GitHub Foundations - Exam Study Guide

Hugo Estrada
February 20, 2024

Introduction to Git and GitHub

Git and GitHub Basics:

- Version Control: Understand the concept of version control and its importance in tracking changes, collaboration, and maintaining the history of a project.
- Distributed Version Control: Define distributed version control and how Git enables multiple developers to work on a project simultaneously.
- Git: Learn the basics of Git, including its functionality as a version control system and its command-line usage.
- GitHub: Differentiate between Git and GitHub, recognizing GitHub as a platform that extends Git's capabilities with additional features like hosting, collaboration tools, and more.
- GitHub Repository: Describe what a GitHub repository is, including its structure and components like README, LICENSE, and CONTRIBUTING files.
- Commits: Understand the concept of commits in Git, how they represent changes, and their role in the version history.
- Branching: Define branching in Git and its use for working on different parts of a project without affecting the main codebase.
- Remotes: Explain what a remote is in Git terminology and how it relates to repository management.

GitHub Entities

- Accounts: Describe the different types of GitHub accounts (personal, organization, enterprise) and their respective products.
- GitHub Markdown: Identify basic Markdown formatting syntax and its use in GitHub for documentation and comments.

GitHub Desktop and Mobile

- GitHub Desktop: Understand the difference between GitHub Desktop and github.com, and the features available with GitHub Desktop.
- GitHub Mobile: Describe the features available with GitHub Mobile and how to manage notifications through the app.

Working with GitHub Repositories

Repository Management

- Creating Repositories: Learn how to create a new repository, including setting up repository templates and understanding repository visibility.
- Cloning and Branching: Describe how to clone a repository and create new branches within it.
- README and Other Files: Understand the components of a good README and the recommended repository files.

Collaboration Features

Issues and Pull Requests

- Issues: Use issues to track tasks, enhancements, and bugs, and understand GitHub's unique issue tracking features[12].

- Pull Requests: Learn how to create and manage pull requests to propose and collaborate on changes to a repository.

Modern Development

Development Workflows

- GitHub Flow: Describe the GitHub flow and how it facilitates collaboration and code review.
- Continuous Integration/Continuous Deployment (CI/CD): Understand the basics of CI/CD and how GitHub Actions can be used to automate workflows.

Project Management

GitHub for Project Management

- Projects: Use GitHub Projects to plan and track work, understanding how to create and customize project boards.
- Milestones and Labels: Utilize milestones and labels to organize and prioritize issues and pull requests.

Privacy, Security, and Administration

Security and Permissions

Repository Roles: Learn about permission levels for repositories and how to manage access.

- Security Features: Understand GitHub's security features, including code scanning and secure deployments with OpenID Connect.

Benefits of the GitHub Community

Community Engagement

- GitHub Community: Explore the benefits of engaging with the GitHub community, including collaboration, learning, and sharing best practices.

Recommendations and Best Practices for Success

- Hands-on Practice: Engage in hands-on exercises and use GitHub for personal or collaborative projects to reinforce learning.
- Utilize Learning Resources: Take advantage of curated learning paths on platforms like Microsoft Learn and LinkedIn Learning.
- Participate in Discussions: Join discussions on platforms like Reddit to ask questions, share knowledge, and connect with other learners.