Gitiquette

Nadia Ahmed

First A Moment of Celebration 🎉

What we have accomplished

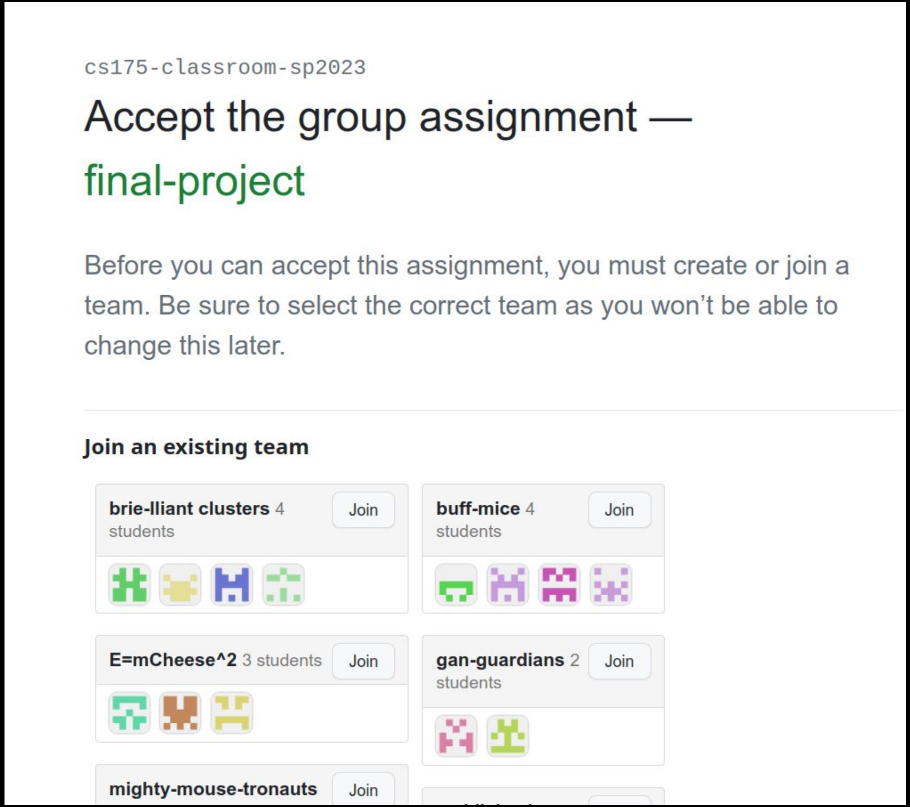
- A Google Ventures Research Design Sprint
- Expert interviews
- Lightning Demo Literature Reviews
- Project Proposal
- Conda environment setup
- Data retrieval utility functions using AWS boto3 API
- PyTorch
  - Custom DataSet
  - Custom Transforms
  - Custom DataLoader
- Test Driven Development
- Reinforcement of Github skills

For the Remainder of the Quarter…

What still needs to be done to succeed in the class?

- Group Project Repository
  - HW to build confidence with baseline models & PyTorch nn.Module
  - Modeling as a group to achieve stretch goal
    - Consistent Pull Requests
    - Status Report (Discord check-ins)
  - Repository Documentation - README.md
- Final Showcase Packet
  - Poster
  - 5 min presentation
  - Tech Memo

Group Project Repository



cs175-classroom-sp2023

Accept the group assignment — final-project

Before you can accept this assignment, you must create or join a team. Be sure to select the correct team as you won't be able to change this later.

Join an existing team

brie-lliant clusters 4 students

buff-mice 4 students

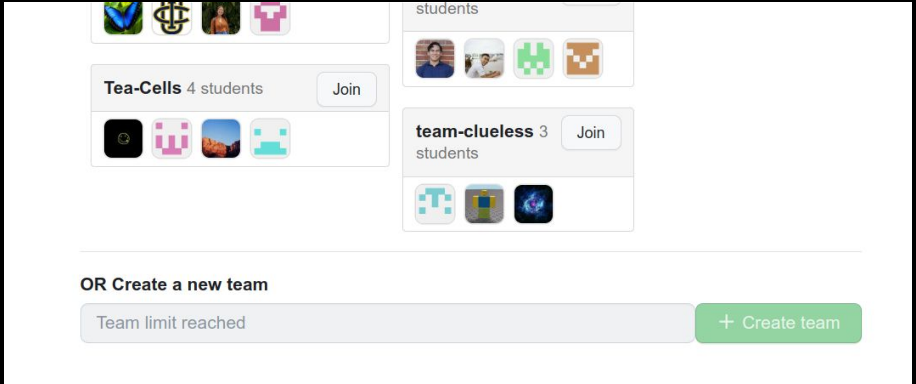
E=mCheese^2 3 students

gan-guardians 2 students

mighty-mouse-tronauts

Step 1: Select Your **Team**

If your team does not exist..



students

Tea-Cells 4 students

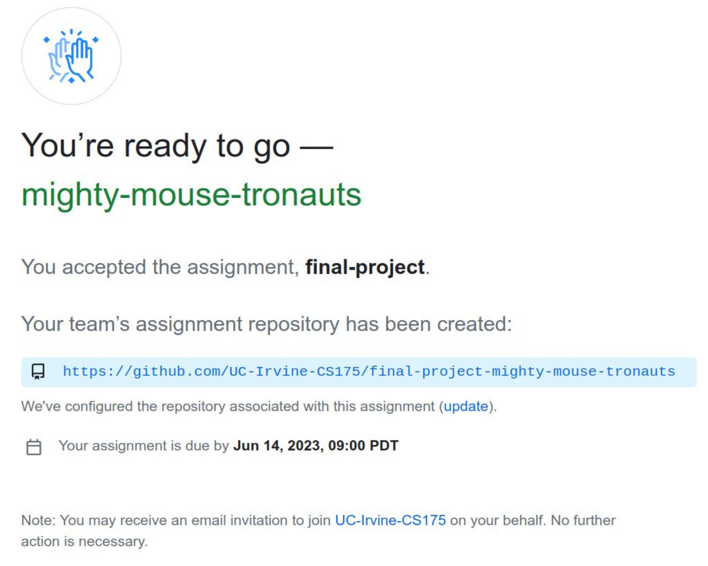
team-clueless 3 students

OR Create a new team

Team limit reached

Create a **new** team.

Clone Team Repository



You're ready to go — mighty-mouse-tronauts

You accepted the assignment, **final-project**.

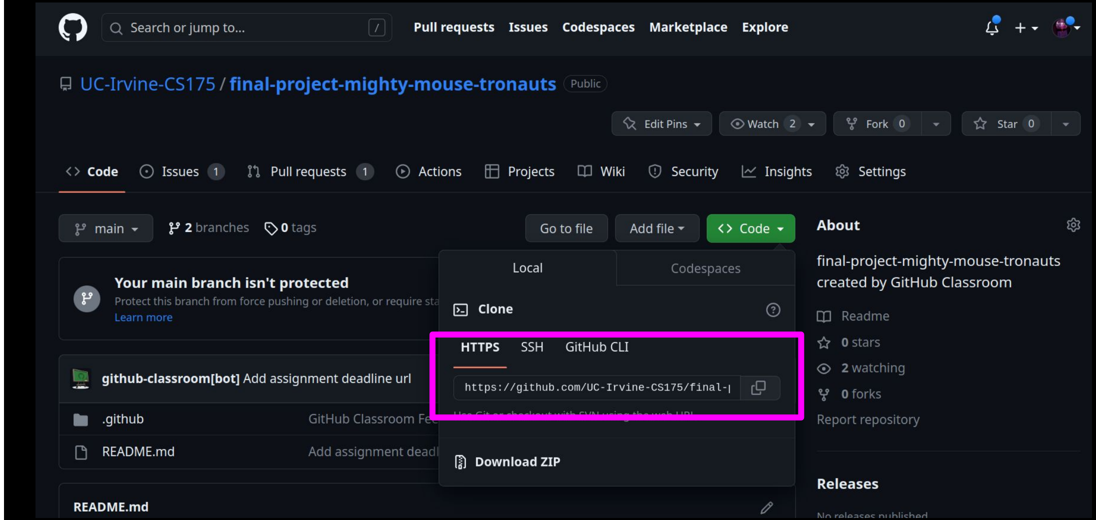
Your team's assignment repository has been created:

https://github.com/UC-Irvine-CS175/final-project-mighty-mouse-tronauts

We've configured the repository associated with this assignment (update).

Your assignment is due by **Jun 14, 2023, 09:00 PDT**

Note: You may receive an email invitation to join UC-Irvine-CS175 on your behalf. No further action is necessary.



UC-Irvine-CS175 / **final-project-mighty-mouse-tronauts** Public

Your main branch isn't protected

Protect this branch from force pushing or deletion, or require sta

Learn more

`https://github.com/UC-Irvine-CS175/final-|`

github-classroom[bot] Add assignment deadline url

.github	GitHub Classroom Fe
README.md	Add assignment deadl

README.md

About

final-project-mighty-mouse-tronauts created by GitHub Classroom

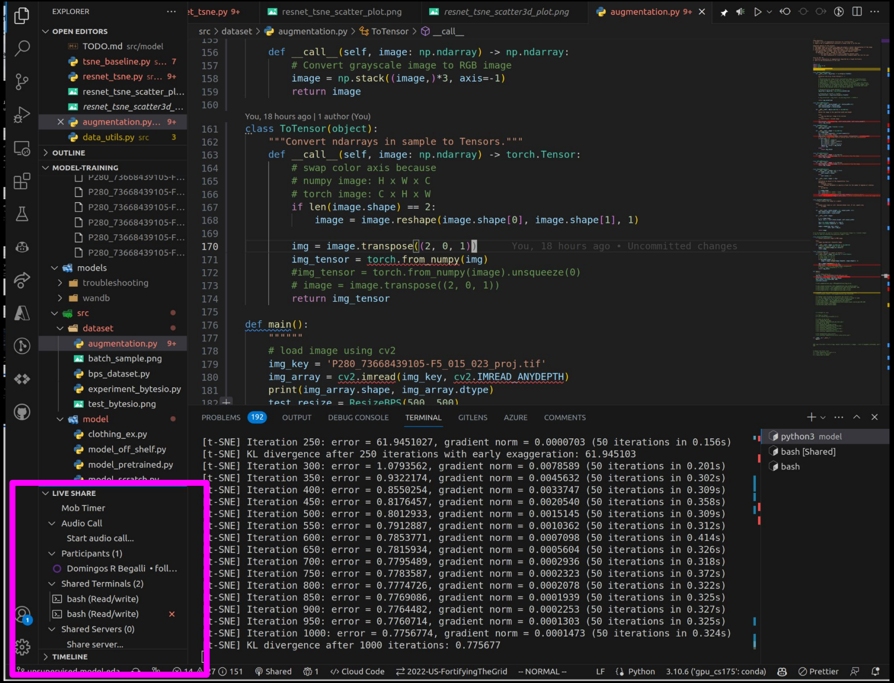
Copy the url and in your terminal type the following to clone:

```
git clone <your_project_url>
```

Working With The TEAM

Visual Studio Code LiveShare is awesome!

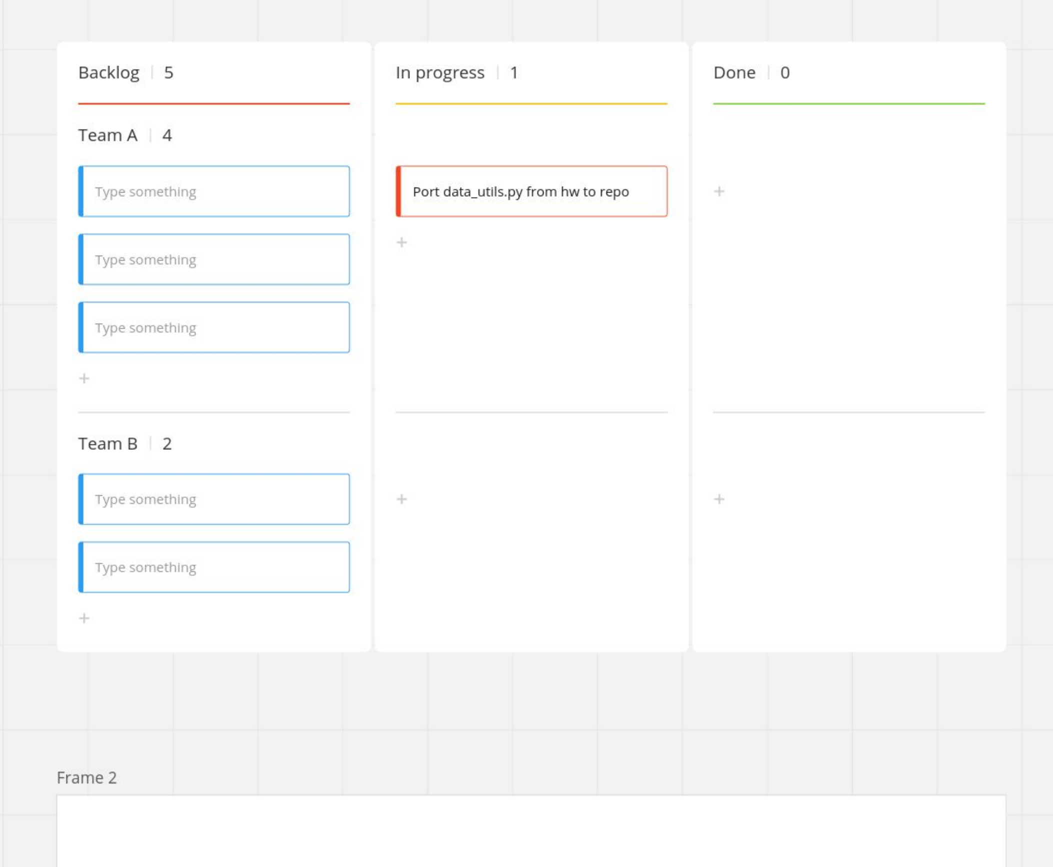
- Code in real-time with your pair programmer or the entire group
- mobti.me to use best ensemble coding methods and patterns to build team morale and flow state
- track progress/plan with Miro Team Kanban
- Utilize team discord channel



```python
    def __call__(self, image: np.ndarray) -> np.ndarray:
        # Convert grayscale image to RGB image
        image = np.stack((image,)*3, axis=-1)
        return image

You, 18 hours ago | 1 author (You)
class ToTensor(object):
    """Convert ndarrays in sample to Tensors."""
    def __call__(self, image: np.ndarray) -> torch.Tensor:
        # swap color axis because
        # numpy image: H x W x C
        # torch image: C x H x W
        if len(image.shape) == 2:
            image = image.reshape(image.shape[0], image.shape[1], 1)

        img = image.transpose((2, 0, 1))      You, 18 hours ago • Uncommitted changes
        img_tensor = torch.from_numpy(img)
        #img_tensor = torch.from_numpy(image).unsqueeze(0)
        # image = image.transpose((2, 0, 1))
        return img_tensor

def main():
    """"""
    # load image using cv2
    img_key = 'P280_73668439105-F5_015_023_proj.tif'
    img_array = cv2.imread(img_key, cv2.IMREAD_ANYDEPTH)
    print(img_array.shape, img_array.dtype)
    test_resize = ResizeBPS(500, 500)
```

```
[t-SNE] Iteration 250: error = 61.9451027, gradient norm = 0.0000703 (50 iterations in 0.156s)
[t-SNE] KL divergence after 250 iterations with early exaggeration: 61.945103
[t-SNE] Iteration 300: error = 1.0793562, gradient norm = 0.0078589 (50 iterations in 0.201s)
[t-SNE] Iteration 350: error = 0.9322174, gradient norm = 0.0045632 (50 iterations in 0.302s)
t-SNE] Iteration 400: error = 0.8550254, gradient norm = 0.0033747 (50 iterations in 0.309s)
t-SNE] Iteration 450: error = 0.8176457, gradient norm = 0.0020540 (50 iterations in 0.358s)
t-SNE] Iteration 500: error = 0.8012933, gradient norm = 0.0015145 (50 iterations in 0.309s)
t-SNE] Iteration 550: error = 0.7912887, gradient norm = 0.0010362 (50 iterations in 0.312s)
t-SNE] Iteration 600: error = 0.7853771, gradient norm = 0.0007098 (50 iterations in 0.414s)
t-SNE] Iteration 650: error = 0.7815934, gradient norm = 0.0005604 (50 iterations in 0.326s)
t-SNE] Iteration 700: error = 0.7795489, gradient norm = 0.0002936 (50 iterations in 0.318s)
t-SNE] Iteration 750: error = 0.7783587, gradient norm = 0.0002323 (50 iterations in 0.372s)
t-SNE] Iteration 800: error = 0.7774726, gradient norm = 0.0002078 (50 iterations in 0.322s)
t-SNE] Iteration 850: error = 0.7769086, gradient norm = 0.0001939 (50 iterations in 0.325s)
t-SNE] Iteration 900: error = 0.7764482, gradient norm = 0.0002253 (50 iterations in 0.327s)
t-SNE] Iteration 950: error = 0.7760714, gradient norm = 0.0001303 (50 iterations in 0.325s)
t-SNE] Iteration 1000: error = 0.7756774, gradient norm = 0.0001473 (50 iterations in 0.324s)
t-SNE] KL divergence after 1000 iterations: 0.775677
```

Miro Team Kanbans



Backlog 5	In progress 1	Done 0
Team A 4		
Type something	Port data_utils.py from hw to repo	+
Type something	+	
Type something		
+		
Team B 2		
Type something	+	+
Type something		
+		

Frame 2

Can issue tickets and keep track of small tasks–the smaller the better and the more rewarding it will be when you finally finish and get to drag and drop to done.

You also have an empty frame to ideate on as a team.

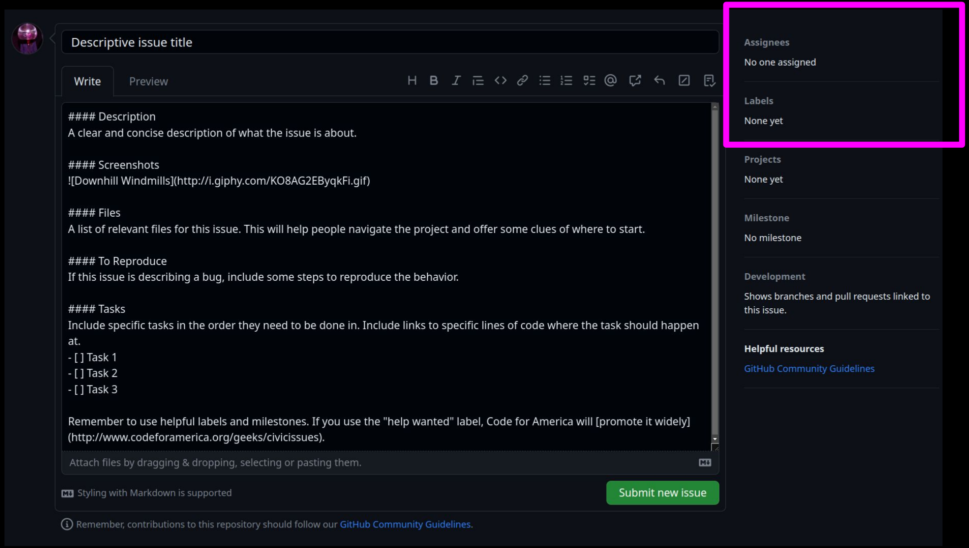
The more activity and use of tools the more you will communicate with your team members and alleviate task planning.

Suggested Workflow: Branching For each Feature and Merging

- Open an issue with the task that needs to be completed and tag reviewers
  - team members
  - course staff
- Create a branch for the issue (best practices here)
  - github GUI
  - manually: example:

```
wip-02-s3-update-data-utils-data-download
```

This name indicates that the task of updating data utility code for downloading from AWS s3 bucket related to issue 02 is a work in progress.



Descriptive issue title

```
#### Description
A clear and concise description of what the issue is about.

#### Screenshots
![Downhill Windmills](http://i.giphy.com/KO8AG2EByqkFi.gif)

#### Files
A list of relevant files for this issue. This will help people navigate the project and offer some clues of where to start.

#### To Reproduce
If this issue is describing a bug, include some steps to reproduce the behavior.

#### Tasks
Include specific tasks in the order they need to be done in. Include links to specific lines of code where the task should happen at.
- [] Task 1
- [] Task 2
- [] Task 3

Remember to use helpful labels and milestones. If you use the "help wanted" label, Code for America will [promote it widely](http://www.codeforamerica.org/geeks/civicissues).
```

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

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

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Shows branches and pull requests linked to this issue.

Helpful resources
GitHub Community Guidelines

More on Branch Naming

Start Name with Category Word

It is recommended to begin the name of a branch with a category word, which indicates the type of task that is being solved with that branch. Some of the most used category words are:

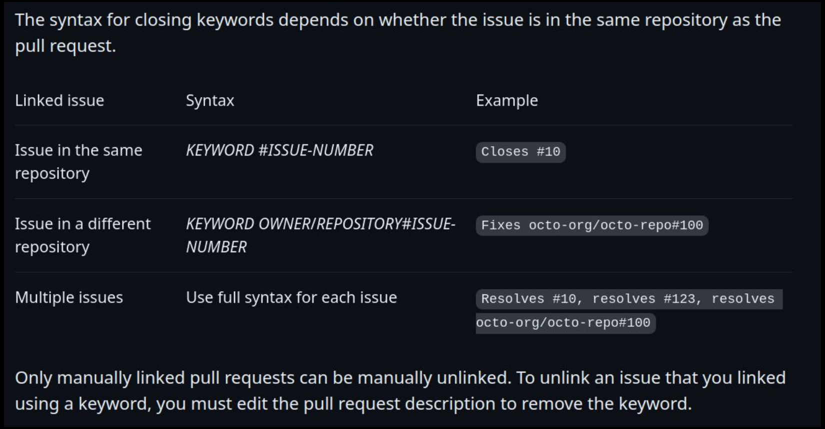
Category Word and Meaning

- hotfix: for quickly fixing critical issues, usually with a temporary solution
- bugfix: for fixing a bug
- feature: for adding, removing or modifying a feature
- test: for experimenting something which is not an issue
- wip: for a work in progress

*source: Best Practices for Naming Branches

Closing the Issue with a Pull Request

- Pull Request is a request to merge your branch to the main branch
  - Best practice for teams: branch and merge often
  - Synchronize pull requests with code reviews
  - Take advantage of the Teaching Assistants' feedback
- Link your pull requests to the issue the merge will address:
  - Use a github supported keyword in the pull request's description or in a commit message to automate
    - close
    - closes
    - closed
    - fix
    - fixes
    - fixed
    - resolve
    - resolves
    - resolved



The syntax for closing keywords depends on whether the issue is in the same repository as the pull request.

Linked issue	Syntax	Example
Issue in the same repository	*KEYWORD #ISSUE-NUMBER*	`Closes #10`
Issue in a different repository	*KEYWORD OWNER/REPOSITORY#ISSUE-NUMBER*	`Fixes octo-org/octo-repo#100`
Multiple issues	Use full syntax for each issue	`Resolves #10, resolves #123, resolves octo-org/octo-repo#100`

Only manually linked pull requests can be manually unlinked. To unlink an issue that you linked using a keyword, you must edit the pull request description to remove the keyword.

  - Link it manually

Pull requests.

*source: github

Pull Request Examples

example 1: descriptive

```
# Description

Please include a summary of the changes and the related issue. Please also include relevant motivation and
context. List any dependencies that are required for this change.

Fixes # (issue)

## Type of change

Please delete options that are not relevant.

- [ ] Bug fix (non-breaking change which fixes an issue)
- [ ] New feature (non-breaking change which adds functionality)
- [ ] Breaking change (fix or feature that would cause existing functionality to not work as expected)
- [ ] This change requires a documentation update

# How Has This Been Tested?

Please describe the tests that you ran to verify your changes. Provide instructions so we can reproduce.
Please also list any relevant details for your test configuration

- [ ] Test A
- [ ] Test B

**Test Configuration**:
* Firmware version:
* Hardware:
* Toolchain:
* SDK:

# Checklist:

- [ ] My code follows the style guidelines of this project
- [ ] I have performed a self-review of my code
- [ ] I have commented my code, particularly in hard-to-understand areas
- [ ] I have made corresponding changes to the documentation
- [ ] My changes generate no new warnings
- [ ] I have added tests that prove my fix is effective or that my feature works
- [ ] New and existing unit tests pass locally with my changes
- [ ] Any dependent changes have been merged and published in downstream modules
```

example 2: minimal

```
## Describe your changes

## Issue ticket number and link

## Checklist before requesting a review
- [ ] I have performed a self-review of my code
- [ ] If it is a core feature, I have added thorough tests.
- [ ] Do we need to implement analytics?
- [ ] Will this be part of a product update? If yes, please write one phrase about this update.
```

example 3: super informative–standard of practice

```
## Scope

<!-- Brief description of WHAT you're doing and WHY. -->

[closes TICKET-###](https://link-to-your-ticket)

## Implementation

With @pair

<!--

Some description of HOW you achieved it. Perhaps give a high level description of the program flow. Did
you need to refactor something? What tradeoffs did you take? Are there things in here which you'd
particularly like people to pay close attention to?

-->

## Screenshots

|         | before | after |
| ------- | ------ | ----- |
| desktop |        |       |
| mobile  |        |       |

## How to Test

<!--

A straightforward scenario of how to test your changes could help colleagues that are not familiar with
the part of the code that you are changing but want to see it in action. This section can include a
description or step-by-step instructions of how to get to the state of v2 that your change affects.

A "How To Test" section can look something like this:

- Sign in with a user with tracks
- Activate `show_awesome_cat_gifs` feature (add `?feature.show_awesome_cat_gifs=1` to your URL)
- You should see a GIF with cats dancing

-->

## Emoji Guide

**For reviewers: Emojis can be added to comments to call out blocking versus non-blocking feedback.**

E.g: Praise, minor suggestions, or clarifying questions that don't block merging the PR.

> 🟢 Nice refactor!

> 🟡 Why was the default value removed?

E.g: Blocking feedback must be addressed before merging.

> 🔴 This change will break something important

| | | |
| --- | --- | --- |
| Blocking | 🔴 ❌ 🚨 | RED |
| Non-blocking | 🟡 💡 🤔 💭 | Yellow, thinking, etc |
| Praise | 🟢 💚 😍 👍 🙌 | Green, hearts, positive emojis, etc |
```

*source: axolo, soundcloud