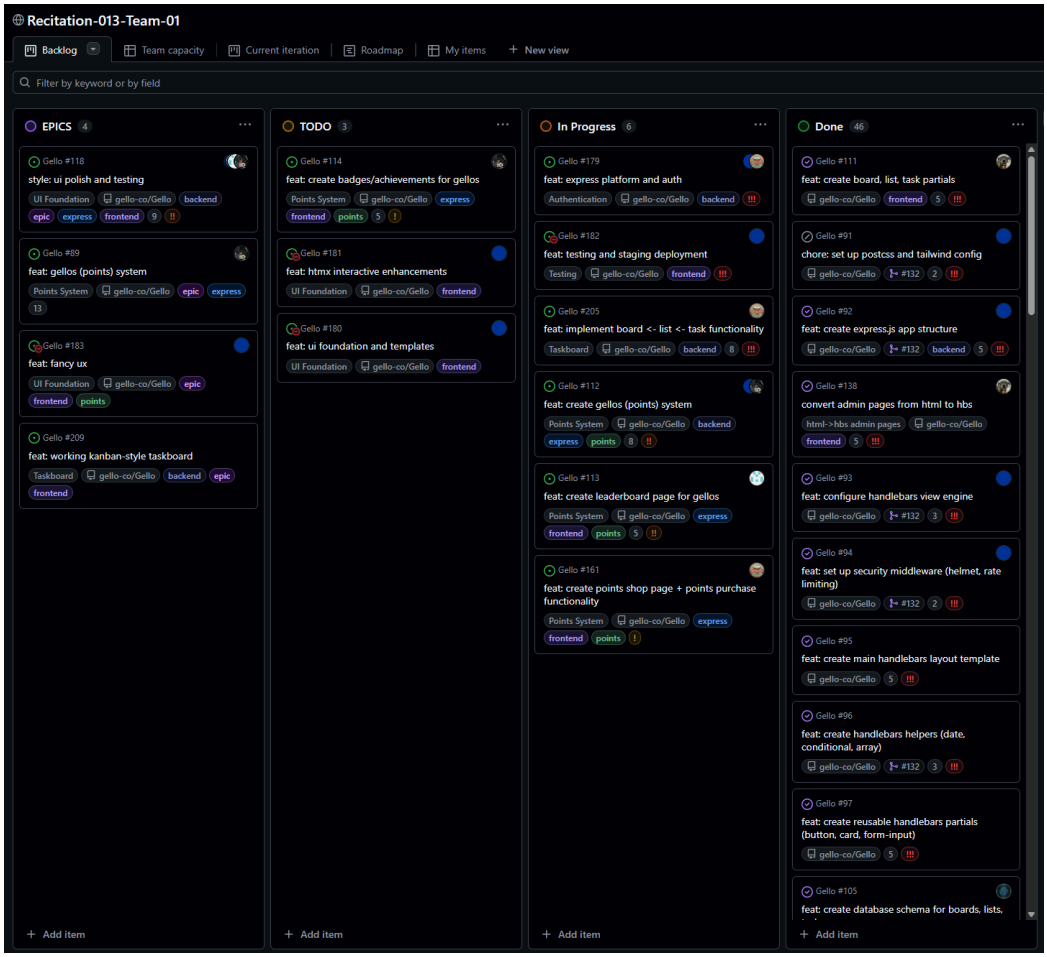
Title: Gello

Who: Jasmitha Gourabathini, Wren Gurung, Payton Hsu, Ronald Gospel Jangam, Joanne Kim, Bradley Wistehuff

Project Description: Our application, Gello (“Jell-oh”), focuses on productivity and team organization. Users can take on admin roles to organize teams and assign tasks/goals for team members. Team members can sign in to specific teams and complete tasks assigned by the admin to earn points. Due to our implemented security and authentication, access is limited to specific teams in case of use of sensitive data.
To differentiate Gello from other productivity applications, our application gamifies productivity by using a point system. This works to introduce friendly competition and motivation into the workplace. Team members are awarded points based on task difficulty and meeting deadlines assigned by the team admin. Points can also be used to redeem rewards on the team member side.

Project Tracker - GitHub project board:
https://github.com/orgs/gello-co/projects/3

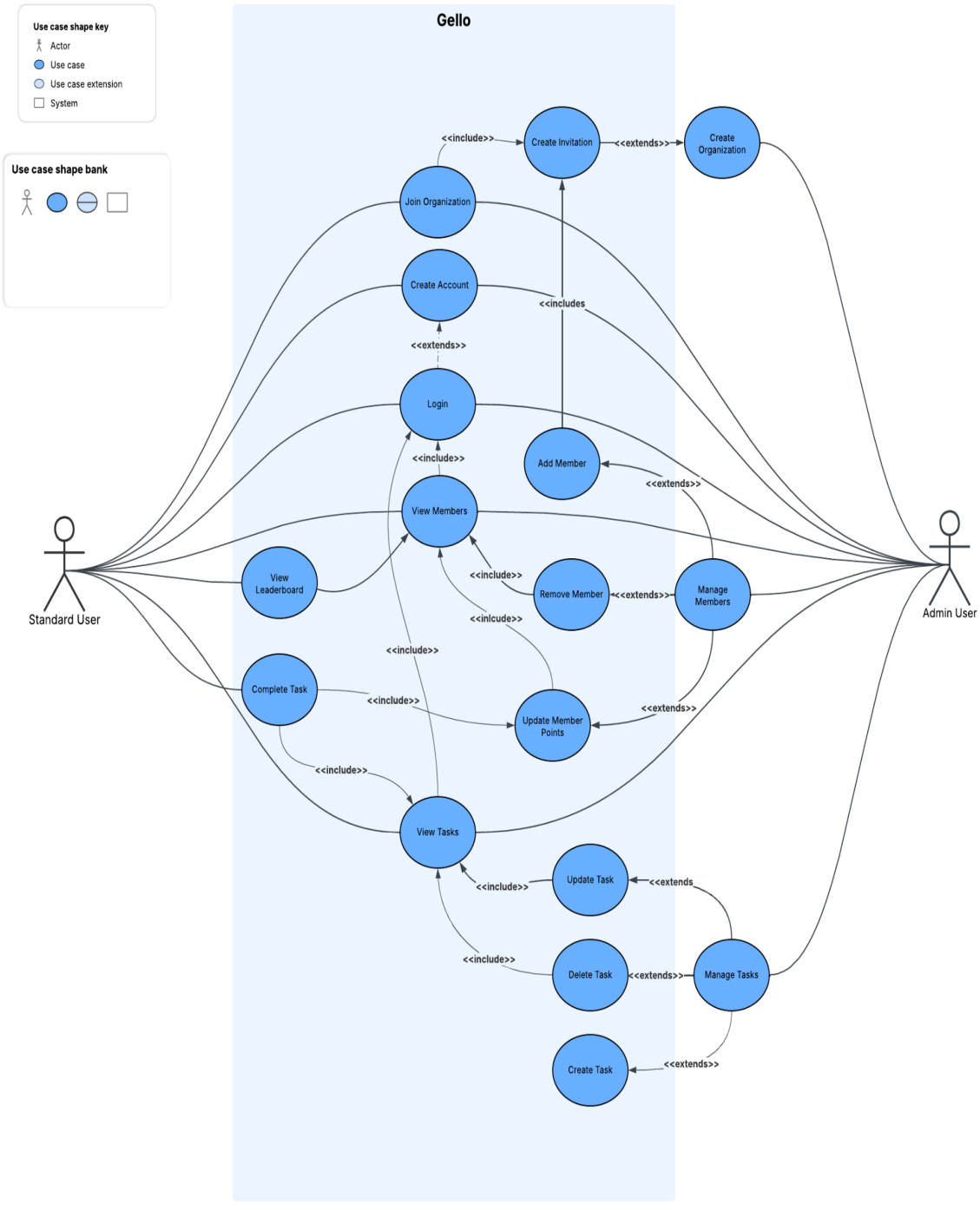
Screenshot showing your project in your project tracker



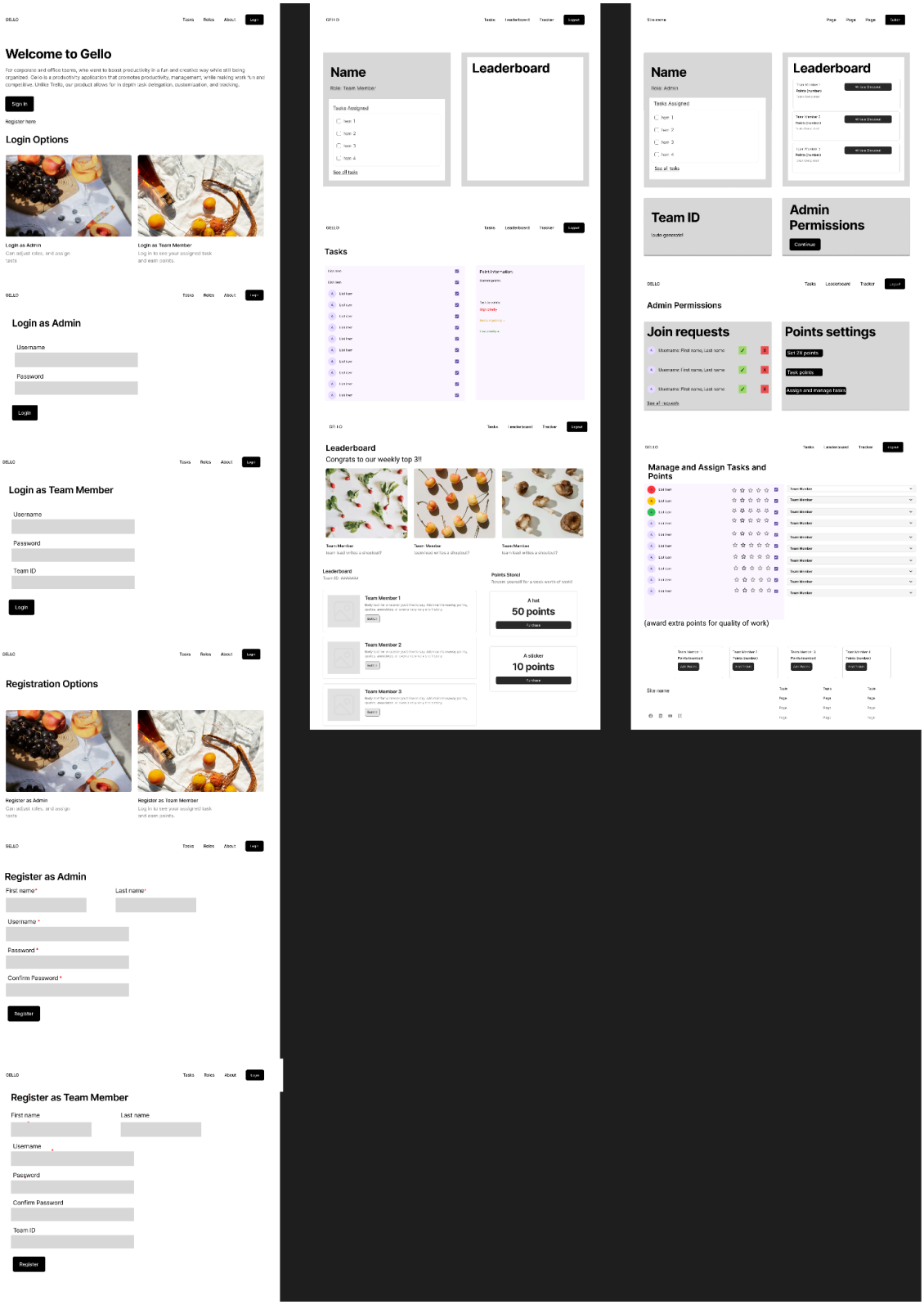
Video: https://www.youtube.com/watch?v=BS2NTseb7lo

VCS: Link to your git Repository
https://github.com/gello-co/Gello

Use Case Diagram:



Wireframe:


Welcome to Gello
Sign In
Register here
Login Options
Login as Admin
Login as Team Member
Login as Admin
Username
Password
Login
Login as Team Member
Username
Password
Team ID
Login
Registration Options
Register as Admin
Register as Team Member
Register as Admin
First name*
Last name*
Username *
Password *
Confirm Password *
Register
Register as Team Member
First name
Last name
Username
Password
Confirm Password
Team ID
Register
Name
Role: Team Member
Tasks Assigned
See all tasks
Leaderboard
Tasks
Leaderboard
Congrats to our weekly top 3!!
Team Member 1
Team Member 2
Team Member 3
Points Store!
A hat
50 points
A sticker
10 points
Name
Role: Admin
Tasks Assigned
See all tasks
Leaderboard
Team ID
Admin Permissions
Continue
Admin Permissions
Join requests
Points settings
Set 2X points
Task points
Assign and manage tasks
Manage and Assign Tasks and Points
(award extra points for quality of work)

Test results:
Feature 1: Admin Pages Access Verification
This test verifies that admin users can visually access all pages in the application, run on localhost. It uses Playwright's visual snapshot testing to capture screenshots of each page for visual verification. If pages render correctly, a message will be sent in the console that it has successfully accessed each page. The user acceptance testers will reflect project team managers.

The user navigated to the registration page successfully. They are entering their information as directed. Their behavior is consistent with the use case. There are no deviations from the expected actions, and it seems as if it is pretty straightforward. After logging in, they are able to click through the different pages and view all of the pages.

Results:
- Pages were rendered successfully on localhost
- Browser displayed properly, and screenshots were successful
- No access or navigation errors

Feature 2: Task Creation and Deletion
This test verifies that admin users can create and delete task, run on localhost. The test should return a response verifying that the task has been added. The user acceptance testers will once again reflect project team managers.

After registering and logging in as an admin, the user has navigated to the tasks page successfully. The user was able to open the new task modal by clicking on the “New Task” button. The user entered the data correctly and the task showed up on the page. This behavior is consistent with the use case. There were no deviations from the expected actions.

Results:
- Team creation, editing, and deletion works as expected
- No permission errors

Feature 3: Completing tasks by users
This test verifies that team members can successfully complete tasks and earn points. It should send a message that the task has successfully been completed. The user acceptance testers will reflect the team members within a team in the workplace.

After registering and logging in as a team member, the user has navigated to the tasks page. The user was able to see all of their assigned tasks. The user was able to click the “Complete” button and complete the task. After looking at the admin view, it was also marked as completed on the admin end.

Results:

- User is able to log in as team member successfully
- Tasks page renders correctly
- Task is able to be completed
- Team member's points reflect correct amount before and after completing task

Feature 4: User Login Verification
This test verifies that both admins and team members can successfully log in using valid credentials. It ensures that the login form works correctly and that users are redirected to the appropriate dashboard after logging in. The test is run on localhost. The user acceptance testers represent typical users logging in to access the system.

The users navigate to the login page, enter their email and password, and click the "Login" button. Their behavior is as expected. Users entered their information and logged in successfully without errors. The login form worked as planned, and no issues were identified during testing.

Results:
- Login page loaded successfully
- Users were able to enter credentials without errors
- Login redirected users to the correct page
- No navigation or authentication issues occurred

Deployment: The app was deployed on Render.
https://gello-31ik.onrender.com/

Team Contributions:

Jasmitha Gourabathini
I mainly focused on working with the front end team beginning with helping with the wireframing process on Figma. After that, I drafted the original home page, navigation bar, and footer in html, and helped migrate the static html pages into handlebars. I also focused on making the leaderboard pages and worked on styling the login and registration pages. Lastly, I also helped with making the project presentation.

Wren Gurung
I primarily focused on front end development, creating visual assets to use for Gello, along with project management. I created all the art assets for Gello while also creating pages such as the home profile pages for admins and team members. The pages were first created in HTML with CSS styling before we converted them to handlebars. Throughout the project I helped to organize the team, making priorities and task delegations clear whilst remediating conflicts and concerns. I also helped to design parts of the wireframe with Figma and handled group communication by creating a Discord server for the project.

Payton Hsu
I mainly did backend development, creating express routes and connecting these routes to the front end. I implemented and debugged some front end functionalities while I was connecting these routes. I also created the front-end UI for the points shop page and worked on some of its backend features. I refactored the project file structure for easier navigability and also did a little bit of work creating a validation schema with Zod and wrote some services that connected the database to the express routes. Also, I made the Use Case Diagram.

Ronald Gospel Jangam
No response given.

Joanne Kim
I mostly helped on the front end team, creating and designing the overall wireframe with Figma and creating the web pages with Handlebars, HTML, and CSS. Specifically, I created the login, register, and tasks pages for both the team member side and admin side. I also styled these pages with CSS to accurately reflect the color scheme and the style of the rest of the pages. Amidst changes to our stack, I helped my team to migrate the original pages from HTML to Handlebars. Lastly, I designed and helped to finish the project presentation, architecture diagram, project proposal, and the final report.

Bradley Wistehuff
My contributions were focused on applying contemporary software development methods, tools, and security practices.
Design: test driven express layered service architecture (routes - services - database), full Zod schema validation, singleton Supabase client.
Auth/Security: Supabase SSR cookie-based sessions and PKCE OAuth flow, CSRF protection, Helmetjs CSP/security headers, RLS policy functions with atomic RPC.
Tests: full pgTAP database policy tests, Playwright E2E tests with visual regression - 51 cases, Supertest integration tests - 153 cases, Vitest unit tests - 162 cases, database mutex protection.
DevOps: CI/CD pipeline with github actions and render blueprints, Supabase schema and migrations, lighthouse auditing.
Docs: C4 architecture diagrams, Revealjs presentations, full type coverage across all layers.