Faysal Aziz

Mastering GitHub

Collaborative Development for Modern Teams 1 - 2 days

FA-0534 · 2 days

Software Development

Introduction

Learn how developers actually use GitHub to collaborate, track work, and ship software.

GitHub is more than a place to store code, it's where modern software development happens. Teams around the world rely on GitHub to collaborate, manage changes, build features, and ship products in a consistent, traceable way.

Understanding GitHub isn't just about knowing commands; it's about mastering workflows that support teamwork, quality control, and rapid iteration.

This course bridges the gap between theory and real-world practice, teaching you the tools and approaches that professional developers use every day.

Learning outcomes

By the end of this course, learners will be able to:

- Navigate GitHub confidently and understand its role in collaborative workflows.
- Use Git (locally and with GitHub) to manage code changes safely.
- Structure and manage repositories, branches, and pull requests.
- Apply best practices for issue tracking, code review, and release management.
- Leverage GitHub features like GitHub Actions, Projects, and Wikis to streamline team work.

Prerequisites

Learners should have:

- Basic knowledge of programming concepts (variables, functions, files).
- A computer with a terminal/command prompt and internet access.
- Familiarity with text editing (VS Code, Sublime, Atom, etc.) is helpful but not required.

Training outline

1. 1. Introduction to Git and GitHub

What is Git?

- Version control fundamentals
- Local vs remote repositories

What is GitHub?

- GitHub's role in the software ecosystem
- GitHub free vs paid plans
- GitHub interface overview

2. 2. Git Foundations

- Installing Git and setting up user identity
- Git configuration and preferences
- Creating a new local repository

  Basic Git commands

  - git init
  - git status
  - git add
  - git commit
- Understanding the Git commit lifecycle

3. 3. Working with Remotes on GitHub

- Creating a repository on GitHub
- Cloning a GitHub repository

  Pushing and pulling changes

    - git push
    - git pull
    - git fetch
- Remote repository management

4. 4. Branching and Merging

- Why branches matter

  Creating and switching branches
    - git branch
    - git checkout / git switch
- Merge strategies
- Resolving merge conflicts
- Best practices for branching

5. 5. GitHub Collaboration Fundamentals

- Forks vs clones vs branches
- Working in teams

  Pull request workflows
    - Creating pull requests
    - Reviewing code
    - Comments and feedback loops
- Merging pull requests

6. 6. Issue Tracking and Project Management

- Creating and managing issues
- Labels, milestones, and assignment

  Using GitHub Projects
    - Boards and workflows
    - Automations
- Linking issues to pull requests

7. 7. Advanced GitHub Features

GitHub Actions

    - CI/CD fundamentals
    - Creating workflows
    - Running tests and deployments

  GitHub Pages
    - Hosting static sites
    - Configuration and publishing
- Wikis and documentation

8. 8. Code Quality and Continuous Integration

- Integrating automated tests
- Status checks
- Protected branches
- Code owners

9. 9. Security and Permission Models

- Repository access levels
- Teams and organizations
- Secrets management
- Dependency scanning

10. 10. Release Management and Tags

- Creating release branches
- Using tags and versioning
- Drafting GitHub releases

11. 11. Real-World GitHub Workflows

- GitHub Flow
- Git Flow
- Trunk-based development
- Choosing the right workflow for your team

12. 12. Tips, Tricks, and Ecosystem Tools

- Using GitHub CLI

- GitHub Desktop
- Integrations (Slack, Jira, VS Code extensions)
- Common pitfalls and how to avoid them

© 2026 Faysal Aziz · Kuala Lumpur