

studyBuddie

Names: Jason Broucek, Josh Churkey, Atsadang Deebunmee Na Chumphae, Aryan Pagar, Abhishek Sachi, Zachary White

Project description:

StudyBuddie is a student focused productivity web app designed to make studying more social, organized, and personalized. The platform helps students manage their academic responsibilities by allowing them to create assignments, track progress, and view everything within an integrated calendar. Users can add friends which allows them to form study groups, helping students implement and schedule collaboration despite busy schedules. When a study group is created, its meeting time automatically appears on each member's personal calendar, ensuring everyone is kept up to date and prepared. StudyBuddie enhances organization by giving students a clear visual layout of upcoming tasks while also making it quick and simple to add assignments. By combining productivity tools with social connection, the app creates an engaging environment where students can stay motivated, maintain structure, and work more efficiently. Overall, StudyBuddie aims to transform studying into a collaborative and streamlined experience that supports academic success.

Project Tracker - GitHub project board:

<https://github.com/orgs/CU-CSCI3308-Fall2025/projects/32/views/1?filterQuery=>

The GitHub Project Board displays three columns of tasks:

- Todo:** 7 items
 - Create Calendar Event to Add Task
 - Delete Calendar Event to Remove Item
 - Navigation bar redirects to correct pages based on requests given by user
 - Create Specific Parameters for each task
- In Progress:** 0 items
- Done:** 6 items
 - Register Page UI
 - Implement Login Form UI
 - Login validation
 - Implement Register Form UI

Video:

youtube.com/watch?v=u-Ej-0bHmWc&feature=youtu.be

VCS:

<https://github.com/CU-CSCI3308-Fall2025/group-project-Abhishrek05>

Contributions:

Jason Broucek:

I mainly worked on the front end for our web design application, although I did help with some back end data and routing for other functions. Some of the main contributions I had were for the login page where I implemented a fully interactive front end page with css styling. I also helped work on some of the calendar pages functions and the overall layout/format. Another main function I helped work on was the styling and interaction for the study groups page.

Josh:

I added database foundations for users, assignments, and assignment-friends, plus supporting SQL logic that ensured data was properly stored in the database. I built the front-end friends page using Handlebars that lists friends and allows you to search for and add new ones, as well as the calendar modal that adapts options based on event type. I also implemented Express /friends backend routes in index.js, delivering search, add, and remove workflows that properly manipulate the database and friends page UI.

Atsa:

So for this project, I mainly handled the backend. I assisted in the design of the tables. After we discussed with the whole team about what our application is supposed to do, the back team and I were able to finalize what information needed to be collected and we created the tables. After that, I handled the assignment aspects. I created the functions to actually add the assignments to the database and also created a function to remove assignments for the database. I also assisted in querying the tables to display the appropriate assignments for each user. Then also I was trying to create the functions to update a specific assignment in the table.

Aryan:

I primarily worked on the front end of the project, building the calendar page, register page, and homepage, all using Handlebars. I also worked on the interactive features on these pages, for example, adding functionality to the calendar page that creates an event when you click anywhere. I also built a navbar that's visible on all pages using Handlebars, making every page clickable to each other. Additionally, I worked on all the functions on each page on the front end, making elements clickable with JS. I also worked on querying the assignments table, ensuring it was filtered by account.

Abhi:

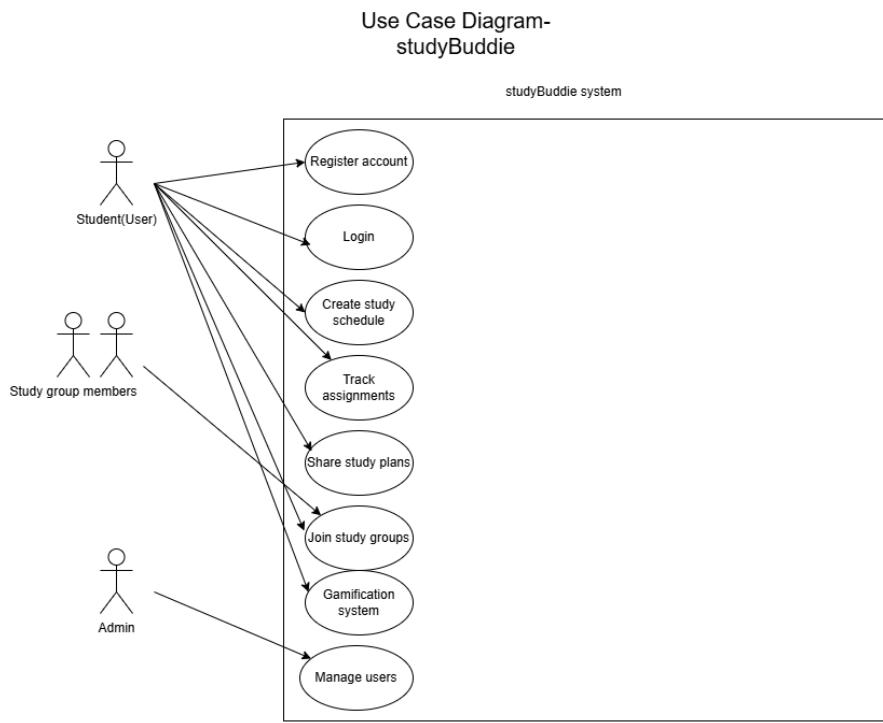
My main focus was the project's backend, mainly working on the study group page and syncing it with the calendar. Using get, post, put, delete, and more routes, I effectively managed the information that needed to be displayed, query the correct data, and also created the SQL tables that will store the information. The Study Group aspect was the main aspect of our app, being able to create groups, edit and delete them, and collaborate with other users. Making sure users

could only see the group they were added to, the groups being created, and that only the host being able to edit and delete groups is crucial. Then syncing it with the calendar, so students are able to clearly see when and what time the group was made, was extremely important. This was extremely important in creating our application and was a great way to learn how all these factors interact with each other and how full-stack development is.

Zachary:

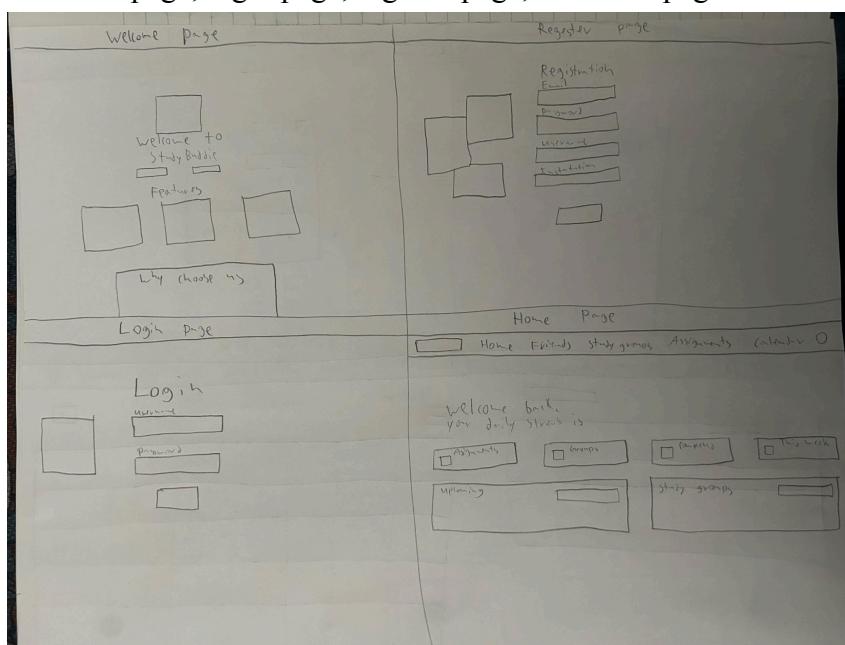
I worked mainly on the back end. I made routes and functions to handle the data for certain aspects. Some of the items I worked on were, login route, add assignments function, and the edit assignments function. Also working with other team members, I got to see and help with the SQL tables. I also helped with debugging/testing, that was another main part after integrating everything together. This also gave me a chance to look at the HTML and see how the front end was developing it, and how the JS can be used.

Use Case Diagram:

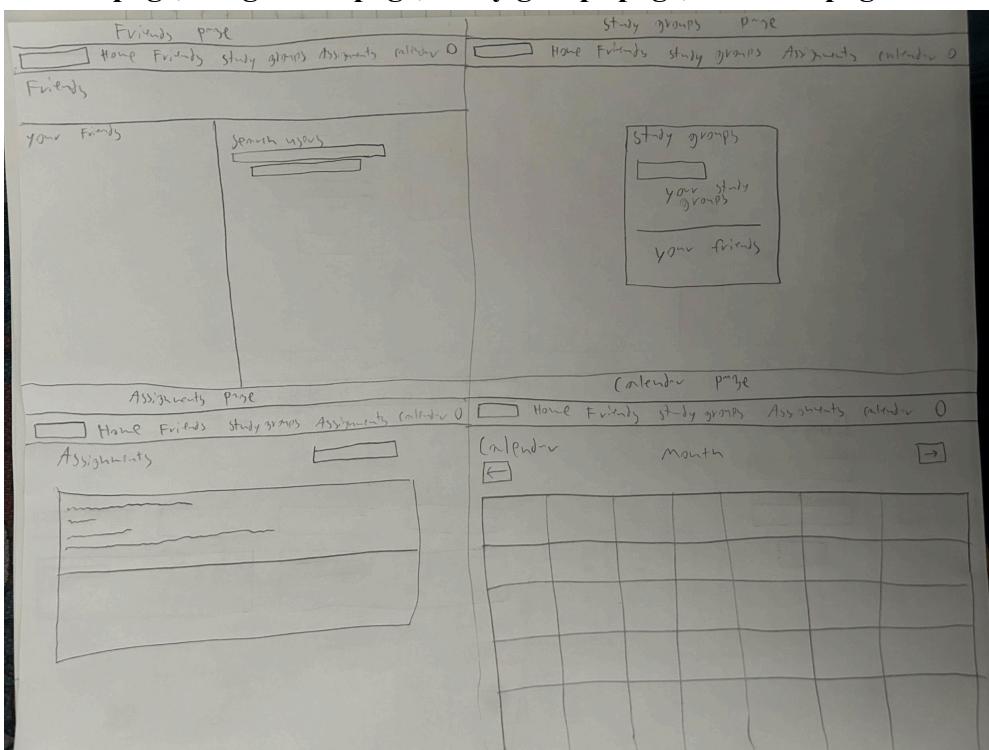


Wireframes:

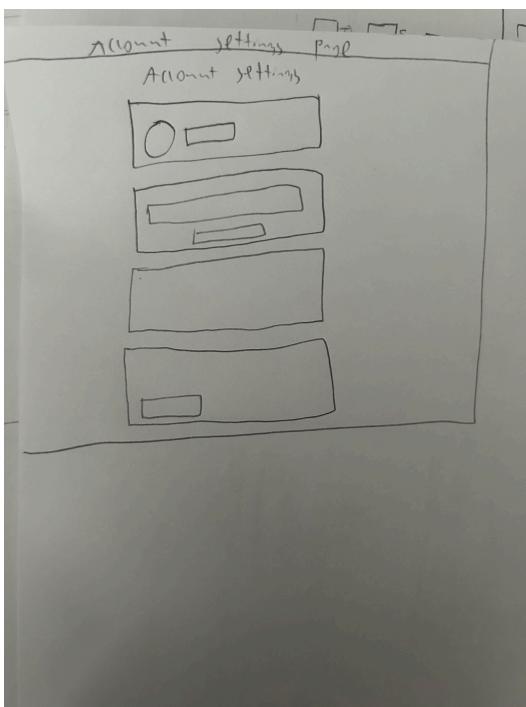
Welcome page, login page, register page, and homepage:



Friends page, assignment page, study groups page, calendar page:



Account page:



Test results:

Feature 1: Assignment Creation & Progress Tracking

Description:

Users should be able to create assignments, set due dates, and track progress through the dashboard.

Test cases:

1, Create a new assignment:

Objective: verify that users are able to create a new assignment.

Steps: go to the assignment page, click on add assignment, in the pop up menu add the assignments name, class, due date, and description.

Test data:

Name: Calc 2 hw #5

Class: Math 2300

Due date: 12/3/2025

Priority: High(since its due soon)

Description: Chapter 5 review problems

Expected results:

Assignment shows up within assignment tab

Assignment shows up on the calendar at the correct date

Observations:

What users did: Users immediately clicked on “Add Assignment” and filled in all fields.

User reasoning: They assumed all fields were required because of the typical structure of assignment forms.

Consistency: Yes

Deviations: One user initially expected the calendar to update without refreshing.

Resulting changes: Implemented calendar refresh after assignment creation.

2, Delete an Assignment:

Objective: Confirm that users can delete an assignment and that it is removed from all sections.

Steps: Go to the calendar page, select an assignment, click delete, confirm that it is removed from the calendar and assignments page.

Test data:

Existing assignment such as the one listed above.

Expected results:

The assignment is removed from the calendar

The assignment is removed from the assignments page

Observations:

What users did: Users clicked the assignment on the calendar.

User reasoning: They expected clicking an event to open a details

Consistency: Yes

Deviations: One user expected clicking to only allow edit and delete not the whole form

Resulting changes: Added a clearer Delete Assignment button

3, Missing Required Fields:

Objective: Confirm that required fields must be filled before submission.

Steps: Try to submit an assignment with a blank field such as no name listed.

Test data:

Name: ""

Class: Math 2300

Due date: 12/3/2025

Priority: High(since its due soon)

Description: Chapter 5 review problems

Expected results:

Error message displayed

Form does not submit

Observations:

What users did: Some users left a field blank to see what would happen

User reasoning: They were testing if all fields were required

Consistency: Yes

Deviations: Most users just filled in the form completely

Resulting changes: Didn't need to change much, just made sure the flag worked when a section was left blank

Feature 2: Study Group Creation & Calendar Sync

Description:

Users should be able to create study groups, invite friends, and automatically sync group meetings to each member's calendar.

Test cases:

1, Create a Study Group

Objective: Verify that users are able to create a new study group.

steps: Go to study groups page, click create study group, fill out form, add friend to group meeting, save

Test data:

Group name: CS Study Group

Category: Exam prep

Date: 12/3/2025

Max participants: 5

Group meeting link:Zoom

Add friends:Josh, Atsa

Expected results:

Group is created

Populates in group meeting tab and personal calendar

Observations:

What users did: Most users added friends before trying to create a group

User reasoning: They expect group apps to require members

Consistency: Yes

Deviations: Some users attempted to create a group without anyone added

Resulting changes: Added a section that says add friends at the bottom of study group form to help notify people of this

2,Calendar Auto-Sync

Objective: Verify that study group meeting times appear on all members calendars

steps:Person A makes a study group with person B added, person B refreshes page and sees the meeting populated on their calendar

Test data:

Group name: CS Study Group

Category:Exam prep

Date:12/3/2025

Max participants:5

Group meeting link:Zoom

Add friends:Person B

Expected results:

The meeting appears on person B's personal calendar.

Observations:

What users did: Person B refreshed the page to check if they were added

User reasoning: Used to this with other calendar style apps

Consistency: Yes

Deviations: All tests refreshed the page to check

Resulting changes: Kept it how it is because users understood refreshing/switching tabs

3, Invalid Group Creation

Objective: Ensure that a study group cannot be created if there is a blank field in the form.

Steps:fill out form but leave one field blank

Test data:

Group name: CS Study Group

Category:Exam prep

Date:12/3/2025

Max participants:5

Group meeting link:""

Add friends:Josh, Atsa

Expected results:

Error message displayed

Form does not submit

Observations:

What users did: Most users filled out the form completely

User reasoning: They understood the need to fill everything out

Consistency: Yes

Deviations: Some users didn't add a link for the meet

Resulting changes: Added please fill out this field for any field that was left blank when creating

Feature 3: User Authentication for login and register

Description:Users must be able to create an account, log in with valid credentials, and receive proper error feedback for invalid attempts.

Test cases:

1, Register a New User

Objective: Users are able to register a new account

Steps:go to register page, fill in all fields, submit form

Test data:

Email:PersonA@gmail.com

Password:Qwertyu1

Username:PersonA

Institution:CU Boulder

Expected results:

Account is created

User is directed to login page

User data is stored

Observations:

What users did: users filled in all sections

User reasoning: They expected all fields to be required

Consistency: Yes

Deviations: Some users left the institution section blank

Resulting changes:Made it so this section is not required because although we are student focused, non students are also able to use our web app.

2, Login With Correct Credentials

Objective: User is able to login when they input correct account info

Steps: go to login page, fill out sections with correct info, click login to be directed to homepage

Test data:

Username:PersonA

Password:Qwertyu1

Expected results:

User successfully is logged in and taken to the homepage

Observations:

What users did: All users were able to successfully login as a result of filling in all fields

User reasoning: Usernames and passwords are always required to login

Consistency: Yes

Deviations: No deviations

Resulting changes: None made

3>Login With Incorrect Password

Objective: User is not able to login to account when they input incorrect account info

steps: Go to login page, fill out sections with incorrect password, click login and be notified that you filled out a Invalid username or password

Test data:

Username:PersonA

Password:11223344(incorrect password)

Expected results:

User is not logged in and receives “Invalid username or password” message

Observations:

What users did: Most users were able to login the first try with correct credentials

User reasoning: Since they just created the account they remembered the info

Consistency: Yes

Deviations: One user mistyped their password and was shown the Invalid username or password pop up

Resulting changes: none made as this user was able to login correctly the second try

Deployment:

<https://group-project-abhishek05.onrender.com>