Github

At a high level, GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code. To understand exactly what GitHub is, you need to know two connected principles:

- Version control
- Git

What Is Version Control?

Version control helps developers track and manage changes to a software project's code. As a software project grows, version control becomes essential. Take WordPress…

At this point, WordPress is a pretty big project. If a core developer wanted to work on one specific part of the WordPress codebase, it wouldn't be safe or efficient to have them directly edit the "official" source code.

Instead, version control lets developers safely work through **branching** and **merging**.

With **branching**, a developer duplicates part of the source code (called the **repository**). The developer can then safely make changes to that part of the code without affecting the rest of the project.

Then, once the developer gets his or her part of the code working properly, he or she can **merge** that code back into the main source code to make it official.

All of these changes are then tracked and can be reverted if need be.

Methodology

Step 0: Install git and create a GitHub account

The first two things you'll want to do are install git and create a free GitHub account

Step 1: Create a local git repository

When creating a new project on your local machine using git, you'll first create a new **repository** (or often, '**repo**', for short).

Step 2: Add a new file to the repo

Go ahead and add a new file to the project, using any text editor you like or running a touch command.

Step 3: Add a file to the staging environment

Add a file to the staging environment using the **git add** command.

Step 4: Create a commit

It's time to create your first commit!

Step 5: Create a new branch
Step 6: Create a new repository on GitHub

If you only want to keep track of your code locally, you don't need to use GitHub. But if you want to work with a team, you can use GitHub to collaboratively modify the project's code.

Step 7: Push a branch to GitHub

Now we'll **push** the commit in your branch to your new GitHub repo. This allows other people to see the changes you've made. If they're approved by the repository's owner, the changes can then be merged into the primary branch.

Step 8: Create a Pull Request (PR)

A pull request (or PR) is a way to alert a repo's owners that you want to make some changes to their code. It allows them to review the code and make sure it looks good before putting your changes on the primary branch.

Step 9: Merge a PR

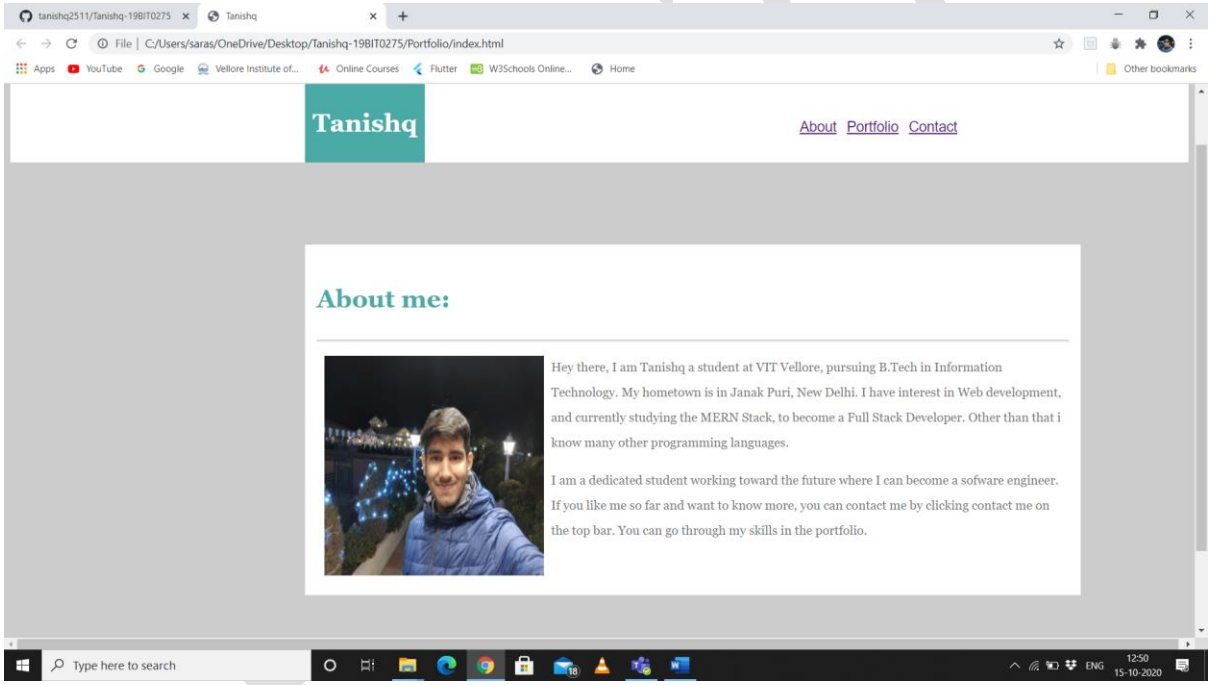
Go ahead and click the green 'Merge pull request' button. This will merge your changes into the primary branch.

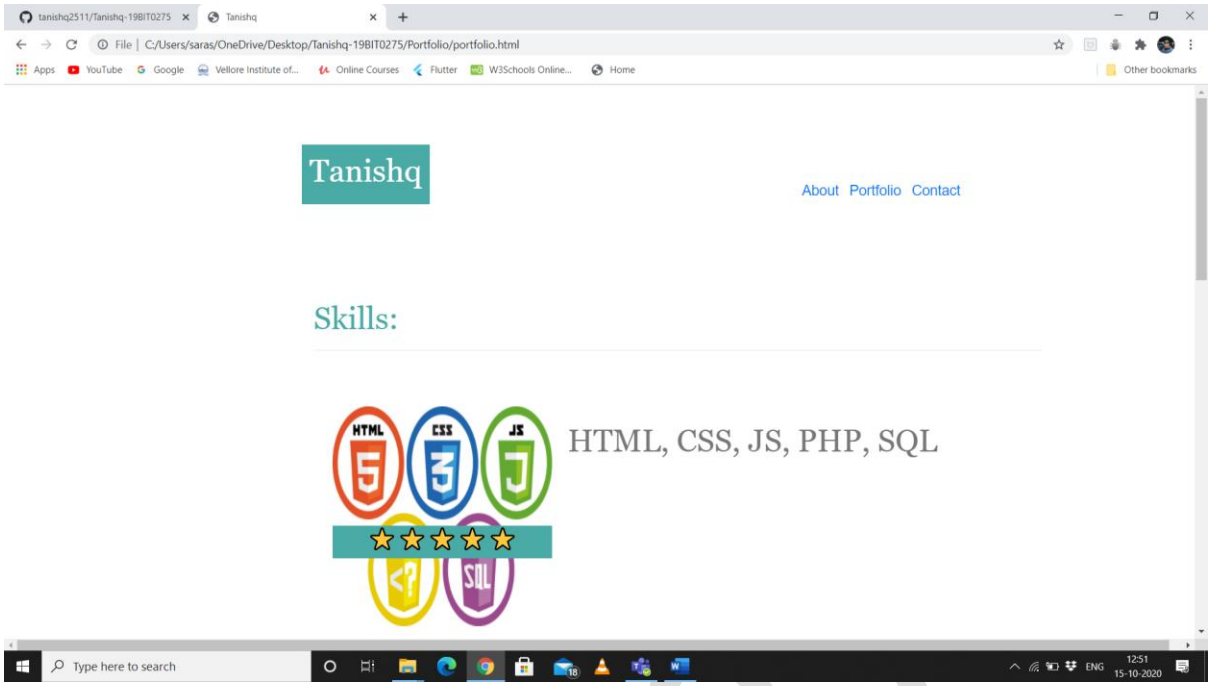
Step 10: Get changes on GitHub back to your computer

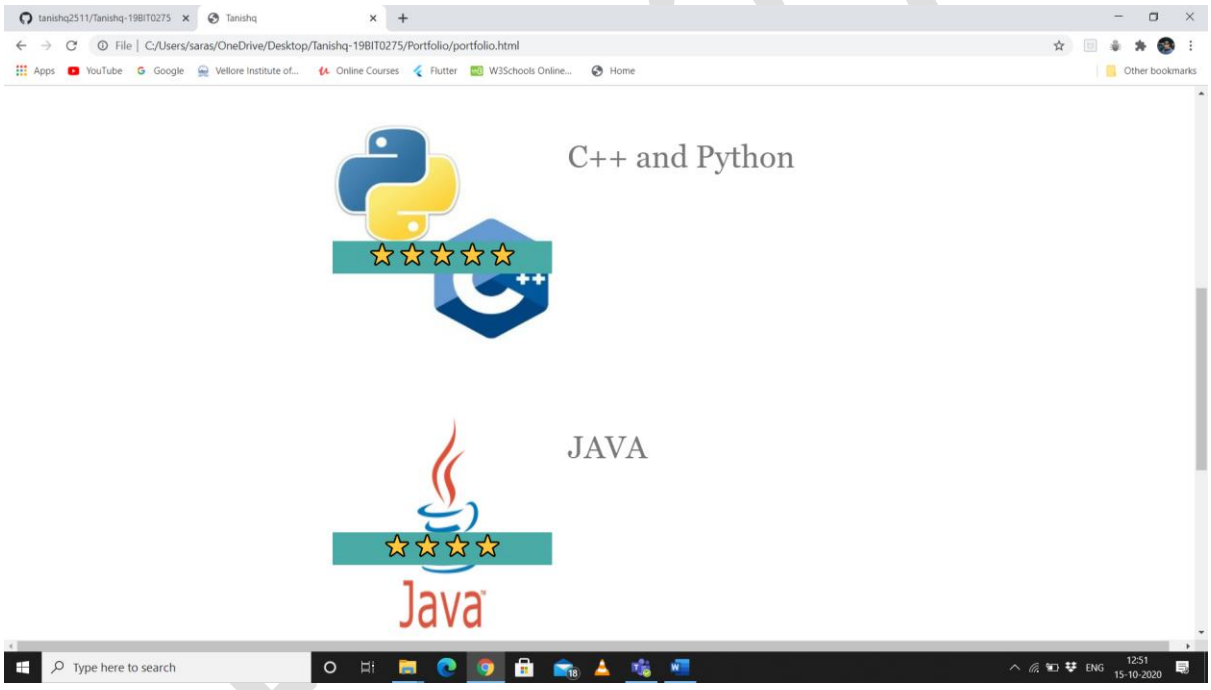
Right now, the repo on GitHub looks a little different than what you have on your local machine. For example, the commit you made in your branch and merged into the primary branch doesn't exist in the primary branch on your local machine.

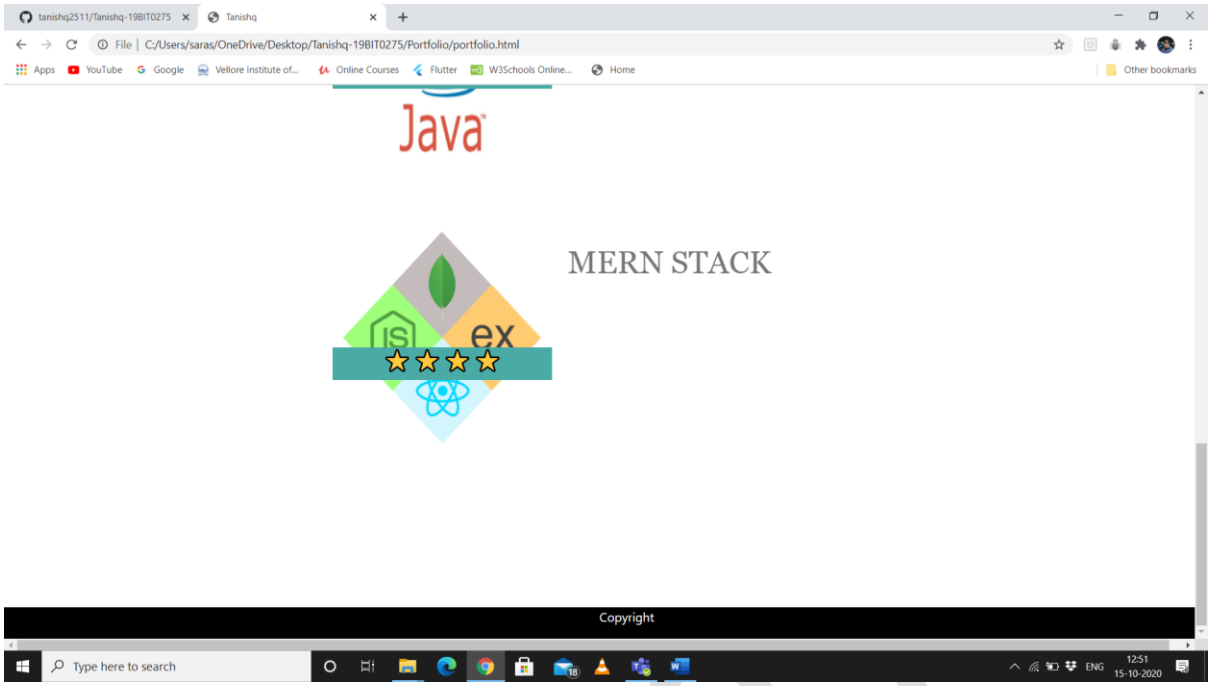
Creating Portfolio website

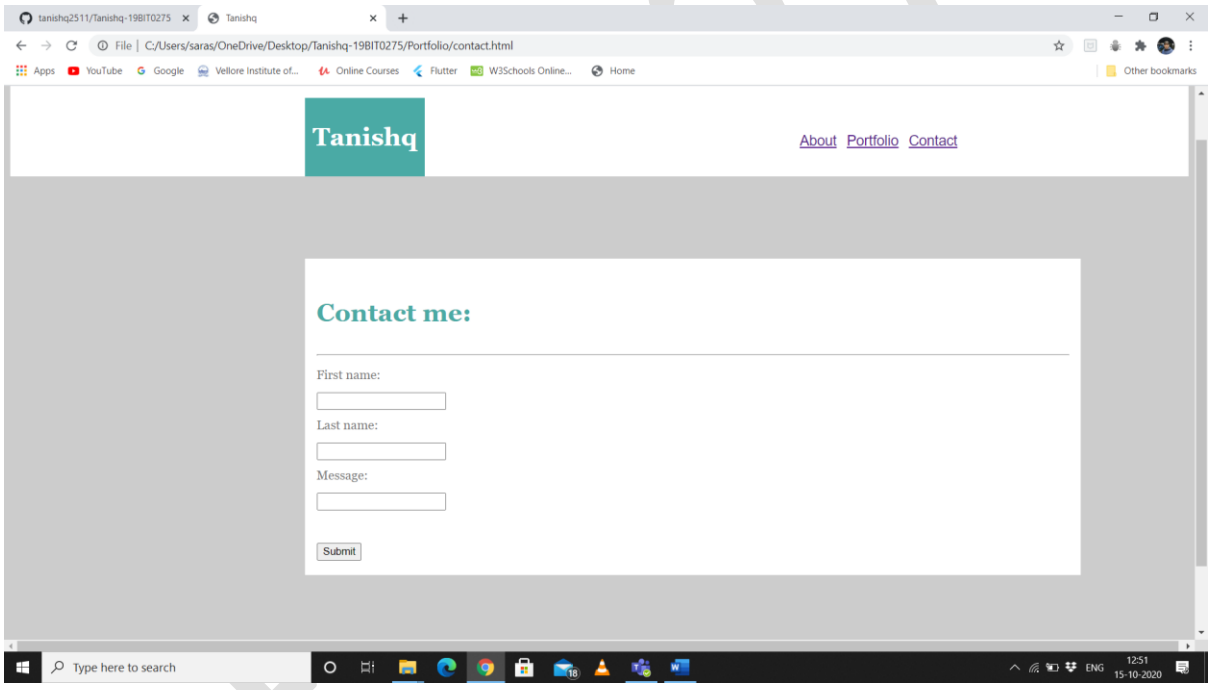
https://tanishq2511.github.io/Portfolio/



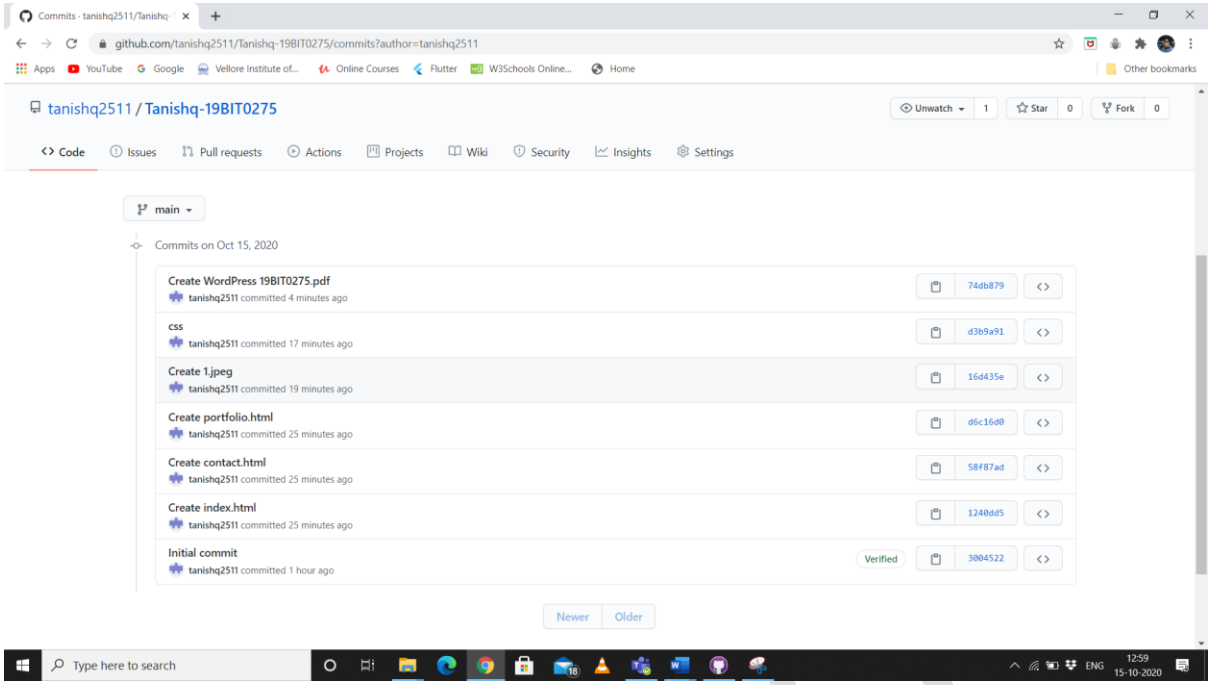

tanishq2511/Tanishq-19BIT0275
Tanishq
File | C:/Users/saras/OneDrive/Desktop/Tanishq-19BIT0275/Portfolio/portfolio.html
Apps
YouTube
Google
Vellore Institute of...
Online Courses
Flutter
W3Schools Online...
Home
Other bookmarks
Tanishq
About
Portfolio
Contact
Skills:
HTML, CSS, JS, PHP, SQL
Type here to search
ENG
12:51
15-10-2020


tanishq2511/Tanishq-19BIT0275
Tanishq
File | C:/Users/saras/OneDrive/Desktop/Tanishq-19BIT0275/Portfolio/portfolio.html
Apps
YouTube
Google
Vellore Institute of...
Online Courses
Flutter
W3Schools Online...
Home
Other bookmarks
C++ and Python
JAVA
Java
Type here to search
ENG
12:51
15-10-2020


tanishq2511/Tanishq-19BIT0275
Tanishq
File | C:/Users/saras/OneDrive/Desktop/Tanishq-19BIT0275/Portfolio/portfolio.html
Apps
YouTube
Google
Vellore Institute of...
Online Courses
Flutter
W3Schools Online...
Home
Other bookmarks
Java
MERN STACK
Copyright
Type here to search
ENG
12:51
15-10-2020


tanishq2511/Tanishq-19BIT0275
Tanishq
File | C:/Users/saras/OneDrive/Desktop/Tanishq-19BIT0275/Portfolio/contact.html
Apps
YouTube
Google
Vellore Institute of...
Online Courses
Flutter
W3Schools Online...
Home
Other bookmarks
Tanishq
About
Portfolio
Contact
Contact me:
First name:
Last name:
Message:
Submit
Type here to search
ENG
12:51
15-10-2020

Version History



Advantages of GitHub

1. It makes it easy to contribute to your open source projects

To be honest, nearly every open-source project uses GitHub to manage their project. Using GitHub is free if your project is open source and includes a wiki and issue tracker that makes it easy to include more in-depth documentation and get feedback about your project. If you want to contribute, you just fork a project, make your changes and then send them a pull request using GitHub web interface.

2. Documentation

By using GitHub, you make it easier to get excellent documentation. Their help section and guides have articles for nearly any topic related to git that you can think of.

3. Showcase your work

Are you a developer and wishes to attract recruiters? GitHub is the best tool you can rely on for this. Today, when searching for new recruits for their project, most companies look into the GitHub profiles. If your profile is available, you will have a higher chance of being recruited even if you are not from a great university or college.

4. Markdown

Markdown allows you to use a simple text editor to write formatted documents. GitHub has revolutionized writing by channeling everything through Markdown: from the issue tracker, user comments, everything. With so many other programming languages to learn for setting up projects, it's really a big benefit to have your content inputted in a format without having to learn yet another system.

5. GitHub is a repository

This was already mentioned before, but it's important to note, GitHub is a repository.
What this means that it allows your work to get out there in front of the public. Moreover, GitHub is one of the largest coding communities around right now, so it's wide exposure for your project.

6. Track changes in your code across versions

When multiple people collaborate on a project, it's hard to keep track revisions—who changed what, when, and where those files are stored. GitHub takes care of this problem by keeping track of all the changes that have been pushed to the repository. Much like using Microsoft Word or Google Drive, you can have a version history of your code so that previous versions are not lost with every iteration.

7. Integration options

GitHub can integrate with common platforms such as Amazon and Google Cloud, services such as Code Climate to track your feedback, and can highlight syntax in over 200 different programming languages.

Disadvantages of GitHub

Security

GitHub does offer private repositories, but this isn't necessarily perfect for many. For high value intellectual property, you're putting all of this in the hands of GitHub as well as anyone who

has a login, which like many sites has had security breaches before and is targeted constantly. It is often better than nothing, but it's not perfect. In addition, some clients/employers will only allow code on their own secure internal Git as a matter of policy.

Pricing

Some of GitHub features, as well as features on other online repositories, are locked behind a SaaS paywall. If you have a large team, this can add up fast. Those who already have a dedicated IT team and their own internal servers are often better off using their own internal git for cost reasons, but for most the cost isn't outrageous.

Features that need to be added in GitHub

- Unlimited free private repositories is something dev teams who are making a product or a service at an early stage really appreciate.
- Mercurial support is attractive to a bunch of dev teams, especially Python-centric teams.
- Unlimited private repositories unlike Github where one has to pay for private repos (If you're a student, then github allows to you have a limited number of private repositories by providing them an .edu email ID)
- Aerobatic hosting for Bitbucket: Similar to Github pages, this integration allows you to host your repos as websites. It has a CD/CI option which Github lacks.

Comparing other version control applications

BitBucket

- BitBucket allows developers to use Git or Mercurial as their VCS.

- Atlassian BitBucket does not charge for personal repositories (public or private), but there is a fee if you have more than 5 users.
- Similar to its competitor, BitBucket can also be run on your own servers via BitBucket Server.
- BitBucket has great integration with Jira (Atlassian's issue tracker and project management tool), so that's a big plus for Jira users.
- BitBucket offers a free issue tracker and wiki, but, to be honest, these are far from their commercial offerings, Jira and Confluence.
- Atlassian now has a build and testing service. See Atlassian Documentation for details on Pipelines.

To be honest, both work and it really comes down to the features you are specifically looking for. I have used both.

Mercurial

Mercurial was introduced close to Git and is also a distributed peer-to-peer system. Mercurial is a distributed revision-control tool which is written majorly in Python and some small sections in C Language.

Features

- High performance and scalability.

- Advanced branching and merging capabilities.
- Fully distributed collaborative development.
- Decentralized
- Handles both plain text and binary files robustly.
- Possesses an integrated web interface.

Pros

- Fast and powerful
- Easy to learn
- Lightweight and portable.
- Conceptually simple

Cons

- Partial checkouts are not allowed.
- Quite problematic when used with additional extensions.

CVS-Concurrent Versions System

CVS, also known as the **Concurrent Versioning System**, is a free client-server revision control system in the field of software. It was developed in the UNIX operating system environment and is available in both Free Software Foundation and commercial versions.

Features

- Client-server repository model.
- Multiple developers might work on the same project parallelly.
- CVS client will keep the working copy of the file up-to-date and requires manual intervention only when an edit conflict occurs
- Keeps a historical snapshot of the project.
- Anonymous read access.

- ‘Update’ command to keep local copies up to date.
- Can uphold different branches of a project.
- Excludes symbolic links to avoid a security risk.
- Uses delta compression technique for efficient storage.

Pros

- Excellent cross-platform support.
- Robust and fully-featured command-line client permits powerful scripting
- Helpful support from vast CVS community
- allows good web browsing of the source code repository
- It’s a very old, well known & understood tool.
- Suits the collaborative nature of the open-source world splendidly.

Cons

- No integrity checking for source code repository.
- Does not support atomic check-outs and commits.
- Poor support for distributed source control.
- Does not support signed revisions and merge tracking.