

GitHub, một hệ sinh thái mạnh mẽ dành cho các lập trình viên, không chỉ là nơi lưu trữ mã nguồn mà còn là một cộng đồng sáng tạo mở rộng. Nhờ vào khả năng làm việc nhóm mạnh mẽ, GitHub trở thành nền tảng hỗ trợ cho mọi ý tưởng và dự án lập trình. Với bài viết "[GitHub là gì?](#) Hướng dẫn cách sử dụng GitHub", hãy cùng [VinaHost](#) khám phá về nền tảng này nhé!

1. GitHub là gì?

[GitHub](#) là một dịch vụ lưu trữ mã nguồn ([source code](#)) trực tuyến dựa trên nền tảng [Git](#). Nó cung cấp một nền tảng cho các nhà phát triển phát triển, quản lý và theo dõi mã nguồn của dự án phần mềm. Dịch vụ này cho phép các nhóm phát triển làm việc cùng nhau trên các dự án phần mềm từ xa và theo dõi lịch sử của mã nguồn.

2. Lịch sử hình thành và phát triển của GitHub

Lịch sử của GitHub thể hiện sự phát triển nhanh chóng và sự tôn trọng của cộng đồng phát triển [mã nguồn mở](#). Đóng góp của GitHub đã giúp định hình cách chúng ta làm việc và hợp tác trong lĩnh vực công nghệ thông tin và phát triển phần mềm.

Cùng VinaHost điểm qua một số sự kiện quan trọng trong lịch sử của Git-Hub sau đây nhé!

- **Năm 2008:** Git-Hub được thành lập bởi Tom Preston-Werner, Chris Wanstrath và PJ Hyett. Ý tưởng ban đầu của họ là tạo ra một nền tảng để sử dụng và mạnh mẽ cho việc quản lý mã nguồn. GitHub ra mắt chính thức tại TechCrunch Disrupt, một sự kiện công nghệ quan trọng. Sự xuất hiện này đã đánh dấu sự khởi đầu của GitHub và thu hút sự chú ý của cộng đồng phát triển.
- **Năm 2009:** Git-Hub bắt đầu thu hút sự quan tâm của các nhà phát triển mã nguồn mở. Các dự án lớn như **Ruby on Rails**, **jQuery** và **Node.js** bắt đầu sử dụng Git-Hub để quản lý mã nguồn của họ.
- **Năm 2010:** GitHub thực hiện vòng gọi gây quỹ đầu tiên và nhận được 7,5 triệu đô la đầu tư từ Sequoia Capital. Điều này giúp họ mở rộng và phát triển nhanh chóng.
- **Năm 2011:** GitHub có hơn một triệu kho lưu trữ trên nền tảng của họ và trở thành một trung tâm quan trọng cho các dự án mã nguồn mở.
- **Năm 2012:** Git-Hub tăng cường tính năng dành cho doanh nghiệp bằng việc ra mắt dịch vụ GitHub Enterprise, cho phép các tổ chức tạo ra máy chủ riêng để quản lý mã nguồn.
- **Năm 2018:** Microsoft mua lại GitHub với giá 7,5 tỷ đô la Mỹ. Sự mua lại này không chỉ tạo ra tiền đề cho sự phát triển mạnh mẽ của GitHub mà còn đồng nghĩa với sự tích hợp của nền tảng này vào hệ thống Microsoft.
- **Năm 2019:** Git-Hub có hơn 40 triệu người dùng và trở thành một phần quan trọng của cộng đồng phát triển mã nguồn mở và ngành công nghệ thông tin.
- **Năm 2020:** GitHub giới thiệu GitHub Codespaces, cho phép các nhà phát triển lập trình trực tiếp từ trình duyệt web mà không cần cài đặt môi trường phát triển trên máy tính cá nhân.



GitHub

vinahost.✓

Bạn có thể tiến hành Github download tại github.com

3. Các loại tài khoản của GitHub

GitHub cung cấp nhiều loại tài khoản để phù hợp với nhu cầu của các cá nhân và tổ chức khác nhau.

Những loại tài khoản này cho phép người dùng ứng dụng GitHub theo cách phù hợp với mục tiêu của họ, từ việc đóng góp vào dự án mã nguồn mở cho đến quản lý dự án phát triển phức tạp trong các tổ chức lớn.

Dưới đây là một số loại tài khoản phổ biến trên Git-Hub:

- **Tài khoản Cá nhân (Individual Account):** Đây là loại tài khoản miễn phí cho các cá nhân. Bạn có thể tạo kho lưu trữ công khai và tham gia vào các dự án mã nguồn mở. Tài khoản cá nhân không mất phí và được sử dụng rộng rãi bởi các nhà phát triển.
- **Tài khoản Tổ chức (Organization Account):** Đây là loại tài khoản dành cho tổ chức, doanh nghiệp hoặc dự án lớn. Tài khoản tổ chức cho phép bạn quản lý các dự án và thành viên trong một không gian tổ chức chung. Git-Hub cung cấp cả phiên bản miễn phí và các phiên bản trả phí với nhiều tính năng mở rộng.
- **Tài khoản Developer:** Đây là phiên bản trả phí của tài khoản cá nhân. Tài khoản này cung cấp các tính năng bổ sung như tích hợp với các công cụ phát triển bên ngoài và hỗ trợ ưu tiên.
- **Tài khoản Team:** Đây là phiên bản trả phí của tài khoản tổ chức. Tài khoản này cho phép bạn tạo nhiều nhóm công việc và quản lý các dự án của bạn dưới dạng nhóm.
- **Tài khoản Enterprise:** Đây là loại tài khoản cao cấp dành cho các tổ chức lớn. Nó cung cấp tính năng cao cấp bao gồm bảo mật nâng cao và tích hợp với các dịch vụ doanh nghiệp khác. Tài khoản Enterprise yêu cầu việc trả phí dựa trên số lượng người dùng.
- **GitHub Free và GitHub Pro:** Đây là hai tùy chọn cho tài khoản cá nhân. GitHub Free là phiên bản miễn phí, trong khi GitHub Pro là phiên bản trả phí với các tính năng bổ sung như tích hợp với các công cụ phát triển bên ngoài và hỗ trợ ưu tiên.

- **GitHub Education:** GitHub cung cấp chương trình giáo dục dành cho học sinh, sinh viên và giáo viên. Tài khoản GitHub Education cung cấp một loạt tính năng mạnh mẽ miễn phí cho mục đích giảng dạy và học tập.

4. Các tính năng nổi bật của GitHub

GitHub, một trong những nền tảng quản lý dự án và phiên bản mã nguồn mở hàng đầu, có nhiều tính năng nổi bật. Những tính năng này làm cho Git-Hub trở thành một nền tảng mạnh mẽ cho phát triển phần mềm và quản lý dự án.

- **Hệ Thống Quản Lý Phiên Bản (VCS):** Git-Hub sử dụng Git làm hệ thống quản lý phiên bản, cho phép theo dõi sự thay đổi trong mã nguồn và quản lý các version khác nhau của project.
- **Repository:** GitHub cung cấp mô hình repository để lưu trữ mã nguồn. Mỗi dự án có thể có một repository riêng, và người dùng có thể sao chép (fork) repository của người khác để đóng góp vào dự án đó.
- **Pull Requests:** Cho phép những người đóng góp đề xuất thay đổi vào repository gốc. Pull requests tạo ra một không gian cho thảo luận, xem xét mã nguồn và các sự điều chỉnh trước khi thay đổi được hợp nhất (merged) vào repository chính.
- **Branching:** Git-Hub hỗ trợ việc tạo ra các nhánh (branches) khác nhau của dự án, cho phép phát triển đồng thời nhiều tính năng mà không ảnh hưởng đến nhau.
- **Commit History:** Cho phép xem chi tiết lịch sử các commit, bao gồm thông tin về người thực hiện, thời gian và nội dung thay đổi.
- **Issues:** Được sử dụng để theo dõi công việc, nhiệm vụ, lỗi, và các cải tiến. Mỗi vấn đề có thể được đặt nhãn, gán cho người thực hiện và được theo dõi qua thời gian.
- **Projects:** GitHub Projects giúp quản lý công việc, theo dõi tiến độ và tổ chức công việc theo cách linh hoạt.
- **Wiki và Pages:** Cho phép tạo ra trang Wiki để chia sẻ thông tin và tài liệu. Git-Hub Pages cung cấp khả năng tạo [website tĩnh](#) từ repository.
- **Collaboration:** GitHub cung cấp các tính năng như nhận xét, thảo luận và cấp quyền truy cập để hỗ trợ sự hợp tác giữa các thành viên trong dự án.
- **Integrations:** Hỗ trợ nhiều tích hợp với các công cụ phát triển phổ biến như **CI/CD (Continuous Integration/Continuous Deployment)**, linter, và các dịch vụ thử nghiệm tự động.
- **Security Features:** Git-Hub cung cấp các công cụ để kiểm tra mã nguồn mở, quét lỗ hổng bảo mật, và theo dõi các phần mềm có thể bị ảnh hưởng.
- **Community:** GitHub tạo ra một cộng đồng lớn với hàng triệu người phát triển trên khắp thế giới, cung cấp cơ hội cho việc học hỏi, chia sẻ và đóng góp.

5. Những khái niệm khi sử dụng Git mà bạn nên biết



Bạn cần nắm một số khái niệm cơ bản để hiểu rõ cách Git hoạt động

Khi sử dụng Git, có một số khái niệm cơ bản nhưng quan trọng giúp bạn hiểu rõ cách Git hoạt động và tận dụng được mọi tính năng của nó:

- **Repository (Repo):** Một thư mục hoặc không gian lưu trữ chứa mã nguồn và lịch sử thay đổi của dự án.
- **Commit:** Một “snapshot” của trạng thái của mã nguồn tại một thời điểm cụ thể. Mỗi commit có một mã hash độ duy nhất.
- **Branch:** Một nhánh là một phiên bản song song của dự án, giúp phát triển độc lập các tính năng mà không ảnh hưởng đến nhau.
- **Merge:** Hợp nhất (merge) là quá trình kết hợp các thay đổi từ một nhánh (branch) vào nhánh khác.
- **Pull Request (PR):** Là một yêu cầu để hợp nhất (merge) các thay đổi từ một nhánh vào nhánh khác. Thường được sử dụng trong các dự án đa người đóng góp.
- **Clone:** Sao chép một repository từ xa (remote) về máy tính cá nhân để có thể thực hiện công việc cục bộ.
- **Fetch:** Là quá trình tải về tất cả các thay đổi từ xa mà không hợp nhất chúng vào nhánh hiện tại.
- **Pull:** Tương tự như fetch, nhưng tự động hợp nhất thay đổi từ xa vào nhánh hiện tại.
- **Push:** Đẩy (push) các commit cục bộ lên repository từ xa.
- **Conflict:** Xung đột xảy ra khi hai nhánh hoặc nhiều người cùng sửa đổi một phần của mã nguồn.
- **Staging Area/Index:** Là nơi tạm thời lưu trữ các thay đổi trước khi commit chúng.
- **Working Directory:** Thư mục trên máy tính cá nhân nơi bạn thực hiện công việc và sửa đổi mã nguồn.
- **HEAD:** Là con trỏ hiện tại đang trỏ đến commit cuối cùng trong nhánh bạn đang làm việc.

- **Tag:** Một “đánh dấu” một commit cụ thể để dễ dàng tìm kiếm và tham chiếu trong tương lai.
- **Remote:** Là repository khác nằm ở một vị trí xa, thường là trên mạng.

6. Lợi ích khi sử dụng GitHub

6.1. Quản lý Source Code dễ dàng

GitHub cung cấp môi trường hiệu quả cho việc quản lý mã nguồn, từ việc theo dõi sự thay đổi đến việc quản lý các nhánh (branches) của dự án.

6.2. Tracking các thay đổi thông qua Version

Git-Hub giúp theo dõi và quản lý các phiên bản của dự án, giúp dễ dàng quay trở lại các phiên bản trước đó.

6.3. Markdown

Git-Hub sử dụng Markdown cho việc viết tài liệu, giúp tạo ra các tài liệu đẹp và dễ đọc.

6.4. GitHub giúp khẳng định giá trị của mình

Git-Hub là một nơi cho các lập trình viên và nhóm hiển thị và chia sẻ công việc của họ, giúp họ xây dựng danh tiếng và khẳng định kỹ năng.

6.5. Giúp cải thiện khả năng code, tracking bug

Sử dụng Git-Hub để theo dõi lỗi (issues) và thực hiện các thay đổi thông qua pull requests giúp cải thiện chất lượng mã nguồn.

6.6. Kho tài nguyên trên Github xịn

GitHub là nguồn cung cấp lớn cho các thư viện mã nguồn mở, framework, và nhiều tài nguyên khác.

6.7. GitHub Action

Dịch vụ tự động hóa các công việc liên quan đến quy trình phát triển, kiểm thử, và triển khai.

6.8. Github Package Registry

Cho phép lưu trữ và quản lý các gói phần mềm và thư viện.

6.9. Mở rộng mối quan hệ

GitHub không chỉ là nơi lưu trữ mã nguồn, mà còn là một cộng đồng lớn với khả năng mở rộng mối quan hệ chuyên nghiệp.

Việc sử dụng Gi-tHub không chỉ làm cho quá trình phát triển phần mềm trở nên hiệu quả mà còn giúp xây dựng cộng đồng và mạng lưới chuyên gia rộng lớn.

7. GitHub được sử dụng để làm gì?



GitHub không chỉ là nơi lưu trữ mã nguồn mà còn là một hệ sinh thái phong phú với nhiều tính năng hỗ trợ quản lý dự án và tương tác sáng tạo

7.1. Kéo, thả và lưu trữ đoạn code

Git-Hub là một nền tảng lưu trữ mã nguồn hàng đầu, cho phép người sử dụng kéo và thả mã nguồn của họ vào kho lưu trữ cá nhân hoặc dự án. Bạn có thể tạo và quản lý các kho (repositories) để chứa mã nguồn của ứng dụng hoặc dự án.

7.2. Tạo thư mục thông qua giao diện web

Người dùng có thể tạo và quản lý các thư mục trực tiếp từ giao diện web github desktop của GitHub, giúp tổ chức mã nguồn một cách hiệu quả.

7.3. Git URL Shortener

Git-Hub cung cấp dịch vụ rút gọn URL sử dụng Git URL Shortener để tạo liên kết ngắn gọn đến các trang, dự án, hoặc file trên Git-Hub.

7.4. Tìm kiếm tập tin/ File

Tính năng tìm kiếm mạnh mẽ của GitHub cho phép người sử dụng nhanh chóng tìm thấy các file và đoạn mã trong kho lưu trữ.

7.5. Github Emoji

Git-Hub hỗ trợ Emoji, giúp tạo thêm biểu cảm và ý nghĩa trong các bình luận và mã nguồn.

7.6. Github CLI

Git-Hub Command Line Interface (CLI) cho phép người sử dụng thực hiện nhiều thao tác GitHub qua dòng lệnh.

7.7. Linking Lines

Git-Hub hỗ trợ việc tạo liên kết giữa các dòng mã nguồn, giúp dễ dàng chia sẻ và thảo luận về từng phần của mã.

7.8. Task Checklist

Người sử dụng có thể tạo danh sách công việc và kiểm tra bằng cách sử dụng Task Lists trong các file Markdown, giúp quản lý công việc và thảo luận.

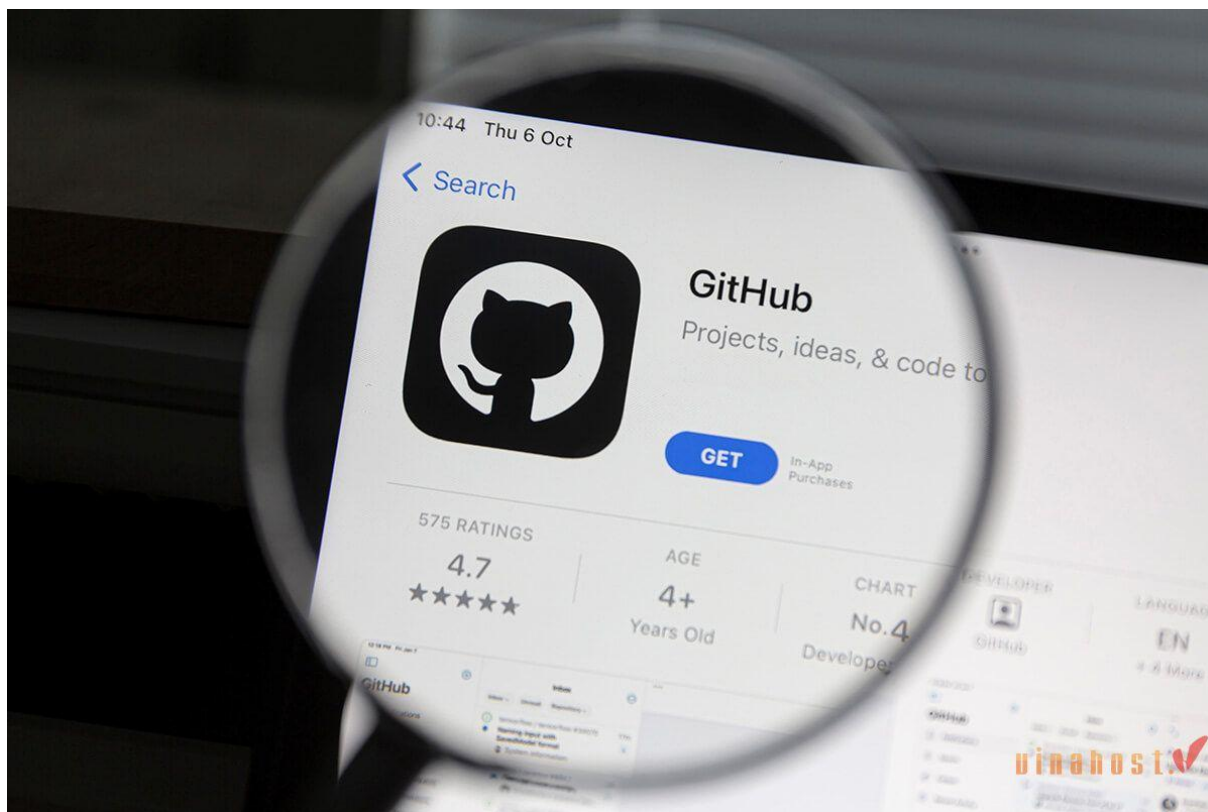
7.9. Định dạng Maps, CSV và 3D Rendering

GitHub hỗ trợ xem trực tiếp các định dạng như Maps, CSV, và có khả năng hiển thị trực tiếp nội dung 3D.

7.10. Get Octodex

Octodex là một bộ sưu tập các hình minh họa về Octocat, biểu tượng của Git-Hub, và được sử dụng để tạo sự thú vị và cá nhân hóa.

8. Hướng dẫn cách sử dụng GitHub chi tiết A-Z



Tải về Github

Hướng dẫn tạo một GitHub Repository

- Trước tiên cài đặt **GitHub install** và tiến hành github login vào **account GitHub** của bạn.
- Trên trang chủ GitHub, bấm vào nút **“New”** để tạo repository mới.
- Điền thông tin về tên, mô tả và lựa chọn các tùy chọn khác.
- Nhấn nút **“Create repository”** để hoàn thành.

Hướng dẫn tạo nhánh trên GitHub

- Trong repository, chọn tab **“Branch”**.
- Nhập tên nhánh mới và nhấn **“Create branch”**.
- Để làm việc trên nhánh mới, chọn nút **“Switch branch/tag”** và chọn nhánh bạn vừa tạo.

Làm việc với Commit Command

- Sử dụng lệnh git **add <tên-file>** để đưa file vào **staging area**.
- Sử dụng lệnh git **commit -m “Thông điệp commit”** để thực hiện **commit**.

Làm việc với Pull Command

- Để cập nhật dự án với thay đổi từ **repository** trên Git-Hub, sử dụng lệnh git pull.

Làm việc với Merge Command

- Để hợp nhất các thay đổi từ một nhánh khác vào nhánh hiện tại, sử dụng lệnh git **merge <tên-nhánh>**.

Cloning dự án

- Trên trang **repository**, nhấn vào nút “**Code**” và sao chép URL của **repository**.
- Trên máy tính của bạn, mở **terminal** và sử dụng lệnh git **clone <URL>**.

9. Tổng kết

Qua bài viết này, VinaHost mong rằng bạn đã nắm vững khái niệm “**GitHub là gì**” và biết cách sử dụng GitHub cơ bản. GitHub không chỉ là một nền tảng lưu trữ mã nguồn mà còn là một công cụ mạnh mẽ giúp các lập trình viên làm việc hiệu quả.

Với khả năng quản lý source code dễ dàng, theo dõi thay đổi qua các phiên bản, hỗ trợ định dạng Markdown, GitHub là một ứng dụng cho mọi nhà phát triển. Tính năng **Pull Request**, **Branching**, và các công cụ như **GitHub Action** giúp tối ưu hóa quy trình làm việc. Bạn có thể tự do tạo, lưu trữ, và chia sẻ mã nguồn một cách thuận tiện. Với một lượng tài nguyên phong phú và sự phổ biến rộng lớn, Git-Hub đã, đang và sẽ trở thành cộng đồng nơi mà các ý tưởng và dự án có thể phát triển mạnh mẽ!