Github & Git

Penn Engineering

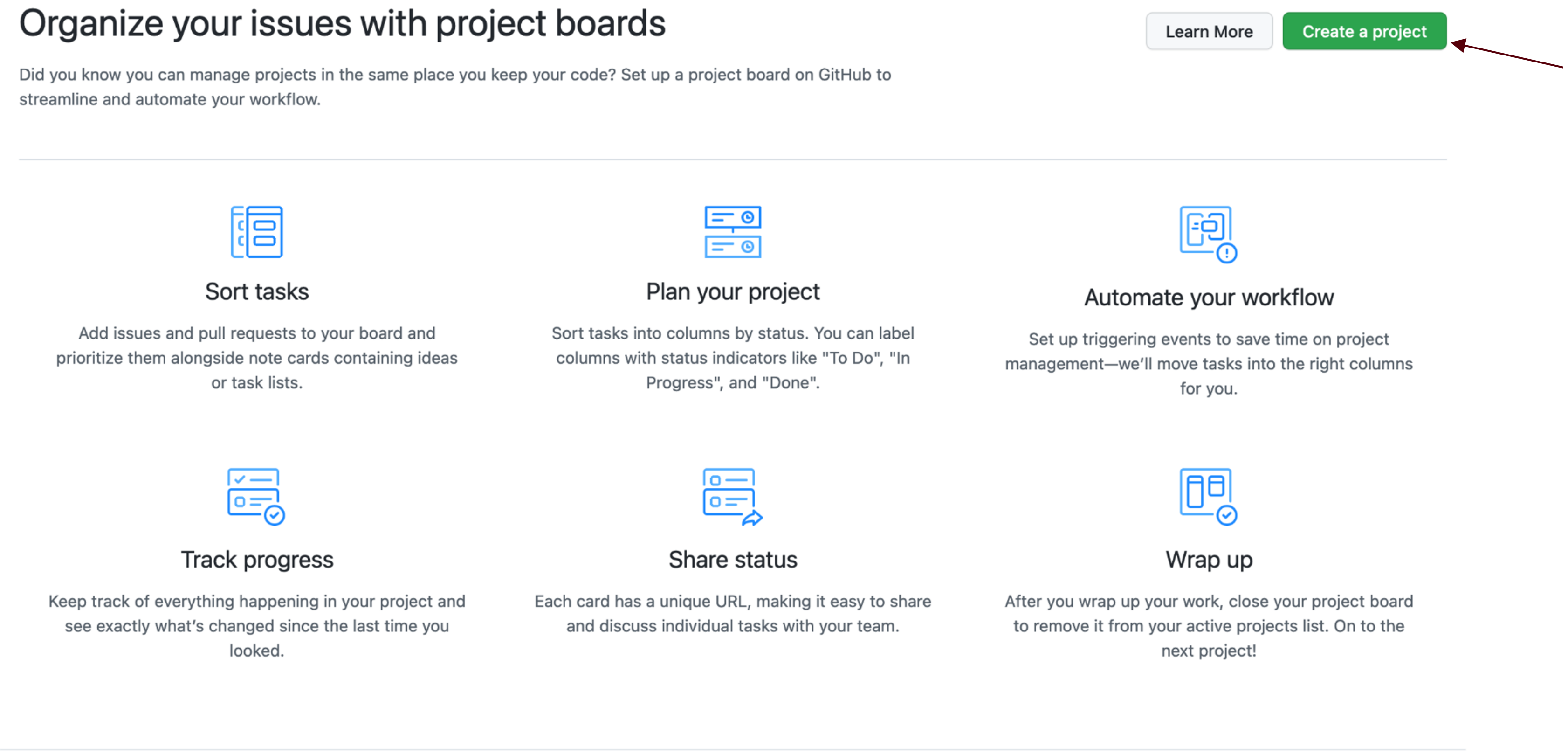
Github Project

Agile Project Management

XP Concepts

- User stories
- Release planning
- Iteration planning
- Project velocity (story points)
- Test driven development
- Pair programming
- Continuous Integration

GitHub Project



Organize your issues with project boards

Did you know you can manage projects in the same place you keep your code? Set up a project board on GitHub to streamline and automate your workflow.

Sort tasks

Add issues and pull requests to your board and prioritize them alongside note cards containing ideas or task lists.

Plan your project

Sort tasks into columns by status. You can label columns with status indicators like "To Do", "In Progress", and "Done".

Automate your workflow

Set up triggering events to save time on project management—we'll move tasks into the right columns for you.

Track progress

Keep track of everything happening in your project and see exactly what's changed since the last time you looked.

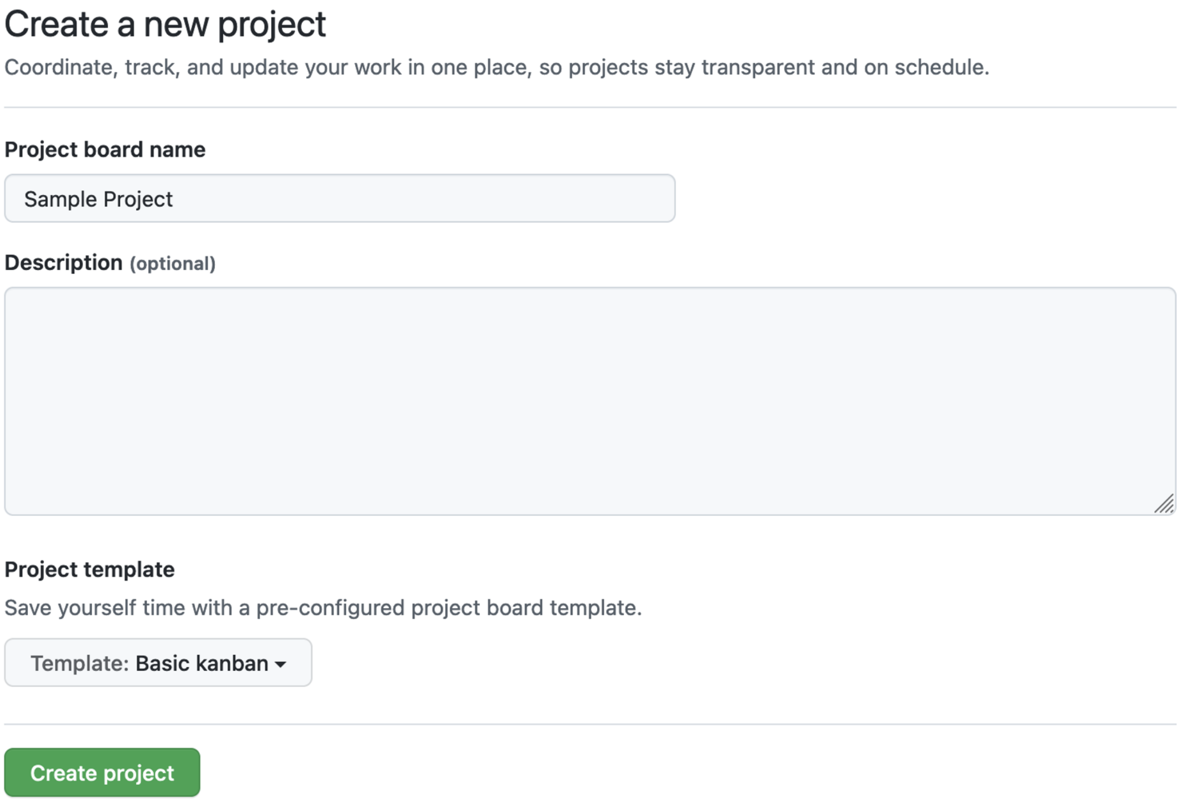
Share status

Each card has a unique URL, making it easy to share and discuss individual tasks with your team.

Wrap up

After you wrap up your work, close your project board to remove it from your active projects list. On to the next project!

GitHub Project



Create a new project

Coordinate, track, and update your work in one place, so projects stay transparent and on schedule.

Project board name

Sample Project

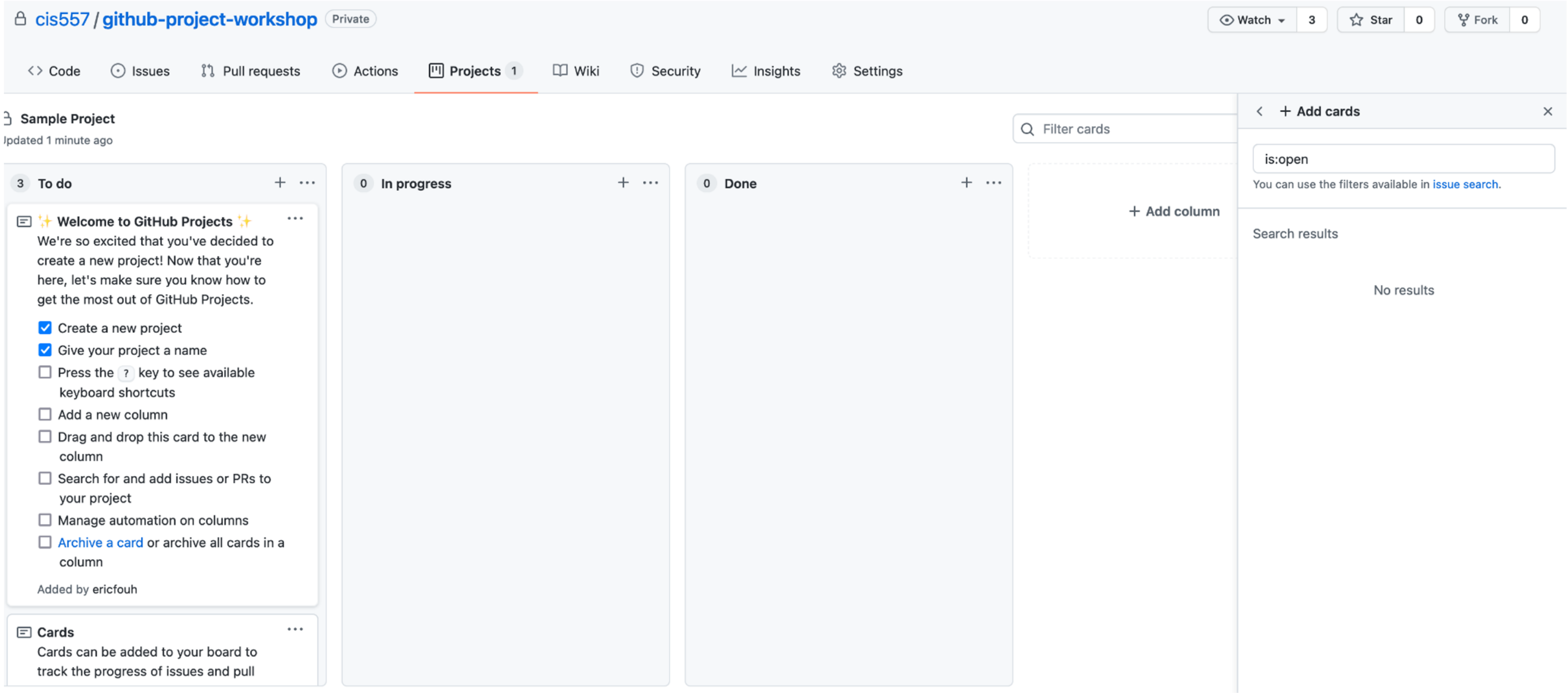
Description (optional)

Project template

Save yourself time with a pre-configured project board template.

Template: **Basic kanban**

GitHub Project Board



Sample Project
Updated 1 minute ago

3 To do

✨ Welcome to GitHub Projects ✨

We're so excited that you've decided to create a new project! Now that you're here, let's make sure you know how to get the most out of GitHub Projects.

- ☑ Create a new project
- ☑ Give your project a name
- □ Press the ? key to see available keyboard shortcuts
- □ Add a new column
- □ Drag and drop this card to the new column
- □ Search for and add issues or PRs to your project
- □ Manage automation on columns
- □ Archive a card or archive all cards in a column

Added by ericfouh

Cards

Cards can be added to your board to track the progress of issues and pull

0 In progress

0 Done

is:open

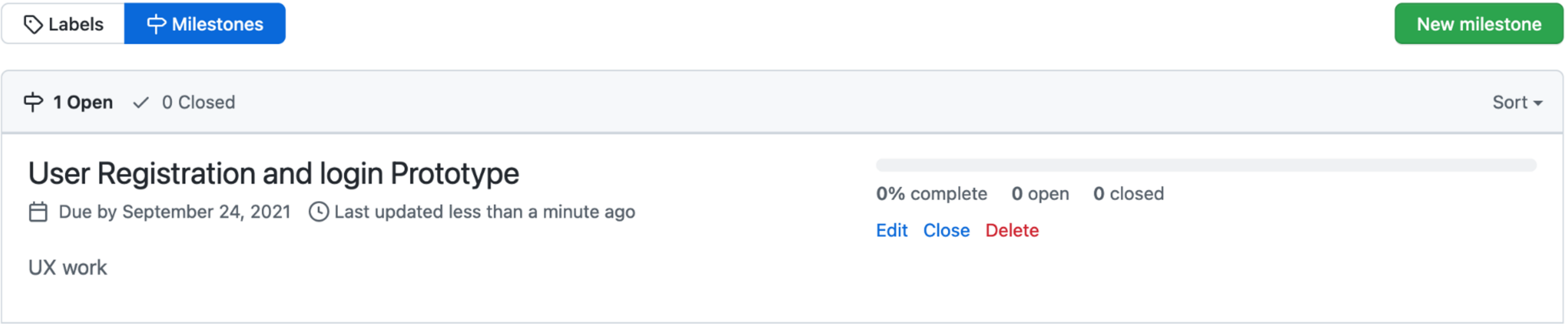
You can use the filters available in issue search.

Search results

No results

Iteration Planning

- Milestones



User Registration and login Prototype

Due by September 24, 2021 Last updated less than a minute ago

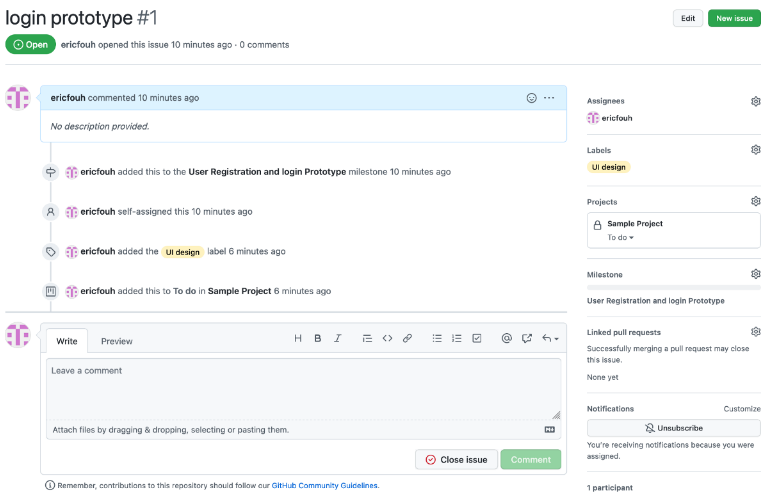
UX work

0% complete 0 open 0 closed

User Stories

- Broken down into tasks called issues in GitHub
- Issues should belong to a milestone
- Issues should have labels
- Assign issues to team members

User Stories



login prototype #1

Open ericfouh opened this issue 10 minutes ago · 0 comments

ericfouh commented 10 minutes ago

No description provided.

ericfouh added this to the **User Registration and login Prototype** milestone 10 minutes ago

ericfouh self-assigned this 10 minutes ago

ericfouh added the UI design label 6 minutes ago

ericfouh added this to To do in Sample Project 6 minutes ago

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Remember, contributions to this repository should follow our GitHub Community Guidelines.

Assignees: ericfouh

Labels: UI design

Projects: Sample Project — To do

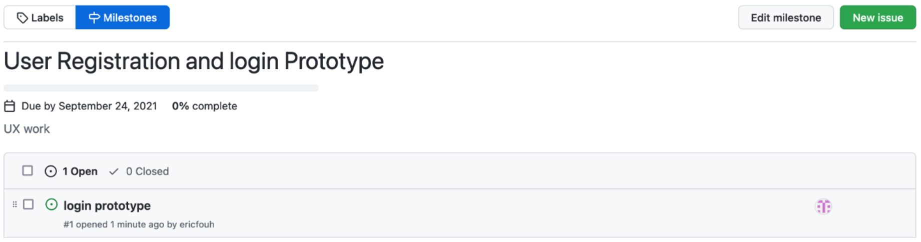
Milestone: User Registration and login Prototype

Linked pull requests: Successfully merging a pull request may close this issue. None yet

Notifications — Customize

You're receiving notifications because you were assigned.

1 participant



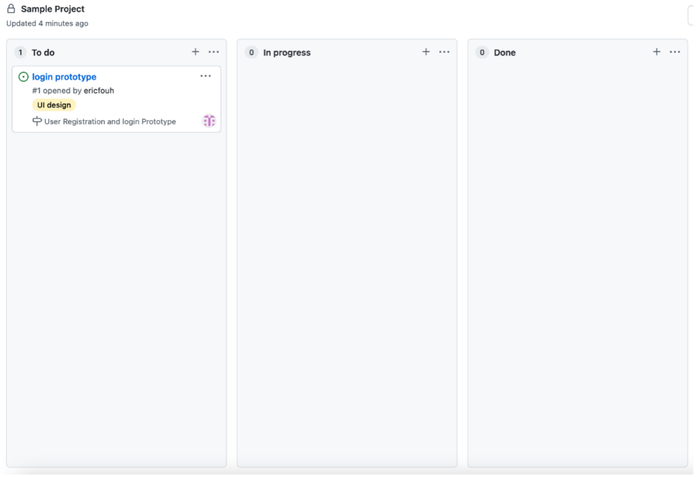
User Registration and login Prototype

Due by September 24, 2021 **0% complete**

UX work

□ 1 Open ✓ 0 Closed

□ **login prototype**
#1 opened 1 minute ago by ericfouh



Sample Project
Updated 4 minutes ago

1 To do	0 In progress	0 Done
login prototype #1 opened by ericfouh UI design User Registration and login Prototype		

Test-Driven Development

- Create a separate test branch
- Use a continuous integration tool to automatically run tests when updates are committed

Pair Programming

- Use pull requests to merge code
- Request code reviews from teammates

Introduction to Github

Installation & Basics

Introduction to Source Control

- Federated management of files and directories
- Track changes over time
- **Recall** previous versions

Introduction to Git

- Created by Linus Torvalds, April 2005
- Cross platform
- Open source & free
- Uses checksums to ensure data integrity

Common Terminology

- "repo" = repository
  - Can contain anything a project needs i.e. folders, files, images, spreadsheets, data, etc.

- "clone" = repository copy on local machine

- "commit" = an atomic change to the repository tracked by Git (Git tracks *changes* not versions)

Git CLI vs. GUI

- GUI makes easy git operations slightly more intuitive but makes complicated issues hard

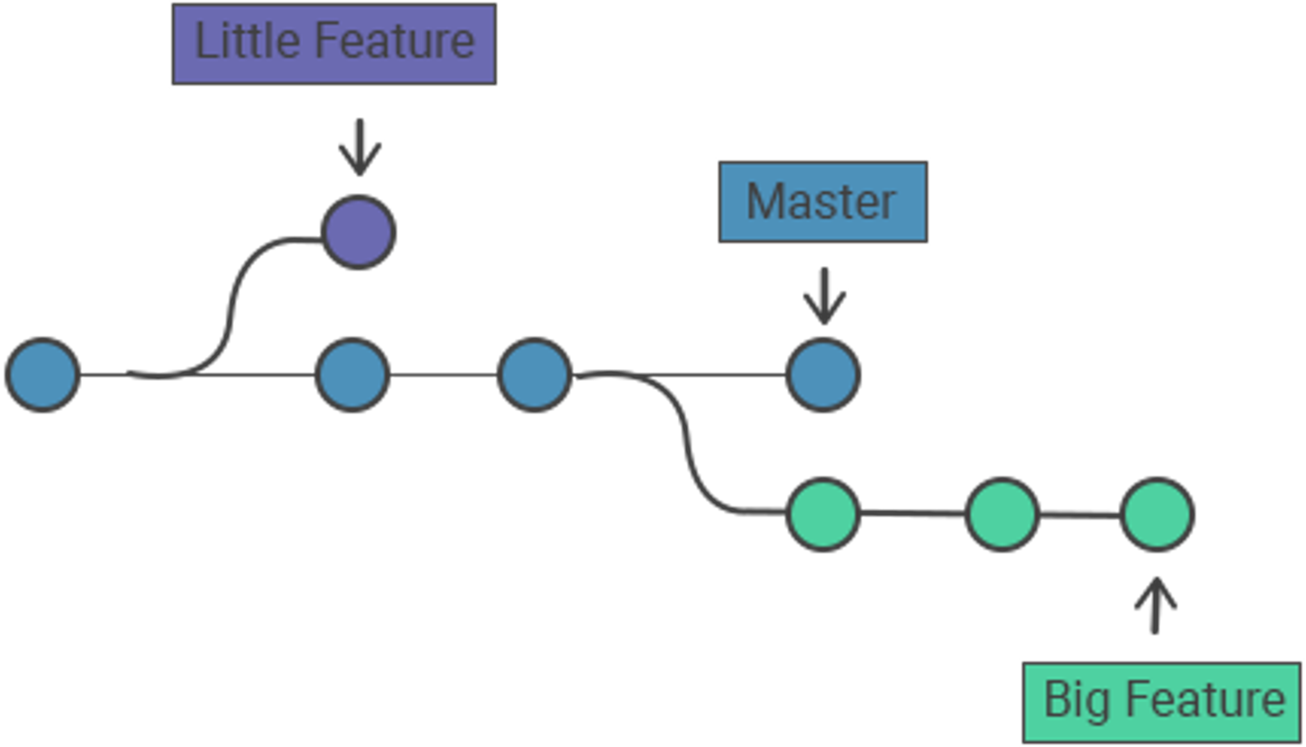
- Most online resources reference Git CLI **not** GUI

- Installation
  - CLI: https://cli.github.com/ or `$brew install gh`
  - GUI: https://desktop.github.com/

Simple Git workflow

- `$mkdir new_project”
  - Make folder named ‘new_project’
- `$cd new_project`
  - Change drive into ‘new_project’
- `$git init`
  - Initialize a git repository in current directory (new_project). `git init` not needed for existing repo
- Make some changes i.e. add new text file
- `$git add .`
  - Add all changes (can replace “.” with filename to add one file) in repository to staging (staging is where all the changes you would like to commit are tracked)
- `$git commit -m “Adds new file”
  - Create a commit with the message “Adds new file” with the contents in staging
- `$git push`
  - Upload all committed changes to remote server (Github)

Git Branching



Git Branching

- Main (master) branch: contains the code that is deployable
- Feature branches: contains code "in progress"
  - Test branch: contains unit tests
  - Development branch: contains code
- Open a pull request with code review when the code is ready to be merged
- Code should only be merged after other team members reviewed it

Git Branching

- `$git checkout -b name_of_branch`
  - Create a new branch with name “name_of_branch”

- Make changes and commit normally

- Push branch for review

A Note about Commit Messages

- Tell it what the commit does (present tense)

- Single line summary

- Keep lines <=100 characters