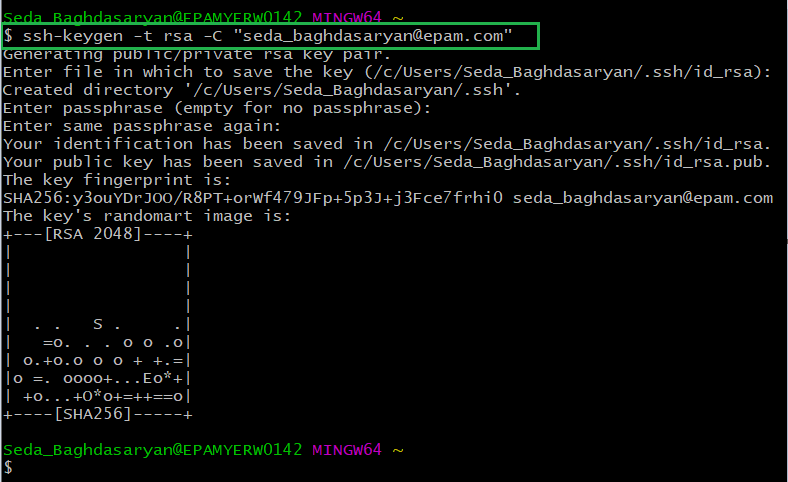
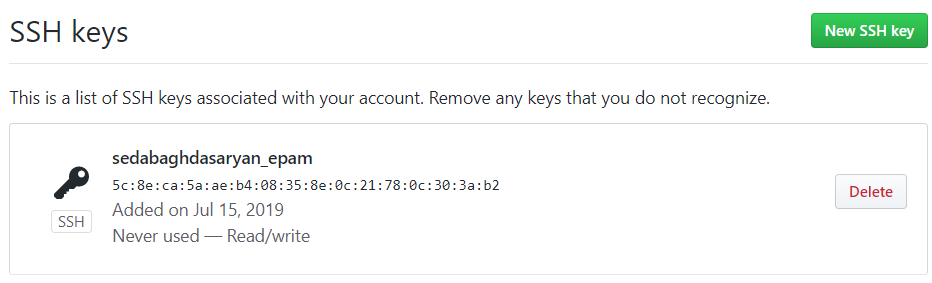
|  |  |
| --- | --- |
| D:\PAL\Logo_4_21_15\Logo_4_21_15\Primary Logo\png_files\EPAM_LOGO_Full_Color_RGB.png | L&D Automated Testing Mentoring Program |

Module 2: version control with git

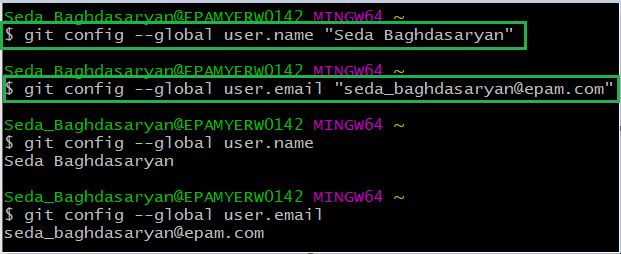
## home task

1. **“I CAN WIN” Task**
   1. Install git and generate a pair ssh keys. Authorize public key in github.com.

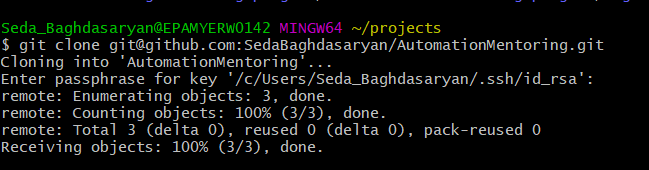
****

****

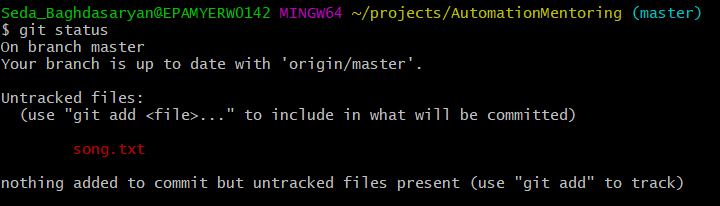
* 1. Set your user.name and user.email in git.

****

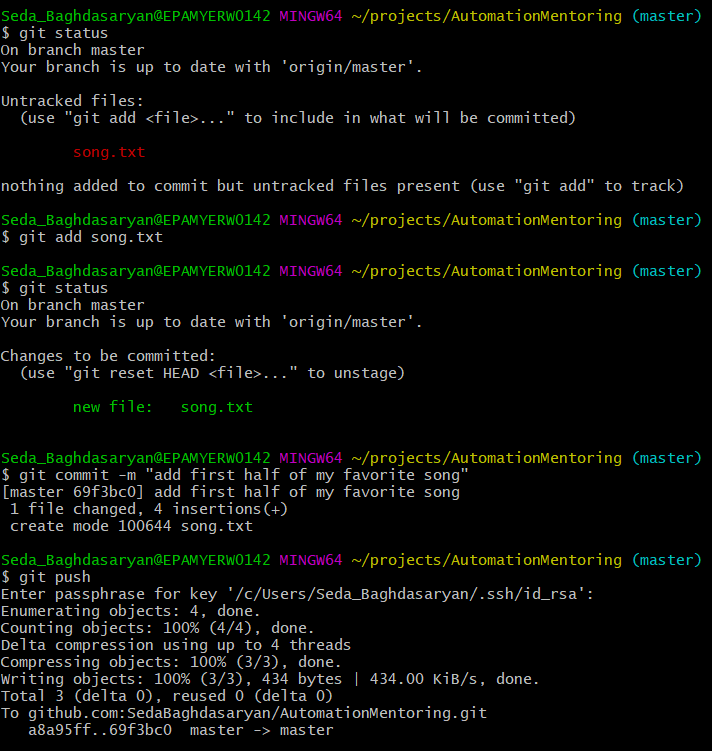
* 1. Create a new repository in github.com and clone it locally to your machine.

****

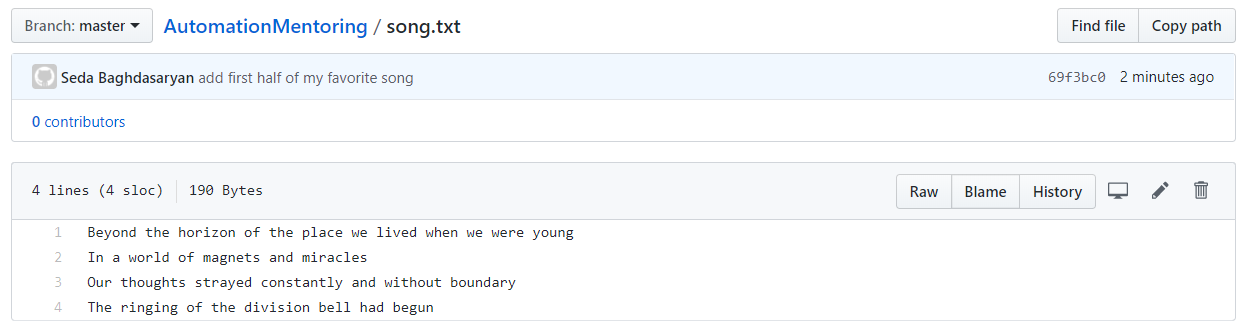
* 1. Create a file with name “song.txt” and put a half of the text of your favorite song to it.

****

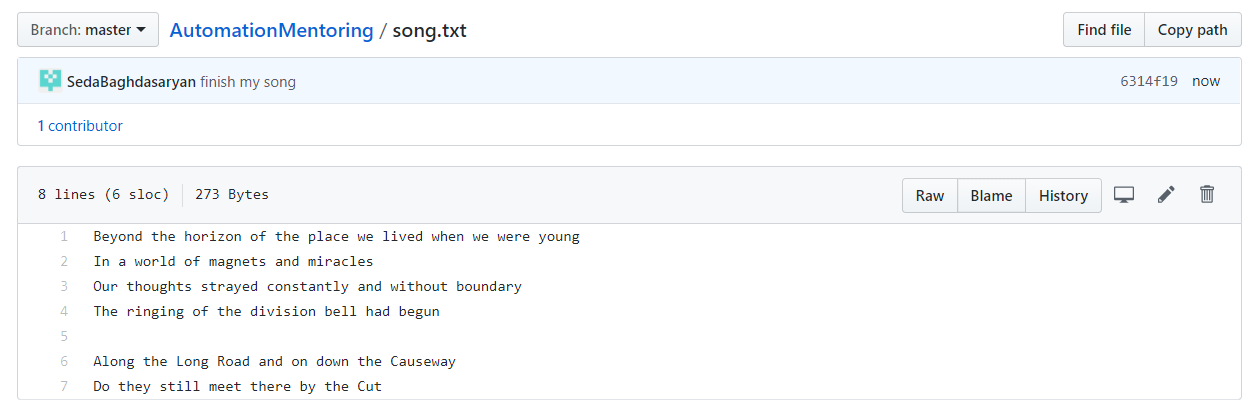
* 1. Make commit with name "add first half of my favorite song" and send it to the server.

****

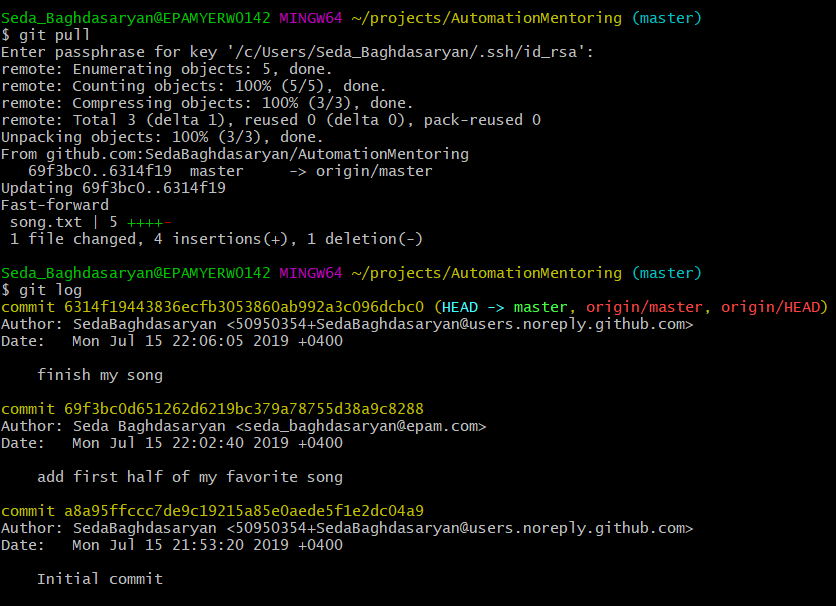
* 1. Make sure there is the file “song.txt” in github now and it contains a half of the song text.

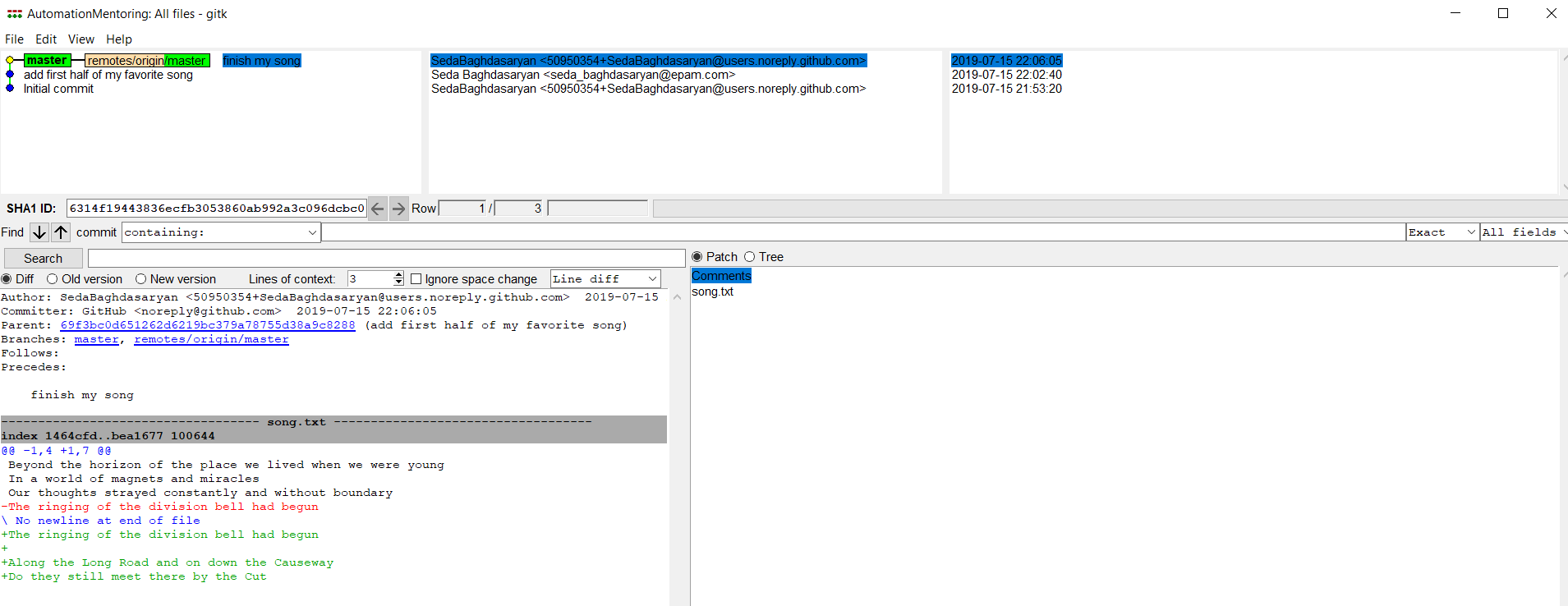
****

* 1. Using github web-interface add the second half of song text and make commit with name "finish my song".

****

* 1. In your local repository pull the changes and make sure that the changes made with the second commit are available, and you are able to see full text of the song.

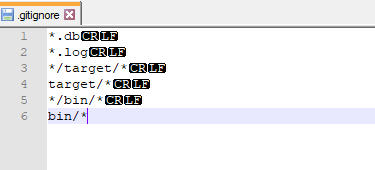
****

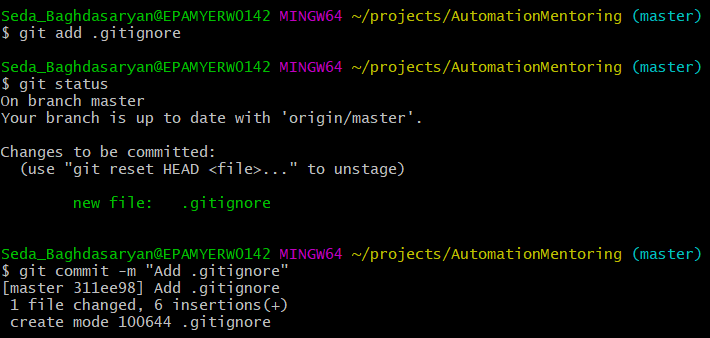
****

1. **“I BRING IT ON” Task**

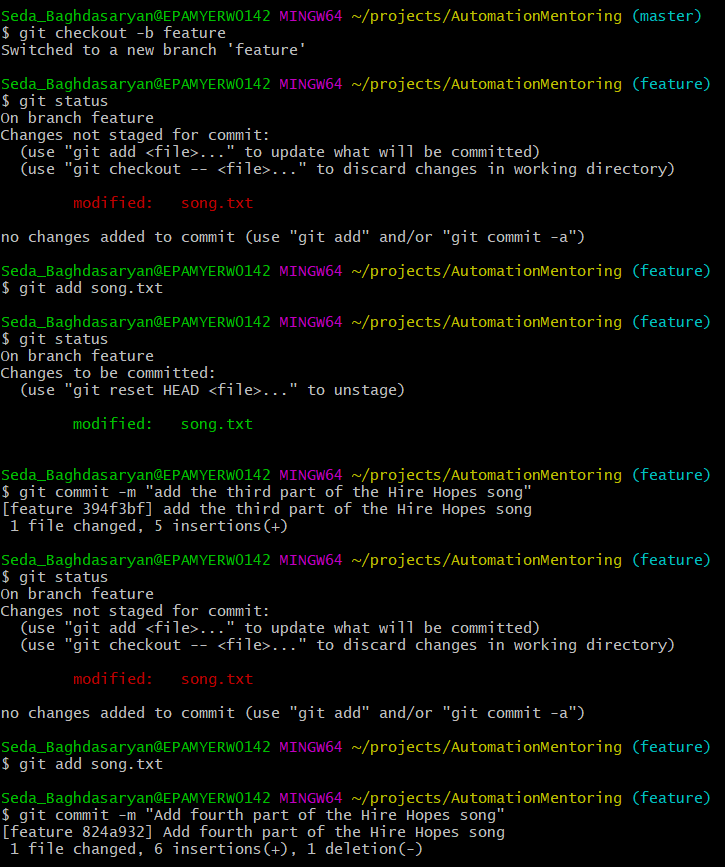
This task is based on the previous one (“I Can Win”)

* 1. Add to your project file .gitignore and set this up to hide files with extensions .db, .log, etc. and folders with names ‘target’ , ‘bin’ etc.

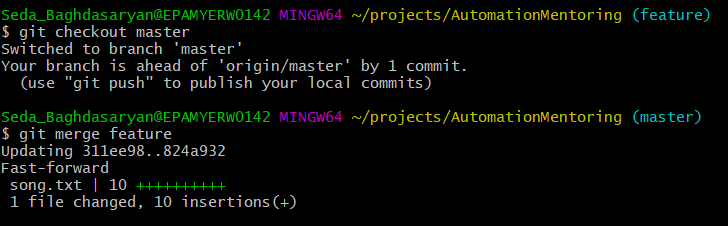




* 1. Create a branch ‘feature’ and make two commits to it.



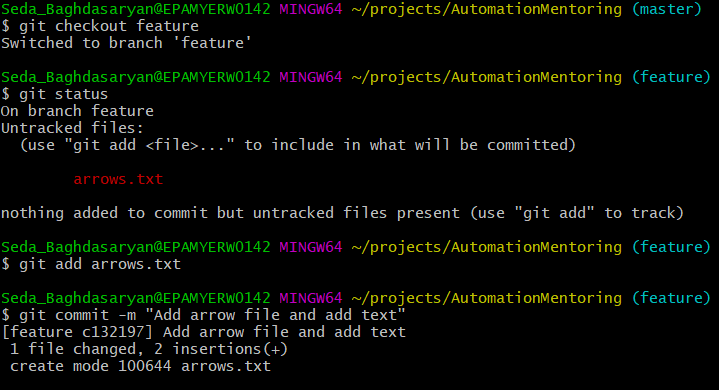
* 1. Merge ‘feature’ branch to ‘master’.



* 1. Navigate back to ‘feature’ branch and create a file ‘arrows.txt’ with the content as below:

*The ship glides gently on the waves  
As day turns into night*

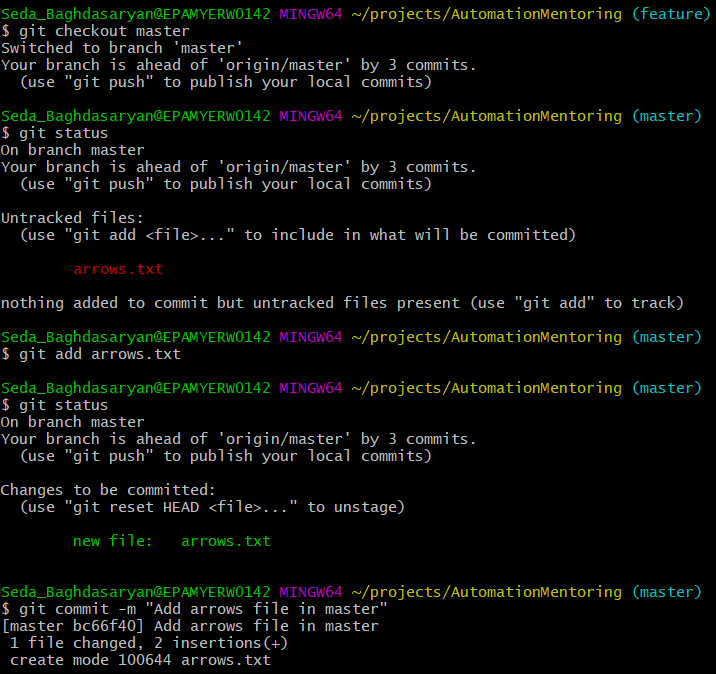
Make commit.



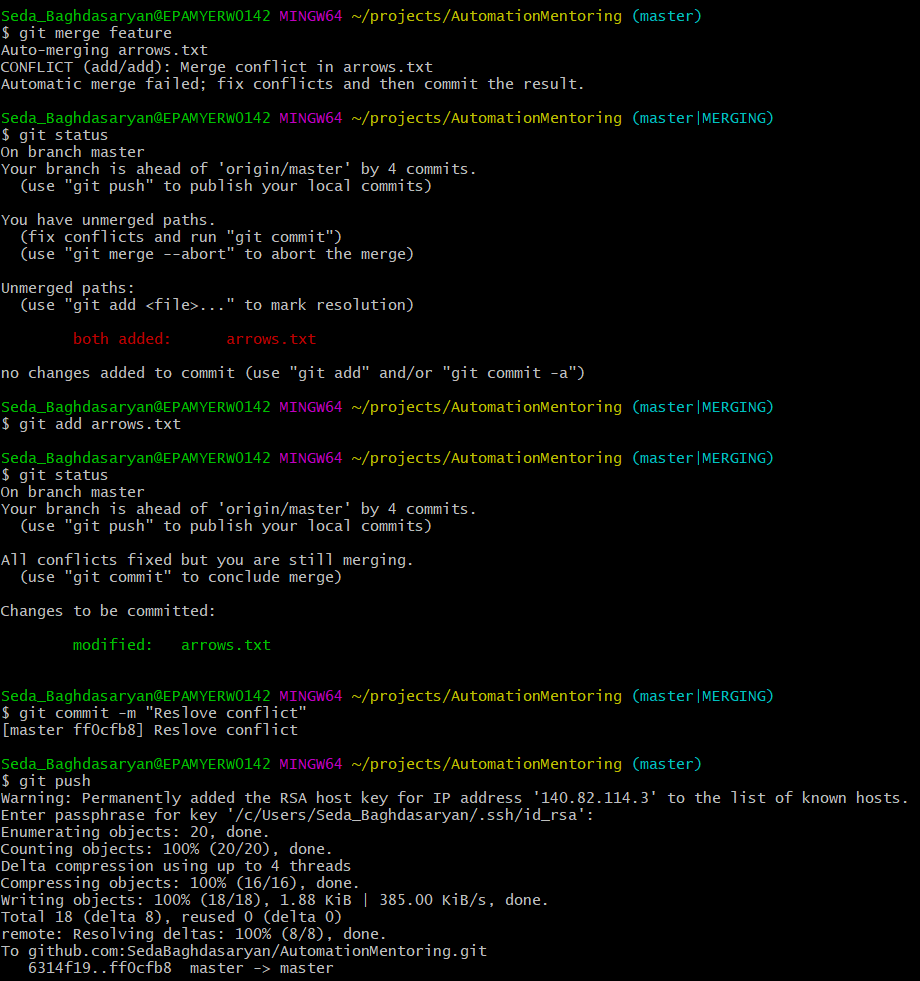
* 1. Switch to ‘master’ branch. Create there a file ‘arrows.txt’ with content as below:

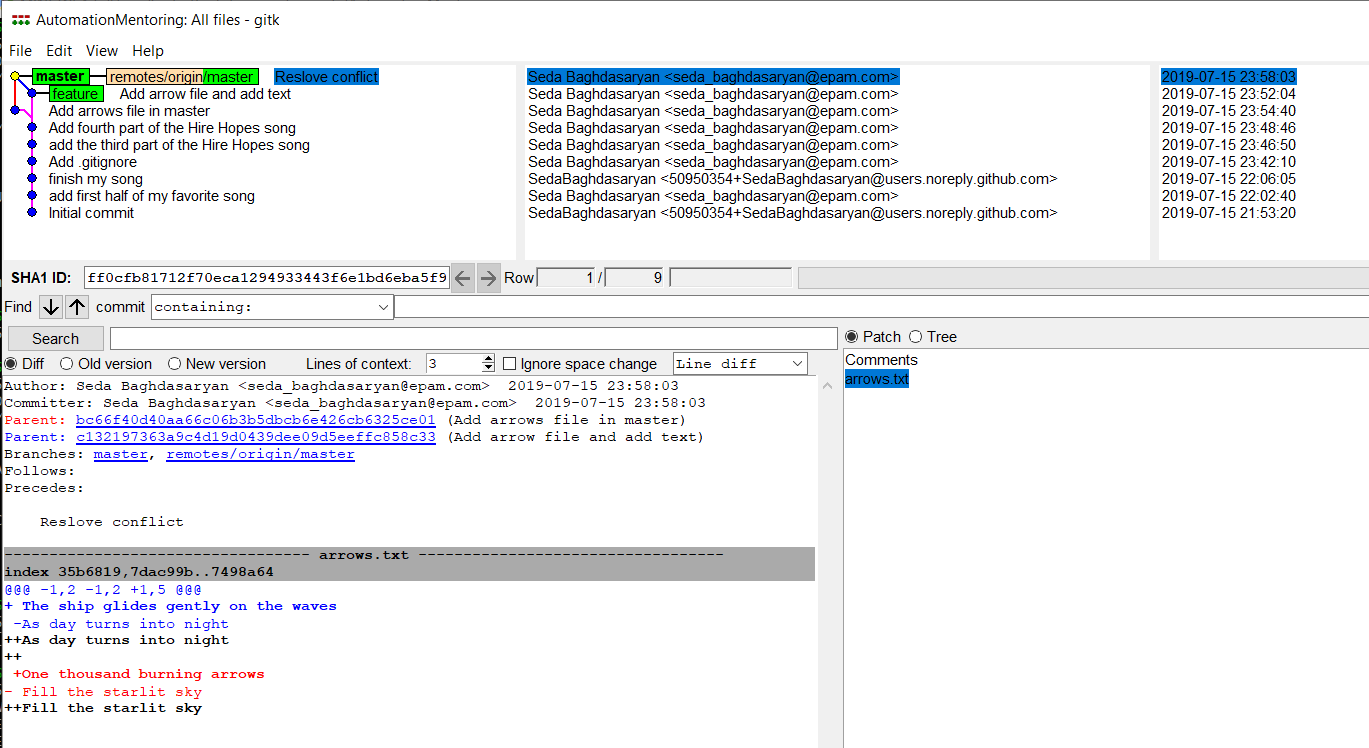
*One thousand burning arrows  
Fill the starlit sky*

Make commit.



* 1. Merge ‘feature’ to ‘master’ resolving conflict: save all 4 strings in file ‘arrows.txt’ in the same order how they were added in steps 2.4 and 2.5.



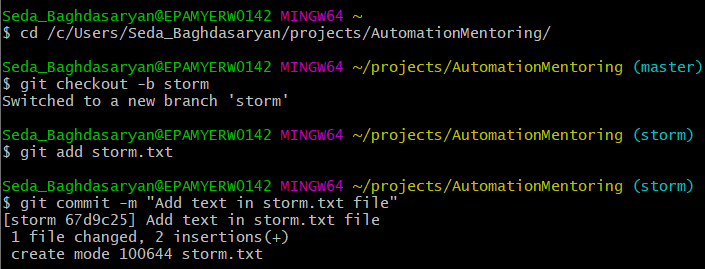


1. **“HURT ME PLENTY” Task**

This task is based on the previous one (“Bring It On”).

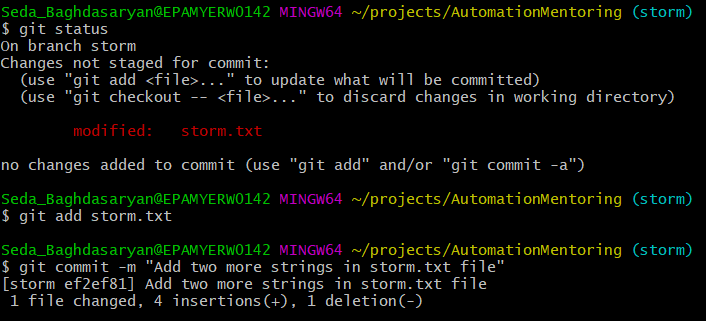
* 1. Create a branch ‘storm’ and add commit to file ‘storm.txt’:

*Twenty ships with Norsemen braves  
Riding the northern wind*

**

* 1. Add 2 more strings in ‘storm.txt’ and make one more commit:

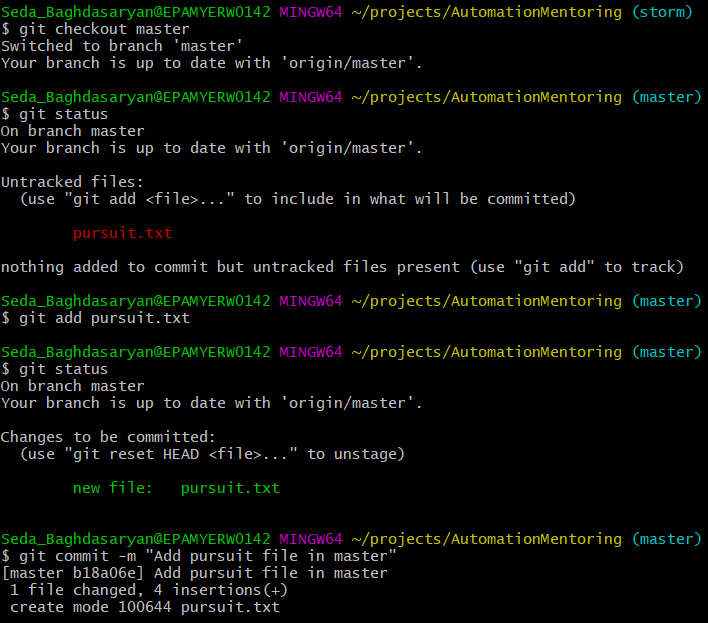
*They left their shores at early dawn  
As a red sun was rising in the east*

**

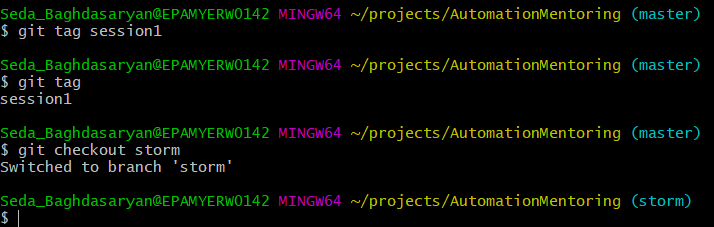
* 1. Switch to ‘master’ and create a file ‘pursuit.txt’ with content as below:

*The warming sun returns again  
And melts away the snow  
The sea is freed from icy chains  
Winter is letting go*

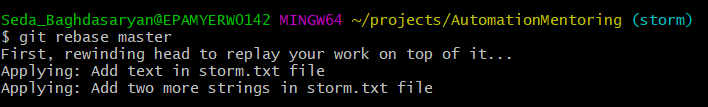
Make commit.

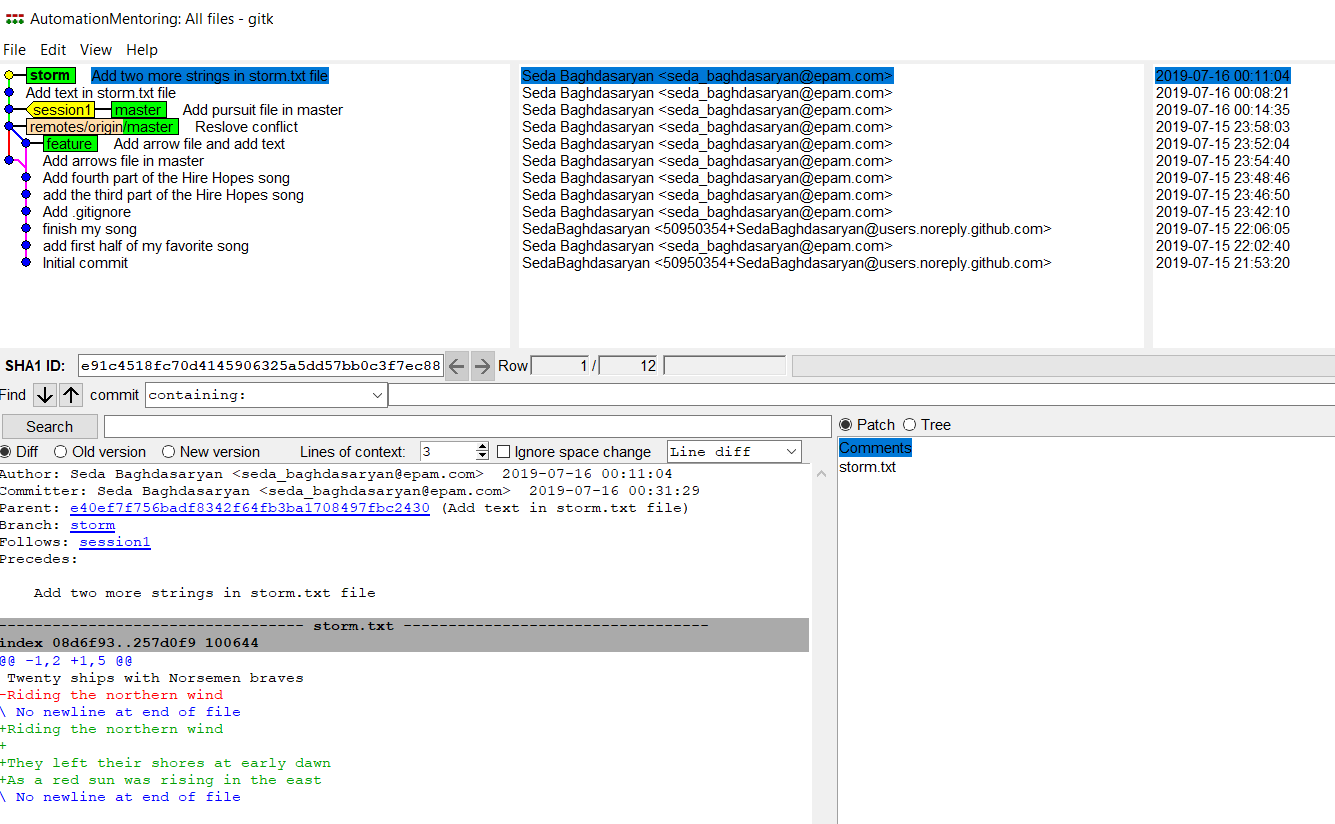


* 1. Mark the commit with tag ‘session1’ and switch to branch ‘storm’



* 1. Make rebase of ‘storm’ branch in the way it contains the last commit from ‘master’ branch.

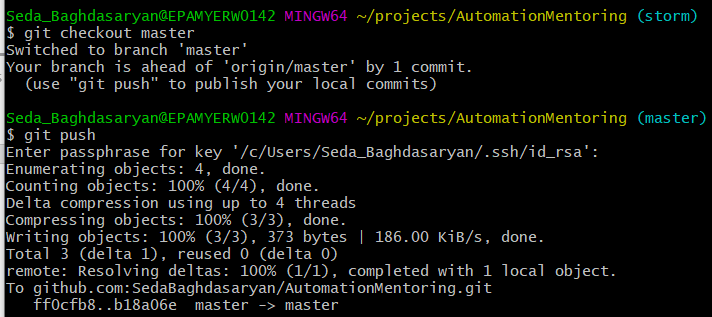




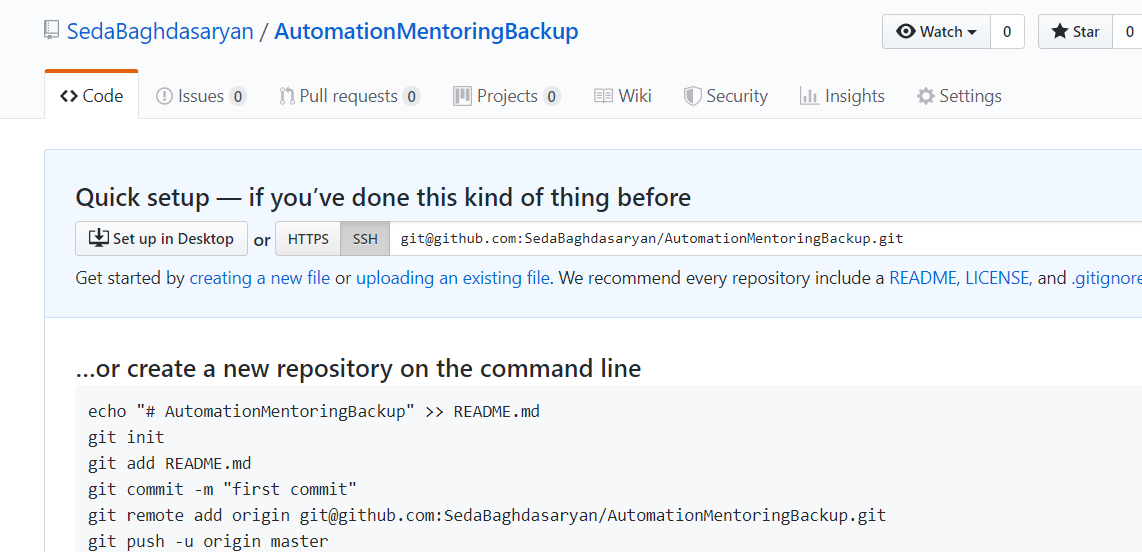
1. **“HARDCORE” Task:**

This task is based on the previous one (“Hurt Me Plenty”).

* 1. Push your repository and make sure all commits appeared in github.



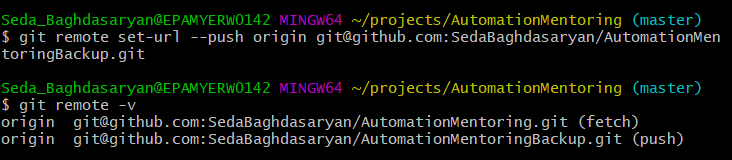
* 1. Create a new repository in github.



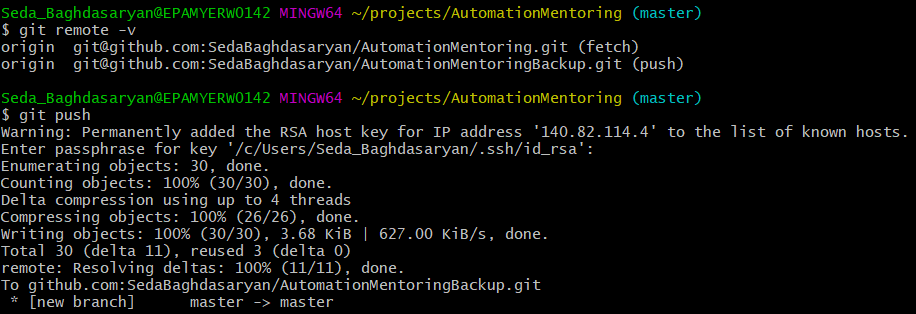
* 1. Set up remote in local repository in the way that allows:

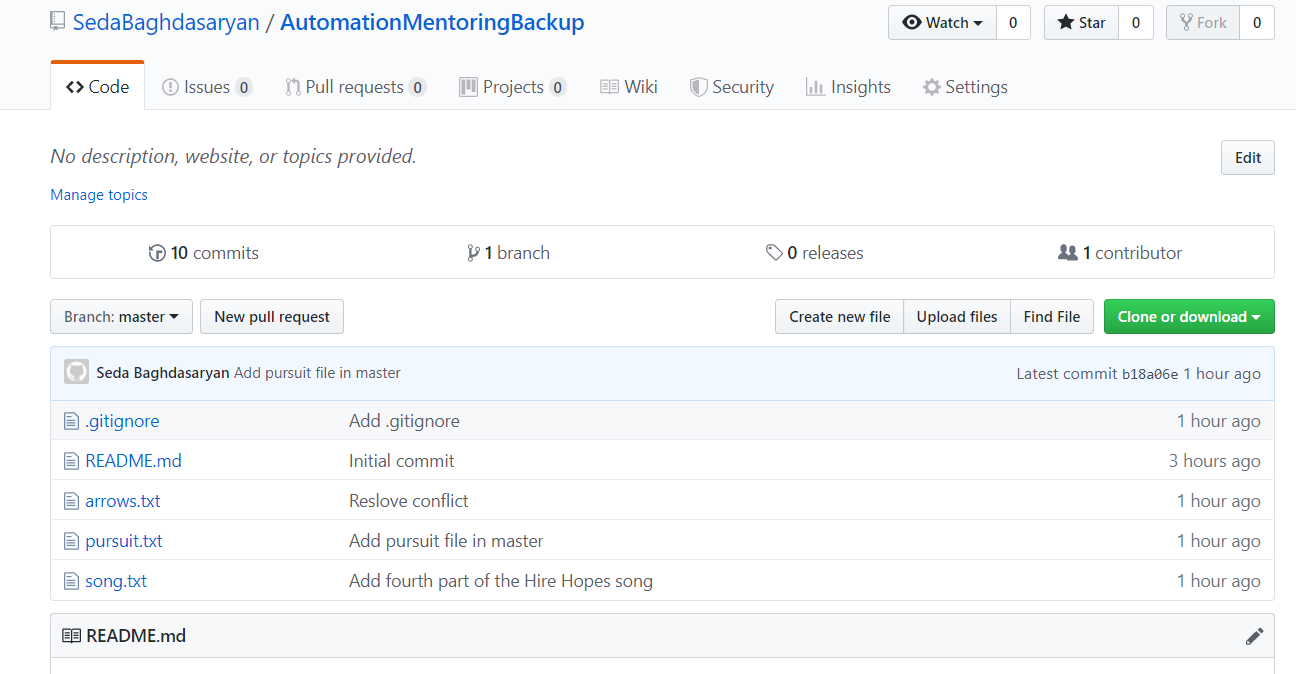
- pull from the first (initial) remote repository,

- push to the second (new) remote repository.

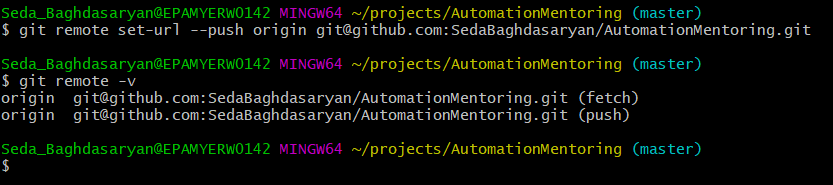


* 1. Make push and make sure the second repository in github is identical to the first one.





* 1. Get initial settings back for remote: pull and push of local repository hit the same remote repository in github.



As the result share the link to the repo with your mentor as well as screenshots of ‘GIT repository browser’ (use gitk command to open it) and history of actions performed.

## BONUS task

To complete this task, you need to repeat all four previous tasks using ONLY command line MINGW64. You are not allowed to use any file managers, text editors with graphic interface, gitk/git gui. You can use a browser only for actions in github. All other steps should be done via console only.

Commands that could help you:

*echo*

*cd*

*cat*

*vi*

## acceptance criteria

1. All steps of Task “I Can Win” done;
2. All steps of Task “Bring It On” done;
3. All steps of Task “Hurt Me Plenty” done;
4. All steps of Task “Hardcore” done.