

CMPT 276 – Project Final Report

Cirkle – Collaborative Study for Better Results

Group Members:

Julianna Morena

Mudasser Mashal

Yogya Agrawal

Nafeesa Leena

Link to Github Repository:

<https://github.com/CMPT-276-SPRING-2025/final-project-07-hills>

Link to Video:

https://drive.google.com/file/d/1nkxcNTN5n1uol579fc_cvPrIqeKvj9dN/view?usp=sharing

Link to Deployed Website:

final-project-07-hills.vercel.app

1. Meeting User Needs & Solving the Problem

Our project aimed to provide a collaborative study tool that helps students stay organized, motivated, and engaged. To assess its success, we had a peer testing session during class, where students from other groups tried our app and provided feedback. The feedback was positive, with users highlighting the intuitive design and ease of navigation. The Pomodoro timer, group leaderboard, and document editing features were the most appreciated. Some users pointed out minor issues, such as needing clearer instructions for authentication or more visual feedback when actions were completed. These insights confirmed that the app successfully addressed the core user needs of study accountability, time management, and group collaboration.

2. SDLC Model Analysis

We followed an Agile method along with a Kanban system. Task tracking was managed through a Kanban board on GitHub, where tasks were organized into backlog, in-progress, and completed stages. Weekly team meetings were held to align with Agile's iterative approach and to assess priorities as the project progressed. Initially, our plan included a Flashcard feature, but we decided to remove it due to time constraints. This change demonstrated Agile's adaptive planning in action. This iterative, feedback driven workflow enabled us to respond to evolving user needs while maintaining project velocity and managing technical challenges effectively.

3. Features Implemented for Each API

1. Google OAuth API (New Addition)

Features Implemented:

- **Secure User Authentication:** Implemented Google Sign-In functionality allowing users to authenticate using their Google accounts.

- Permission Management: Configured OAuth scopes to request appropriate access permissions for Google Drive integration.
- Sign-out Functionality: Implemented Sign-Out functionality allowing users to easily use different google accounts

Changes Since Planning Stage:

- This API was not part of the original proposal but was added to enhance security and provide a streamlined authentication experience.
- Integration of OAuth eliminated the need for creating a separate authentication system while providing secure access to users' Google resources.

2. Google Drive API (Expanded)

Features Implemented:

- Upload files: Implemented file upload functionality allowing users to add files and study materials directly from their devices. Additionally, the users were able to view the files uploaded by other members in the group.
- Create documents: Allowed the user to create documents that get saved as google docs files in the group, allowing everyone to access and edit them.
- Renaming and deleting files and documents: Added functionality to rename and delete files and documents that you have created inside the group.

Changes Since Planning Stage:

- Originally, Google Docs creation and management was planned as a separate API implementation.
- Consolidated functionality by using Google Drive API for both file uploads and document creation, simplifying the architecture

3. Firebase Database (New Addition)

Features Implemented:

- Group Management System: Stored group and membership information, along with the files and documents associated with each group and information about the creator.
- Group Joining Functionality: Generated unique group IDs to facilitate easy joining of groups via ID sharing

- Pomodoro Timer Leaderboard: Implemented a leaderboard for the pomodoro page and stored the leaderboard points using firebase.

Changes Since Planning Stage:

- Firebase Database was not included in the original proposal but was added to provide essential data storage and real-time functionality
- The addition of Firebase enabled the implementation of leaderboards and group joining elements.

4. CI/CD Pipeline Overview

We used GitHub Actions as our continuous integration tool to automatically run build checks and unit tests on each pull request. This helped identify errors early in the development process and maintained code stability. For deployment, we integrated Vercel with our GitHub repository, enabling automatic deployments to a live environment every time changes were merged into the main branch. This allowed the team to instantly view and verify changes in a production-like environment.

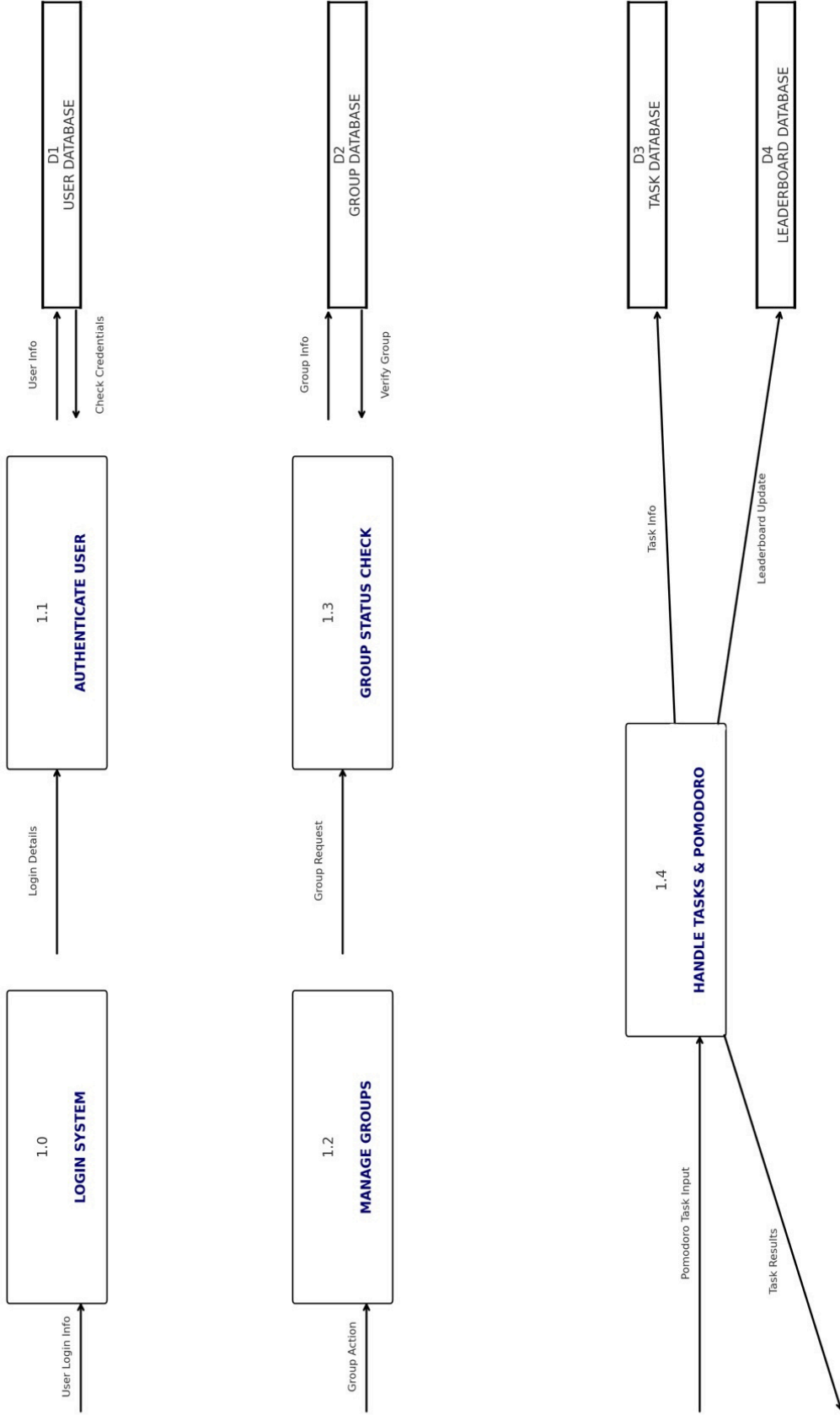
Testing was integrated into the CI pipeline to ensure that unit and integration tests ran automatically, providing immediate feedback on code quality. Additionally, preview deployments by Vercel enabled us to conduct manual testing on isolated feature branches before merging. For monitoring, we relied on Vercel's built-in analytics and error logging tools to track site performance, deployment issues, and runtime errors. This full CI/CD setup helped us maintain a stable and iterative development flow throughout the project lifecycle.

5. Testing Strategy and Implementation

We used combined automated unit tests, integration tests, and manual peer testing to make sure we had feature reliability and stability. We used Jest as our testing framework and React Testing Library to simulate user interactions and assert expected behaviors in components. Unit tests were written for critical components such as the Pomodoro timer and leaderboard to ensure they functioned correctly in isolation. Integration tests verified that different parts of the app worked together seamlessly, particularly in areas where authentication and data retrieval were involved. These tests were executed automatically through GitHub Actions in our CI pipeline, ensuring any problems were caught early.

Beyond automated testing, we also had a peer testing session where classmates interacted with the deployed app and provided feedback through a survey. Their feedback allowed us to identify usability improvements and address bugs that automated testing might not catch.

6. Project Architecture and Data Flow



1.0 LOGIN SYSTEM

Purpose:

Serves as the user entry point into the system, enabling users to log in using either standard credentials or third-party authentication (e.g., Google OAuth).

Inputs:

- User login information (email/password or Google login)

Outputs:

- Authenticated session or login failure message

Data Flows:

- Sends credentials to 1.1 AUTHENTICATE USER
- Receives verification data from D1: USER DATABASE

1.1 AUTHENTICATE USER

Purpose:

Validates the login credentials received from the login system and authorizes the user upon successful verification.

Inputs:

- Login credentials from 1.0 LOGIN SYSTEM

Outputs:

- Session data (name, ID, token) or authentication failure message

Data Flows:

- Queries D1: USER DATABASE for user record
- Returns validation status and user data to 1.0 LOGIN SYSTEM

1.2 MANAGE GROUPS

Purpose:

Handles all group-related operations including creation, joining, and leaving groups.

Inputs:

- Group creation or join request
- Leave group action

Outputs:

- Confirmation of action
- Group status update (e.g., "Joined", "Left", or "Invalid")

Data Flows:

- Communicates with 1.3 GROUP STATUS CHECK for group code validation
- Reads/writes group data in D2: GROUP DATABASE

1.3 GROUP STATUS CHECK

Purpose:

Validates group existence and checks whether a user is a member of a specific group.

Inputs:

- Group join code or group ID

Outputs:

- Group validity status
- Membership information

Data Flows:

- Accesses D2: GROUP DATABASE to verify group and user status

1.4 HANDLE TASKS & POMODORO

Purpose:

Manages Pomodoro timer sessions, task entry, and leaderboard tracking.

Inputs:

- User actions to start/stop timers
- Task description input

Outputs:

- Logged task or session
- Updated leaderboard rankings

Data Flows:

- Stores tasks in D3: TASK DATABASE
- Updates performance metrics in D4: LEADERBOARD DATABASE

Data Stores

D1: USER DATABASE

Stores user credentials, profile information, and login/session data.

D2: GROUP DATABASE

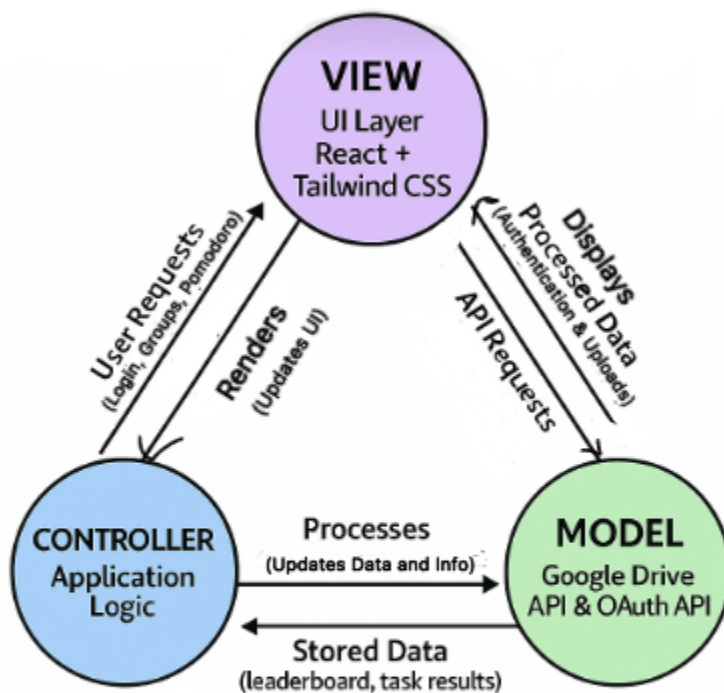
Maintains all group records, member associations, and join codes.

D3: TASK DATABASE

Stores task entries, timer logs, and related task metadata.

D4: LEADERBOARD DATABASE

Tracks user performance and Pomodoro completions for ranking.



View (UI Layer)

Technologies Used: React, Tailwind CSS

The View represents the user interface and is responsible for presenting data to users and capturing user interactions. Built using React and styled with Tailwind CSS, the View dynamically updates based on the state of the application.

- **Responsibilities:**
 - Displays user-facing components such as Pomodoro timers, leaderboards, and group interfaces.
 - Sends user requests to the Controller (e.g., login attempts, group creation, timer start/stop).
 - Reacts to changes in data by re-rendering the interface.

Controller (Application Logic)

Function: Manages user input, controls the flow of data, and applies logic.

The Controller serves as the intermediary between the View and the Model. It receives input from the View (e.g., user actions like starting a Pomodoro session or signing in), applies logic, and coordinates the necessary updates.

- **Responsibilities:**
 - Handles logic for user actions such as authentication, task tracking, and group operations.
 - Validates user input and manages state transitions.
 - Communicates with the Model to fetch or update data (e.g., updating leaderboard scores).

Model (Data + API Integration)

Technologies Used: Google Drive API and OAuth API

The Model is responsible for managing the application's data and external API interactions. This includes securely integrating with Google Drive and OAuth for file and user management.

- **Responsibilities:**
 - Stores and processes user data such as task completion, group information, and session logs.
 - Handles API requests from the Controller and returns results (e.g., updated leaderboard, user files).
 - Maintains persistent data such as saved Pomodoro results and shared documents via Google Drive.

7. List of Known Bugs and Issues

Bug #	Description	Steps to Reproduce	Severity	GitHub Issue Link
1.	Long group names extend the page.	1. Create a group with a very long name 2. Navigate to the group page 3. Observe layout overflow	Low	https://github.com/CMPT-276-SPRING-2025/final-project-07-hills/issues/41

2.	Login requires two attempts after logout.	1. Log in 2. Log out 3. Log back in (will take two attempts to succeed)	Medium	https://github.com/CMPT-276-SPRING-2025/final-project-07-hills/issues/42
3.	Pomodoro timer not synced across tabs.	1. Open the site in two tabs 2. Start the timer in one tab 3. Observe desync in the other tab	High	https://github.com/CMPT-276-SPRING-2025/final-project-07-hills/issues/43
4.	No confirmation when leaving a group.	1. Join a group 2. Click 'Leave Group' 3. Observe immediate removal without warning	Medium	https://github.com/CMPT-276-SPRING-2025/final-project-07-hills/issues/44
5.	Leaderboard loads without visual feedback.	1. Navigate to the Pomodoro page 2. Observe delay before leaderboard data loads 3. No loading spinner is displayed	Medium	https://github.com/CMPT-276-SPRING-2025/final-project-07-hills/issues/45
6.	No confirmation when logging out.	1. Log in 2. Click logout 3. Observe redirection with no confirmation message	Medium	https://github.com/CMPT-276-SPRING-2025/final-project-07-hills/issues/46

7.	No prompt when closing page during active Pomodoro.	1. Start a Pomodoro timer 2. Try closing or refreshing the tab 3. Observe no warning about losing progress	High	https://github.com/CMPT-276-SPRING-2025/final-project-07-hills/issues/47
----	---	--	------	---

8. Future Work and Potential Improvements

Although every planned M1 feature was effectively carried out, we thought about including a flashcard feature to help students with their effective studying. Users could build decks of question-and-answer cards using the flashcard system. We would create an interactive interface to track progress and flip through cards using JavaScript and React. Flashcards could be made, modified, and reviewed either alone or in a group setting. There could be a leaderboard and points would be given every time that a user successfully finished going through the deck. Should we use Firebase as the backend database to keep flashcard data, it would enable group access and real-time syncing. Decks could be arranged like our present document structure and connected to current study groups.

9. Lessons learned and Project Takeaways

This project provided valuable experience in full-stack development, team collaboration, and problem-solving. One major challenge was integrating Google authentication with Firebase while securely handling user data. We overcame this by learning how to manage access tokens and structure user profiles effectively. Another challenge was building real-time group features and the Pomodoro leaderboard, which required careful database design and synchronization using Firebase's real-time tools. Time management was crucial, and prioritizing core features helped keep the project on track. We learnt valuable technical skills; including how to implement APIs in an application, how to work with frameworks like React, and how to use modern improvements to existing skills such as Tailwind CSS. Overall, the project strengthened our abilities and taught us how to work efficiently as a team.

10. Appendix

Contributions:

Julianna Morena:

Milestone 1: The SDLC model, risk assessment, the overview of the project in the video, editing and subtitles in the video, appendix.

Milestone 2: Lessons learned and project takeaways, appendix, future work and potential improvements in the video, editing and subtitles in the video.

Project: Added the ability to delete documents, the ability to change the name of files from google docs and update it on the interface, the ability to leave groups. Worked on the frontend of the project, which includes the popup menus and the ability to copy and paste the group code.

Mudasser Mashal:

Milestone 1: Google Drive API description and features, work breakdown structure, project schedule, overview of the chosen APIs in video.

Milestone 2: Meeting User Needs & Solving the Problem, SDLC Model Analysis, Features Implemented for Each API, CI/CD Pipeline Overview. Project overview in the video, project takeaways and lessons in the video.

Project: Worked on Pomodoro Timer functionality and leaderboard feature. Fixed issues related to Group and File names.

Yogya Agrawal:

Milestone 1: The mid-fidelity prototype, walkthrough of prototype in video.

Milestone 2: API features with their descriptions, demo video recording and voiceover

Project: Worked on the google authentication and connecting firebase to the project. Worked on creating a group and storing it in the back-end database. Also worked on storing all the other required information using firestore. Connected the Google drive API and ensured ability to create documents and upload files, with the ability to rename and delete files. Also worked on the ability to view associated google accounts with the group.

Nafeesa Leena:

Milestone 1: Google Docs API description and features, Data Flow Diagrams, MVC model diagram, overview of application data flow in video.

Milestone 2: Data Flow updated Level 1 Diagram, Updated MVC diagram, Bugs and Issues, and CI/CD pipeline in the video

Project: Worked on the Group the visual group creation, worked on the popup menu to put the group name for group creation, worked on the UI and visibility of groups on the main page, and worked on the Pomodoro UI as well as the ability to change time on Pomodoro.

Changelog:

We used Google OAuth instead of Google Docs API, as the Docs API requires security clearance and other documents before the developer has access to the API.

Peer Testing Feedback Form:

<https://docs.google.com/forms/d/e/1FAIpQLScGyZRLewlC1RoW8K3j1kUGDD4g0pxgotogN5zW4KagzAdcaw/viewform?usp=sharing>

Peer Testing Feedback:

Website Cirkle Rating

Name (not needed)

2

rate from 1-10

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5
- ☐ 6
- ☐ 7
- ☐ 8
- ☐ 9
- ☒ 10

Things you liked about Cirkle

the colours were very pretty! and i also like making a document. i liked the timer it too

Things you disliked about Cirkle

maybe add instructions for more purpose and stuff

Recommendations for Cirkle

how do i see members of my cirkle jk make it more clear

Other thoughts or comments

very good guys

This content is neither created nor endorsed by Google.

Google Forms

Website Cirkle Rating

Name (not needed)

rate from 1-10

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5
- ☐ 6
- ☐ 7
- ☐ 8
- ☒ 9
- ☐ 10

Things you liked about Cirkle

I like the pomodoro points fire

Things you disliked about Cirkle

more spacing for the top nav

Recommendations for Cirkle

sign out and pomodoro on the 2 ends

Other thoughts or comments

fire app

This content is neither created nor endorsed by Google.

Google Forms

No Additional Diagrams/Charts/Tables