10 key Features of Github

# Table of Contents

1. Repositories

2. Branches

3. Pull Requests

4. Issues

5. GitHub Actions

6. Wikis

7. GitHub Pages

8. Projects (Kanban Boards)

9. Code Review

10. Security and Vulnerability Alerts

1. Repositories

#Overview

A repository, or "repo," is a central place where all the files for a project are stored. Each repository contains all the project’s files and revision history. Repositories can be public or private.

#Creating a Repository

1. Go to your GitHub homepage.

2. Click the "New" button next to "Repositories".

3. Fill in the repository name, description, and choose its visibility (public/private).

4. Click "Create repository".

# Usage

Store code, manage files, and track changes over time.

# Code Snippet

bash # Clone a repository

git clone https://github.com/username/repo.git (https://github.com/username/repo.git)

2. Branches

#Overview

Branches are used to develop features, fix bugs, or safely experiment with new ideas in a contained area of your repository.

# Creating a Branch

1. Navigate to your repository.

2. Click the branch dropdown list, then enter a branch name.

3. Click "Create branch".

# Usage

Isolate development work without affecting other branches, typically the main branch.

#Code Snippet

bash # Create a new branch

git checkout -b new-feature

3. Pull Requests

# Overview

Pull requests let you tell others about changes you've pushed to a branch in a repository on GitHub. Once a pull request is opened, you can discuss and review the potential changes with collaborators.

# Creating a Pull Request

1. Navigate to the repository.

2. Click the "Pull requests" tab, then "New pull request".

3. Select the branch with the changes and the branch you want to merge into.

4. Click "Create pull request".

# Usage

Facilitate discussions and code reviews before merging changes.

# Code Snippet

bash # Merge a pull request

git merge feature-branch

4. Issues

# Overview

Issues are used to track tasks, enhancements, and bugs for your projects.

# Creating an Issue

1. Go to your repository.

2. Click the "Issues" tab, then "New issue".

3. Fill in the title and description.

4. Click "Submit new issue".

# Usage

Manage and organize project tasks and bugs.

# Code Snippet

markdown <!-- Link to an issue -->

Fixes #123

5. GitHub Actions

# Overview

GitHub Actions help you automate workflows right in your repository. You can build, test, and deploy your code based on specified events.

# Creating an Action

1. Go to your repository.

2. Click the "Actions" tab, then "New workflow".

3. Choose a template or create your own from scratch.

4. Commit the workflow file.

# Usage

Automate CI/CD processes and other tasks.

# Code Snippet

yaml name: CI

on: [push]

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Set up Node.js

uses: actions/setup-node@v1

with:

node-version: '14'

- run: npm install

- run: npm test

6. Wikis

# Overview

Wikis provide a space to document your project, making it easier for others to understand and contribute.

# Creating a Wiki Page

1. Go to your repository.

2. Click the "Wiki" tab.

3. Click "New page" and start writing your documentation.

# Usage

Provide comprehensive project documentation and guides.

# Code Snippet

markdown <!-- Example of a wiki link -->

Home (https://github.com/username/repo/wiki)

7. GitHub Pages

# Overview

GitHub Pages is a static site hosting service that takes HTML, CSS, and JavaScript files straight from a repository on GitHub.

# Setting Up GitHub Pages

1. Go to your repository.

2. Click the "Settings" tab.

3. Scroll down to the "GitHub Pages" section.

4. Select the source branch and folder.

5. Click "Save".

# Usage

Host project documentation, portfolios, and more.

# Code Snippet

html <!-- Example index.html for GitHub Pages -->

<!DOCTYPE html>

<html>

<head>

<title>GitHub Pages</title>

</head>

<body>

<h1>Hello, GitHub Pages!</h1>

</body>

</html>

8. Projects (Kanban Boards)

# Overview

GitHub Projects provide Kanban-style boards for managing work and tracking progress.

# Creating a Project Board

1. Go to your repository.

2. Click the "Projects" tab, then "New project".

3. Choose a template or start from scratch.

4. Add columns and cards to manage tasks.

# Usage

Organize and prioritize work visually.

# Code Snippet

markdown <!-- Link an issue to a project board -->

- [ ] Add feature X #123

9. Code Review

# Overview

Code review is a critical part of the development process, where peers review each other's code to catch bugs and improve quality.

# Conducting a Code Review

1. Navigate to the pull request.

2. Click "Files changed" to review the changes.

3. Add comments, suggestions, or request changes.

4. Click "Submit review".

# Usage

Improve code quality and share knowledge.

# Code Snippet

markdown <!-- Example code review comment -->

// Consider using a more descriptive variable name here

let descriptiveName = value;

10. Security and Vulnerability Alerts

# Overview

GitHub provides security features to help you find and fix vulnerabilities in your dependencies.

# Enabling Security Alerts

1. Go to your repository.

2. Click the "Settings" tab.

3. Under "Security" click "Enable security alerts".

# Usage

Stay informed about and address security vulnerabilities in your project.

# Code Snippet

markdown <!-- Example of a security alert badge -->

GitHub security alerts (https://img.shields.io/github/vulnerabilities/username/repo)

By leveraging these features, you can effectively manage your projects on GitHub, ensuring better collaboration, quality, and security. Happy coding!