

Project setup (GitLab)

CS 346: Application
Development

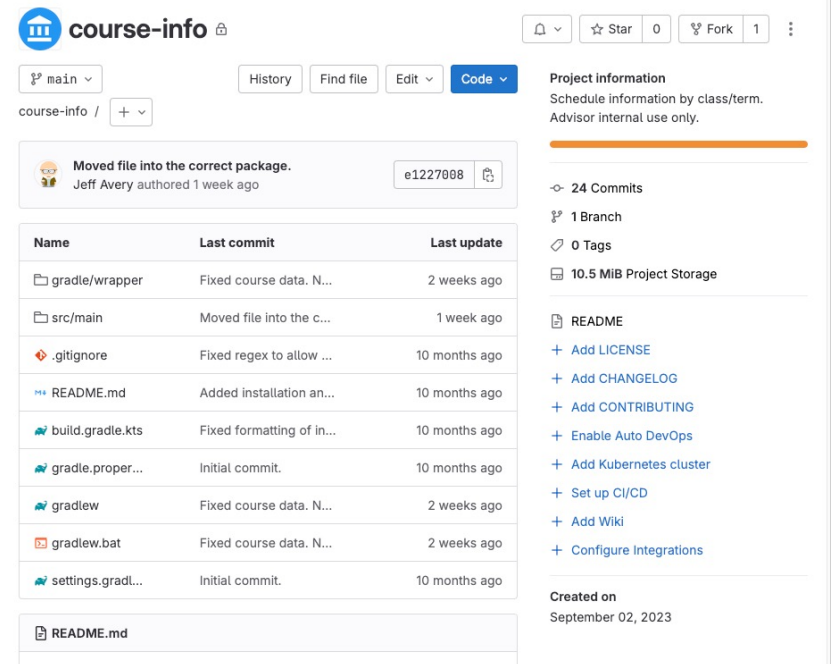
What is GitLab?

GitLab is a project tracking system.

- UW has our own self-hosted instance.
<https://git.uwaterloo.ca/>

Features

- **Project planning, tracking**
- Source code management (Git, merging)
- Continuous integration (running tests, deployment).
- Security auditing (out-of-scope for us)
- Wiki for documentation, diagrams.



The interface looks familiar.

How will we use it?

Here's the main functionality that we'll be using in this course:

- **Project tracking.** We will use project issues and milestones to track your work towards your project.
- **Source control.** Your GitLab repository is a Git repo. You should be cloning it to your personal computers and using the repo for your source code (and any other documents not in the wiki).
- **Software releases.** We'll use the built-in mechanisms to tag and release software for each milestone.
- **Wiki.** All project documents should be stored as pages in your Wiki, written in Markdown.

For a small sample project, see <https://git.uwaterloo.ca/j2avery/mm>

GitLab Setup

Basic GitLab project instructions.

Getting Started

1. Navigate to <https://git.uwaterloo.ca>
2. Click on + > **New Project** > **Create blank project**.

Project name
Team-101-5

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL
https://git.uwaterloo.ca/ j2avery

Project slug
team-101-5

Visibility Level ⓘ

Private
Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.

Internal
The project can be accessed by any logged in user except external users.

Public
The project can be accessed without any authentication.

Project Configuration

Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Enable Static Application Security Testing (SAST)
Analyze your source code for known security vulnerabilities. [Learn more](#).

Create project Cancel

The URL should be under a team-mates name e.g., j2avery

Give your project a descriptive name

Make sure access is private

Setup: Members

It doesn't matter who "owns" the project, since we will add other team members and give them full access.

Manage > Members

- Add team members as Owners.
- Add course staff as Developers.



Members were successfully added

Project members Import from a project Invite a group Invite members

You can invite a new member to `notae` or invite another group.

Members 2

Filter members Account

Account	Source	Max role	Expiration	Activity
 Caroline Kierstead @ctkierst ead	Direct member by Jeff Avery	Owner	Expiration date	User created: Apr 27, 2015 Access granted: Aug 06, 2024 Last activity: Aug 05, 2024
 Jeff Avery It's you @j2avery	Direct member by Jeff Avery	Owner	Expiration date	User created: Aug 20, 2015 Access granted: Jan 05, 2024 Last activity: Aug 05, 2024

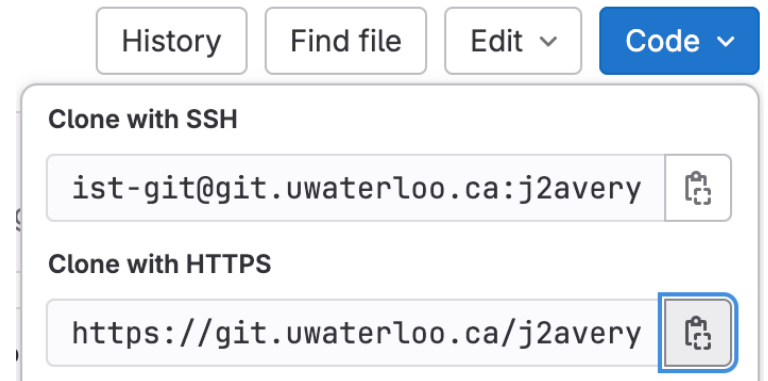
Setup: Source Code

Once the repo is created, everyone that has access should be able to `git clone` it to their local machine.

Code button > Copy the URL

```
$ git clone https://git.uwaterloo.ca/j2avery/mm
```

We'll add code later.



Setup: Starting Files

Add a **README.md** file to your repository.

- This is a markdown file that is shown when you open the project
 - i.e., it's the “Landing Page” for your project.

Add a **.gitignore**

- Just a plain-text file that you can add to the repo at the top-level.
- Include files and directories that you do NOT want in your repository. Git will skip over these files when performing operations.

Example files here: <https://git.uwaterloo.ca/j2avery/mm>

README.md contents

```
# SUPER-COOL-PROJECT-NAME
```

```
## Title A description of your project.
```

```
## Team Details
```

Basic team information including:

- Team number from forming a team step.
- Team members. Full names and email addresses.
- Link to your **Team Contract** wiki page.

```
## Other sections over time!!
```

This is the minimum required
for the Project Proposal stage.

You will add more as the
project progresses.

.gitignore contents

```
build/  
.gradle  
.idea  
.vscode/  
.DS_Store
```

You can probably just start with “build/” and add more over time.

Setup: Wiki pages

A [wiki is a simple content management system](#) – a place to store documents on the web.

- GitLab includes a simple wiki engine that lets you produce documents in Markdown format, and include them in your project.
 - Plan > Wiki, Create Your First Page
- You will need to create Wiki pages for all your project documents i.e., Project Proposal, Design Proposal.
- You can link Wiki pages to issues and vice-versa.