execution than Java. PHP, known for its simple syntax and extensive support for server-side scripting, is easy to learn and deploy. While it's effective for web-related tasks, it is less efficient for complex algorithmic operations and has known security concerns.

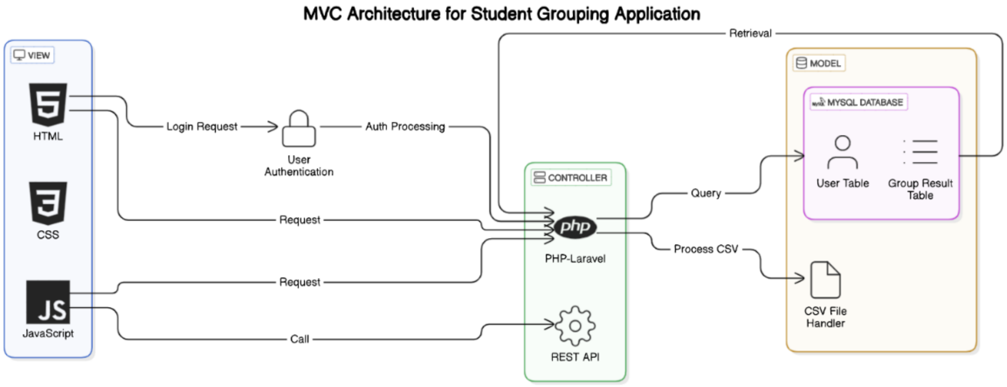
Website Language: JavaScript, Python, HTML, and CSS were considered. JavaScript remains the top choice for creating dynamic, interactive user interfaces, supported by powerful libraries like React and Vue that facilitate rapid development and responsiveness. However, handling complex asynchronous programming can be challenging. Python, known for its readable syntax, can be used in the front-end through tools like Brython but often needs JavaScript integration, limiting its standalone application. HTML and CSS are essential for structuring and styling web content but must be combined with a scripting language to handle interactivity and logic.

Database: MySQL, SQLite, and Firebase were evaluated. MySQL is an industry standard with proven reliability for managing large data volumes and handling complex queries, making it ideal for scalability and high-read applications. However, it requires careful schema design due to its rigid structure. SQLite, a lightweight and serverless database, is easy to implement and suitable for small projects with minimal setup, but it struggles with scalability and concurrent write operations. Firebase, a NoSQL, cloud-based database, stands out for real-time data synchronization and automatic scaling, making it a strong option for applications that need immediate updates. Nonetheless, its dependency on the Google ecosystem can raise long-term concerns, and it lacks the flexibility of SQL in complex data retrieval.

Frameworks: Laravel, React, and Flask were considered. Laravel, a PHP-based framework, offers robust built-in tools for rapid development with a strong MVC structure but is not as lightweight as other options. React, a JavaScript library for dynamic UIs, supports reusable code and full-stack development but requires familiarity with JSX and hooks. Flask, a Python-based microframework, offers flexibility for custom applications but requires more setup compared to other frameworks.

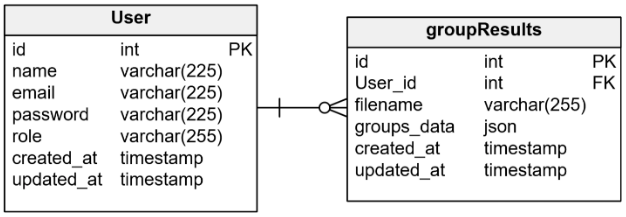
File Type: CSV and Excel were evaluated. CSV files are simple, lightweight, widely supported, and ideal for school systems but lack complex data types and can lead to parsing errors. Excel files are user-friendly with advanced analysis features but are less portable and require specific handling libraries.

Design Selection:



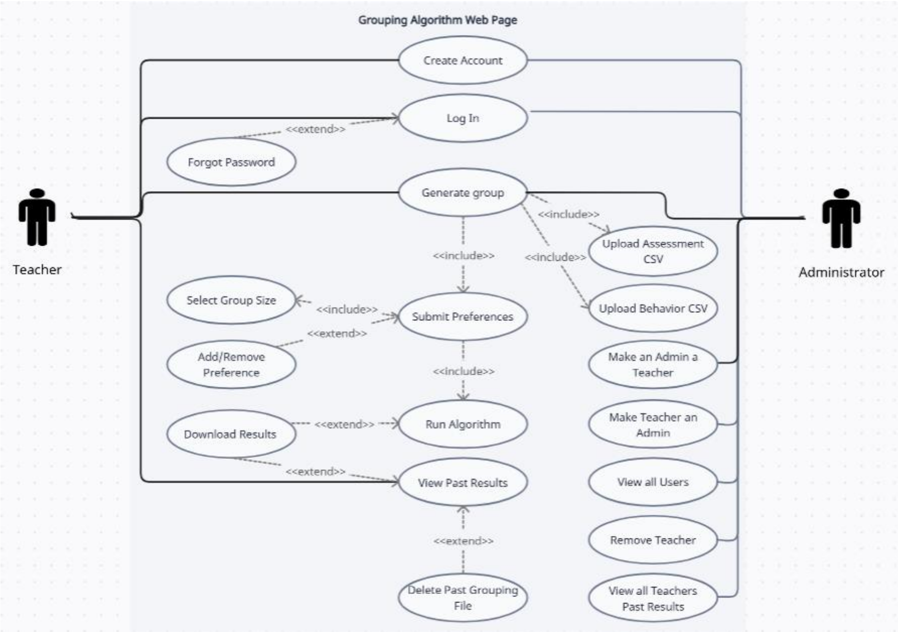
PHP was chosen as the main programming language for my project due to its strong support for web applications, ease of use, and extensive community resources, making it ideal for a responsive, server-side solution. Basic knowledge of this language was acquired through w3schools tutorials. Laravel, a PHP framework, enhances this choice with its robust MVC architecture, built-in Rest API, and built-in tools like authentication and database management, ensuring rapid development and a maintainable codebase. The Laravel 11.x documentation for guidance was used for reference and guidance. This combination supports creating an efficient, scalable platform for teachers to input data and manage group results. Laravel's structured approach, while resource-intensive, is worth it for its developer-friendly environment and scalability.

For the database, MySQL was selected due to its reliability and scalability. Its structured design and fast data retrieval ensure secure, efficient management of user information and group results. The database was developed using "[MySQL Tutorial - Learn MySQL Fast, Easy and Fun"] as a reference. The ERD outlines how user and group result data are stored, prioritizing student data security by processing behavior and assessment files directly without storing raw data. Users have authentication information and the role of administrator or teacher. The group results include the filename, group data, and user identification, establishing a one-to-many relationship—a user can have multiple group results, but each result is linked to one user.



For the front end, HTML and CSS were chosen for structure and styling, creating a clear, appealing interface. JavaScript was added for interactivity and responsiveness, improving user experience with real-time feedback and dynamic content. CSV files were chosen for data uploads and downloads due to their ease of use for teachers. This front-end setup, combined with the PHP-Laravel back end, provides a comprehensive full-stack solution tailored for effective, secure student grouping.

User Interfaces:



The web application provides an intuitive, user-friendly interface for both teachers and administrators. Teachers can log in using their email and password or create a new account by entering their name, email, and password. A password recovery option is available for resetting forgotten credentials. Once logged in, teachers can access past group results and use the grouping algorithm by uploading assessment and behavior CSV files. They can set group sizes and customize student pair preferences (together or apart) before submitting. The algorithm generates a table of student groupings, downloadable as a CSV file with compatibility insights. Teachers can also manage past results by viewing, downloading, renaming, or deleting files through the "File" page, and access best practices and resources on group formation through the "Why Group?" page. Administrators can do everything a teacher can do and can view all teacher accounts and associated group data. They can also promote a teacher to an admin or demote an admin to a teacher and delete a user. The application is designed to be responsive, ensuring consistent experience across different device sizes.

Development Environment: The development environment for this project primarily used PHP Storm as the integrated development environment (IDE), chosen for its comprehensive support for PHP, built-in version control integration, and ease of use. PHP Storm's features streamlined the development process with tools for code inspection and efficient debugging. The project was developed on a Windows operating system, using Git within PHP Storm for version control to manage code changes and ensure backups at key functional milestones. Unit testing and manual testing through log statements and the dd function were performed to verify data flows and page functionality. Sample CSV files were created manually to test assessment and behavior data handling for different class sizes, ensuring the algorithm's robustness.

Deployment Environment: The project is deployed on the University of Mississippi's Turing service, a reliable platform for hosting academic web projects. It is a stable environment for deploying PHP-based applications without additional security measures. A MySQL account was created on Turing to handle the application's database requirements, supporting data management and retrieval. The Laravel framework must be configured to run on this service, requiring basic PHP and Laravel setups to ensure smooth operation.

Test Plan: The test plan for this project will use a layered strategy to validate individual components and overall system functionality. Unit testing will be performed for major components of the algorithm, including the compatibility formula and group generation, to ensure their correctness. Log statements and the dd function will be used separately to check each step of development and identify errors in the code. End-to-end testing will involve running the complete application with sample CSV files to validate the seamless flow from file upload to result in file generation and download without storage. Responsive testing will be conducted using the browser's inspect tool to simulate various devices with mobile views prioritized. This will confirm consistent user experience. External user testing will include having new users interact with the application to ensure it is intuitive and functional and providing feedback on usability. Additionally, a future works document will outline an evaluation plan to assess the grouping algorithm's real-world impact, focusing on time savings, group productivity, student collaboration, and classroom optimization over the next semester.

Project Timeline:

November 10: Implementation of the "Forgot Password" feature.

November 17: Creation of an administrator user.

November 22: Addition of error handling to the algorithm and preferences.

November 25: Preparation of the "Future Works" document.

December 1: User manual due.

December 2-6: Final oral presentation (presentation order to be announced).

December 8: Final report and product delivery due.

December 9-11: Exit interviews, exit survey, and sponsor evaluation form to be completed.

References

Creately. (n.d.). Retrieved November 8, 2024, from https://app.creately.com/d/

Dev.java: The Destination for Java Developers. (n.d.). Dev.Java: The Destination for Java Developers. Retrieved November 7, 2024, from https://dev.java/

Eraser – AI co-pilot for technical design. (2024, November 8). https://app.eraser.io/dashboard/all

Firebase | Google's Mobile and Web App Development Platform. (n.d.). Retrieved November 7, 2024, from https://firebase.google.com/

Getting started | PhpStorm. (n.d.). PhpStorm Help. Retrieved November 7, 2024, from https://www.jetbrains.com/help/phpstorm/getting-started.html

Laravel—The PHP Framework For Web Artisans. (n.d.). Retrieved November 7, 2024, from https://laravel.com/

MySQL Tutorial—Learn MySQL Fast, Easy and Fun. (n.d.). MySQL Tutorial. Retrieved November 7, 2024, from https://www.mysqltutorial.org/

PHP: Introduction - Manual. (n.d.). Retrieved November 7, 2024, from https://www.php.net/manual/en/introduction.php

PHP Tutorial. (n.d.). Retrieved November 7, 2024, from https://www.w3schools.com/php/

Vertabelo—Design Your Database Online. (n.d.). Vertabelo - Design Your Database Online. Retrieved November 8, 2024, from https://my.vertabelo.com/drive

Welcome to Flask—Flask Documentation (3.0.x). (n.d.). Retrieved November 7, 2024, from https://flask.palletsprojects.com/en/stable/

Welcome to Python.org. (2024, November 7). Python.Org. https://www.python.org/

What is JavaScript? - Learn web development | MDN. (2024, August 3). https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

APPENDIX E. USER MANUAL

Collab Connect User Manual

Optimized Student Grouping for Enhanced

Classroom Performance

Kathryn Reardon | kreardo1@go.olemiss.edu

Dr. Yixin Chen (Sponsor) | ychen@cs.olemiss.edu

Table of Contents

I. PROJECT OVERVIEW

This product is a Laravel-based web application designed to group students based on assessment data and behavior data as well as teacher input. This web application aims to optimize performance in the classroom by providing heterogeneous student groups that support a studentcentered classroom where students support students. The web application is located at https://turing.cs.olemiss.edu/~kreardo1/seniorProject/collabConnect/public/index. This manual can be used for people who want to create their own version of the application or users of the Collab Connect web application.

II. DEPENDENCIES

II.I OPERATING ENVIRONMENT/SYSTEM

This project was developed on a Windows operating system and is compatible with Linux or macOS systems with minimal modifications. It requires a PHP version of 8.2 or higher and MariaDB version 10.3 or higher as the database system. The web server used is Apache, but Nginx can also be used.

For frontend asset compilation, Node.js and npm must be installed. Linux and macOS users need to ensure proper file permissions are set for Laravel's Storage and Bootstrap/Cache directories. On all systems, the environment should have PHP extensions including ctype, cURL, DOM, fileinfo, filter, hash, mbstring, OpenSSL, PCRE, PDO, session, tokenizer, and XML enabled.

For further information on the server and database requirements, refer to the Laravel documentation here: Server Requirements and Database Requirements.

II.II SOFTWARE, TOOLS, AND FRAMEWORKS DEPENDENCIES

The backend is developed using Laravel (10.x), a PHP framework, and the frontend interface uses HTML, CSS, and JavaScript. Composer is used to manage PHP dependencies, and Node.js is employed for managing frontend assets. The database system is MariaDB, a drop-in replacement for MySQL, which stores users and grouping data. The project requires two CSV file inputs:

- Assessment File: Exported from PowerSchool, containing student names and assessment data.
- Behavior File: Teachers access this file via a link on the website. Clicking the link forces a copy of a Google Spreadsheet, where teachers paste the student names from the assessment file. Teachers manually fill out behavior information for each student in the spreadsheet and download it as a CSV file for upload to the application.

II.III BUILDING THE PRODUCT LOCALLY

To build a project locally, download the repository using GitHub and navigate to the project folder on your local machine. Once inside the folder, install the necessary backend dependencies by running composer install, and then install the required Node.js packages for the frontend assets by executing `npm install`. This will ensure the backend and frontend dependencies are set up for the application.

```
DB_CONNECTION=sqlite
# DB_HOST=127.0.0.1
# DB_PORT=3306
# DB_DATABASE=laravel
# DB_USERNAME=root
# DB_PASSWORD=
```

Next, configure the environment. Copy the provided `.env.example` file to `.env`. Edit the `.env` in a text editor to set up the database credentials, specifying the database name, username, and password that corresponds with the local MySQL or MariaDB installation. For example, if your database is named “laravel_project” and your credentials are “laravel_user” with the password “laravel_password”, update the .env file accordingly. Ensure the `DB_CONNECTION` in the `.env` file is set to mysql for MariaDB or MySQL, instead of the default sqlite. Once the environment configuration is complete, proceed to set up the database. Using MariaDB, create a new database and user for the application. Log into MariaDB and execute commands to create the database, assign the user privileges, and flush them. With the database ready, run the Laravel migrations using `php artisan migrate` to create the necessary tables, and seed the database using the command `php artisan db:seed`.

To enable email functionality for features like password resets or notifications, configure Laravel’s mailer settings to use Gmail. To enable Laravel’s email functionality, configure the mailer settings in the .env file as follows:

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME= gmail_address@gmail.com
MAIL_PASSWORD=gmail_app_password
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS= gmail_address@gmail.com
MAIL_FROM_NAME="${APP_NAME}"
```

Replace `gmail_address@gmail.com` with your actual Gmail address and `gmail_app_password` with an app password generated from your Google account. App passwords must be used instead of your normal Gmail password for security. To generate a Gmail app password, log in to your Google account, navigate to the Security section, and enable '2-Step Verification if not already enabled. Then, under 'App Passwords,' generate a new password specifically for this application or set your app password here: Create and manage Gmail app passwords.

During development, execute `npm run dev` to compile the frontend assets in development mode. For a production-ready build, execute `npm run build` to generate optimized assets. Once complete, start the Laravel development server by running `php artisan serve --port=9000`. Open a browser and navigate to `http://localhost:9000` to access the application locally. If port 9000 is in use, replace it with an unused port number.

II.IV BUILDING THE PRODUCT EXTERNALLY

When deploying the project to an external server, ensure the server meets the required specifications, including PHP 8.2 or higher, MariaDB, Composer, Node.js, and the necessary PHP extensions. To install the required PHP extensions, use the following command:

```
sudo apt install php-cli php-ctype php-curl php-dom php-fileinfo
phpmbstring php-openssl php-pdo php-tokenizer php-xml php-session
phpbcmath php-filter
```

Transfer the project files to the server. Once the files are on the server, SSH into the server and navigate to the project directory. From there, run `composer install` to install the backend dependencies. For frontend assets, install the Node.js packages, using `npm install` and compile the assets using `npm run build`.

Next, configure the environment by copying the `.env.example` file to `.env` and updating it with the server's database credentials. Configure the application URL to ensure the application works correctly in the deployed environment. In the `.env` file, set the `APP_URL` value to the server’s domain or IP address where the application is hosted. For example, `APP_URL=http://yourdomain.com`. This ensures correct URL generation for emails, asset links, and redirects.

After configuring the environment file, set the appropriate file permissions for Laravel’s storage and bootstrap/cache directories. Use commands `chmod -R 775 storage bootstrap/cache` and `chown -R www-data:www-data storage bootstrap/cache` to grant the necessary permissions and ownership.

To prepare the database on the server, create a new database and user in MariaDB, then run the

Laravel migrations with `php artisan migrate` followed by `php artisan db:seed`. After completing the database setup, configure the web server. For Apache, create a virtual host file that specifies the server name, document root, and directory permissions. Enable the site configuration using the `a2ensite` command and restart the Apache service to apply the changes.

Once everything is configured, the application will be accessible via the server’s domain or IP address. For example, if the server is set up with the domain your-domain.com, the application will be accessible at `http://your-domain.com`. Ensure all features of the application are functioning as expected after deployment to confirm a successful build.

II.V CREATING THE DATABASE

The application uses a MariaDB, a drop-in replacement for MySQL. Ensure the MariaDB version supports JSON data types (version 10.2 or higher). For earlier versions, adjust the schema to store JSON as a text field. It can also use SQLite, MySQL, or PostgreLite with modifications.

To create the database, you should source the file at https://turing.cs.olemiss.edu/~kreardo1/seniorProject/collabConnect/database/migrations/student Grouping_create (1).sql. Then download the file from the location and save it on the system that holds the MySQL database. If on the same system with MySQL, then go to the link, and it will download. If on a different system, go to the link, and it will download. Then copy the file to the system with MySQL database. After downloading and once on the system with the MySQL database, source the file by typing at the terminal, replacing the strings in [] with the correct information for your database:

```
mysql -u [username] -p [databasename] < studentGroup_create_(1).sql
```

The database schema is defined using Laravel migrations, which is executed using the php artisan migrate command. If manual setup is required, the following SQL statements can be used:

```
-- tables
-- Table: User CREATE TABLE User (
id int  NOT NULL,     name
varchar(225)  NOT NULL,     email
varchar(225)  NOT NULL,     password
varchar(225)  NOT NULL,     role
varchar(255)  NOT NULL,
created_at timestamp  NOT NULL,
updated_at timestamp  NOT NULL,
CONSTRAINT User_pk PRIMARY KEY (id)
);

-- Table: groupResults CREATE
TABLE groupResults (     id int
NOT NULL,     User_id int  NOT
NULL,     filename varchar(255)
NOT NULL,     groupings json  NOT
NULL,     groups_data json NOT
NULL,     created_at timestamp
NOT NULL,     updated_at
timestamp  NOT NULL,
    CONSTRAINT groupResults_pk PRIMARY KEY (id)
);

-- foreign keys
-- Reference: groupResults_User (table: groupResults)
```

```
ALTER TABLE groupResults ADD CONSTRAINT groupResults_User FOREIGN KEY
groupResults_User (User_id)
    REFERENCES User (id);
```

Below is the migration script for the users table.

```
<?php  use
Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
 return new class extends
Migration
{
    /**
     * Run the migrations.
     */     public function
up(): void
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->string('role')->default('teacher');
            $table->rememberToken();
            $table->timestamps();
        });

        Schema::create('password_reset_tokens', function (Blueprint
$table) {
            $table->string('email')->primary();
            $table->string('token');
            $table->timestamp('created_at')->nullable();
        });

        Schema::create('sessions', function (Blueprint $table) {
            $table->string('id')->primary();
            $table->foreignId('user_id')->nullable()->index();
            $table->string('ip_address', 45)->nullable();
            $table->text('user_agent')->nullable();
            $table->longText('payload');
            $table->integer('last_activity')->index();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down():
void
    {
        Schema::dropIfExists('users');
        Schema::dropIfExists('password_reset_tokens');
        Schema::dropIfExists('sessions');
    }
};
```

Below is the migration script for the group results table.

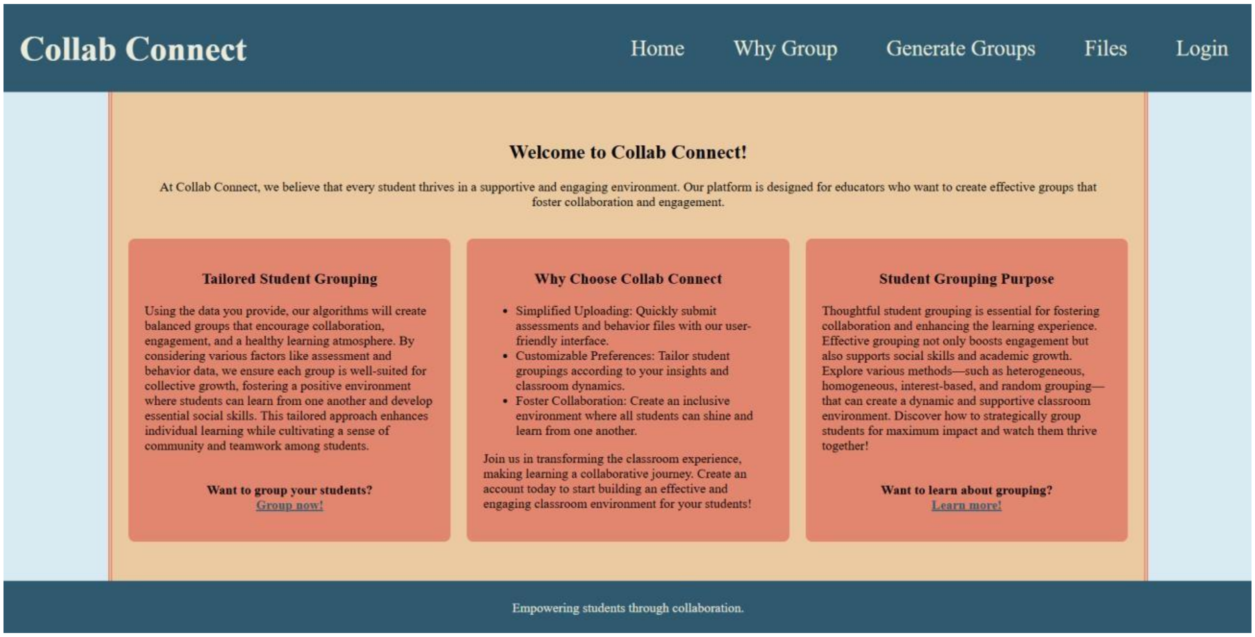
```
<?php  use
Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
 return new class extends
Migration
{
    /**
     * Run the migrations.
     */     public
function up()
    {
        Schema::create('group_results', function (Blueprint $table) {
            $table->id();
            $table->foreignId('user_id')->constrained()-
>onDelete('cascade');
            $table->string('filename');
            $table->json('groups');
            $table->json('groups_data');
            $table->timestamp('generated_at');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down():
void
    {
        Schema::dropIfExists('group_results');
    }
};
```

II.VI CURRENT HOSTING SITE

The application is hosted on the University of Mississippi’s Turing System. Teachers and administrators can access the live site at: https://turing.cs.olemiss.edu/~kreardo1/seniorProject/collabConnect/public/index.
Apache, php-mysql, and Composer are pre-installed on the University of Mississippi’s Turing System. Additional packages required for this project include php-cli, php-mbstring, php-xml, and php-bcmath. Apache is the web server, php-mysql is the package needed for communicating to the MariaDB, and composer is a php package manager.

III. USE: PUBLIC PAGES

III.I HOME PAGE OVERVIEW



Welcome to Collab Connect!

At Collab Connect, we believe that every student thrives in a supportive and engaging environment. Our platform is designed for educators who want to create effective groups that foster collaboration and engagement.

Tailored Student Grouping

Using the data you provide, our algorithms will create balanced groups that encourage collaboration, engagement, and a healthy learning atmosphere. By considering various factors like assessment and behavior data, we ensure each group is well-suited for collective growth, fostering a positive environment where students can learn from one another and develop essential social skills. This tailored approach enhances individual learning while cultivating a sense of community and teamwork among students.

Want to group your students?
Group now!

Why Choose Collab Connect

- Simplified Uploading: Quickly submit assessments and behavior files with our user-friendly interface.
- Customizable Preferences: Tailor student groupings according to your insights and classroom dynamics.
- Foster Collaboration: Create an inclusive environment where all students can shine and learn from one another.

Join us in transforming the classroom experience, making learning a collaborative journey. Create an account today to start building an effective and engaging classroom environment for your students!

Student Grouping Purpose

Thoughtful student grouping is essential for fostering collaboration and enhancing the learning experience. Effective grouping not only boosts engagement but also supports social skills and academic growth. Explore various methods—such as heterogeneous, homogeneous, interest-based, and random grouping—that can create a dynamic and supportive classroom environment. Discover how to strategically group students for maximum impact and watch them thrive together!

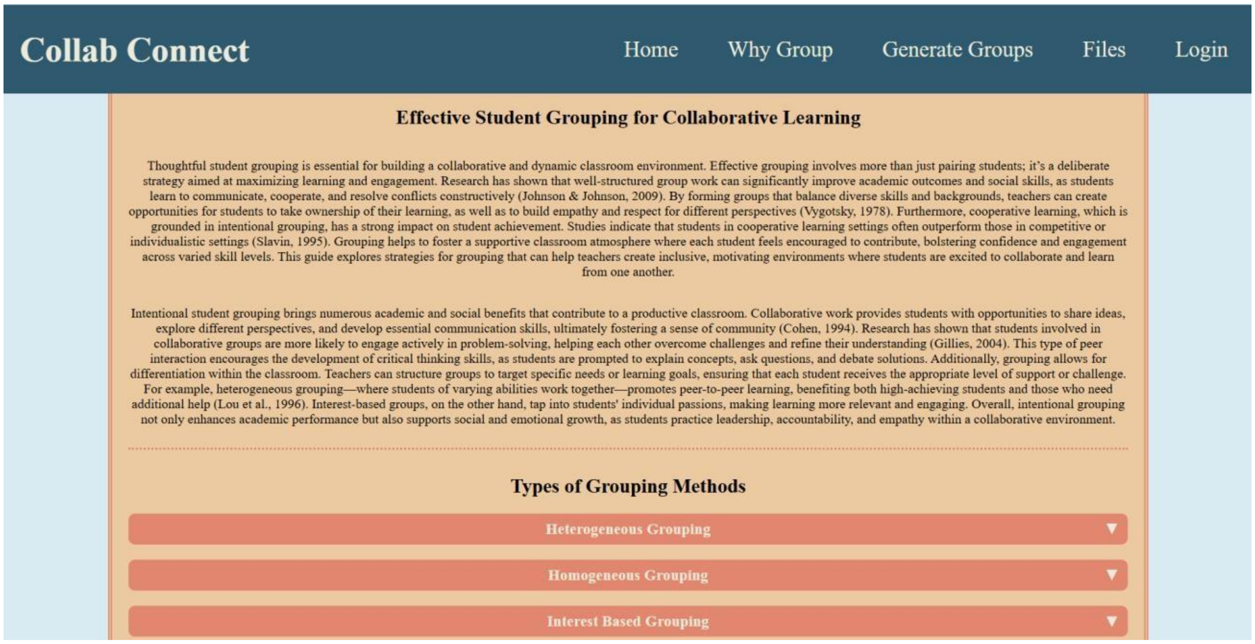
Want to learn about grouping?
Learn more!

Empowering students through collaboration.

Figure A: The home page is the initial page a user will see when coming to the website. It provides a brief overview of the purpose of the website.

The Home page introduces the purpose of the Collab Connect website and is accessible without logging in. It serves to allow new visitors to learn about the website's purpose. It explains the website's mission to create balanced groupings based on assessment data, behavior data, and teacher input to optimize classroom performance and promote collaboration in the classroom. This page features an intuitive navigation bar at the top of each page, allowing users to access other pages of the website such as Why Group, Generate Groups, Files, and Admin.

III.II WHY GROUP PAGE OVERVIEW



Effective Student Grouping for Collaborative Learning

Thoughtful student grouping is essential for building a collaborative and dynamic classroom environment. Effective grouping involves more than just pairing students; it's a deliberate strategy aimed at maximizing learning and engagement. Research has shown that well-structured group work can significantly improve academic outcomes and social skills, as students learn to communicate, cooperate, and resolve conflicts constructively (Johnson & Johnson, 2009). By forming groups that balance diverse skills and backgrounds, teachers can create opportunities for students to take ownership of their learning, as well as to build empathy and respect for different perspectives (Vygotsky, 1978). Furthermore, cooperative learning, which is grounded in intentional grouping, has a strong impact on student achievement. Studies indicate that students in cooperative learning settings often outperform those in competitive or individualistic settings (Slavin, 1995). Grouping helps to foster a supportive classroom atmosphere where each student feels encouraged to contribute, bolstering confidence and engagement across varied skill levels. This guide explores strategies for grouping that can help teachers create inclusive, motivating environments where students are excited to collaborate and learn from one another.

Intentional student grouping brings numerous academic and social benefits that contribute to a productive classroom. Collaborative work provides students with opportunities to share ideas, explore different perspectives, and develop essential communication skills, ultimately fostering a sense of community (Cohen, 1994). Research has shown that students involved in collaborative groups are more likely to engage actively in problem-solving, helping each other overcome challenges and refine their understanding (Gillies, 2004). This type of peer interaction encourages the development of critical thinking skills, as students are prompted to explain concepts, ask questions, and debate solutions. Additionally, grouping allows for differentiation within the classroom. Teachers can structure groups to target specific needs or learning goals, ensuring that each student receives the appropriate level of support or challenge. For example, heterogeneous grouping—where students of varying abilities work together—promotes peer-to-peer learning, benefiting both high-achieving students and those who need additional help (Lou et al., 1996). Interest-based groups, on the other hand, tap into students' individual passions, making learning more relevant and engaging. Overall, intentional grouping not only enhances academic performance but also supports social and emotional growth, as students practice leadership, accountability, and empathy within a collaborative environment.

Types of Grouping Methods

Figure B: The Why Group page provides information on the benefits of groupings students and the various ways to group students.

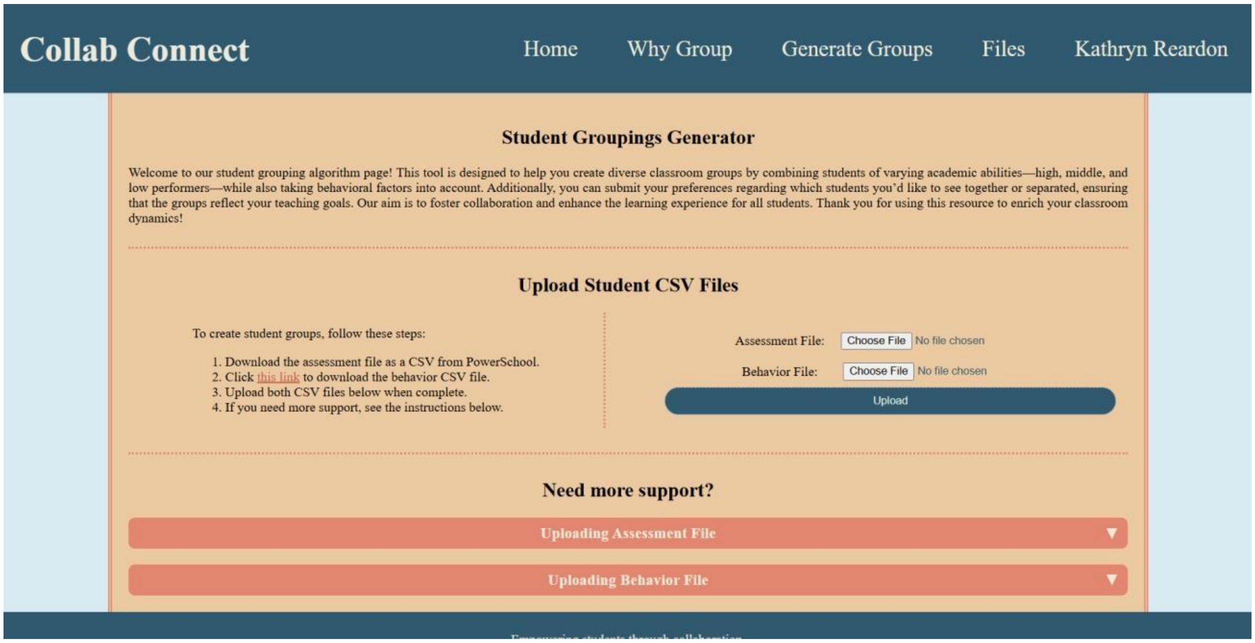
The Why Group page provides a comprehensive overview of the educational benefits of intentional student grouping and is accessible without being logged in. The page uses educational research to emphasize how meaningful groupings can improve academic performance, foster collaboration, and build essential social skills within the classroom. Users are introduced to various grouping strategies, including heterogeneous, homogeneous, interest-based, and random groupings. This serves as a tool to help teachers make informed decisions about grouping strategies that best fit their classroom dynamics. This page also provides resources that teachers can use to integrate group-based learning in the classroom. This page is a valuable resource for educators wanting to deepen their understanding of effective grouping methods.

IV. USE: TEACHER PAGES

IV.I GENERATE GROUPS PAGES

The Generate Groups section is the heart of the Collab Connect platform, guiding teachers through a three-step process to create student groupings. This feature includes three interconnected pages—Upload Files, Preferences, and Results—which streamline the grouping process and allow for customization. Each page is designed to be user-friendly so that teachers of all skill levels can use the platform effectively. To access these pages, the user must be logged in to their account.

IV.I.I Upload Files Page Overview



Student Groupings Generator

Welcome to our student grouping algorithm page! This tool is designed to help you create diverse classroom groups by combining students of varying academic abilities—high, middle, and low performers—while also taking behavioral factors into account. Additionally, you can submit your preferences regarding which students you'd like to see together or separated, ensuring that the groups reflect your teaching goals. Our aim is to foster collaboration and enhance the learning experience for all students. Thank you for using this resource to enrich your classroom dynamics!

Upload Student CSV Files

To create student groups, follow these steps:

1. Download the assessment file as a CSV from PowerSchool.
2. Click this link to download the behavior CSV file.
3. Upload both CSV files below when complete.
4. If you need more support, see the instructions below.

Assessment File: Choose File No file chosen

Behavior File: Choose File No file chosen

Need more support?

Uploading Assessment File ▼
Uploading Behavior File ▼
Empowering students through collaboration

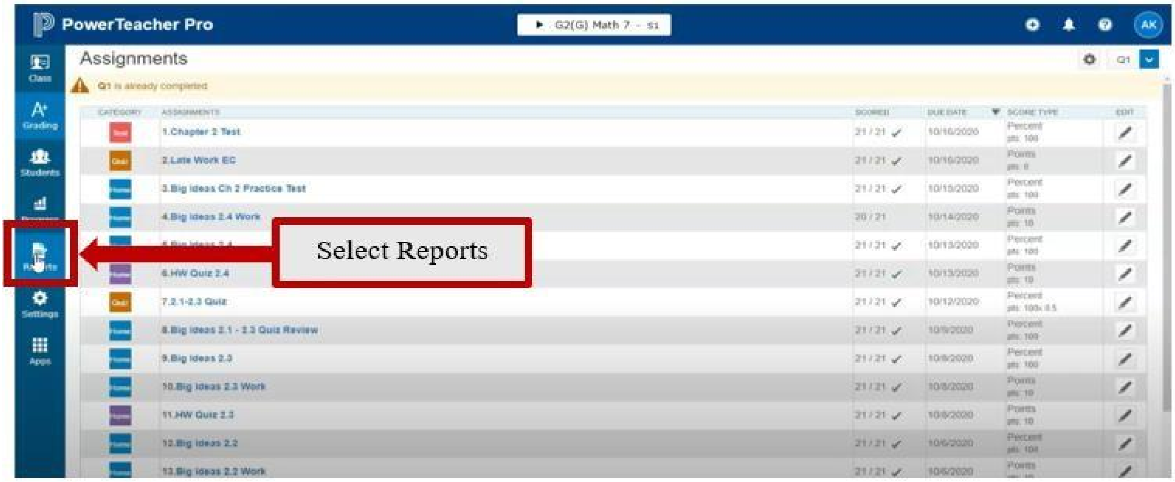
Figure C: The Upload Files page is the first step in generating student groupings. This is where the teacher will upload the assessment and behavior CSV files, and it provides instructions on how to upload the assessment and behavior files.

The Upload Files page is the first step in generating student groupings. Teachers begin by preparing the necessary data files. The assessment file is downloaded as a CSV from PowerSchool, while the behavior file is generated by clicking a link that forces a copy of a preformatted Google Spreadsheet. Teachers then paste student names into the spreadsheet, fill in the behavior data, and download the completed file as a CSV. Once both files are ready, they are uploaded using the "Choose File" buttons on the page. Clicking the "Upload" button submits the files to the system, initiating the grouping process. This page's clear instructions and simple interface ensure that teachers can complete the upload process with ease. An error message will display when the user clicks upload if the files do not have students, do not have the same number of students, or are not in the correct format so that the teacher can check the files and correct the errors before proceeding.

IV.I.II Assessment File Uploading

To upload an assessment file, first download the file as an Excel sheet from PowerSchool. Log into PowerSchool, and go to PowerTeacher Pro. Follow the steps shown in the images below to download the report or use this video by Marie Henderson for a more guided explanation: https://www.youtube.com/watch?v=WSGyc4LvlBU. If following the video, be sure to select the specific class in classes under the Criteria tab and the output as Excel under the Format tab. These are the only necessary selections to make in the criteria, student, and format tabs. Once the scoresheet is downloaded, follow along with the picture guide to adjust the file before downloading it as a CSV file and uploading it to the website.

Step 1: Select Reports on the far-left side of the PowerSchool page.

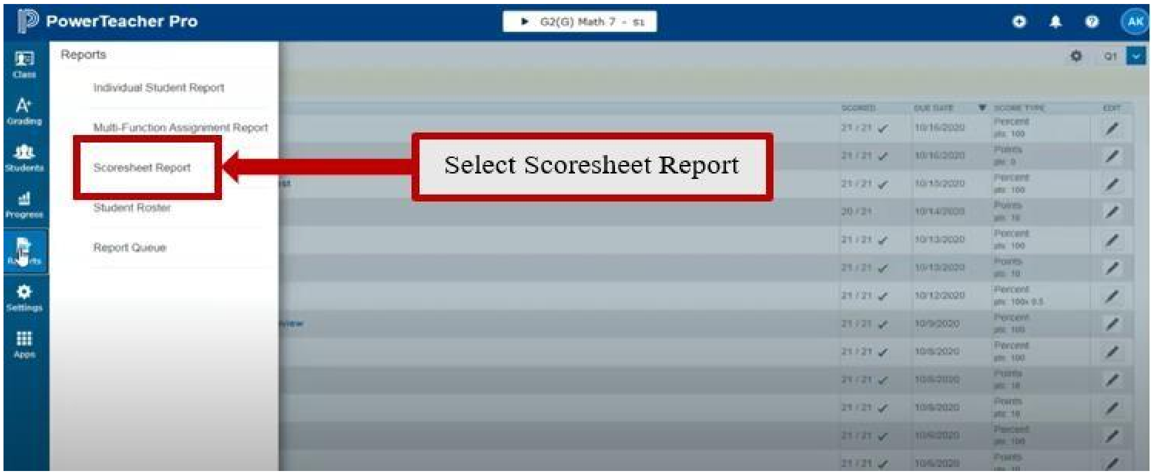


Assignments

Q1 is already completed

CATEGORY	ASSIGNMENTS	SCORED	DUE DATE	SCORE TYPE	EDIT
Test	1.Chapter 2 Test	21 / 21 ✓	10/16/2020	Percent pts: 100	✎
Quiz	2.Late Work EC	21 / 21 ✓	10/16/2020	Points pts: 0	✎
Home	3.Big Ideas Ch 2 Practice Test	21 / 21 ✓	10/15/2020	Percent pts: 100	✎
Home	4.Big Ideas 2.4 Work	20 / 21	10/14/2020	Points pts: 10	✎
	5.Big Ideas 2.4	21 / 21 ✓	10/13/2020	Percent pts: 100	✎
Home	6.HW Quiz 2.4	21 / 21 ✓	10/13/2020	Points pts: 10	✎
Quiz	7.2.1-2.3 Quiz	21 / 21 ✓	10/12/2020	Percent pts: 100x 0.5	✎
Home	8.Big Ideas 2.1 - 2.3 Quiz Review	21 / 21 ✓	10/9/2020	Percent pts: 100	✎
Home	9.Big Ideas 2.3	21 / 21 ✓	10/8/2020	Percent pts: 100	✎
Home	10.Big Ideas 2.3 Work	21 / 21 ✓	10/8/2020	Points pts: 10	✎
Home	11.HW Quiz 2.3	21 / 21 ✓	10/8/2020	Points pts: 10	✎
Home	12.Big Ideas 2.2	21 / 21 ✓	10/6/2020	Percent pts: 100	✎
Home	13.Big Ideas 2.2 Work	21 / 21 ✓	10/6/2020	Points	✎

Step 2: Select the Scoresheet Report in the Reports options list.

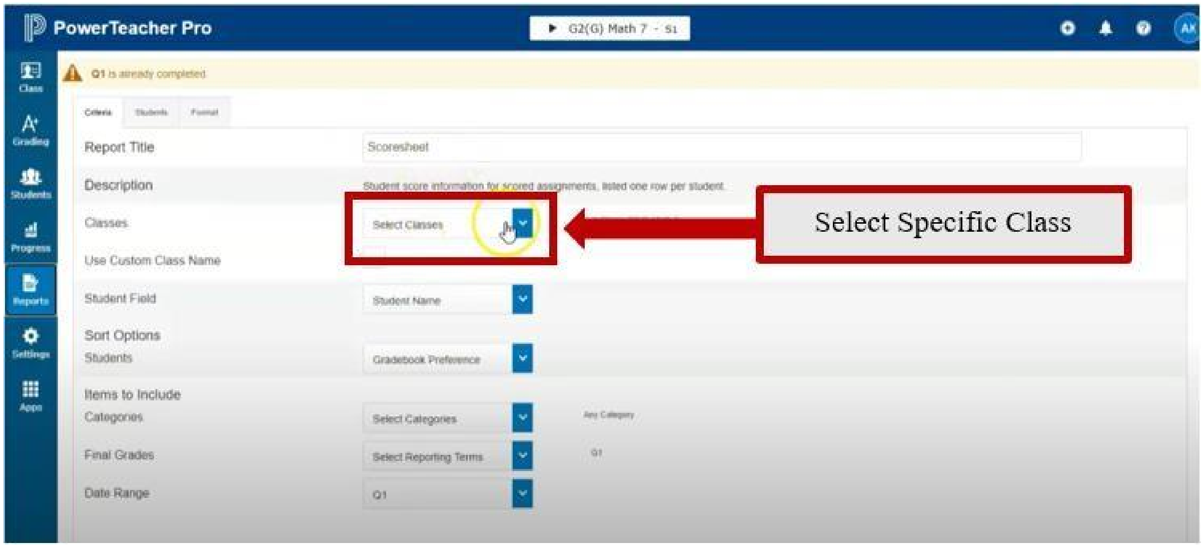


Reports

- Individual Student Report
- Multi-Function Assignment Report
- Scoresheet Report
- Student Roster
- Report Queue

Select Scoresheet Report

Step 3: Select the specific class to be grouped in classes under the Criteria tab.



Q1 is already completed.

Report Title: Scoresheet

Description: Student score information for scored assignments, listed one row per student.

Classes: Select Classes

Use Custom Class Name

Student Field: Student Name

Sort Options

Students: Gradebook Preference

Items to Include

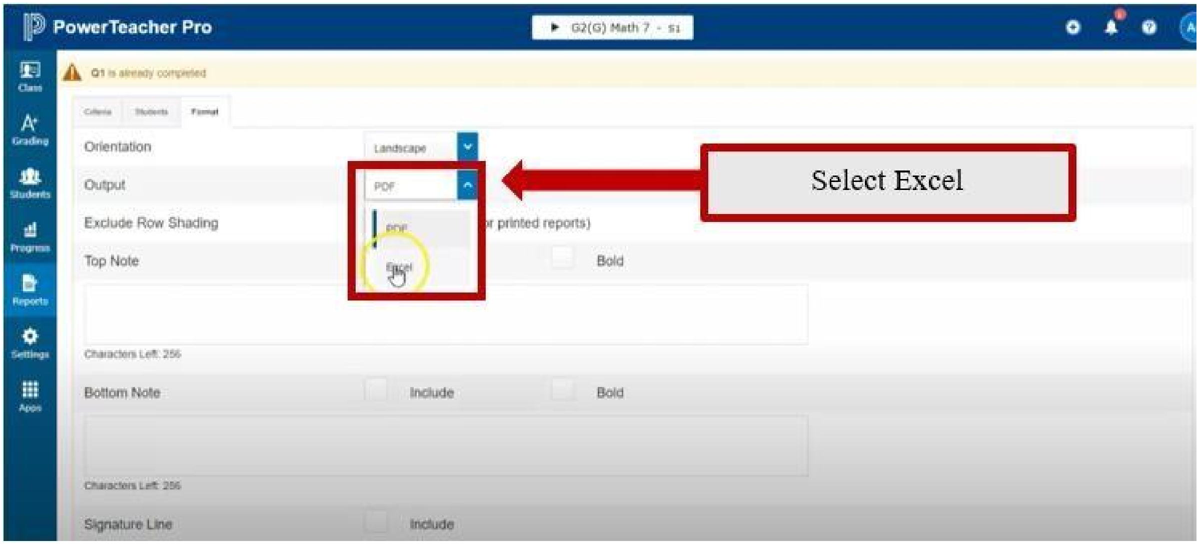
Categories: Select Categories — Any Category

Final Grades: Select Reporting Terms — Q1

Date Range: Q1

Select Specific Class

Step 4: Select Excel as the output under the Format tab.



Q1 is already completed

Orientation: Landscape

Output: PDF

Exclude Row Shading (Saves ink for printed reports)

Top Note □ Include □ Bold

Characters Left: 256

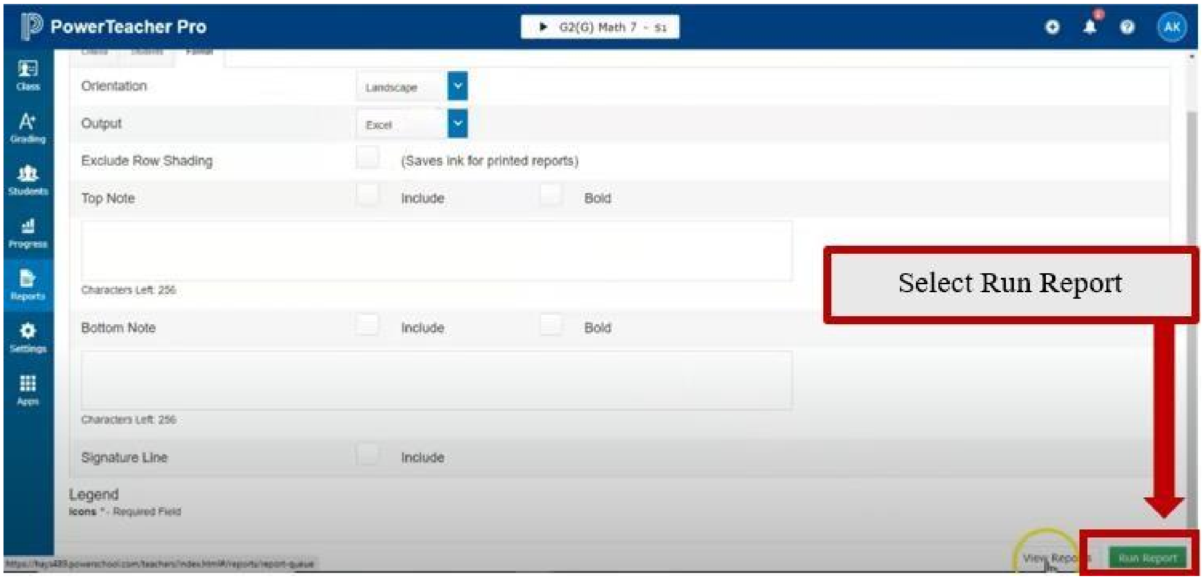
Bottom Note □ Include □ Bold

Characters Left: 256

Signature Line □ Include

Select Excel

Step 5: Click Run Report to generate the student assessment report.



Orientation: Landscape

Output: Excel

Exclude Row Shading □ (Saves ink for printed reports)

Top Note □ Include □ Bold

Characters Left: 256

Bottom Note □ Include □ Bold

Characters Left: 256

Signature Line □ Include

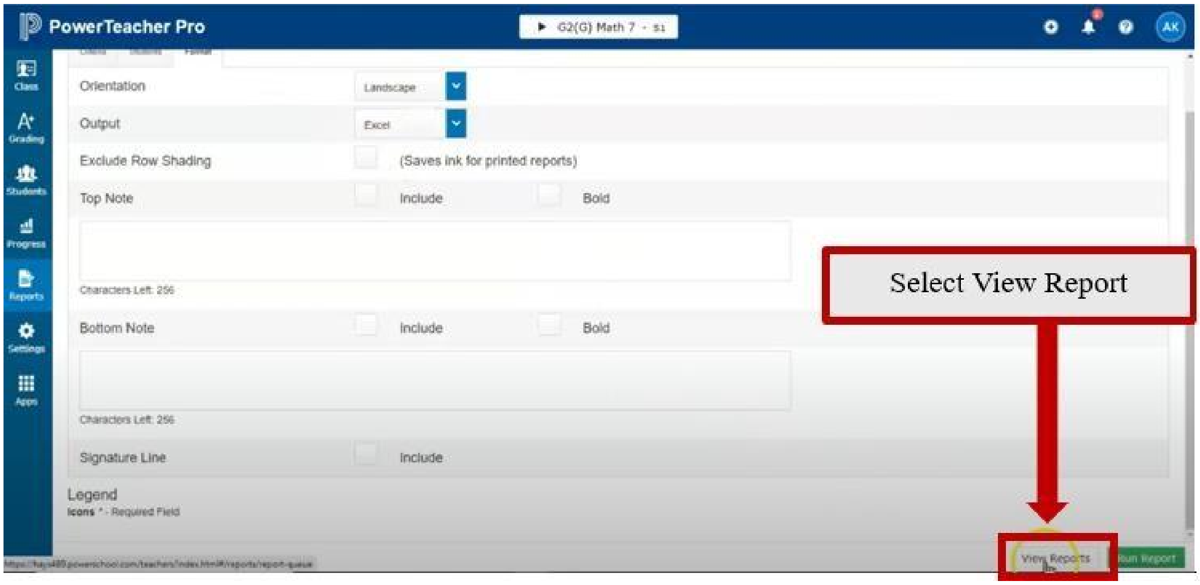
Legend

Icons * - Required Field

https://hays489.powerschool.com/teachers/index.html#/reports/report-queue

Select Run Report

Step 6: Select View Report to view the report that was generated and prior reports.



Orientation: Landscape

Output: Excel

Exclude Row Shading □ (Saves ink for printed reports)

Top Note □ Include □ Bold

Characters Left: 256

Bottom Note □ Include □ Bold

Characters Left: 256

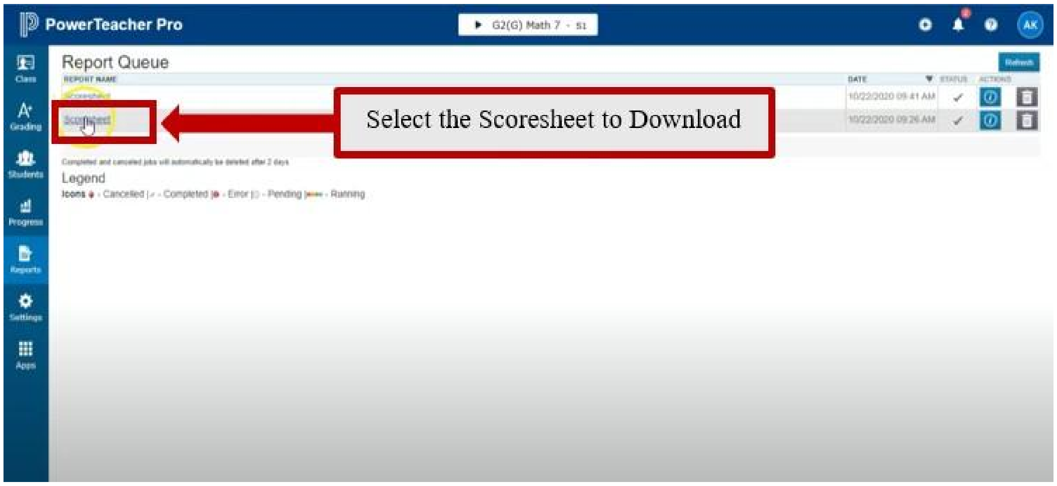
Signature Line □ Include

Legend

Icons * - Required Field

Select View Report

Step 7: Select Scoresheet for the scoresheet that was generated. Check the date on the scoresheet to make sure it is the scoresheet that was just generated.



Report Queue

REPORT NAME	DATE	STATUS	ACTIONS
Scoresheet	10/22/2020 09:41 AM	✓	
Scoresheet	10/22/2020 09:26 AM	✓	

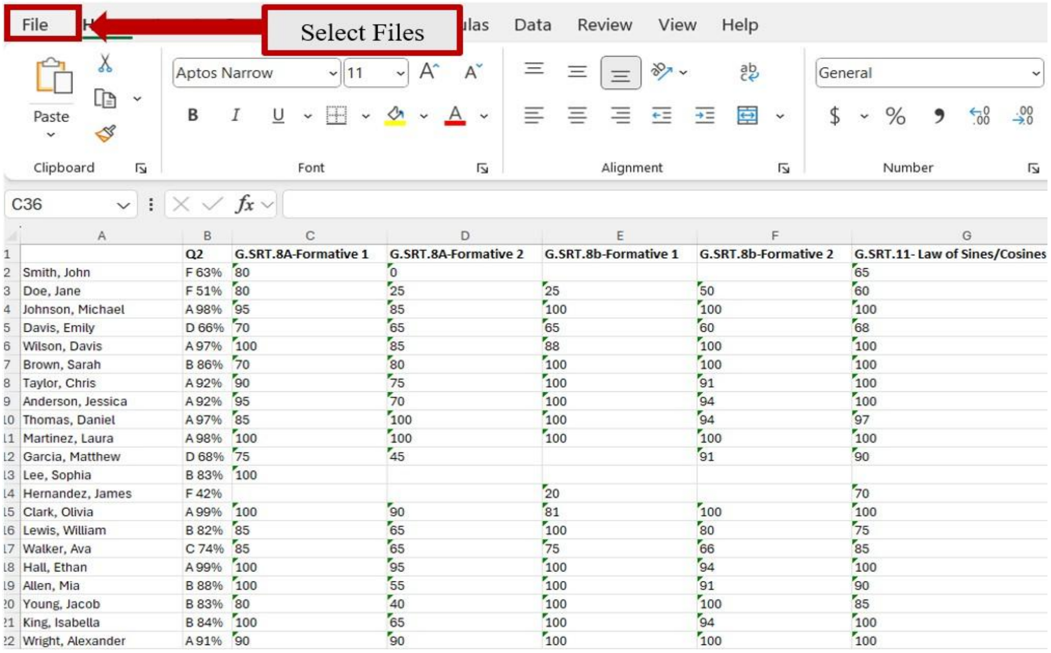
Completed and canceled jobs will automatically be deleted after 2 days

Legend

Icons - Canceled - Completed - Error - Pending - Running

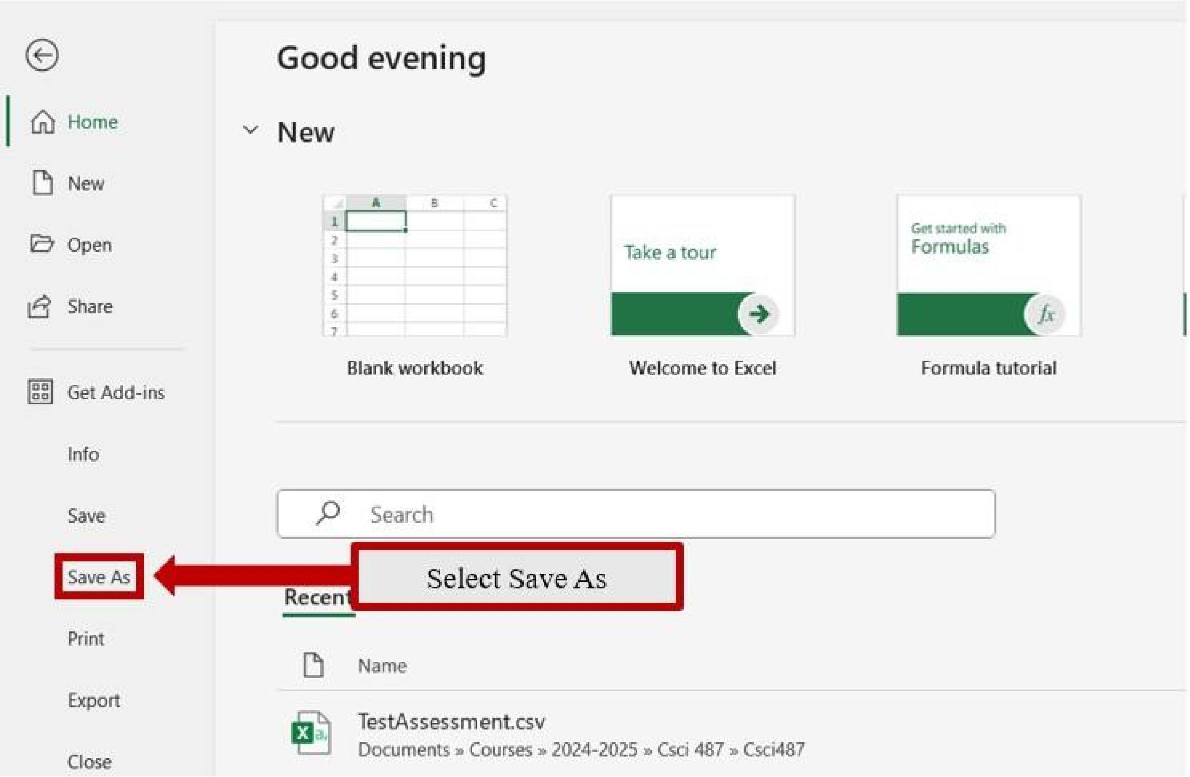
Select the Scoresheet to Download

Step 8: Open the scoresheet file that was just downloaded. Any assessment score columns that do not need to be included can be deleted by right clicking the column, clicking delete, and clicking the entire column. Leave the quarter percentage column. Once done, select File in the top right corner.



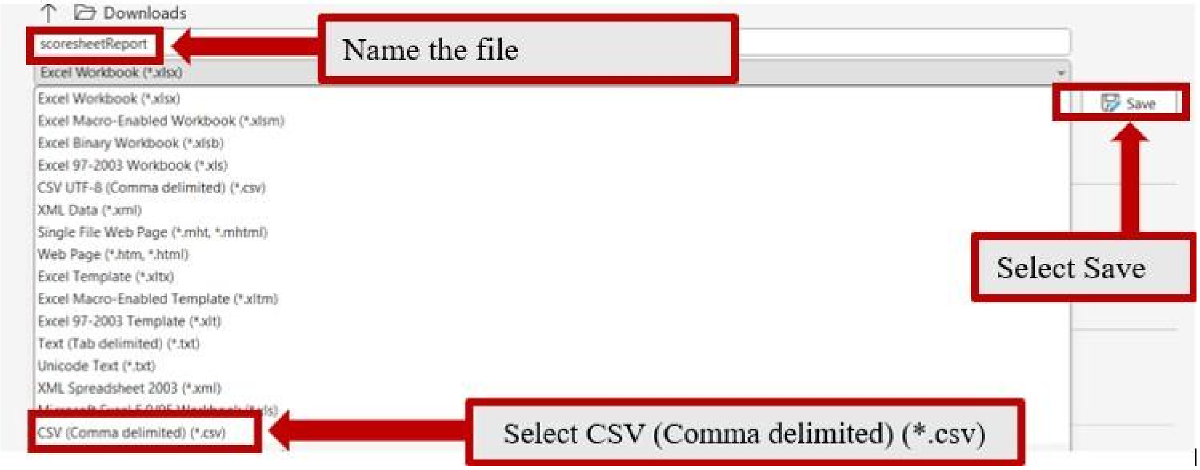
	A	B	C	D	E	F	G
1		Q2	G.SRT.8A-Formative 1	G.SRT.8A-Formative 2	G.SRT.8b-Formative 1	G.SRT.8b-Formative 2	G.SRT.11- Law of Sines/Cosines
2	Smith, John	F 63%	80	0			65
3	Doe, Jane	F 51%	80	25	25	50	60
4	Johnson, Michael	A 98%	95	85	100	100	100
5	Davis, Emily	D 66%	70	65	65	60	68
6	Wilson, Davis	A 97%	100	85	88	100	100
7	Brown, Sarah	B 86%	70	80	100	100	100
8	Taylor, Chris	A 92%	90	75	100	91	100
9	Anderson, Jessica	A 92%	95	70	100	94	100
10	Thomas, Daniel	A 97%	85	100	100	94	97
11	Martinez, Laura	A 98%	100	100	100	100	100
12	Garcia, Matthew	D 68%	75	45		91	90
13	Lee, Sophia	B 83%	100				
14	Hernandez, James	F 42%			20		70
15	Clark, Olivia	A 99%	100	90	81	100	100
16	Lewis, William	B 82%	85	65	100	80	75
17	Walker, Ava	C 74%	85	65	75	66	85
18	Hall, Ethan	A 99%	100	95	100	94	100
19	Allen, Mia	B 88%	100	55	100	91	90
20	Young, Jacob	B 83%	80	40	100	100	85
21	King, Isabella	B 84%	100	65	100	94	100
22	Wright, Alexander	A 91%	90	90	100	100	100

Step 9: Select Save As in the panel to the right.



TestAssessment.csv
Documents » Courses » 2024-2025 » Csci 487 » Csci487

Step 10: Name the file to be identifiable for uploading or leave it as scoresheetReport. Then in the dropdown below the filename, select CSV (Comma delimited) (*.csv). Then click save.



scoresheetReport

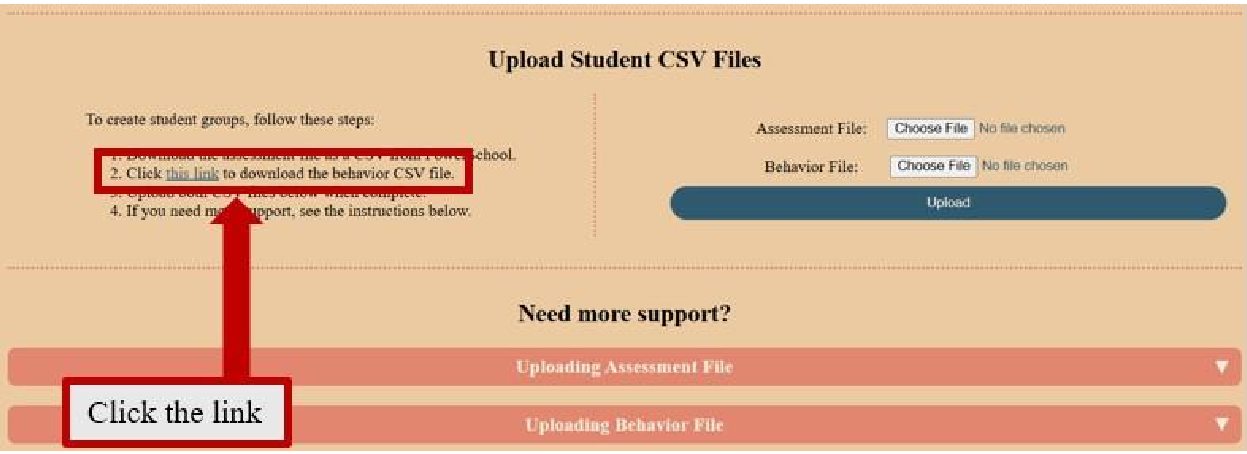
Excel Workbook (*.xlsx)

Excel Workbook (*.xlsx)
Excel Macro-Enabled Workbook (*.xlsm)
Excel Binary Workbook (*.xlsb)
Excel 97-2003 Workbook (*.xls)
CSV UTF-8 (Comma delimited) (*.csv)
XML Data (*.xml)
Single File Web Page (*.mht, *.mhtml)
Web Page (*.htm, *.html)
Excel Template (*.xltx)
Excel Macro-Enabled Template (*.xltm)
Excel 97-2003 Template (*.xlt)
Text (Tab delimited) (*.txt)
Unicode Text (*.txt)
XML Spreadsheet 2003 (*.xml)
Microsoft Excel 5.0/95 Workbook (*.xls)
CSV (Comma delimited) (*.csv)

IV.I.III Behavior File Uploading

To upload the behavior file, first, click the behavior file link on the upload student CSV file page, fill out the file based on student behavior, and then download it as a CSV. Following the following steps for step-by-step instructions to do this:

Step 1: Click the link in step 2 of the website instructions in the Upload Student CSV Files section of Generate Groups.



Upload Student CSV Files

To create student groups, follow these steps:

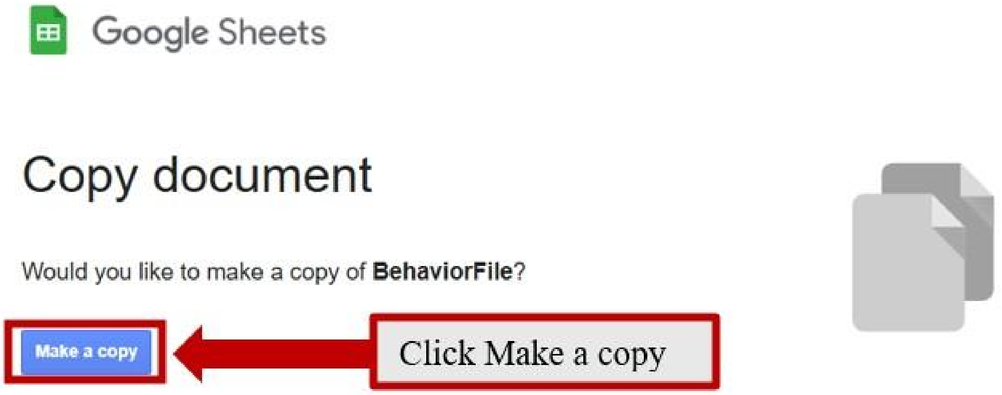
1. Download the assessment file as a CSV from PowerSchool.
2. Click this link to download the behavior CSV file.
3. Upload both CSV files below when complete.
4. If you need more support, see the instructions below.

Assessment File:

Behavior File:

Need more support?

Step 2: Click Make a copy to make a force copy of the behavior file.

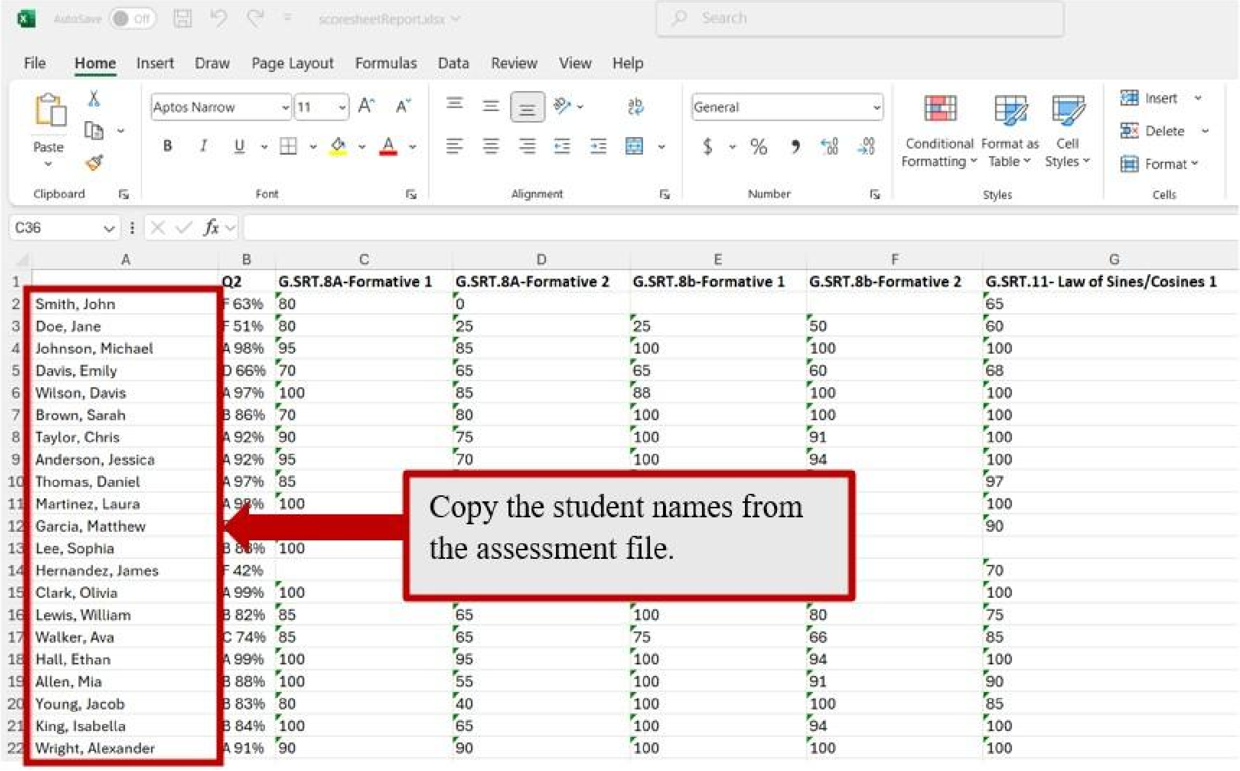


Copy document

Would you like to make a copy of **BehaviorFile**?

Click Make a copy

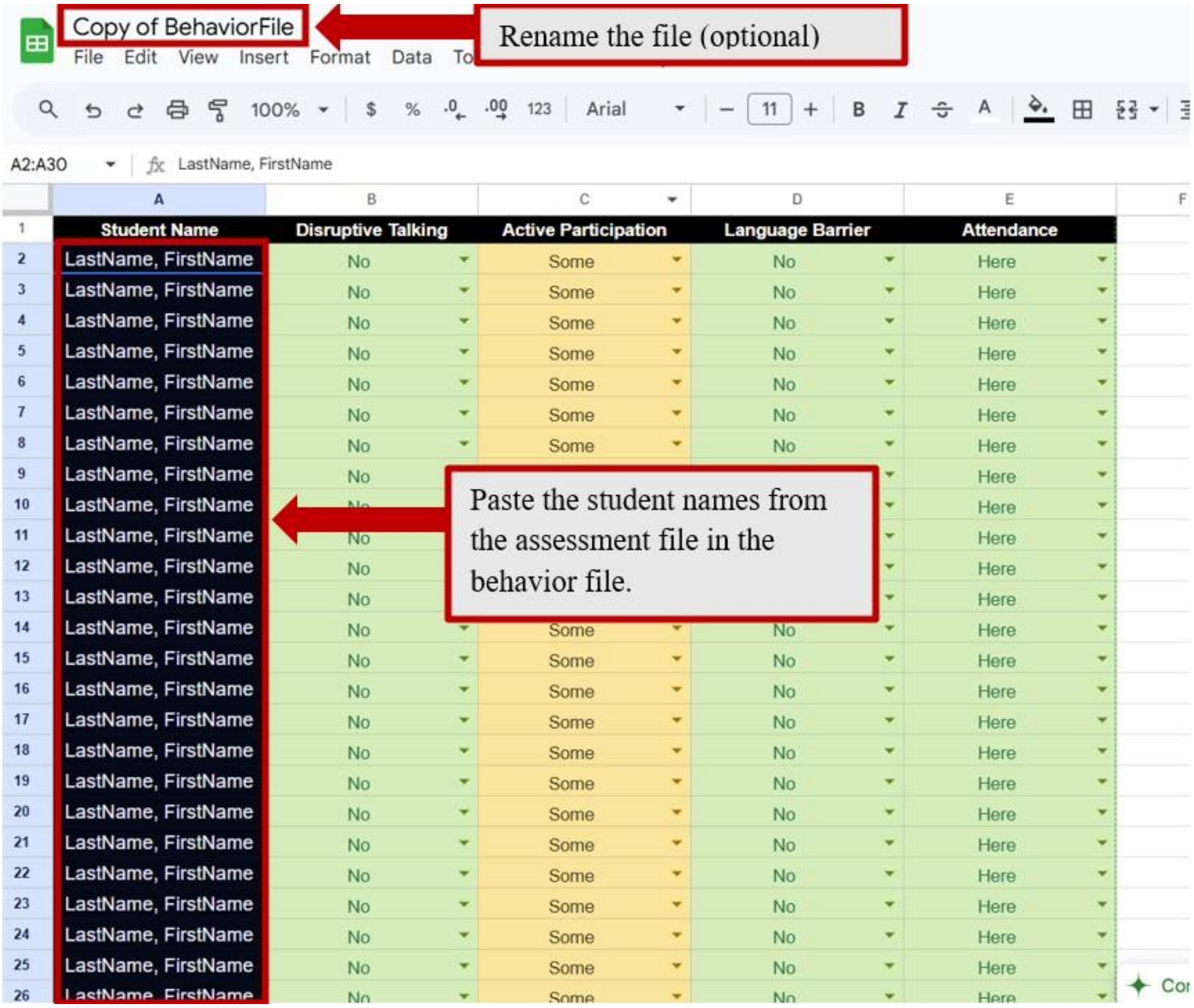
Step 3: Highlight and copy the student names from the assessment scoresheet file. These names are already formatted correctly.



	A	B	C	D	E	F	G
1		Q2	G.SRT.8A-Formative 1	G.SRT.8A-Formative 2	G.SRT.8b-Formative 1	G.SRT.8b-Formative 2	G.SRT.11- Law of Sines/Cosines 1
2	Smith, John	F 63%	80	0			65
3	Doe, Jane	F 51%	80	25	25	50	60
4	Johnson, Michael	A 98%	95	85	100	100	100
5	Davis, Emily	D 66%	70	65	65	60	68
6	Wilson, Davis	A 97%	100	85	88	100	100
7	Brown, Sarah	B 86%	70	80	100	100	100
8	Taylor, Chris	A 92%	90	75	100	91	100
9	Anderson, Jessica	A 92%	95	70	100	94	100
10	Thomas, Daniel	A 97%	85				97
11	Martinez, Laura	A 9[illegible]%	100				100
12	Garcia, Matthew						90
13	Lee, Sophia	B 8[illegible]%	100				
14	Hernandez, James	F 42%					70
15	Clark, Olivia	A 99%	100				100
16	Lewis, William	B 82%	85	65	100	80	75
17	Walker, Ava	C 74%	85	65	75	66	85
18	Hall, Ethan	A 99%	100	95	100	94	100
19	Allen, Mia	B 88%	100	55	100	91	90
20	Young, Jacob	B 83%	80	40	100	100	85
21	King, Isabella	B 84%	100	65	100	94	100
22	Wright, Alexander	A 91%	90	90	100	100	100

Copy the student names from the assessment file.

Step 4: Highlight the template names in the behavior file copy, and paste the names from the assessment file in the column. Be sure to highlight starting on row 2. Rename the file to the desired filename (optional).

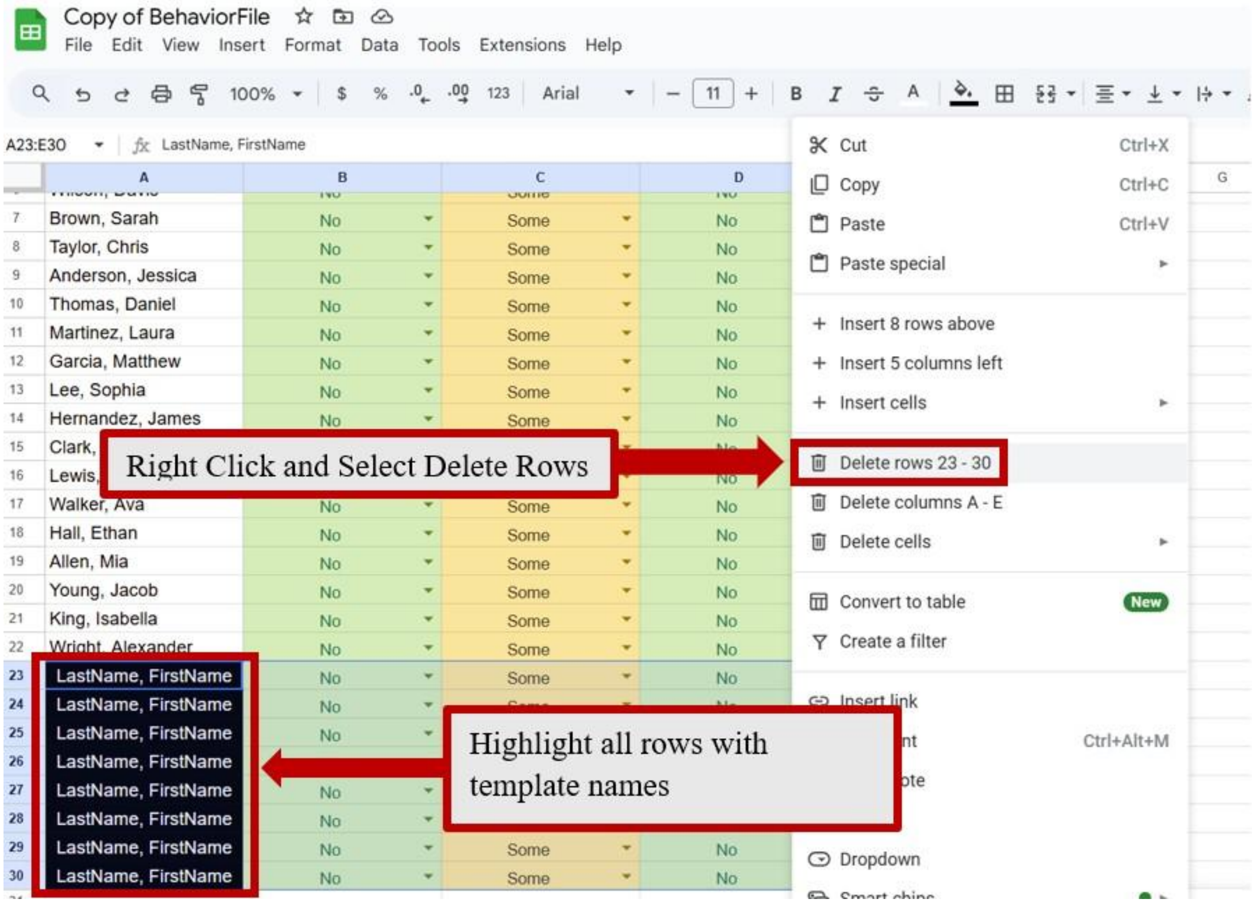


Rename the file (optional)

Paste the student names from the assessment file in the behavior file.

	Student Name	Disruptive Talking	Active Participation	Language Barrier	Attendance
2	LastName, FirstName	No	Some	No	Here
3	LastName, FirstName	No	Some	No	Here
4	LastName, FirstName	No	Some	No	Here
5	LastName, FirstName	No	Some	No	Here
6	LastName, FirstName	No	Some	No	Here
7	LastName, FirstName	No	Some	No	Here
8	LastName, FirstName	No	Some	No	Here
9	LastName, FirstName	No			Here
10	LastName, FirstName	No			Here
11	LastName, FirstName	No			Here
12	LastName, FirstName	No			Here
13	LastName, FirstName	No			Here
14	LastName, FirstName	No	Some	No	Here
15	LastName, FirstName	No	Some	No	Here
16	LastName, FirstName	No	Some	No	Here
17	LastName, FirstName	No	Some	No	Here
18	LastName, FirstName	No	Some	No	Here
19	LastName, FirstName	No	Some	No	Here
20	LastName, FirstName	No	Some	No	Here
21	LastName, FirstName	No	Some	No	Here
22	LastName, FirstName	No	Some	No	Here
23	LastName, FirstName	No	Some	No	Here
24	LastName, FirstName	No	Some	No	Here
25	LastName, FirstName	No	Some	No	Here
26	LastName, FirstName	No	Some	No	Here

Step 5: Select the rows with template names if there are leftover template names. Right click and select delete rows.

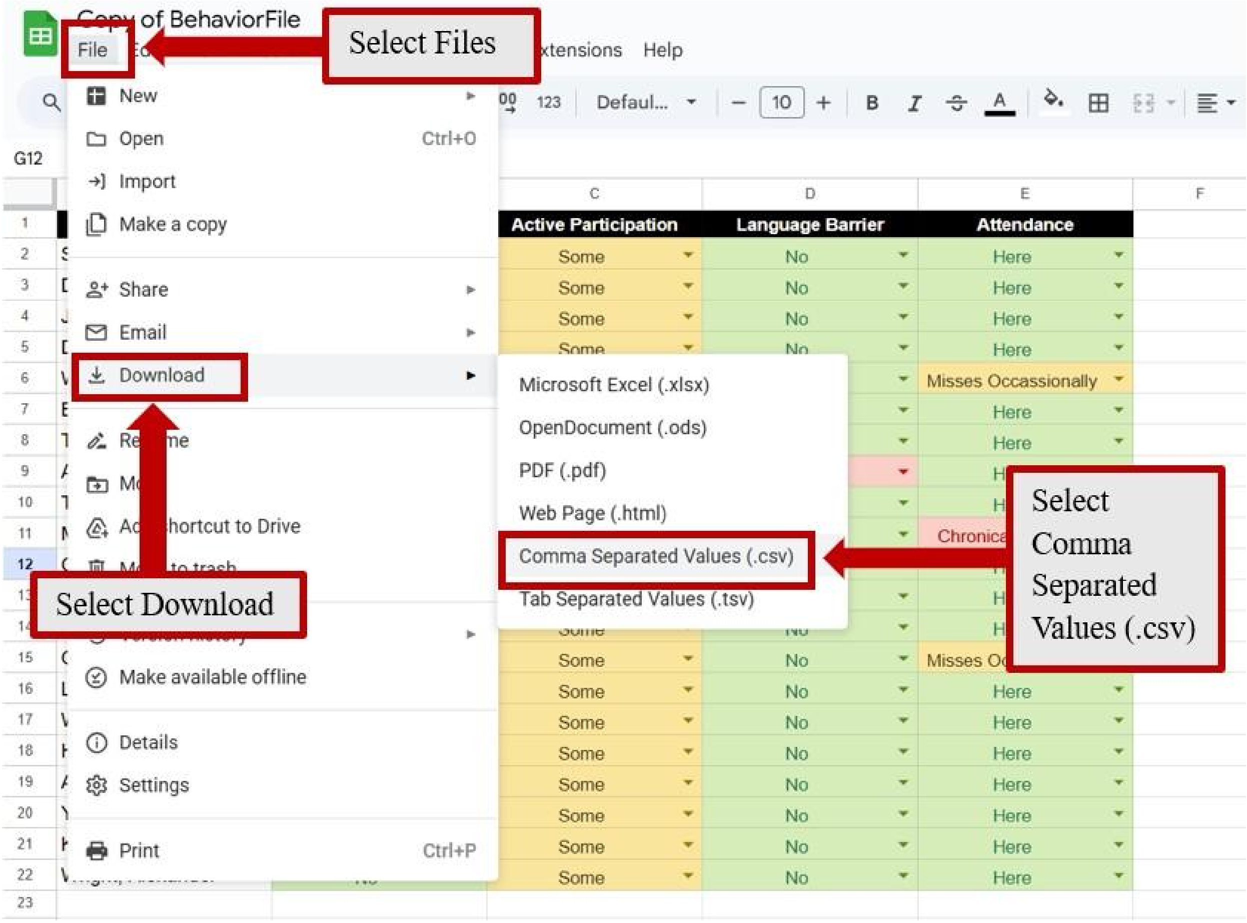


	A	B	C	D
7	Brown, Sarah	No	Some	No
8	Taylor, Chris	No	Some	No
9	Anderson, Jessica	No	Some	No
10	Thomas, Daniel	No	Some	No
11	Martinez, Laura	No	Some	No
12	Garcia, Matthew	No	Some	No
13	Lee, Sophia	No	Some	No
14	Hernandez, James	No	Some	No
15	Clark,			No
16	Lewis,			No
17	Walker, Ava	No	Some	No
18	Hall, Ethan	No	Some	No
19	Allen, Mia	No	Some	No
20	Young, Jacob	No	Some	No
21	King, Isabella	No	Some	No
22	Wright, Alexander	No	Some	No
23	LastName, FirstName	No	Some	No
24	LastName, FirstName	No		
25	LastName, FirstName	No		
26	LastName, FirstName			
27	LastName, FirstName	No		
28	LastName, FirstName	No		
29	LastName, FirstName	No	Some	No
30	LastName, FirstName	No	Some	No

Right Click and Select Delete Rows

Highlight all rows with template names

Step 6: Fill out file using the drop down menus based on student behavior. Then select files, click download, and then select comma separated values (.csv). This will download the file as a CSV file.



Select Files

Select Download

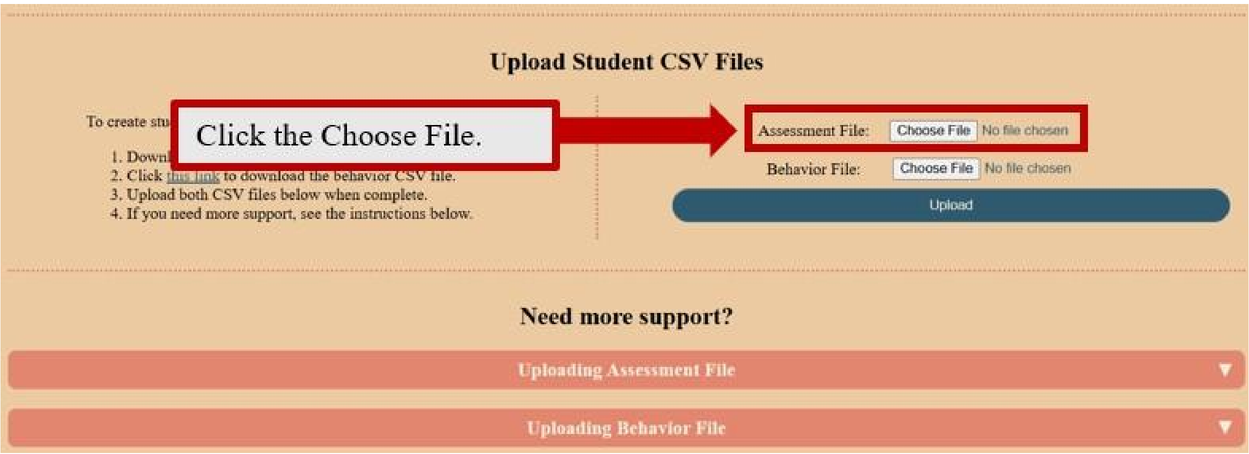
Select Comma Separated Values (.csv)

	C	D	E
1	Active Participation	Language Barrier	Attendance
2	Some	No	Here
3	Some	No	Here
4	Some	No	Here
5	Some	No	Here
6			Misses Occassionally
7			Here
8			Here
9			H
10			H
11			Chronica
12			
13			H
14			H
15	Some	No	Misses Oc
16	Some	No	Here
17	Some	No	Here
18	Some	No	Here
19	Some	No	Here
20	Some	No	Here
21	Some	No	Here
22	Some	No	Here

IV.I.IV Uploading Files Generate Groups

To upload the assessment and behavior files to the generate groups, the teacher must follow the following steps:

Step 1: Click the assessment file Choose File button.



Upload Student CSV Files

To create stu

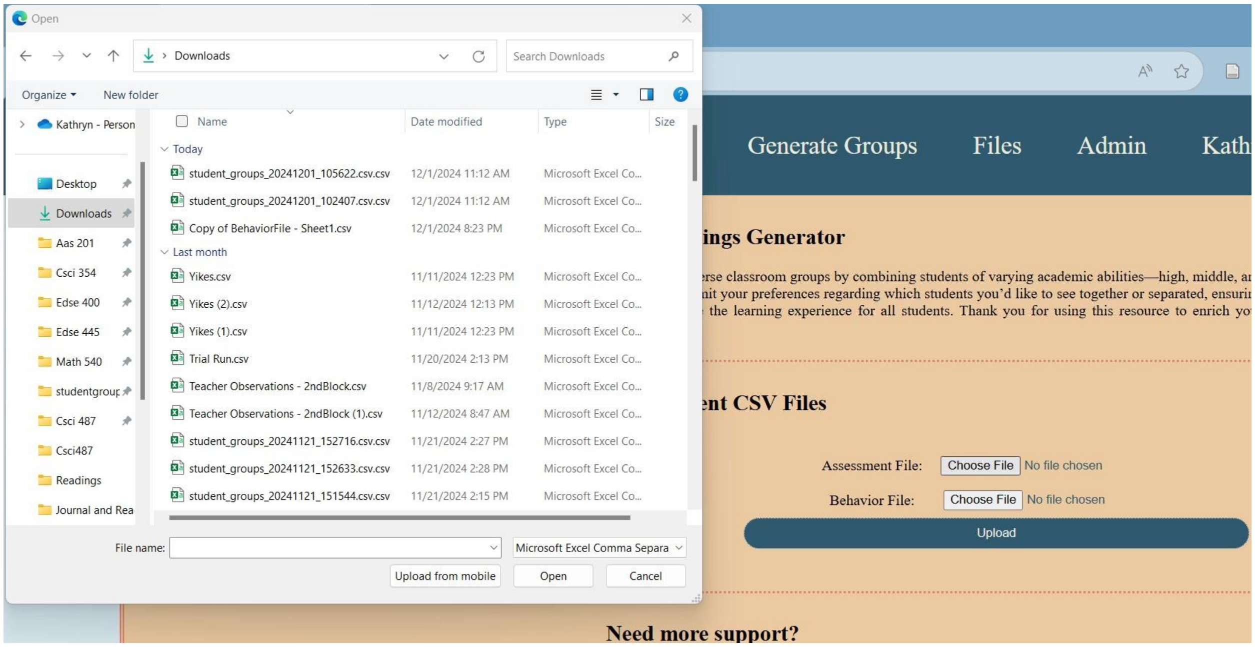
1. Downl
2. Click this link to download the behavior CSV file.
3. Upload both CSV files below when complete.
4. If you need more support, see the instructions below.

Click the Choose File.

Assessment File: Choose File No file chosen

Behavior File: Choose File No file chosen

Step 2: Select the Assessment File with the student assessment data to group based on.



Name	Date modified	Type	Size
Today			
student_groups_20241201_105622.csv.csv	12/1/2024 11:12 AM	Microsoft Excel Co...	
student_groups_20241201_102407.csv.csv	12/1/2024 11:12 AM	Microsoft Excel Co...	
Copy of BehaviorFile - Sheet1.csv	12/1/2024 8:23 PM	Microsoft Excel Co...	
Last month			
Yikes.csv	11/11/2024 12:23 PM	Microsoft Excel Co...	
Yikes (2).csv	11/12/2024 12:13 PM	Microsoft Excel Co...	
Yikes (1).csv	11/11/2024 12:23 PM	Microsoft Excel Co...	
Trial Run.csv	11/20/2024 2:13 PM	Microsoft Excel Co...	
Teacher Observations - 2ndBlock.csv	11/8/2024 9:17 AM	Microsoft Excel Co...	
Teacher Observations - 2ndBlock (1).csv	11/12/2024 8:47 AM	Microsoft Excel Co...	
student_groups_20241121_152716.csv.csv	11/21/2024 2:27 PM	Microsoft Excel Co...	
student_groups_20241121_152633.csv.csv	11/21/2024 2:28 PM	Microsoft Excel Co...	
student_groups_20241121_151544.csv.csv	11/21/2024 2:15 PM	Microsoft Excel Co...	

File name: Microsoft Excel Comma Separa

ings Generator

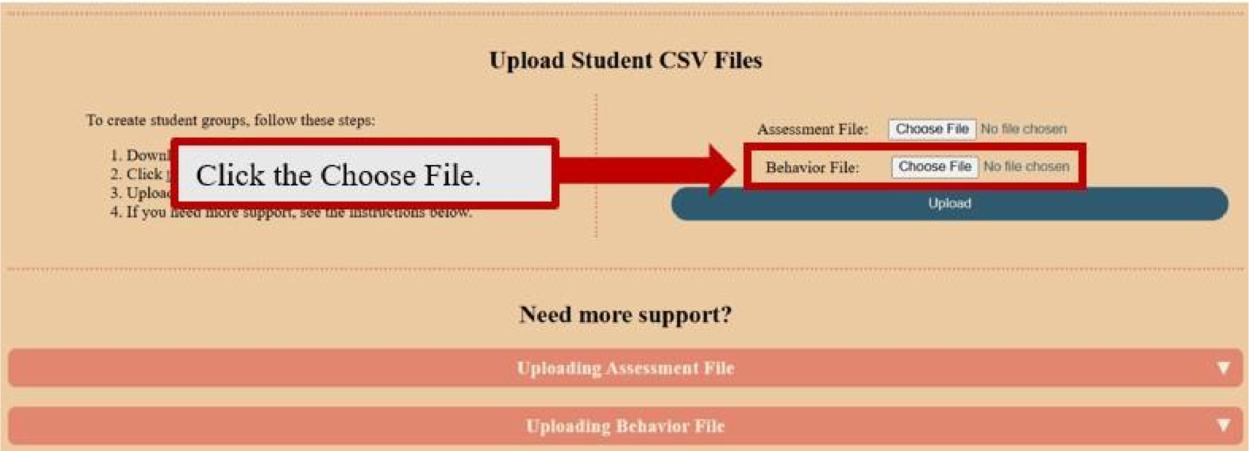
rse classroom groups by combining students of varying academic abilities—high, middle, a
nit your preferences regarding which students you'd like to see together or separated, ensuri
the learning experience for all students. Thank you for using this resource to enrich yo

ent CSV Files

Assessment File: Choose File No file chosen

Behavior File: Choose File No file chosen

Step 3: Click the behavior file Choose File button.



Upload Student CSV Files

To create student groups, follow these steps:

1. Downl
2. Click
3. Upload
4. If you need more support, see the instructions below.

Click the Choose File.

Assessment File: Choose File No file chosen

Behavior File: Choose File No file chosen

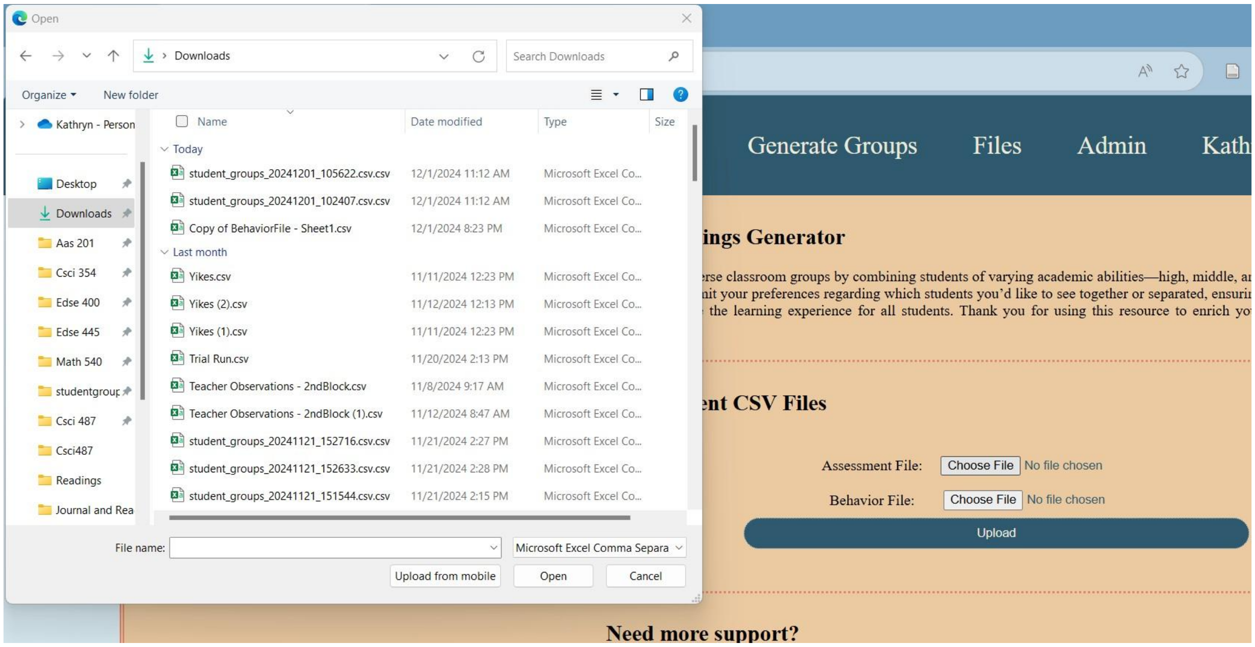
Upload

Need more support?

Uploading Assessment File ▼

Uploading Behavior File ▼

Step 4: Select the Behavior File with the student assessment data to group based on.



Name	Date modified	Type	Size
Today			
student_groups_20241201_105622.csv.csv	12/1/2024 11:12 AM	Microsoft Excel Co...	
student_groups_20241201_102407.csv.csv	12/1/2024 11:12 AM	Microsoft Excel Co...	
Copy of BehaviorFile - Sheet1.csv	12/1/2024 8:23 PM	Microsoft Excel Co...	
Last month			
Yikes.csv	11/11/2024 12:23 PM	Microsoft Excel Co...	
Yikes (2).csv	11/12/2024 12:13 PM	Microsoft Excel Co...	
Yikes (1).csv	11/11/2024 12:23 PM	Microsoft Excel Co...	
Trial Run.csv	11/20/2024 2:13 PM	Microsoft Excel Co...	
Teacher Observations - 2ndBlock.csv	11/8/2024 9:17 AM	Microsoft Excel Co...	
Teacher Observations - 2ndBlock (1).csv	11/12/2024 8:47 AM	Microsoft Excel Co...	
student_groups_20241121_152716.csv.csv	11/21/2024 2:27 PM	Microsoft Excel Co...	
student_groups_20241121_152633.csv.csv	11/21/2024 2:28 PM	Microsoft Excel Co...	
student_groups_20241121_151544.csv.csv	11/21/2024 2:15 PM	Microsoft Excel Co...	

File name:

Microsoft Excel Comma Separa

Upload from mobile | Open | Cancel

Generate Groups   Files   Admin   Kath

ings Generator

erse classroom groups by combining students of varying academic abilities—high, middle, a
nit your preferences regarding which students you'd like to see together or separated, ensurir
the learning experience for all students. Thank you for using this resource to enrich yo

ent CSV Files

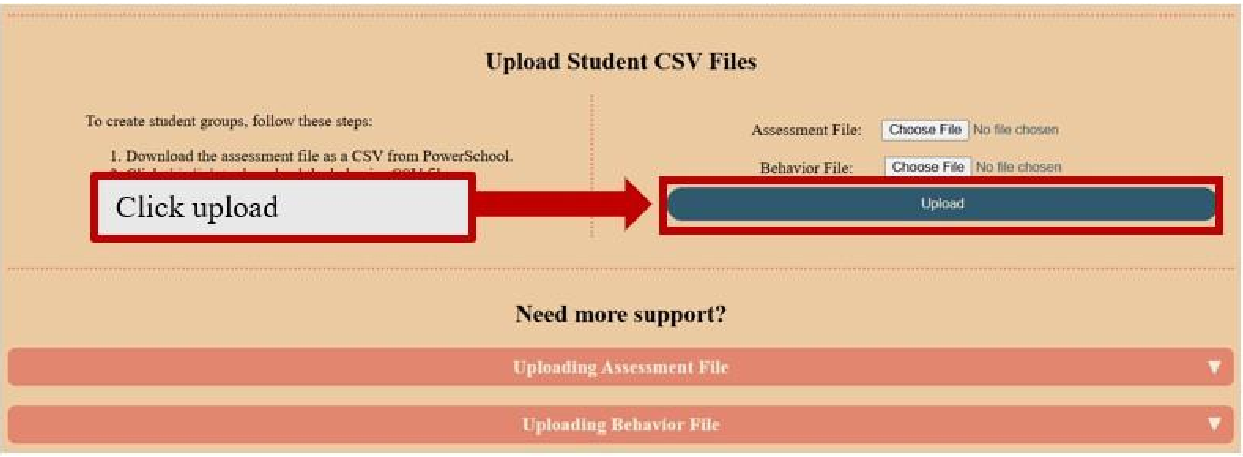
Assessment File: Choose File No file chosen

Behavior File: Choose File No file chosen

Upload

Need more support?

Step 5: Select upload when both files are selected.



Upload Student CSV Files

To create student groups, follow these steps:

1. Download the assessment file as a CSV from PowerSchool.

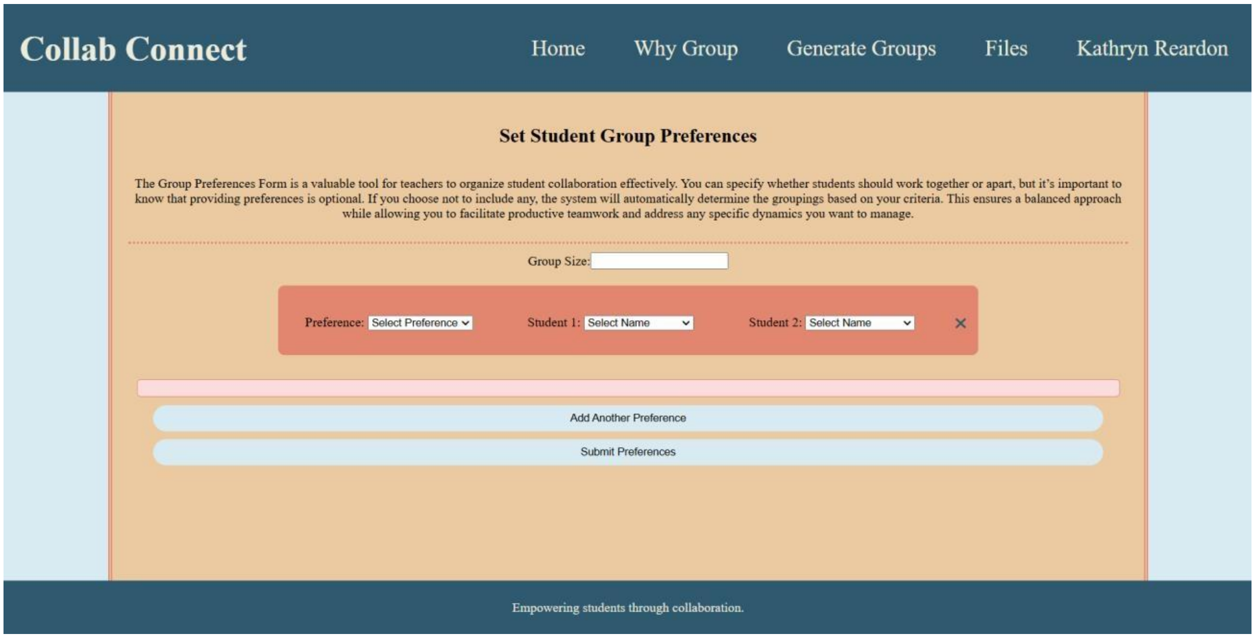
Assessment File: Choose File No file chosen

Behavior File: Choose File No file chosen

Click upload

Need more support?

IV.I.V Preferences Page Overview



Collab Connect

Set Student Group Preferences

The Group Preferences Form is a valuable tool for teachers to organize student collaboration effectively. You can specify whether students should work together or apart, but it's important to know that providing preferences is optional. If you choose not to include any, the system will automatically determine the groupings based on your criteria. This ensures a balanced approach while allowing you to facilitate productive teamwork and address any specific dynamics you want to manage.

Group Size:

Preference: Select Preference   Student 1: Select Name   Student 2: Select Name

Empowering students through collaboration.

Figure D: The Preferences page allows teachers to select the group sizing they want in their classroom and specify any specific student pairs they want to be together or apart when the groups are generated.

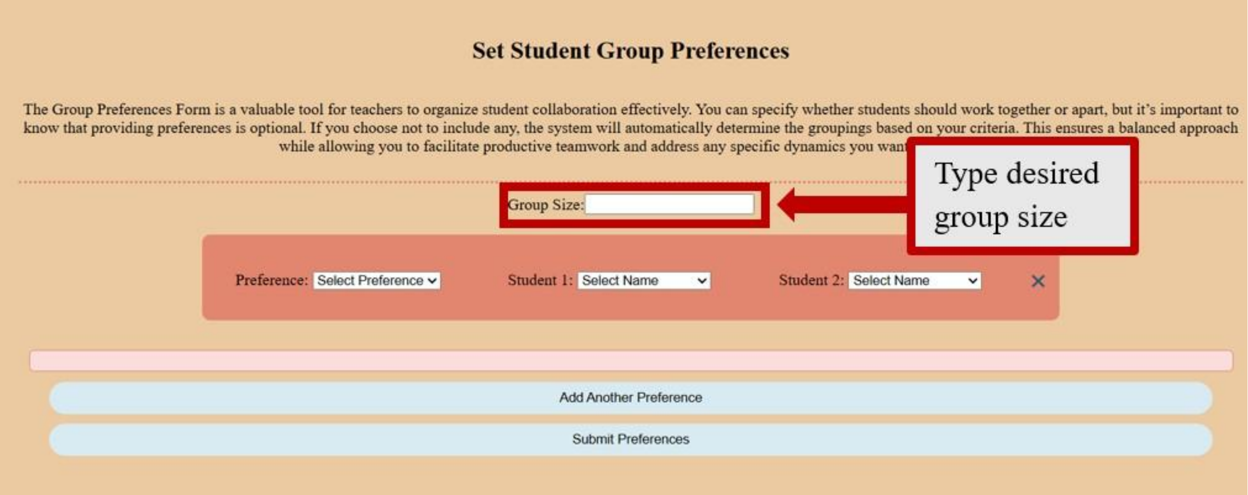
The Preferences page allows teachers to customize the grouping process. It empowers teachers to create groupings tailored to their classroom needs, ensuring a personalized and effective solution. Teachers can specify the desired group size. The group size will be used to generate as many full or close to full groups as possible. For example, for a class size of 21 students, a teacher may select to have group sizes of 4. This will generate three groups of 4 and three groups of 3 as opposed to creating five groups of 4 and one group of 1 to ensure that

all students are benefiting from the groupings. Further, teachers can input preferences about student grouping. For instance, teachers may choose to group certain students or ensure that specific students are placed in separate groups. These preferences are added through an intuitive form, with the option to include multiple criteria by clicking “Add Another Preference.” Once all preferences are entered, teachers click “Submit Preferences” to finalize their input. An error message displays if student is paired to be together with more students than the group size or the same student is not selected for student 1 and student 2 in the same preference.
Disclaimer: If a teacher selects group sizes of 2, one student will be left on their own by the way the algorithm is designed. The teacher will manually have to add that student to a group if they do not want a student on their own.

IV.I.VI Inputting Preferences

To input the desired group size and grouping preferences, the teacher must follow the following steps:

Step 1: Type the desired group size or use the up and down arrows on the far right side of the group size box to select the desired group size.

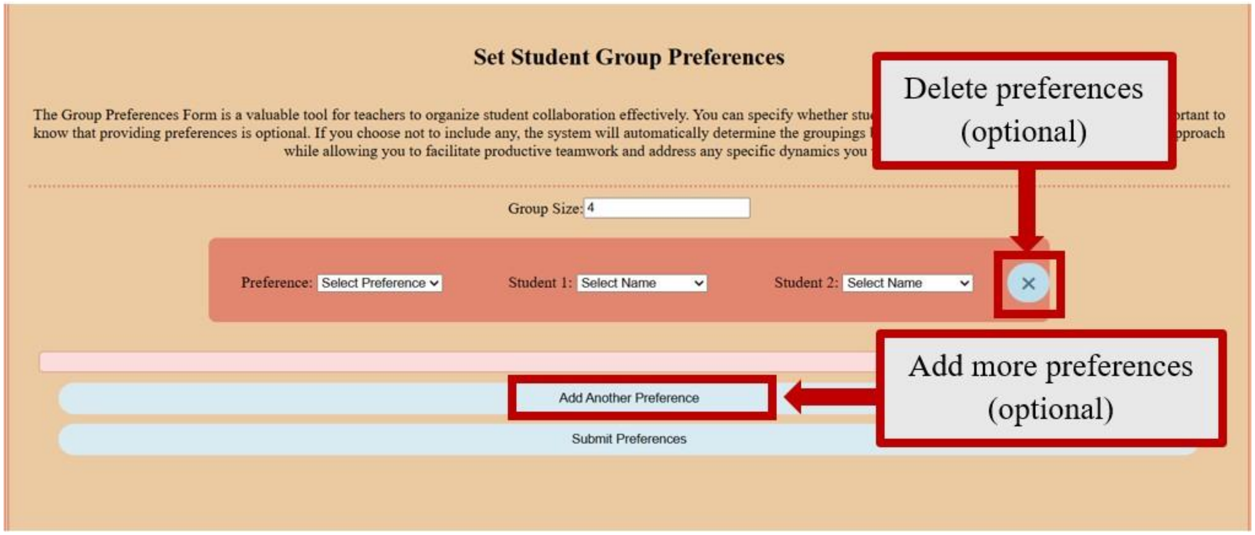


Set Student Group Preferences

The Group Preferences Form is a valuable tool for teachers to organize student collaboration effectively. You can specify whether students should work together or apart, but it’s important to know that providing preferences is optional. If you choose not to include any, the system will automatically determine the groupings based on your criteria. This ensures a balanced approach while allowing you to facilitate productive teamwork and address any specific dynamics you want

Type desired group size

Step 2: Delete any or all preferences that are not desired by clicking the blue "X". Add more preferences if desired by clicking the Add Another Preference button.



Set Student Group Preferences

The Group Preferences Form is a valuable tool for teachers to organize student collaboration effectively. You can specify whether stu rtant to know that providing preferences is optional. If you choose not to include any, the system will automatically determine the groupings b pproach while allowing you to facilitate productive teamwork and address any specific dynamics you

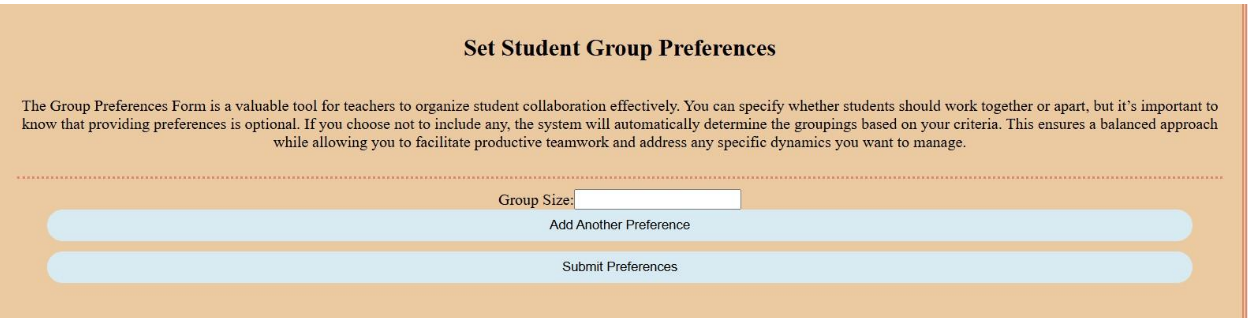
Group Size: 4

Preference: Select Preference Student 1: Select Name Student 2: Select Name

Delete preferences (optional)

Add more preferences (optional)

Step 3: If no preferences are desired, then delete all preference boxes from the page. It should look like the following. A group size still must be input to submit the preference page.

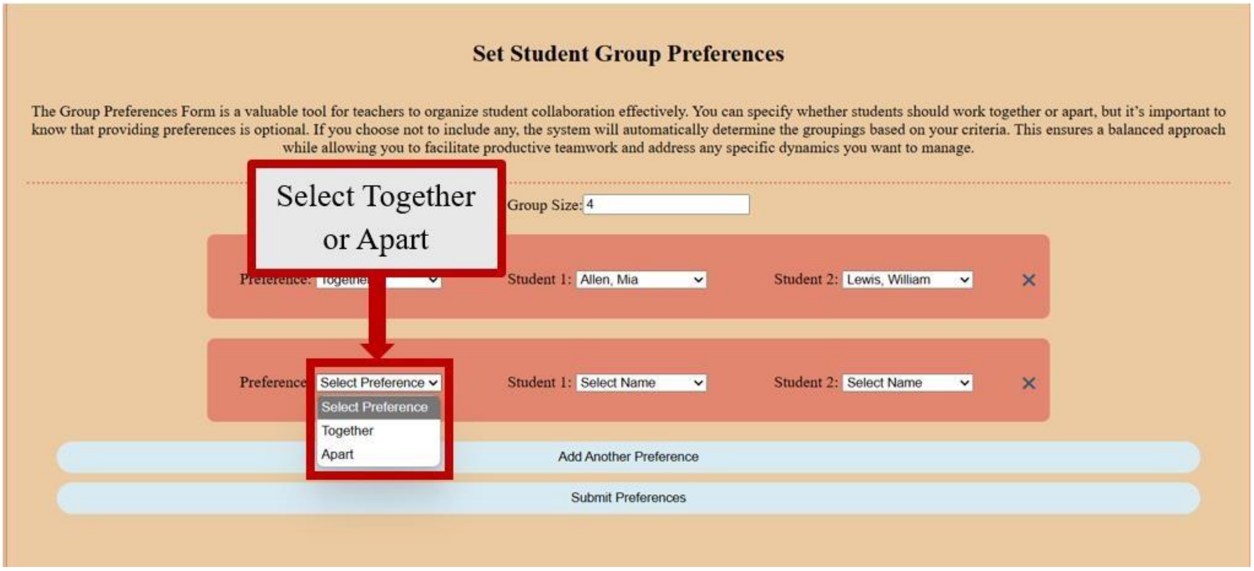


Set Student Group Preferences

The Group Preferences Form is a valuable tool for teachers to organize student collaboration effectively. You can specify whether students should work together or apart, but it's important to know that providing preferences is optional. If you choose not to include any, the system will automatically determine the groupings based on your criteria. This ensures a balanced approach while allowing you to facilitate productive teamwork and address any specific dynamics you want to manage.

Group Size:

Step 4: For desired preferences, select the Preference dropdown menu, and select either Together or Apart. One of these two options must be selected for the preference to be valid.



Set Student Group Preferences

The Group Preferences Form is a valuable tool for teachers to organize student collaboration effectively. You can specify whether students should work together or apart, but it's important to know that providing preferences is optional. If you choose not to include any, the system will automatically determine the groupings based on your criteria. This ensures a balanced approach while allowing you to facilitate productive teamwork and address any specific dynamics you want to manage.

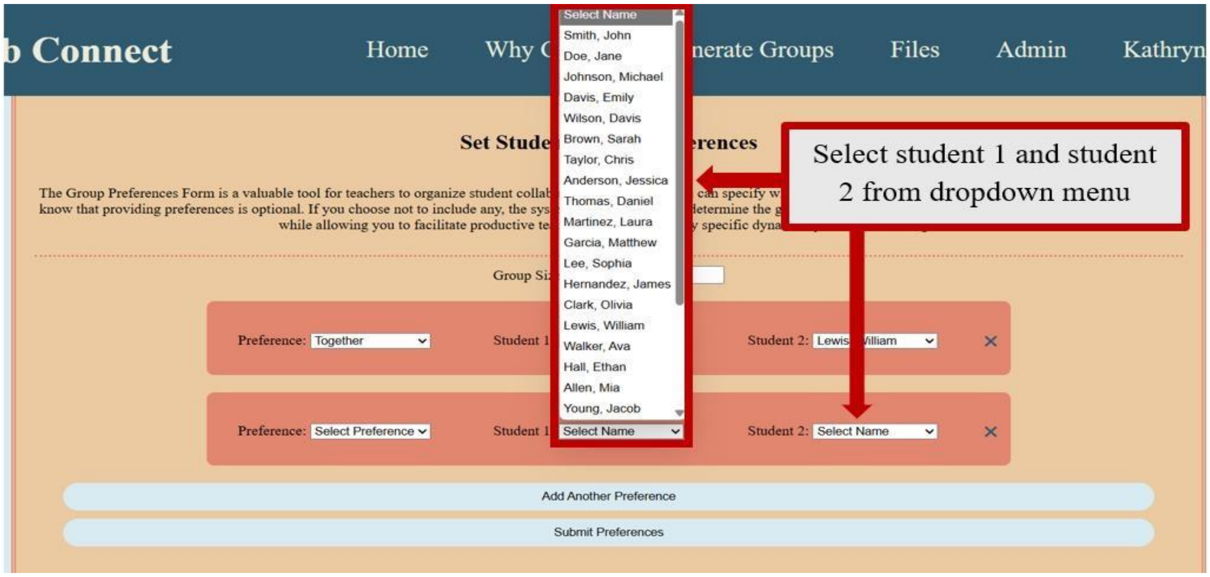
Select Together or Apart

Group Size: 4

Preference: Together | Student 1: Allen, Mia | Student 2: Lewis, William | ×

Preference: Select Preference | Student 1: Select Name | Student 2: Select Name | ×

Step 5: For desired preferences, select student 1 and student 2, the two students that should be put together or apart, from the dropdown menu. Student 1 and student 2 must be selected for it to be a valid preference. The same student cannot be selected as student 1 and student 2. A student may be selected more than once from the dropdown list for preferences as long as they are not selected as student 1 and student 2 in the same preference. A student cannot exceed one less than the group size in together pairings.

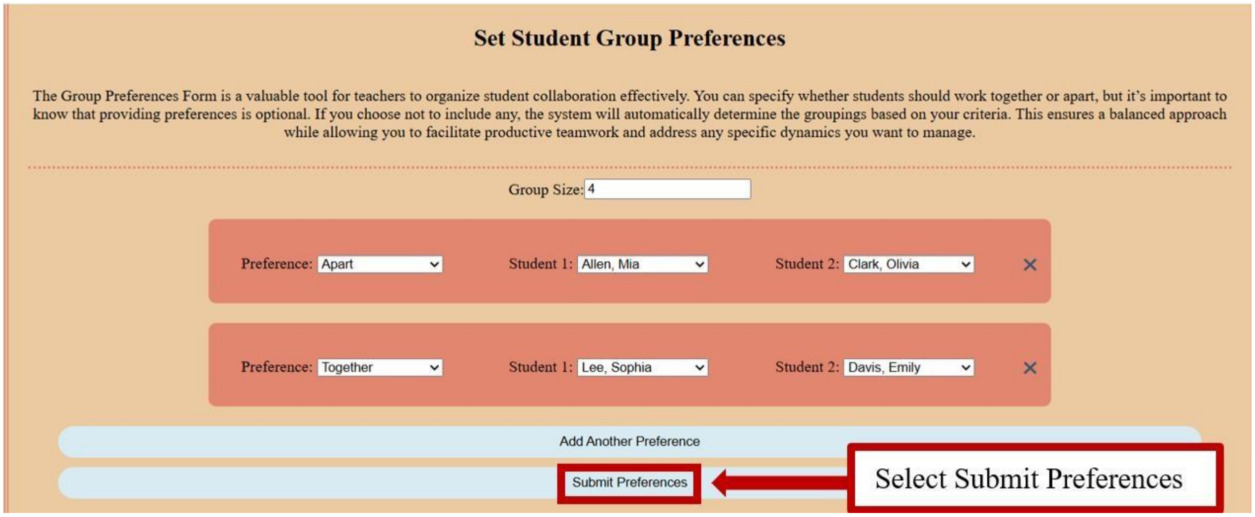


Select student 1 and student 2 from dropdown menu

Preference: Together | Student 1 | Student 2: Lewis, William | ×

Preference: Select Preference | Student 1: Select Name | Student 2: Select Name | ×

Step 6: When group size and all desired preferences are selected, click Submit Preferences to generate groups.



Set Student Group Preferences

The Group Preferences Form is a valuable tool for teachers to organize student collaboration effectively. You can specify whether students should work together or apart, but it's important to know that providing preferences is optional. If you choose not to include any, the system will automatically determine the groupings based on your criteria. This ensures a balanced approach while allowing you to facilitate productive teamwork and address any specific dynamics you want to manage.

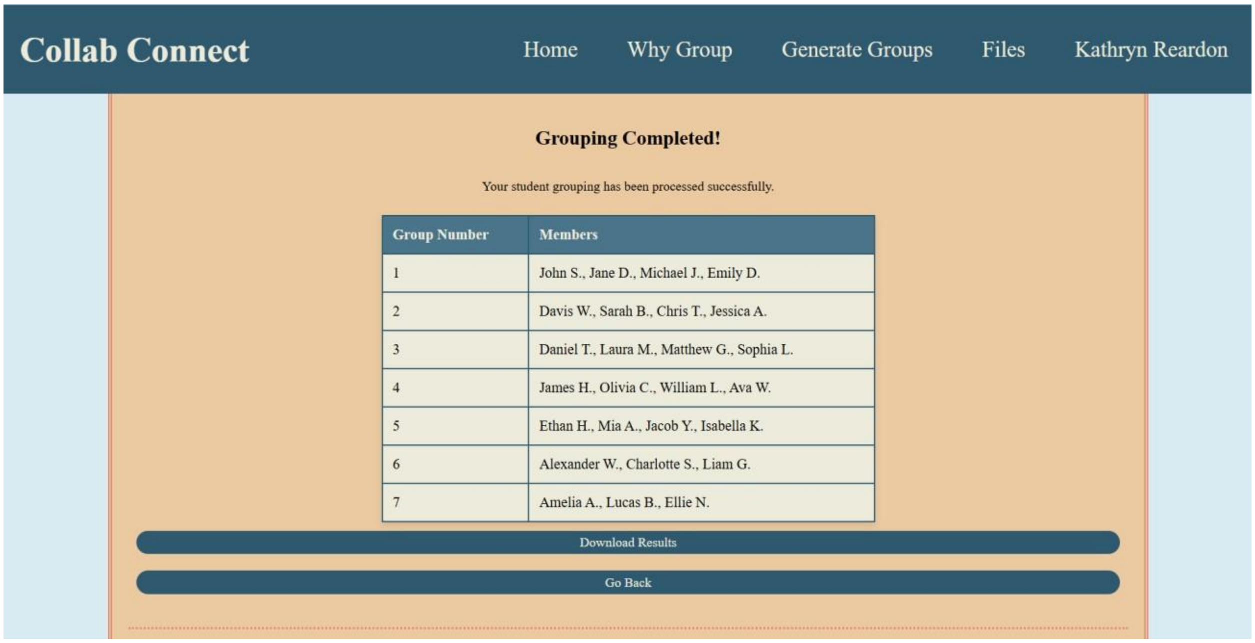
Group Size: 4

Preference: Apart Student 1: Allen, Mia Student 2: Clark, Olivia

Preference: Together Student 1: Lee, Sophia Student 2: Davis, Emily

Select Submit Preferences

IV.I.VII Group Results Page Overview



Grouping Completed!

Your student grouping has been processed successfully.

Group Number	Members
1	John S., Jane D., Michael J., Emily D.
2	Davis W., Sarah B., Chris T., Jessica A.
3	Daniel T., Laura M., Matthew G., Sophia L.
4	James H., Olivia C., William L., Ava W.
5	Ethan H., Mia A., Jacob Y., Isabella K.
6	Alexander W., Charlotte S., Liam G.
7	Amelia A., Lucas B., Ellie N.

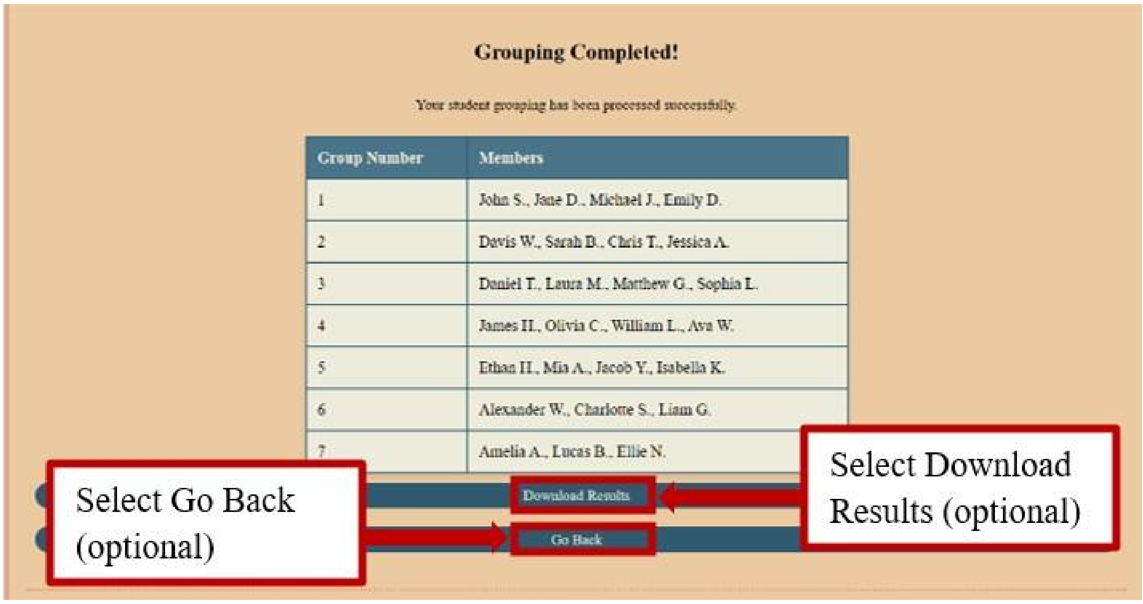
Figure E: The Result page displays the results in a table with group numbers and the members of each group based on the assessment data, behavior data, and teacher preferences. A teacher has the option to download the results.

The Results page displays the completed student groupings in an organized table format. Each group is labeled with a group number, and its members are listed beneath it. Teachers can review the groupings directly on this page to ensure they align with their preferences and classroom goals. The page also provides an option to download the groupings as a CSV file for offline use or further sharing. If adjustments are needed, the “Go Back” button allows teachers to return to the Upload Files pages to refine their inputs. This page ensures that teachers have quick and convenient access to their finalized groupings.

IV.I.VIII Result File

Select Download Results to download a CSV file with the group number, group members, member compatibility scores, and group compatibility scores. The member compatibility scores inform the teacher of how compatible each pair of students within each group is to be able to best support each other. The group compatibility scores, like the member compatibility score, inform the teacher of how well the group as a whole can support each other and is the average of the member compatibility scores. Lower compatibility scores mean that a teacher may want to provide more support to the group.

A teacher may also want to rerun the generated group and to do this, they may select the go-back button. However, this button will take the teacher back to the upload file page. The algorithm for generating groups is deterministic, meaning that it will produce the same groups if run with the exact same data and preferences, so to get new groups, the teacher must update either the assessment file, behavior file, group size, or preferences.

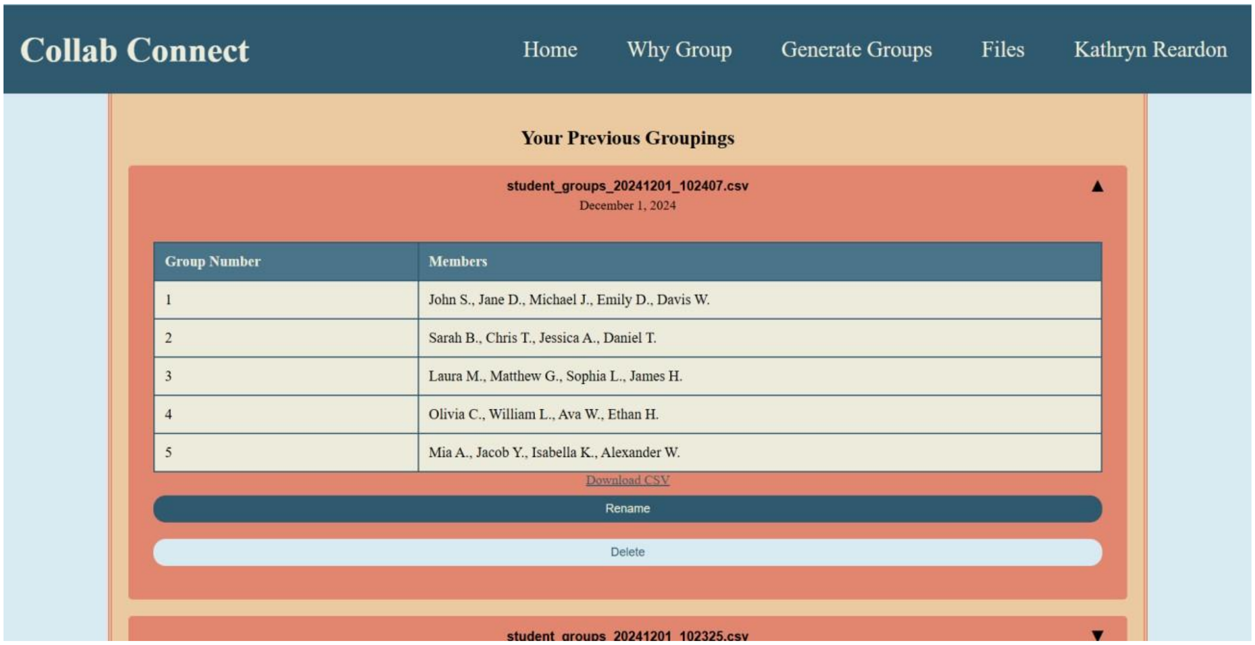


Grouping Completed!

Your student grouping has been processed successfully.

Group Number	Members
1	John S., Jane D., Michael J., Emily D.
2	Davis W., Sarah B., Chris T., Jessica A.
3	Daniel T., Laura M., Matthew G., Sophia L.
4	James H., Olivia C., William L., Ava W.
5	Ethan H., Mia A., Jacob Y., Isabella K.
6	Alexander W., Charlotte S., Liam G.
7	Amelia A., Lucas B., Ellie N.

IV.II FILES PAGE OVERVIEW



Your Previous Groupings

student_groups_20241201_102407.csv
December 1, 2024

Group Number	Members
1	John S., Jane D., Michael J., Emily D., Davis W.
2	Sarah B., Chris T., Jessica A., Daniel T.
3	Laura M., Matthew G., Sophia L., James H.
4	Olivia C., William L., Ava W., Ethan H.
5	Mia A., Jacob Y., Isabella K., Alexander W.

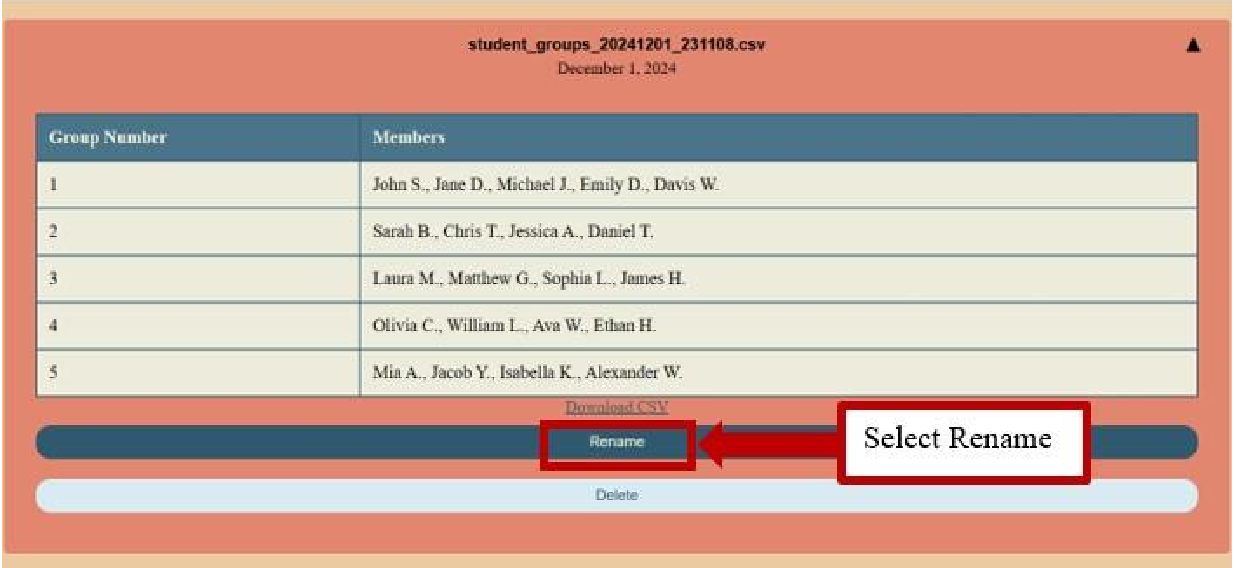
student_groups_20241201_102325.csv

Figure F: The File page displays all prior grouping results in a dropdown format. The teacher can rename the file, download the file, or delete past files.

The Files page serves as a repository for all previously generated groupings that a user can access when logged into their account. It displays a list of past groupings, organized by filenames and timestamps, allowing teachers to easily locate and manage their files. Teachers can rename files for better organization, delete files that are no longer useful, or download groupings as CSV files for a compatibility report on groupings. The rename function will display an error message if the filename they input contains forbidden characters such as “/” or “\.” This is to ensure that the file is downloadable with the filename the teacher inputs. Its straightforward layout makes file management simple and efficient.

IV.II.I Rename File

Step 1: Select Rename to start the renaming file process.

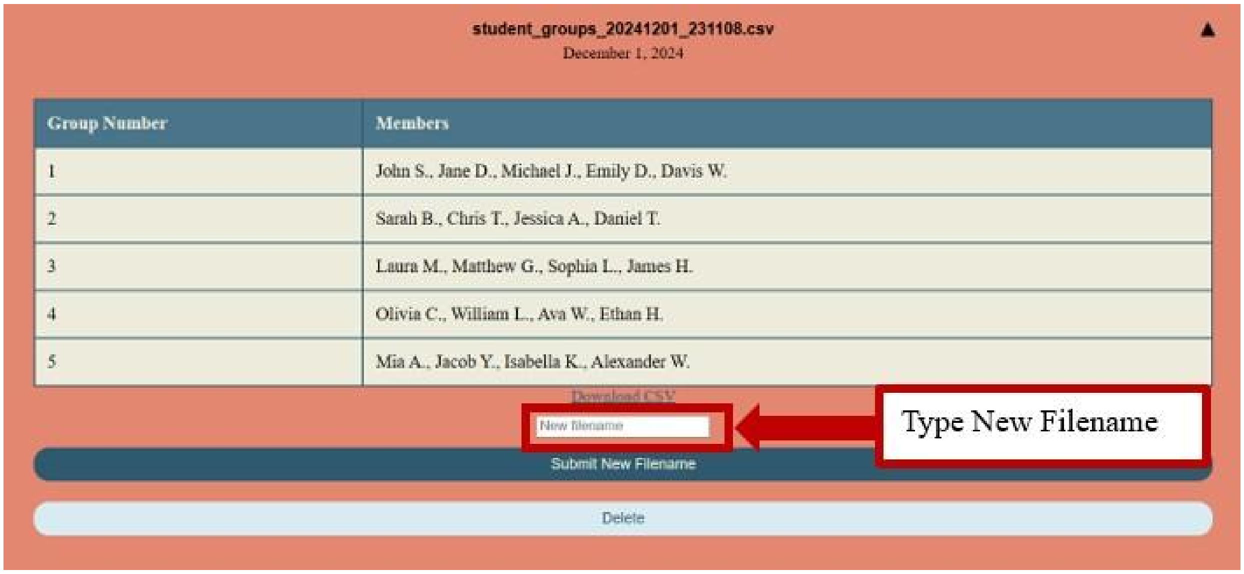


student_groups_20241201_231108.csv

December 1, 2024

Group Number	Members
1	John S., Jane D., Michael J., Emily D., Davis W.
2	Sarah B., Chris T., Jessica A., Daniel T.
3	Laura M., Matthew G., Sophia L., James H.
4	Olivia C., William L., Ava W., Ethan H.
5	Mia A., Jacob Y., Isabella K., Alexander W.

Step 2: Type in the new filename in the new filename box. Be sure not to use "/" or "\" as these are forbidden characters when downloading a file.

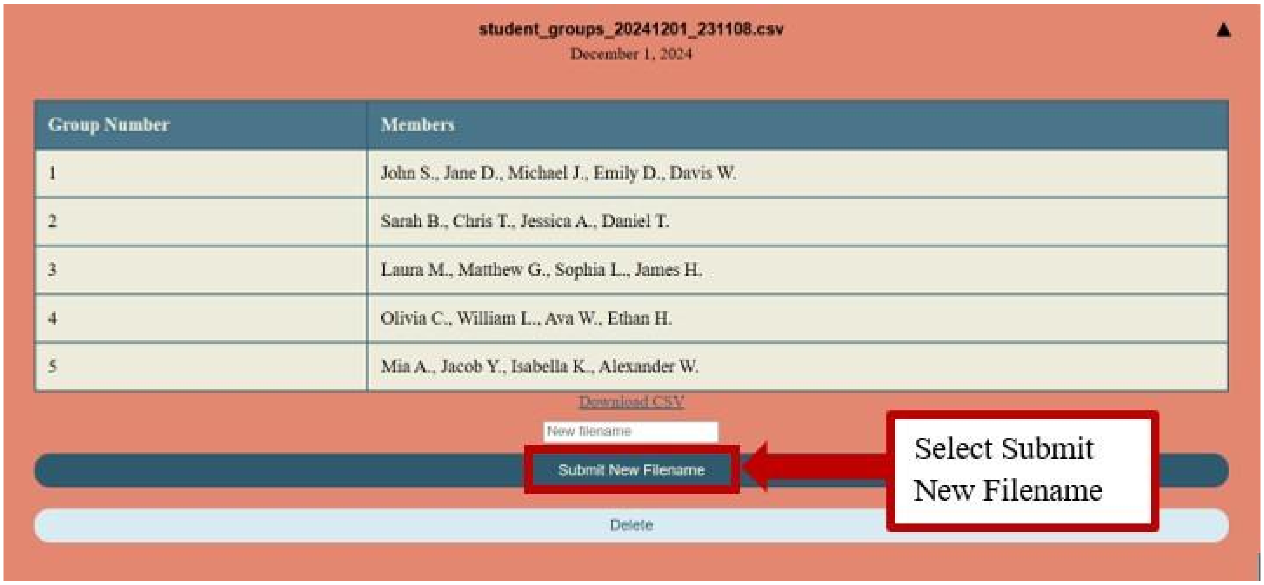


student_groups_20241201_231108.csv

December 1, 2024

Group Number	Members
1	John S., Jane D., Michael J., Emily D., Davis W.
2	Sarah B., Chris T., Jessica A., Daniel T.
3	Laura M., Matthew G., Sophia L., James H.
4	Olivia C., William L., Ava W., Ethan H.
5	Mia A., Jacob Y., Isabella K., Alexander W.

Step 3: Select Submit New Filename when the filename is typed in the new filename box to update the filename.



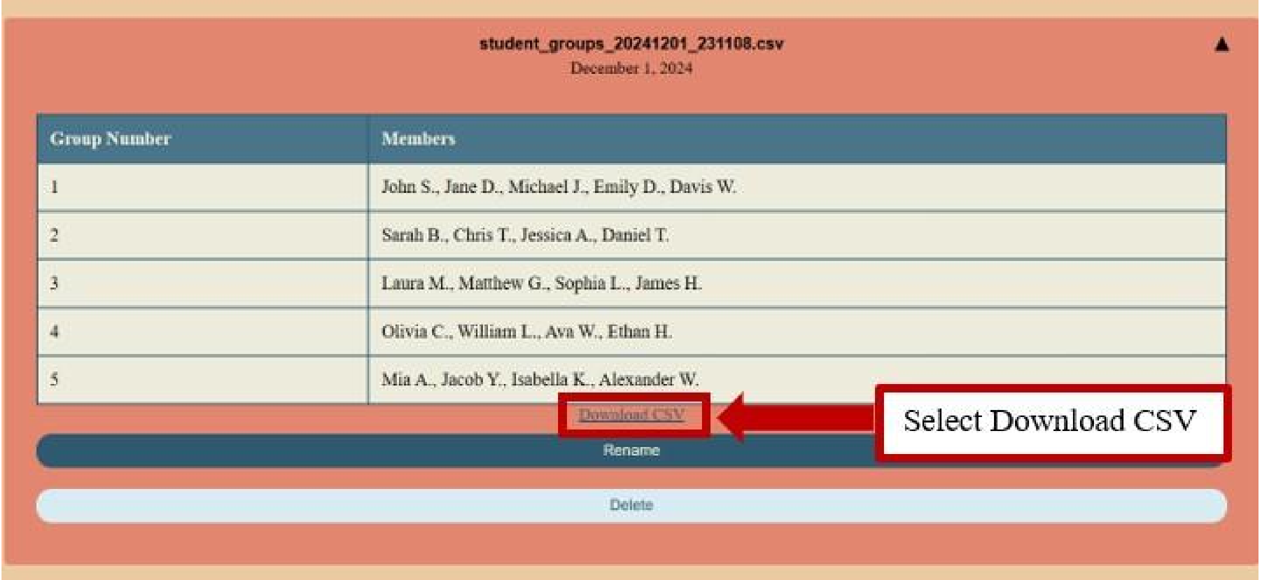
student_groups_20241201_231108.csv
December 1, 2024

Group Number	Members
1	John S., Jane D., Michael J., Emily D., Davis W.
2	Sarah B., Chris T., Jessica A., Daniel T.
3	Laura M., Matthew G., Sophia L., James H.
4	Olivia C., William L., Ava W., Ethan H.
5	Mia A., Jacob Y., Isabella K., Alexander W.

Select Submit New Filename

IV.II.II Download File

Select the Download CSV link to download the CSV file with the group number, the members, the member compatibility scores, and the group compatibility score.



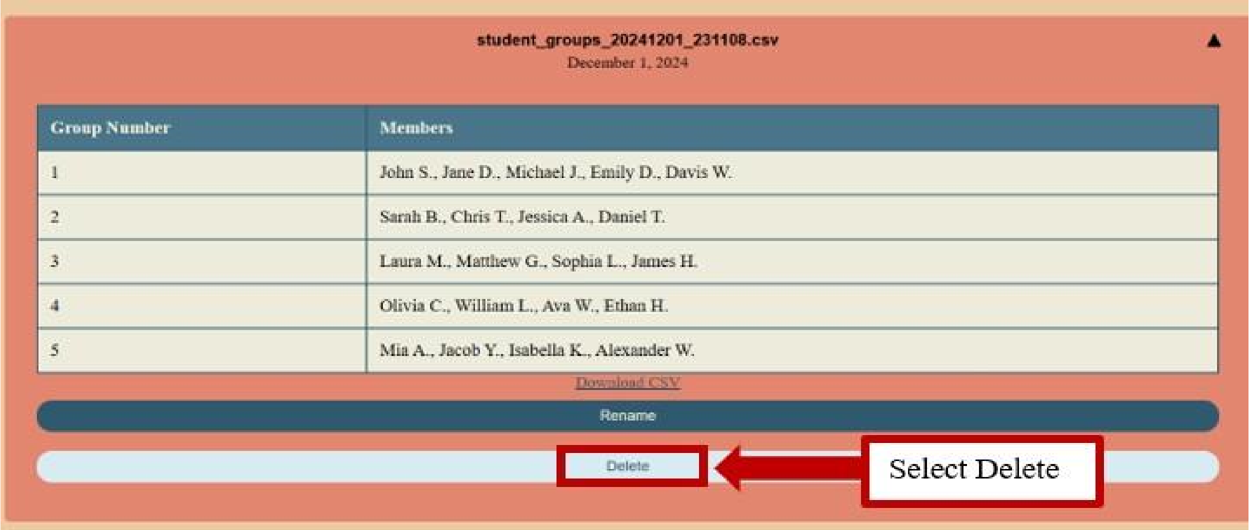
student_groups_20241201_231108.csv
December 1, 2024

Group Number	Members
1	John S., Jane D., Michael J., Emily D., Davis W.
2	Sarah B., Chris T., Jessica A., Daniel T.
3	Laura M., Matthew G., Sophia L., James H.
4	Olivia C., William L., Ava W., Ethan H.
5	Mia A., Jacob Y., Isabella K., Alexander W.

Select Download CSV

IV.II.III Delete File

Select the Delete button to delete the group result. This will permanently remove the file. It will have a box pop up that asks if you are sure you want to delete the file. Select ok if you are sure you want to delete the file. Select cancel otherwise.



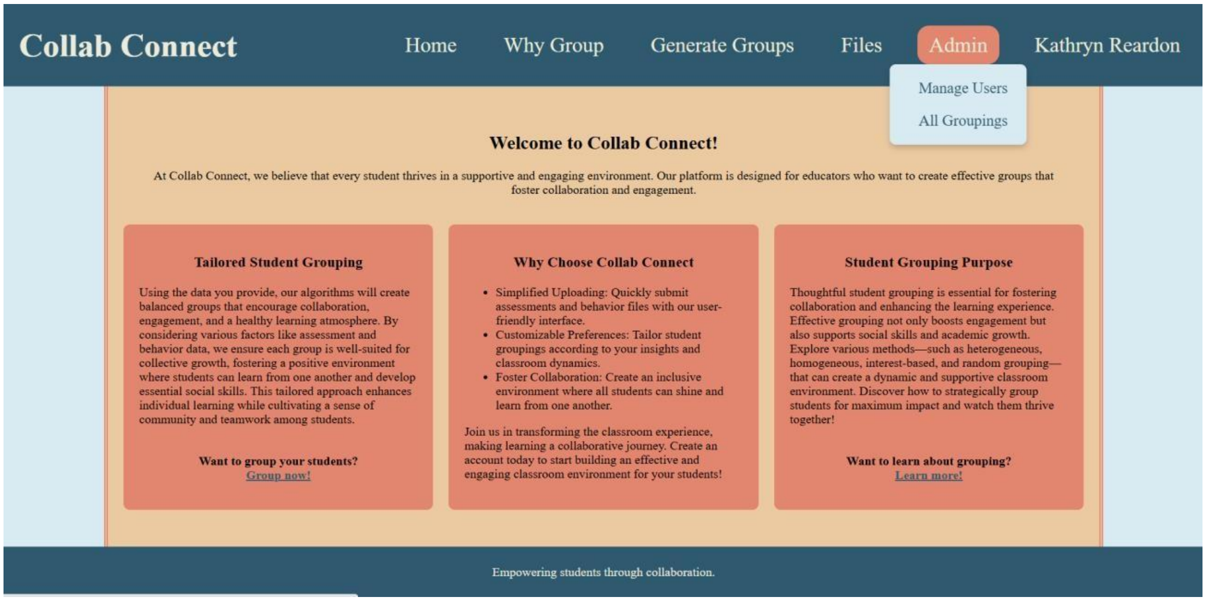
student_groups_20241201_231108.csv

December 1, 2024

Group Number	Members
1	John S., Jane D., Michael J., Emily D., Davis W.
2	Sarah B., Chris T., Jessica A., Daniel T.
3	Laura M., Matthew G., Sophia L., James H.
4	Olivia C., William L., Ava W., Ethan H.
5	Mia A., Jacob Y., Isabella K., Alexander W.

V. USE: ADMIN PAGES

V.I ADMIN VIEW



Welcome to Collab Connect!

At Collab Connect, we believe that every student thrives in a supportive and engaging environment. Our platform is designed for educators who want to create effective groups that foster collaboration and engagement.

Tailored Student Grouping

Using the data you provide, our algorithms will create balanced groups that encourage collaboration, engagement, and a healthy learning atmosphere. By considering various factors like assessment and behavior data, we ensure each group is well-suited for collective growth, fostering a positive environment where students can learn from one another and develop essential social skills. This tailored approach enhances individual learning while cultivating a sense of community and teamwork among students.

Want to group your students?
Group now!

Why Choose Collab Connect

- Simplified Uploading: Quickly submit assessments and behavior files with our user-friendly interface.
- Customizable Preferences: Tailor student groupings according to your insights and classroom dynamics.
- Foster Collaboration: Create an inclusive environment where all students can shine and learn from one another.

Join us in transforming the classroom experience, making learning a collaborative journey. Create an account today to start building an effective and engaging classroom environment for your students!

Student Grouping Purpose

Thoughtful student grouping is essential for fostering collaboration and enhancing the learning experience. Effective grouping not only boosts engagement but also supports social skills and academic growth. Explore various methods—such as heterogeneous, homogeneous, interest-based, and random grouping—that can create a dynamic and supportive classroom environment. Discover how to strategically group students for maximum impact and watch them thrive together!

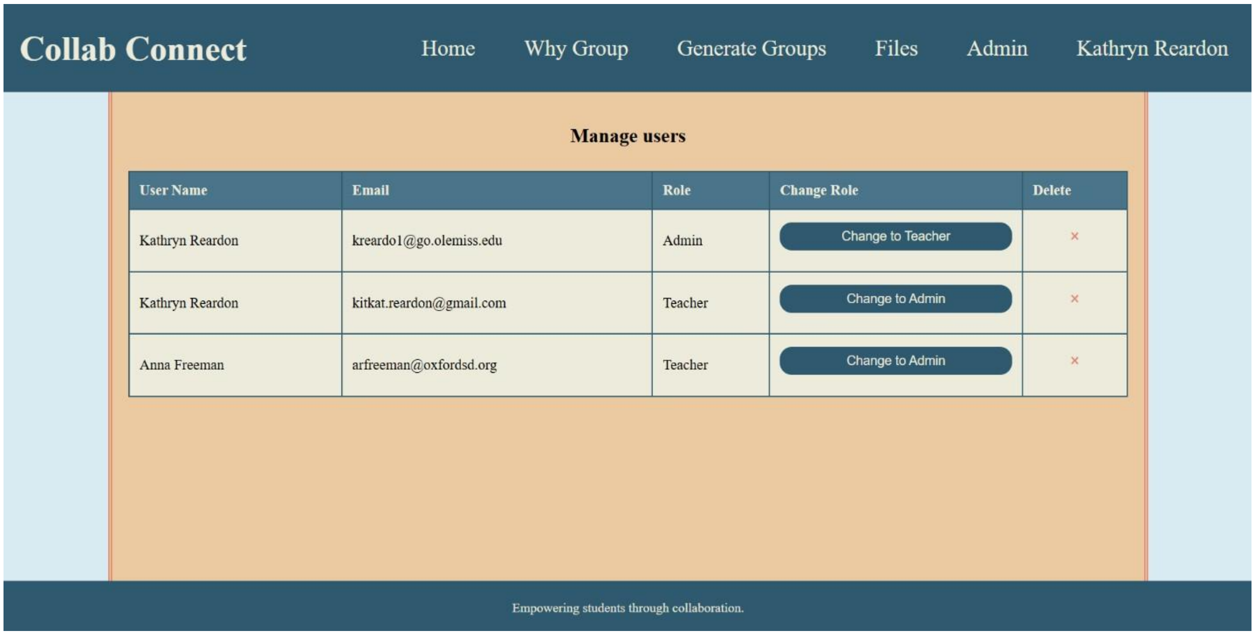
Want to learn about grouping?
Learn more!

Empowering students through collaboration.

Figure G: The administrators have an extra option in their navigation bar to view all users and all users' groupings.

Admin View provides administrators with tools to manage users and oversee platform activity. This section is divided into two main pages: Manage Users and User Groupings. Together, these pages ensure that administrators can maintain platform functionality, assign roles, and monitor the group generation process. The admin must be logged in and their role must be set to admin to have access to these pages.

V.I.I Manage Users Page Overview



Manage users

User Name	Email	Role	Change Role	Delete
Kathryn Reardon	kreardo1@go.olemiss.edu	Admin	Change to Teacher	×
Kathryn Reardon	kitkat.reardon@gmail.com	Teacher	Change to Admin	×
Anna Freeman	arfreeman@oxfordsd.org	Teacher	Change to Admin	×

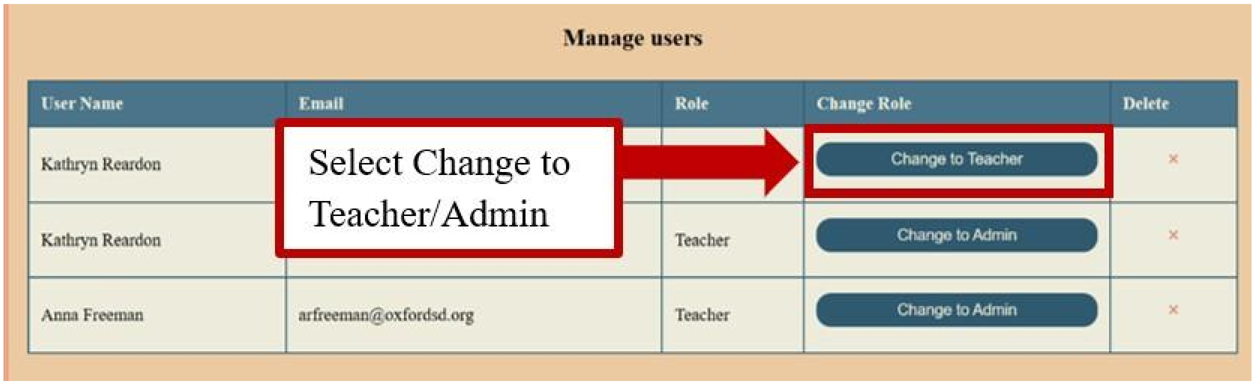
Empowering students through collaboration.

Figure H: The Manage Users page is where administrators can update the role of a user or delete a user. They can also view user information such as their email.

The Manage Users page is where administrators manage user accounts. It provides a comprehensive table with all registered users, including their names, email addresses, and user role. There are two user roles: administrators and teachers. Administrators can use the "Change Role" button to demote administrators to teachers or promote teachers to administrators. Further, they can use the pink "X" button to delete inactive accounts. This page ensures that administrators can efficiently maintain an organized user base and assign roles based on the platform's needs.

V.I.II Changing User Role

Select Change to Teacher to change the role of an admin to a teacher or select Change to Admin to change the role of the teacher to an admin.



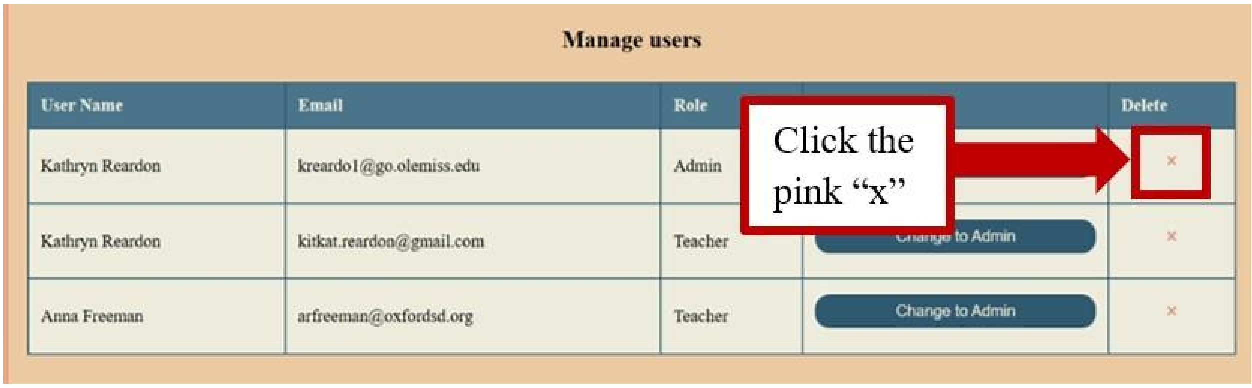
Manage users

User Name	Email	Role	Change Role	Delete
Kathryn Reardon			Change to Teacher	×
Kathryn Reardon		Teacher	Change to Admin	×
Anna Freeman	arfreeman@oxfordsd.org	Teacher	Change to Admin	×

Select Change to Teacher/Admin

V.I.III Deleting A User

Click the pink "X" to remove a user and all their groupings.

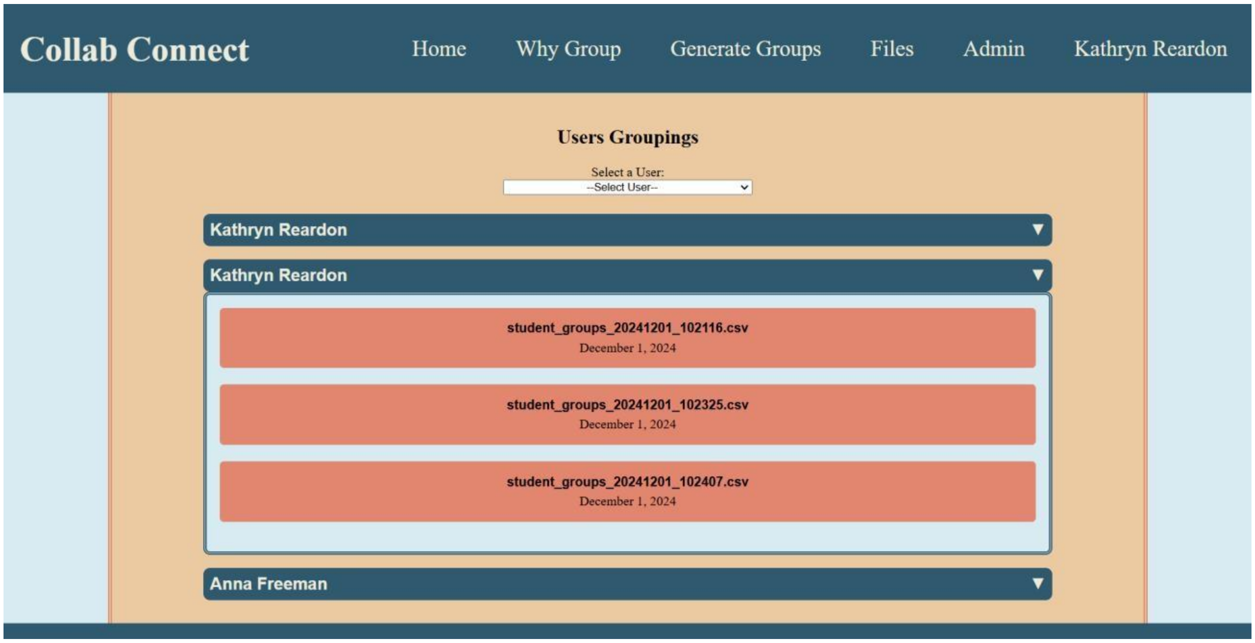


Manage users

User Name	Email	Role		Delete
Kathryn Reardon	kreardo1@go.olemiss.edu	Admin		×
Kathryn Reardon	kitkat.reardon@gmail.com	Teacher	Change to Admin	×
Anna Freeman	arfreeman@oxfordsd.org	Teacher	Change to Admin	×

Click the pink "x"

V.I.IV User Groupings Page Overview



Users Groupings

Select a User:
--Select User--

Kathryn Reardon ▼

Kathryn Reardon ▼

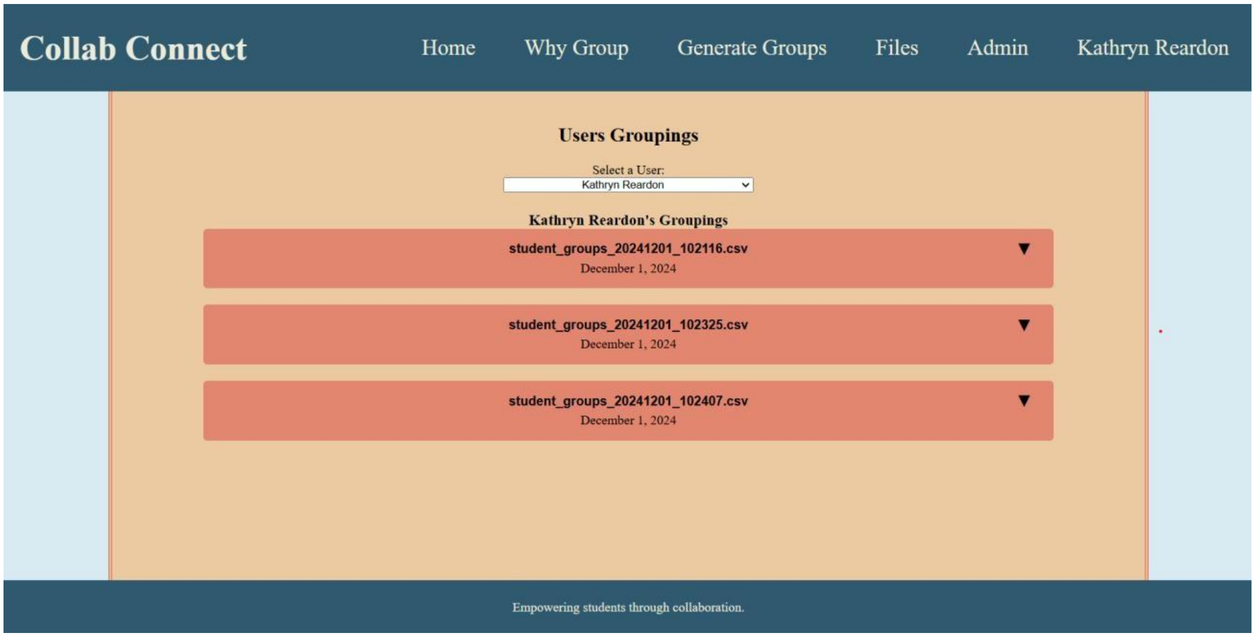
student_groups_20241201_102116.csv
December 1, 2024

student_groups_20241201_102325.csv
December 1, 2024

student_groups_20241201_102407.csv
December 1, 2024

Anna Freeman ▼

Figure I: The user groupings page displays a list of all the users and the groupings they have created. An admin may click a specific users name to view their groupings.



Users Groupings

Select a User:
Kathryn Reardon

Kathryn Reardon's Groupings

student_groups_20241201_102116.csv ▼
December 1, 2024

student_groups_20241201_102325.csv ▼
December 1, 2024

student_groups_20241201_102407.csv ▼
December 1, 2024

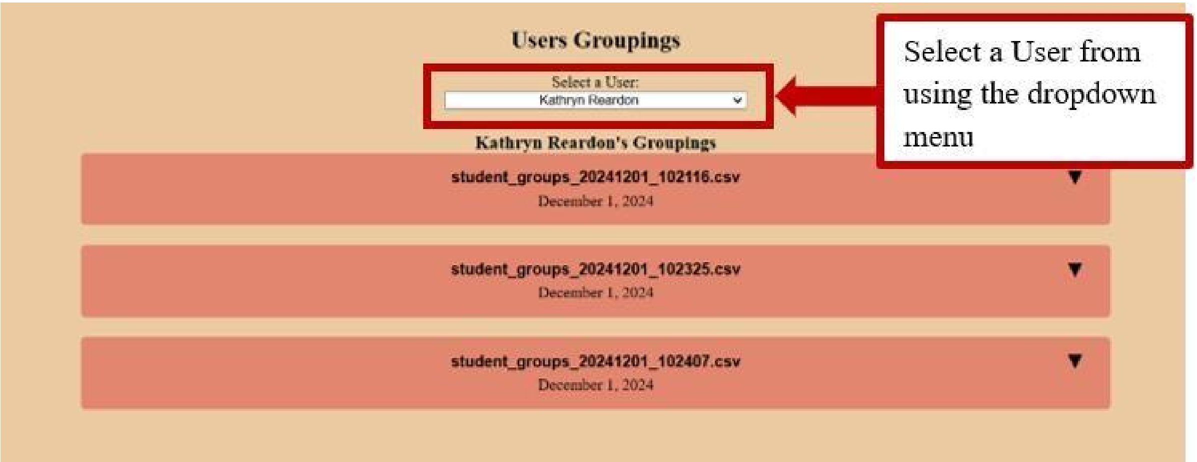
Empowering students through collaboration.

Figure J: An administrator can also use the select a user dropdown to select a specific user to view their groupings. This is an ideal option when the list of users is very long to quickly find the specific user.

The User Groupings page allows administrators to view groupings created by specific teachers. An administrator can click a user's name to view their grouping as seen in Figure 1 or use the select a user dropdown to select a user as seen in Figure 2. Using a dropdown menu, administrators can select a teacher and view their generated groupings, which are displayed in an expandable list format. Each grouping includes a filename, timestamp, and options to view or download the associated CSV file. This page provides administrators with insights into how teachers are using the platform and ensures accountability in the grouping process. Its organized structure makes it easy for administrators to monitor activity and access relevant data.

V.I.V Select a User's Groupings

Select a user from the dropdown menu to view that specific user's groupings.



Users Groupings

Select a User: Kathryn Reardon

Kathryn Reardon's Groupings

student_groups_20241201_102116.csv
December 1, 2024

student_groups_20241201_102325.csv
December 1, 2024

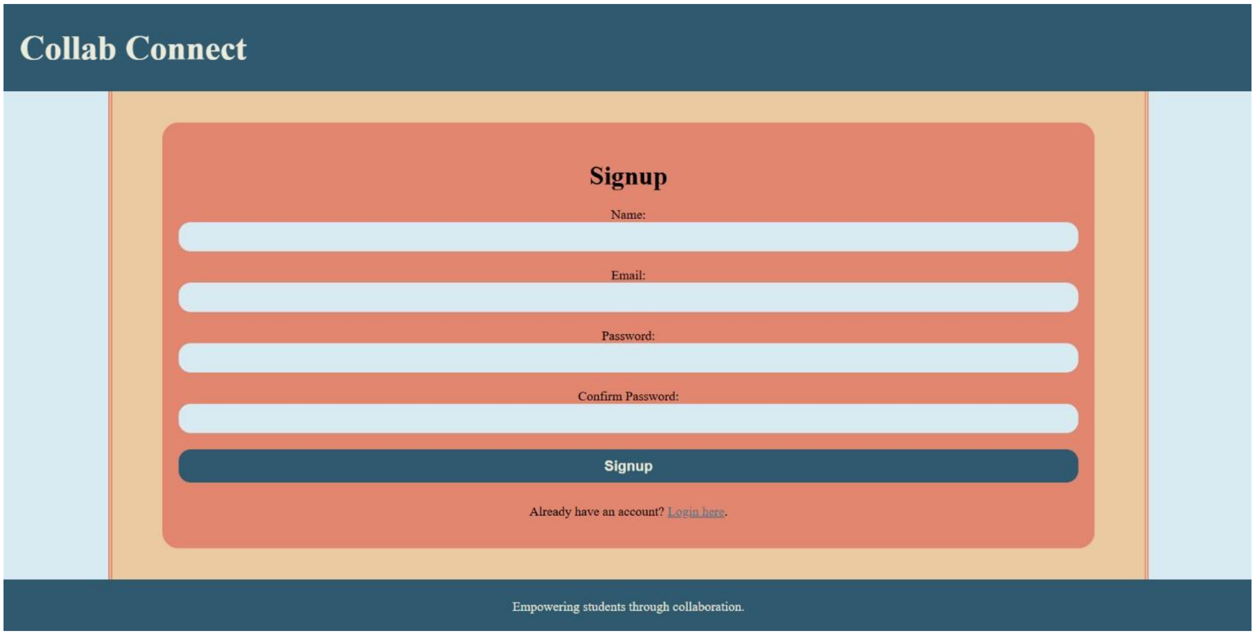
student_groups_20241201_102407.csv
December 1, 2024

Select a User from using the dropdown menu

VI. USE: AUTHENTICATION PAGES

The Collab Connect platform includes three authentication pages to manage user access. While on an authentication page, the navigation bar is unavailable. However, to go back to the home page, the user can click on Collab Connect.

VI.I.I Register Page



Already have an account? Login here.

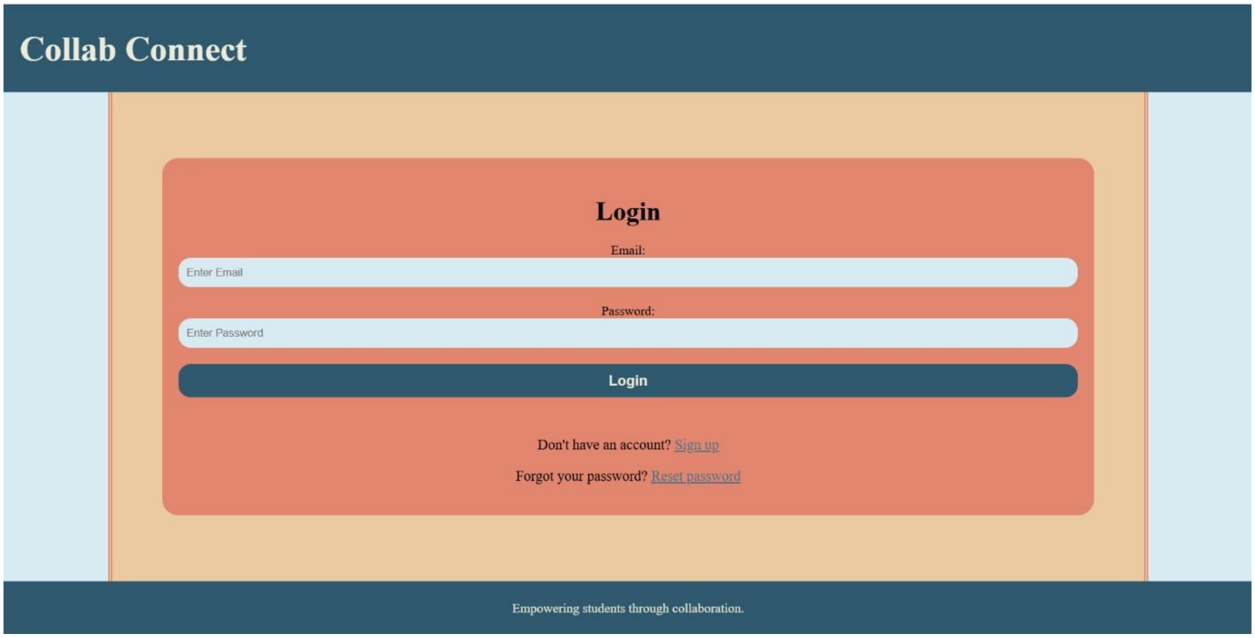
Empowering students through collaboration.

Figure K: The Signup page allows a user to create an account. The user must register using their name, email, and password.

The Register page enables new users to create an account by providing their name, email address, and password. The page includes validation to ensure that all fields are filled correctly. An error message will display for instances such as the password and confirm password being different and an email is used that is already registered. A To complete registration, a verification email will be sent to the email specified in the sign up. The user must verify their email to fully register to gain access to the features of the website. A user will be automatically set to a teacher role. To gain access to the administrator role, another administrator must go in and promote them to an administrator.

Disclaimer: The first initial administrator will have to be added manually through the database instead of through the web application as there is no other administrator to upgrade the user to an administrator.

VI.I.II Login Page



Don't have an account? Sign up

Forgot your password? Reset password

Empowering students through collaboration.

Figure L: The Login page allows a user to input their email and password to access their account. They can also sign up or reset their password if needed.

The Login page provides an interface for users to access their accounts. They must be logged in to access the Generate Group pages, Files page, and Admin View pages. Teachers and administrators can log in using their email addresses and passwords. Error messages guide users if they enter incorrect credentials.

VI.I.III Forgot Password Page

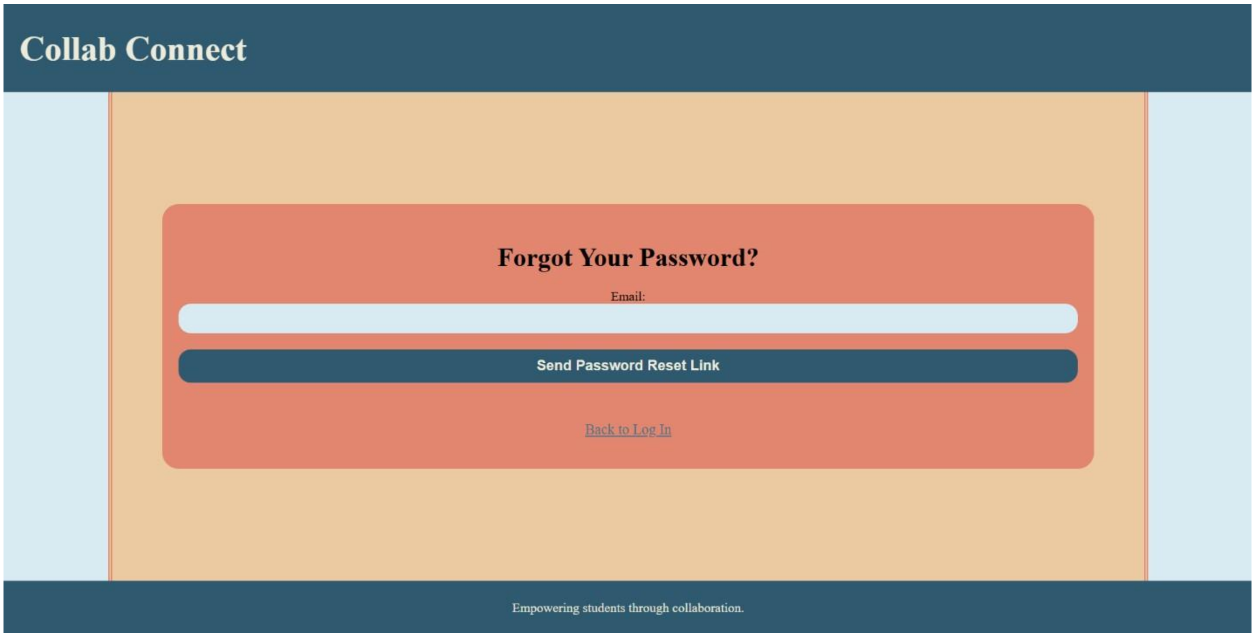
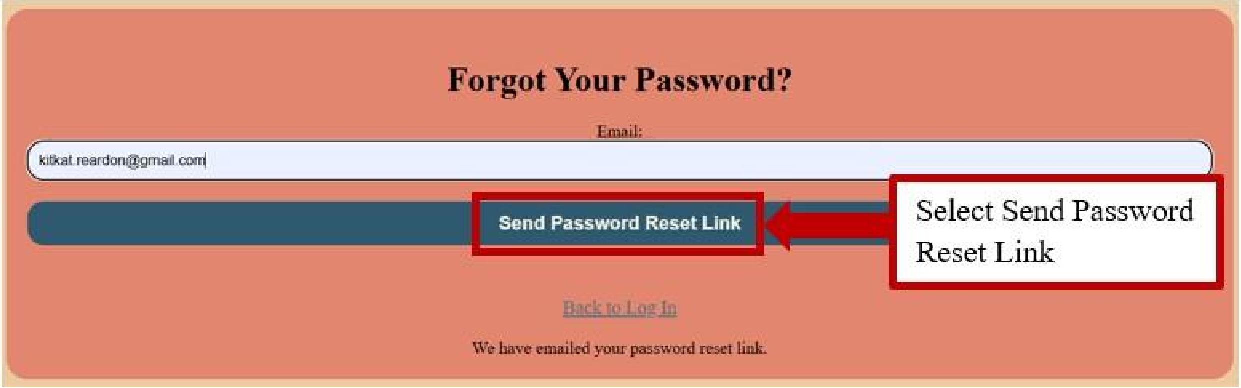


Figure M: The Forgot Password page allows a user to input a registered email to send a reset password link to the email.

The Forgot Password page allows users to reset their passwords by entering their registered email address. A reset link is sent to their email, enabling users to regain access quickly and securely. An error message is displayed if the email input is not a registered email. A verification message will display if a reset password email is sent.

VI.I.VI Resetting a Password

Step 1: Type in a registered email for the website. Then select Send Password Reset Link to send a password reset email to the email that was input. A "We have emailed your password reset link." notification will show on the password reset page when the email is sent.

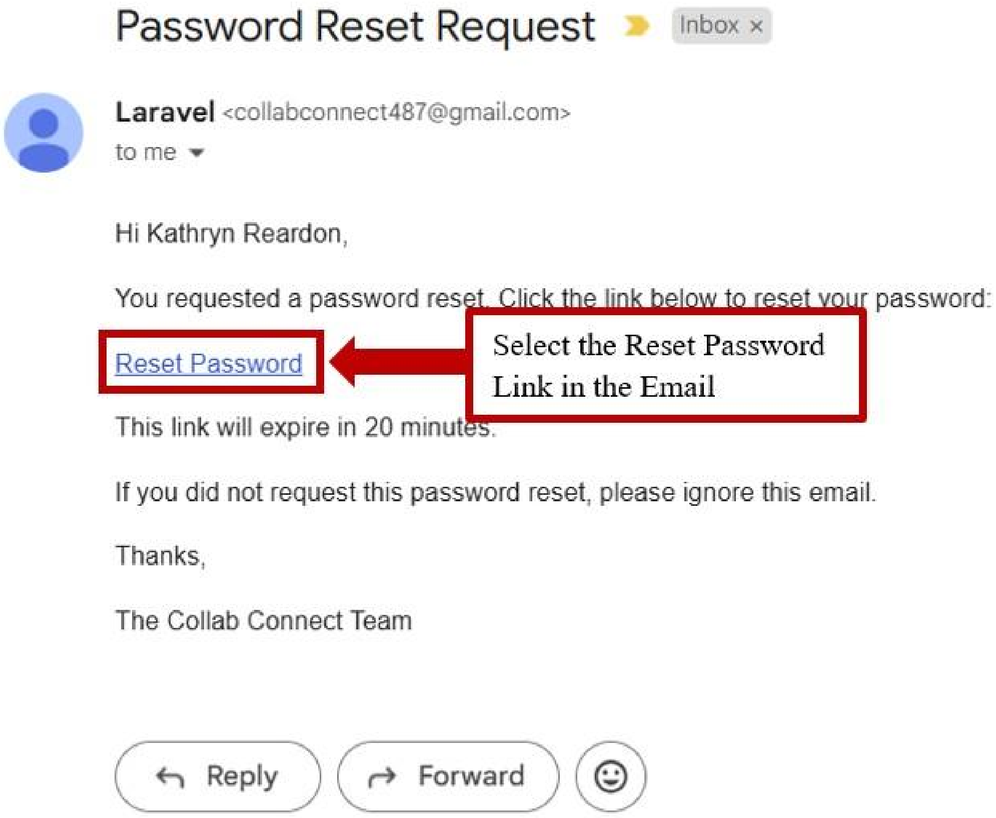


Email: kitkat.reardon@gmail.com

We have emailed your password reset link.

Step 2: Go to the email that was input, and open the Password Reset Request email from Laravel.

Select the Reset Password link to change your password. This link only lasts for 20 minutes.



Password Reset Request

Laravel <collabconnect487@gmail.com>

to me

Hi Kathryn Reardon,

You requested a password reset. Click the link below to reset your password:

Reset Password

This link will expire in 20 minutes.

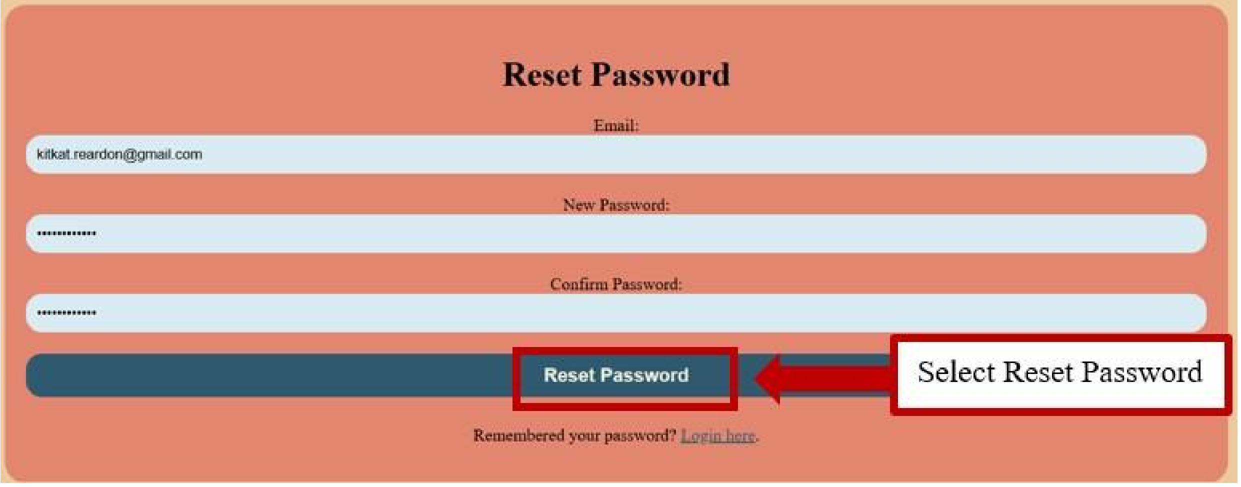
If you did not request this password reset, please ignore this email.

Thanks,

The Collab Connect Team

Select the Reset Password Link in the Email

Step 3: Type in your email, the password, and confirm the password. Then select Reset Password to update your password.



Reset Password

Email:

kitkat.reardon@gmail.com

New Password:

••••••••••••

Confirm Password:

••••••••••••

Select Reset Password

Remembered your password? Login here.

VI.I.V ACCESSIBILITY FEATURES

Collab Connect is designed to be accessible and user-friendly for all users, regardless of technical expertise or physical challenges. The platform features a simple layout to enhance readability. Informative error messages guide users through troubleshooting, while step-by-step instructions on each page ensure that even non-technical users can navigate the platform with ease. By prioritizing intuitive design and accessibility, Collab Connect ensures that all users can benefit from its features.

VII. ADMINISTRATION AND MAINTENANCE

VII.I ROUTINE MAINTENANCE

Routine maintenance for the Collab Connect platform ensures the system operates efficiently and continues to meet the needs of its users. Since the platform relies on CSV file uploads for generating groupings and does not store the uploaded files, the focus of maintenance is on ensuring that the results data, user accounts, and the hosting environment remain well-managed.

IV.I.I User Management

Administrators should regularly review user accounts to ensure the platform's security and usability. They should verify that all users have the correct roles (e.g., Teacher or Admin), and promote or demote users as necessary through the Manage Users page. Administrators should delete accounts that are no longer active or necessary. This reduces clutter and maintains security. Also, all users should regularly update their passwords.

IV.I.II Managing Results Data

Since CSV files uploaded for grouping are not stored on the platform, only the generated group results are saved. At the end of an academic year or when groupings are no longer relevant, users should delete outdated results via the Files Page. This ensures storage efficiency and reduces confusion when working with new data.

IV.I.III Maintaining Datasets

The platform relies on the structure of CSV files exported from PowerSchool. If the PowerSchool format changes, adjustments may need to be made to the platform's algorithms to ensure compatibility with the updated file structure. Administrators should periodically review updates or notifications from PowerSchool regarding changes to their CSV export formats and collaborate with developers to modify the algorithm to process the updated file structure correctly. Conduct thorough testing after adjustments to ensure data is handled accurately.

IV.I.IV Software and Security Updates

The platform relies on several software components that must be kept up-to-date to ensure optimal performance. Use composer update to ensure the latest versions of Laravel and its backend dependencies are installed. Update frontend assets using npm update. Keep the PHP version (minimum 8.2), MariaDB, Apache or Nginx, and all required PHP extensions up-to-date to maintain compatibility with Laravel.

IV.I.V Server and Storage Monitoring

Administrators should regularly check the hosting server's health and performance. They should remind teachers to delete unnecessary files to free up space. They need to also check server logs for errors or slow response times and address any issues promptly. Ensure the server has adequate resources (RAM, CPU, and disk space) to handle user activity.

IV.I.VI Regular Testing

Administrators should perform routine tests to ensure the platform's features are working correctly. They need to test the upload and grouping process regularly to ensure it handles CSV files accurately and generates expected results. They need to also verify that the administrative features, such as role changes ad user deletion function as intended. As SIS systems or file templates evolve, administrators should confirm that the platform can process the new formats without errors.

IV.I.VII User Training and Support

Regular training and support for teachers and administrators ensure the platform is used effectively. The user manual with the latest features should be updated regularly. Administrators should conduct training sessions, especially at the start of a new academic year, to familiarize users with platform updates and new tools, and collect user feedback to address issues or suggestions to improve the platform's usability.

VII.II NEW HOSTING SERVICE

Migrating the Collab Connect platform to a new hosting service involves several steps to ensure a seamless transition without data loss or functionality issues. The process begins with verifying the requirements of the new hosting environment. The new hosting service must support PHP version 8.2 or higher, MariaDB version 10.3 or higher, and either Apache or Nginx as a web server. Additionally, the server should have SSH access to facilitate setup and support for essential PHP extensions such as ctype, cURL, DOM, fileinfo, mbstring, PDO, session, tokenizer, XML, and bcmath.

Once the environment is verified, the platform files and database need to be transferred. Start by exporting the existing database using the `mysqldump` command, which creates a backup file of the group results and user data. This file is then imported into the new server's database using the `mysql` command. For the application files, transfer all relevant project directories, including the `storage` and `bootstrap/cache` folders, using a secure copy protocol (SCP) or an FTP client. These files must be placed in the appropriate directory on the new server.

Next, configure the application for the new environment by updating the `.env` file. Modify the database connection settings to reflect the new server's database credentials, including the host, username, password, and database name. Additionally, update the `APP_URL` setting to the new domain or IP address where the application will be hosted. For example, if the new domain is `http://new-domain.com`, ensure `APP_URL` reflects this. It is also crucial to set appropriate permissions for Laravel's `storage` and `bootstrap/cache` directories using the `chmod` and `chown` commands to ensure the application can write logs and cache files.

Once the configuration is complete, install the necessary dependencies. Run `composer install` to restore backend dependencies and `npm install` to set up frontend assets. Compile the frontend assets with the `npm run build` command to ensure all styles and scripts are production-ready. Afterward, restart the web server to apply the changes and test the application to ensure all features, such as file uploads, grouping algorithms, and administrative tools, work as expected. Pay particular attention to the grouping workflow to verify data accuracy and functionality.

Finally, ensure the new hosting environment is secure and optimized. Schedule regular backups of the database and files, and monitor server performance to address any bottlenecks or issues promptly. By following these steps, the Collab Connect platform can be smoothly migrated to a new hosting service with minimal disruption to its users.

VIII. FUTURE ENHANCEMENTS

VIII.I ENHANCEMENTS FOR TEACHERS

VIII.I.I Automating Behavior File Population

Teachers would no longer need to manually copy student names into the behavior file or upload it separately. The platform can be enhanced to automatically populate the behavior file template using data from the uploaded assessment file. This prefilled behavior file would include student names and other relevant details, requiring teachers to simply review and edit behavior data directly within the platform. This automation would save time and reduce errors during the grouping process.

VIII.I.II AI Integration for Group Optimization

AI algorithms could help teachers optimize groupings by analyzing historical grouping data, student behavior trends, and performance metrics. Teachers would receive AI-driven suggestions for the most effective groups based on a mix of compatibility factors like academic scores, behavior, and social dynamics. This feature would provide tailored recommendations to support classroom goals.

VIII.I.III Multiple Grouping Algorithms

Teachers would have the flexibility to select from various grouping algorithms tailored to their instructional goals:

- **Homogeneous Grouping**: Students with similar skills or traits are grouped together for targeted learning.
- **Heterogeneous Grouping**: Balanced groups with diverse skills and behaviors encourage peer learning.
- **Interest-Based Grouping**: Groups formed based on shared interests or project preferences.
- **Random Grouping**: Neutral, randomly assigned groups for unbiased collaboration.

- **Testing Arrangements:** A special algorithm would generate seating arrangements for classroom testing. The system would consider factors such as academic performance or behavior to ensure that students who might collaborate or distract each other are placed apart.

VIII.I.IV Customizable Compatibility Formula

Teachers would be able to adjust the compatibility formula used to generate groupings, giving weight to factors such as assessment scores, behavioral traits, or specific teacher-defined preferences. This flexibility ensures that groupings align with the unique needs of each class.

VIII.I.V Customizable File Sorting

Teachers would be able to create folders to store groupings for specific classes or subjects as well as filter previous groupings based on the date the file was generated, the subject or class the groups were generated for, the unit the groupings were for, and other related characteristics. This would allow teachers to quickly filter and find specific past groupings.

VIII.I.VI Specified Assessment Based Grouping

Teachers would be able to select which assessment data is used to group students. This would allow teachers to group students based on similar standards or units to help target students' needs for the unit they are currently focused on learning.

VIII.II ENHANCEMENTS FOR ADMINISTRATORS

VIII.II.I Automated Notifications and Maintenance Tools

Administrators would have access to automated tools to ensure platform maintenance such as regular password update notifications, data cleanup reminders to clear out old groupings, and bulk deletion of groupings to simplify data cleanup for new academic years.

VIII.II.II Automated Data Import

Administrators could integrate the platform directly with student information systems (SIS) such as PowerSchool. This enhancement would allow assessment data to sync automatically, eliminating the need for manual CSV uploads. Automated data import ensures accuracy and saves time for both teachers and administrators.

VIII.II.III School District and Multi-Tenant Support

The platform could be extended to support multiple schools or districts under a single instance. Each school would have isolated data, with its own administrators managing users, groupings, and results. At the district level, an overarching admin role could oversee all schools, providing insights and managing configurations across the system. This structure would support scalability and appeal to larger educational organizations.

VIII.II.IV Data Export and Import in Multiple Formats

Administrators could manage data in various formats such as CSV, Excel, JSON, or PDF. This feature would allow for easy sharing of group results, user data, and reports across different systems and stakeholders. It would also facilitate seamless data migration or integration with other tools.

VIII.II.V Detailed Usage Analytics

A detailed analytics dashboard would allow administrators to monitor platform usage and performance. Key insights could include:

- The number of groupings generated by teachers.
- Active and inactive users.
- Trends in grouping preferences or algorithms.
- System performance and storage usage. These analytics would help administrators optimize platform use and identify areas for improvement.

VIII.II.VI AI Integration for Group Optimization

AI algorithms could also assist administrators by providing data-driven insights into overall platform usage and suggesting strategies to improve grouping outcomes across the organization. Administrators could use AI to review patterns in group generation and recommend platformwide optimizations.

VIII.II.VII Enhanced Security Features

Administrators would benefit from advanced security tools, such as two-factor authentication for all users, security audits, and role-based access controls. These features would ensure data safety and compliance with organizational policies.

VIII.III SHARED ENHANCEMENTS FOR TEACHERS AND ADMINISTRATORS

VIII.III.I Customizable Reporting Features

Both teachers and administrators could use customizable reports tailored to their specific needs. Teachers might focus on group-specific insights, while administrators could review system-wide data, such as group success metrics or overall engagement.

VIII.III.II Bulk Actions for Data Management

The ability to delete all groupings or archive results in bulk would be useful for both teachers and administrators. This feature ensures that the platform remains organized and ready for the new academic year.

VIII.III.III Enhanced Accessibility

Improving platform accessibility through screen reader compatibility, better keyboard navigation, and multi-language support would benefit all users, making the system more inclusive.

APPENDIX F. FINAL REPORT

Optimized Student Grouping for Enhanced Classroom Performance

University of Mississippi | CSCI 354: Senior Project | Fall 2024

Kathryn Reardon | kreardo1@go.olemiss.edu

Dr. Yixin Chen (Sponsor) | ychen@cs.olemiss.edu

Abstract

Traditional methods for grouping students in a classroom are time-intensive, subjective, and often result in inconsistent group dynamics. This project addresses these challenges by employing a data-driven approach to optimize student groups based on behavior, attendance, academic performance, and teacher preferences. The minimum viable product is a web application that groups students based on student data and teacher input and stores generated groupings for later reference. By automating the grouping process, this resource will save teachers time, enabling them to focus more on developing meaningful instruction.

The application provides functionality for the users: teachers and administrators. The teachers can create optimized student groups and access past grouping, while administrators have additional capabilities to manage users and view all groupings created in the system. The deliverables for this project include a fully functional web application, source code, and a report detailing the algorithm's design, implementation, and evaluation framework.

Final Report

Developing this project was both a challenging and rewarding journey that pushed me to integrate my technical skills, teaching experience, and problem-solving abilities. The project aimed to create a web application that enables teachers to group students based on assessment and behavior data, while also providing user management features for administrators. Reflecting on the entire process, I can confidently say the project's goals were successfully achieved, despite a few hurdles along the way.

Project Goals and Realization

Initially, I envisioned an application that streamlines the student grouping process that grouped based on assessment and behavior data, allowing teachers to focus more on instruction rather than administrative tasks. These goals were successfully achieved. Teachers can efficiently group students, review past groupings, and input preferences, while administrators can manage user roles, delete accounts, and review all groupings. These functionalities met the practical needs of users, which became clearer as the project developed. The grouping generator had significant changes, transitioning from the stable marriage algorithm to a constraint-based greedy algorithm, to address the need for larger group sizes. I chose to revise the algorithm to mimic the manual grouping process familiar to teachers. I tested the algorithm by using sample test files that I used to manually group and run through the algorithm to compare the results and time spent. This web application reduced the time required for grouping after the initial setup of behavior files from the sample tests. While the best method would be to test with an actual classroom, due to permission and time frame, this was not possible, so I tested using test files. I made a future evaluation plan to test the algorithm in an actual class setting for later work. While the original project prospectus included broad and vague admin functionalities, the development process helped me to clarify and refine these features into practical tools aligned with user needs.

Successes and Challenges

Several aspects of the project were successful and contributed to its overall effectiveness. The algorithm development phase was particularly smooth, as it mirrored the logical process of manual student grouping, which I was already familiar with as a teacher. Using together and apart preferences as well as teacher preferences seemed to make well-constructed groups when comparing them to manual groupings. This surprised me on how well the group results were, but upon further reflection, I feel like having the teacher's preferences allows for grouping considerations that an algorithm could not consider on its on like social dynamics. Backend development was prioritized early in the project, which allowed me time to develop skills in frontend programming and ensured a smoother overall workflow. I decided to keep the database design simple, with only two tables, to minimize complexity and allow me to effectively manage the data despite limited experience. Deploying the project using Laravel on Turing was also successful, as its compatibility with Apache servers streamlined the process with minimal issues.

Challenges arose primarily due to an initial choice of programming language. Python, while effective for certain data tasks, presented significant obstacles in connecting the frontend to the backend. I considered using Flask, but this had a steep learning curve and posed challenges when putting the project onto Turing. Switching to PHP and Laravel addressed this issue as Laravel handles all the frontend to backend connection for me, but I had to rewrite the algorithm, delaying progress by a couple weeks. Similarly, I had to switch from SQLite to MySQL. This was a slight challenge as the default database supported by Laravel is SQLite, but it does support MySQL. I had to look through a lot of the Laravel documentation to determine how to setup a MySQL. Transitioning the database from SQLite to MySQL added setup work, although it did not disrupt functionality. Despite these challenges, the project progressed on schedule, largely due to effective planning and adaptability. My interactions with the project sponsor were constructive, and early discussions clarified that real-world classroom evaluations were beyond the project timeline. This understanding helped define the project's scope and allowed me to focus on creating an evaluation plan for future use.

Opportunities for Improvement

If I were to repeat this project, I would prioritize selecting the most suitable programming language during the planning phase. Choosing Python initially, without fully considering its limitations for deployment and frontend-backend integration, introduced unnecessary complexity and delays. PHP and Laravel ultimately provided a more cohesive development framework. Additionally, I would refine the algorithm development timeline. While focusing on the algorithm early on was valuable, it delayed progress in integrating the frontend and database. A more balanced approach would have allowed for iterative refinement later, improving workflow and enabling better documentation. I would also explore algorithmic adjustments, such as addressing "together" and "apart" preferences simultaneously, to reduce the risk of ungrouped students. These changes would streamline the process, enhance the final product, and reduce stress during development.

Lessons Learned

This project provided a comprehensive learning experience that exceeded my expectations. Technically, I learned how to use PHP, MySQL, and Laravel, and developed the ability to integrate frontend, backend, and database components into a functional web application. I enhanced my skills in algorithm design, learning to iteratively refine the logic to meet project requirements. Debugging errors systematically became a key strength, allowing me to resolve issues efficiently. Beyond technical skills, this project emphasized the importance of adaptability. Switching programming languages mid-project, while challenging, demonstrated my ability to pivot without abandoning progress. Moreover, this project highlighted the interdisciplinary nature of my studies in mathematics, education, and computer science, illustrating how these fields can intersect to solve complex problems.

Future Enhancements

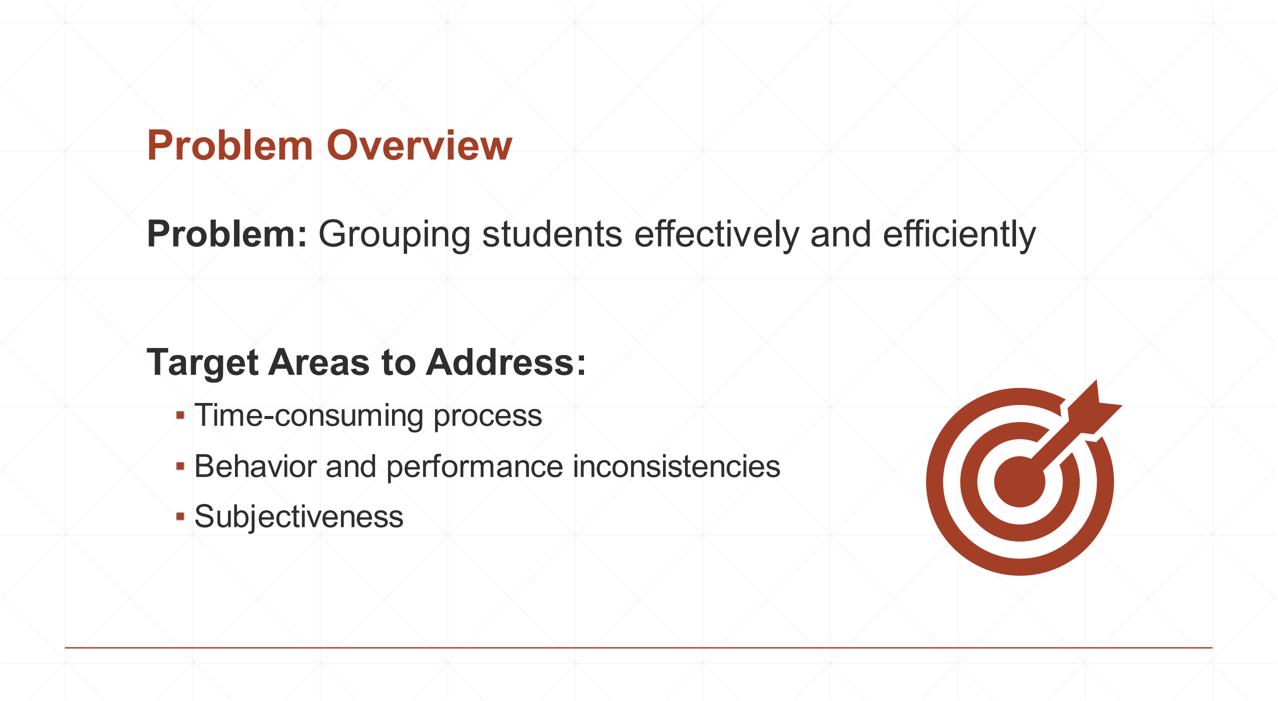
Several extensions and enhancements could add value to this project. Automating the population of behavior files from assessment data would streamline setup and reduce repetitive tasks for teachers. Expanding the grouping options to include heterogeneous or homogeneous algorithms would offer flexibility for different classroom needs. Integrating the platform with external systems, such as PowerSchool, could simplify data import and improve usability. Additionally, implementing a detailed evaluation plan to measure the application’s effectiveness in real-world classrooms is a priority. This plan would involve collecting baseline data on manual grouping practices, comparing it with results from using the application, and analyzing metrics such as engagement, collaboration, and academic performance. Teacher and student feedback would provide valuable insights for refining the platform. Finally, scaling the platform for broader adoption in schools would ensure a greater impact on classroom dynamics and instructional efficiency.

APPENDIX G. FINAL ORAL PRESENTATION SLIDES

Optimized Student Grouping for Enhanced Classroom Performance

Presenter: Kathryn Reardon

Sponsor: Dr. Yixin Chen

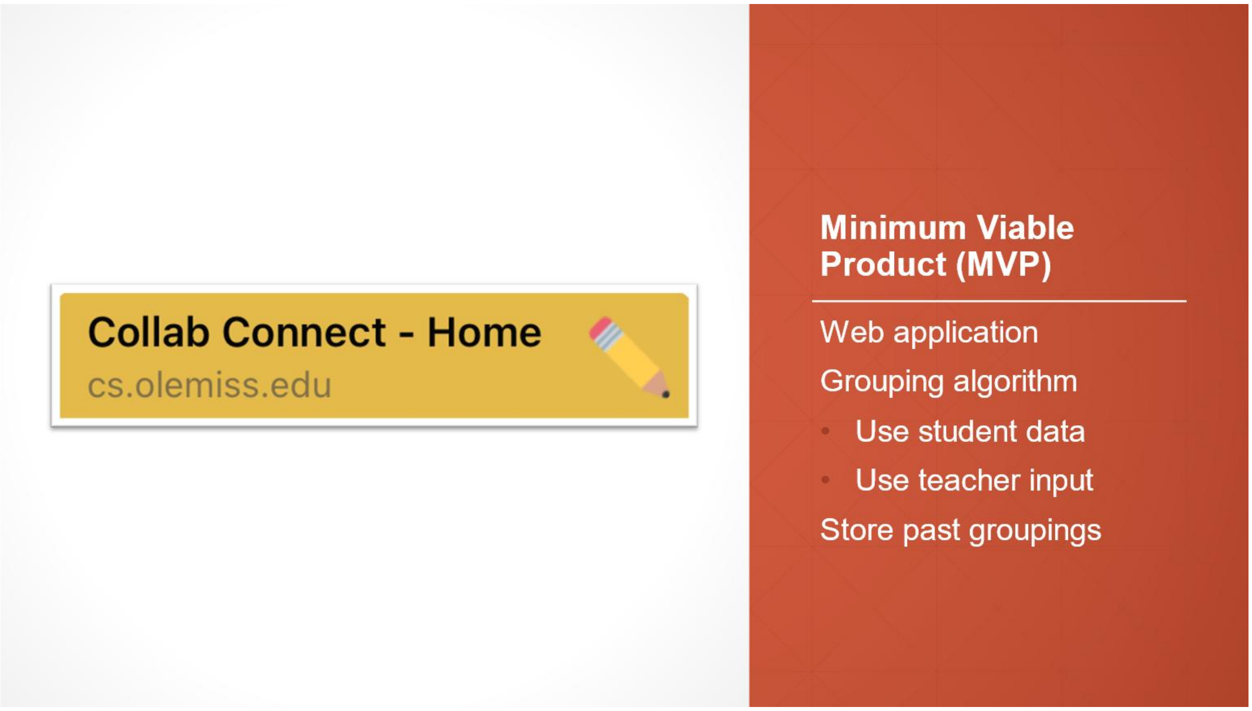


Problem Overview

Problem: Grouping students effectively and efficiently

Target Areas to Address:

- Time-consuming process
- Behavior and performance inconsistencies
- Subjectiveness



Collab Connect - Home
cs.olemiss.edu

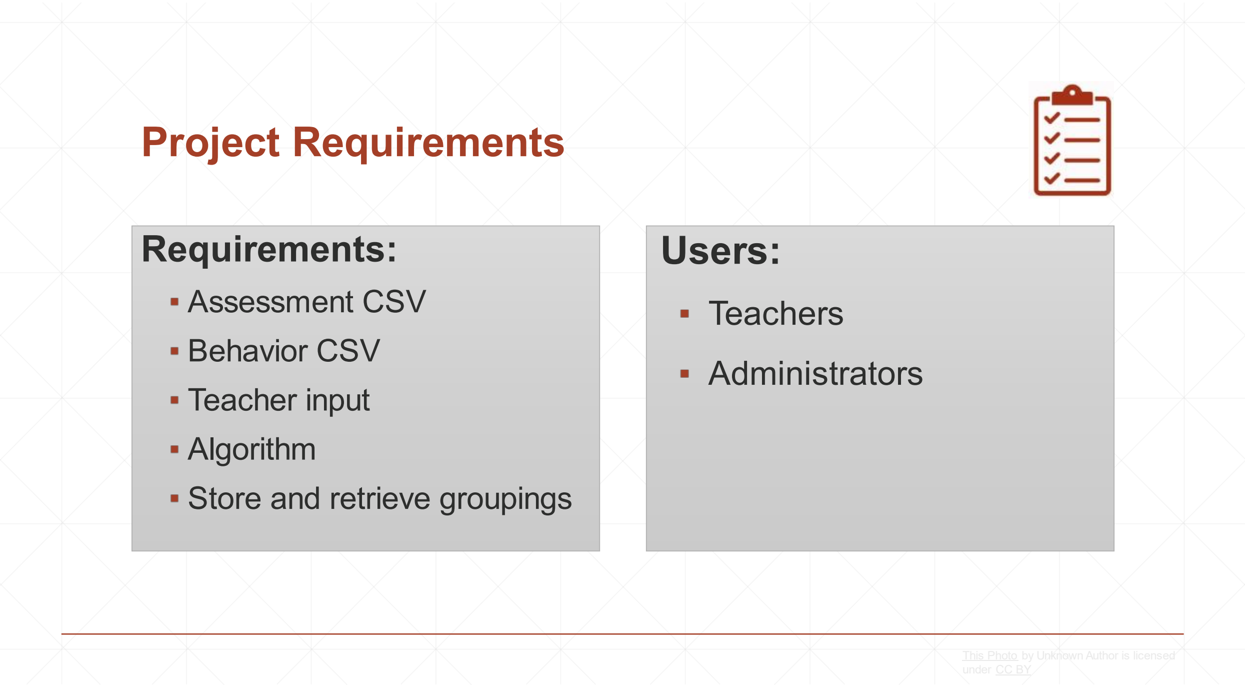
Minimum Viable Product (MVP)

Web application

Grouping algorithm

- Use student data
- Use teacher input

Store past groupings



Project Requirements

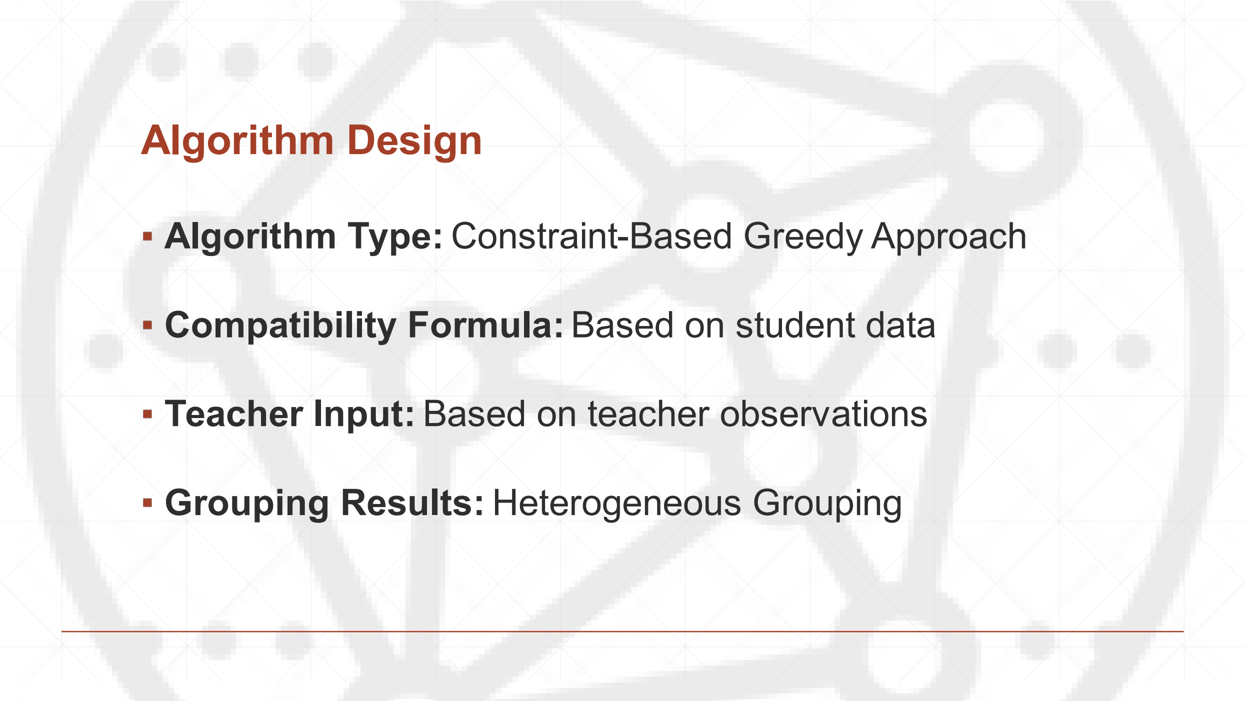
Requirements:

- Assessment CSV
- Behavior CSV
- Teacher input
- Algorithm
- Store and retrieve groupings

Users:

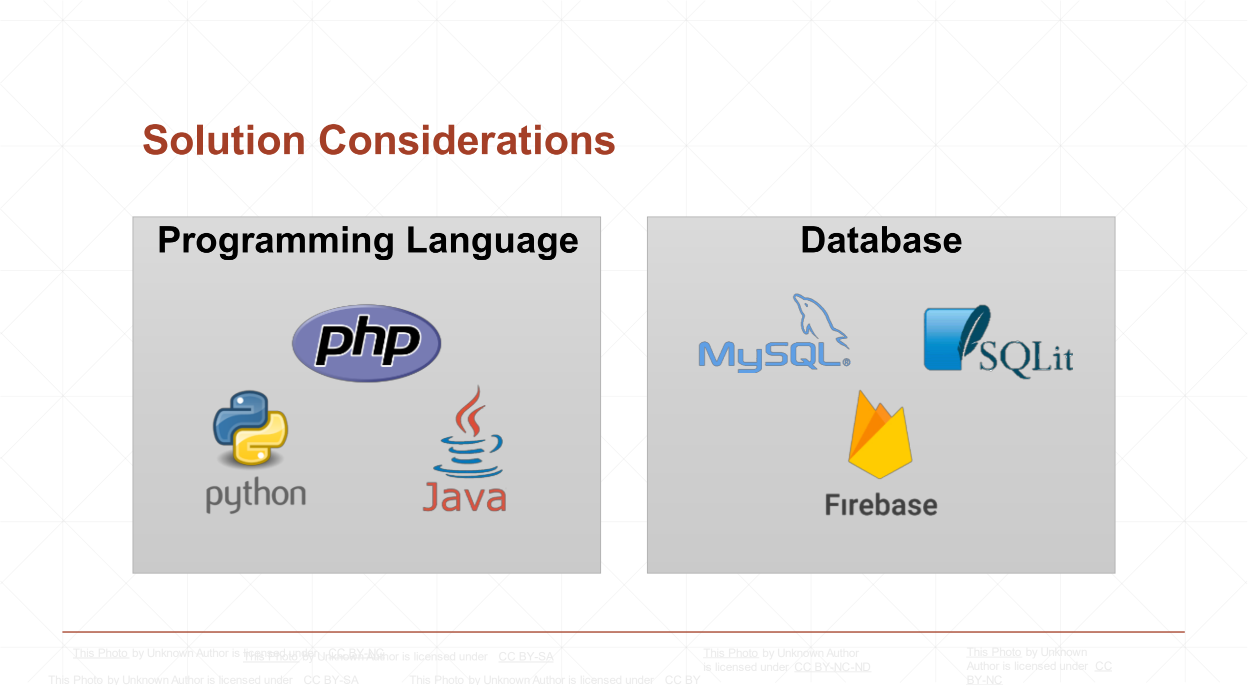
- Teachers
- Administrators

This Photo by Unknown Author is licensed under CC BY



Algorithm Design

- **Algorithm Type:** Constraint-Based Greedy Approach
- **Compatibility Formula:** Based on student data
- **Teacher Input:** Based on teacher observations
- **Grouping Results:** Heterogeneous Grouping



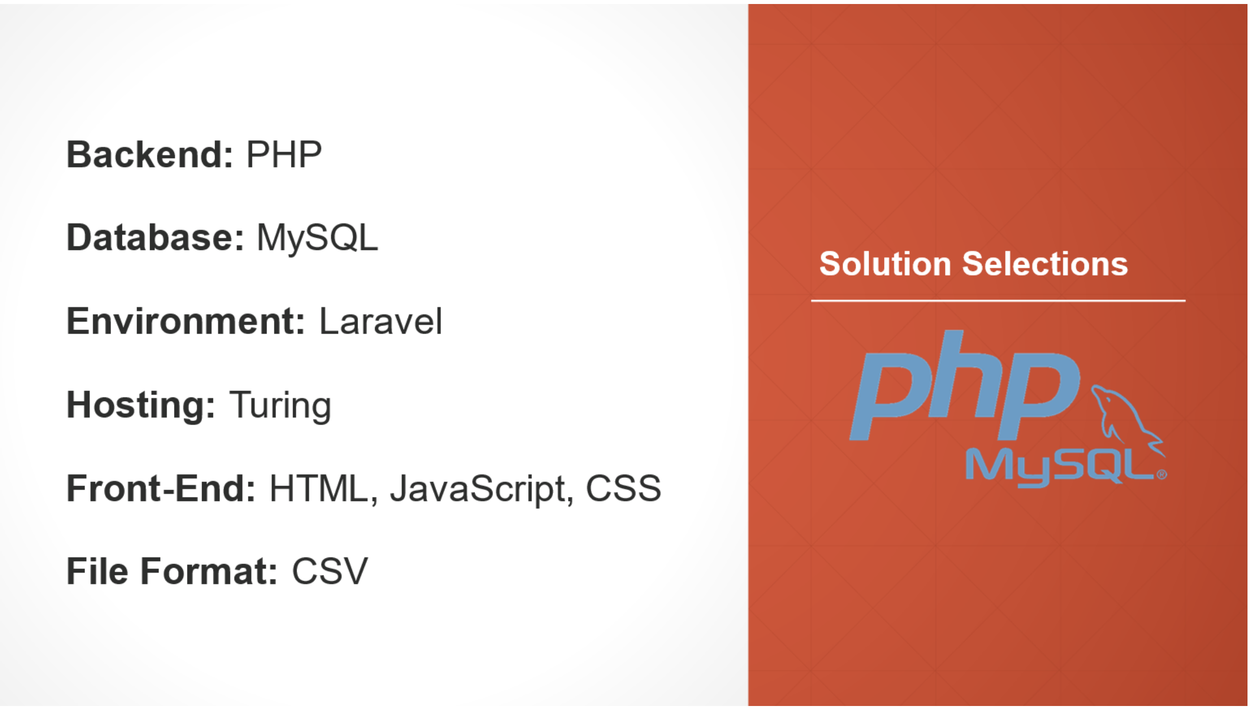
Solution Considerations

Programming Language

php
python
Java

Database

MySQL
SQLit
Firebase



Backend: PHP

Database: MySQL

Environment: Laravel

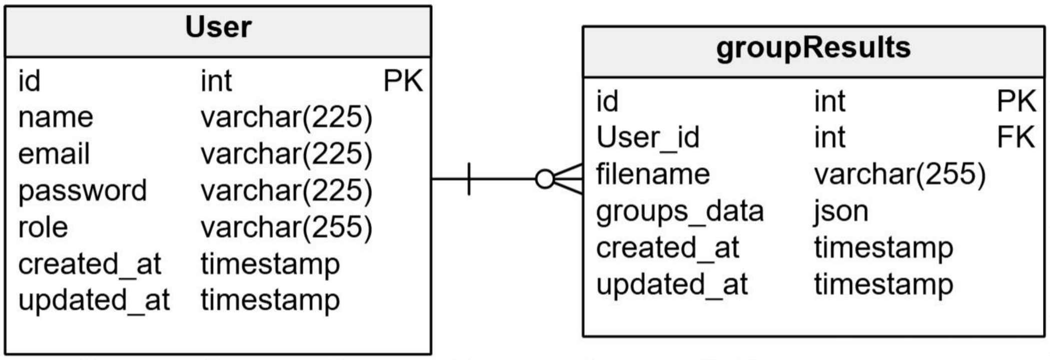
Hosting: Turing

Front-End: HTML, JavaScript, CSS

File Format: CSV

Solution Selections

ERD



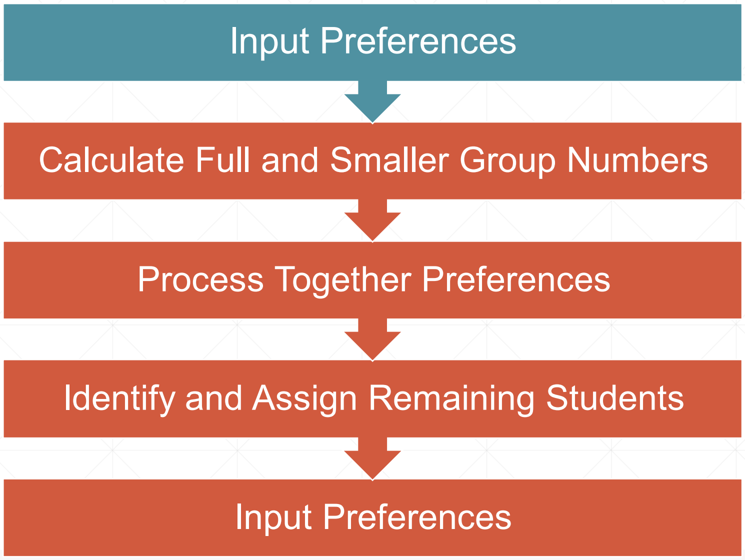
User

id	int	PK
name	varchar(225)	
email	varchar(225)	
password	varchar(225)	
role	varchar(255)	
created_at	timestamp	
updated_at	timestamp	

groupResults

id	int	PK
User_id	int	FK
filename	varchar(255)	
groups_data	json	
created_at	timestamp	
updated_at	timestamp	

Algorithm Logic



Input Preferences

Calculate Full and Smaller Group Numbers

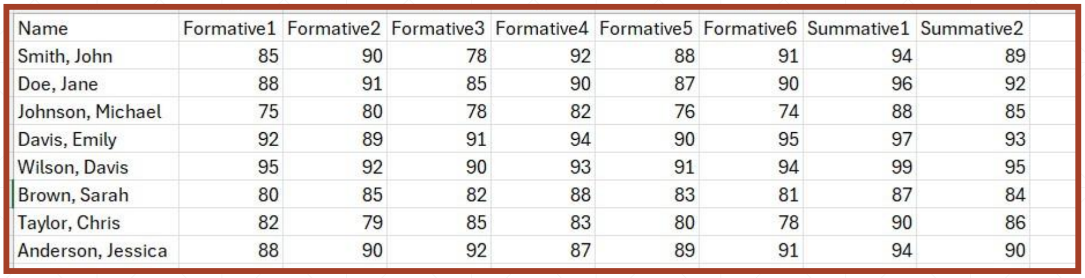
Process Together Preferences

Identify and Assign Remaining Students

Input Preferences

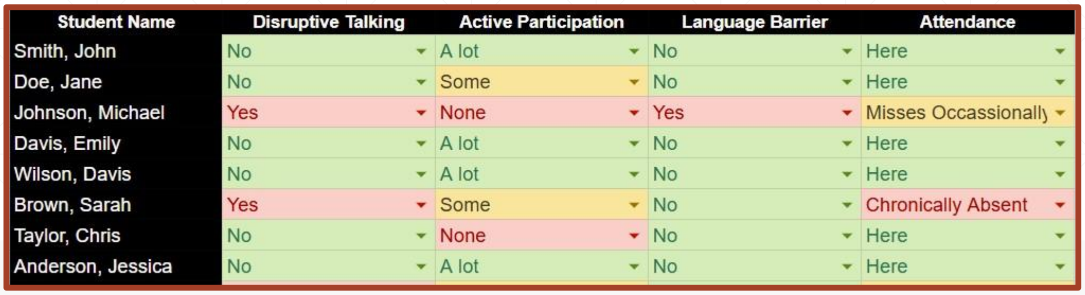
CSV Upload Files

Assessment

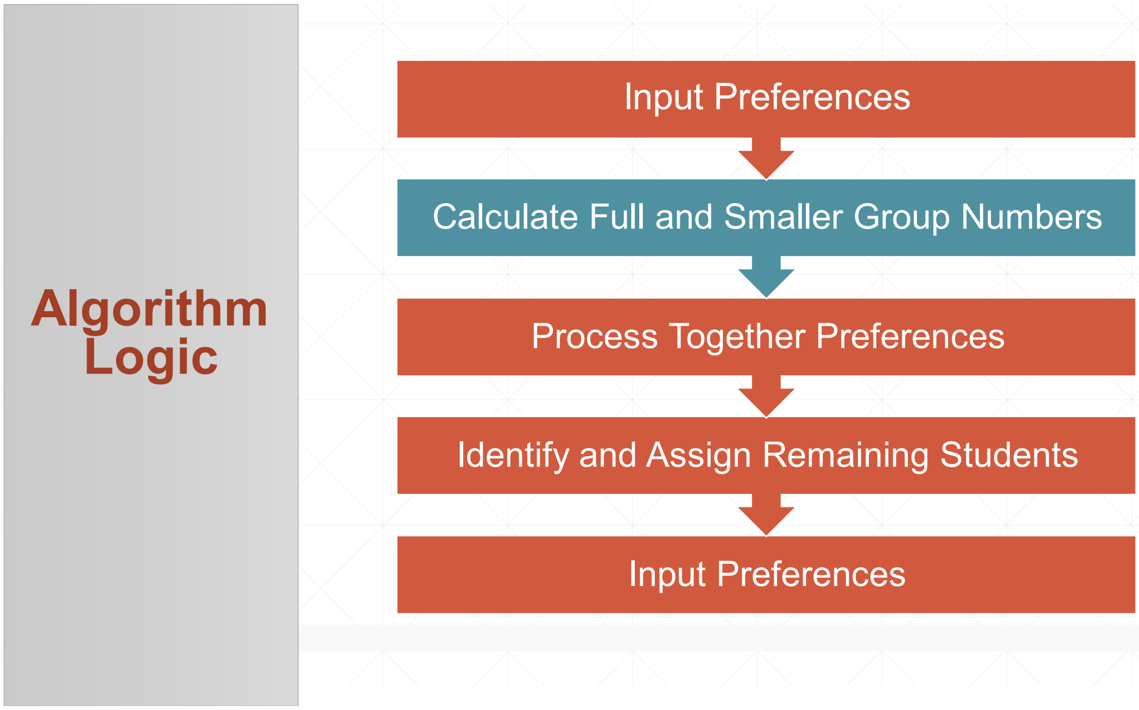


Name	Formative1	Formative2	Formative3	Formative4	Formative5	Formative6	Summative1	Summative2
Smith, John	85	90	78	92	88	91	94	89
Doe, Jane	88	91	85	90	87	90	96	92
Johnson, Michael	75	80	78	82	76	74	88	85
Davis, Emily	92	89	91	94	90	95	97	93
Wilson, Davis	95	92	90	93	91	94	99	95
Brown, Sarah	80	85	82	88	83	81	87	84
Taylor, Chris	82	79	85	83	80	78	90	86
Anderson, Jessica	88	90	92	87	89	91	94	90

Behavior



Student Name	Disruptive Talking	Active Participation	Language Barrier	Attendance
Smith, John	No	A lot	No	Here
Doe, Jane	No	Some	No	Here
Johnson, Michael	Yes	None	Yes	Misses Occassionally
Davis, Emily	No	A lot	No	Here
Wilson, Davis	No	A lot	No	Here
Brown, Sarah	Yes	Some	No	Chronically Absent
Taylor, Chris	No	None	No	Here
Anderson, Jessica	No	A lot	No	Here



Algorithm Logic

Input Preferences

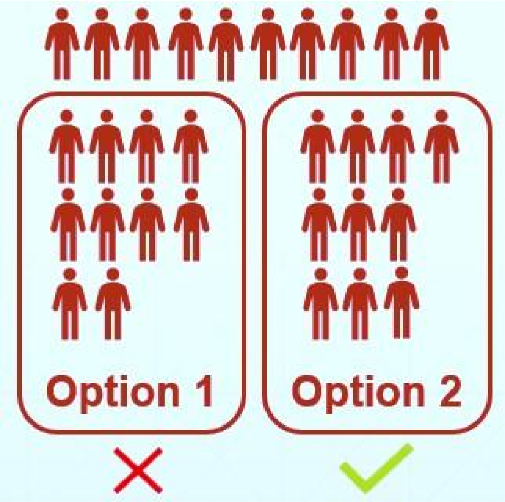
Calculate Full and Smaller Group Numbers

Process Together Preferences

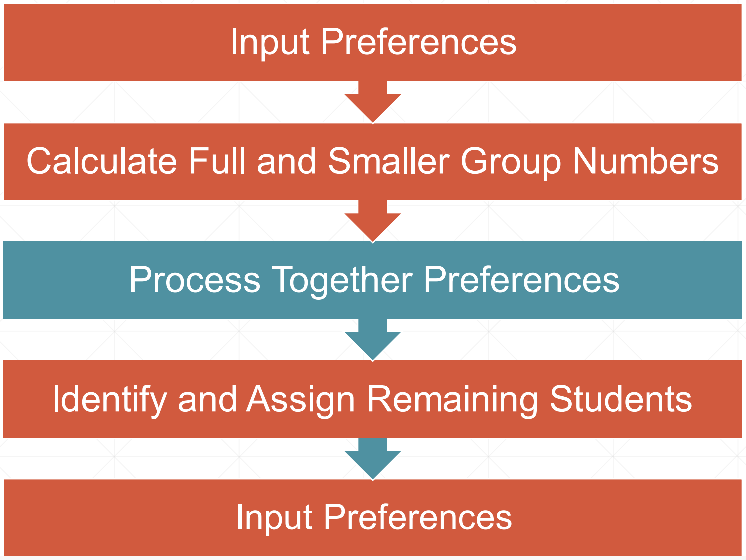
Identify and Assign Remaining Students

Input Preferences

Determine Group Sizing



Algorithm Logic

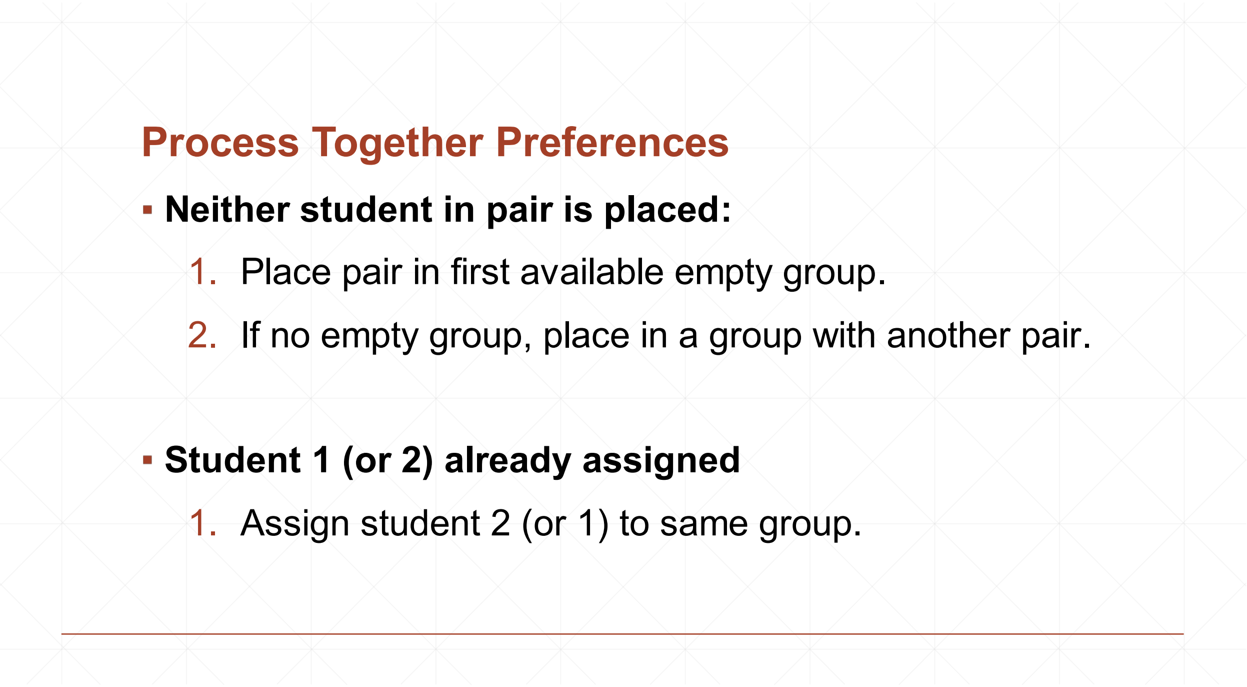

Input Preferences

Calculate Full and Smaller Group Numbers

Process Together Preferences

Identify and Assign Remaining Students

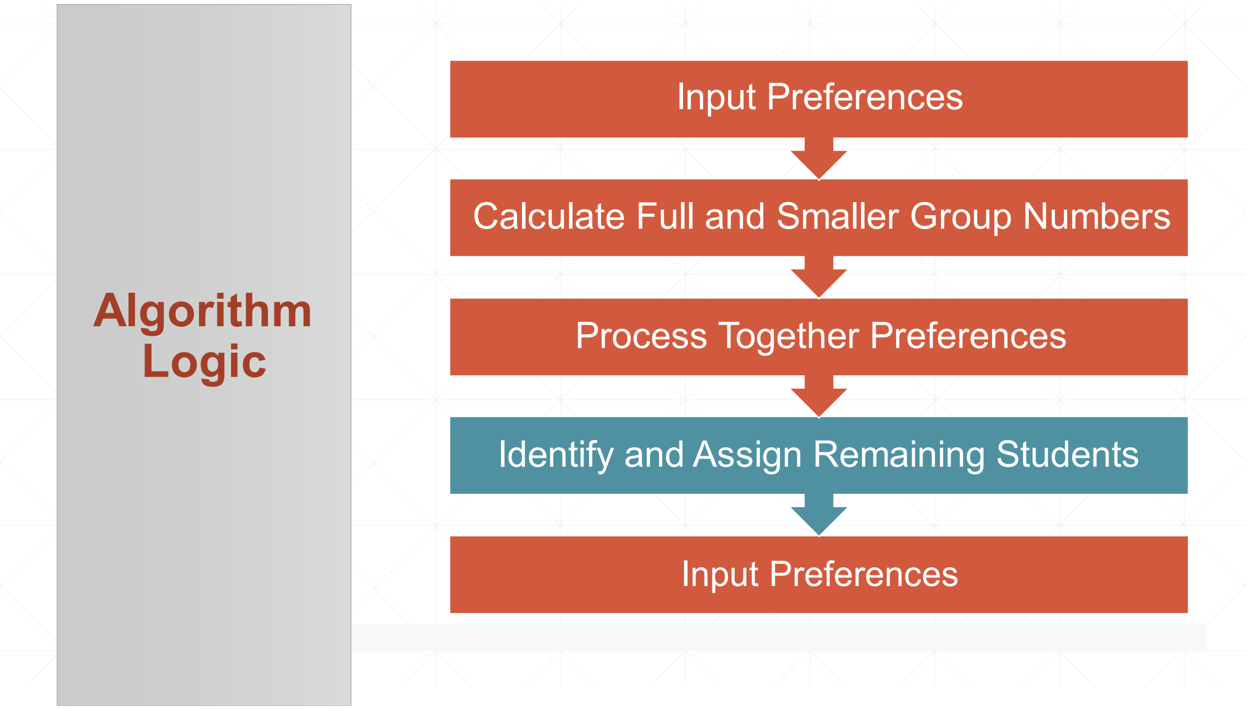
Input Preferences



Process Together Preferences

- **Neither student in pair is placed:**
  1. Place pair in first available empty group.
  2. If no empty group, place in a group with another pair.

- **Student 1 (or 2) already assigned**
  1. Assign student 2 (or 1) to same group.



Algorithm Logic

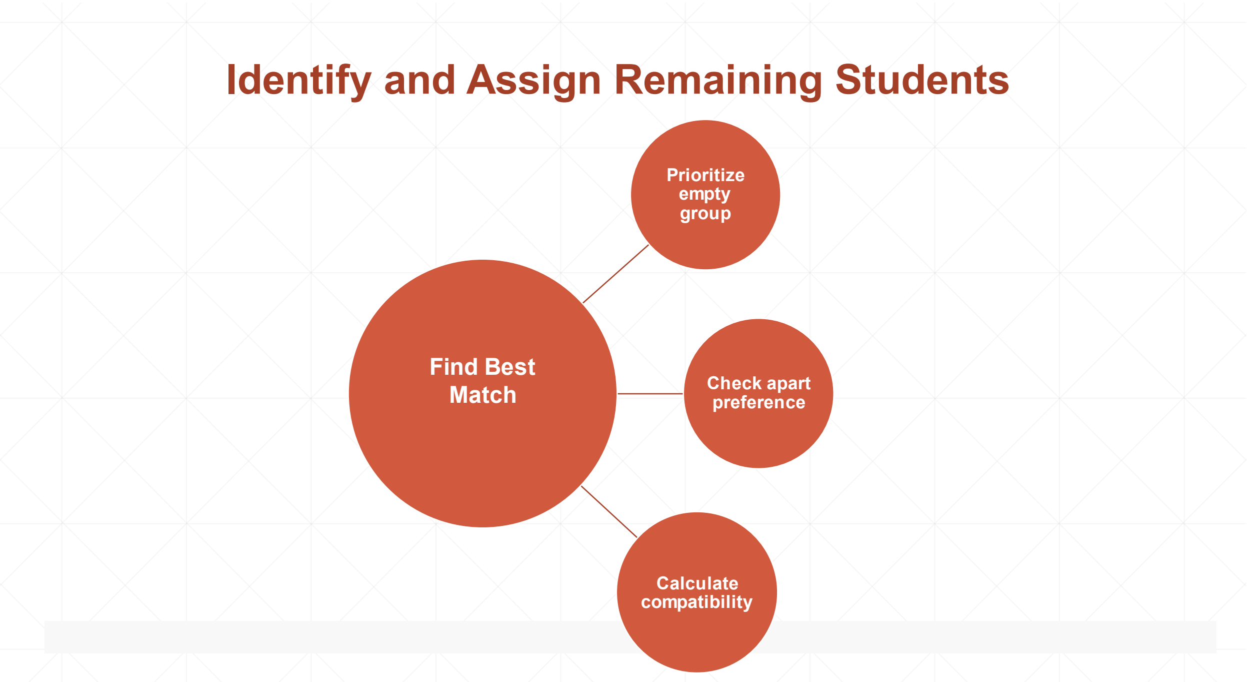
Input Preferences

Calculate Full and Smaller Group Numbers

Process Together Preferences

Identify and Assign Remaining Students

Input Preferences



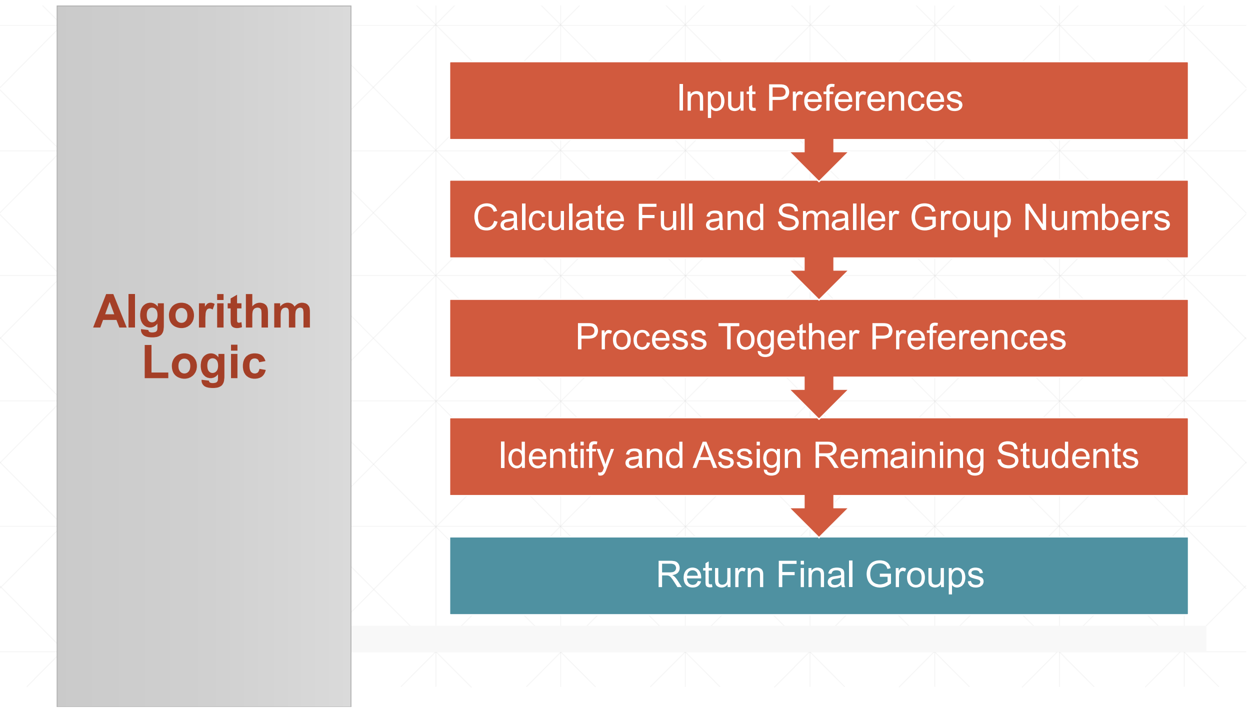
Identify and Assign Remaining Students

Find Best Match

Prioritize empty group

Check apart preference

Calculate compatibility



Algorithm Logic

Input Preferences

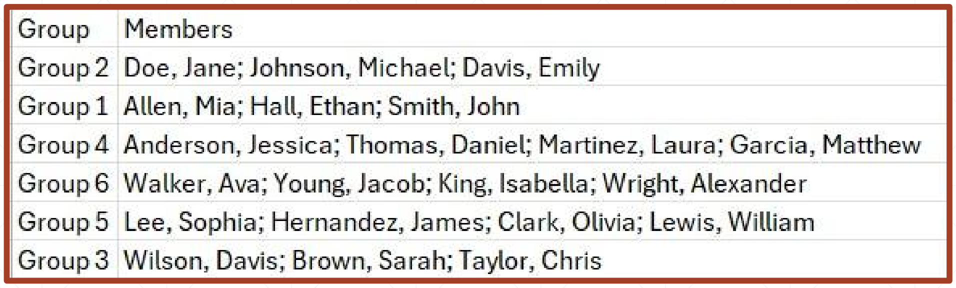
Calculate Full and Smaller Group Numbers

Process Together Preferences

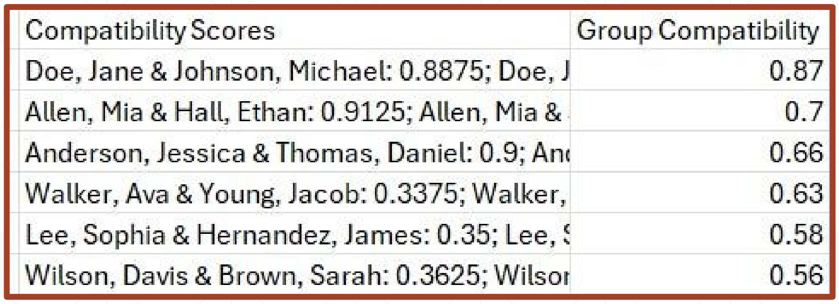
Identify and Assign Remaining Students

Return Final Groups

Grouping Results File



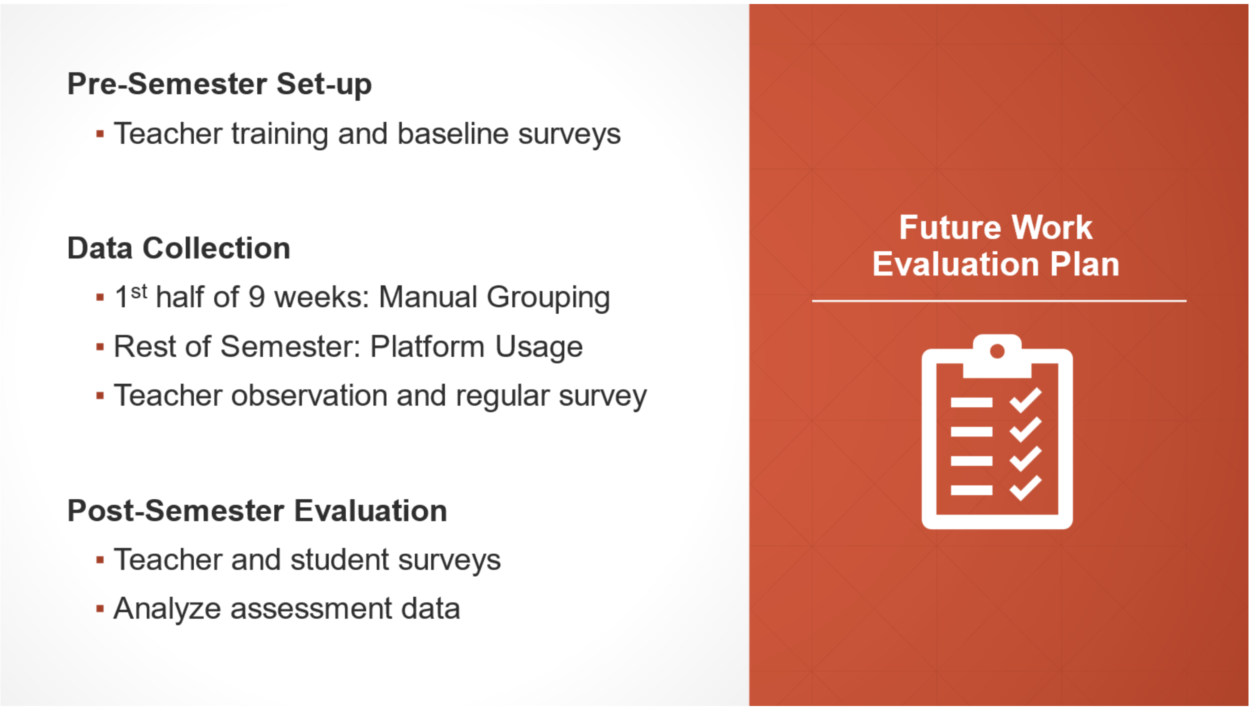
Group	Members
Group 2	Doe, Jane; Johnson, Michael; Davis, Emily
Group 1	Allen, Mia; Hall, Ethan; Smith, John
Group 4	Anderson, Jessica; Thomas, Daniel; Martinez, Laura; Garcia, Matthew
Group 6	Walker, Ava; Young, Jacob; King, Isabella; Wright, Alexander
Group 5	Lee, Sophia; Hernandez, James; Clark, Olivia; Lewis, William
Group 3	Wilson, Davis; Brown, Sarah; Taylor, Chris



Compatibility Scores	Group Compatibility
Doe, Jane & Johnson, Michael: 0.8875; Doe, J	0.87
Allen, Mia & Hall, Ethan: 0.9125; Allen, Mia &	0.7
Anderson, Jessica & Thomas, Daniel: 0.9; And	0.66
Walker, Ava & Young, Jacob: 0.3375; Walker,	0.63
Lee, Sophia & Hernandez, James: 0.35; Lee, S	0.58
Wilson, Davis & Brown, Sarah: 0.3625; Wilso	0.56

Demo

https://turing.cs.olemiss.edu/~kreardo1/seniorProject/collabConnect/public/



Pre-Semester Set-up

- Teacher training and baseline surveys

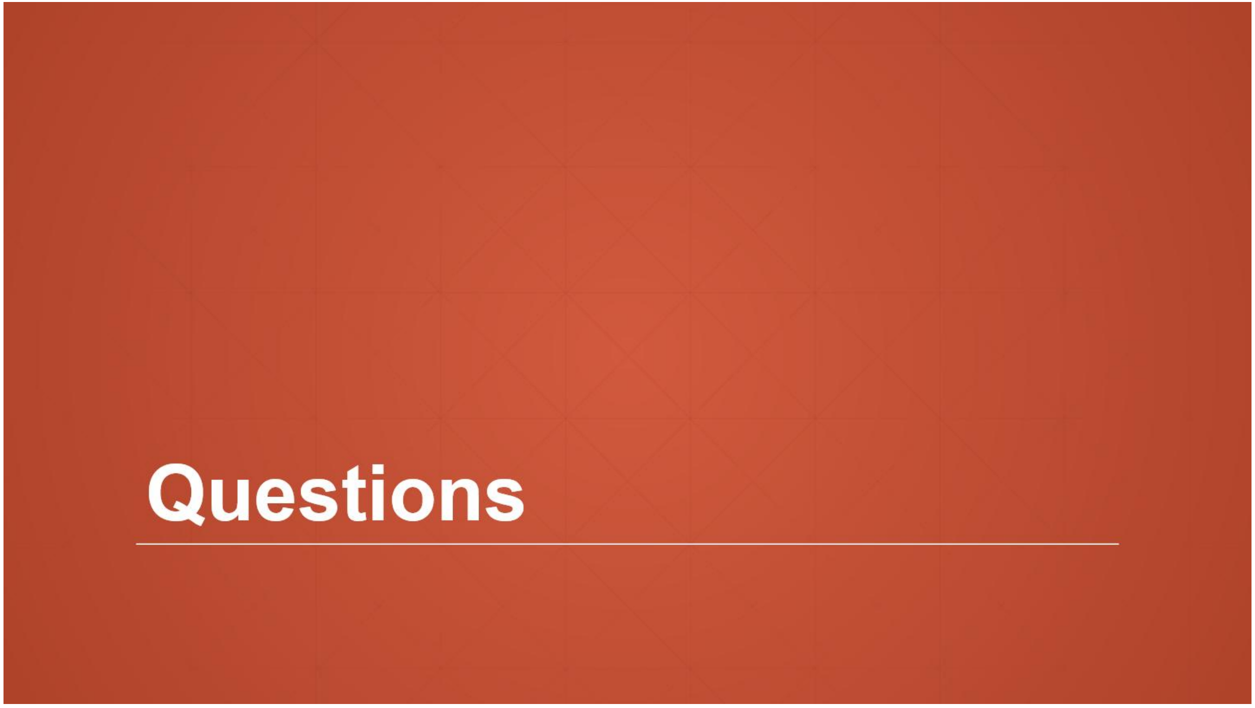
Data Collection

- 1st half of 9 weeks: Manual Grouping
- Rest of Semester: Platform Usage
- Teacher observation and regular survey

Post-Semester Evaluation

- Teacher and student surveys
- Analyze assessment data

Future Work
Evaluation Plan



Questions

APPENDIX H. EVALUATION PLAN

Problem: Traditional student grouping methods are time-consuming and depend on subjective criteria derived from student scores and teacher observation but are inconsistent in producing effective group dynamics. Collab Connect aims to address these challenges by employing a data-driven algorithm to produce optimal student groups based on behavior, attendance, and academic performance.

Target Areas: To evaluate the effectiveness of Collab Connect, there are five main target goals: time-saving for teachers, classroom collaboration, and engagement, academic impact, teacher attitude towards grouping, and student perspectives.

1. **Time Saving:** One goal of the algorithm is to reduce the time teachers spend grouping students so that they can spend more time focusing on the lesson plan and instruction. This will be measured using a time log of time spent grouping students, comparing it with a pre-semester estimate of manual grouping times.
2. **Engagement:** A key goal of the algorithm is to increase classroom engagement and collaboration. This will be evaluated by classroom observations using a structured rubric that focuses on participation, collaboration, task completion, and on-task behavior during groupwork.
3. **Academic Impact:** Assessment scores will be analyzed to determine whether the algorithm leads to measurable improvements in student performance. Scores from pre- and post-semester assessments will be compared, using statistical methods to evaluate significant changes.
4. **Teacher Attitudes:** Teacher perceptions of grouping will be assessed through surveys and interviews. This will include questions on the platform's usability and satisfaction with the grouping results. An open-ended feedback section will allow teachers to share suggestions for improvement.
5. **Student Perspectives:** Student feedback is important to understand the social and academic dynamic of the generated groups. Surveys will be conducted to gather student opinions on group functionality, collaboration, and perceived fairness. Questions will address whether students felt supported by their peers and if the groups helped them stay engaged and achieve learning goals.

Implementation: Evaluating the algorithm will take place over one semester and will be completed in three phases: pre-semester setup, data collection, and post-semester evaluation.

1. **Pre-Semester Set-up:** Teachers will participate in a training session to familiarize them with the Collab Connect platform, including how to upload files, set preferences, and interpret the group results generated. During this training, teachers will have opportunities to test the platform with sample data to address any technical or usability concerns before full implementation. To get a baseline for evaluation, teachers will complete a detailed survey about their current grouping practices. This survey will gather

data on the time they typically spend forming groups manually, the frequency of group-based learning in their classrooms, and their attitudes toward group dynamics. Students will also complete a short survey assessing their experiences with group-based activities, including their perceptions of collaboration and fairness in previous group assignments. These pre-semester surveys will be a reference for evaluating changes after the platform's use.

2. **Data Collection:** During the first half of the first nine weeks, teachers will use manual grouping to determine how the surveys align with classroom practices. For the rest of the semester, teachers will use the Collab Connect platform to form student groups for lessons. Teachers will be encouraged to use the platform for various group related activities, such as collaborative projects, peer teaching activities, and differentiated instruction tasks. This ensures a broad evaluation of the algorithm’s adaptability to different classroom needs. Classroom observations will occur bi-weekly. To gather student feedback, short surveys will be conducted after major group-based activities and after summative or end of unit assessments. These surveys will assess how students felt about their group experiences, including whether they found the groupings fair, collaborative, and supportive of their learning. Teachers will also be asked to complete brief observation forms, sharing their observations on group functionality and student engagement. Regular check-ins with teachers will provide an opportunity to address any technical issues and gather informal feedback. These check-ins will also reinforce the importance of using the platform consistently and help identify any adjustments needed to improve usability.
3. **Post-Semester Evaluation:** The post-semester phase will focus on analyzing collected data and obtaining final feedback from teachers and students. Teachers will complete a detailed survey assessing their overall experience with the platform, including perceived time savings, satisfaction with group results, and the impact on classroom dynamics. They will also report on whether the platform influenced the frequency of group-based learning in their classrooms and whether they observed improvements in student engagement and performance. Students will participate in a final survey evaluating their experiences with group work throughout the semester. This survey will include questions about how well they collaborated with their group members, whether they felt supported, and whether the groups helped them stay engaged and achieve their learning goals. Finally, assessment data will be analyzed to determine changes in student performance. Pre- and post-semester scores will be compared, with a focus on identifying trends related to grouping results.

Rubrics and Surveys:

Pre-semester Teacher Survey

Thank you for participating in the Collab Connect evaluation. Your input will help us understand your current grouping practices and set a baseline for comparison after using the platform. Please answer the following questions based on your experiences with student grouping.

Section 1: Time and Frequency

Approximately how much time do you currently spend forming student groups for each activity?

a) Less than 10 minutes
b) 10–20 minutes
c) 21–30 minutes
d) More than 30 minutes

How often do you group students in your classroom?

a) Never
b) Occasionally (e.g., a few times per semester)
c) Weekly
d) Daily

Section 2: Grouping Effectiveness

On a scale of 1 to 5, how effective do you find your current grouping methods in achieving balanced and collaborative groups? (1 = Not Effective, 5 = Very Effective)

What challenges do you face when creating student groups?

Section 3: Expectations

What outcomes would you like to achieve by using Collab Connect?

Pre-Semester Student Survey

Your feedback is important to understand your experiences with group work in the classroom. Please answer the following questions honestly.

Section 1: Group Experiences

How well do you usually work with classmates in group settings?

a) Not well at all
b) Somewhat well
c) Very well

Do you feel that group assignments in the past have been fair?

a) Not fair at all
b) Somewhat fair
c) Very fair

Do you feel supported by your group members during group work?

a) Not supported at all
b) Somewhat supported
c) Very supported

Does working in groups help you understand the material better?

a) Not at all
b) Somewhat
c) Very much

Section 2: Improvements

What could make group work better for you?

Classroom Observation Form

Observer Name: _____________ **Date: ________________**
Group Activity Observed: ____________________ **Class/Subject: ________________**

Section 1: Engagement

Rate each statement on a scale of 1 (Strongly Disagree) to 5 (Strongly Agree):

1. Students were actively participating in the group activity.
   □ 1 □ 2 □ 3 □ 4 □ 5
2. Students were focused on the task for the majority of the activity.
   □ 1 □ 2 □ 3 □ 4 □ 5
3. Group members encouraged one another to stay engaged.
   □ 1 □ 2 □ 3 □ 4 □ 5
4. Students demonstrated enthusiasm or interest in the activity.
   □ 1 □ 2 □ 3 □ 4 □ 5

Section 2: Collaboration

Rate each statement on a scale of 1 (Strongly Disagree) to 5 (Strongly Agree):

1. Group members communicated effectively and respectfully.
   □ 1 □ 2 □ 3 □ 4 □ 5
2. Students actively listened to each other’s ideas.
   □ 1 □ 2 □ 3 □ 4 □ 5

3. Students resolved disagreements constructively.
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
4. All group members contributed meaningfully to the task.
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Section 3: Task Completion

Rate each statement on a scale of 1 (Strongly Disagree) to 5 (Strongly Agree):

1. The group successfully completed the assigned task.
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
2. The group demonstrated understanding of the task's objectives.
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
3. The quality of the group’s work met or exceeded expectations.
   ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Section 4: Behavioral Dynamics

Check all behaviors observed during the activity:

☐ Students were inclusive of all members.

☐ Dominant behavior by one or more students negatively impacted the group.

☐ Students sought help or clarification when needed.

☐ Conflict, if any, was managed effectively by the group.

☐ Group members encouraged one another’s contributions.

Section 5: Observer Notes

Strengths observed during the activity:

Challenges faced by the group:

Specific examples of group dynamics or behaviors (positive or negative):

Suggestions for improvement:

Section 6: Overall Effectiveness

Rate the overall success of this group activity on a scale of 1 (Not Successful) to 5 (Highly Successful):
□ 1 □ 2 □ 3 □ 4 □ 5

Post-Activity Student Feedback Form

We'd like to hear about your experience with your group. Your feedback will help us improve how groups are formed in the future.

Section 1: Group Functionality

How well did your group work together during this activity?

a) Not well at all
b) Somewhat well
c) Very well

Did working in this group help you stay focused on the task?

a) Not at all
b) Somewhat
c) Very much

Did you feel the group assignment was fair?

a) Not fair at all
b) Somewhat fair
c) Very fair

Did working in this group help you understand the material better?

a) Not at all
b) Somewhat
c) Very much

Section 2: Open Feedback

What did you like about your group during this activity?

What could have been improved about your group?

Post-Semester Teacher Survey

We would like to hear your experience with Collab Connect. The feedback will help us improve the platform.

Section 1: Time and Frequency

How much time did you save using Collab Connect compared to your previous grouping methods?

a) None
b) Less than 10 minutes per session
c) 10–20 minutes per session
d) More than 20 minutes per session

Did Collab Connect encourage you to group students more frequently?

a) Less frequently than before
b) About the same as before
c) More frequently than before

Section 2: Effectiveness and Satisfaction

On a scale of 1 to 5, how effective were the groups generated by Collab Connect? (1 = Not Effective, 5 = Very Effective)

Did you observe an improvement in student engagement and collaboration during group activities? (1 = No Improvement, 5 = Greatly Improved)

How easy was it to use Collab Connect? (1 = Very Difficult, 5 = Very Easy)

Section 3: Feedback

How did using Collab Connect affect your ability to focus on lesson planning?

What changes would you recommend for Collab Connect?

Post-Semester Student Survey

We want to learn about your experiences with group work this semester. Please share your thoughts about the groups you were part of.

Section 1: Group Experiences

How well did your groups work together this semester?

a) Not well at all
b) Somewhat well
c) Very well

Did your group members support you during group work?
a) Not at all
b) Somewhat
c) Very much

Did working in groups help you stay engaged in classroom activities?
a) Not at all
b) Somewhat
c) Very much

Did working in groups help you learn and understand the material better?
a) Not at all
b) Somewhat
c) Very much

Do you think the groups assigned through Collab Connect were fair?
a) Not fair at all
b) Somewhat fair
c) Very fair

Section 2: Open Feedback

What did you like about the groups you were part of?

What could have made your group experiences better?

Success Criteria:

- **Time Savings:** Teachers report a 50% reduction in time spent forming groups.
- **Engagement:** Classroom engagement report improvement from at least 70% of teachers.
- **Academic Outcomes:** Statistically significant improvement in student assessment scores from pre- to post-semester.
- **Teacher Satisfaction:** At least 80% of teachers rate the platform as effective or highly effective.
- **Student Feedback:** At least 75% of students report positive experiences with the groupings.

APPENDIX I. COLLAB CONNECT

To access the product, visit the following website link while on The University of Mississippi campus: CollabConnect Website.

APPENDIX J. SAMPLE FILES

To run the grouping algorithm on the website, use the following test files:

- AssessmentData.csv and BehaviorData.csv (version 1).
- AssessmentData2.csv and BehaviorData2.csv (version 2).