**GITHUB**

**Introduction to GIT**

Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files.

**GitHub**

GitHub brings together the world's largest community of developers to discover, share, and build better software.

**Why GitHub?**

Consider two people X and Y working on a same project and different files. X is working on first component and Y is working on second component. X push his code on to normal repository and when Y pulled that code from there, Y lost his original code which he made on second component. This is the problem of normal repository.

GitHub has a smart intelligence to merge codes done by multiple people to one code without creating any conflict for the other person’s code. But sometimes it throws an error called Merge Conflict. For example, two people A and B working on a same file parallelly and try to push the code to GitHub, it says “Don’t push there is merge conflict”. This error should be managed manually by contacting with the other person.

**Install and Download Git**

To talk to GitHub from a local machine, first you need to download the GIT in your local machine.

Link: <https://git-scm.com/downloads>

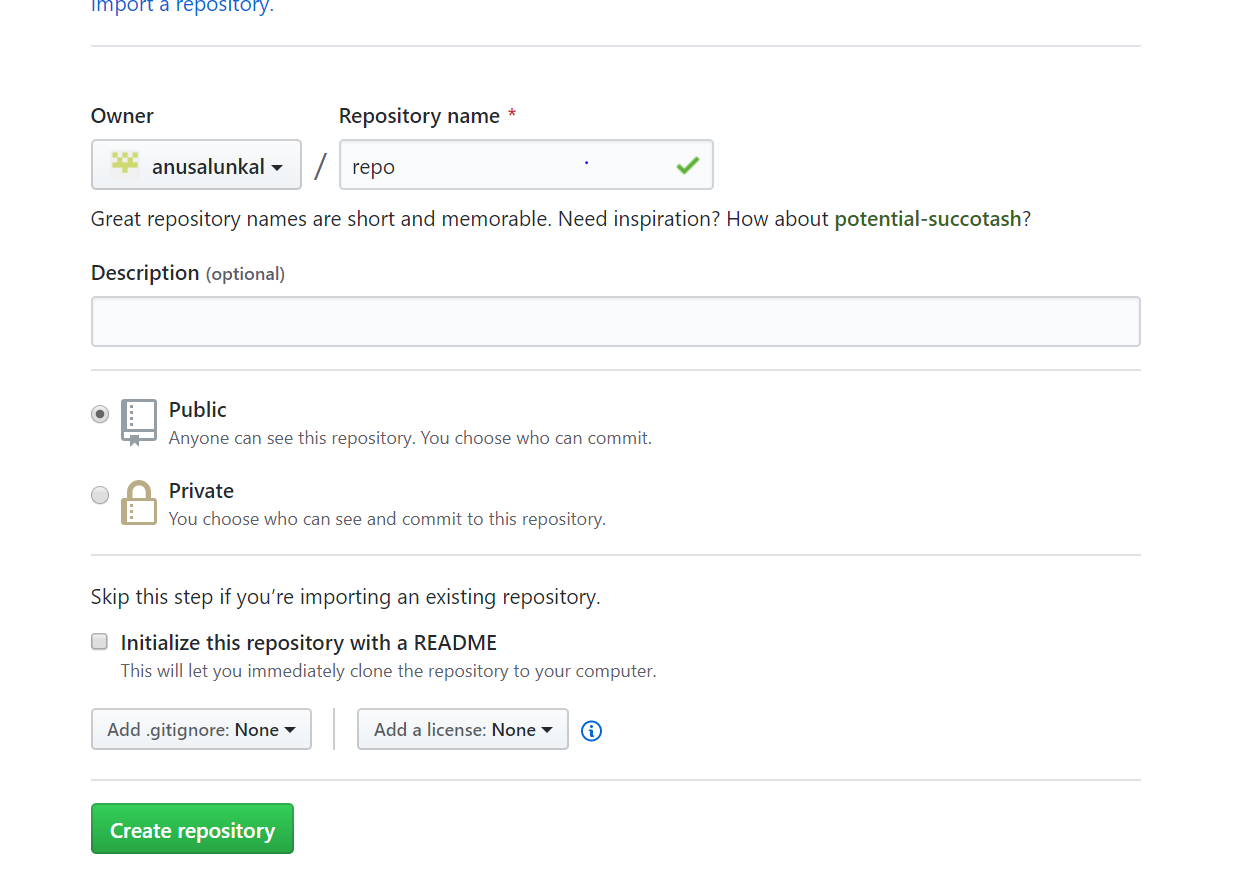
**GitHub Account Creation**

After installing Git in your local machine, create an account in GitHub to push or pull your code to your corresponding central repository.

Link: <https://github.com>

Create your own account using a unique username, email id and password.

Once your account is ready, **Create a new repository** with name and privacy settings.



**Basic Git Commands**

1. **Creating Git config**

git config -- global user.name “user1”

git config – global user.email “[user1@gmail.com](mailto:user1@gmail.com)”

1. **Push code to GitHub**

* Go to your project location in command prompt.
* Initialize a git file to read your code from local machine by GitHub.

git init

* Commit your changes
  + Staging is the first level of commit

git add \*

or

git add <filename>

* To know the files you added to staging, check the status

git status

* Commit your files added to staging.

git commit -m “First Commit Message”

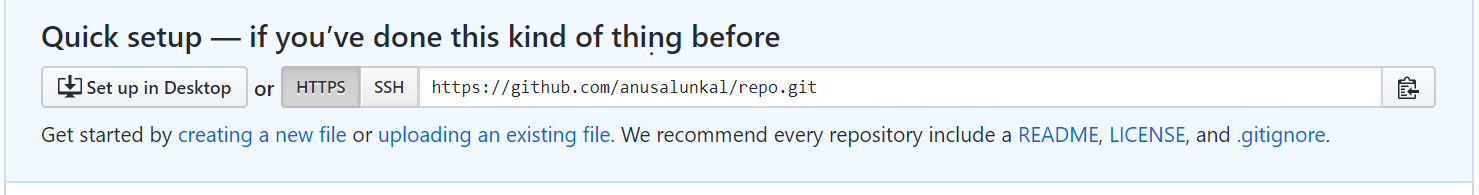
* Adding remote repository

git remote add origin <server>

origin: Name for the remote repository

remote: Remote connection

server: Address of the remote repository, i.e., URL of your repository.



* Push your code

git push origin master

master: default branch

* Undo the recent change before pushing

git fetch orgin

git reset –hard orgin/master

* Amending the most recent commit message

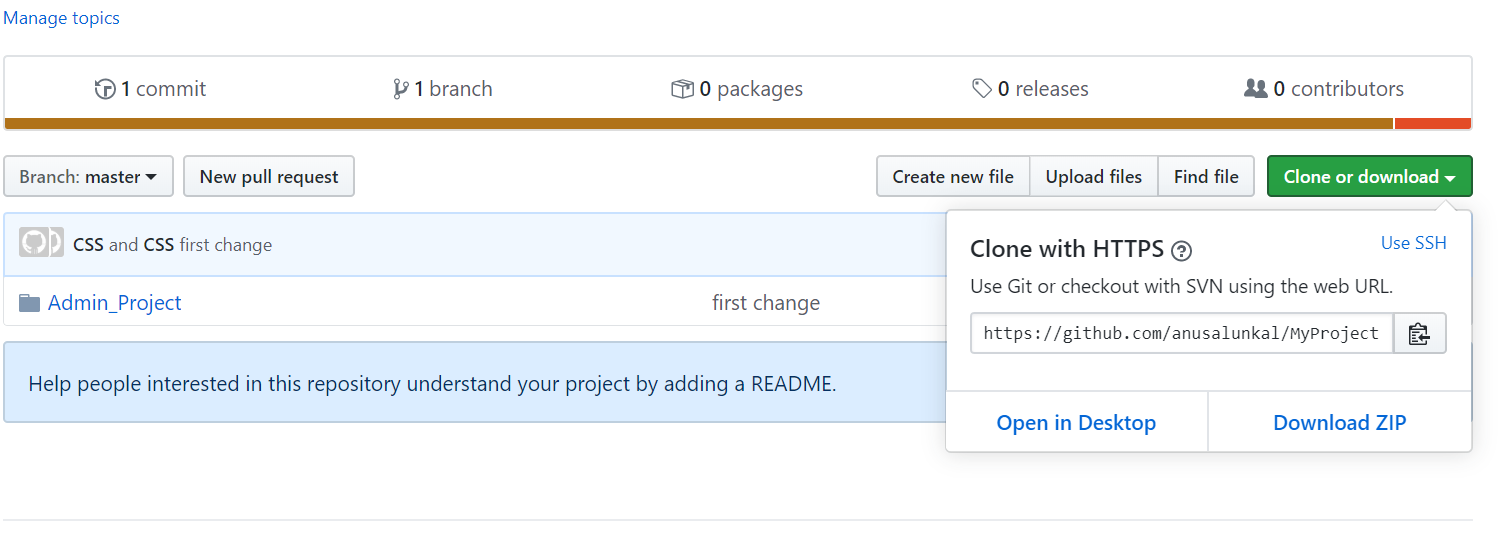
This is to modify an existing unpushed commit message.

git commit --amend -m "New commit message"

1. **Clone a fresh repository**

To download an entire fresh repository for the first time to your local repository, clone or download the code.

git clone <link>



1. **Pull the latest code**

To take the latest code to your local machine instead of downloading the entire repository, just pull only the latest changes.

git pull origin master

1. **Branching in GitHuB**

Consider a scenario where GitX is the master branch of one repository and an architect starts to change the structure of code in GitX and pushes to master branch itself, then the earlier code will not work properly due to structural changes. In this case a new sub branch needs to be created for Git, so that the changes made on sub branch will not affect the master branch unless it is merged. Once the changes is completed in sub branch it can be replaced as master branch.

* To create a new branch

git checkout -b <branchName>

* To know the current branch

git branch

* Push to branch

git push origin <branchName>

* Pull from branch

git pull origin <branchName>

* Switch to another branch

git checkout <branchName>

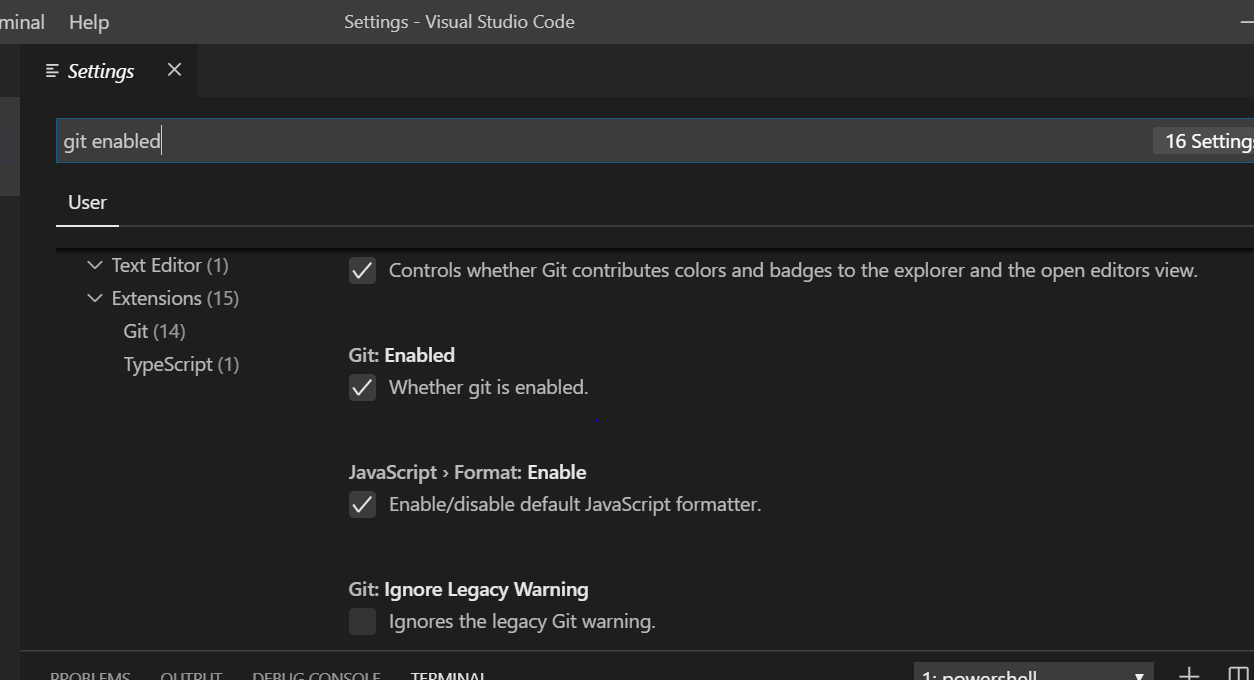
* To merge the branch to master
  + Checkout to master branch
  + Merge the branch

git merge <branchName>

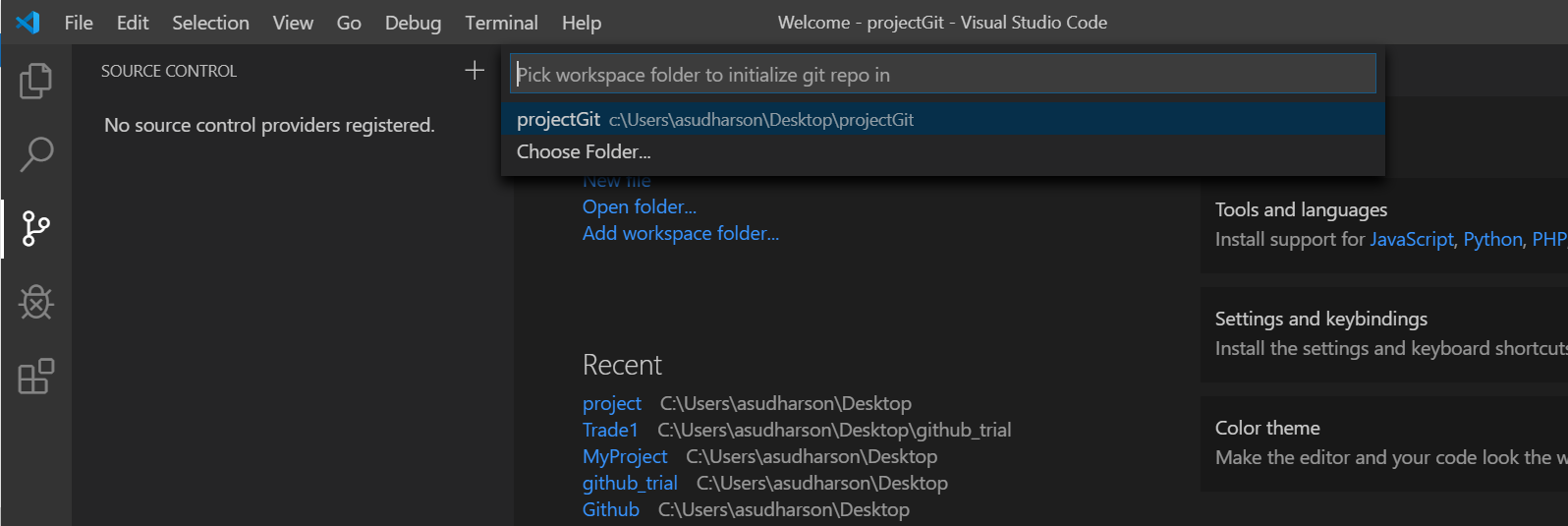
**Using Git with Visual Studio Code**

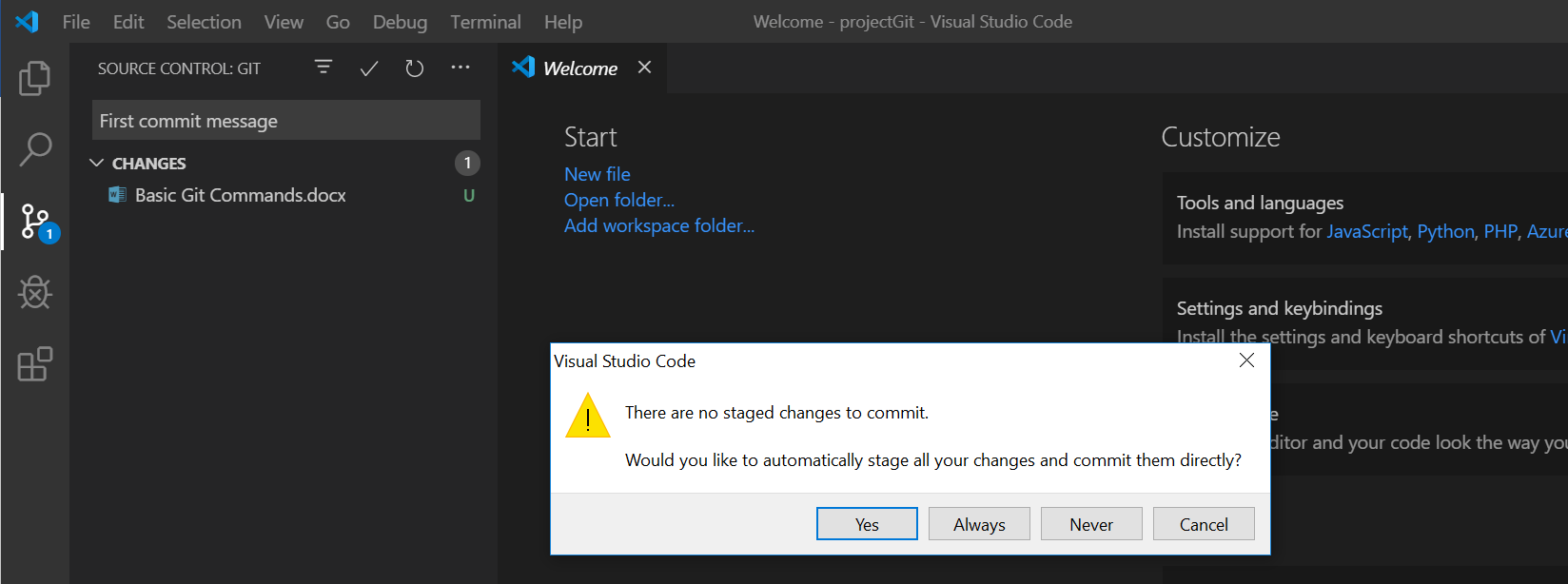
1. **Add to git the existing project** 
   1. Install Git on machine
   2. Create an account on GitHub
   3. Create a new repository and copy URL
   4. Go to visual studio and open project folder
   5. Check git is enabled in vs code

Path: settings -> search git enabled -> mark checkbox



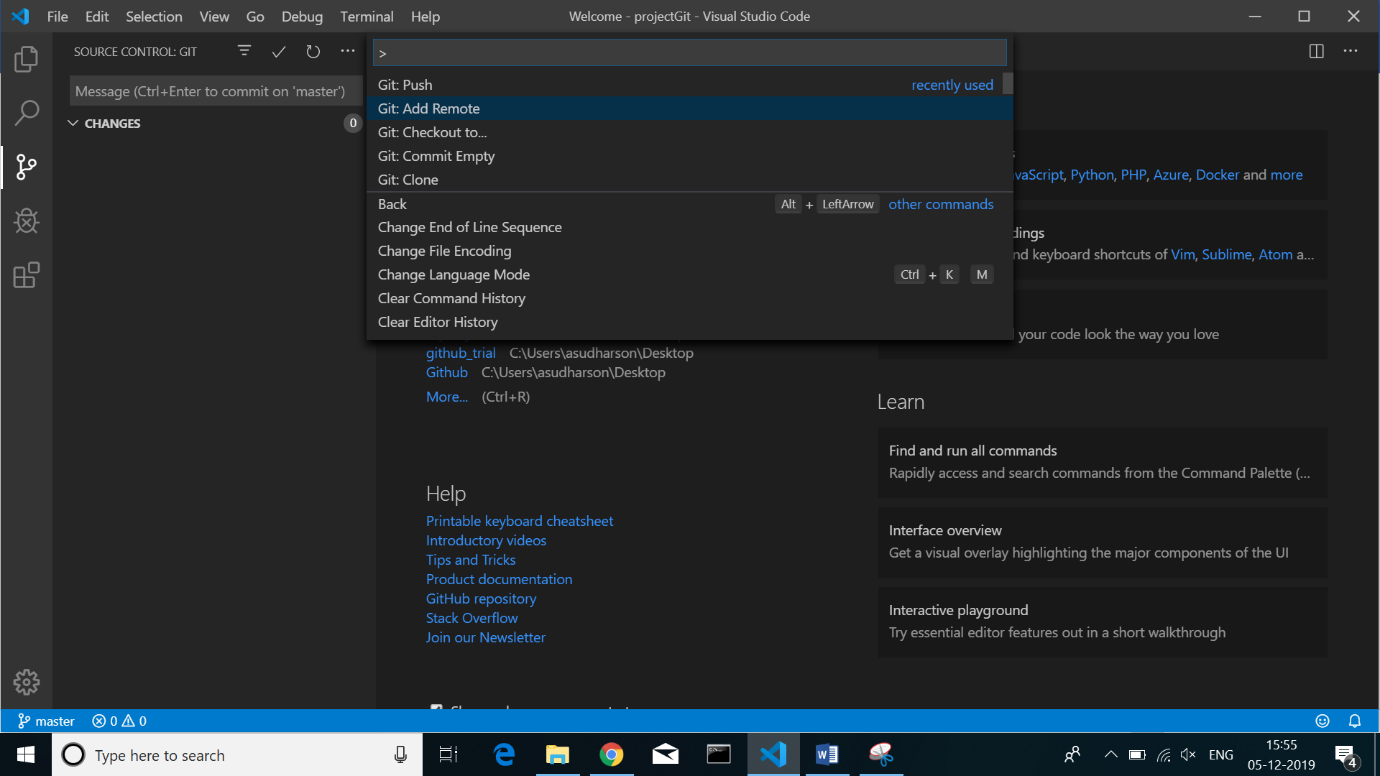
* 1. Go to source control
  2. Choose folder of project, write the commit message on text box and click tick mark on commit changes

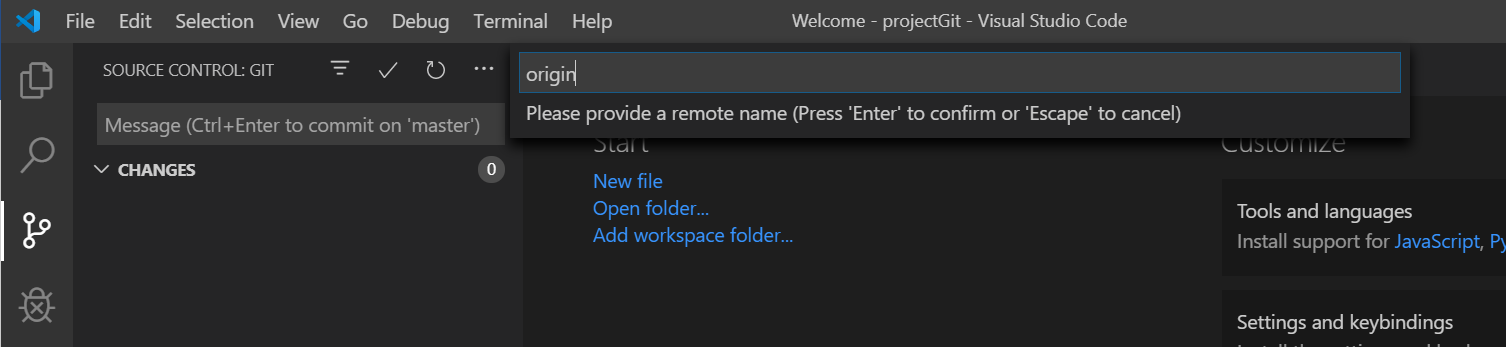


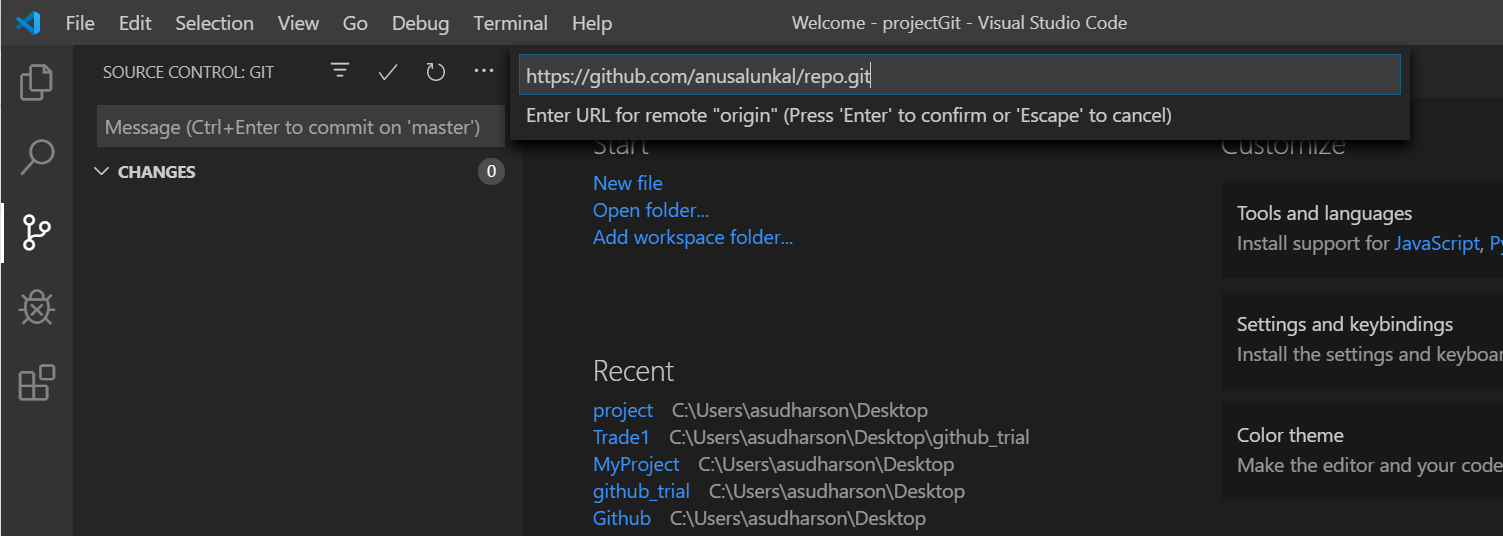


* 1. Add remote repository

Path: view -> command palette ->enter add remote ->give name -> enter ->give URL

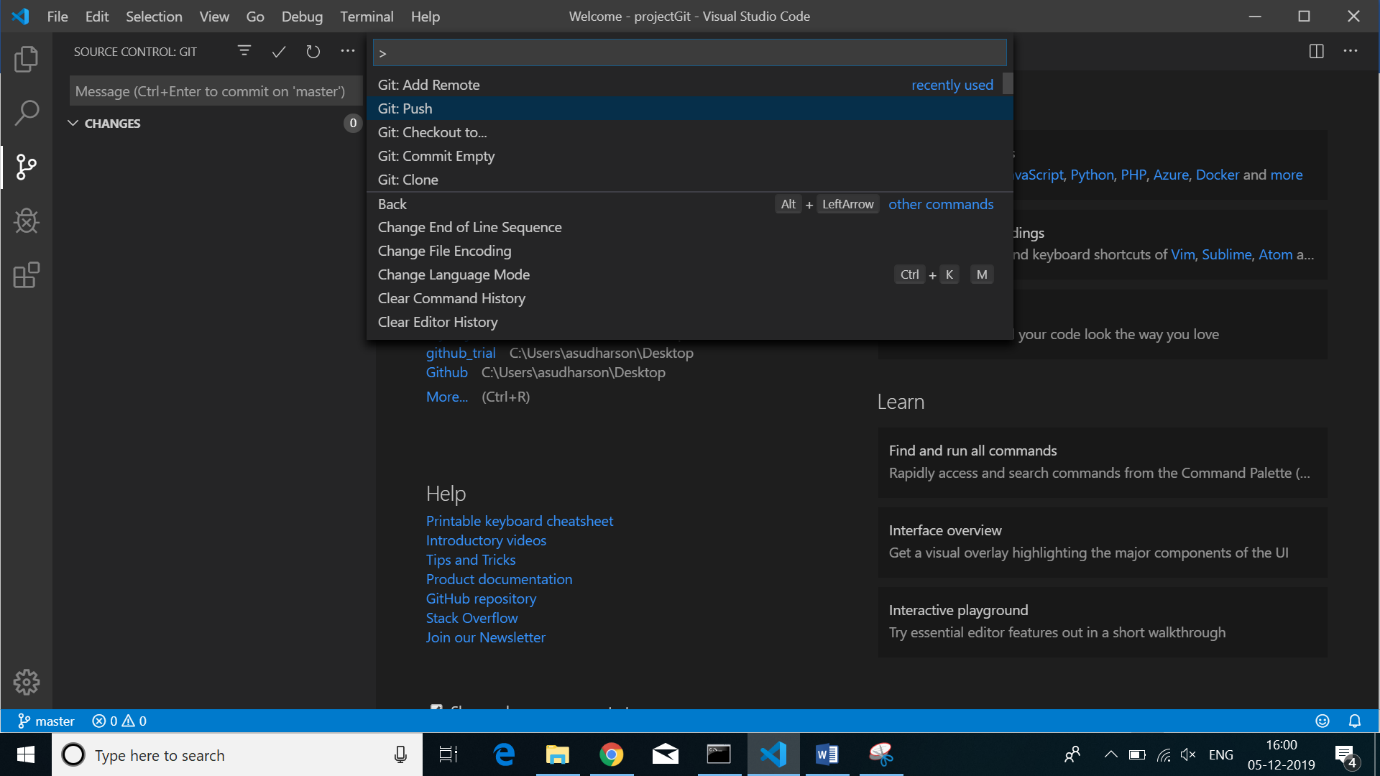


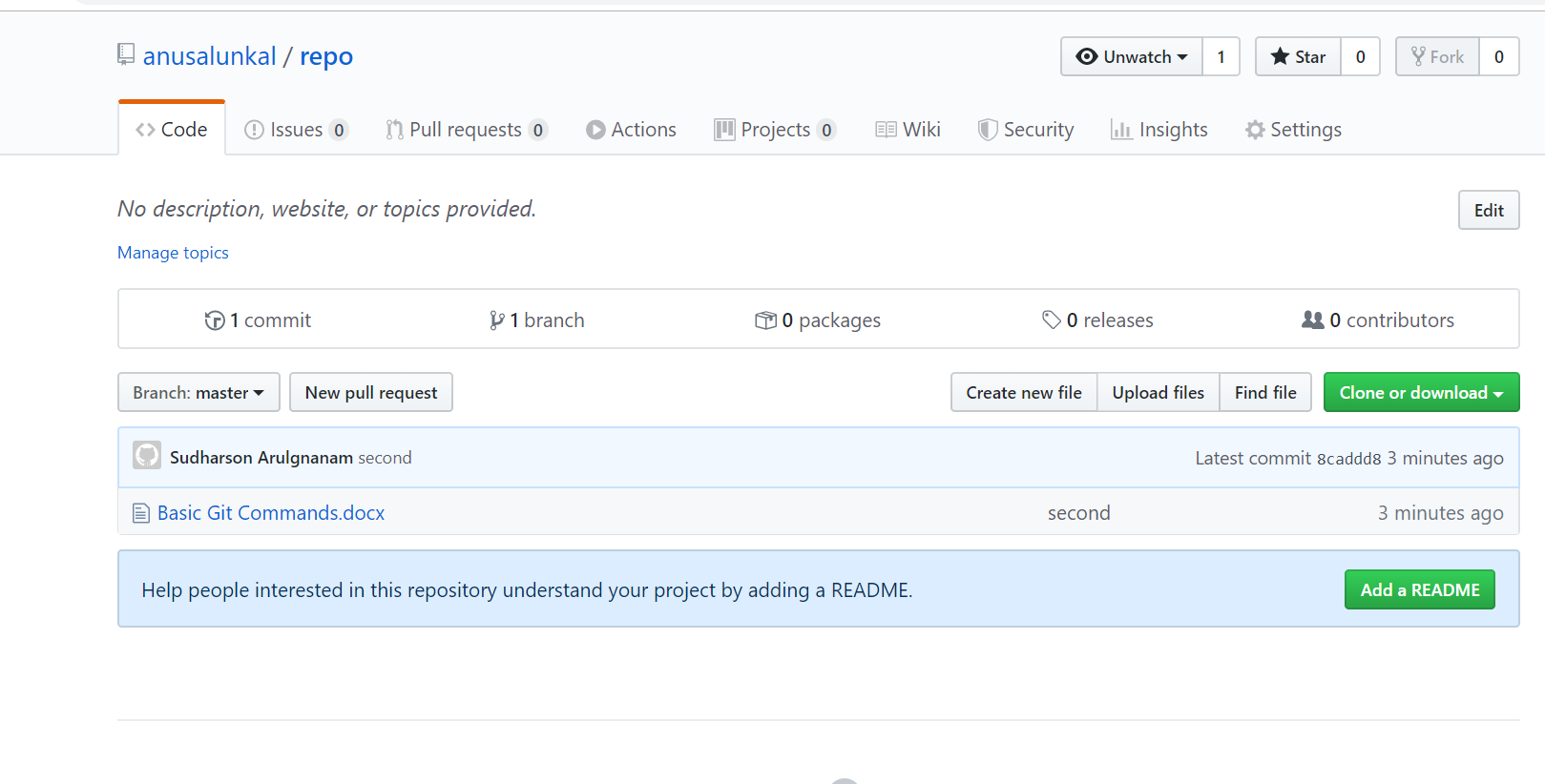




* 1. Push the committed changes

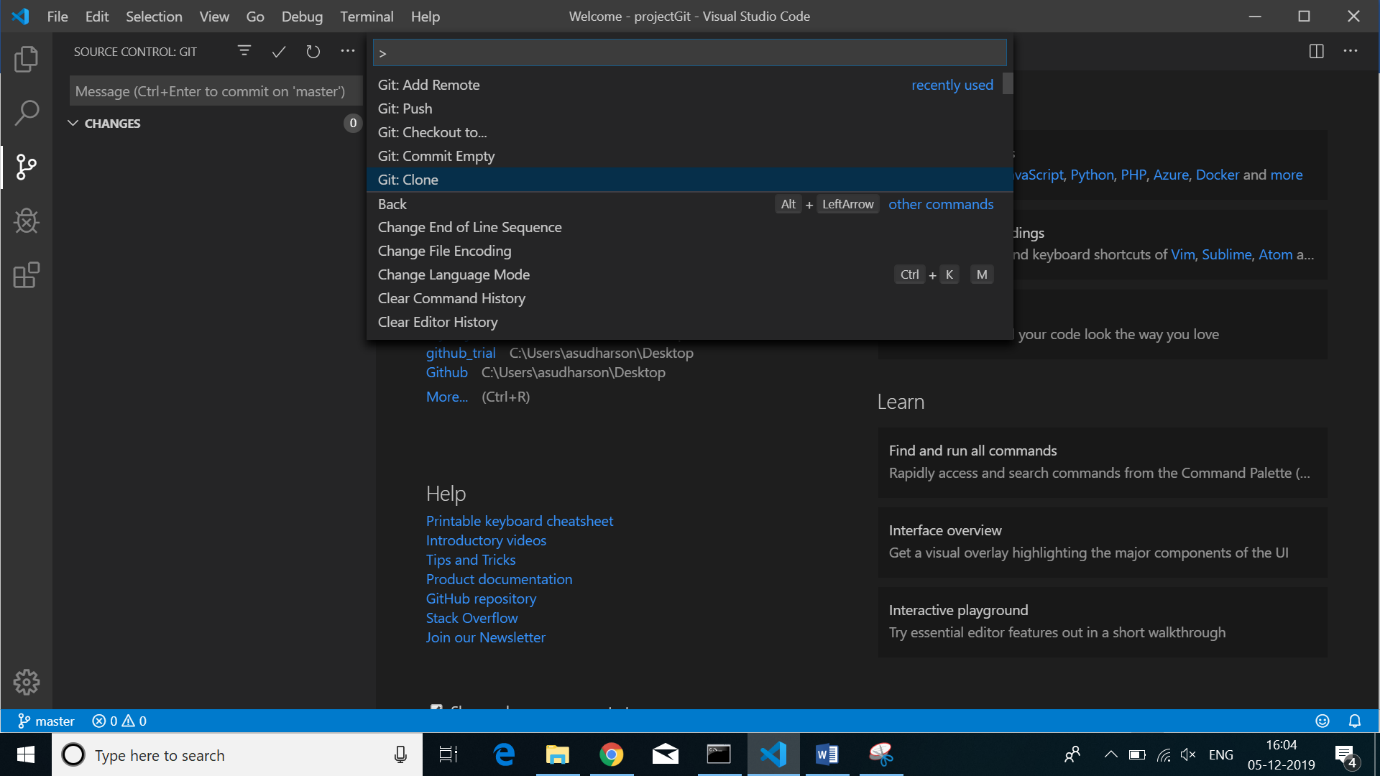
Path: view -> command palette ->enter git push

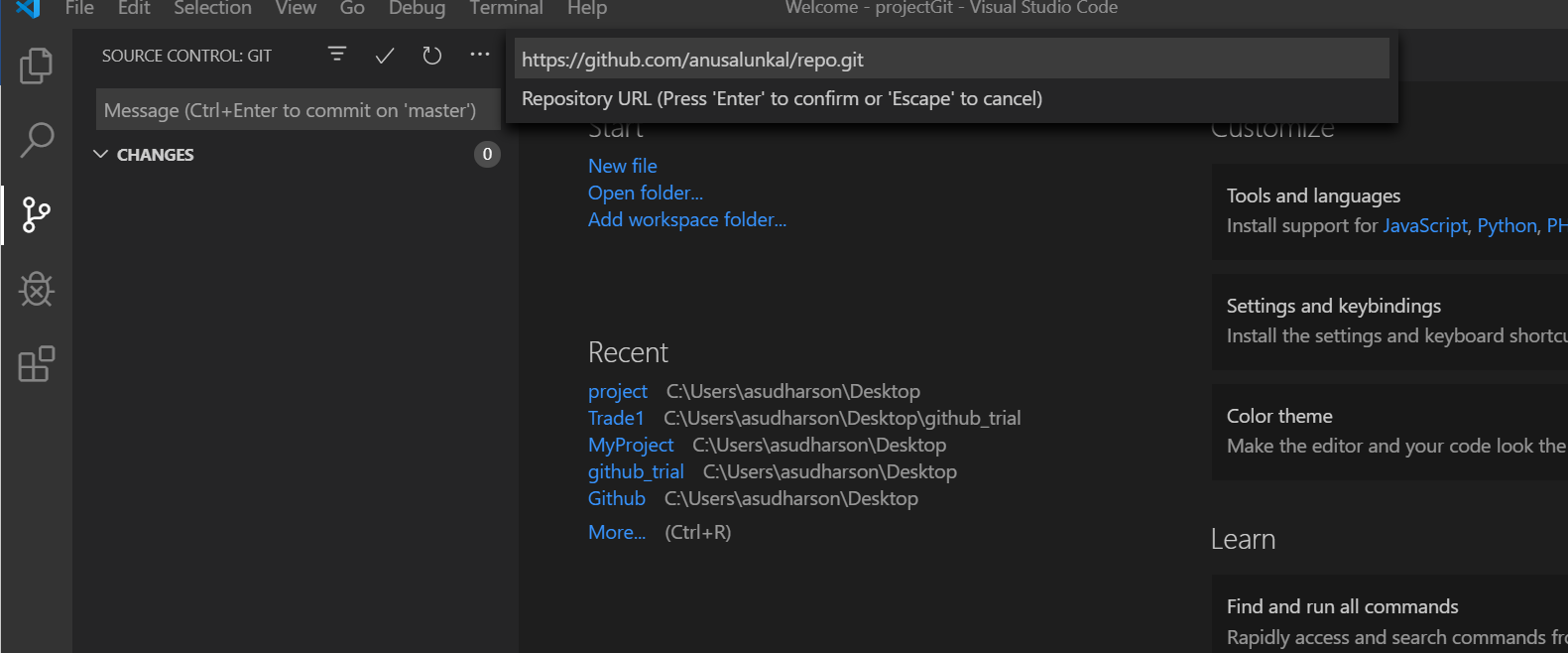


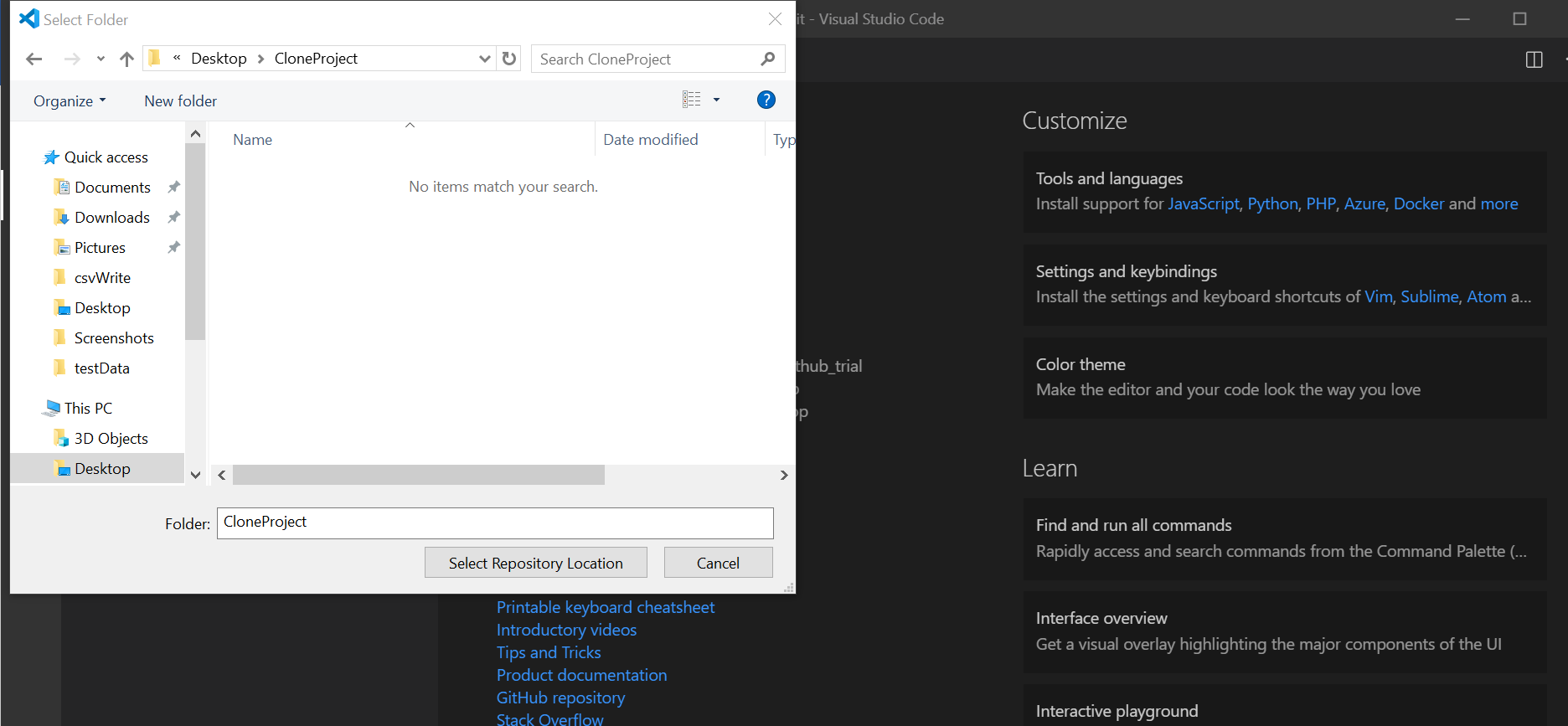


1. **Clone from GitHub**

Path: view -> command palette ->enter git clone ->give URL -> select local repository location -> click ‘Select Repository Location’

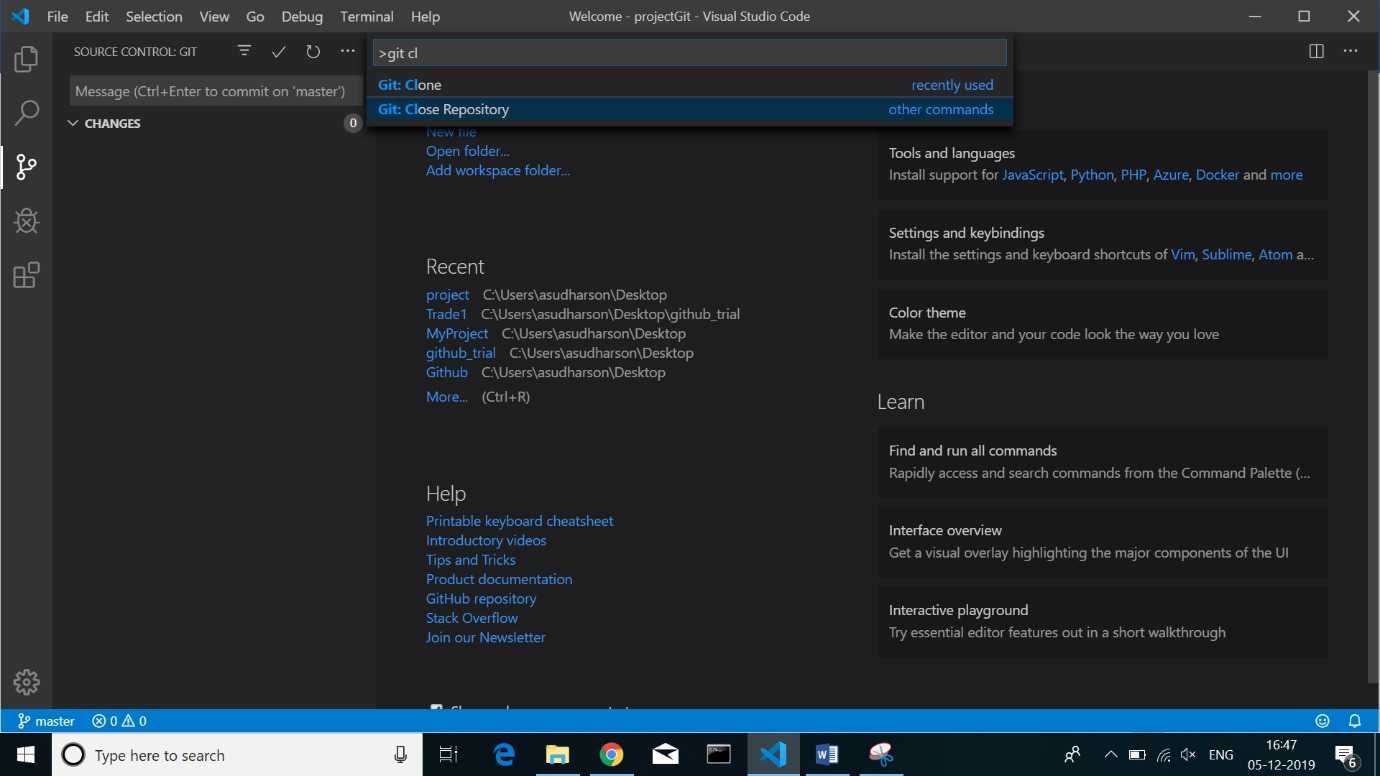


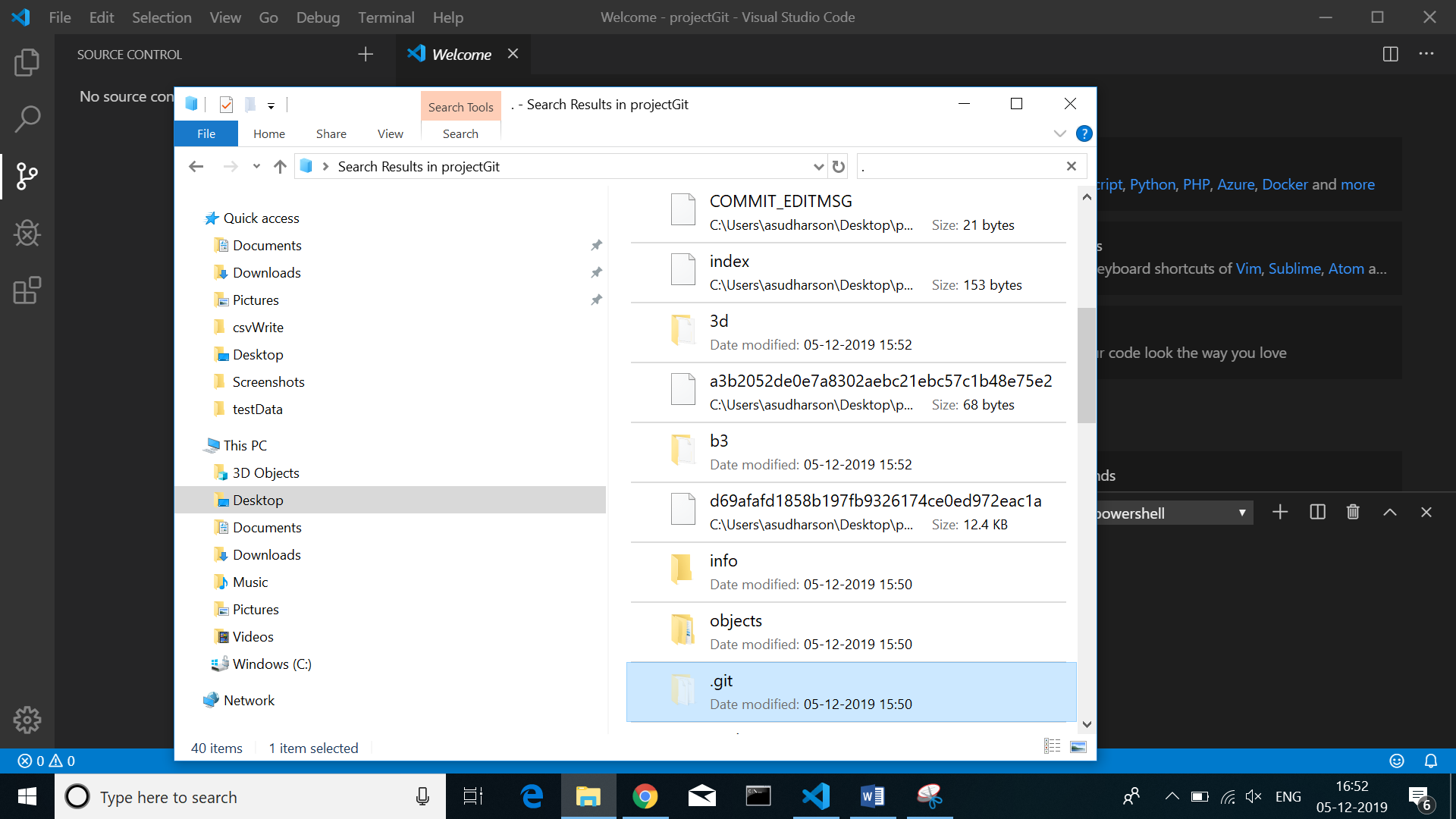




1. **Remove git mapping**

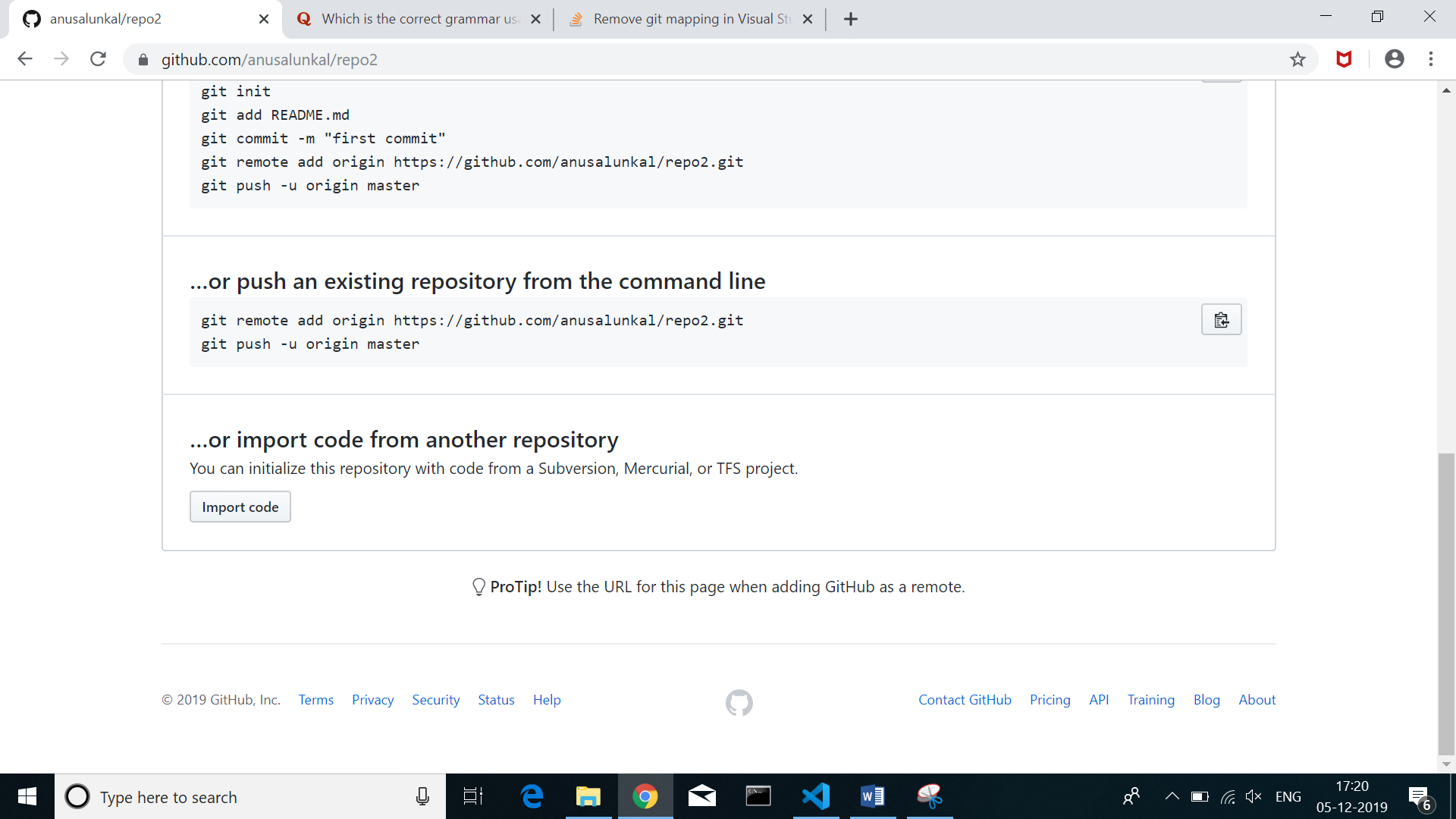
Path: view -> command palette ->enter git close repository -> delete .git hidden file from your project manually



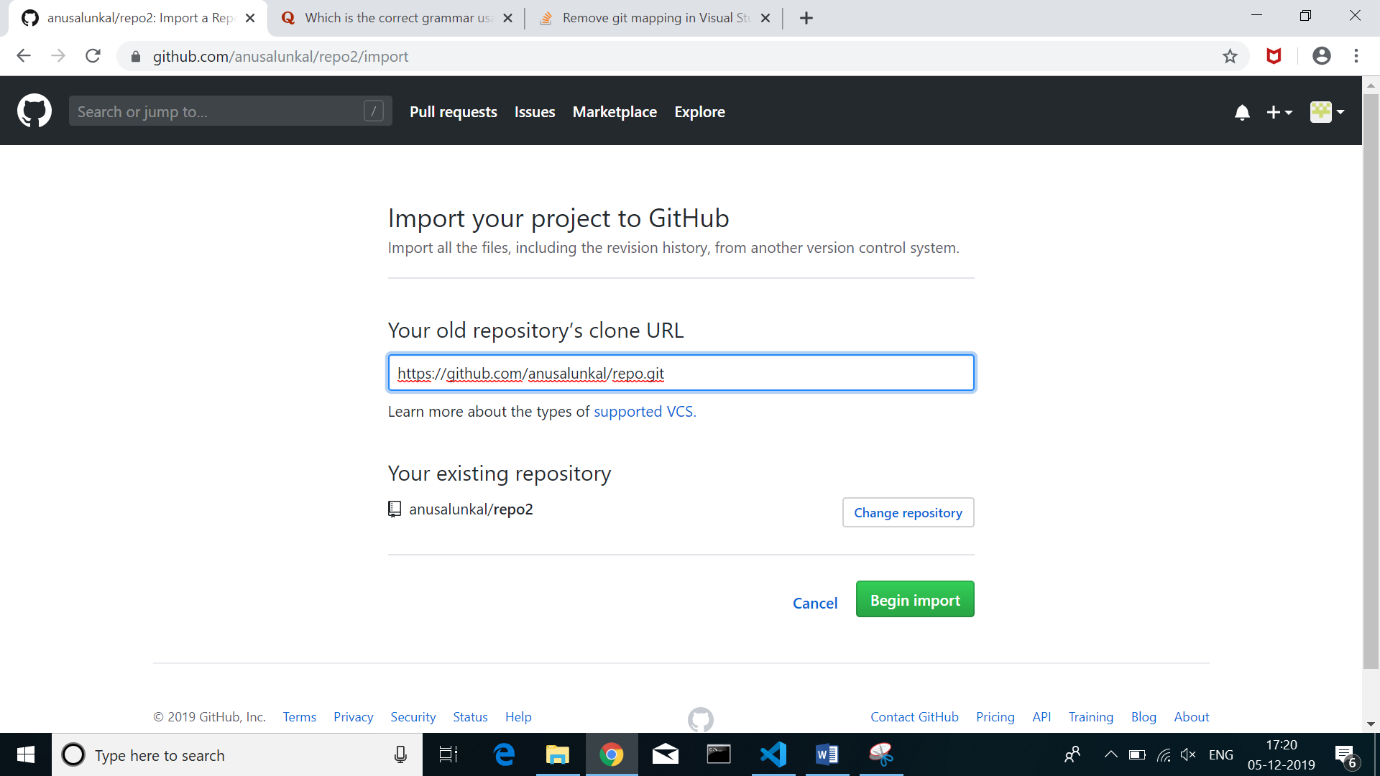


**Importing a repository directly using GitHub UI**

1. Create a new repository
2. Click on import code button

****

1. Paste the URL of the repository which you want to import



1. Click Begin import button

