



Merly Mentor

User's Manual

Version 0.0.56 — © 2024 Merly, Inc.

Overview

Software developers spend around 70% of their time trying to understand code. Developers spend this time on a variety of tasks such as identifying defects, finding technical debt, or generally trying to understand the code for feature extension, amongst others. We created Merly Mentor to help programmers with these and many other problems. In this guide, we walk through Mentor to help provide an intuition of how it can be helpful and encourage users to continue to exploring using Mentor in even new novel ways, not described here.

*We build software that builds software..
Your privacy is our #1 priority.*

Note: This user manual is being iteratively updated.

Contents

1	What Is Merly Mentor?	4
2	What is Merly Mentor?	5
2.1	Terms to Know	5
2.2	Requirements, Prerequisites, and Dependencies	6
3	Quick Start Guide Installation Steps	7
3.1	Prepare the Installation	7
3.1.1	No Licence Key? Register for Trial Access on Merly.ai	7
3.1.2	Already Have a License Key?	8
3.2	Installing Merly Mentor	10
3.3	Logging in	10
3.4	Adding the First Repository	11
3.5	Adding the First Repository	12
4	Navigation Overview	13
4.1	Homepage Elements	14
5	Mentor Service: Adding a Repository	14
5.0.1	New Repositories	15

5.0.2	Advanced Add Options	15
5.1	Preparing the Repository and Running Inference	18
5.1.1	Running Inference	19
5.1.2	Running Inference on a Specific Branch	19
5.2	Analyzing the Results	20
5.2.1	Browsing An Instance	21
5.3	Configuring the Mentor Service	23
5.3.1	Configuration	23
5.4	Tags and Filtering	24
5.4.1	Sorting and Filtering on the Home Page	24
5.5	Repo Issues and Quick Actions	25
5.6	Managing Users	26
5.6.1	Adding Users	26
5.6.2	Users List	26
5.6.3	Editing Users	27
5.6.4	Deleting Users	27
5.6.5	Resetting a Password	27
6	Navigating the Mentor Service	29
6.1	Log In Screen	29
6.2	Repo Lists	29
7	Mentor Service: Mentor Summaries	31
7.1	Overview	31
7.2	Accessing Mentor Summaries	32
8	Mentor Service: Mentor Insights	33
8.1	Overview	33
8.2	Accessing Mentor Insights	33
8.3	Browsing Insights	34
8.4	The Insights Panel	34
8.5	Privacy Settings	35
9	Mentor Service: User Privileges	36
9.1	Managing User Privileges	36
9.1.1	Default Roles and Privileges	36
9.1.2	Overview of Privileges by Role	36
9.1.3	Editing User Privileges	38
9.2	Managing Repository Privileges	39
9.2.1	Opening and Editing Repo Settings	39
9.2.2	Restricting Repo Access	40
9.2.3	Saving Changes	40
10	Mentor Service: Exporting Data	41
11	Using Mentor with GitHub Actions	42
11.1	Setting up the Environment	42
11.2	Creating the Workflow	43
11.3	Running Mentor with GitHub Actions	46
11.4	Exploring Results	47
12	Mentor Service: Repository Comparison Tool	48
12.1	Comparison Tool - Selecting	48
12.1.1	Notes on Functionality	48
12.1.2	To Select and Compare Repositories	49

13 Frequently Asked Questions (FAQs)

51

14 Release Notes

53

1 What Is Merly Mentor?

At a high level, Merly Mentor can be thought of as a code-based reasoning system. More specifically, Merly Mentor (also known as Mentor) is a self-supervised system which utilizes federated learning and an iterative multi-tiered code abstraction model to help reason about the semantics of code.

An overview of Mentor’s system design is shown in Figure 1. Its distributed learning framework enables it to learn from billions of lines of code in a single day on commodity hardware. In learning from such an abundance of code, Mentor can learn both good and bad code syntax, interesting and uninteresting (copied) patterns, semantics, and even recommend patches for defects it finds. Once trained, Mentor’s model can be used for a variety of tasks such as: (i) detecting potential technical debt or defects and recommending fixes in existing code, (ii) grading the quality of an existing repository, and (iii) guiding programmers through the important aspects of any code repository, to name a few.

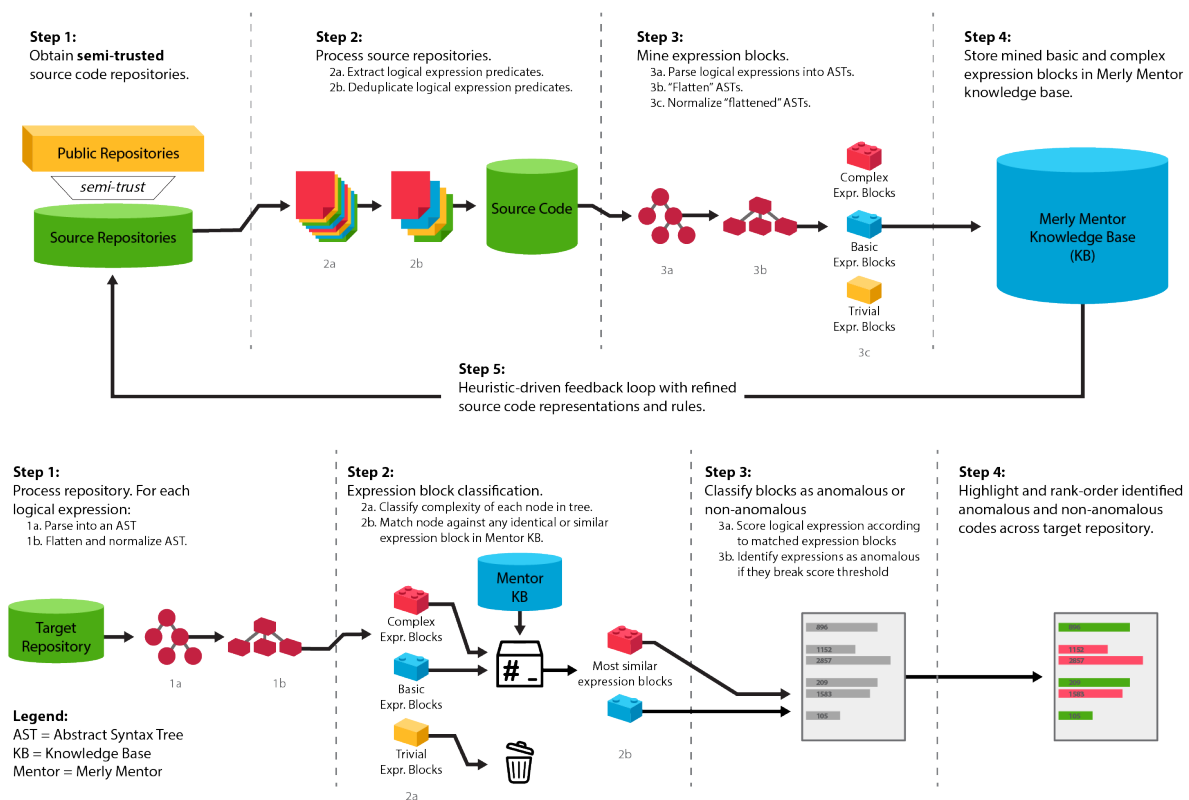


Figure 1: System Overview of Merly Mentor.

In this one aspect, Merly Mentor is a machine programming (MP) system that learns how to identify anomalous code fragments in source code by training on billions of lines of source code across dozens of programming languages. These anomalous code fragments, also known as anomalies, are often latent defects in the existing code that programmers have failed to identify or correct. Mentor helps programmers find these anomalies and correct them, thereby improving the overall quality of the existing software. For this version of Mentor, we only include Mentor’s ability to perform inference (i.e., detect good or bad patterns) on code. In subsequent releases, we may also include the ability to train new models on other code bases, including users’ own proprietary ones.

For answers to Frequently Asked Questions, see [Section: FAQs](#).

2 What is Merly Mentor?

Developers spend 70% of their time understanding code. Yet, code understandability tools don't exist.

We're here to change that. Merly is building an autonomous AI software platform called Mentor. It moves at the speed of AI and helps in many ways: reliable code generation, repository lifetime analysis, defect detection and mitigation, and more!

Mentor helps developers understand, fix, and maintain deeply complex code. It also provides insight, tracking, and summarizations on code quality and trends over time. And we trained it on a trillion of lines of code. So, yeah-it's our mentor.

2.1 Terms to Know

Merly.ai is a startup looking to massively accelerate development and improve software quality. **Merly Mentor**, it's first flagship product, is a software ecosystem intended to help developers understand, fix, and maintain deeply complex code. It also provides insight, tracking, and summarizations on code quality and trends over time. And we trained it on a trillion of lines of code. So in a way, it *is* a mentor!

Mentor works by running **inference**. In simple terms, it will compare the code you're working on against Merly's trained knowledge base, and then help you understand it with a summary of the repository. Merly Mentor will point out Issues, a term meaning an expression the service has flagged as needing review because it's unfamiliar or interesting in some way.

Some of these Issues are **Anomalies**, expressions that could lead to bugs, technical debt, unexpected behavior, memory issues, or even crash errors. Others are marked as **Notables**, which are code fragments that potentially have a high mental cost associated with understanding them. **Uniques** is a term used to describe an expression pattern that Merly Mentor hasn't seen before—which could be fine, just novel.

For each issue, you'll be presented with **Mentor Insights**, a system which leverages Merly's proprietary foundation model (FM). With Mentor Insights, all anomalous and notable expressions that Mentor identifies can be analyzed more in-depth. This analysis includes multiple explanations of the code, potential fixes, as well as other considerations. It also includes three different levels of code security, including a fully-private mode where no source code is sent to Merly servers.



Mentor Summaries scores your repository's health, reasons about trends, and provides customized improvement strategies tailored to your specific software needs based on a lifetime of analysis, heuristics, and generative AI tools. Those Summaries are represented by **Mentor Scores**, which can be thought of almost like a credit score, but for code. Instead of looking at the codebase as it is *now*, it uses the lifetime and uses the entire history of commits to develop a metric of health using heuristic-based scoring. **Mentor Compare** is a tool that is used to compare up to four repositories' metrics side-by-side.

...but that's not all!

If you're feeling excited, or even a little overwhelmed at this rapid introduction to Merly Mentor, this **User's Manual** is meant to get you familiar with the tool and have up and running as quick as possible with a step-by-step installation.

2.2 Requirements, Prerequisites, and Dependencies

The Merly Installer will download the application binaries as well as models and other dependencies.

- Operating Systems
 - **macOS** Merly Mentor is designed to work on macOS Monterey (version 12) and above.
 - **Windows** Merly Mentor is designed to work on Windows 10 and Windows 11 systems.
 - **Linux** Merly Mentor is designed to work on most Linux distros, such as ones based on Debian and it's variants—such as Ubuntu—as well as Red Hat and Fedora, and others.
- Runtimes
 - **Microsoft Visual C++ Redistributable** To avoid of missing runtime issues (like missing DLLs, i.e. *msvcp140.dll* or *vcruntime140.dll*), be sure that the MSVC++ redistributable is installed. The installer will handle this for you if it detects the runtime is missing.
Link: [The Microsoft Visual C++ Redistributable download page](#) 
- Source Control
 - **Git** Git is a free and open source distributed version control system widely used in the industry. Mentor uses some git commands when processing and manipulating repositories behind the scenes. The installer will handle this for you if it detects that git is missing from the system.
Link: [Git's download page](#) 

A Note on Antivirus Software If your installation is halted or flagged by antivirus software, consider temporarily disabling antivirus software, as some may interfere with Mentor's installation process—particularly with essential third-party software like Git.

Alternatively, you can manually install these dependencies before installing Mentor. Remember to re-enable antivirus software afterward to ensure security.

3 Quick Start Guide Installation Steps

In this section, you will be walked through our quick start guide so you can get Mentor up and running in no time. The two things you will need on-hand is your license key and a repository URL—the installer will walk you through each step of the process.

3.1 Prepare the Installation

To prepare the installation of Merly Mentor in your operating system's graphical user interface, create folder you'd like the installation directory to be, or open a terminal window, and create a directory for where you want it installed. For the command line:

🍏 macOS/Linux:

```
$ mkdir mentor
```

```
$ cd mentor
```

🪟 Windows 10/11:

```
$ mkdir mentor
```

```
$ cd mentor
```

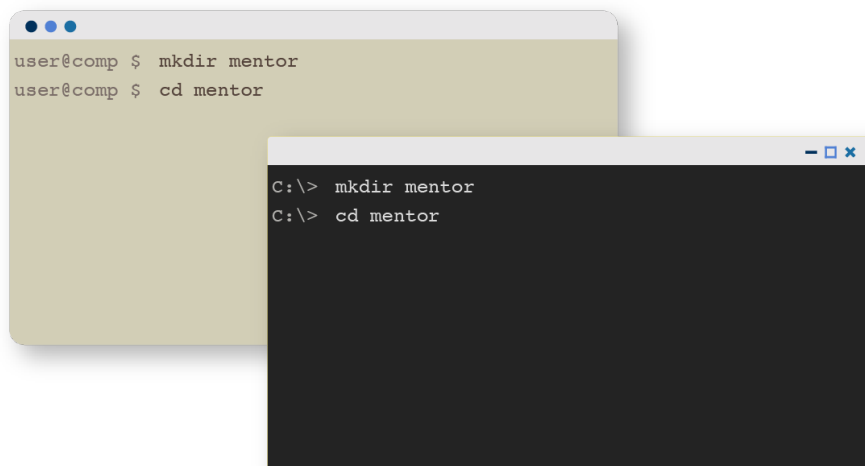


Figure 2: For Linux or macOS, refer to the left (tan) window in the figure; for Windows, refer to the black one on the right.

3.1.1 No Licence Key? Register for Trial Access on Merly.ai

Open your browser and navigate to <https://merly.ai/>

In the navigation bar at the top of the page, click or tap “Early Access” in the navigation bar which will direct a user to the portal to register for a Mentor Trial license.

EARLY ACCESS

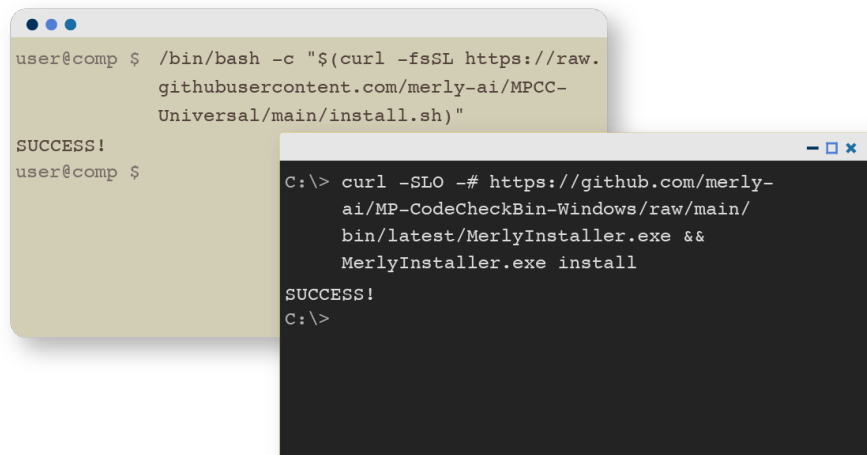
Figure 3: The Early Access button.

Filling out the form on this page will provide access to a free 90-day trial of Merly Mentor. Submitting the form will do the following:

1. Register your name and email address with Merly
2. Generate your license key and prepare instructions
3. Deliver you an email specifically tailored to your activation
4. Send instructions on how to install the Merly Mentor Service

3.1.2 Already Have a License Key?

In the navigation bar at the top of the page, click or tap “Early Access” in the navigation bar which will direct a user to the portal to register for a Mentor Trial license.




The image shows two terminal windows. The left window is a tan-colored terminal window with a light background. It shows a user at a prompt 'user@comp \$' entering a command to run a shell script from a GitHub repository. The command is: `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/merly-ai/MPCC-Universal/main/install.sh)"`. The output is 'SUCCESS!' followed by the prompt 'user@comp \$'. The right window is a black terminal window with a dark background. It shows a user at a prompt 'C:\>' entering a command to download and run a Windows installer. The command is: `curl -SLO -# https://github.com/merly-ai/MP-CodeCheckBin-Windows/raw/main/bin/latest/MerlyInstaller.exe && MerlyInstaller.exe install`. The output is 'SUCCESS!' followed by the prompt 'C:\>'.


Figure 4: For Linux or macOS, refer to the left (tan) window in the figure; for Windows, refer to the black one on the right.

Copy and paste the command below. This script will:


1. Downloads the Merly Installer, which will run the setup
2. Ask for your license key to register your product and activate your subscription
3. Download Merly Mentor, trained language models, assets, and dependencies

4. Install and register the Mentor Service
5. Launch Merly Mentor when complete

Attention! Visit [Visit Merly.ai/](https://merly.ai)  for the most up-to-date 1-click-copy installation command, or Support for any issues.

 macOS/Linux:

```
$ <paste most recent command in>
```

 Windows 10/11:

```
$ <paste most recent command in>
```

3.2 Installing Merly Mentor

In the folder you created on your local machine, follow the instructions in the email you received, or paste the installation command in your operating system: either into a terminal or Command Prompt, and press enter.

What to expect for your installation:

1. **First, the command will download and execute the installation script.**

The installation script will detect your operating system and then download the Merly Installer which corresponds.

(You will see this download as a progress bar across the output.)

2. **Next, you will see a welcome message.**

In addition, information confirming your registered name and email address which corresponds to your license and activation.

3. **Detection of open internet port and Service name.**

The Merly Installer will look for an open port to serve the Mentor Service on, and display that in the output.

4. **Then, the Installer will download everything necessary.**

The trained language and other models, service assets, and dependencies will be downloaded, showing download progress in the output window.

5. **Then, the Merly Installer will register Merly Mentor as a service.**

Once started, loaded, and confirmed to be running, access via the localhost as well as the computer's IP will be shown in the output.

6. **The Installation completes.**

When successful, in the output will be shown: "Install complete!" and "Service 'MerlyMentorService' Successfully Started." Any errors or other notifications are shown as well.

Merly Mentor will begin in the background as a system service.

7. **Then, the final step.**

After a few seconds, the browser will be opened and the "first launch" experience begin, allowing you to sign in and add a repository.

Congrats! You are now ready to start using Merly Mentor!

3.3 Logging in

Upon successful installation, the user needs to set a password and accept the EULA. You can see an example of this in Figure 5.

Notes:

- Your licensed name and email are displayed in the First Launch window.
- The password you set must be verified to match (but can be reset later).
- The EULA must be read and agreed to before logging in for the first time.

After setting up the first user account, log in.

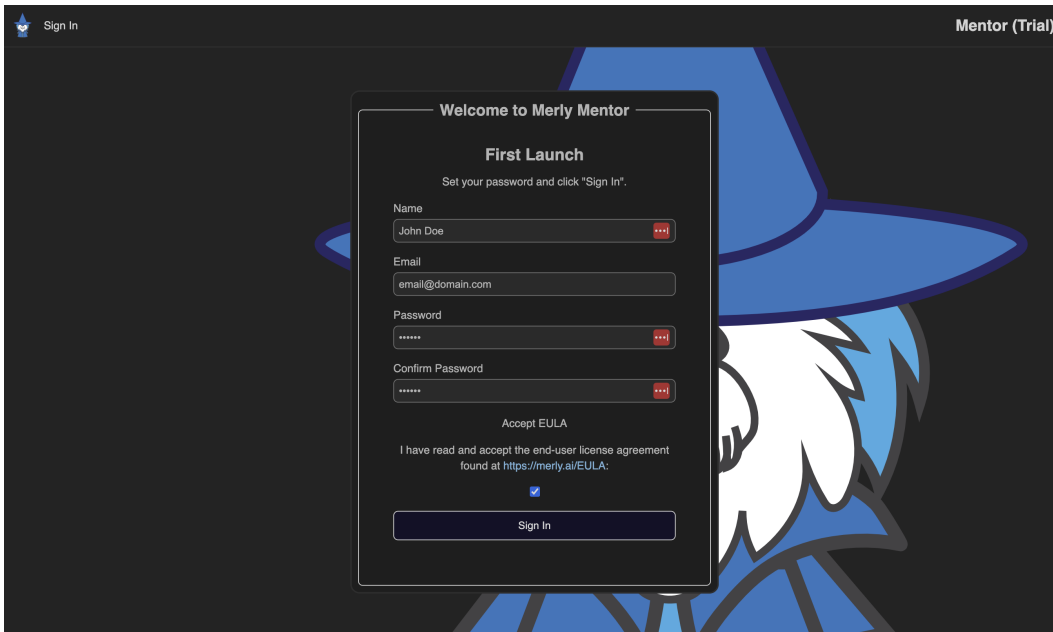


Figure 5: Logging in for the first time.

3.4 Adding the First Repository

Though you don't have to, Merly Mentor will guide you in adding your first repository after successfully installing.

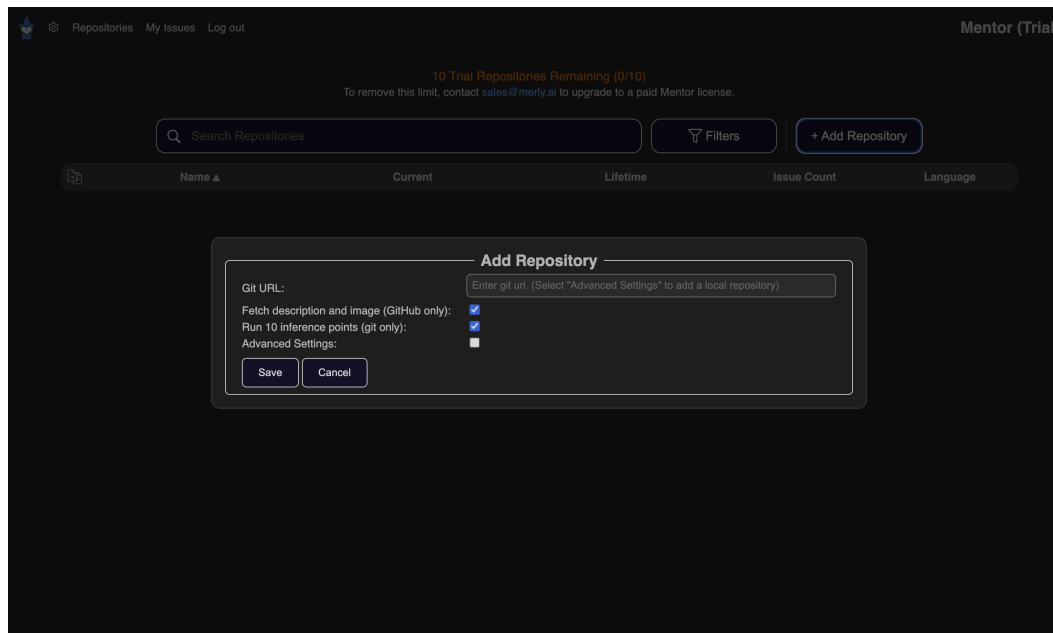


Figure 6: Logging in for the first time.

More details will be provided for adding a repository in Section 5.

In the window, you'll see a localhost URL. Back in your browser of choice, navigate to localhost:4200 (or whatever port you specified), where you'll be greeted by the Mentor Service.

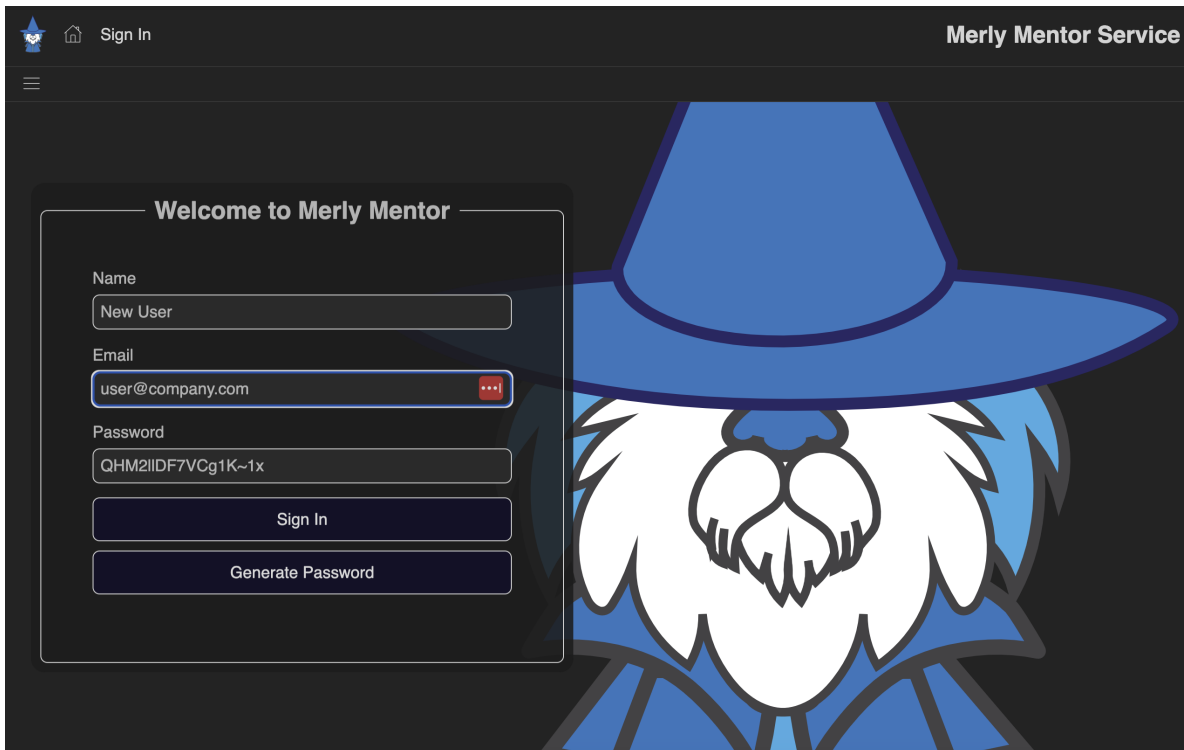


Figure 7: On the first time to the Mentor Service, you will be asked to set a password.

Verify the log-in information, set your password, and log in.

All Done!

That's it! You've installed Merly Mentor with the quick start installer. You are now ready to browse the inferences, dig into the code metrics and performance, read and analyze Mentor Summaries, compare projects, and explore Mentor Insights.

3.5 Adding the First Repository

The Mentor Service is installed and registered as a system service and accessed through a browser.

By visiting localhost:port or double clicking the link file in your install folder, you will be taken to the login page. If you've just launched the service for the first time and set your password, you'll be prompted to add a repository. *(Note this is optional. A repository can be added at any time).*

More will be covered in Section 5 (Adding a Repository), but as a quick guide if a user wanted to add one, paste a Git link into the provided box, and click `Save`.

This will add the repository to the service, clone it locally, pull metadata if available (from GitHub for instance), and if the option was checked, run inference on 10 points.

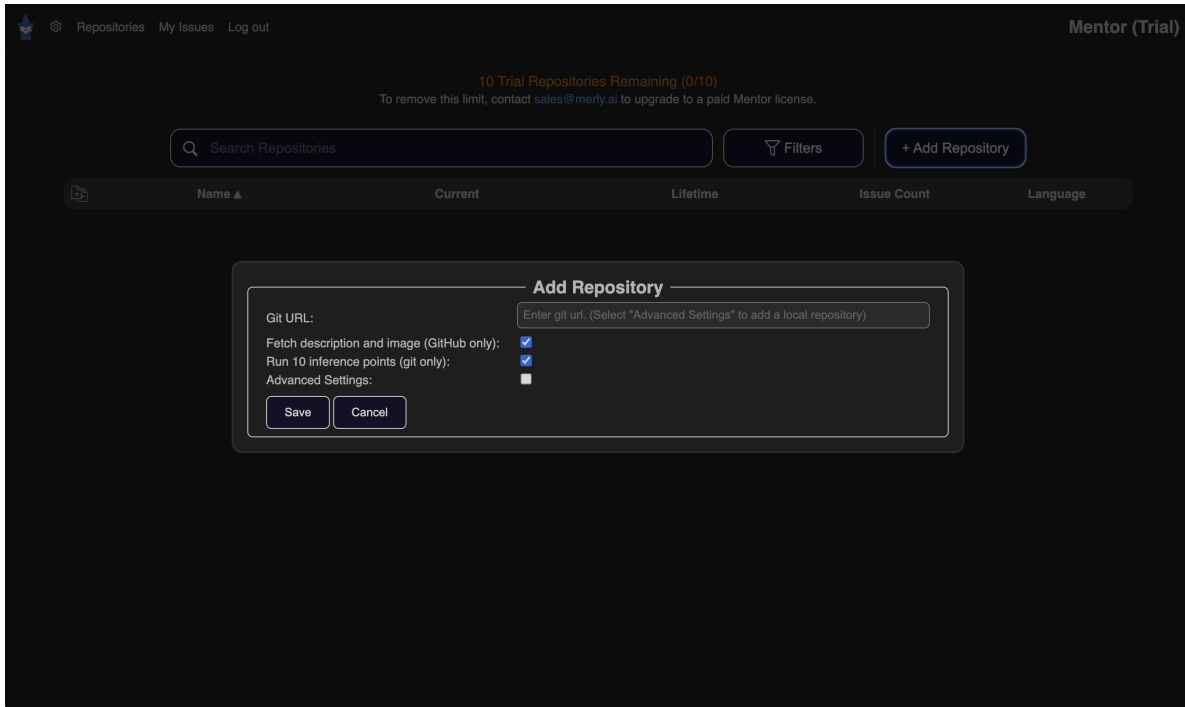


Figure 8: When there are no repositories on the Mentor Service, you will be prompted to add one.

4 Navigation Overview

Using a web browser, navigate to the configured address. The default is localhost:4200, but this can vary depending on your specific installation and system settings.

At the top of the page is a title bar. You'll see the name of the service on the right and various navigation options on the left.



Figure 9: A first-launch of the Mentor Service on a new Merly Mentor install with default configuration.

The Merly logo links to the merly.ai website, and "Repositories" will take you to the Repository List, which is also the homepage. Next to that is a button to **Sign In** or **Log Out**, depending on if there is a user logged in.

Once you're logged in, more items will appear on the title bar, such as a link to "My Repo Issues" and the gear icon (⚙️) which will navigate to the Configuration page for settings.

Clicking or tapping the sign in button opens a dialog box in the center of the page. The username is the assigned email address, and the password is what the service administrator generated and provided, or the user themselves set after first login.

If this is the first time logging in after installation, these fields will be pre-populated and ready to have a password set.

If a user has forgotten a password, a password can be reset with the "Forgot password" link. Please request a password reset from the Merly Mentor service administrator. For more on user management, see Section 5.6.5 (Mentor Service: Users - Resetting a Password).

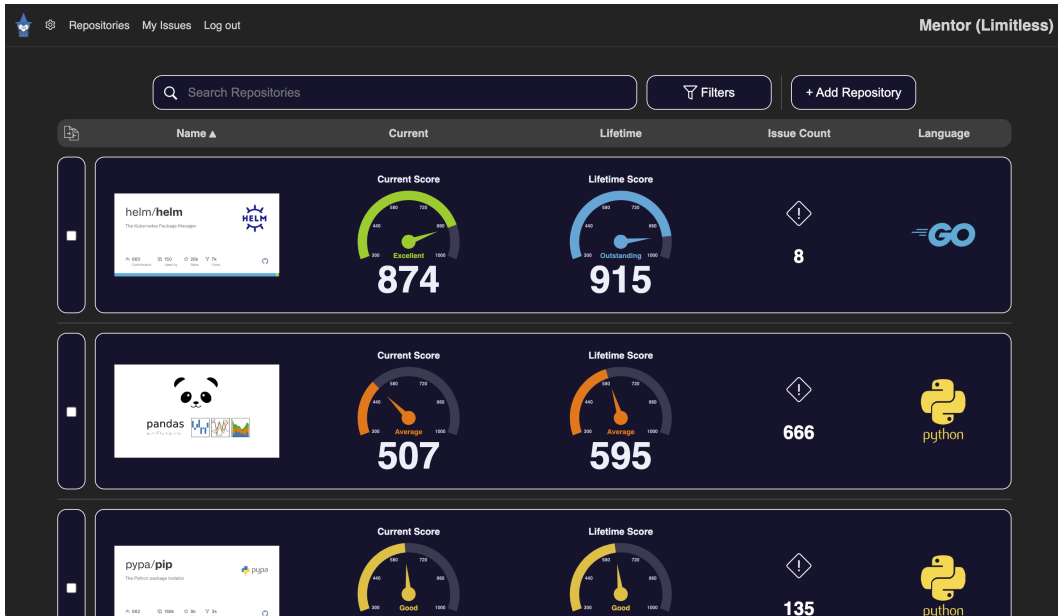


Figure 10: A first-launch of the Mentor Service on a new Merly Mentor install with default configuration.

4.1 Homepage Elements

The title bar is the topmost navigation menu. Below that, the main navigation menu. Once a user is logged in, the options change to include links to Repositories, Settings, My Repo Issues, and Log out.

Hovering over the log out button will display the name of the logged in user.

On pages where further navigation is possible, a hamburger menu (☰) will appear on a final menu below the configurable navigation bar. This is where any links you add in the configuration will be available, navigation between Instance points inside of a repository, and browsing Anomalies, Notables, Files, or other views.

The main window of the Mentor Service's page will contain things such as leaderboards, repository performance graphs, issues lists, expression details, actionable items, configuration, and more.

5 Mentor Service: Adding a Repository

This section outlines the process to add a new repository. Doing so will allow inference to be performed, Summaries and Insights to be investigated, metrics to be displayed, and more.

1. In the menu bar at the top, click or tap **Repositories**. This page not only shows repositories currently being monitored by the Mentor Service, but allows users to add repositories with the navy blue **+** Add Repository button.



Figure 11: The button used to add a repository.

2. An "Add Repository" context menu will open, where project and repository information can be configured.

5.0.1 New Repositories

The quickest way to add a repository to the service is to type in or paste a Git URL into the Add Repository dialog box.

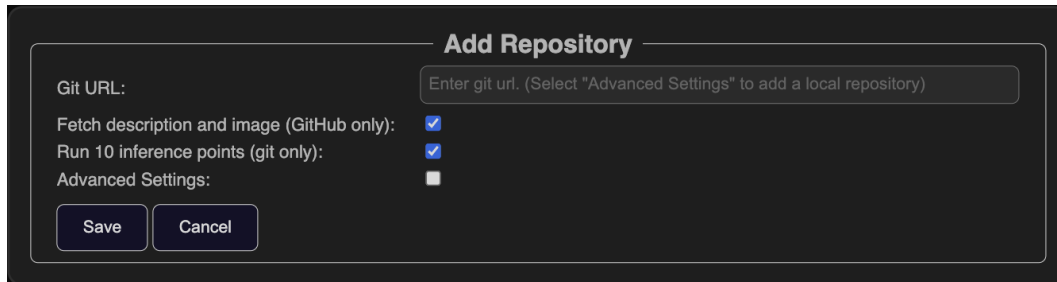


Figure 12: The Quick Add dialog.

There are three options in this me:

- **Fetch description and image (GitHub only):**

Checking this box will pull a title image and the project description from the GitHub link provided. If unselected, the repository's name will appear instead of a logo. A repo logo (or "icon" as it's called in Mentor), can be added later manually by editing the repository details.

- **Run 10 inference points (git only):**

If this box is checked, the repository will be added, cloned locally, and inference for 10 points will be queued to run. If you would like to run more than 10 points of inference, unselect this box. Saving will only add the repository to the service.

To run inference on more or less than 10 points, choose the repository from the list, in the "Jobs" tab, click "Git Clone Now," and after cloning, choose "Run Inference Points." There, type in a number, or use the default of 50, and then start inference by clicking "Infer with Points."

- **Advanced Settings:**

Checking this box will expand the dialog to include more settings that advanced users may want to modify.

5.0.2 Advanced Add Options

Required Field In the newly opened dialog, the only required field is the "Git URL." In the case of a local repository, the "Git URL" can be omitted in favor of the "Git Path" field.

For advanced configurations, you *can* fill in the following fields:

- **Name** : labels the repository; this will be displayed as the *section* title on the /Repository or homepage. This will be set automatically if using a git URL.
- **Description** : a short description of the project. This will be populated automatically if the option above it is selected, and it is a GitHub project
- **Icon Link** : image URL to represent the project—used in various places and views
- **Project Link** : a URL link to the project's website or Git page
- **Git Root** : local path to the source code after Mentor clones it
- **Git Remote** : reference to the git remote (in most cases, origin). **Mentor will automatically detect and populate this.**

Add Repository

Git URL:

Fetch description and image (GitHub only):

Run 10 inference points (git only):

Advanced Settings:

Name:

Description:

Icon Link:

Project Link:

Git Root:

Git Remote:

Git Branch:

Git Username:

Git Token:

Source Root:

Exclusion Folders:

Exclusion Files:

Auto Update Interval:

Tab Size:

Tags:

All Files:

Public Insights:

Prefetch Insights:

Share Votes:

Restrict Repo Access:

Figure 13: Adding a repository with advanced settings

- **Git Branch** : reference to the git branch to be analyzed. **Mentor will automatically detect and populate this—but inference can choose a different branch later if you choose**
- **Git Username** : username for GitHub authentication
- **Git Token** : GitHub token used for authentication (On GitHub, account settings, Developer, Tokens)
- **Source Root** : local path for source code or other repositories
- **Exclusion Folders/Files** : in the form of a comma separated list—will be split into parts with ',' as split char folder. Exclusions are for the 'directory' part of any file, and "?" matches any single char, and "*" matches any range of characters. Exclusions for the "file" setting are for the filenames specifically, therefore forward- and back-slashes should not be used for that pattern (as it will return no matches), and the wildcards "?" and "*" are treated the same as for folders
- **Auto Update Interval** : sets the time in a format before git will fetch an update and if there are new commits, inference will again. The minimum is currently 1 day. Format supported: 0-9d|day|days
- **Tab Size** : number of spaces per tab (default is 2)

- **Tags** : configurable per repo; a comma separated list of tags used to be able to sort and organize repositories on the homepage list
- **All Files** : if checked; the repo will show all files in the "Files View," not just ones with identified anomalies or notables
- **Nested IFs** : an experiential feature that is turned off by default, used to also analyze nested IFs as a single expression
- **Public Insights** : if checked, Mentor Insights will default to "public"
- **Prefetch Insights** : if checked, this option will utilize background jobs to fetch Mentor Insights for faster retrieval
- **Share Votes** : if checked, Mentor Insight feedback votes will be shared with Merly's foundation model
- **Restrict Repo Access** : part of the Privilege System, checking this box will allow a user to customize repository access on a per-user basis

Note: All you need to add a repository is a Git URL in the Git URL field. The Mentor Service will auto-populate the name field and automatically detect which git branch to use.

5.1 Preparing the Repository and Running Inference

If you choose not to select any options and only add a URL, you must take several steps to run inference and generate results.

1. Clone the repository
2. Set number of inference points and let Mentor run inference

On the Repositories page (Click or tab "Repositories" from the navigation bar to get to the home page), the newly added repository can be accessed. Clicking or tapping on the title of the repository will display the Repository Overview page.

The Repository Overview page contains tabs for the information contained within, such as Mentor Summaries, charts, repository details and actions, and instance SHAs. **Until the repository is cloned and inference is finished, most of these tabs are empty.**

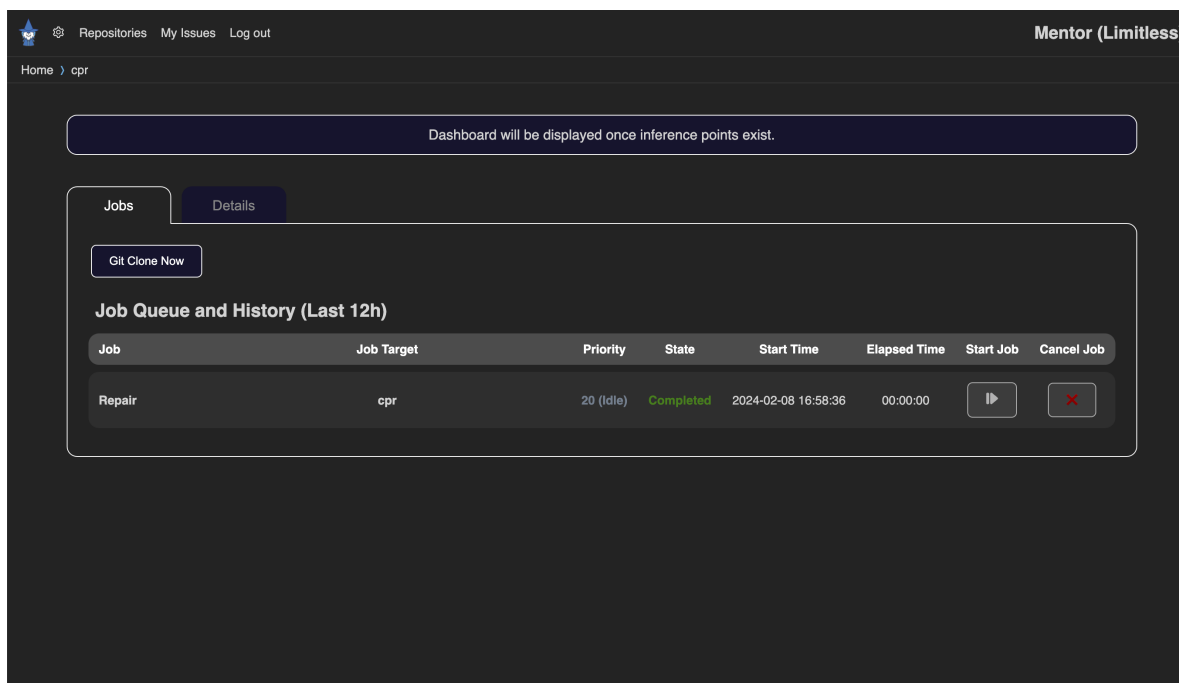


Figure 14: A newly added repository in the Mentor Service, before inference.

Step 1 - Open the Jobs Tab First, navigate to the "Job" tab if it is not already selected (see Figure ??). In this tab, you will see the settings which were used when adding the project to the service. If it is not a local repo, then the repo must be cloned before inference can run.

Step 2 - Clone the Project Locally The **Git Clone Now** button will clone the repository locally based off of the previously configured path. Note: in the "Details" tab, paths may show as "missing" until a repository action is taken. If the directories don't exist, they will be created automatically.



Figure 15: The button used to clone the project locally

Note: Repo modification actions such as editing, deleting, running or rerunning inference, cloning, and fetching need a user role of *Lead* or higher access level (by default).

5.1.1 Running Inference

Next, based off of your needs there are two options for inference.

- **Run Inference:** allows an inference to be run on specified commit by unique identifier (commonly called a git SHA) or branch. This field lists the most recent 100 commits, or alternatively, can accept a specific SHA pasted in.
- **Run Inference Points:** sets a quantity of inference points to run, and then runs inference* (note: existing inference data will be removed and points re-run, if inference points exist.)

*The first inference point will be the earliest commit and the last is the most recent. Mentor will divide all remaining points in between as those points and matches them to a date that fits closely within an allowable margin.

Note: specific SHAs and points can be added or removed.

5.1.2 Running Inference on a Specific Branch

To run inference on a specific branch, click or tap the **Run Inference** or **Run Inference Points** buttons. **This is optional.** By leaving it blank, Mentor will automatically detect where the HEAD is and choose the branch accordingly (usually main—but this is user-configurable).

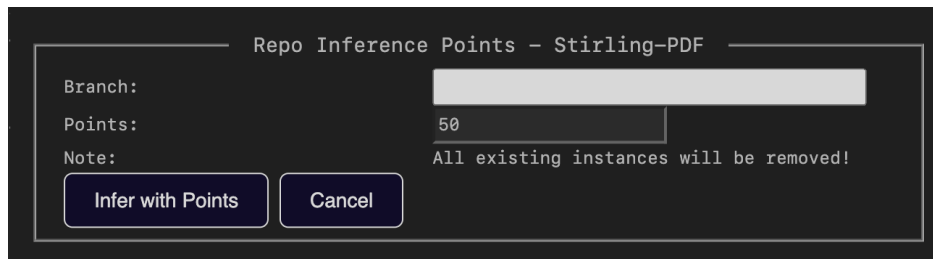


Figure 16: When running inference, you have the option to choose the branch as seen with this drop-down menu.

5.2 Analyzing the Results

Analyzing the results can be performed in a number of ways. First, on the Repository Overview, a Dashboard is presented with a number of metrics available at a glance. These metrics represent items pertinent at the project-level.

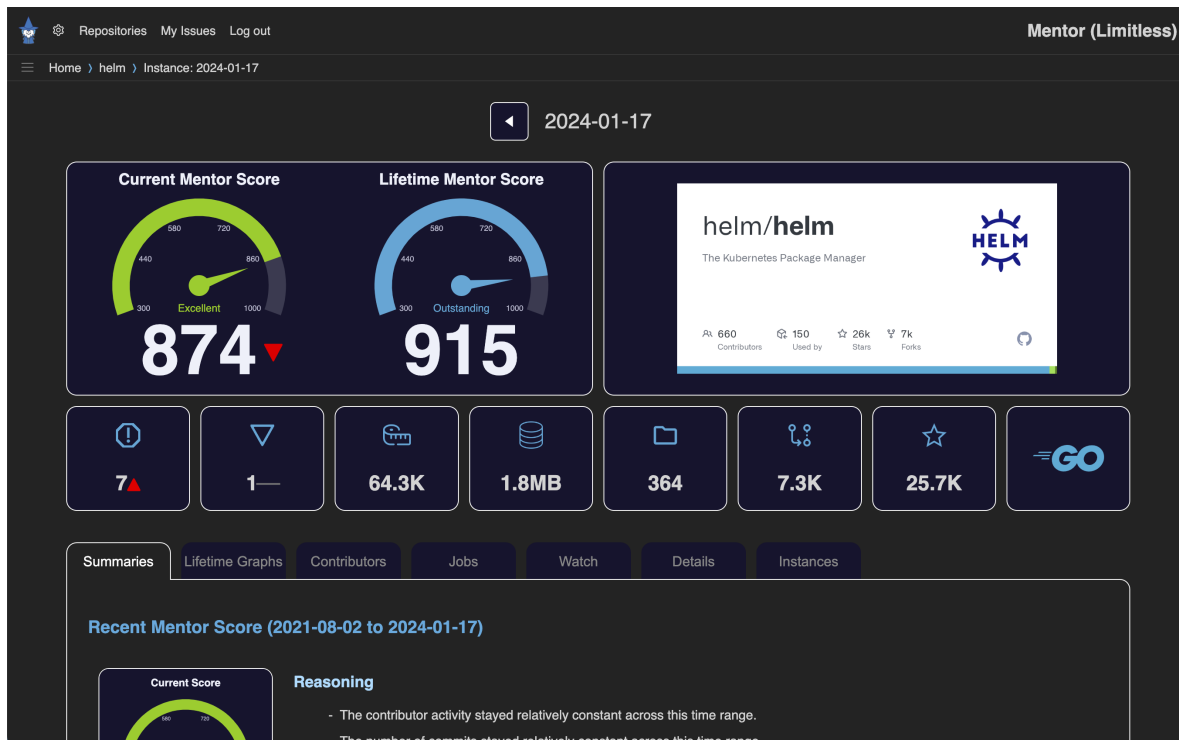


Figure 17: Mentor Service Repository Overview; a detailed page with a Dashboard of metrics and details in tabs.

The items displayed in the Dashboard are:

- Current Mentor Score: the project's score as it stands at the latest time point
- Lifetime Mentor Score: the project's score across the repository's lifetime
- Repo icon: a logo for the project
- Anomalies (octagon with exclamation point): number of Mentor-identified out-of-distribution code fragments
- Notables (inverted triangle): number of Mentor-detected unusually complex and out-of-ordinary code fragments
- Lines of Code (LoC): lines of code in the project
- Size: size, on disk, of the repository
- Files: total number of files
- Commits: Number of git commits (if available)
- Stars: how many "stars" a project has (GitHub/GitLab)
- Language: what programming language the project is written in

After Running Inference After cloning the repository and an inference action is taken, more options become available in the tabs below:

- **Re-Run All Points ("Jobs" tab):** Will use the same dates and SHAs, but delete the data from those points, and run inference again
- **Instances tab:** below the dashboard, more details for each Instance are available: date, SHA, size, lines of code, and number of issues, Mentor Score, and other metrics
- **Issues "Details" tab:** navigating here will show a list of all identified project issues and other associated data. Note that Mentor uses this term in various places to refer to both anomalies and notable expressions (more on these in later sections).

5.2.1 Browsing An Instance

At each inferred point in time, Mentor will provide lists of what anomalies and notables it's found. Together, these are referred to as "Issues."

There are multiple ways of viewing browsing Issues:

1. The Issues list, found in the "Details" tab
2. From a time point chosen from the Lifetime graphs (in the tab called "Lifetime Graphs")
3. From a time point chosen from the list in the "Instances" Tab

1 - The Issues List A list of all current issues can be reached from here:

1. From the homepage, click or tap the repository you'd like to browse if not already viewing a repository
2. From the that overview page, click or tap the "Repository Details" tab
3. Click or tap the "Issues" button



Figure 18: The "Issues" button will show a list of all anomalies and notables for the repository.

This list contains all anomalies and notables currently identified in the repo.

2 - From the Lifetime Graphs Each lifetime chart shows data across the entire lifetime of a repository; along the x-axis are the dates of inference points.

Hovering over the performance graph shows metrics at-a-glance. Clicking or tapping at any point will display that date's inference.

Once on the Repository Instance, there are several ways to explore Issues:

- From the hamburger menu, navigating to one of the following:
 - New: issues which are new to this instance (ones which weren't in the previous time slice)
 - Anomalies: anomalous expressions which are present on this instance



Figure 19: Showing the the hover effect of a repository’s Lifetime Code Overview graph; in this example, clicking or tapping will navigate to the 2021-01-07 instance.

- Notables: notable expressions which are present on this instance
- From the Dashboard, navigating to one of the following:
 - Anomalies icon: anomalous expressions which are present on this instance
 - Notables icon: notable expressions which are present on this instance

3 - From the Instance Tab List The last tab under the Dashboard on the Repository Overview page contains a list of all inferred points. This list is sortable, shows repo size at the given date, number of lines of code, number of Issues, and other metrics.

- “AIL” stands for “Average issues per n lines of Code” - “SR” stands for “Score Ratio,” which is a ratio of expression complexity.

Clicking a SHA’s row will navigate to that Instance. Once there, follow the steps found in 5.2.1.

5.3 Configuring the Mentor Service

Note: These are advanced features. Backup your .mentor folder in the install directory to restore previous settings.

In a web browser, after navigating to the Mentor Service home page, the Settings menu will become available by clicking or tapping on the gear icon (⚙️) in the menu bar at the top of the page. Further sub-menus will expand by clicking or tapping the title of that section.

5.3.1 Configuration

Advanced daemon configuration options are as follows:

- **IP** : Mentor will detect and bind the machine's IP address automatically during the installation. Here, a specified IP address to connect to the service can be configured. 0.0.0.0 will bind to `localhost` on local installs)
- **Port** : specified internet port used to connect to the Mentor Service. Note that changing this could result in the service not being able to connect and properly be accessed. Changing the port will not take effect until service restart.
- **SSL Cert** : if necessary, your own provided certificate; a path to a .pem file (i.e. one that has been split from a PFX file)
- **SSL Key** : if necessary, your own provided certificate; a path to a .key file (i.e. one that has been split from a PFX file)
- **Git/Source Root Folder, Output Folder, etc.** : source directories for various Mentor processes and output. Shows on local installs, but not present on hosted installations
- **Git Username** : username to GitHub or other service
- **Git Token** : generated *token* which is used as a password to authenticate. See GitHub docs to generate a developer access token
- **Exclusion** : excludes files or folders as designated, and can be provided in the form of a comma separated list—will be split into parts with ',' as split char. Folder Exclusions are for the 'directory' part of any file. A "?" matches any single character, and "*" matches any range of characters. Exclusions for the "file" setting are for the filenames specifically, therefore forward and back-slashes should not be used for that pattern (as it will return no matches), and the wildcards "?" and "*" are treated the same as for folders; setting this globally will be applied to added repositories
- **Tab Size** : number of spaces per tab (default is 2)
- **Auto Update Interval** : sets the time in a format before inference will again—minimum is currently 1 day. Format supported: 0-9d|day|days
- **Git Lag Log Threshold** : internal Git diagnostic setting; a number (in seconds) that will create a log for the various git commands
- **Title** : sets a page title for this instance the Mentor Service as seen on the homepage on the Navigation bar
- **Links** : This allows custom text and links to be displayed under the first line of the menu bar; the syntax is similar to markdown, the format being: `[display text] (http-link) [text2] (http-link2)`. Note: for external links, a full path including "http://" is required
- **Hosted Service** : a toggle which turns on hosted-specific features, such as hiding local filesystem full paths and certain options
- **Users Per Page** : controls the number of users displayed on the Users configuration tab
- **All Files** : if checked, the files view will display all files inferred, not just ones with Issues (anomalies and notables)

- **Public Insights** : sets the default privacy option when retrieving Insights
- **Prefetch Insights** : this option control if Insights are prefetched during background cycles or if they are fetched on-demand when an Insight is selected
- **Share Votes** : this option controls if Insights feedback votes are used to inform Merly's foundation model (FM)
- **Enable Debug Logs** : check to enable debug logs (verbose output for commands), found in the /logs subfolder
- **Detailed Score Reasons** : Special debug option not available in every release; gives detailed reasoning on Mentor Score
- **Service URL** : used as the public accessible URL to the service. If set, welcome email and password reset options become available. (additional configuration beyond the scope of the current mentor is required.)
- **Using Merly SMTP Server** : if checked, email notifications (such as new user creation) will generate and come from Merly's server
- **SMTP Host** : URL of the smtp server
- **SMTP Port** : port of the smtp server
- **SMTP User** : which mailbox authorizes the email notification. Format: valid email address
- **SMTP Password** : password for the SMTP server
- **SMTP Sender** : name of the sender (i.e. Merly Inc. No-reply)
- **SMTP From** : which mailbox the email comes from. Format: valid email address
- **SMTP Reply To** : which mailbox will be addressed when replying. Format: valid email address
- **SMTP Use TLS** : selects whether to use secure protocols (this depends on your settings)
- **Color Template** : this is for a future feature to allow user customization for color templates

Additional page elements:

- **Save**: saves current configuration settings
- **Clear Cache**: the cache is an internal feature used to speed up the system, and clearing this can be used to troubleshoot odd behavior; this button will clear the cache—which re-generated when the service restarts.

Note on SSL settings: These settings are needed if one wants the Merly Mentor service to only be available via SSL. If the SSL has a password, Merly Mentor will ask for it while starting, via openssl.

5.4 Tags and Filtering

5.4.1 Sorting and Filtering on the Home Page

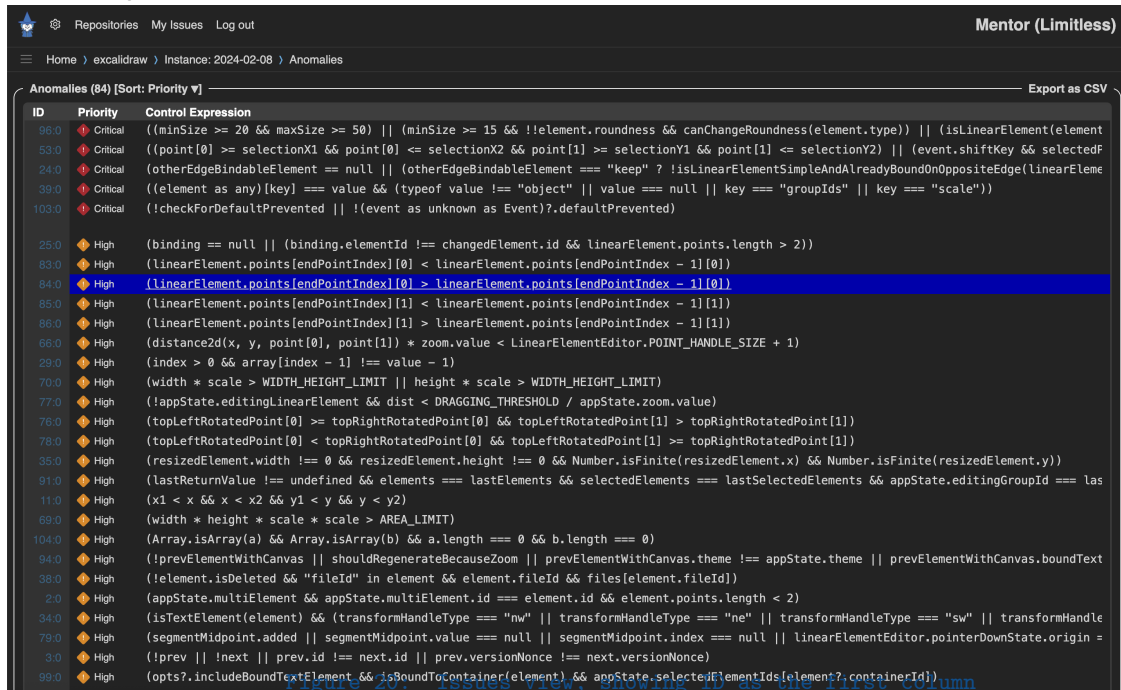
Each repo supports the use of tags for organization and sorting. These are configurable per repo and utilize the **Links** portion of the Configuration. Both the /leaderboard and /repo/list pages can be filtered by appending `?tag=tag1` at the end of the URL.

Additional arguments can be passed by appending an `&`, such as name and language to filter it even further. In this example, the link will filter all the added repos and only display ones matching the tag "tag1" and repos containing "ex" in their name, but also which have been identified as being C project:

```
localhost:4200/repo/list?tag=tag1&name=ex&language=c
```


Custom links for these filters can be set in the Configuration. See [the section on Configuration](#) for more information on the format of Links.

5.5 Repo Issues and Quick Actions



In many views (code, anomaly, issues, etc.), a list of "Repo Issues" are listed. In the leftmost column, each of those are assigned an ID. Hovering over the ID number displays quick actions in the dialog. In the hover menu, clicking or tapping the edit icon (✎) will allow you to assign a task, categorize it, and comment. Once the fields are filled out, click or tap the save icon (💾) in the top right corner of the hover box. These fields are drop down menus (which require another click or tap), or short response form (like for the Comment field).

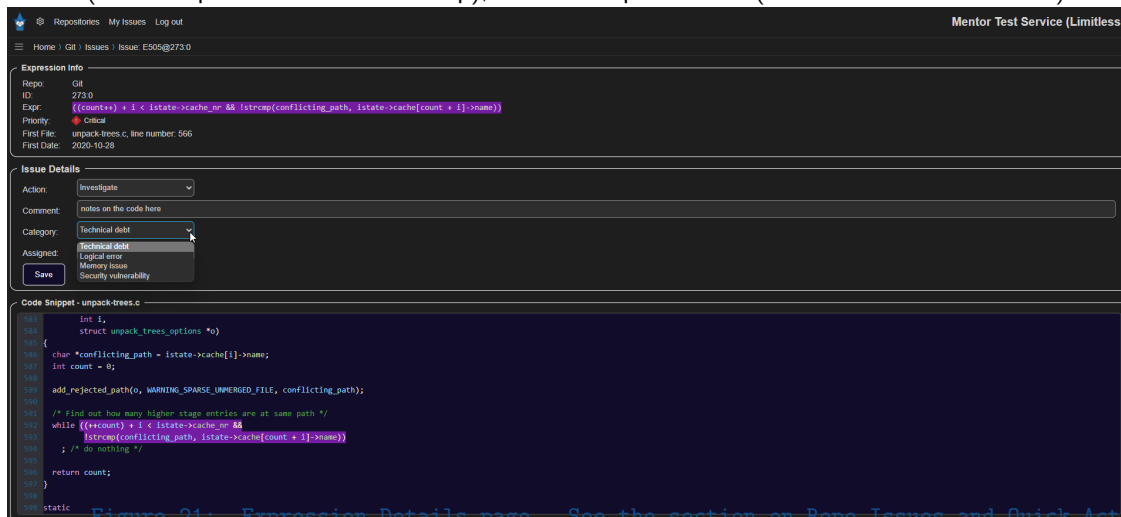


Figure 21: Expression Details page. See the section on Repo Issues and Quick Actions

In the Expression Details box, there are user-definable fields (see Figure 21):

- Action options can be one of the following:
 - "Ignore"

- "Investigate"
- "Simplify"
- "Fix"
- "Potential Crash"
- "Potential Security Issue"
- **Comment:** allows for a comment to be made
- **Options for the Category** can be one of the following:
 - "Technical debt"
 - "Logical error"
 - "Memory issue"
 - "Security vulnerability"
- **Reporter:** user to report (from list of all created users)
- **Assigned:** assigned user (from list of all created users)

In addition to the *Quick Action* hover menu, clicking or tapping the ID number will bring up the "Expression Info" and details page. Contained within is the information, such as filename and line number, as well as first date it's seen in the repo. A score is issued, along with a class and cost (see Section ?? for more information), along with a code snippet. From here, the same kind of categories, notes, and assignments can be made.

A note about permissions: a *developer* cannot remove a repo issue by selecting an empty action or empty assigned user.

5.6 Managing Users

Users can be managed via the web GUI. User modifications require an access level of *owner* or *admin*.

5.6.1 Adding Users

1. To add a user, log in as the first user, click or tap the gear icon (⚙️) in the menu bar.
2. Navigate to the "Users" section. A current list of users will expand.
3. Click or tap the plus icon (+) icon for a context box with new user information.
4. Fields:
 - **Name:** the name of the user, e.g. "John Doe"
 - **Email:** the full email address, e.g. "john.doe@domain.com." Mentor will indicate if the email address has been used before.
 - **Role:** the drop down menu allows selection of roles: Owner, Administrator, Lead, or Developer.

The `Send welcome email` button will then create the user and send them a welcome email with a unique link to log in for the first time. During this first log-in, the user will have a chance to set their password, or use a generated one.

5.6.2 Users List

Under the *Users* configuration section, there is a table with a unique generated ID, the role the user was registered with or assigned, and online status (which indicates if a welcome email has been sent, if they're inactive, or if a user is active and online).

5.6.3 Editing Users

Here in the same Settings submenu, navigate to "Users" and expand the section. In the information table, you can view user status, role, emails, names, and unique IDs. Click or tap on an ID to edit a user. The same fields appear as when creating a user. A user can change their password, but when editing another user, if a user has a password set, it will appear as "***". Setting or erasing it will subsequently set or erase it. If you need to re-generate the password, you can click or tap that button. Additionally, to have the user reset their own password, use the Reset Password button to send them an email similar to the welcome email which will let them set a new password.

5.6.4 Deleting Users

In the same Users submenu, after opening a user to edit, click the Delete (🗑️) icon at the bottom of the context window. **Note you must confirm the deletion with the confirmation dialog.**

5.6.5 Resetting a Password

It is possible for a user account password to be reset. If a user has forgotten the password, there are methods to reset or recover the information.

Note: The Service URL field needs to be set for the "Forgot?" reset link to appear on the log in screen. For local installations, this must be set to `localhost:<port-number>`.

Two methods can be employed:

1. A user can request a password reset:
 - (a) On the "Login" page, click or tap the "Forgot?" button

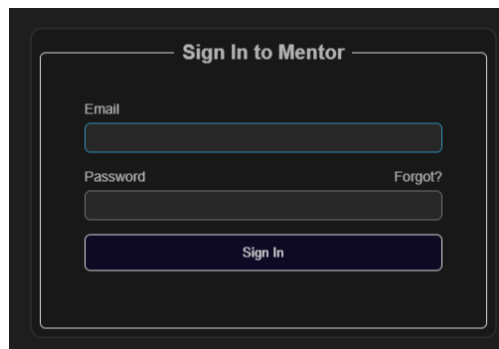


Figure 22: The log in screen with the 'Forgot?' button present.

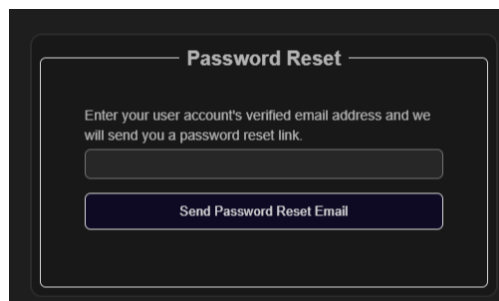



Figure 23: A user will type the email address associated with the account.

- (b) In the top field, type the email address associated with the user account.
 - (c) Click or tap the the **Send password reset email** button. If that user and email combination exists, an email will be sent.
 - (d) In that email, a "Reset Link" button will be provided. Click or tap the button and the user will be redirected to the Mentor Service, and able to set a password.
2. A user account with sufficient privileges (by default: Admin and Owner roles) can send the user a password reset email by performing the following steps:
- (a) Navigate to the Configuration settings (the  icon in the Navigation bar)
 - (b) Expanding the "Users" section by clicking or tapping the header
 - (c) Click or tap the user you would like to edit
 - (d) Perform one or more of the following actions:
 - Set a password by typing it into the Password field.
 - Automatically prepare and send a password reset email by clicking or tapping the Reset Password button.

6 Navigating the Mentor Service

This section will highlight the various pages found within the Mentor Service and describe the different ways of navigating it.

6.1 Log In Screen

Present on the **log in page** (Fig. ??) is:

- Home Page button (🏠) - when logged in, this will take a user to the homepage, the Leaderboard view
- Sign In button - this link in the navigation bar opens the user sign in dialog box
- Links bar - this is configurable in the Configuration settings

On the **log in dialog box** (Fig. 24):

- User Name - this field takes the user's name, username, or email address
- Password - this field takes the user's password

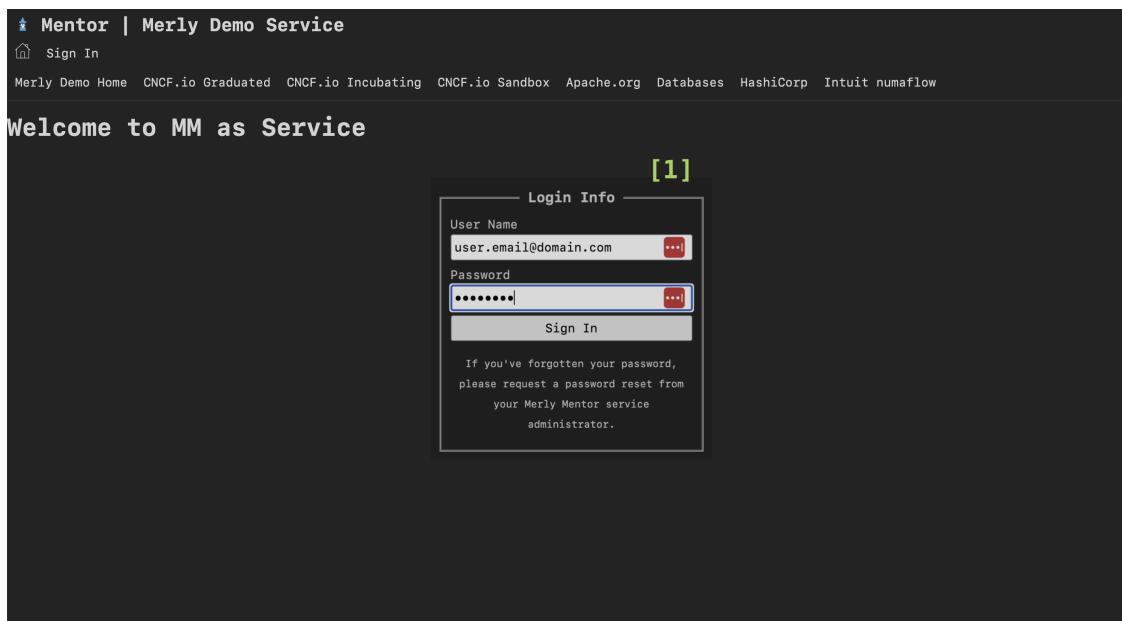


Figure 24: The log in dialog box.

6.2 Repo Lists

There are various ways to view what repositories are on the service.

Looking at the **Homepage (Repository List)** (Fig. 25), you will notice things like:

- 1 - The main navigation bar at the top
- 2 - Links bar: this is user-settable in the configuration settings
- 3 - A search bar and filter button
- 4 - The "(+) Add Repository" button which will open the dialog box to add a repository

5 - List of all repos on the service

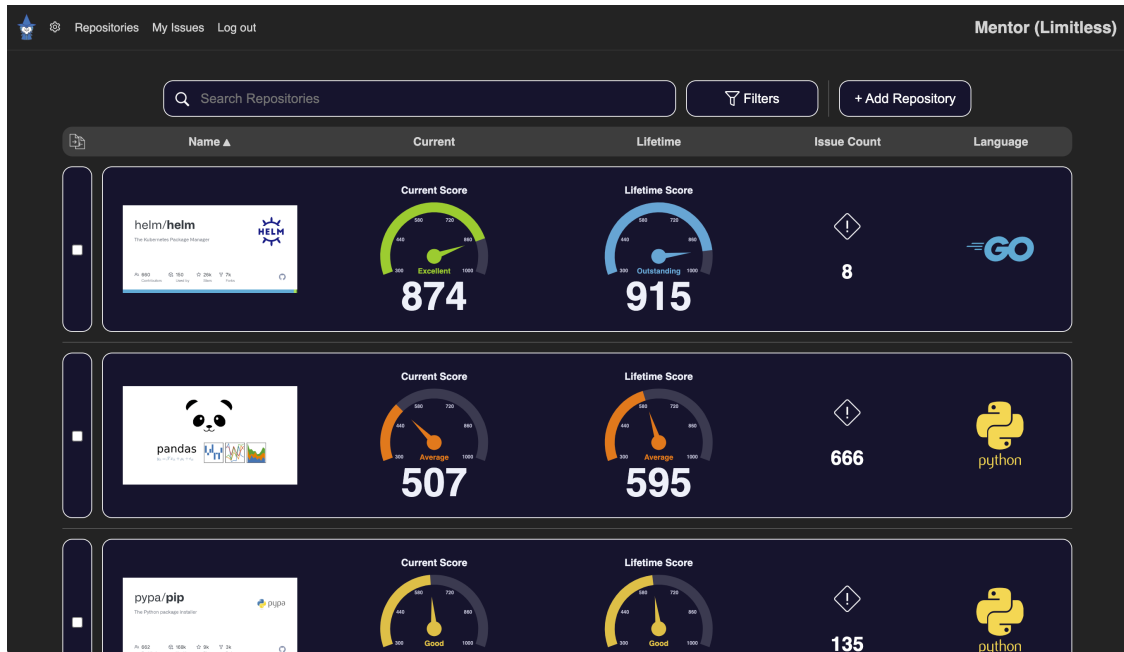


Figure 25: The repository list view, or homepage.

7 Mentor Service: Mentor Summaries

Mentor Summaries provides the following key behaviors:

- (i) it scores your repository's short-term and long-term quality,
- (ii) it provides analytical feedback on temporal trends over the lifetime of the repository, and
- (iii) it generates actionable feedback for users to consider in multiple areas.

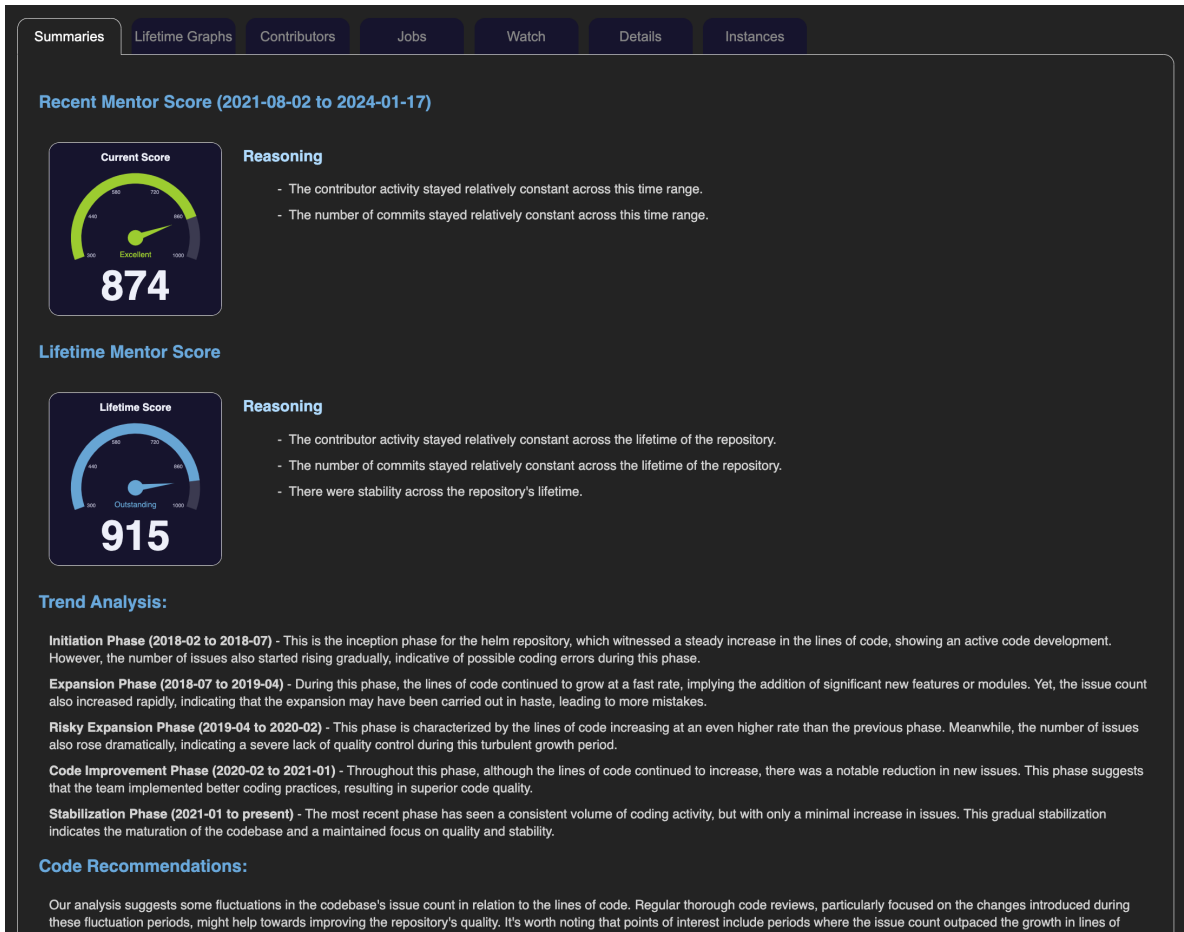


Figure 26: An example of Mentor Summaries

7.1 Overview

Mentor Summaries produce a report for any repository with enough data in the Mentor Service. These sections are:

Activity Scores

This score ranges from 0 to 1000—anchored by heuristic metrics and driven by AI. Summaries will score both the recent activity and the lifetime activity of a repository and give reasons as to each, using both metrics from Merly Mentor as well as generative AI support. “Recent Activity” uses the most recent 24-month (2 year) time span. The “Lifetime Activity Score” refers to the entire history of the repository.

Trend Analysis

Summaries will separate the history of the repository into discrete phases, such as “Organic Growth,” “Stabilization,” “Risk Mitigation,” or other phases. These are determined by metrics *like*:

- Patterns in lines of code changes
- Issue removal and introduction rate relative to codebase size
- Issue scores and severity over time
- Calls attention to quiet periods that were retrospectively optimistic

Code Recommendations

Summaries informs the design process by calling attention to periods where changes created or worsened issues.

Contributor Recommendations

Summaries guides project planning by correlating Mentor-collected and analyzed contributor activity with feature introduction and noticeable changes in code quality.

Potential Consequences of Inaction

This section makes a best guess for potential challenges and risks the codebase faces without remediation and proactive efforts.

Learning From the Past

When applicable, Mentor references previous struggle periods to reinforce lessons learned during development and to prevent repeating what is determined to be more volatile periods of development

7.2 Accessing Mentor Summaries

To access, view, and interact with Mentor Summaries, from a list of repositories in the Mentor Service, click or tap a repository to navigate to the repo’s detail page.

On this repo page is a chart of the repo’s lifetime history, author and contributor data, Mentor Summaries, the repository’s configuration settings, and list of inferred references.

Necessary Inference Points Needed Mentor Summaries will generate when there are at least ten (10) inference points.

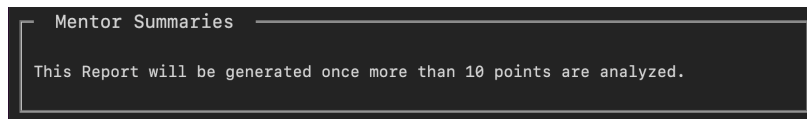


Figure 27: The message that appears when Mentor indicates more points of inference are needed to generate Mentor Summaries.

Regenerating the Report Mentor uses a rubric of scoring heuristics as well as generative AI into produce the report. To regenerate the report, click or tap the “Regenerate Report” button. This can be helpful if more points of inference have been added since the last Summaries was generated.

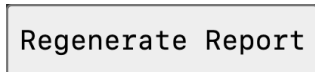


Figure 28: This button will cause Mentor to regenerate the report.

8 Mentor Service: Mentor Insights

Mentor Insights scores your repository's health, reasons about trends, and provides customized improvement strategies tailored to your specific software needs based on a lifetime of analysis.

8.1 Overview

Mentor Insights will analyze an anomalous or notable expression and provide insight into why it was identified and suggestions to possibly improve it. These sections are:

8.2 Accessing Mentor Insights

To access, view, and interact with Mentor Insights, first, identify an issue (an expression identified as an anomaly or notable) on the repository's "Issues" page, or on a specific inference instance's "Anomalies" (including the "New" filter) or "Notables" page.

Figure 32 shows this view. In this figure, item number [1] shows the various ways to navigate and view Issues. Item number [2] shows an identified anomaly. Hovering over this will show a snippet of the code in context. Clicking or tapping the anomaly will navigate to Mentor Insights.

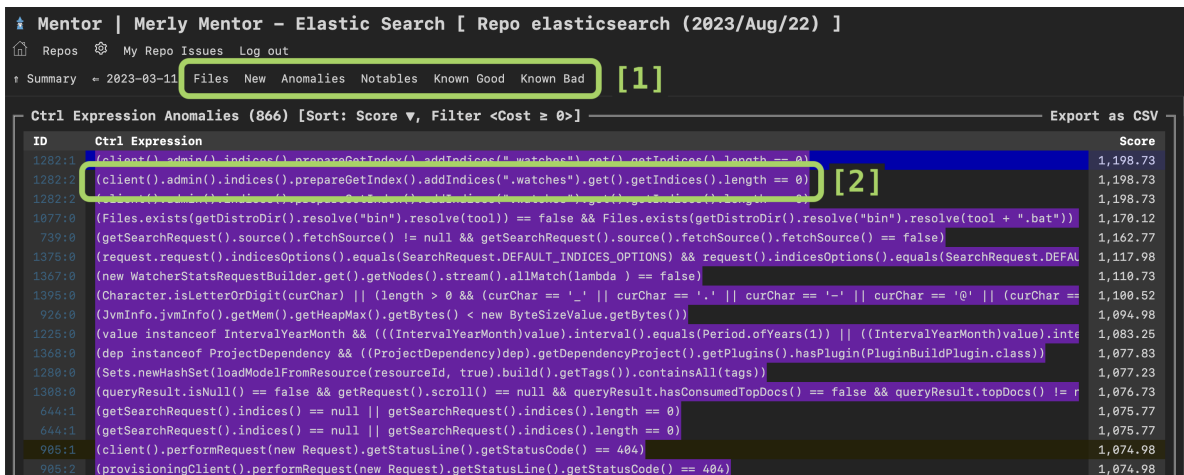


Figure 29: An Issues view on an example repo's inference instance.

8.3 Browsing Insights

When browsing Mentor Insights, the page is split into two halves, left and right. Figure 30 shows this. On the left, [1] is the expression and the context from the file itself. The right side of the page, number [2], is the Mentor Insight itself. The Insights Panel (shown at number [3]) navigates the pages of Insights and controls the various privacy options.

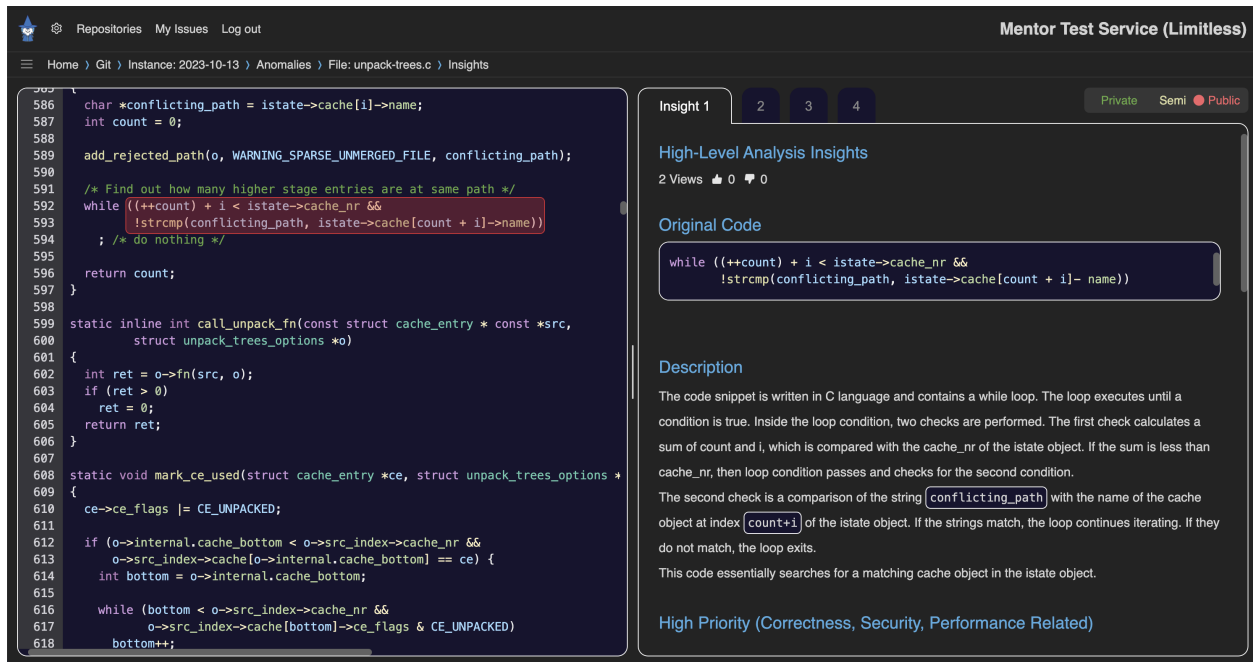


Figure 30: This is an example of Mentor Insights.

8.4 The Insights Panel

The Insights Panel controls navigation between various pages and approaches for Insights, the size of the Panel itself, as well as privacy controls.



Figure 31: The Insights Panel

- 1 - The tabs navigate between the different pages of Insights. Each page could be a different approach and provide various suggestions (there are currently 4 to browse)
- w - The white bar in between the panels will adjust the size left and right of the Insights and code context panels.
- 4 - The privacy controls allow for expression privacy.

Insight Feedback Using reinforcement learning from human feedback, giving the recommendation a thumbs up informs the foundation model which answer is best and will be used for future Insights.

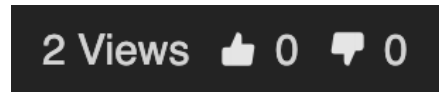


Figure 32: Reinforcement learning by voting. These icons are part of the Panel, found in the top right corner of each page of Insights.

In addition, both up and down votes will show for every user and help move the insights that developers have found most useful or appropriate to the front for everyone on the Service to see.

8.5 Privacy Settings

Mentor Insights provides for fine-tuned control from within the Panel on how each expression is processed.

There are three levels of privacy:

- **Private:** Mentor removes all syntactic details of the code, but keeps in place the structure for analysis by Mentor. Insights analysis on this setting is done completely within the service, behind your company's firewall.
- **Semi-private:** the structure of the expression is preserved, like in the private Mode, but uses Merly's servers for analysis.
- **Public:** the original code fragment is used sent to Merly's foundation model system which provides rich and detailed Insight analysis.

Notes:

- The default setting is **private**.
- Default Insights privacy can be set on a repository level from within the repository's configuration settings
- Feedback given through voting (thumbs up and thumbs down) can be controlled in the repository's configuration settings under "Share Votes."

9 Mentor Service: User Privileges

The privilege system exists to allow the view and edit access to various components of the Mentor Service. These are managed on a per-user and per-repository basis.

9.1 Managing User Privileges

To view and edit user privileges, click or tap on the gear icon (⚙️) in the navigation menu. Next, open “Users” by clicking or tapping the corresponding tab.

In the Users section, clicking or tapping one of the users will allow for a user’s default privileges to be modified (if the editing user has the privileges themselves).

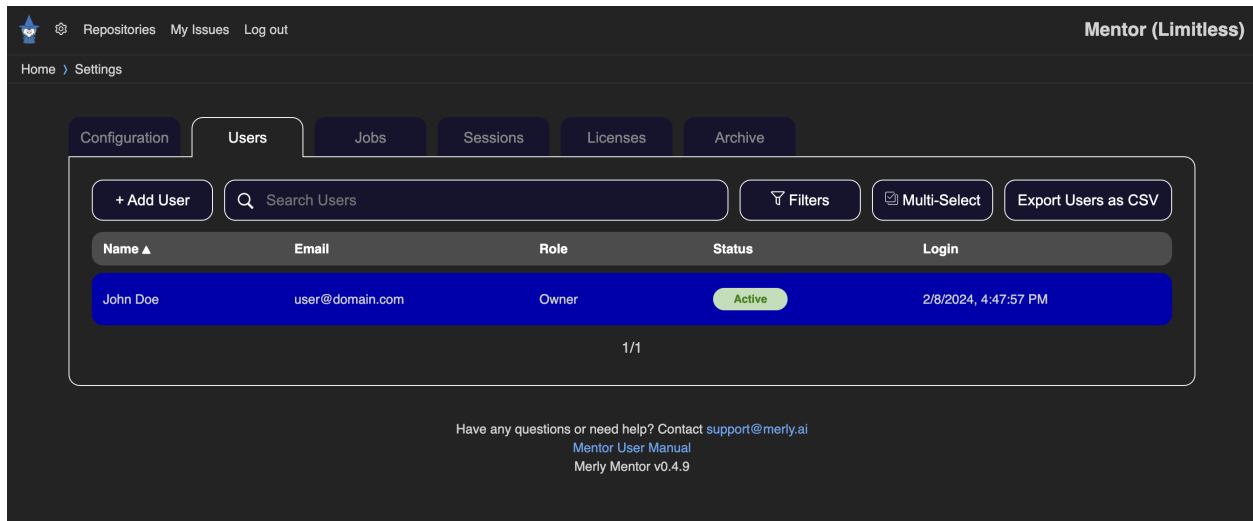


Figure 33: Navigating to the User configuration settings.

9.1.1 Default Roles and Privileges

By default, Mentor defines privileges by role, though these can be changed on a per-user basis. The four assignable roles in the Mentor Service are:

- Admin
- Owner*
- Lead
- Developer

*This is also the first user of the system, typically the license holder, is assigned this role by default.

9.1.2 Overview of Privileges by Role

As an overview, here is a breakdown of default privileges exist for each defined role, and which ones can be modified with a toggle in the user settings. Current discrete privileges are:

Current User's Privileges

- Use default role privileges:

Repository-Related Privileges

Modify (Repositories Added by This User)

- Quick add repositories:
- Advanced add repositories:
- Edit repository configurations:
- Delete repositories:
- Delete repository instances:
- Assign issues:

Jobs (Repositories Added by This User)

- Run git clone and fetch jobs:
- Run infer and rerun jobs:
- Run infer points jobs:
- Regenerate Mentor Summaries:
- Abort jobs:

Modify (Any Accessible Repository)

- Edit repository configurations:
- Delete repositories:
- Delete repository instances:
- Assign issues:

Jobs (Any Accessible Repository)

- Run git clone and fetch jobs:
- Run infer and rerun jobs:
- Run infer points jobs:
- Regenerate Mentor Summaries:
- Abort jobs:

Contributors

- View contributor data:
- View contributor recommendations in Mentor Summaries:

User-Related Privileges

- View all users:
- Add users:
- Edit users:
- Delete users:
- View user privileges of all users:
- Edit user privileges of other users:

Settings-Related Privileges

- View Mentor configuration:
- Edit Mentor configuration:
- View languages:
- View jobs:
- View sessions:
- View licenses:
- Add licenses:
- Regenerate summaries for all repositories:

Figure 34: User privilege list.

Note: Developer user accounts can only change the assignment, not close an assigned issue.

9.1.3 Editing User Privileges

By default, admins and owners have the meta-privilege of editing privileges, as these user roles have default access to the entire service. Privilege settings are enabled by checking or unchecking the checkbox while editing a user. Changes will take effect when this dialog box is closed by clicking or tapping “Save.”

Currently, the settable user privileges are:

Allow user to run repository actions and regenerate Mentor Summaries

When this privilege is enabled, users are allowed to add repositories from the Repos page, edit repository information and configuration settings, delete repositories, and run or re-run inference.

Mentor Summaries are also available to be viewed and regenerated.

Allow user to view contributor data charts

This privilege setting allows for these charts to be viewed, browsed, and downloaded. As part of the two On each repo’s detail page there are two charts with project contributor metrics:

On the repo detail page:

- Author Summary Chart (LoC per contributor over time)

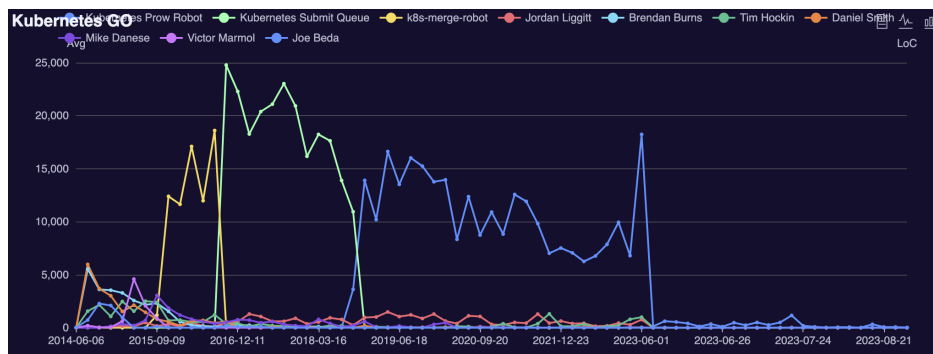


Figure 35: Example Author Summary Chart

On the repo detail *and* inference instance page:

- Author Chart (LoC per contributor since last inference point)

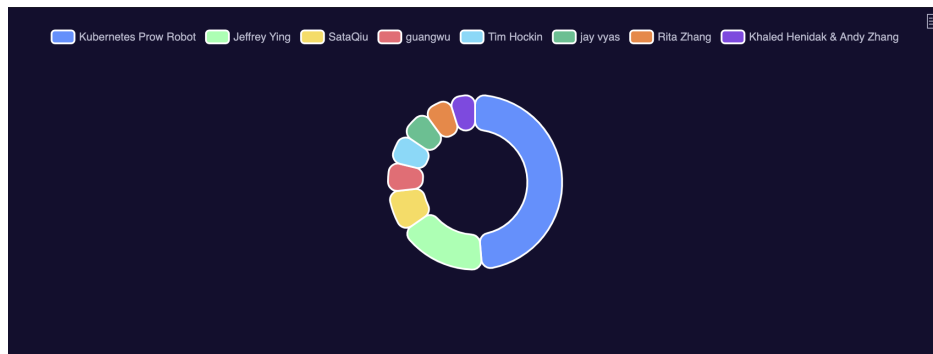


Figure 36: Example Author Chart

Allow user to view contributor recommendations in Mentor Summaries

Mentor Summaries provides detailed information after analyzing the entire the lifetime of a repository. Part of this report is a section on contributor impact and recommendations. This privilege setting shows or hides this section.

Notes:

- These privileges are also checked against repository permissions. It is possible, for instance, for a user to be able to run inference (with the “modify repo” privilege enabled), but not have repo access, thus not allowing them to perform actions in that repo.
- If the **Use Default Role Privileges** option is checked, the other options become disabled. If there is anything checked or unchecked, the “default” option overrides it.
- It is possible to make fine-tuned adjustments to a specific User’s privileges. The default checkbox must be unchecked, and then various privileges can be enabled or disabled.

9.2 Managing Repository Privileges

Repositories can also be access controlled on a per-user basis. This setting will hide or show repos on the Mentor Service’s repo list and leaderboards. Because of the access-controlled nature as well, any user with privileges to view and modify repos only apply to the ones they have access to.

These settings are not managed in the User configuration settings section, rather, on a per-repo basis. Modification of this setting is based upon the editing user account’s privileges.

9.2.1 Opening and Editing Repo Settings

From the homepage Leaderboards or repository list, navigate to a repo’s details page:



Figure 37: Step 1: Navigate to the repo’s settings.

Under the lifetime charts and Mentor Summaries are the repo details and settings. To edit, click or tap the edit button () next to the repo name—noted in 37 as number [1]. Note the default access is “Open,” shown at [2].

9.2.2 Restricting Repo Access

Now, while editing the repository, check the box (☑) to enable access control for this repository.



Figure 38: Step 2: Check the box to enable repo access control.

When checked, a new dropdown field appears with a list of users:

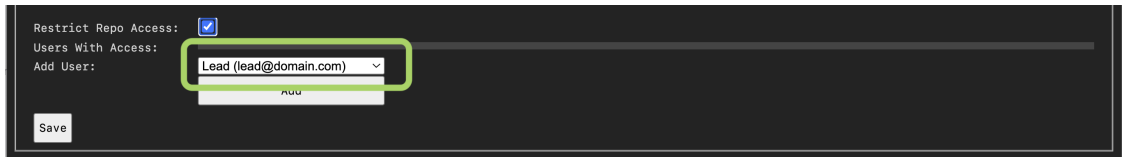


Figure 39: Step 3: Add users to the repository from the dropdown field.

Clicking or tapping a user from this list will select them. To allow them to have repo access, next click or tap the “Add” button.

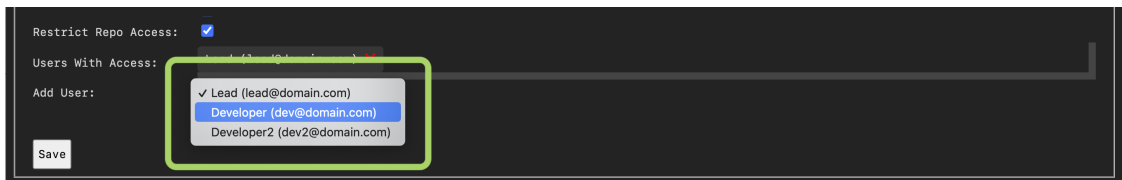


Figure 40: Step 4: Choose the user accounts to have access.

9.2.3 Saving Changes

To save the repo permissions, click or tap the “Save” button (number [3] in Figure 41). To remove a user from the repository access list, click or tap the red “X” (✘) next to their name and email address (number [1] in the same figure). To add or re-add users, click or tap the “Add” button ([2]).



Figure 41: Step 5: Save the settings.

10 Mentor Service: Exporting Data

In most places tables or other data is presented, an option exists to download a copy of the table or list. The following is the current list of places Export button is seen:

Leaderboards:

- Leaderboard List

Repo Charts:

- Score Summary Chart
- Author Charts
- Score Chart

Repo Information:

- Repository Summary
- Repo Instance Table

Issues Tables:

- Repo Issues Table
- New Issues / All Issue / Assigned Issues
- My Repo Issues

Repository Views:

- Files
- New
- Assigned Only
- Anomalies
- Notables
- Known Good / Known Bad

Configurations:

- Service Configuration
- User List


The location to where the file is downloaded can be set in your browser settings.

11 Using Mentor with GitHub Actions

Another mode of operation with a graphical user interface (GUI) is GitHub Actions. Here, we use Mentor directly in your software development repository workflow. Using this feature, you can use predefined triggers to call and run Mentor as an automatic part of your continuous integration/continuous deployment (CI/CD) process. Running Mentor with a GitHub Action (GHA) will produce results that can be viewed directly within GitHub as part of your CI/CD process.

This section will describe the set up process for GitHub Actions, how to configure the workflow, and provide an example workflow on how to get started.

Note: If you do not initially see the “Actions” link in the GitHub repository’s toolbar, navigate to “Settings,” then “Actions,” and from there, enable Actions by selecting the appropriate “Allow...” setting according to your needs and policies.

GitHub Actions Documentation Using Mentor via GitHub Actions is user configurable. For the full set of documentation and instructions on how to create and customize GHAs, refer to the [GitHub Action documentation here](#) .

11.1 Setting up the Environment

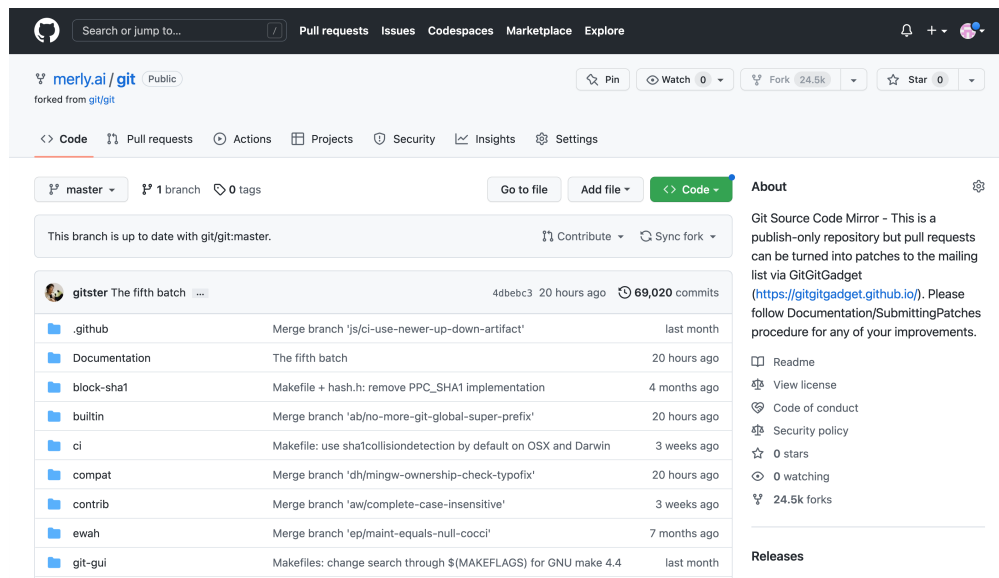


Figure 42: A GitHub repository’s project page. In the navbar, “Actions” and “Settings” can be found.

On the repository’s main page, click or tap the “Settings” tab in the navigation bar to bring up repository settings. Here, you will set a secret environment variable which the triggered GitHub Action will use and pass into the workflow.

In the left-hand navigation panel, look for “Secrets” and then “Actions.” Once you open this menu up, click or tap the green “New Repository Secret” button in the top right (as seen in Figure 44).

Set the key/value pair of this new secret to:

- Name: MM_KEY
- Secret: your Merly Mentor license key (e.g., ABCD-0123-QWER-9876)

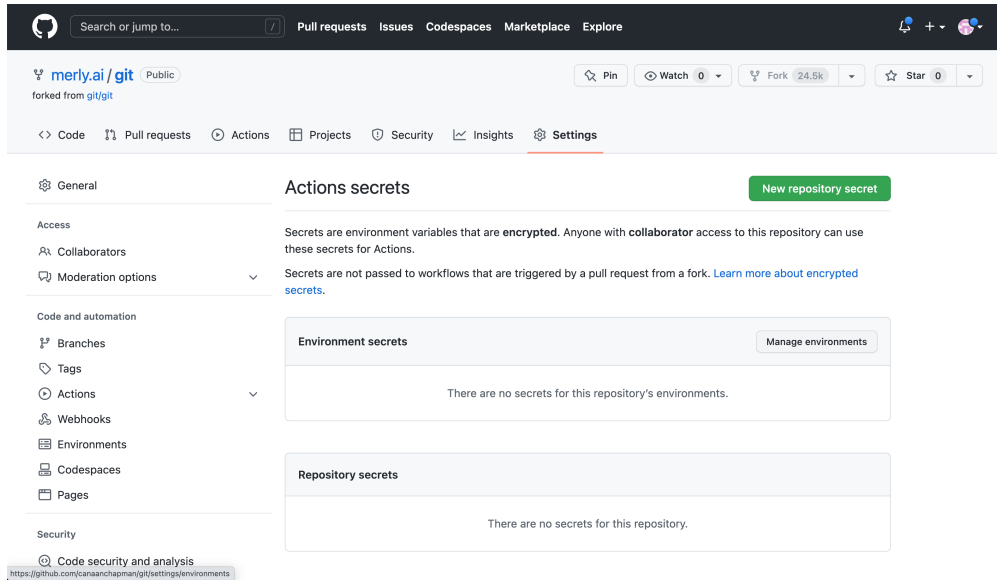


Figure 43: The settings page found under the left-side navigation menu Security > Secrets and Variables > Actions.

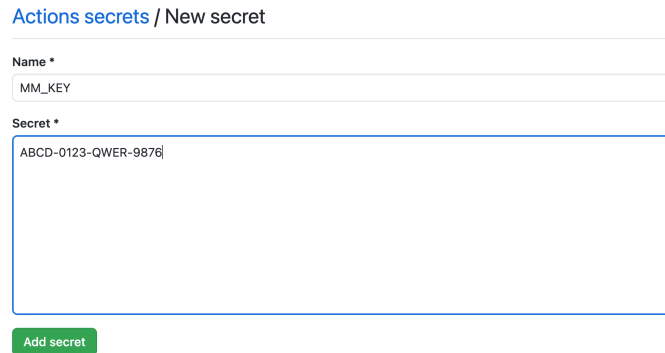


Figure 44: Adding your Merly Mentor license key as an environmental secret to be passed into the Action's container.

To save this secret environment value, click or tap the green “Add secret” button below the input boxes. You will see a notification “Repository secret added” which means it was saved successfully. You can verify this by seeing that “MM_KEY” has been added to your Repository secrets section.

Now that you've set your key, Mentor will take this variable and use it as part of the workflow.

11.2 Creating the Workflow

Now again from a project's repository main page on GitHub, click or tap the “⚙️ Actions” button in the navigation bar.

If you are working in a *forked repository*, you may have to acknowledge a dialog box from GitHub before you view or run any existing workflows. For these forked repos, you can click or tap the appropriate response in acknowledgment.

Notice that workflows are stored in the `<your repository>/ .github/workflows/` directory. This is the place from which all workflows for GHA must be stored to run.

Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.

Skip this and [set up a workflow yourself](#) →

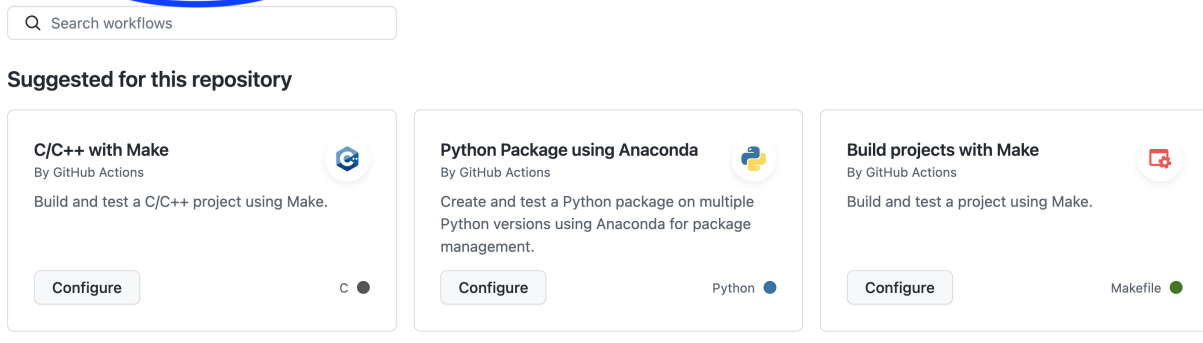


Figure 45: GitHub's repo "Actions" page with the button to set up a new workflow circled.

Once the new action page is open, GitHub will ask you to "Choose a workflow." Navigate to the link to "set up a workflow yourself" where a template will open and a file editor appears.

Start with code that specifies what will trigger the Action. Copy and paste this snippet into the file editor and then give your workflow a name.

```
name: MerlyMentor

on:
  push:
    branches: [ "master" ]
  pull_request:
    branches: [ "master" ]
```

In our example, the workflow has been named "MerlyMentor". Once the workflow is named, the "on:" section defines when the trigger will call the workflow to run.

For the full set of documentation and instructions on how to create and customize GHAs for your own needs, refer to the [GitHub Action documentation here](#) ↗.

Next, you will describe the environment and jobs in which the containerized version of Merly Mentor will run.

```
env:
  MM_Install: ../../MM-Install
  MM_KEY: ${ secrets.MM_KEY }
```

Here is the environment set up, where the container will pass the MM_KEY secret you set up in the first step. Then, copy and paste this snippet into the editor to tell GHA that you want to use the latest Ubuntu container using the "runs-on" command. In our example below, the first thing this workflow does is to get the code using one of GitHub's standard library actions, "checkout@v3" with a fetch-depth of 0. This indicates retrieving all history for all branches and tags.

The container's environment is now set up, and when triggered, will check out the code. Continuing in our example, the inference Merly Mentor performs is run in two back-to-back jobs:

```
jobs:
  inference:
    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v3
      with:
```

```
fetch-depth: '0'

- name: install MerlyMentor
  run: |
    echo "::add-mask::${{ secrets.MM_KEY }}"
    mkdir "$MM_Install"
    cd "$MM_Install"
    curl -fsSL https://raw.githubusercontent.com/merly-ai/MPCC-Universal/main/install.sh | bash -s -- -q -k "$MM_KEY"

- name: Run MerlyMentor
  run: |
    echo "::add-mask::${{ secrets.MM_KEY }}"
    cd "$MM_Install"
    ./MerlyMentor --github-actions -D ${github.workspace} check
```

First, the named job “install MerlyMentor” will run the same commands a manual installation would run to get the binaries and models downloaded, as well as passing in your license key.

Next, the job “Run MerlyMentor” will move into the installation directory and run Mentor with the command and the --github-actions flag, which formats the output in a specific way GitHub can use to overlay annotations on the repository.

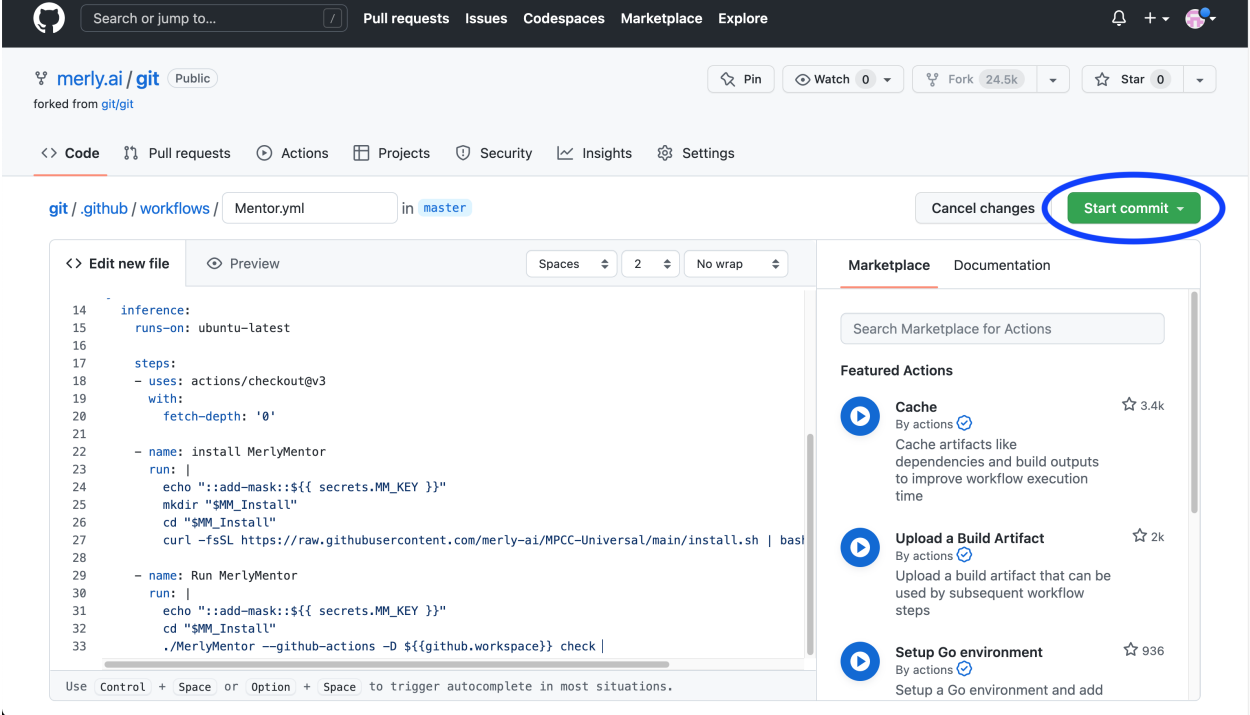


Figure 46: What the finished workflow may look like.

Put all of these together in the editor and save this .yml file by clicking or tapping the green “Start commit” button on the right side of the page.

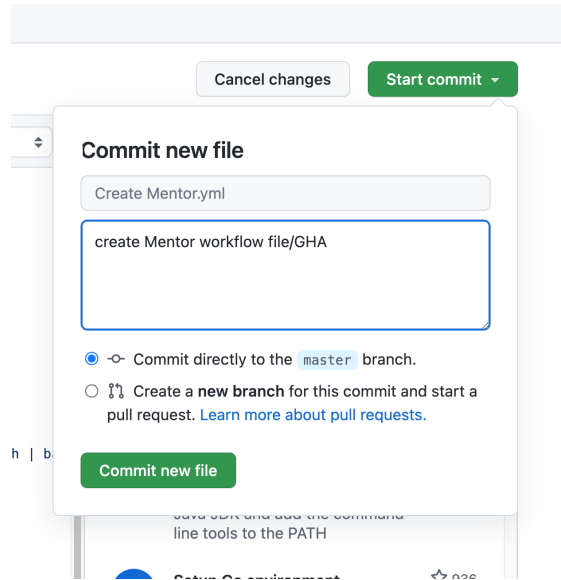


Figure 47: To save your workflow, you must commit your change by clicking or tapping the green button.

11.3 Running Mentor with GitHub Actions

When running Mentor as a GitHub Action, no manual activation is required. The setup you just performed is enough to have inference run and generate output as you update the repository.

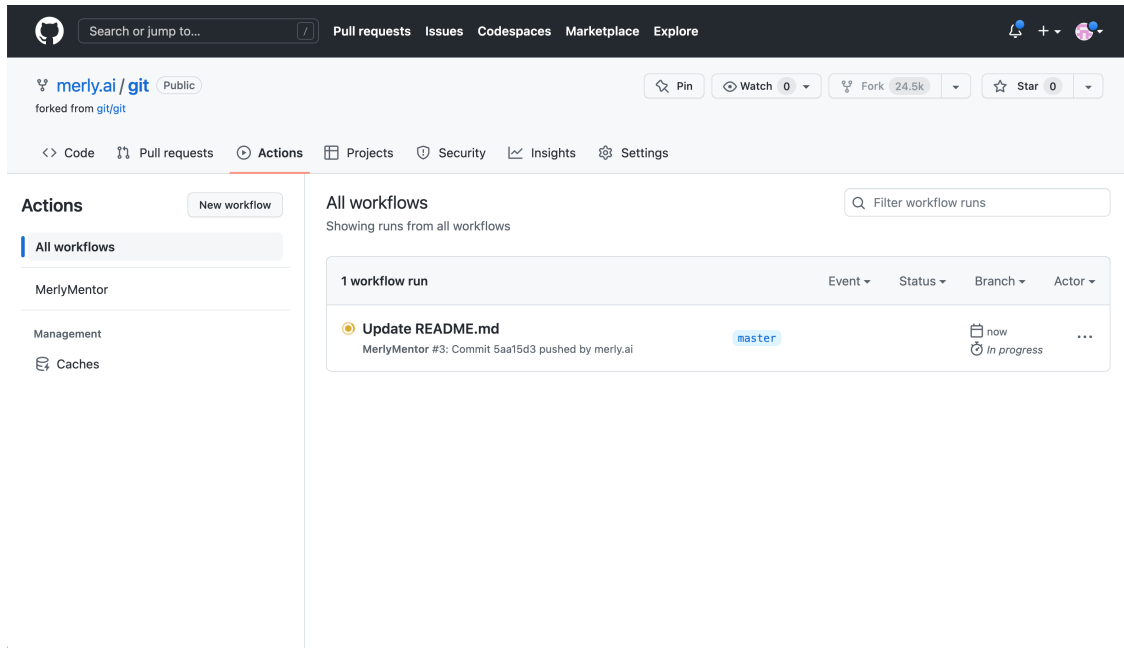


Figure 48: Actions page showing a recently triggered workflow running.

In the example script, the workflow is called on any `git push` or `git pull`. This can be configured to your preferences (refer to the GitHub Actions documentation).

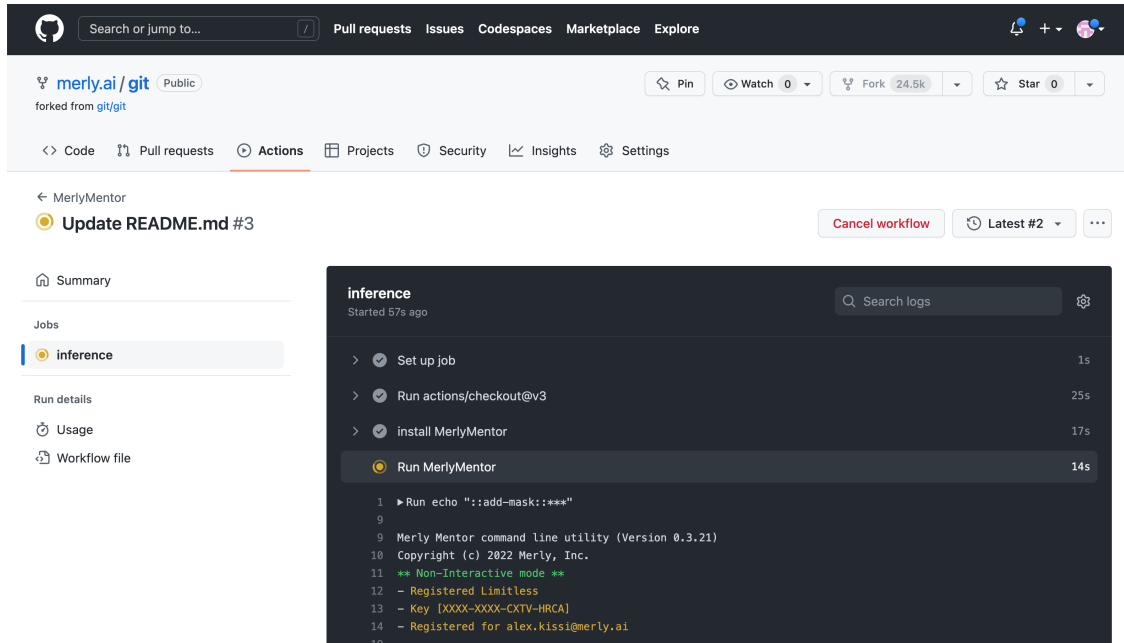


Figure 49: Monitoring the details of Merly Mentor running as a GitHub Action from within the Jobs page.

Because the system is effectively already configured and ready to trigger, you can make a test run to verify everything was set up correctly by simply making an edit in the repository and pushing a change. By navigating to your Actions, and viewing running jobs, you can view each step of the containerized process, as seen in Figures 48 and 49.

11.4 Exploring Results

When Merly Mentor is run as a GitHub action, a special flag is used, making the default output both readable and usable by GitHub. Because of this, once a workflow is complete, the job will end and the container is shut down. Results can be viewed in the Actions tab. This tab can be found under “All workflow runs”, as seen in Figure 50.

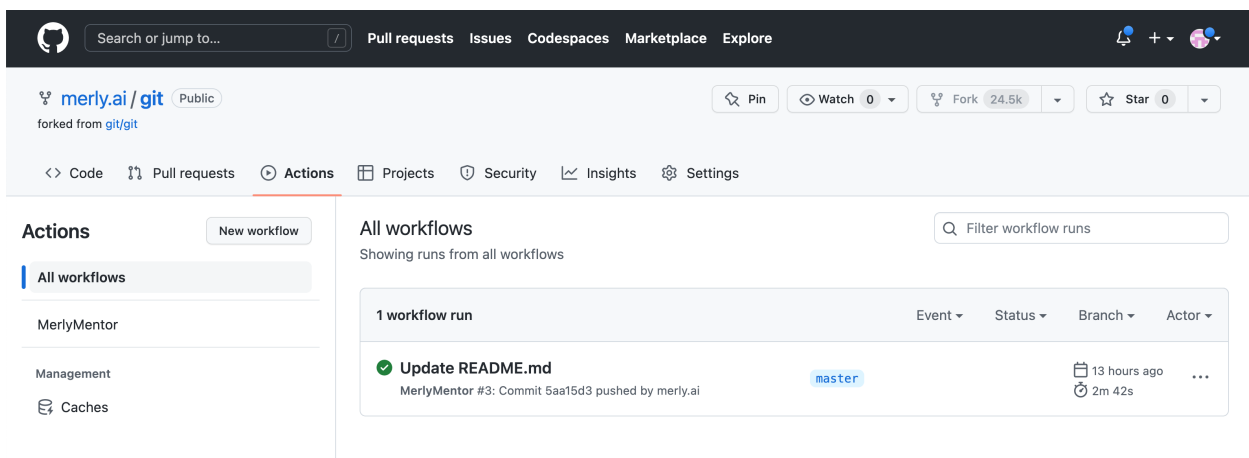


Figure 50: Workflow results and reporting page from a recently run Mentor job.

12 Mentor Service: Repository Comparison Tool

Built into Merly Mentor is a tool which allows users to directly compare projects side-by-side. Instead of viewing one set of metrics or charts at a time, choosing projects to compare is a way to view Mentor's collected metrics in a new way.

12.1 Comparison Tool - Selecting

From the homepage (also called the Repo Overview), the left-hand column, as indicated by the comparison icon (the stacked documents with arrows). Below that column are checkboxes, which can be used to select up to four (4) repositories for comparison.

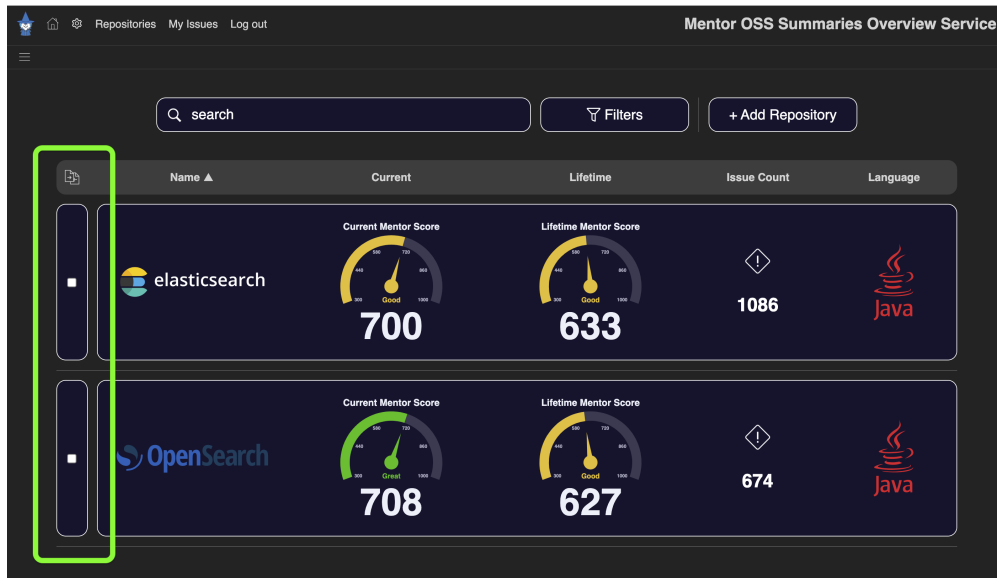


Figure 51: Mentor Compare check boxes on the Repo Overview.

12.1.1 Notes on Functionality

The steps for choosing which projects to compare is relatively straightforward. A few notes on functionality:

- Repos can only be selected only when there are no background jobs running (i.e. inference or repo repair.)
- Repos can be selected while utilizing the search and/or filter tool—the selection will be persistent
- The minimum comparison is two (2) repos and the maximum is four (4)

12.1.2 To Select and Compare Repositories

1) **Selecting the Repositories** Scrolling the list, using the search function, and/or after using the filters, look for the projects for comparison.

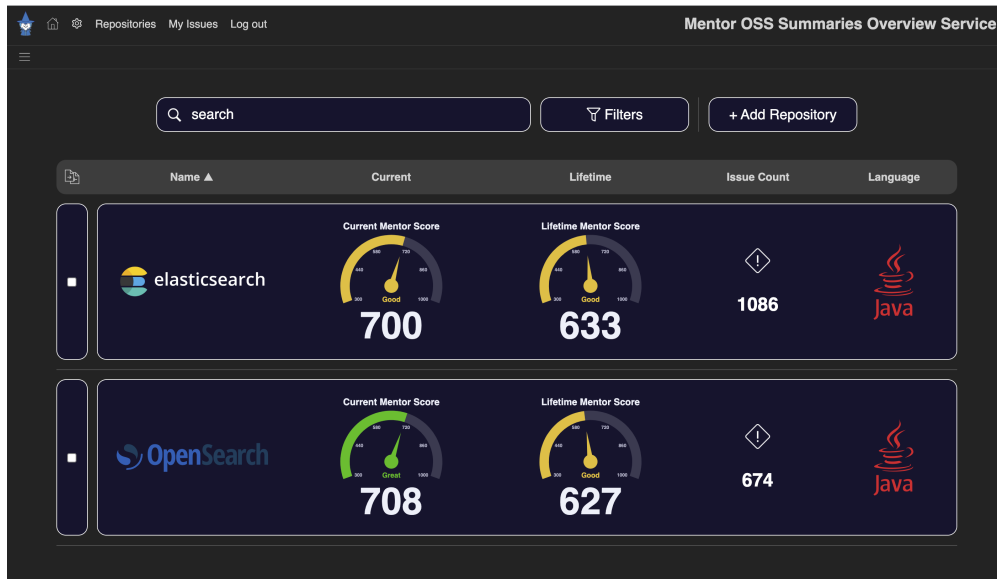


Figure 52: Home page with no repos selected.

2) **Select the First Repo** Select the *first* repo you'd like to use for comparison by clicking or tapping the checkbox in the left-hand compare column.

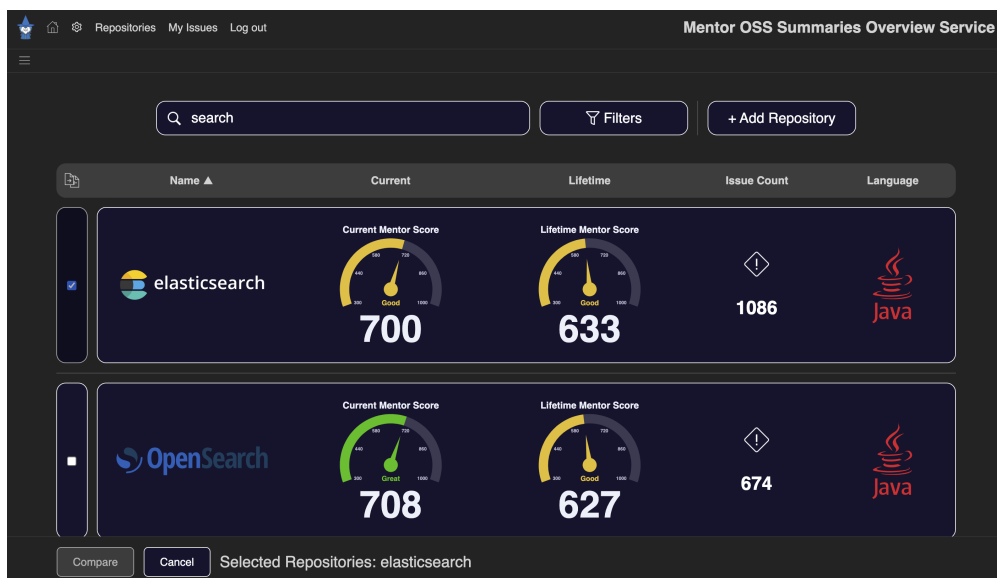


Figure 53: The first repository being selected.

Note: You can further choose up to two more repos to compare, for a total of four.

3) **Select the First Repo** Scrolling the list, using the search function, and/or after using the filters, select the *next* repo you'd like to use for comparison:

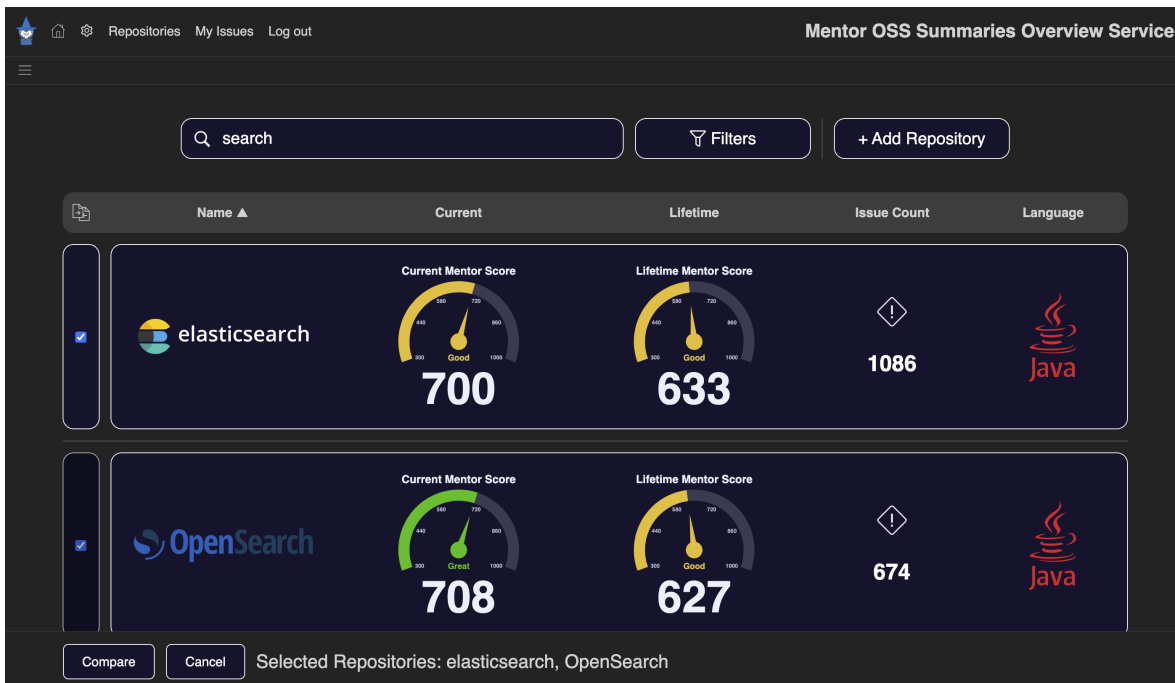


Figure 54: The next repository being selected.

4) **Select the First Repo** At the bottom of the page, click or tap the "Compare" button to load the page to compare. The "Cancel" button will deselect any selected repositories.

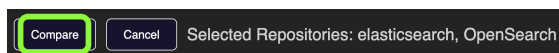


Figure 55: The Compare button will then load the side-by-side comparison.

13 Frequently Asked Questions (FAQs)

We know learning a new tool can be difficult—but we at Merly want to quickly address some high-level questions many have about Mentor. The user experience is a guiding principal at Merly.ai. If you have further questions, please let us know if there is something in this manual you'd like further clarification on!

About Merly Mentor

- **What is Merly Mentor?**

Merly Mentor can be seen as a reasoning system for code semantics. Because a significant portion of software development is understanding code, Mentor has created a tool that provides insights into development by characterizing anomalous and notable expressions, scoring them, tracking them over time, and displaying deeper Insights and reasonings.

- **How does Merly Mentor work?**

At a high level, Mentor is given a code repository where it will infer characteristics from that codebase's history and compare it to Merly's trained models. Once completed, metrics and a report will be shown to the user.

- **What does Merly Mentor look at in a codebase?**

Mentor will evaluate the abstracted syntax of expressions and compare that to Merly's knowledge base and trained models and proprietary foundation model used for Mentor Insights and Mentor Summaries. More of Mentor's features evaluate metrics based on the codebase.

- **How are expressions tracked by Merly Mentor?**

Mentor is designed to also compare software over time. Using multiple inference runs on a provided repository, Mentor will indicate any new issues as well as displaying easy-to-understand graphs which show performance over time. Mentor will also follow anomalies and indicate how many of those issues are fixed over time.

- **What types of insights will Merly Mentor provide?**

Merly Mentor's strength is not just advanced anomaly detection, but code understanding as well. Using Merly's proprietary foundation model system in addition to the anomaly detection algorithms to provide Insights—each anomaly and notable expression gets a deep analysis, plain English explanations, and code recommendations for revisions. Additional tools for insights are code metrics, repository performance over its lifetime, and scores for expression complexity.

- **Does Merly Mentor track human hours put into code?**

Currently, it does not. While there are ways to study a codebase's history—like showing lines of code and issues over time—Mentor does not track this kind of development data; it tries to help a developer understanding the code itself. Mentor does track author contributions over time though, and with Mentor Summaries, can share information about the development process.

- **What are Mentor Insights?**

Currently, it does not. While there are ways to study a codebase's history—like showing lines of code and issues over time—Mentor does not track this kind of development data; it tries to help a developer understanding the code itself. Mentor does track author contributions over time though, and with Mentor Summaries, can share information about the development process.

- **What are Mentor Summaries?**

Mentor Summaries is a report created to help analyze a repository. Summaries scores your repository's health, reasons about trends, and provides customized improvement strategies tailored to your specific software needs based on a lifetime of analysis.

Using Merly Mentor

- **What are the ways to run Merly Mentor?**

We think of Mentor in terms of different modes of operation, mainly the Mentor Service (accessed through a browser-based front end) and CI/CD pipelines with a containerized version.

- **What platforms does Merly Mentor support?**

Merly Mentor is designed to work on Windows 10 and Windows 11 systems, macOS Monterey (version 12) and above, and Linux.

- **Can I integrate Merly Mentor with my CI/CD pipeline?**

Yes—Mentor already supports GitHub Actions. In addition, in-depth Insights, analysis, and recommendations can be viewed by the Mentor Service—it will fetch the latest commits and run inference on an interval of your choosing.

- **Does Merly Mentor support GitHub Actions?**

It does! Mentor can be deployed in containerized applications like workflows in GitHub actions. Because it's designed to work well with commodity hardware, triggers can be set to run on git actions and then annotate those check-ins to provide a report to the developer of any potential issues.

- **What kind of codebase does Merly Mentor work with?**

Currently, a git repository is needed to get the best experience with Mentor's automatic classification. In addition to local repositories, GitHub repositories (both public and private) are supported.

How Mentor Works

- **How is Merly Mentor hosted?**

Mentor is flexible enough to fit any organization's software development pipeline. The Mentor application can be installed and launched within minutes to provide a simple on-premises local instance of Mentor which can be used immediately. In addition, the Mentor Service supports multiple users over a local network or wider-area network. Lastly, in certain cases, large organizations can talk to Sales and discuss a Merly-hosted option.

- **How much Merly Mentor cost?**

The Mentor ecosystem is competitively priced to tackle the issue of time efficiency. There are different tiers of corporate subscriptions: Enterprise, Enterprise+, and Limitless.

Enterprise only offers basic support and limited Mentor Insights, Enterprise+ expands this virtual calls, and Limitless puts no limits on anomaly detection, Insight, Summaries, and recommendation browsing, as well as weekly support calls. For any questions including ones about possible product trials, consult the Merly.ai website, or contact info@merly.ai directly.

14 Release Notes

Merly Mentor v. 0.4.13 Changelog:

- Updates to Mentor Watch, Mentor Score, user privileges, navigation, and bug fixes
- Performance updates, CI/CD added to Mentor Trial usage, subscription improvements,
- Many updates to the Mentor service, style, Merly Installer, and more. Full changelog will be updated in a future version.

Merly Mentor v. 0.4.12 Changelog:

- Introduction of Subscription tab and automatic purchasing and upgrading from a Trial in global settings
- Many updates to the Mentor service, style, Merly Installer, and more. Full changelog will be updated in a future version.

Merly Mentor v. 0.4.11 Changelog:

- Merly Installer updates, improvements to job and the Job Queue, bug fixes, and Mentor Watch updates
- Various other improvements to many other systems

Merly Mentor v. 0.4.10 Changelog:

- Introduction of Merly Mentor Trial
- Many updates to the Mentor service, style, Merly Installer, and more. Full changelog will be updated in a future version.

Merly Mentor v. 0.4.10 Changelog:

- Various other improvements to many systems
- Many updates to the Mentor service, style, Merly Installer, and more. Full changelog will be updated in a future version.

Merly Mentor v. 0.4.9 Changelog:

- Various other improvements to many systems
- Many updates to the Mentor service, style, Merly Installer, and more. Full changelog will be updated in a future version.

Merly Mentor v. 0.4.8 Changelog:

- Many updates to the Mentor service, style, Merly Installer, and more. Full changelog will be updated in a future version.

Merly Mentor v. 0.4.7 Changelog:

Mentor Service:

- “Issues/10k LoC” is now “AIL” (Average issues per n LoC)
- “Score/10k LoC” is now “SR” (Score Ratio)
- Lifetime Overview default series changed to: AIL, SR, LoC
- Improvements to charts and graphs display
- Hover overlay on charts can be toggled on by pressing the “control” key
- Minor visual improvements to charts
- Charts now have titles and improved legends
- Updates to colors on charts
- New “Administration” settings panel
- Regenerate All (Mentor Summaries) button added
- SMTP settings updates w/ Merly API and emails
- Improved verbiage for instructions on Welcome and Password Reset Request emails
- New “Add Repository” button
- Improved how Mentor handles malformed Git URLs
- All elements on projects column clickable
- Many major and minor CSS updates

New feature: Repository Dashboard

- New dashboard on repository overview page
- Dashboard indicators link to appropriate views
- New gauges and animations
- New display of repository metrics
- Many minor improvements for display of page sections

Performance:

- Improved rendering for Files View
- Improved Repo Summary rendering by reading file count from the summary table
- Improved performance with more efficient use of DB requests
- Sorting by date performance improved
- Various fixes to jobs queue
- Fixed some instances where jobs will get stuck in aborting status
- Improvements to inference dialog loading times

Mentor Summaries:

- Updated Summaries layout with improved Mentor Score display
- Mentor Summaries sections redesigned
- Updated Summaries heuristic scoring ranges and grades
- Updates to the colors used to display scores
- New functions to allow for grades/scores to be decoupled from “Regeneration”
- Trend Analysis section alignment improved
- Minor Summaries section layout improvements

Mentor Insights:

- Font size and readability improved for Mentor Insights

Merly Mentor v. 0.4.6 Changelog:

- Improvements to Mentor Summaries
- Improvements to user privilege system
- Bug Fixes
 - Repository list now sorts alphabetical regardless of case
 - Greatly improved performance on how the My Repo Issues page was being queried and rendered
 - Minor UI updates to align elements

Merly Mentor v. 0.4.5 Changelog:

- New feature: User Privileges - Can now enable or disable certain actions on a per-user basis (in addition to default role privileges)
- New feature: Repo Privileges - Privileges can be assigned to each user to control the level of access they get.
- New Feature: Automatic password reset request
- Bug Fixes
 - Fixed an issue with users with same name colliding; better handling of login function with names and emails
 - Repository list now sorts alphabetical regardless of case
 - Greatly improved performance on how the My Repo Issues page was being queried and rendered
 - Minor UI updates to align elements

Merly Mentor v. 0.4.4 Changelog:

- New feature: Mentor Summaries
- Graph “Issue” line is now red
- Charts improved how numbers are displayed when hovering
- Instances table is now sortable

- My Repo Issues is now sortable
- My Repo Issues now show which repo the issue comes from
- Issues table is now sortable
- Number of concurrent background tasks is configurable
- Queued jobs can now be aborted in the Jobs section of the configuration page
- Repository information page rearranged to account for Mentor Summaries (Repo configuration details, action buttons, moved down; contributor charts below that)
- JSON configuration and output converted to database
- Repo home page changed to Leaderboard style
- Add repo (+) icon moved to top of the repo page
- Bug Fixes
 - Names no longer block certain elements of the author chart
 - Repo descriptions now properly wrap lines on the leaderboard page
 - Fixed error regarding missing git credentials on certain repository actions
 - Updated the way Mentor handles certain special characters in repo names
 - Updated data view alignment on certain charts
 - Fixed git clone issue if repo name and display name were configured to be different
 - Fixed git clone failed issue if name was changed in the Installer

Merly Mentor v. 0.0.22 Changelog:

- Repo modification actions (edit/delete/rerun/infer/clone/fetch) need `lead` or higher access level
- User modifications in the Mentor Service require `admin` or `owner` access level
- Fixed issue with adding a repo in the Mentor Service resulting in 404 error (switched from **+** to **#** as the symbol)
- Added check to ensure username/email is unique on `new user` and `edit user`
- Added option to select Merly SMTP or custom SMTP server
- Added option to switch between public and private service with `service_url`. Public service will generate emails for add/edit user, and private service will just add/edit without sending an email (as there is no public URL that can be used to access the service, so an email link cannot be used to access the Mentor Service)
- New: User list table display; now with role, ID, and status (email sent, inactive, active, etc.)
- Quick actions now has now an edit-then-save mode, which allows all fields to be changed and then saved; fix for erroneous behavior when in edit mode the dialog would disappear when the mouse is moved outside the dialog. Now, the dialog will close once "Save" is clicked and the mouse leaves the dialog
- Fixed a bug which allowed a `developer` to remove an repo issue by selecting an empty action or empty assigned user
- Fixed a crash bug found with the `repo issues` page

The Merly development team is constantly adding new features and enhancements to the Merly Mentor ecosystem. To report any issues, please email support@merly.ai.



THE MACHINE PROGRAMMING COMPANY