Getting started with GitHub

In this train, weâ€™ll learn the basics of using GitHub including repository management, collaboration features, and integrating Git with GitHub.

Learning objectives

By the end of this train, you should be able to:

- Set up a basic GitHub repo and explain repository settings.
- Collaborate on GitHub by forking repositories and creating pull requests.
- Explore GitHub features like issues, wikis, and project boards for effective project management.

Outline

1. Understanding the structure and setup of a GitHub repository
2. Collaborating effectively on GitHub
3. Utilising GitHub for effective project management

1. Understanding the structure and setup of a GitHub repository

What is a GitHub repository?

A GitHub repository (or 'repo') is our project's storage space. It can hold various files and data essential to the project. Crucially, it functions as a version control system, maintaining a record of changes and facilitating collaboration.

How to create a new repository:

- Sign in to GitHub.
- Click the + icon at the top right and select 'New repository'.

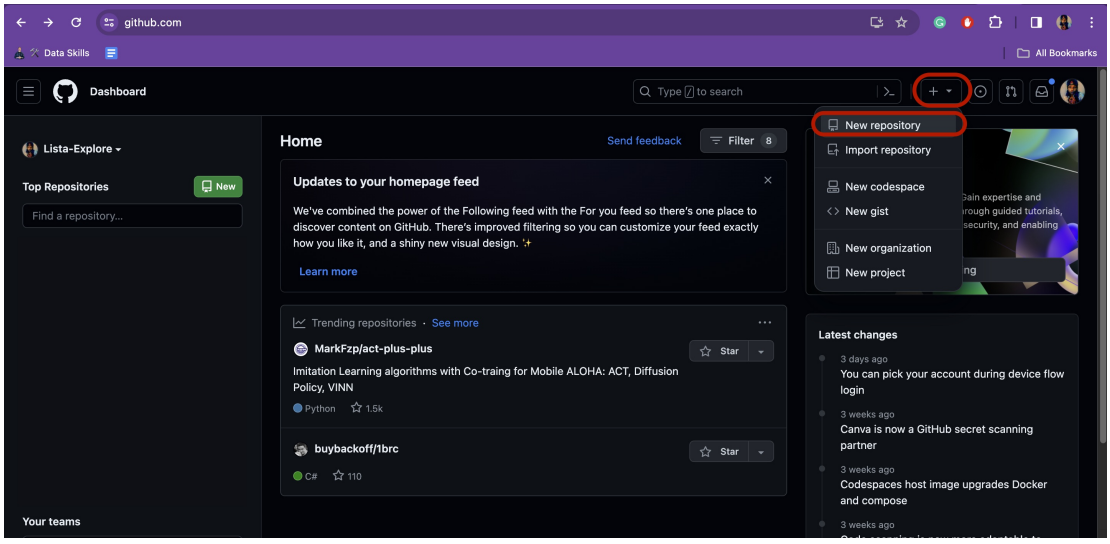

Figure 1: Create a new repository (a)

- Name the repository, add a description, and choose between public or private visibility.

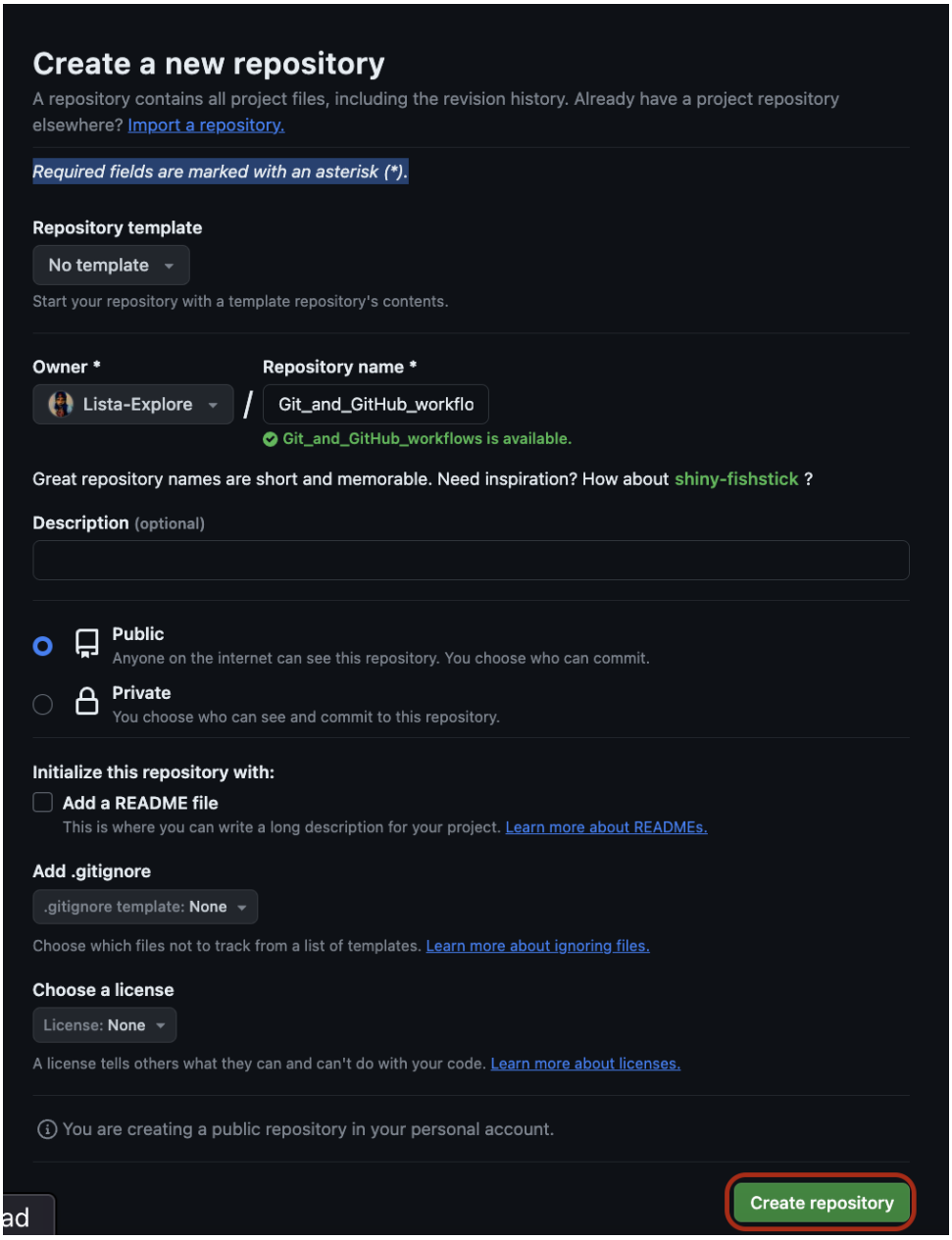

Figure 2: Create a new repository (b)

- Optionally, initialise with a README, .gitignore, and select a license.

Configuring repository settings:

- **Visibility settings:** Control who can view and interact with the repository. Public repositories are visible to everyone while private repositories are limited to certain collaborators.
- **Branch protection rules:** Establish rules to safeguard specific branches. For instance, require pull reviews before merging, ensuring code quality and collaborative assessment.
- **Collaborators and teams:** Manage who can contribute to the repository to ensure secure and controlled collaboration.

2. Collaborating effectively on GitHub

Forking repositories

Forking a repository means creating a personal copy on our GitHub account. This enables us to experiment with changes without impacting the original project.

How to fork: Click the Fork button on the repository's page. We can then clone our fork, make alterations, and propose these changes to the original repository.

Creating pull requests

After forking and adjusting the repository, we can suggest these changes to the original repository through a pull request.

How to initiate pull requests:

- Navigate to the original repository on GitHub.
- Click New pull request.
- Select our fork and the branch containing our changes.
- Review and create the pull request, detailing our contributions.

3. Utilising GitHub for effective project management

Issues

Issues in GitHub are a fundamental tool for tracking tasks, enhancements, or bugs in our projects. They allow for an organised and detailed way to manage different aspects of our project's development and maintenance.

Creating and managing issues: Begin by opening a new issue for each distinct task, challenge, or bug we encounter. We can categorise these issues with labels such as "bug", "feature request", or "enhancement" to make them easily searchable and sortable. Assign each issue to relevant team members for resolution. This practice helps in delegating tasks and tracking their progress.

Advanced features: Utilise GitHub's milestones and projects to link issues to specific project goals or phases. This helps in tracking the progress of a set of issues towards a particular objective.

Comments and discussions: Team members can use the comment section in each issue to discuss solutions, provide updates, or ask for further clarification. This feature ensures that all relevant discussions are kept in one place and are easily accessible.

Wikis

GitHub wikis offer a flexible and user-friendly space for storing extensive documentation or important project information. This feature is essential for maintaining comprehensive and accessible knowledge bases for our projects.

Creating and editing wikis: Wikis are accessible directly from our repository and offer a markdown-based system for easy editing. Team members can collaboratively contribute to the wikis, making it an ideal place for storing project documentation, detailed guides, meeting notes, or any other relevant information.

Organisation and navigation: Organise our wiki pages with a table of contents and categorise them into sections for easy navigation. This helps users find the information they need.

Revision history: Wikis maintain a revision history, allowing us to track changes and revert to previous versions if necessary. This feature is crucial for maintaining the integrity of our project's documentation.

Project boards

Project boards in GitHub provide a visual and interactive way to track and manage the progress of tasks in our project. These boards are highly customisable and can be adapted to fit various project management methodologies like Kanban or Scrum.

Setting up project boards: Start by creating a new board and customising its columns to reflect our workflow stages, such as 'To Do', 'In Progress', 'In Review', and 'Done'. We can then add tasks or issues to these columns, providing a transparent and real-time overview of our project's progress.

Automation features: Utilise GitHub's automation tools to automatically move tasks through different stages of our workflow based on triggers like issue assignments or pull request merges. This reduces manual updates and keeps our project board current.

Integration with issues and pull requests: Link our project board with specific issues and pull requests. This integration allows for a seamless transition of tasks from the planning stage to implementation and final review.

By effectively using these tools, GitHub can serve as a comprehensive platform for managing our software projects, enhancing collaboration, and improving overall productivity.