ITE1008 – Open Source Programming

Digital Assignment – I Fall Semester 2020 - 2021

GitHub Version Control

Submitted to: Mr.Jayakumar S

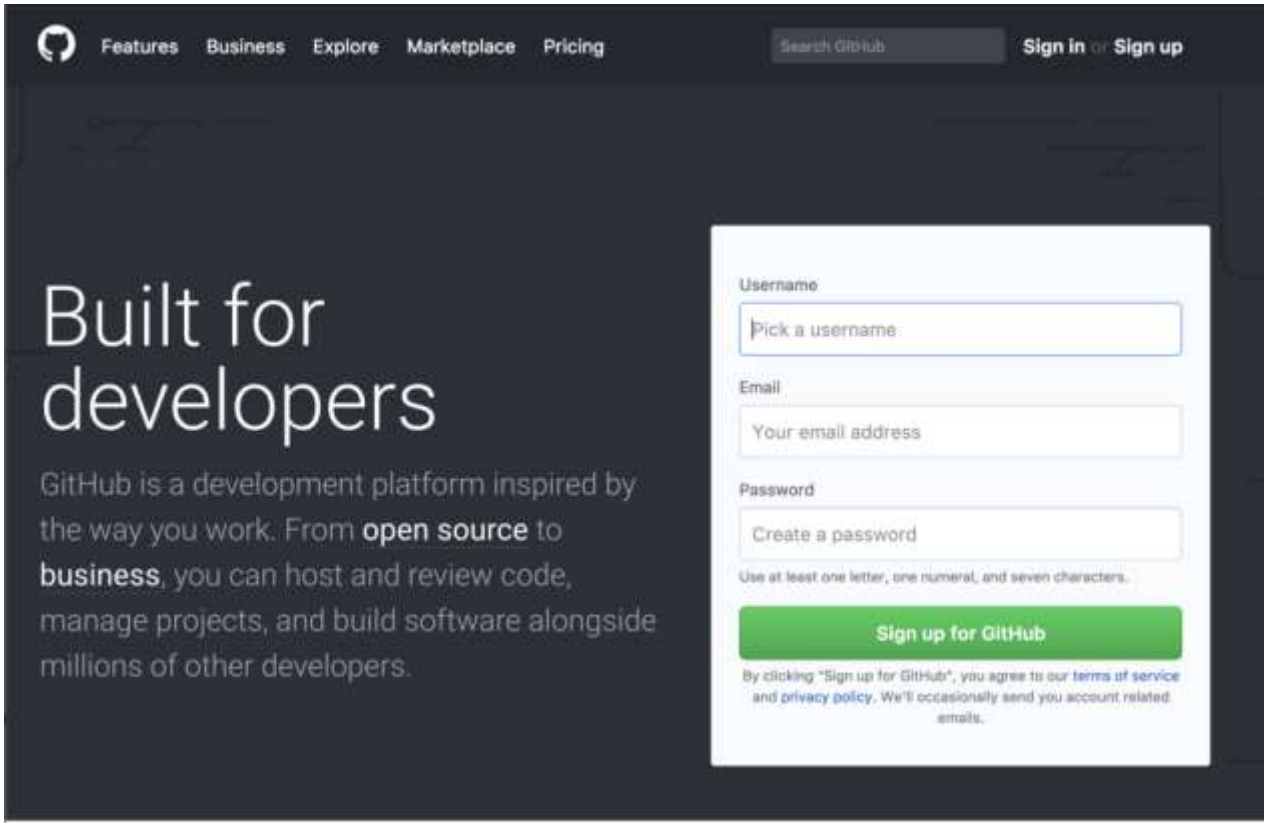
Submitted by: Dhruv Sood

19BIT0234

GitHub working methodology

GitHub is a web-based platform, which is used for version control. Git simplifies the process of working with other people and makes it easy to collaborate on projects. Team members can work on files and easily merge their changes in with the master branch of the project.

Step 1: *Create a GitHub account on GitHub.com*

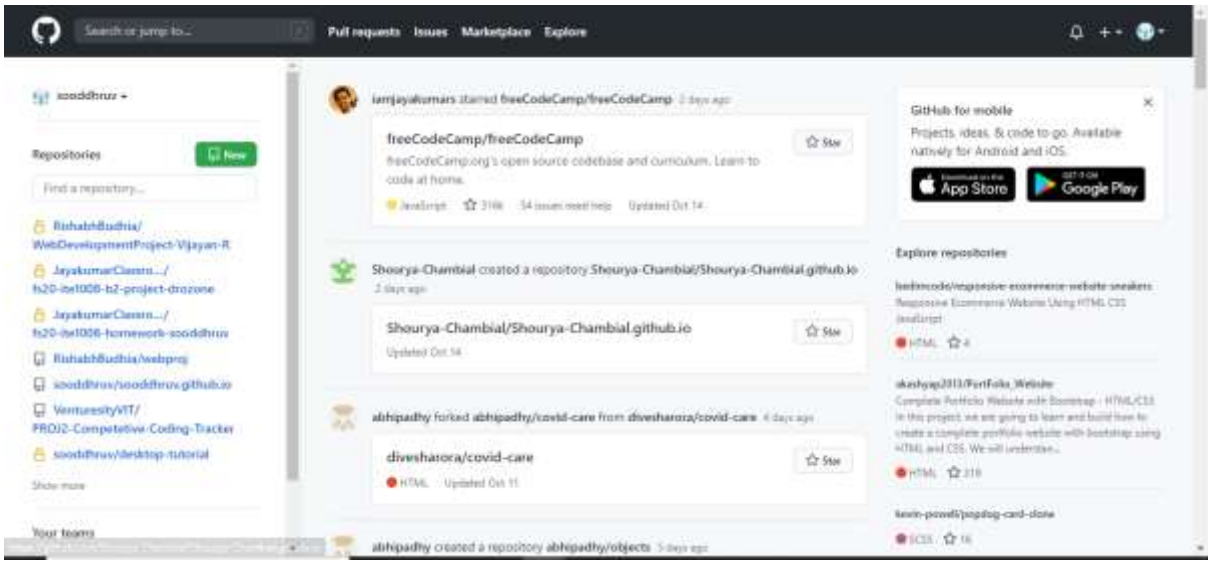


Step 2: *Create a repository*

A repository is a storage space where your project lives. It can be local to a folder on your computer, or it can be a storage space on GitHub or another online host. You can keep code files, text files, images or any kind of a file in a repository. You need a GitHub repository when you have done some changes and are ready to be uploaded.

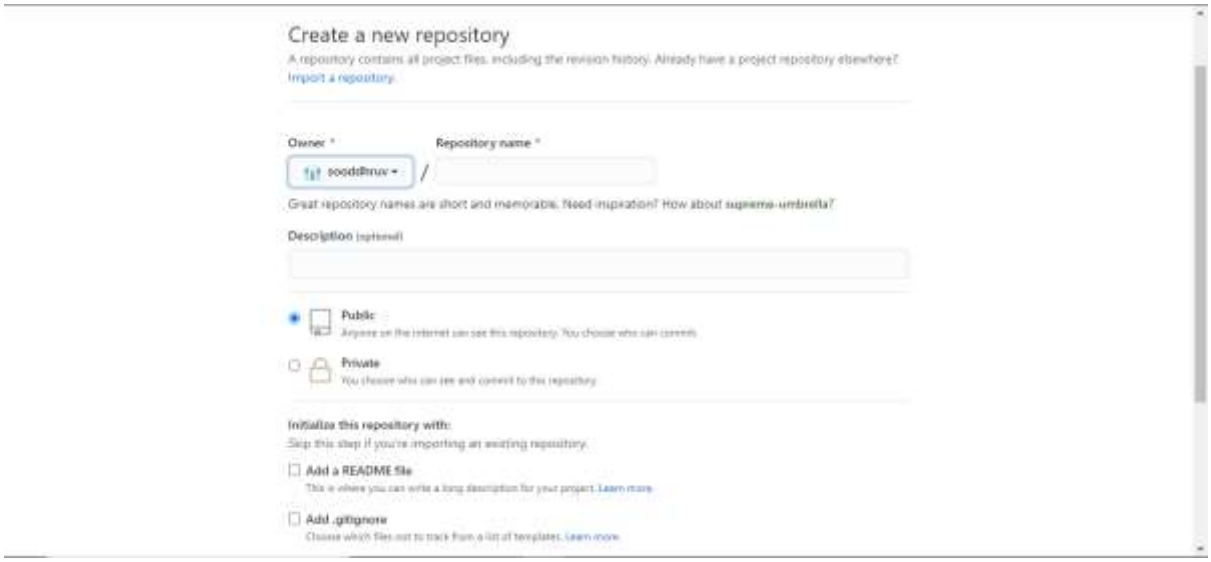
To create a new repository

Just press on new



And create a repository with any name with .github.io

For example dhruv.github.io

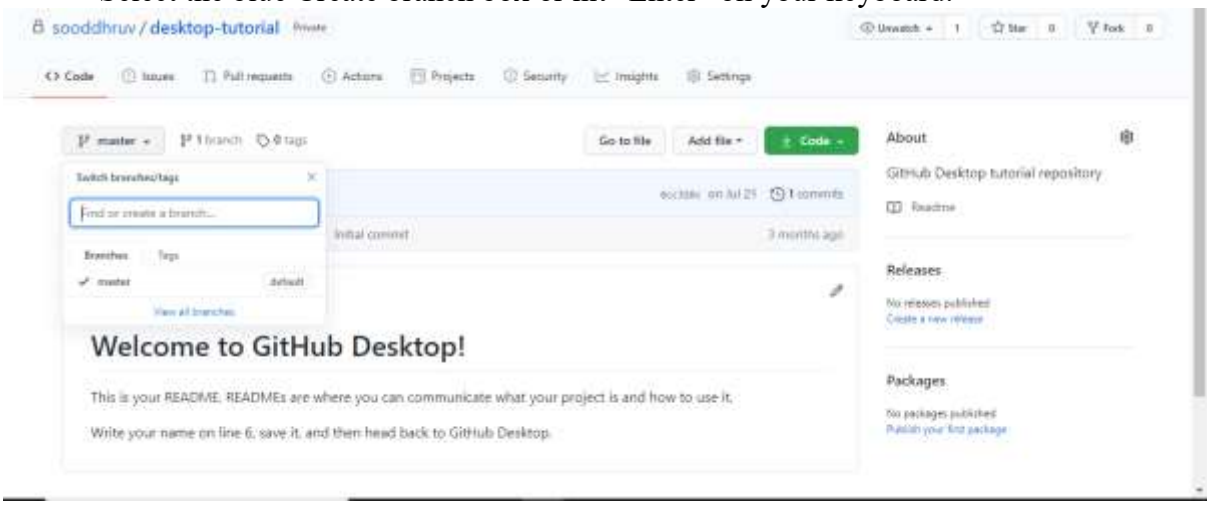


Step 3: Create a branch

Branches help you to work on different versions of a repository at one time. Let's say you want to add a new feature (which is in the development phase), and you are afraid at the same time whether to make changes to your main project or not. This is where git branching comes to rescue. Branches allow you to move back and forth between the different states/versions of a project.

To create a new branch: -

- Go to your new repository.
- Click the drop down at the top of the file list that says branch: main.
- Type a branch name of your wish into the new branch text box.
- Select the blue Create branch box or hit "Enter" on your keyboard.



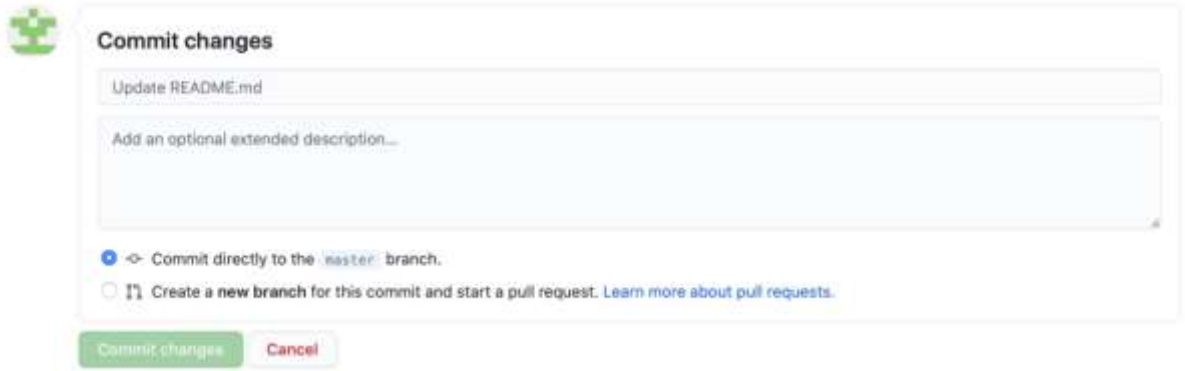
How to use GitHub Operations:

Make and commit changes

This operation helps you to save the changes in your file. When you commit a file, you should always provide the message, just to keep in the mind the changes done by you. Though this message is not compulsory but it is always recommended so that it can differentiate the various versions or commits you have done so far to your repository.

To make and commit changes: -

- In the editor, write a bit about yourself.
- Write a commit message that describes your changes.
- Click Commit changes button.



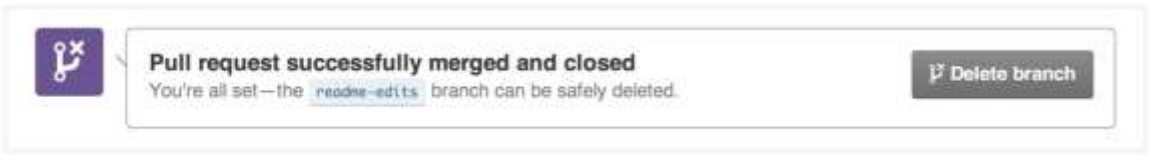
Open a pull request

Pull command is the most important command in GitHub. It tells the changes done in the file and request other contributors to view it as well as merge it with the master branch. Once the commit is done, anyone can pull the file and can start a discussion over it. Once it's all done, you can merge the file. Pull command compares the changes which are done in the file and if there are any conflicts, you can manually resolve it.

Merge Command

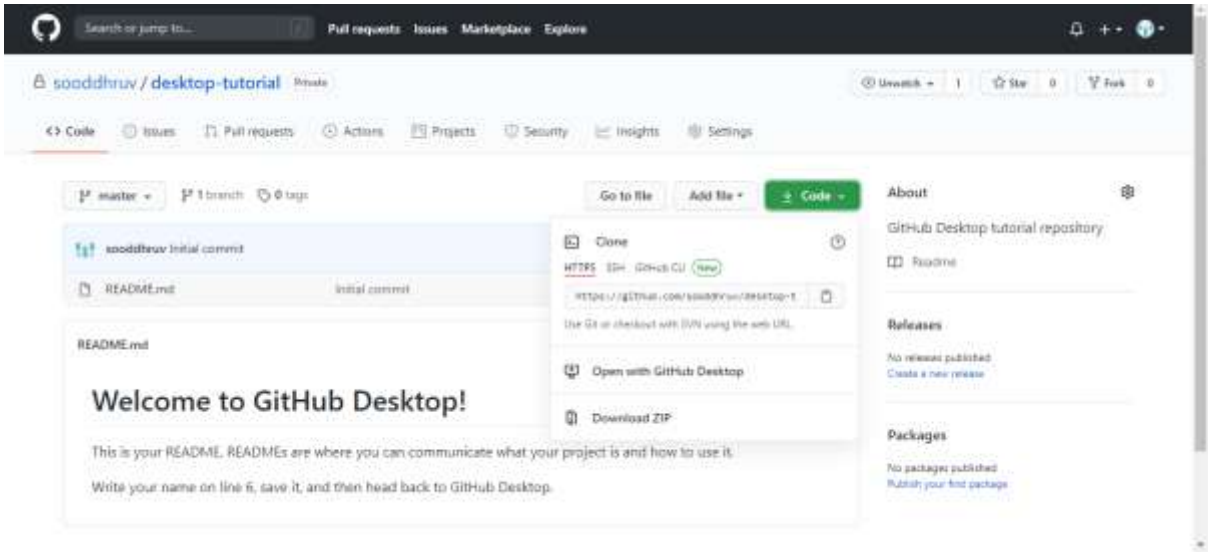
Here comes the last command which merge the changes into the main master branch. We saw the changes in pink and green color, now let's merge the "readme- changes" file with the master branch/ read-me. Go through the below steps to merge pull request.

- Click on "Merge pull request" to merge the changes into master branch.
- Click "Confirm merge".



Cloning and forking GitHub repository

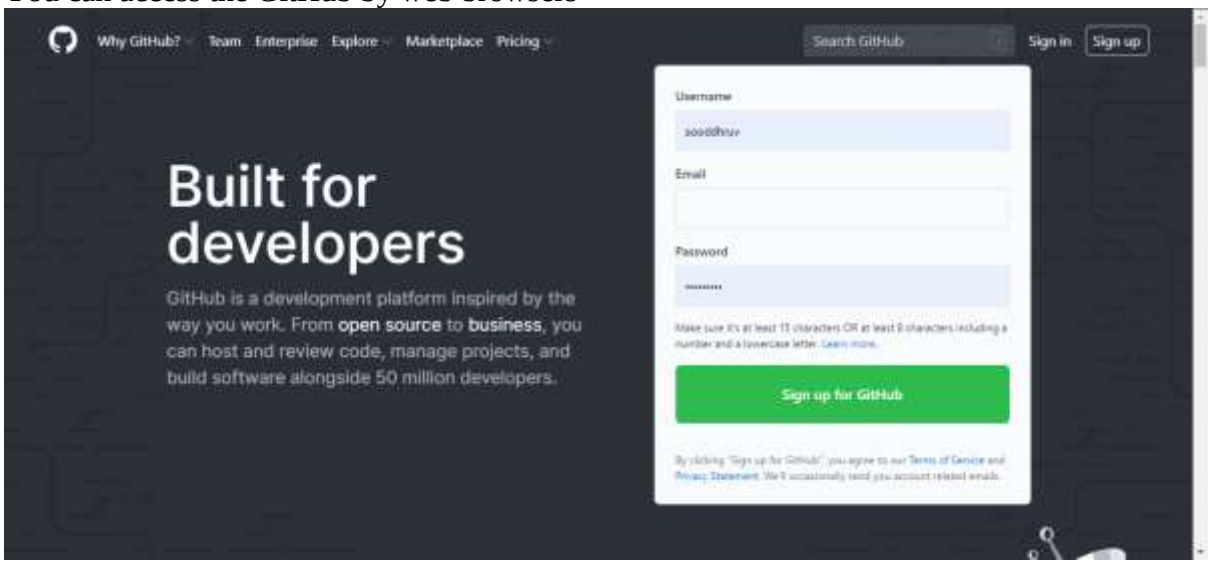
If you want to use some code present in a public repository, you can directly copy the contents by cloning or downloading.
If, you need some code which is present in a public repository, under your repository and GitHub account. For this, we need to fork a repository.
Changes done to the original repository will be reflected back to the forked repository.
If you make a change in forked repository, it will not be reflected to the original repository until and unless you have made a pull request.



By what means we can use GitHub

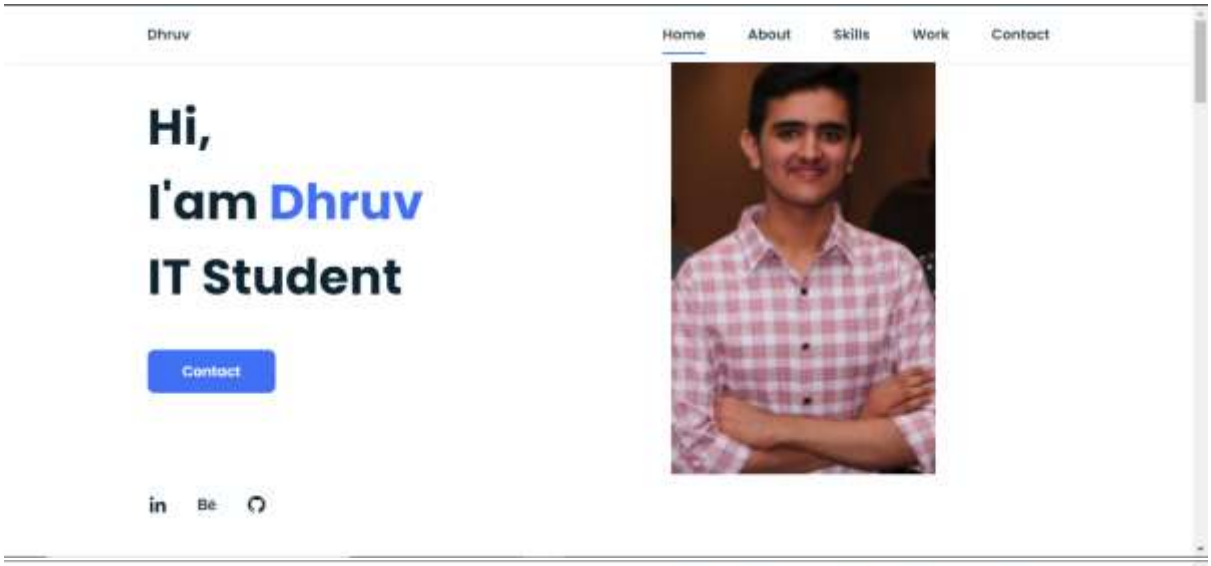
There are two ways to access the GitHub both are really simple and user friendly.
So two ways as follows

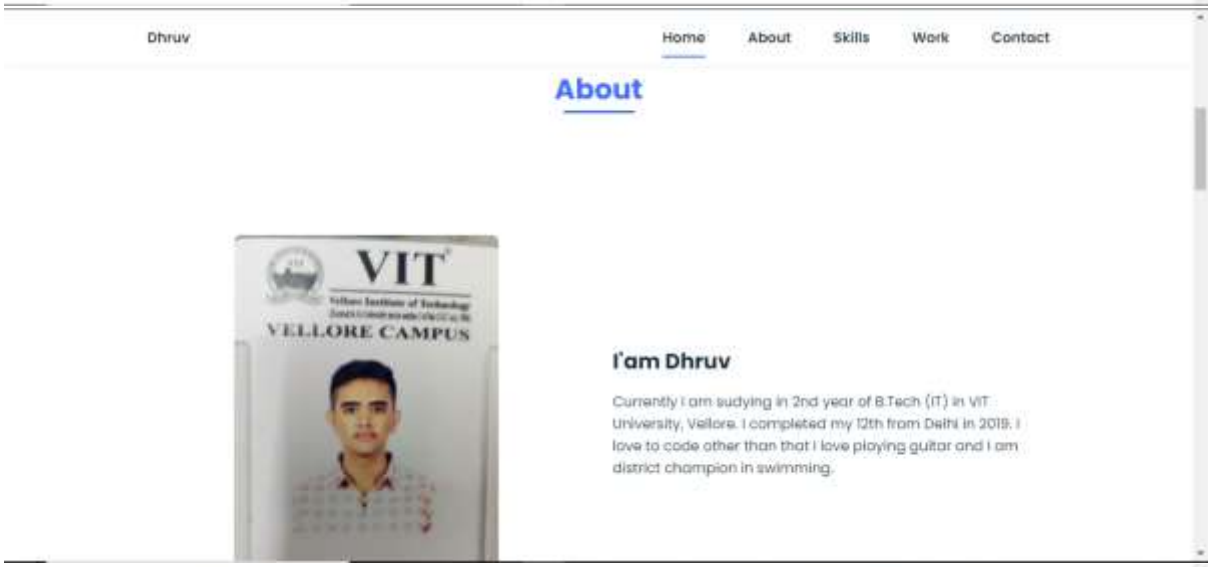
1) GitHub web version
   You can access the GitHub by web browsers

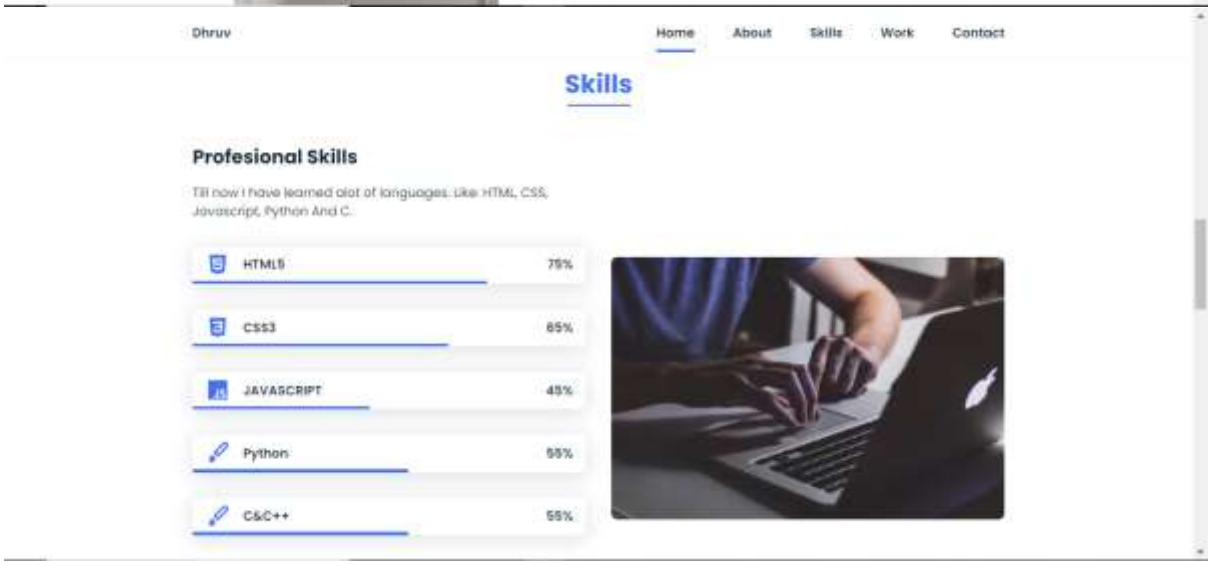


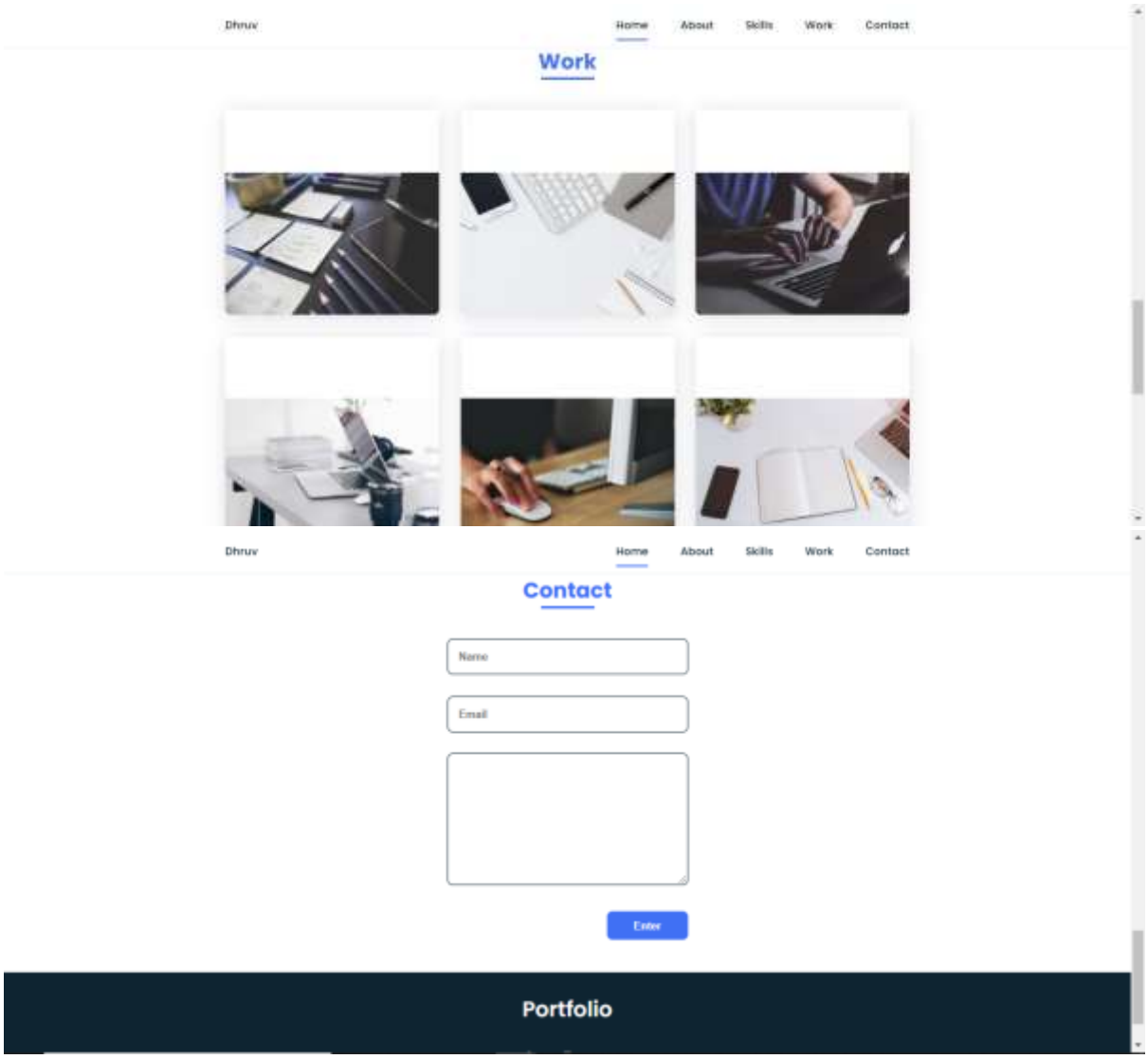
2) You can also access the GitHub by it's app or SO CALLED DESKTOP VERSION

My portfolio

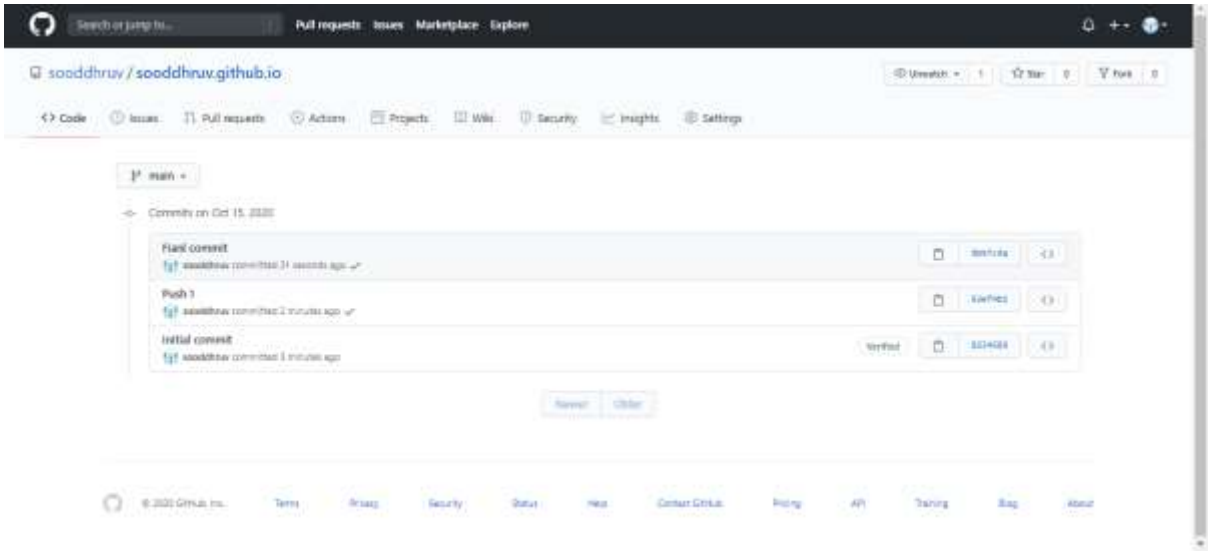

Dhruv
Home
About
Skills
Work
Contact
Hi,
I'am Dhruv
IT Student
Contact
Be


Dhruv
Home
About
Skills
Work
Contact
About
VIT
VELLORE CAMPUS
I'am Dhruv
Currently I am sudying in 2nd year of B.Tech (IT) in VIT University, Vellore. I completed my 12th from Delhi in 2019. I love to code other than that I love playing guitar and I am district champion in swimming.


Dhruv
Home
About
Skills
Work
Contact
Skills
Profesional Skills
Till now I have learned alot of languages. Like HTML, CSS, Javascript, Python And C.
HTML5
75%
CSS3
65%
JAVASCRIPT
45%
Python
55%
C&C++
55%



Version History



Advantages of GitHub

- It is free and it is open source: GitHub is completely free and we can use it without paying and since it is an open source you can download the source code and can make changes as per the requirements.
- Multiple developers can work: GitHub allows multiple developers to work on a single project at a time. It helps all the team members to work together on a single project at a time from different locations.
- It is fast: Since most of the operations are preferred locally, it allows huge benefit in terms of speed
- It provides good backup: Here chance of losing data is very low as it provides the multiple copies of it.
- 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.
- 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.

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.
- Reviewing large pull requests can be tedious and it can be tough to identify recent changes (e.g. a one line change) in new files or files with lots of changes.
- You have to be careful with merge operations; a bad merge can be painful to reverse.

Difference between different version control applications:

GitHub	GitLab	BitBucket
• GitHub was launched in 2008. It is git based repository hosting platform.	• GitLab was launched as a project in 2011 providing an alternative to the available repository management solutions. But the site GitLab.com was launched in 2012	• Bitbucket was launched in 2008 initially supporting Mercurial Projects. In 2010, it was acquired by Atlassian and from 2011 it also started to support Git hosting.
• Is only hosts projects that use the Git VCS	• It do almost everything that GitHub does, so it is like Github, but here we have free private repositories that github doesn't	• It supports the Mercurial VCS(version control system) in addition to Git
• It is free for public repositeries and for private one it is paid.	• GitLab Community Edition is free and open sourced.	• It is not open source but by buying the self-hosted version the full source code is provided.
• GitHub is written using Ruby and Erlang	• GitLab is written using C, Perl, Tcl, Python.	• Bitbucket is written in Python and uses the Django web framework.

• GitHub provides CI pipeline for sharing of codes.	• GitLab provides free CI for easy access to users.	• This platform does not give free CI to its users.

Features that should be added to GitHub

- Doesn't offer very good API development
- Slightly expensive for those in search of private repository
- It does not have features as compared to GitLab or other applications.