**\*my recommended git workflow:\***

0. \*Make sure you are currently on your master branch\*

1. pull down latest master branch:

`git pull origin master`

2. check out your own working branch:

`git checkout -b my-new-branch`

3. when you’re done working, pull down latest version of master

(others may have made changes while you were working)

`git pull origin master`

4. merge the newly updated local version of master into your working branch:

(while your current branch is `my-new-branch`!)

`git merge master`

5. resolve merge conflicts if necessary

6. push your working branch to github

`git push origin my-new-branch`

7. and finally create a pull request on github to merge `my-new-branch` into master (edited)

**Example Branch:**

git checkout -b *fontbranch*

git pull origin master (based on if you want to pull latest)

git merge master (based on if you want to pull latest)

git push origin *fontbranch*

git add .

git commit -m "new fonts"

git push

git push --set-upstream origin fontbranch

**# Group Do: Designing your MVP**

## Instructions

\* With your group, answer each of the following questions:

1. Who is your target audience?

2. What is the problem that the product will address?

3. What is the primary goal of the product?

4. Identify and prioritize essential user stories (limit this to 3 or fewer)?

\* Remember, your MVP should be the absolute \*minimum\* product that you need to pitch your idea to investors (or your classmates). Think bare essentials.

\* As an example, the essential features of an instagram-like application might look like:

1. As a user, I want to post pictures so that I can share my life.

2. As a user, I want to be able to search for other users so that I can find their list of images.

3. As a user, I want to be able to view other people’s images so that I can keep up with what’s happening in their lives.

\* Things like liking photos, sending photos to friends, making instagram stories, etc are cool, but they aren’t essential. They should \*\*not\*\* be part of the MVP.

**Creating Issues**

## Instructions

\* Start by listing the tasks involved in completing each of your user stories.

\* Once you have 1-5 tasks written up for each user story, open a GitHub issue for each task by clicking on `Issues` and the green `New Issue` button.

Use the template below, replacing the user stories and acceptance criteria with your own.

```## User Story

As a user, I want to be able to input search queries so that I can search Stack Overflow when I'm stuck.

## Acceptance Criteria

- [ ] index.html has a text input

- [ ] index.html has a search button```

\* \*\*This is a critical task.\*\* Don’t be afraid to ask instructional staff for help during this part. Make sure that each task is clear and everyone on the team understands how to implement each one.

As an example:

In a dating website application...

\*\*User Story\*\*: As a user I want to see my closest match, so that I can meet someone with whom I share interests.

\* \*\*BAD TASK\*\*:

1. On new member form submit, find and display best match.

\* \*\*GOOD TASKS\*\*:

1. build a form that captures user scores (1-5) to ten questions.

2. On submit, save the ten numerical answers to as an array.

3. Compare the user array to each existing user array and find the closest match.

4. Display the closest match to the user.

**Stand-ups**

\* In this activity you will take part in your first stand-up with your group mates. This will make sure everyone is on the same page with what they are working on and everyone gets help they might need.

## Instructions

\* Before your first stand-up, each member of the team should go through the To Do column of the Project Board and self-assign at least one issue.

\* To do this, click on the issue and on the right choose “self-assign”.

\* Everyone should have a clear idea of what they intend to accomplish today. Now it’s time for your first stand-up!

\* Stand if you are able - yep, stand-ups take place standing; this helps to ensure that the meetings are short and to the point.

\* Each member of the team should say what they did yesterday, what they plan to do today, and what, if anything, is blocking their progress.

\* Stand-ups should be held \*\*daily\*\* from this point forward - yes even days that you don’t have class (use Slack).