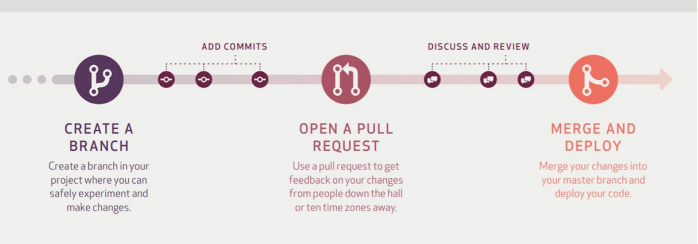
Digital Assignment - 1

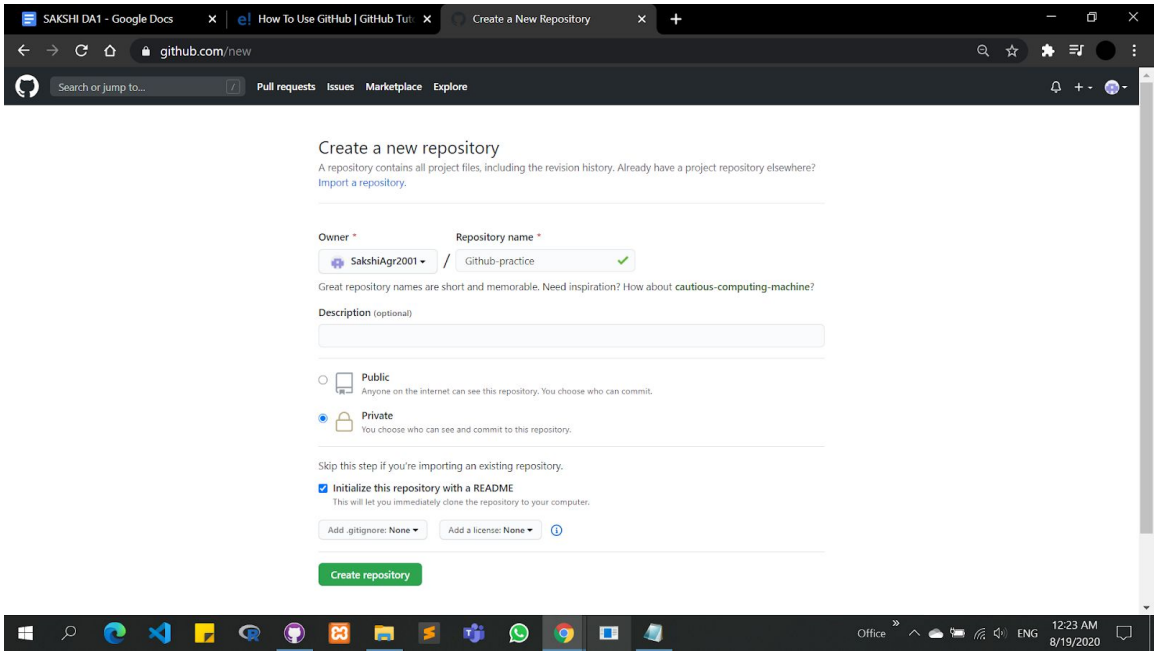
Q1. Write down the step by step process of GitHub working methodology and different ways to access GitHub.

Working methodology

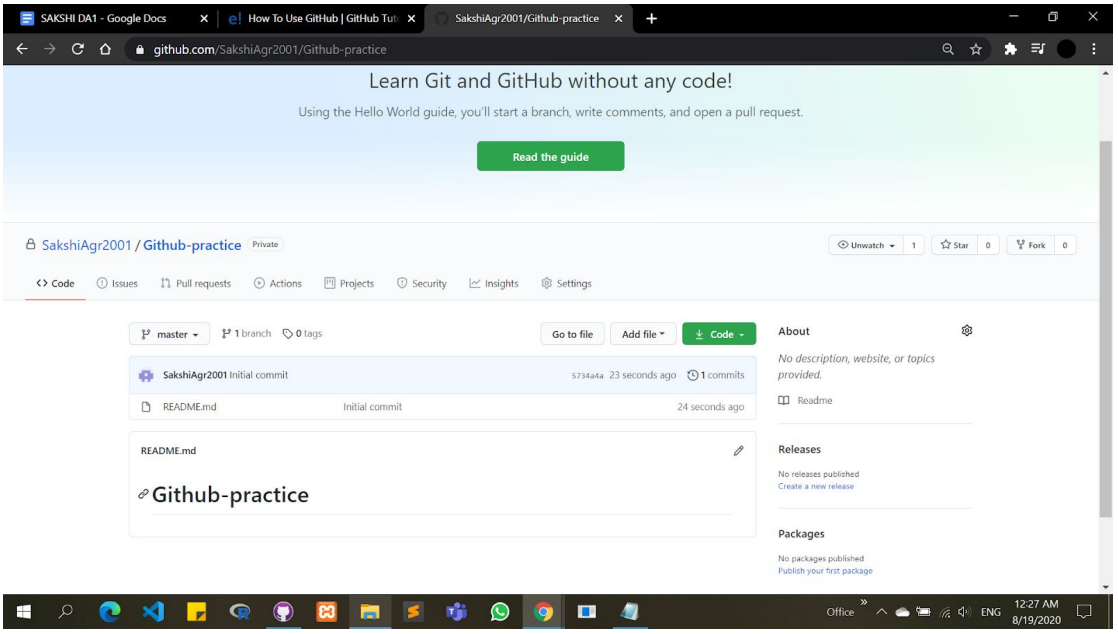


1. **create a GitHub Repository**

- Go to the link: https://github.com/ . Fill the sign up form and click on “Sign up for Github”.
- Click on “Start a new project”.
- Enter any repository name and click on “Create Repository”. You can also give a description to your repository (optional).
- Now you can either choose public (anyone can view the content)
- or private (you can choose who can view the content).



- your repository is successfully created! It will look like the below screenshot



2.Create Branches and Perform Operations

- Click on the dropdown “Branch: master”
- As soon as you click on the branch, you can find an existing branch or you can create a new branch.

<u>OPERATIONS</u>

1. ***Commit Command:***

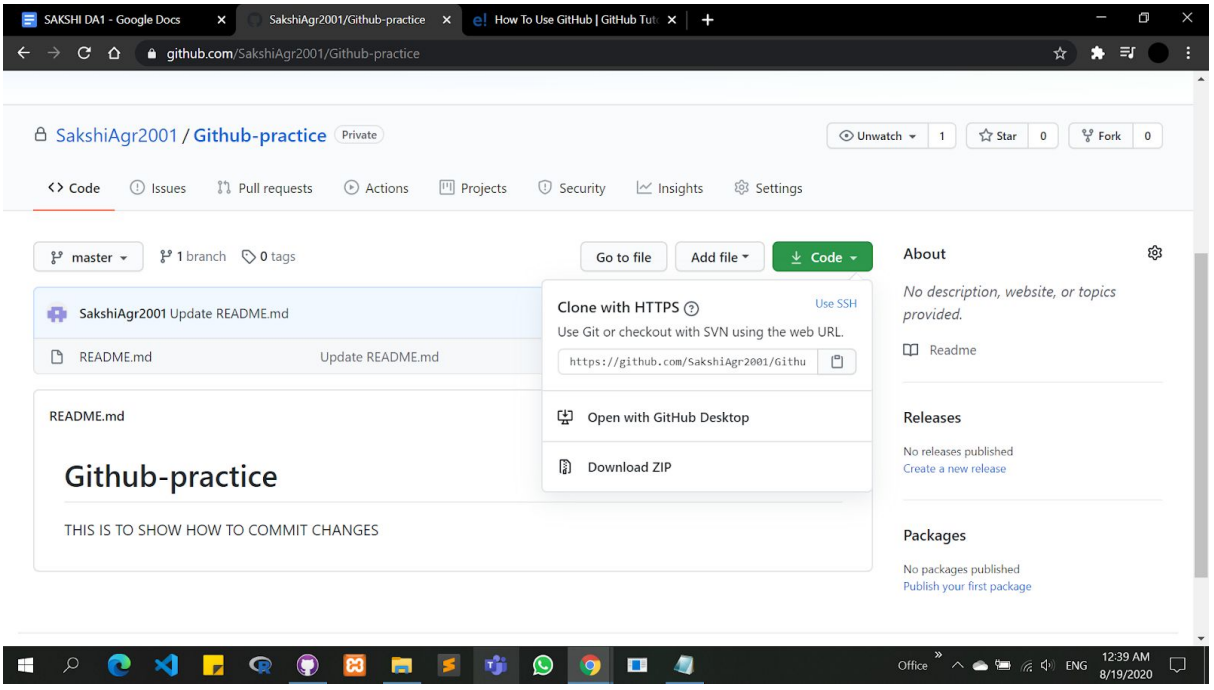
- Click on the “readme- changes” file which we have just created.
- Click on the “edit” or a pencil icon in the right most corner of the file.
- Once you click on that, an editor will open where you can type in the changes or anything.
- Write a commit message which identifies your changes.
- Click commit changes in the end.

2. ***Pull Command***

- Click the ‘Pull requests’ tab.
- Click ‘New pull request’.
- Once you click on pull request, select the branch and click ‘readme- changes’ file to view changes between the two files present in our repository.
- Click “Create pull request”.
- Enter any title, description to your changes and click on “Create pull request”..

3. Merge Command

- Click on “Merge pull request” to merge the changes into master branch.
- Click “Confirm merge”.
- You can delete the branch once all the changes have been incorporated and if there are no conflicts.



3. Cloning and Forking GitHub Repository Cloning is very simple in github. It can easily be cloned by copying the url and opening it with github desktop or zip.

fork a repository

- Go to Explore and search for public repositories.
- Click “fork”. Note that this “tangent” repository is already forked 27 times and it is under “google” account. Refer the below image for better understanding.

Q3. Write down the pros and cons of GitHub

PROS

- GitHub supports Markdown for the issue tracker, user comments, wikis – everything.
- It is a big benefit to have your content inputted in a format without having to learn yet another system.
- ***GitHub flavored markdown*** – a feature that adds changes to the usual markdown in order to make it more useful in programming environments.
- ***Well-padded*** guide in GitHub helps us in finding any section of article related to git.
- ***Gists*** a feature in github which lets you convert one or several files into a working git repository. This helps sharing and tracking changes made to configuration.

- Github helps in collaboration. It is beneficial to those working collaboratively on a project that are not part of a professional environment – particularly open source projects.
- It provides a simple way to have their code and version history available online regardless of what happens to the local machine.
- Compatible with existing systems and protocols like HTTP, FTP, ssh.
- Easily maintainable and robust

CONS

- Complex and the bigger history log become difficult to understand.
- Does not support keyword expansion and timestamp preservation
- Some of GitHub features, as well as features on other online repositories, are locked behind a SaaS paywall.
- Github does offer private repositories but isn't necessarily perfect for many.
- When creating 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.

Q4. List down the features that need to be added in GitHub.

1. Ability to apply a patch from a .patch file
2. Ability to manage tags
3. Ability to stash changes (stashing currently works when changing branches, but there is no dedicated UI to create a stash with custom name and apply a stash selectively / manage stashes.)
4. Ability to create a commit template with preset text that can be reused across various commits
5. Ability to delete multiple local/remote branches in a single action.
6. Ability to selectively stage/discard hunks
7. Ability to capture the creative process or for recording ideas.
8. Ability to track design. It can seem a little bit of a grey area when designs are needed to be translated into code or for when you need to export designs to a production setting.

Q5. Compare the minimum of three version control applications

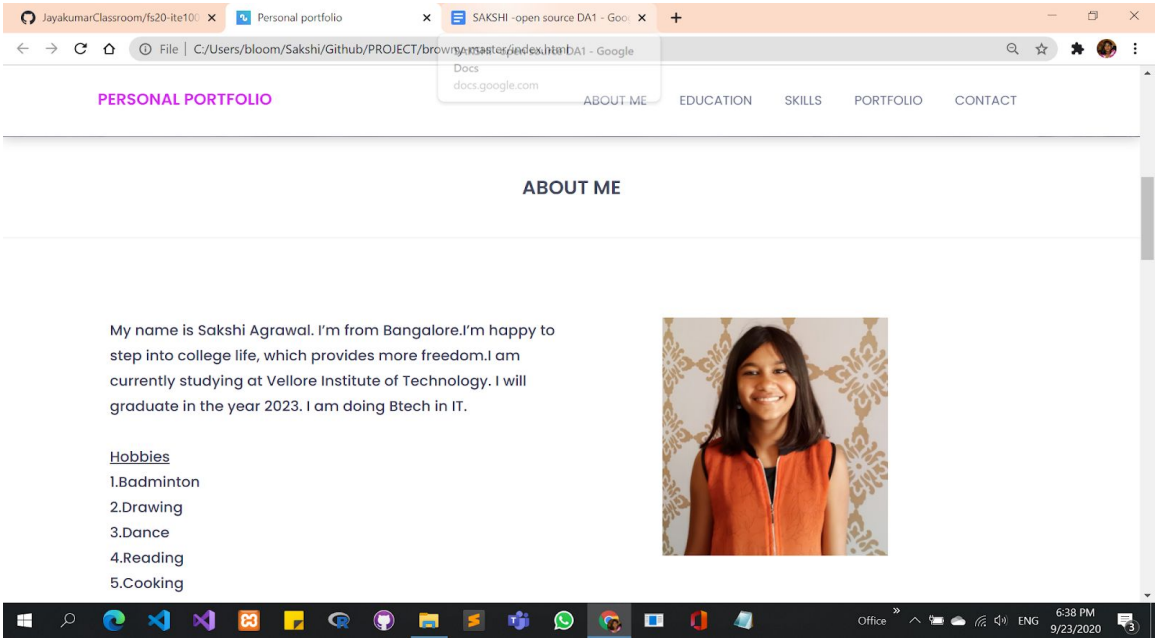
Version control applications:
These applications are used when multiple developers are continuously working or changing the source of code. It helps track and compare the code in different versions.

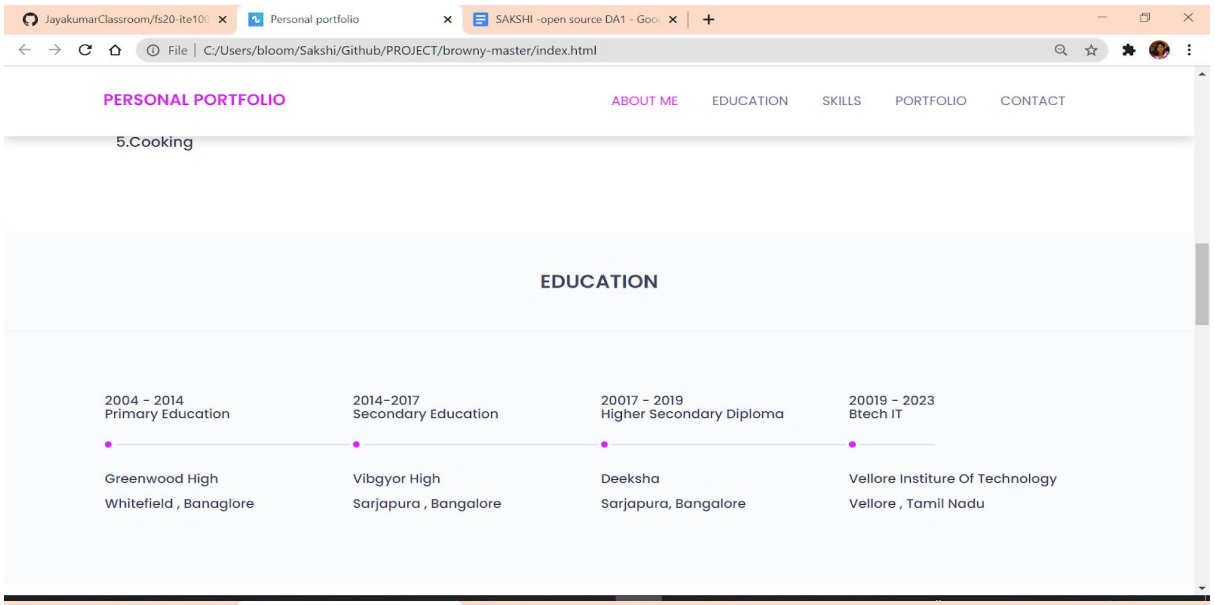
Github	CVS	TFS
Global Information Tracker	Concurrent Versions System	Team Foundation Server

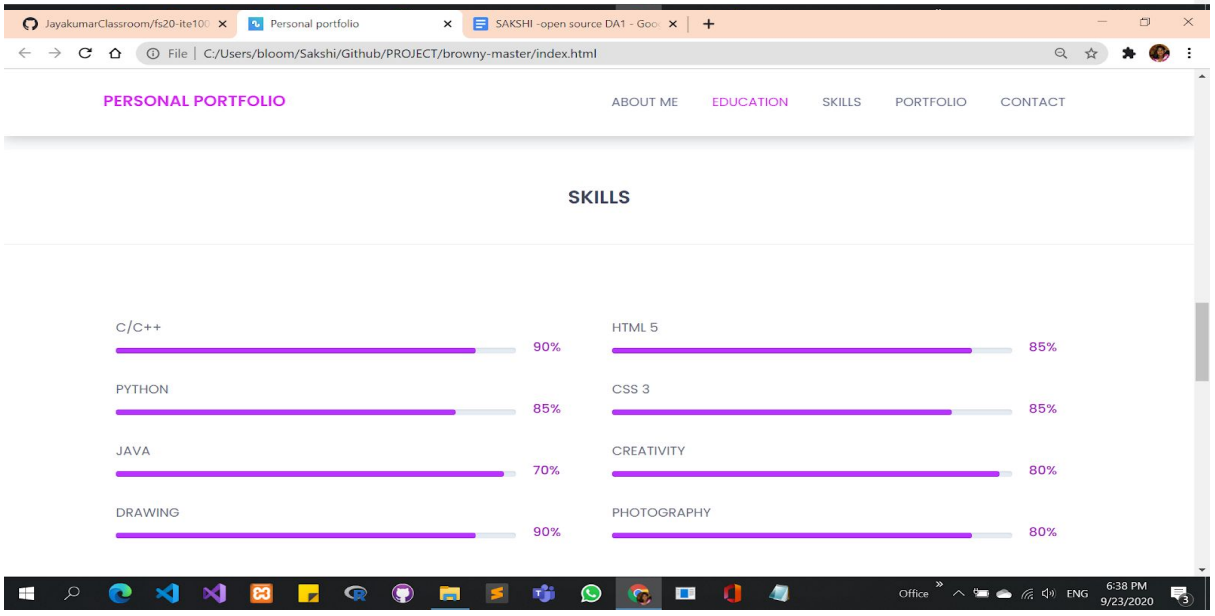
Git is one of the best version control tools that is available in the present market	CVS is one of the oldest version control systems and is a well-known tool among both commercial and open source developers.	TFS is a distributed, client-server repository model based version control tool provided by Microsoft.
Setting up a repository. Git stores repositories in .git directory in top directory of your project;	**Setting up a repository**. CVS require setting up CVSROOT, a central place for storing version control info for different projects (modules).	**Setting up a repository**. repository for different platforms from C# to Python, from Windows to Android; means a variety of languages and platforms.
Git is distributed as everyone has a full copy of the whole repo and its history.	CVS is a version symbolic link. It is a free form versioned meta data.	TFS is a centralized version while
Git users do commits based on distributed full versions with difference checking (commit-before-merge)	CVS forces you to first update your working directory and resolve conflicts before allowing you to commit. (merge-before-commit)	TFS has frequent merge conflicts. It is mandatory to be connected to a central repository. Users “check-in” which invokes files.

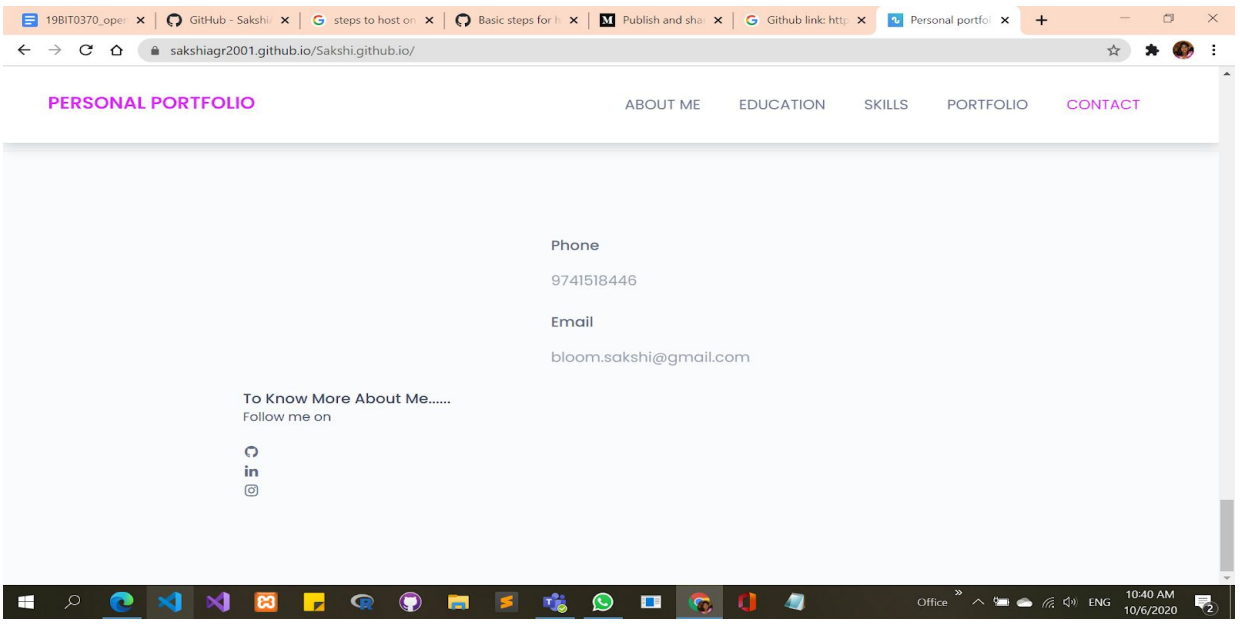
Open Source: Yes **Cost:** Free	**Open Source:** Yes **Cost:** Free	**Open Source:** No **Cost:** Free of cost for up to 5 users in the VSTS or for open source
Git provides a “stash area” away from items being committed.	Credentials are stored by default in $HOME/.cvspass this is a bit of a barrier (you have to know the name of the guest account, or pay attention to CVS server messages)	TFS provides a “shelf area” to hold local changes temporarily.

Q2 . Host your Personal Portfolio in GitHub and provide the screenshot of the project and version history.

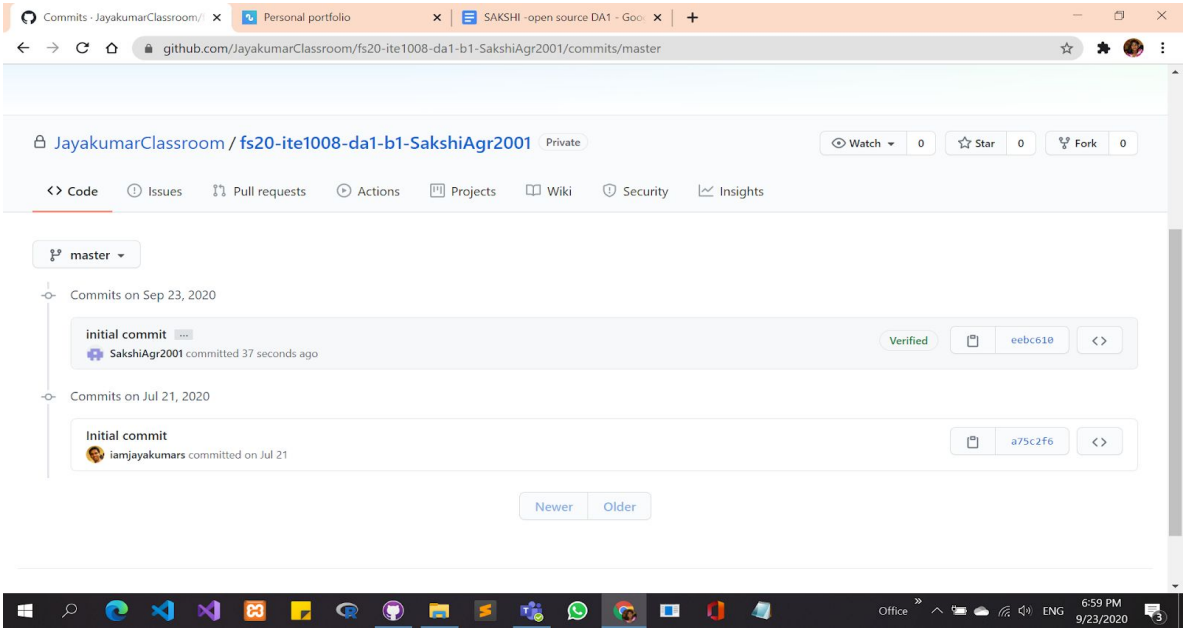



JayakumarClassroom/fs20-ite100
Personal portfolio
SAKSHI -open source DA1 - Goo
File | C:/Users/bloom/Sakshi/Github/PROJECT/browny-master/index.html
PERSONAL PORTFOLIO
ABOUT ME
EDUCATION
SKILLS
PORTFOLIO
CONTACT
5.Cooking
EDUCATION
2004 - 2014
Primary Education
Greenwood High
Whitefield , Banaglore
2014-2017
Secondary Education
Vibgyor High
Sarjapura , Bangalore
20017 - 2019
Higher Secondary Diploma
Deeksha
Sarjapura, Bangalore
20019 - 2023
Btech IT
Vellore Institure Of Technology
Vellore , Tamil Nadu


JayakumarClassroom/fs20-ite100
Personal portfolio
SAKSHI -open source DA1 - Goo
File | C:/Users/bloom/Sakshi/Github/PROJECT/browny-master/index.html
PERSONAL PORTFOLIO
ABOUT ME
EDUCATION
SKILLS
PORTFOLIO
CONTACT
SKILLS
C/C++
90%
PYTHON
85%
JAVA
70%
DRAWING
90%
HTML 5
85%
CSS 3
85%
CREATIVITY
80%
PHOTOGRAPHY
80%
Office
ENG
6:38 PM
9/23/2020


19BIT0370_oper
GitHub - Sakshi
steps to host on
Basic steps for h
Publish and sha
Github link: http
Personal portfol
sakshiagr2001.github.io/Sakshi.github.io/
PERSONAL PORTFOLIO
ABOUT ME
EDUCATION
SKILLS
PORTFOLIO
CONTACT
Phone
9741518446
Email
bloom.sakshi@gmail.com
To Know More About Me.....
Follow me on
Office
ENG
10:40 AM
10/6/2020

VERSION HISTORY



Github link: https://sakshiagr2001.github.io/Sakshi.github.io/

STEPS TO HOST ON GITHUB

1. Create your project's repository. Login to your GitHub account and go to https://github.com/new or click the New repository icon from your account homepage.
2. Name your repository username.github.io, replacing username with your GitHub username.
3. Create an index.html page by clicking the plus icon next to your repository name and typing the file name directly in the input box that appears.
4. Commit index.html. At the bottom of the page, there is a text input area to add a description of your changes and a button to commit the file
5. Link to your CSS file inside your HTML document's <head>. Go back to index.html and select the "Edit" button
6. Visit http://username.github.io to see your styled website.