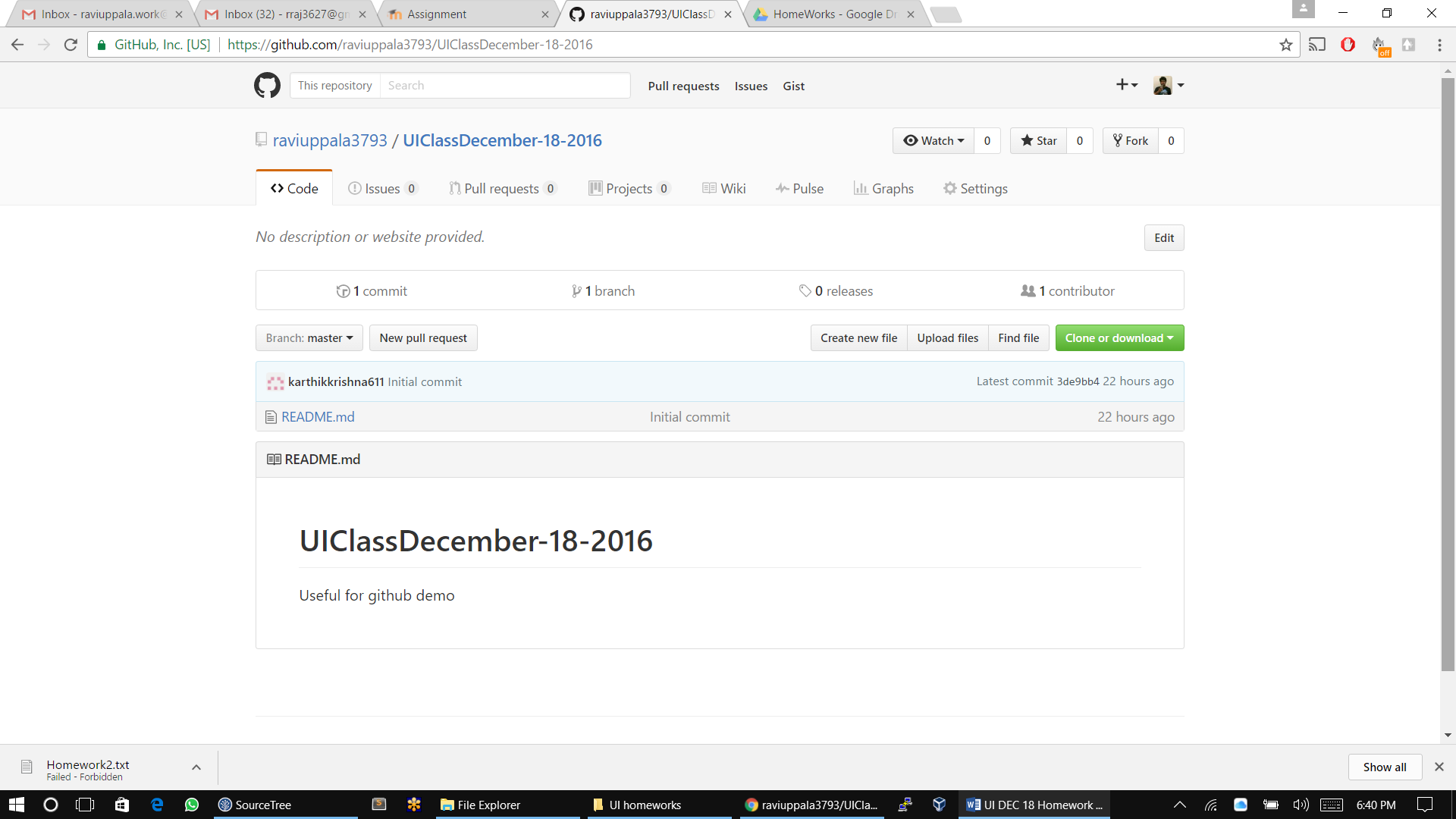
**Home work 2**

Look into all the keywords below

1. **try to clone my repo**

[**https://github.com/karthikkrishna611/UIClassDecember-18-2016**](https://github.com/karthikkrishna611/UIClassDecember-18-2016)



1. **attempt to create a own repo for submitting homwork and list down all the steps in creating repo in github.com**

In the top left corner of the github account, by clicking on the add symbol a list will drop down and click on new repository option.

After, a new page will be opened in the page type the repository name and click on the public option and check the initialize the repository option

And click on create repository.

1. **research about all git hub commands and write an explanation for all the following commands in details and when to use them**

**git hub cheat sheet link: https://www.git-tower.com/blog/git-cheat-sheet/**

**Clone**: This command is used to copy the existing git repository

**Init**: this command is used to make directory as the git repository.

**Add**: this command is to add the unstaged files to staged files in the command line then commit the changes.

**MV:**  this command is used to rename any file or to rename a file in the local repository.

**Reset:** This command is used to make the current branch and reset to somewhere. There are four types reset soft, hard, mixed, merge.

**Rm**: this command is used to remove the files from the repository and not from the file system.

**Bisect:** This command is to find the problem creating commit. Suppose we have series of commits, in that a commits one commit creates a bug in the code we can use bisect command.

**Grep:** This command is used to find a specific folder or file and a specific pattern files.

**Log:** this command is used to view commit history when we made series od commits

**Show:** This command is used to show the log messages, tag messages and trees.

**Status:** This command is used to see staus of the repository wheather the source code files is statged are unstaged.

**Branch:** command will list all the local branches in the repository.

**Checkout:** This command is used to switch to a specific branch. For suppose if A , B and C are the branches if we want to switch B branch this command is used.

**Commit:** this command is used to save the changes made in the index with a commit message.

**Diff:** this shows the changes in the working tree and index and changes in the two files

**Merge:** This command is used to merge the two commits.

**Rebase:** This command is basically to move from one commit to another commit.This command is a process of moving a branch to initial commit

**Tag:** This is to add tags to commit ID’s.

**Fetch:** this command is used to import the remote repository to local repository and the next commits will be saved as remote branchs.

**Pull:** This command is used to get the code from remote repository when a two persons are working the code.

**Push:** used to sent the working code to the remote repository.

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

reset Reset current HEAD to the specified state

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

checkout Switch branches or restore working tree files

commit Record changes to the repository

diff Show changes between commits, commit and working tree, etc

merge Join two or more development histories together

rebase Forward-port local commits to the updated upstream head

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects