VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

DIGITAL ASSIGNMENT 1

NAME	:	**M S RITIK KUMAR NAIDU**
REG.NO	:	**19BIT0195**
COURSE CODE	:	**ITE1008**
COURSE TITLE	:	**OPEN SOURCE PROGRAMMING**
SLOT	:	**B1+TB1**
GUIDED BY	:	**PROF. JAYAKUMAR SADHASIVAM**
GITHUB LINK	:	**https://github.com/ritiknaidu/Portfolio.git**

Case Study on GitHub Version Control

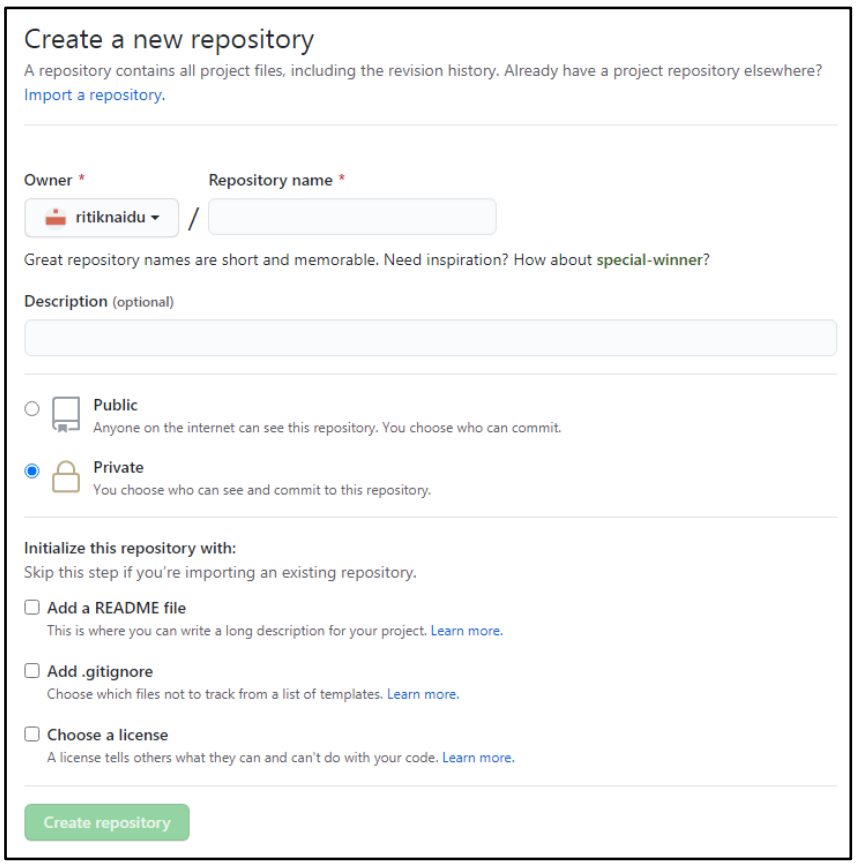
GitHub working methodology:

GitHub is a code hosting platform for version control and collaboration. It lets us and others work together on projects from anywhere. Version control allows us to keep track of our work and helps us to easily explore the changes we have made, be it data, coding scripts, notes, etc.

All the files relating to a project are stored in a single repository. A repository is used to organize a single project. Repositories can contain folders and files, images, videos, spreadsheets, and data sets etc.

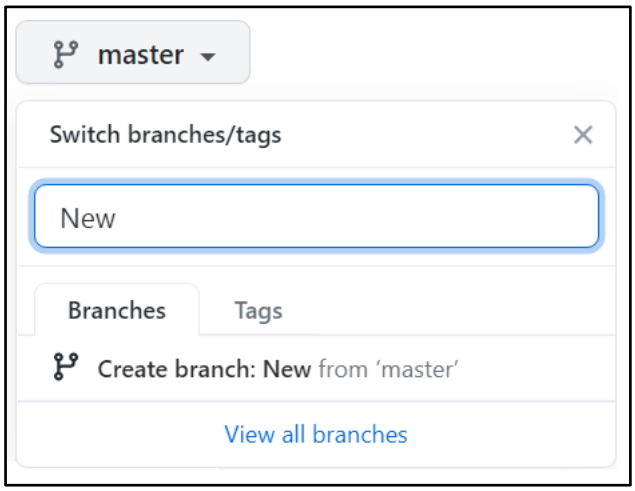
- Creating a repository:

Step 1 : In the upper right corner, click on the "+" icon and select "new repository".
Step 2 : Type the name of the repository and give it a description.
Step 3 : Set the status to either public or private.
Step 4 : Optionally one can initialize the repository with a readme file.

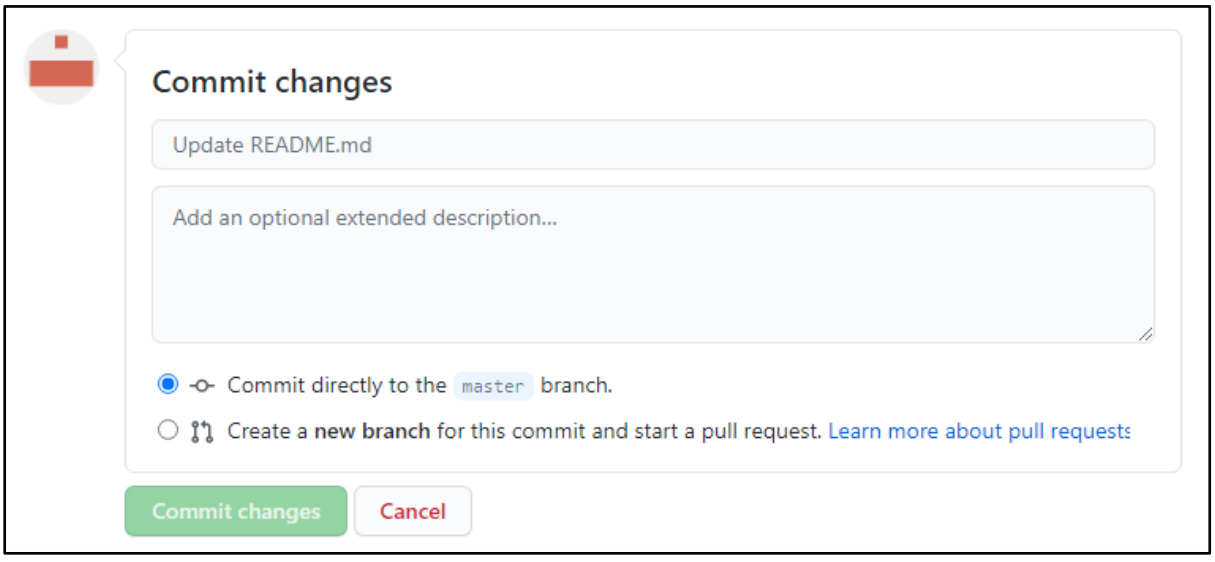


- Creating a branch: Branching is the way to work on different versions of a repository at one time.

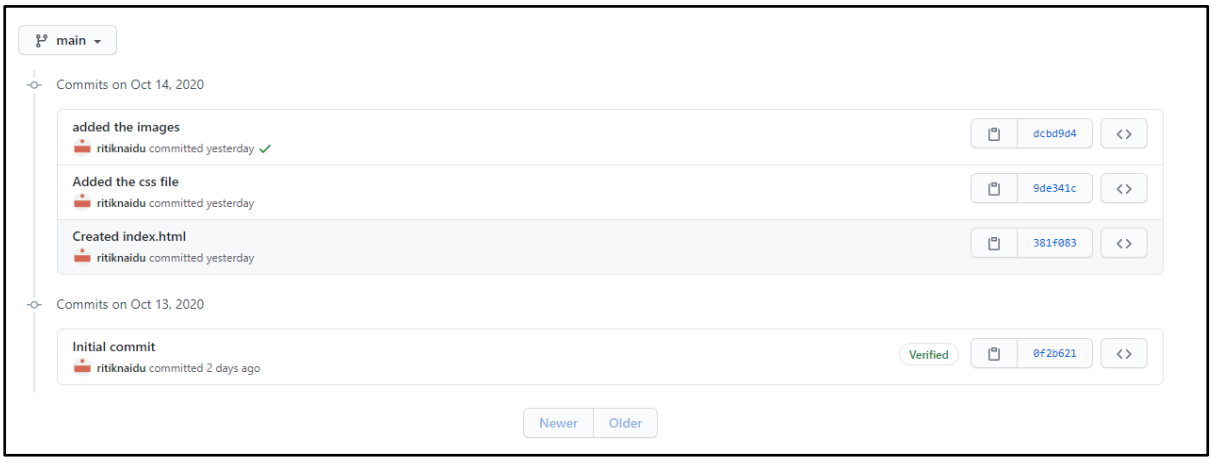
Step 1 : Go to the repository of choice.
Step 2 : Select the dropdown menu with the name master.
Step 3 : In the create new branch text box type the name of the branch.
Step 4 : Hit Enter.

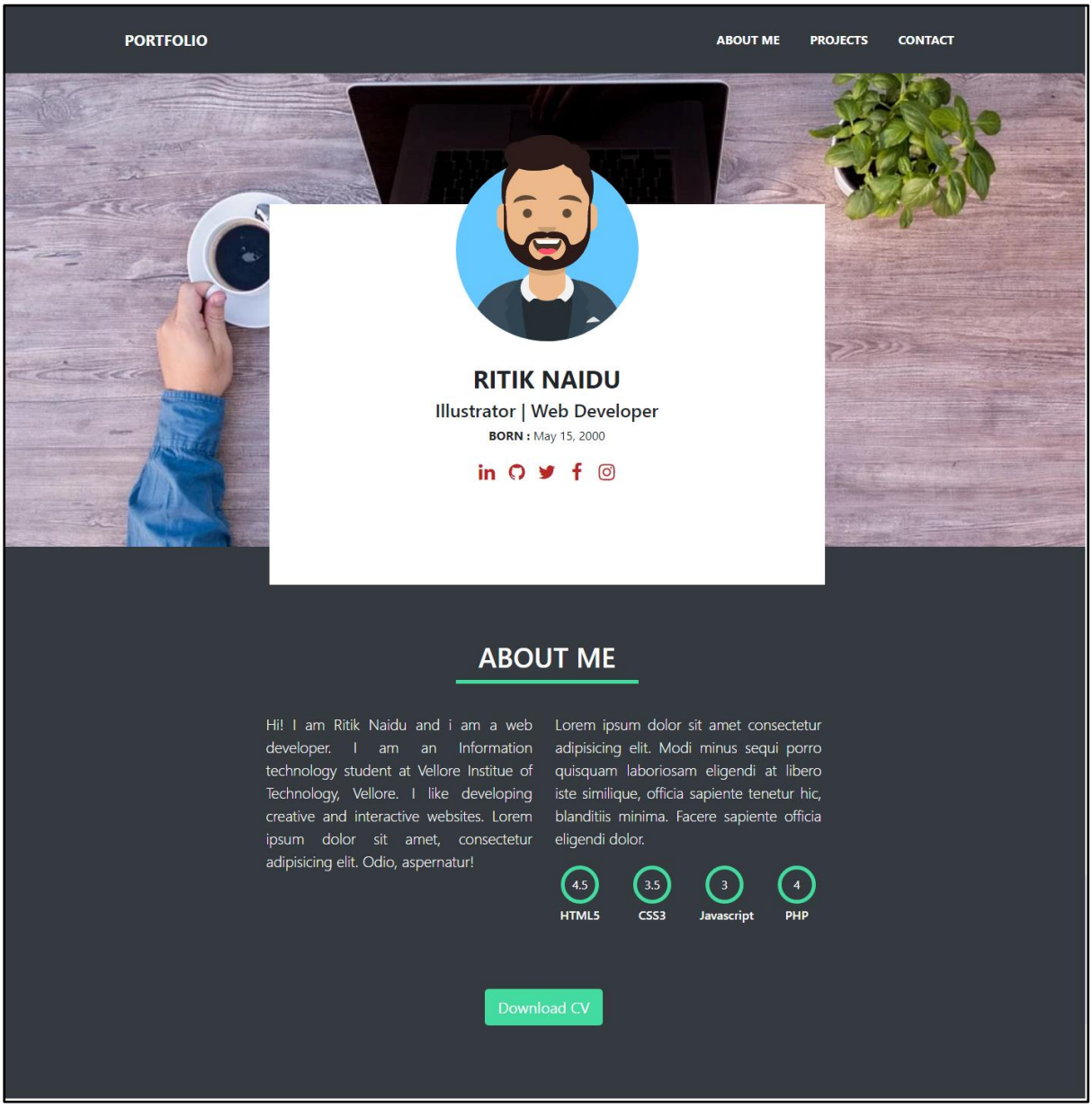


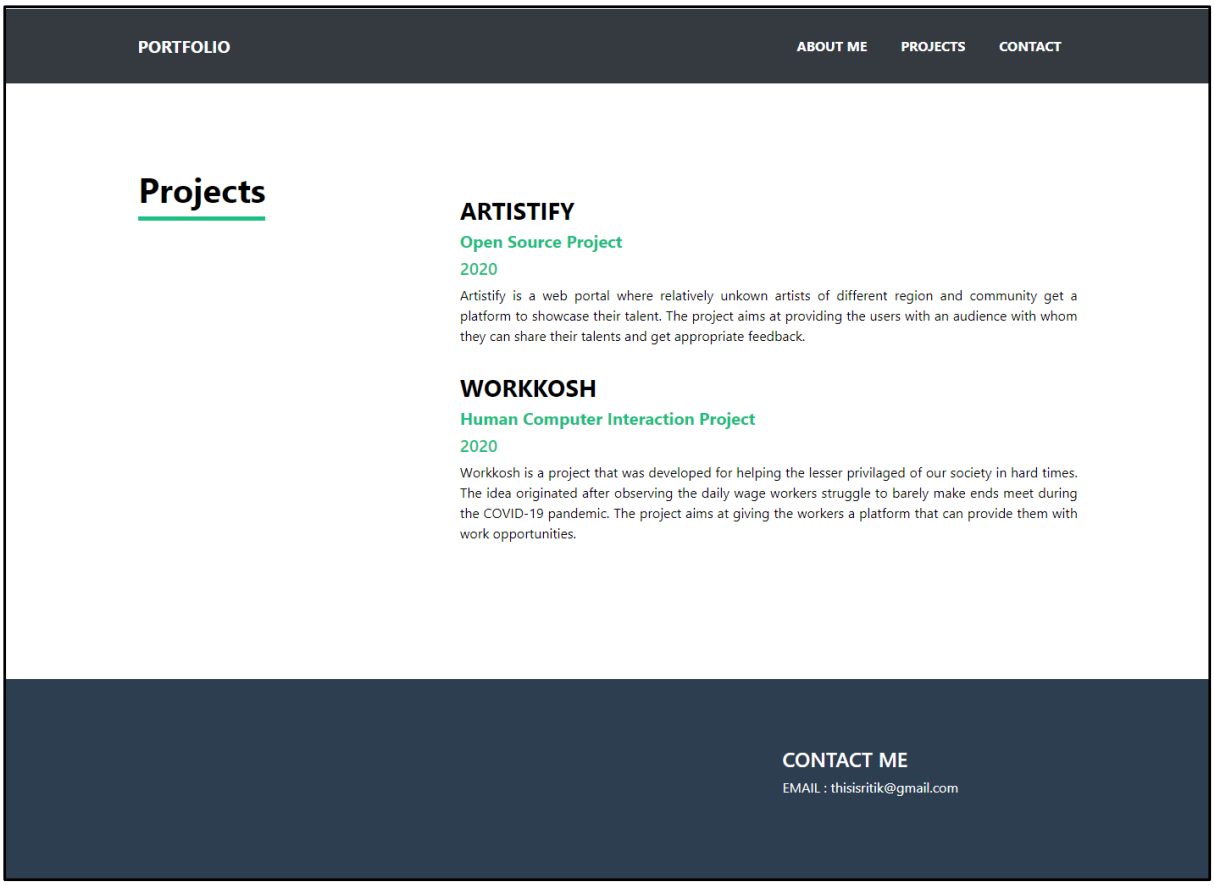
- Making Commits: Changes made to the project files are saved to the repository using commits. After a change is made to the file a commit description is given describing why the said change was made. Commit tracks the changes made to the project so that other team members can understand how and why the changes were made.



Screenshots of Portfolio:







Pros of GitHub:

- Version control – GitHub makes it easy to develop projects in versions and go to a specific version when necessary.
- Review – With GitHub's pull requests features it is very easy to keep track of the user suggestions and the UI has powerful tools like inline commenting.
- GitHub can be integrated to other tools very easily.
- GitHub is available for multiple platforms. So the project can be organized from various devices with ease.

Cons of GitHub:

- Some of the GitHub features are locked due to pricing, but the cost can go very high depending on the number of people in the team.
- The merging operation need to be done properly as the reversing process is a bit difficult to do.
- The search feature for repositories is not very good and the result can be less than average sometimes.

Features that can be added to GitHub:

- Incorporating two – factor authentication for improving user account security.
- Private repositories for the users of the free version.

Comparison with other VCS:

GIT	CVS	SVN	Mercurial
▪ Distributed repository model. ▪ Cross platform support available. ▪ Provides strong support for non-linear development.	▪ Client-server repository model. ▪ Excellent cross-platform support. ▪ It's an old tool and hence is very understandable.	▪ Client-server repository model but also permits distributive branches. ▪ Directories are versioned. ▪ Have better windows support than GIT.	▪ High performance and scalability. ▪ Advanced branching and merging capabilities. ▪ Fully distributed collaborative development. ▪ Is lightweight.