Git’r Done - Project Milestone 1:

Team Name: Git’r Done

Members: Jack Marty, Brandon King, Margad Batamgalan, Miles McCorkendale, Noah Berry, and Sanjana Shenoy

Description: The basis for our project will be to create a calendar in which users can add events to specific days and keep track of their daily tasks. By default, if will notify the user of an event 10 minutes prior to the event happening unless otherwise specified. A unique feature of our app will be that if a location is specified for an event, the app will give weather updates based on that location. The goal is that our users will be able to more accurately plan their schedule around the weather and have the ability to know if the weather will impact their plans ahead of time. This should make it easier for people to quickly reschedule events that would otherwise be cancelled by giving them immediate alternatives when an unfavorable weather change is detected. Our app will be a website, so notifications will happen via JavaScript Alert and email if a user email is provided.

The display of our calendar will resemble a handheld calendar where each month has its own picture. Users can upload their own pictures to customize the display. Since it is an online a calendar, users will be able to change the view by selecting a day, month, or a whole year displaying format. Users will benefit from this calendar because it will be a fully-functional, customizable event planner that is simple to use, easy to navigate, and effective in day-to-day needs.

Vision Statement: Our vision is to have an online calendar that lets users keep track of their personal schedule and gives weather updates based on the time and location of the event.

Motivation: Our team thought of this idea when we realized how much someone’s scheduled events depend on the weather. Boulder is a town with an outdoor vibe and many people would like to coordinate their events according to the weather. We all thought it’d be beneficial for users to track weather and have their calendar on one platform. This would also help people in making sure their events can run smoothly without any delays or cancellations due to weather issues.

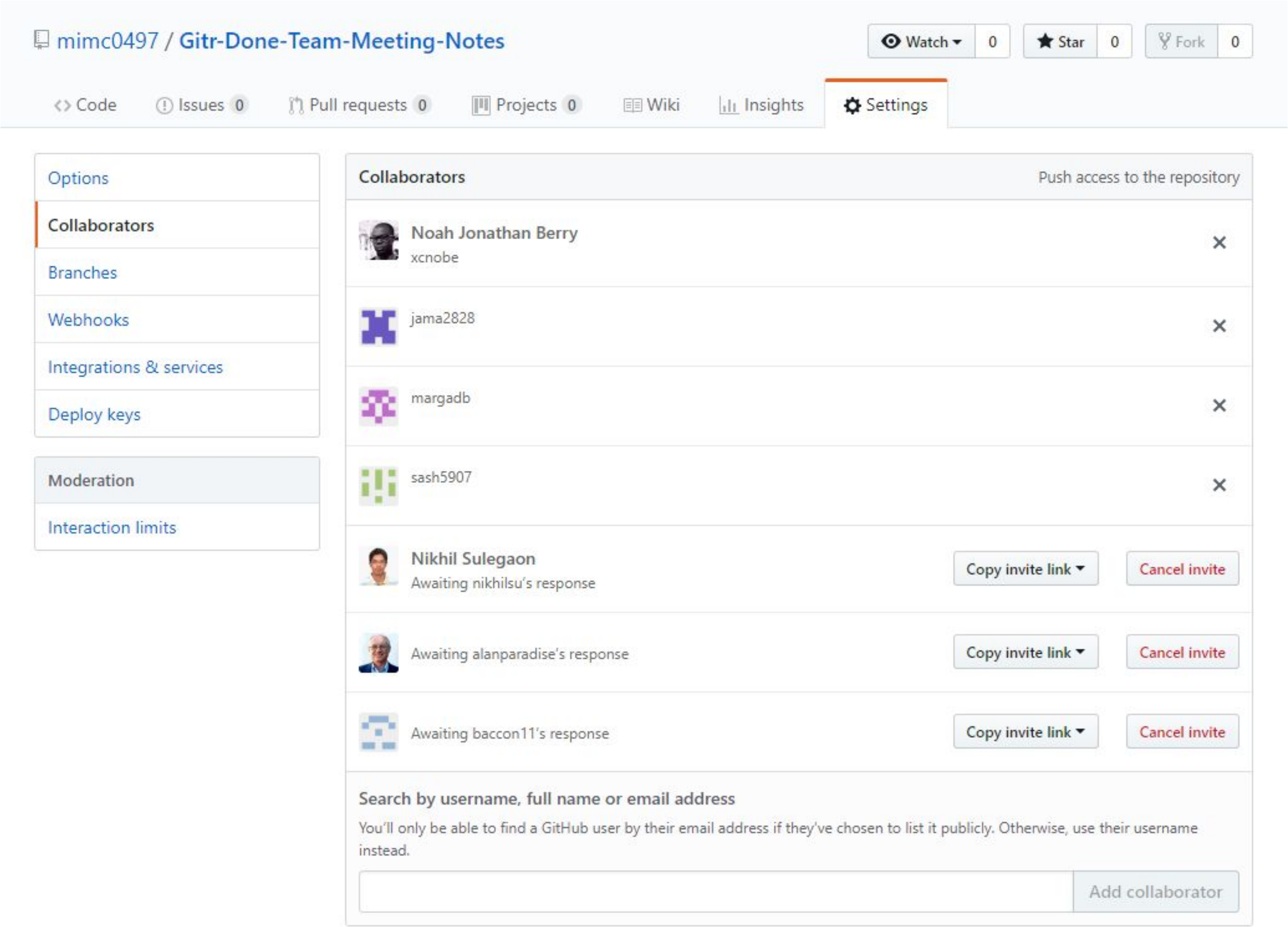
Risks:

- Not everyone in the group is experienced in all the programming languages we intend to use and that may cause delay when we assign everyone tasks to complete
- Everyone has different schedules with various amounts of school work and timings of work shifts, so all of us always meeting at the same time may gradually get harder as the semester continues
- There is the risk that our intended product might be developed outside of the planned scope and objectives, resulting in unplanned features or additions to the goal. (AKA scope creep)
- Since this is everyone's first time working together, there maybe some poor communication within the team.
- Compatibility issues may arise when trying to integrate scripts and APIs for things like the calendar and weather data since our team will likely be importing ones we have never used before.
- Obtaining a domain name and adequate server space for the website may be expensive or troublesome.

Risk Mitigation Plan:

- We plan on communicating our times available for the weekly face to face meeting ahead of time so we can work around any scheduling conflicts.
- We also plan to allow for extra time when giving assignments to someone who is not fluent in the specific programming language the assignment requires.
- We plan to use programming languages and that do not greatly exceed our group's experience, and plan to thoroughly research language and tools before committing to using them.
- We will try not to use tools with which only one group member is familiar. This will allow other members to provide support if someone becomes stuck, or overwhelmed.
- To deal with scope creep, we have to plan for complications and setbacks as far as production goes, and make adjustments accordingly. We also have to be realistic in terms of resource availability and deadlines in order to achieve quality results.
- We are using collaborative software to work together, so that everyone stays informed.
- Our team has access to personal computers powerful enough to host a web server if needed, but this is not a long term solution.

Version Control:
https://github.com/mimc0497/Gitr-Done-Team-Meeting-Notes



Collaborators — Push access to the repository

Noah Jonathan Berry
xcnobe

jama2828

margadb

sash5907

Nikhil Sulegaon
Awaiting nikhilsu's response

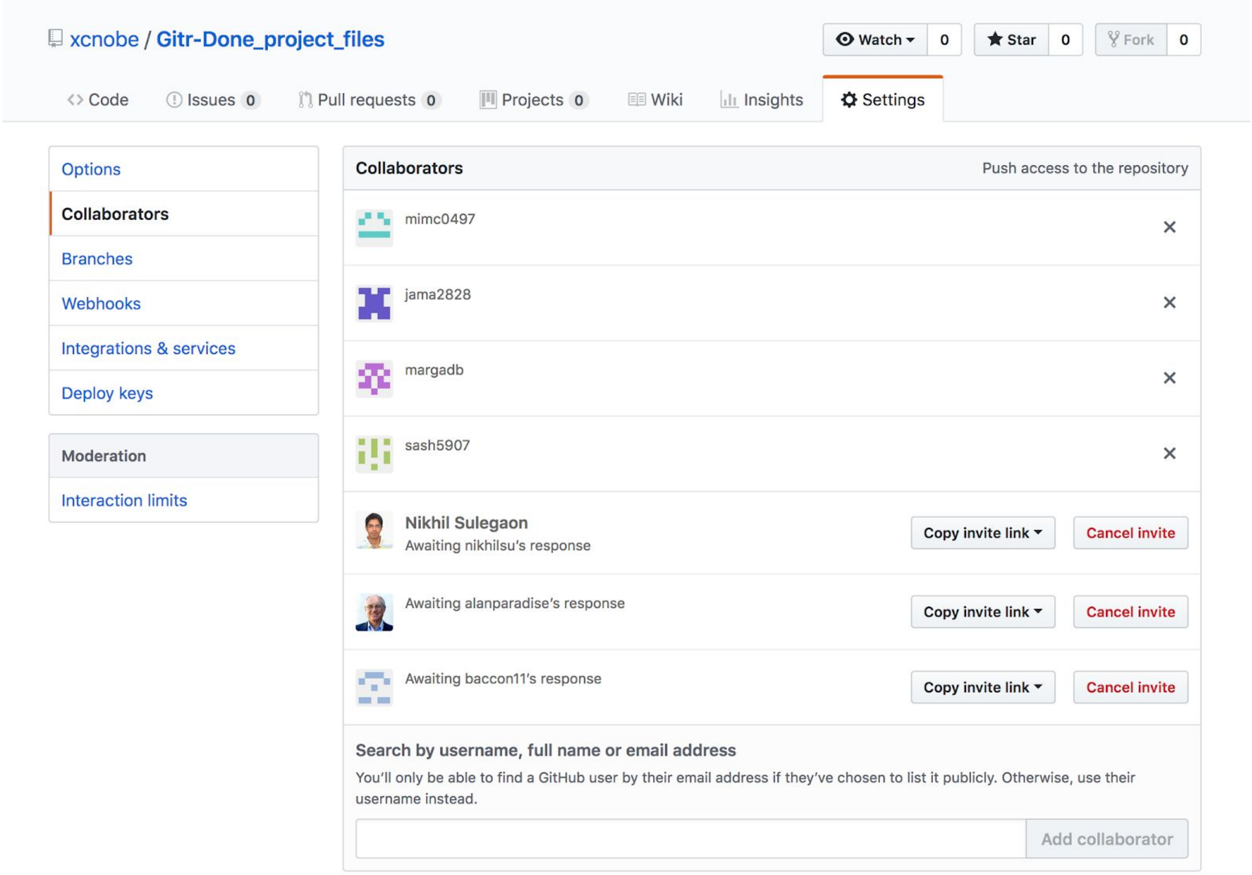
Awaiting alanparadise's response

Awaiting baccon11's response

Search by username, full name or email address

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

https://github.com/xcnobe/Gitr-Done_project_files



Collaborators Push access to the repository

mimc0497 ✕

jama2828 ✕

margadb ✕

sash5907 ✕

Nikhil Sulegaon
Awaiting nikhilsu's response — Copy invite link ▾ — Cancel invite

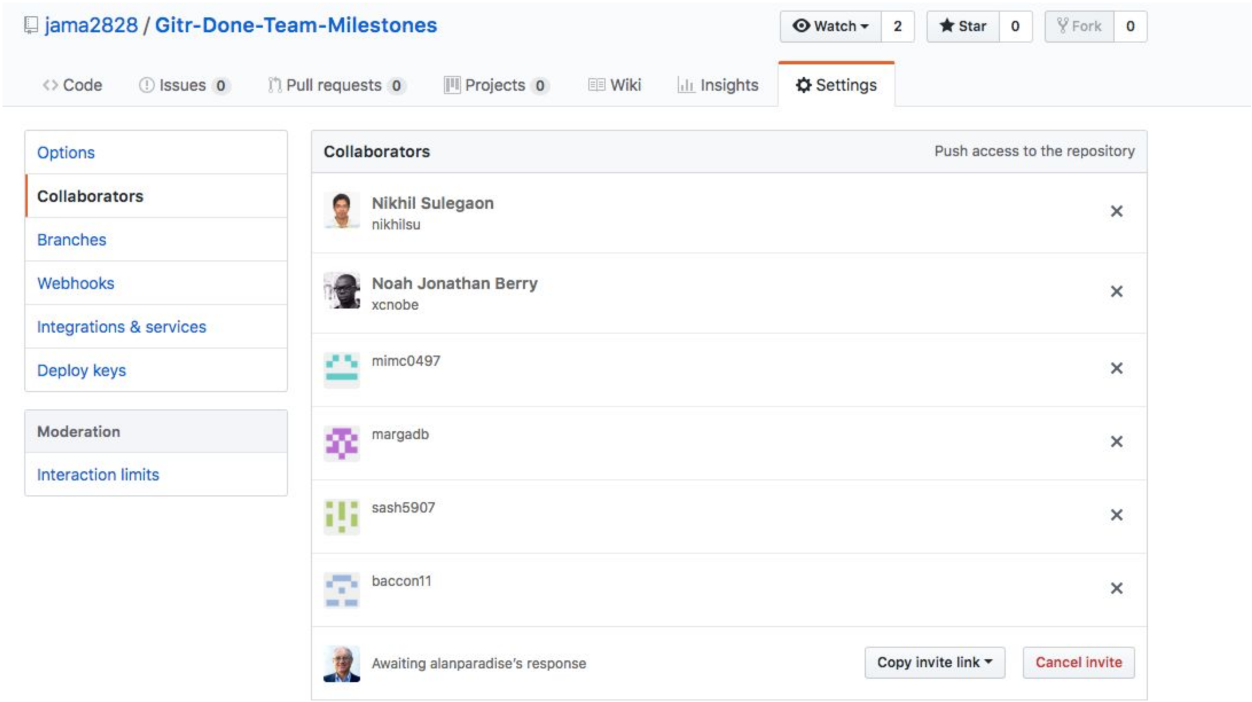
Awaiting alanparadise's response — Copy invite link ▾ — Cancel invite

Awaiting baccon11's response — Copy invite link ▾ — Cancel invite

Search by username, full name or email address

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

https://github.com/jama2828/Gitr-Done-Team-Milestones



jama2828 / Gitr-Done-Team-Milestones

Watch 2 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

Options
Collaborators
Branches
Webhooks
Integrations & services
Deploy keys

Moderation
Interaction limits

Collaborators — Push access to the repository

Nikhil Sulegaon
nikhilsu ×

Noah Jonathan Berry
xcnobe ×

mimc0497 ×

margadb ×

sash5907 ×

baccon11 ×

Awaiting alanparadise's response — Copy invite link — Cancel invite

Development Method: We have decided to use an Agile development structure because we are working with many tools and ideas that we have not used before and therefore need to be flexible. If we run into a problem early on in a step along the way, Agile will allow us to make necessary changes without derailing other development cycles before or after the fact. Agile will allow us to separate creating the webpages, creating the database, integrating the two, and specific user functions into separate "sprints" so that we can deal with the unique issues of each area separately and specifically.

Collaboration Tool: We have made a group chat through the app Slack.

Proposed Architecture:
Frontend: HTML, CSS, JavaScript.
Backend: SQL, C
Midlayer: NodeJS

The user interface of the website will be through HTML webpages using JavaScript and CSS for style and functionality. We will import a style sheet and integrate scripts and APIS for calendar and weather data. These webpages will get and edit user information such as usernames, passwords, and calendar events from a SQL database. A NodeJS layer will facilitate reading from and writing to the database. Once the webpages are working as intended, we will need to use a service such as Google Sites to go live.