CIS*3750 Lab 2

REPOSITORY SETUP AND USER STORIES

SUNDAY, SEPTEMBER 21, 2025

Lab Objectives

- What we’ll do today:
  - **Finalize Teams**: Officially decide on your team name (members should already be assigned) □ Any issues? Bring them up to the TA **after the presentation.**
    - It lets us ensure the lab curriculum is covered in time, and we can focus our full attention on resolving issues afterwards.
  - **Set Up Your Project**: Create a shared team repository using GitHub Classrooms.
  - **Brainstorm**: Define requirements and collaborators for your Intelligent Tutoring System (ITS) project.
  - **Write User Stories**: Document your brainstormed ideas in the standard user story format.
  - **Document Your Work**: Commit your team information and user stories to your new GitHub repository.

Part I – Team and Repository Setup

- Step 1: Pick a Team Name
  - This is the name your team will use for the whole semester.
  - Choose something **unique** and **professional**
- Step 2: Designate One Person
  - ***ONLY ONE PERSON*** from your team should be the first to click the assignment link.
  - This person will be responsible for creating the team on GitHub. **Everyone else** will join it.
  - ****IF YOU CREATE SEPARATE TEAMS, YOU WILL BE UNABLE TO JOIN EACH OTHER. SO PLEASE FOLLOW THE INSTRUCTIONS****

GitHub Classroom – First Team Member

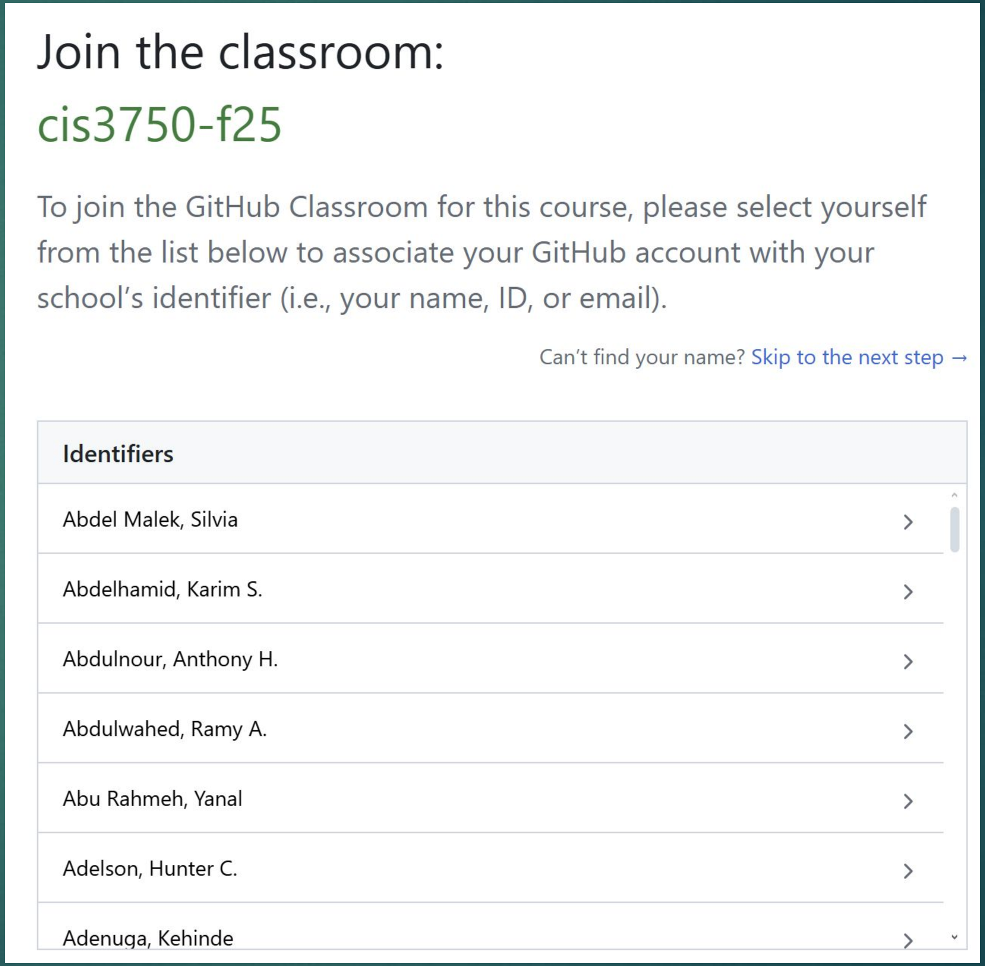
- Action: The ***designated member*** clicks the link
  - Navigate to the GitHub Classroom assignment link: https://classroom.github.com/a/JWjp7sah
  - You will be prompted to create a new team

Navigation (1/4)

Follow the instructions on the link.

Locate your identifier (i.e., name) and, if required, sign into your GitHub account.

**** If you don't have a GitHub account yet, I suggest waiting until after the presentation is finished so I can help out.**



Join the classroom:

cis3750-f25

To join the GitHub Classroom for this course, please select yourself from the list below to associate your GitHub account with your school's identifier (i.e., your name, ID, or email).

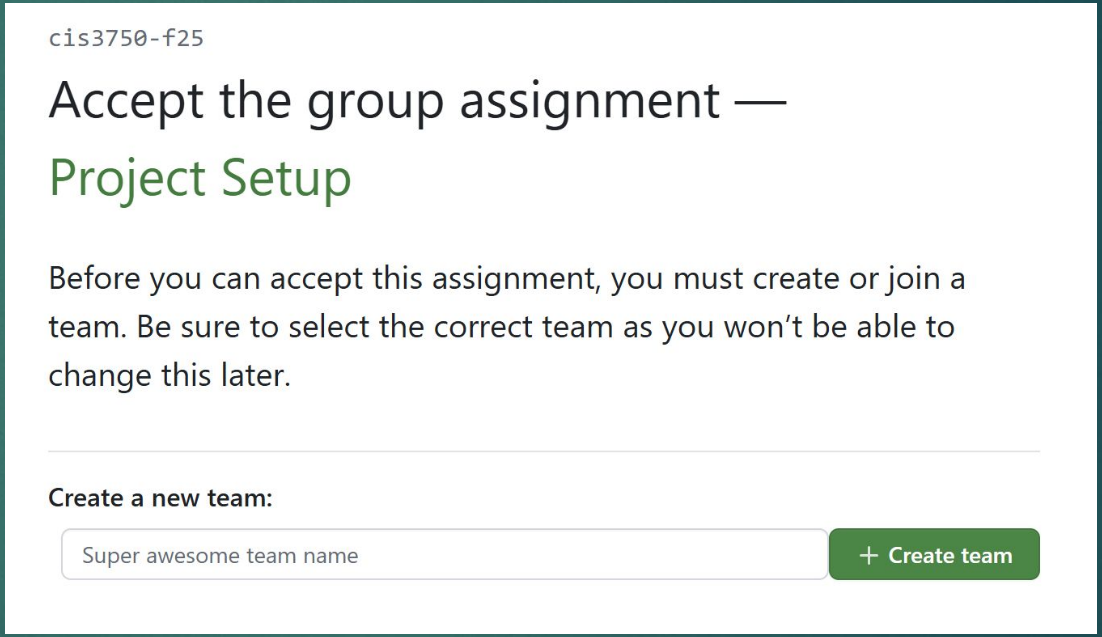
Can't find your name? Skip to the next step →

Identifiers
Abdel Malek, Silvia
Abdelhamid, Karim S.
Abdulnour, Anthony H.
Abdulwahed, Ramy A.
Abu Rahmeh, Yanal
Adelson, Hunter C.
Adenuga, Kehinde

Navigation (2/4)

The designated member will enter the exact team name decided on in Step 1.

GitHub will then create a new repository for your team. This may take a minute or two.



cis3750-f25

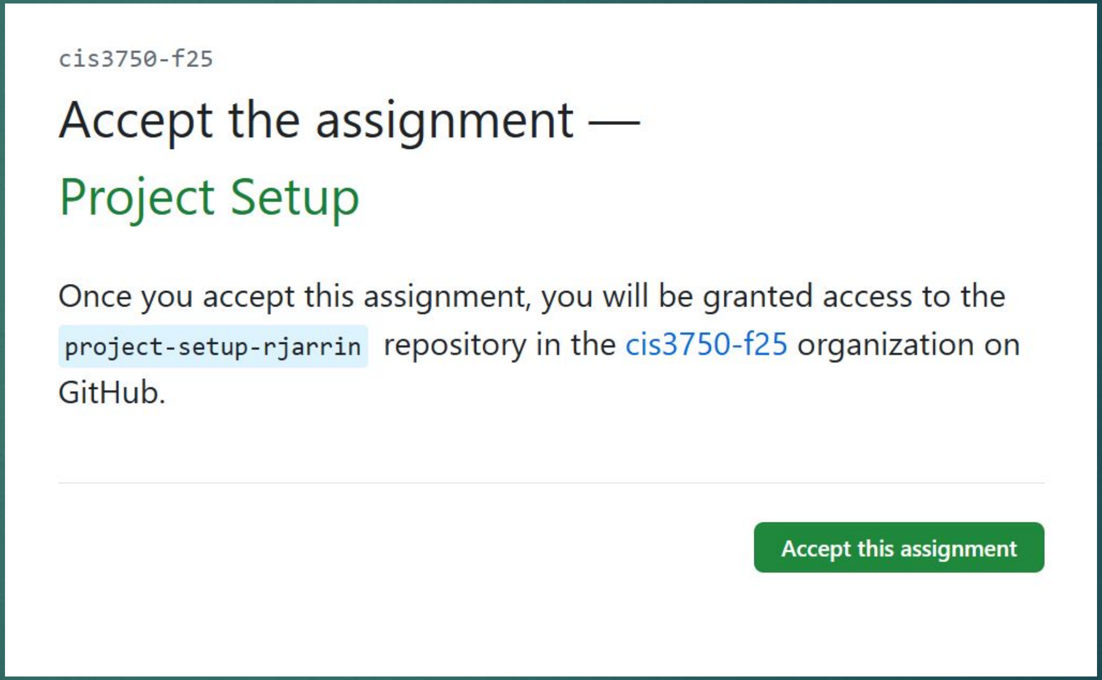
Accept the group assignment — Project Setup

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.

Create a new team:

Navigation (3/4)

You'll gain access to the repo after accepting the assignment. Simply click on the green 'Accept this assignment' button.



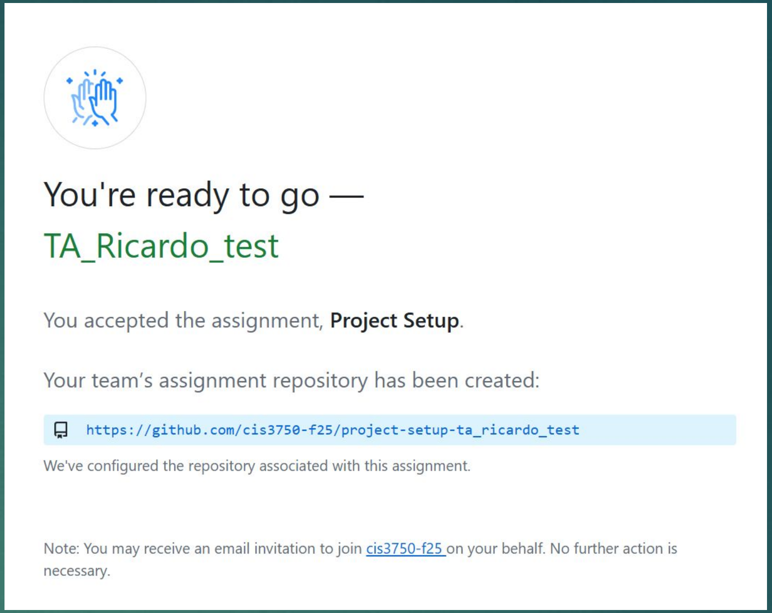
cis3750-f25

Accept the assignment — Project Setup

Once you accept this assignment, you will be granted access to the `project-setup-rjarrin` repository in the cis3750-f25 organization on GitHub.

Navigation (4/4)

You're all set! Click on the link to be redirected to your GitHub repo.



You're ready to go — TA_Ricardo_test

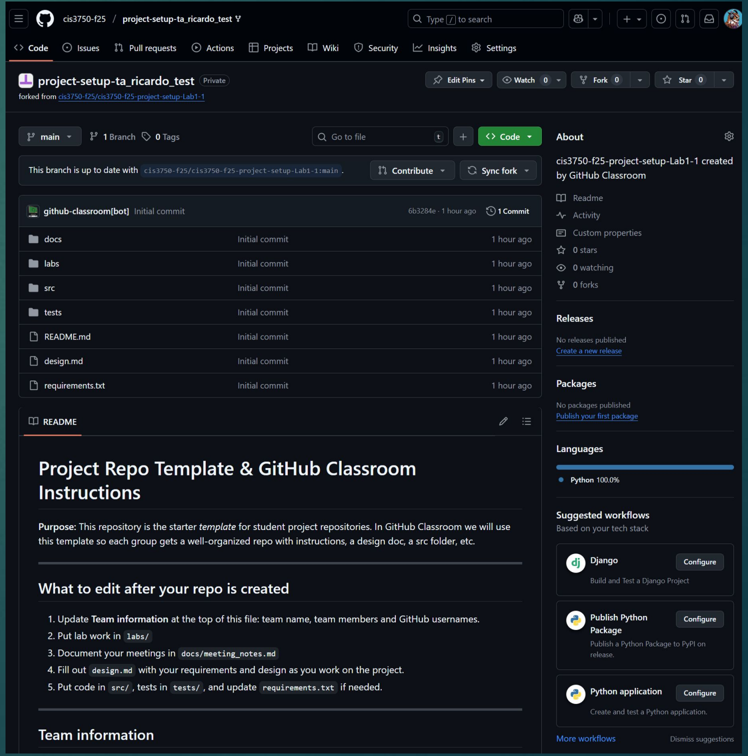
You accepted the assignment, **Project Setup**.

Your team's assignment repository has been created:

`https://github.com/cis3750-f25/project-setup-ta_ricardo_test`

We've configured the repository associated with this assignment.

Note: You may receive an email invitation to join cis3750-f25 on your behalf. No further action is necessary.



project-setup-ta_ricardo_test Private
forked from cis3750-f25/cis3750-f25-project-setup-Lab1-1

This branch is up to date with `cis3750-f25/cis3750-f25-project-setup-Lab1-1:main`.

github-classroom[bot] Initial commit 6b3284e · 1 hour ago 1 Commit

docs	Initial commit	1 hour ago
labs	Initial commit	1 hour ago
src	Initial commit	1 hour ago
tests	Initial commit	1 hour ago
README.md	Initial commit	1 hour ago
design.md	Initial commit	1 hour ago
requirements.txt	Initial commit	1 hour ago

Project Repo Template & GitHub Classroom Instructions

Purpose: This repository is the starter *template* for student project repositories. In GitHub Classroom we will use this template so each group gets a well-organized repo with instructions, a design doc, a src folder, etc.

What to edit after your repo is created

1. Update **Team information** at the top of this file: team name, team members and GitHub usernames.
2. Put lab work in `labs/`
3. Document your meetings in `docs/meeting_notes.md`
4. Fill out `design.md` with your requirements and design as you work on the project.
5. Put code in `src/`, tests in `tests/`, and update `requirements.txt` if needed.

Team information

About

cis3750-f25-project-setup-Lab1-1 created by GitHub Classroom

Readme
Activity
Custom properties
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Python 100.0%

Suggested workflows
Based on your tech stack

Django — Build and Test a Django Project

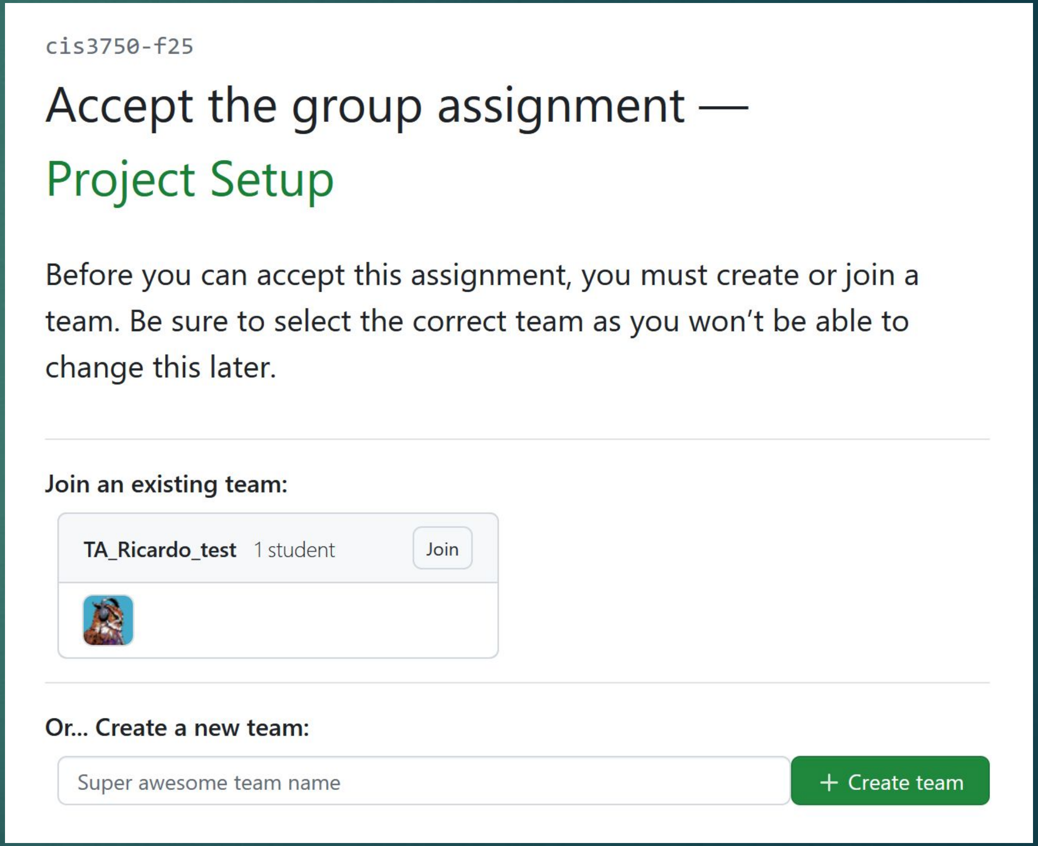
Publish Python Package — Publish a Python Package to PyPI on release.

Python application — Create and test a Python application.

More workflows
Dismiss suggestions

GitHub Classroom – The Rest of the Team

- Action: **All other members** join the team
  - Once the first person has created the team, the rest of the members should click the same assignment link
  - Find and select your identifier
  - Instead of just seeing the "Create a team" option, you should now see your team's name in a list.
  - Join your team and you will all have access to the same shared repository


cis3750-f25

Accept the group assignment — Project Setup

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.

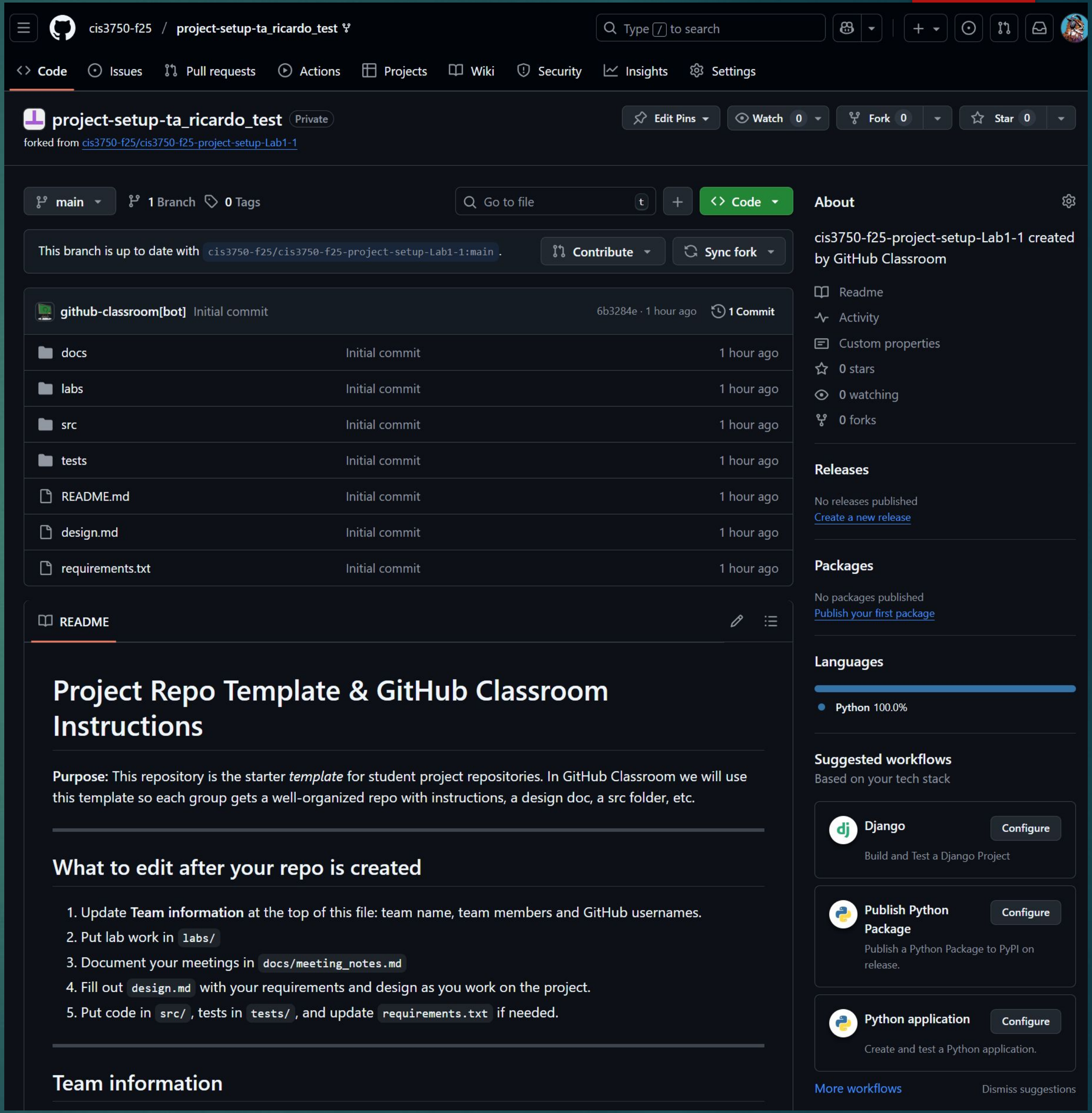
Join an existing team:

TA_Ricardo_test 1 student

Or... Create a new team:

Navigating Your Repository

- Let's get familiar with the repository page
  - **Code Tab**: This is the main page for showing all your files and folders
  - **Files and Folders List**: You'll see folders like docs, labs, and src, and files like README.md
  - **Edit Button (✎)**: This is how you edit a file directly in your browser (*but we'll look at how to edit locally via Git*)



project-setup-ta_ricardo_test Private

forked from cis3750-f25/cis3750-f25-project-setup-Lab1-1

This branch is up to date with cis3750-f25/cis3750-f25-project-setup-Lab1-1:main .

github-classroom[bot] Initial commit 6b3284e · 1 hour ago 1 Commit

docs	Initial commit	1 hour ago
labs	Initial commit	1 hour ago
src	Initial commit	1 hour ago
tests	Initial commit	1 hour ago
README.md	Initial commit	1 hour ago
design.md	Initial commit	1 hour ago
requirements.txt	Initial commit	1 hour ago

README

Project Repo Template & GitHub Classroom Instructions

Purpose: This repository is the starter *template* for student project repositories. In GitHub Classroom we will use this template so each group gets a well-organized repo with instructions, a design doc, a src folder, etc.

What to edit after your repo is created

1. Update **Team information** at the top of this file: team name, team members and GitHub usernames.
2. Put lab work in `labs/`
3. Document your meetings in `docs/meeting_notes.md`
4. Fill out `design.md` with your requirements and design as you work on the project.
5. Put code in `src/`, tests in `tests/`, and update `requirements.txt` if needed.

Team information

About

cis3750-f25-project-setup-Lab1-1 created by GitHub Classroom

Readme
Activity
Custom properties
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Python 100.0%

Suggested workflows

Based on your tech stack

Django — Configure
Build and Test a Django Project

Publish Python Package — Configure
Publish a Python Package to PyPI on release.

Python application — Configure
Create and test a Python application.

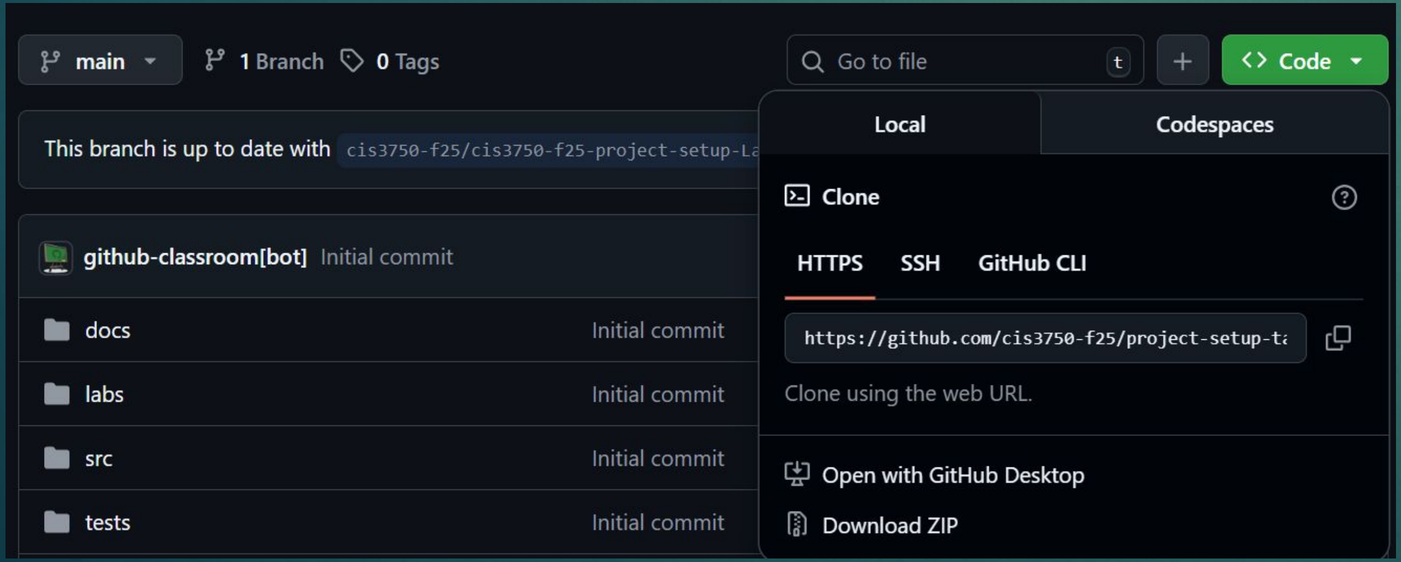
More workflows
Dismiss suggestions

Working Locally

- This is the standard workflow. You should get comfortable with the command line.
- Why?
  - Use your own code editor
  - Work offline
  - Learn/emulate how professional developers work better to do this now in school where teaching staff can help you
- What you'll need:
  - Git installed and a terminal/command prompt
  - You can see if git is already installed by running 'git' in the prompt
    - If you get a help message, Git is already installed

Getting Started (1/3)

- Clone (Copy the repo to your PC):
  - On GitHub, click the green <> Code button, and copy the HTTPS URL
  - In your terminal, run: git clone [paste-the-URL-here]
  - *You can also use SSH, but …*



This branch is up to date with `cis3750-f25/cis3750-f25-project-setup-La`

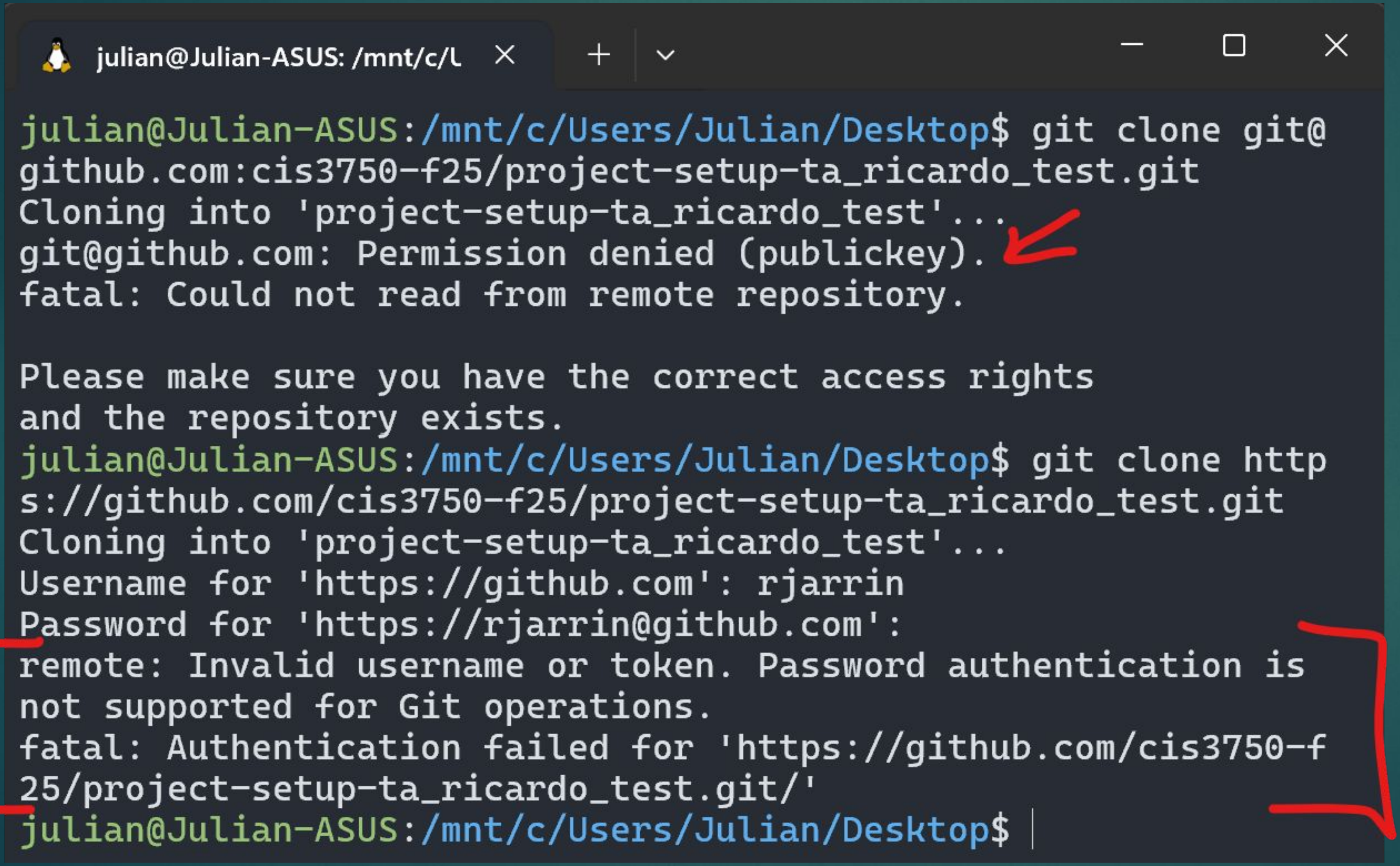
github-classroom[bot] Initial commit

docs	Initial commit
labs	Initial commit
src	Initial commit
tests	Initial commit

`https://github.com/cis3750-f25/project-setup-ta`

Clone using the web URL.

But wait! Something went wrong!



```
julian@Julian-ASUS:/mnt/c/Users/Julian/Desktop$ git clone git@
github.com:cis3750-f25/project-setup-ta_ricardo_test.git
Cloning into 'project-setup-ta_ricardo_test'...
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
julian@Julian-ASUS:/mnt/c/Users/Julian/Desktop$ git clone http
s://github.com/cis3750-f25/project-setup-ta_ricardo_test.git
Cloning into 'project-setup-ta_ricardo_test'...
Username for 'https://github.com': rjarrin
Password for 'https://rjarrin@github.com':
remote: Invalid username or token. Password authentication is
not supported for Git operations.
fatal: Authentication failed for 'https://github.com/cis3750-f
25/project-setup-ta_ricardo_test.git/'
julian@Julian-ASUS:/mnt/c/Users/Julian/Desktop$
```

<u>Using SSH method:</u>
Permissions denied

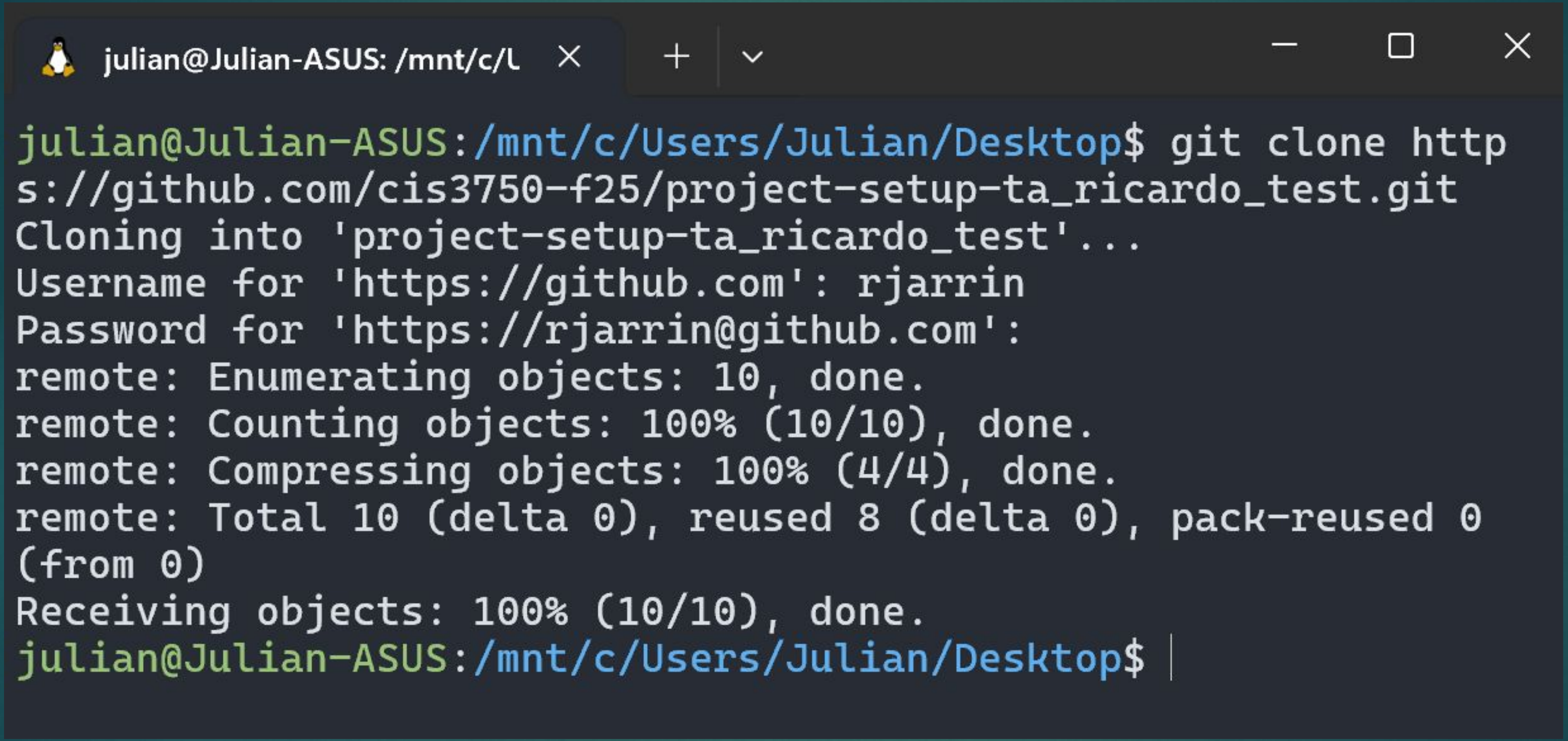
<u>Using HTTPS method:</u>
Asks for username and password, but says 'Password authentication not supported'

<u>So what happened?</u>
GitHub stopped allowing password authentication in 2021 (https://github.blog/changelog/2021-08-12-git-password-authentication-is-shutting-down)

Setting up a classic token for HTTPS

- Click on your profile picture (top right) **Settings**
- On the left menu, all the way at the bottom, click on **Developer Settings**
- Click on **Personal access tokens Tokens (classic)**
- Click on **Generate new token (classic)** from the dropdown menu
- Fill out the form:
  - Note: A descriptive name (e.g., CIS3750 Laptop)
  - Expiration: Up to you, though ensure it lasts for the whole semester
  - Scope: Check the main repo box. This is essential
- Click on Generate Token
- COPY THE TOKEN IMMEDIATELY. You will **never** see it again. Save it in a password manager or alternative safe places. When the terminal asks for a Password, you will paste this token.

Success!



```
julian@Julian-ASUS:/mnt/c/Users/Julian/Desktop$ git clone http
s://github.com/cis3750-f25/project-setup-ta_ricardo_test.git
Cloning into 'project-setup-ta_ricardo_test'...
Username for 'https://github.com': rjarrin
Password for 'https://rjarrin@github.com':
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 10 (delta 0), reused 8 (delta 0), pack-reused 0
(from 0)
Receiving objects: 100% (10/10), done.
julian@Julian-ASUS:/mnt/c/Users/Julian/Desktop$
```

What about using SSH?

- PATs are easier to get through in the lab time.
- But if anyone really wants to:
  - Check for any existing keys first. In the terminal run **ls -a ~/.ssh**. If you see files named id_rsa.pub or id_ed25519.pub, you already have a key.
  - If you don't have a key, generate one by running **ssh-keygen -t ed25519 -C "your_email@example.com"**. Press Enter through all the prompts
  - Copy the public key. Run **cat ~/.ssh/id_ed25519.pub** (or **id_rsa.pub**) and copy the entire output.
  - Add to GitHub □ Settings □ SSH and GPG keys □ New SSH key. Paste the key in the input field, give it a name, and save.

Getting Started (2/3)

- Edit
  - Open the new folder in a code editor (e.g., VS Code)
  - Change the README.md and labs/lab2.md files and save them
    - Change them how?

README.md

- Follow the instructions on the lab manual
- Update your file with your **Team name** and list all your **Team Members**
- Think of a descriptive message for the next step (e.g., “Added team name and members”) for the next step

Getting Started (3/3)

- Status, Add, Commit, and Push
  - ‘git pull’: Update the repo with latest info
  - ‘git status’: Checks status of git repo
  - ‘git add .’ : Stage all changed files. **You can name specific files too**
    - **git add README.md**
  - ‘git commit –m “Message here”: Save a snapshot locally
  - ‘git push’: Upload your saved changes to the shared repo

Brainstorming Requirements

- Thinking about your project:
  - As a group, brainstorm as many requirements as you can for your ITS project
  - **Identify stakeholders**
    - Who will use or be affected by this system (e.g., students, teachers)?
  - Think about both types of requirements:
    - **Functional**: What the system must do
    - **Non-functional**: Performance, usability, security, etc.

Writing User Stories

- Capturing Requirements
  - Convert your ideas into User Stories
  - Use this specific format:
    - As a [user type], I want [some action/function], so that [benefit]
    - Ex: "As a **student**, I want **to receive immediate feedback on my answers** so that **I can learn from my mistakes**"
  - Your goal is to aim for at lead **50** user stories as a team
    - Your goal is **quantity over quality** right now; don't worry about overlap

Organizing Your Stories

- Structuring Your Ideas:
  - Now, group your 50+ user stories into categories
  - You can choose any method that makes sense for your team, such as:
    - By Stakeholder (e.g., all “student” stories together)
    - By Feature (e.g., all stories about the tutoring model)
    - By Priority (e.g., “must-have”, “should-have”, “nice-to-have”)

Where to document stories: lab2.md vs Issues

- Some of you may already be familiar with using Issues for stories
- **For this lab, we will use labs/lab2.md**
  - This is perfect for the initial “brain-dump” and creates a single document with all your initial ideas
- In the real world (and a consideration for future work): Each user story often becomes a GitHub issue
  - Issues are trackable tasks. They can be assigned to people, labeled, and discussed individually. They form your project’s “backlog” or to-do list

Common GitHub issues

- “Failed to push some refs…” (Collaboration Conflict)
  - What it means: A teammate has pushed changes to the repo since the student last pulled.
  - How to fix: Run git pull before you git push

Common GitHub issues

- “Merge conflicts”
  - What it means: Both teammates edited the same lines in the same file. Git doesn’t know which version is correct, so it’s asking the human to decide
  - How to fix:
    - Don’t panic! This is a normal part of teamwork
    - Open the conflicting file
    - Look for the <<<<, ====, >>>> markers. The code between HEAD and the ==== is **your** change. The code after the === is the teammate’s change
    - Edit the file to be correct. You must delete the conflict markers and decide which version of the text to keep, or write a new version that combines both.
    - Save the file
    - Add, commit, and push

Lab Submission Checklist

Before you leave today, make sure you have

- ❑ Created a team on GitHub Classroom with all members joined
- ❑ Updated the README.md with your team name and members
- ❑ Brainstormed at least 50 user stories
- ❑ Organized the user stories into categories
- ❑ Added the categorized user stories to the labs/lab2.md file
- ❑ Committed all your changes to the repository, which should be updated through the semester.

Questions?

- Any issues?
- Did you want me to return to a specific slide?
- Clarification on project?