Title: *Perfect Planner*

Group Members:

- Patrick Liu
- Abby Haines
- Julio Gonzalez
- Luke Fitzpatrick
- Dylan Bomgardner

Project Description:

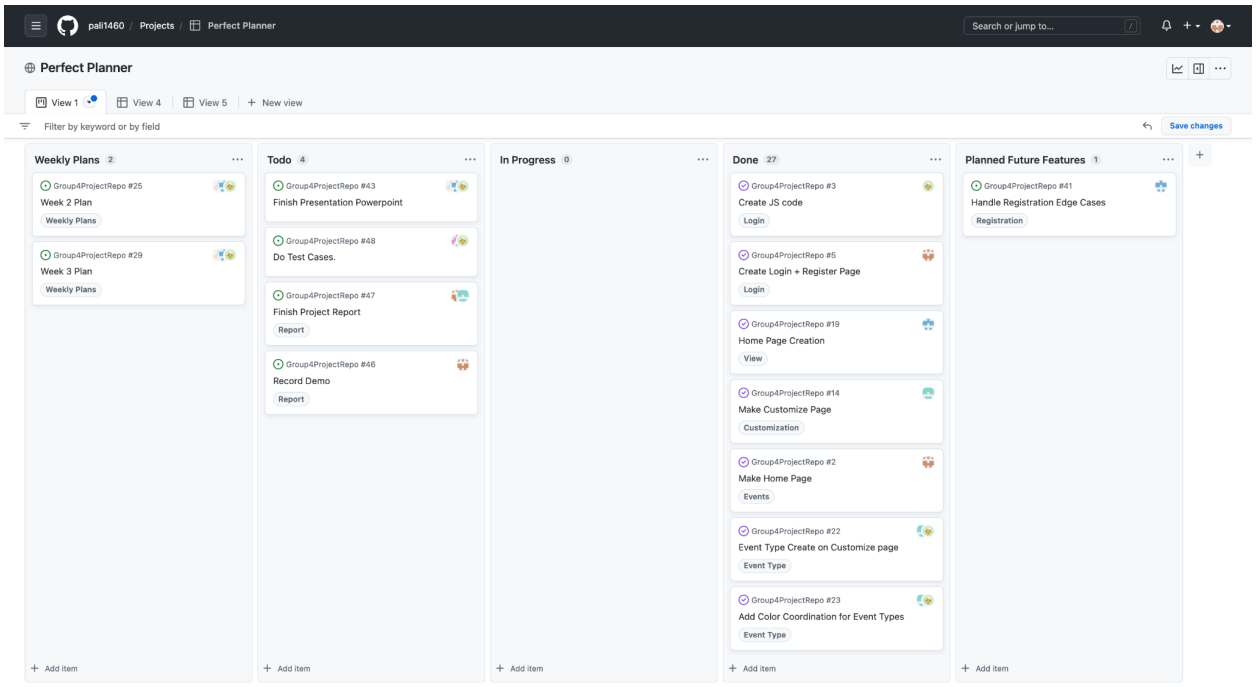
The Perfect Planner is an app that helps organize everyday tasks. This planner makes it easy to see what you have to accomplish that day. The Perfect Planner will have users enter all events and assignments, filter them, and help organize them for the user.

The user will be able to create and log in to an account and access their list of assignments and other events, as well as create their own. They will further be able to sort events by type, and be able to filter their list of events by type. Users will be able to allow this application to send them alerts for the events. Users will be able to set the time these alerts appear. Such as a week before, a day before, or the day of. To further customize the experience users will have the option to adjust element colors as well as add their own pictures and images.

The Perfect Planner is a simple but practical application that allows college students to effectively plan and manage their time.

Project Tracker - GitHub project board:

https://github.com/users/pali1460/projects/1



Video:

https://drive.google.com/file/d/1-tqnosHJqWMCSNe1iycFVhLKF9Yrnie9/view?usp=share_link

VCS:

https://github.com/pali1460/Group4ProjectRepo

Contributions:

Patrick Liu:

- I primarily worked on parts of the SQL database (the portions related to events in particular) and contributed to the event display page, event adding page, home page, and admin page within the project. I also worked on the Javascript functions related to said pages, such as the functions to add events to the database, fetch relevant information from the database, and delete information from the event, eventtype, and user databases. I also worked on configuring the Docker container such that the project ran correctly and in the right time zone.

Abby Haines:

- In the first week Luke and I were responsible for creating the UI for our project in order to have some direction to go in with our code. Then I created the initial calendar view page and started trying to put events onto the correct calendar. I also created a default event type that is automatically added when each new user is registered. This fixed a problem where new users did not have a default event type and could not create events unless they first created an event type. With this change, all new users could immediately start creating events without needing to first create an event type.

Julio Gonzalez:

- My contributions to this project weren't as large as all of my other teammates but what I did do was give out my idea for the project. I had it thought out on what it should do. I also helped around because I just couldn't find my one spot like the UX or Javascript. I took care of testing, I had people in my dorm test the website to see if it was user friendly and I took notes on how they interacted with the product, whether they traversed through it smoothly or if there was confusion with certain features.

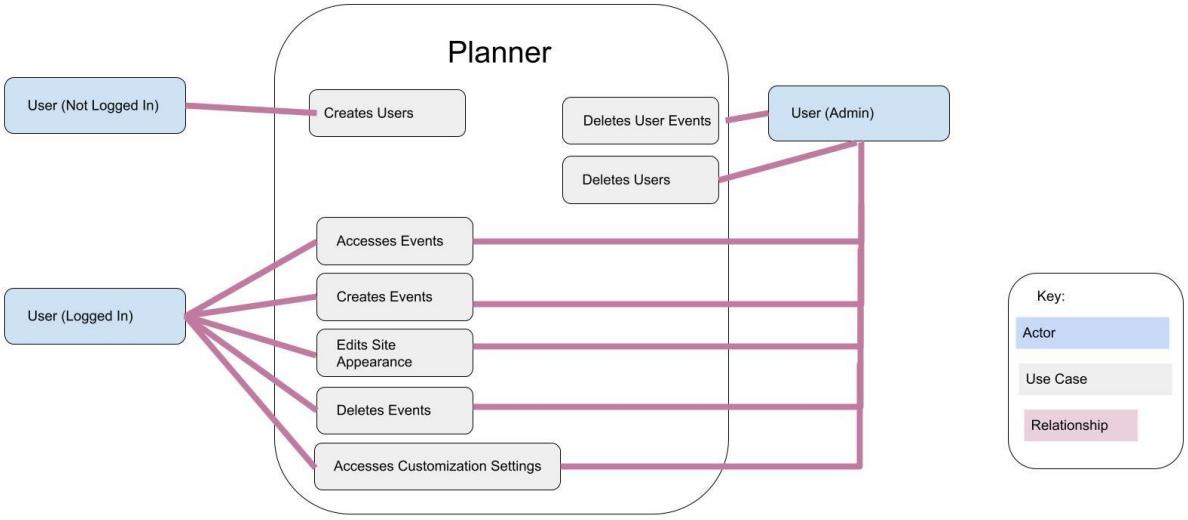
Luke Fitzpatrick:

- For this project, I focused on background customization, event types, and the calendar page. I also worked on a simple UI prototype with Abby throughout the first week. For customization I used node js, SQL, HTML, and javascript to allow the user to change the background. I implemented event types using node js, SQL, HTML, javascript. I was able to build upon the event creation framework already implemented. For the calendar pages I made a dynamic and responsive calendar that would allow the user to toggle between months. For this I used only HTML and javascript.

Dylan Bomgardner:

- Created the notification system so that when rendering the home page, a notification will be sent to the desktop of the user. The notification will include events that had the warning set to that day. The website will first ask for permission to allow notifications, then once rendering the home page it will be sent to the desktop. This was done with js logic on the home.ejs page utilizing variables from ejs.
- Fixed a bug that crashed docker when a duplicate username was registered.

Use Case Diagram:



Test results:

Feature: Login and Registration	
Acceptance Criteria	Accepted (Y/N)?
A user cannot submit the login/registration form without completing the required fields: Username and Password	Y
Information from the form is stored in the “users” table in the database	Y
The password entered is hashed and stored in the database.	Y
No two users can have the same username.	Y
Feature: Add/Delete Events	
Acceptance Criteria	Accepted (Y/N)?

A user cannot submit an event without providing a name and date. The user, however, should be able to submit an event with or without any of the remaining fields.	Y
Information from the form is stored in the “events” table in the planner_db database	Y
The calendar should be able to display all events created by the user, but *not* events created by other users.	Y
Feature: User is Alerted	
Acceptance Criteria	Accepted (Y/N)?
User should receive an alert when the “warn date” and “warn time” pass	Y
Events that have had a “warn date” and “warn time” pass should be displayed on an “urgent” calendar on the user’s homepage in the format of a list.	Y
Feature: Data Saved Between Sessions	
Acceptance Criteria	Accepted (Y/N)?
Upon logging out and back in, the user should be able to view the events they created in a previous session. This applies even if other users logged in between sessions.	Y
Feature: Visual Customization	
Acceptance Criteria	Accepted (Y/N)?
User should be able to use a menu listing each element they may customize	Y
New information should then update each element and be saved into the database	Y

Upon reloading the page the page should take on the new appearance specified by the user	Y
Each element the user can choose a color from a color wheel or input and image url	Y
Overall Acceptance?	**Y**

User testing:

We let a couple of CU students test our website to see how someone with no knowledge of the features or implementation of our idea would navigate through our product. The testers were able to register, log in, create events, look at their events, etc. with ease with some confusion on a certain feature because they didn't explore a page. The page was the customize page, which isn't a required page for the functionality of the website but it's definitely a helpful tool to make your website look more personalized or for you to organize your events better. After they found that page, they added an event type and tested it out but I saw it as a little flaw because we knew that feature existed and would've gone to that page first if we wanted to create an event whereas a new person wouldn't explore that page until after playing with the main functionality of the website.

Deployment:

To deploy the project, navigate to the Project subdirectory within the downloaded Group4ProjectRepo folder. Once there, run the “docker compose up -d” command and navigate to localhost:3000 once Docker has finished deploying.

To shut the planner down, run the “docker compose down” command in the same directory. If you wish to delete all data, run the “docker compose down -v” command as before.

To create an administrative account, register an account with the username “admin” and any password. You will then be able to access admin features once you log in to the system.