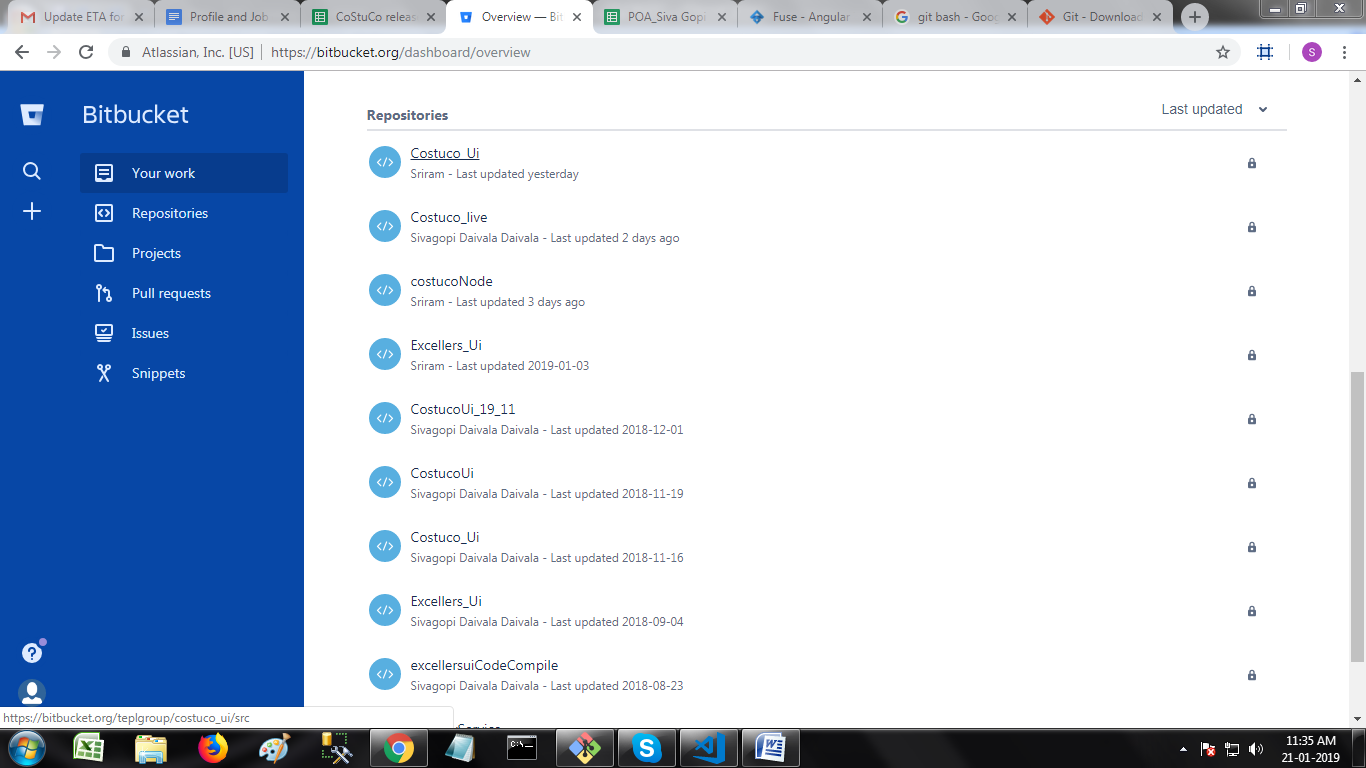
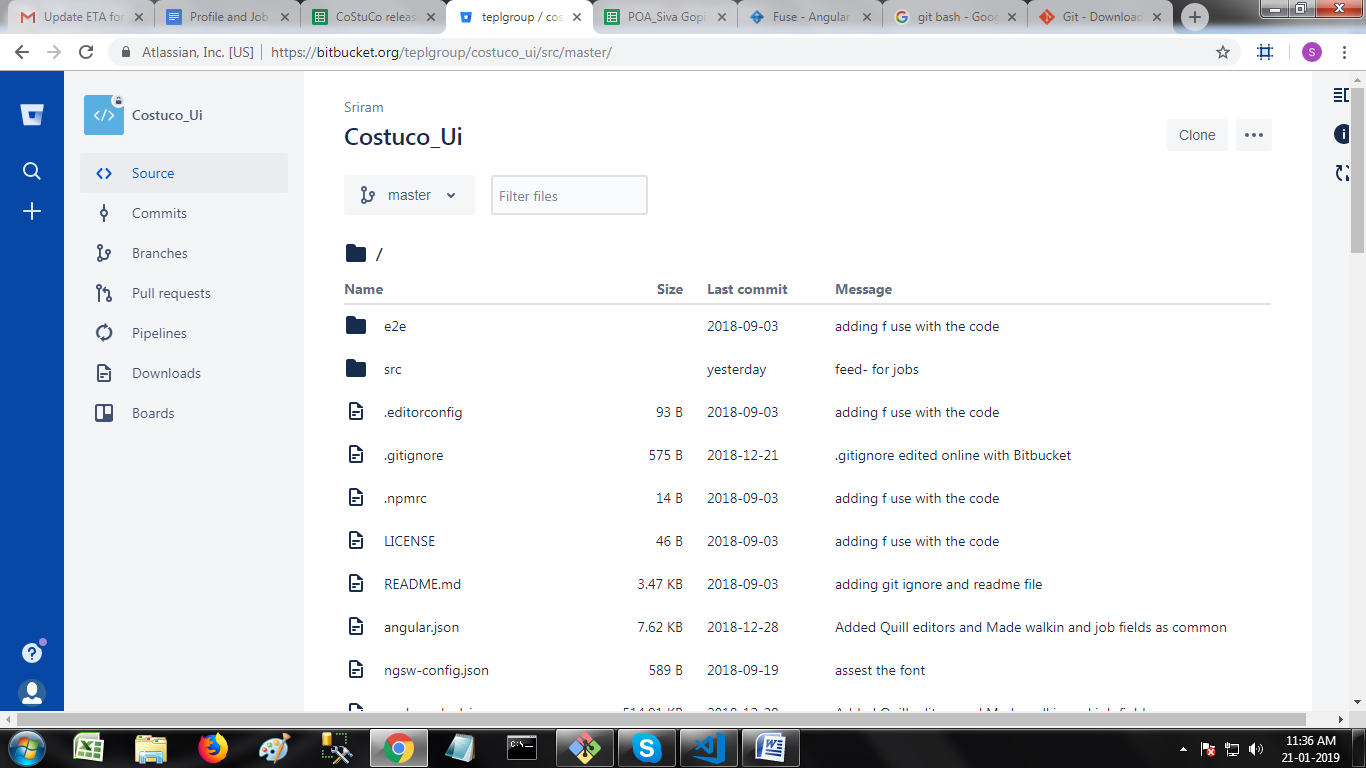
BIT BUCKET

First time use:

1. Open the bit bucket with official website <https://bitbucket.org>

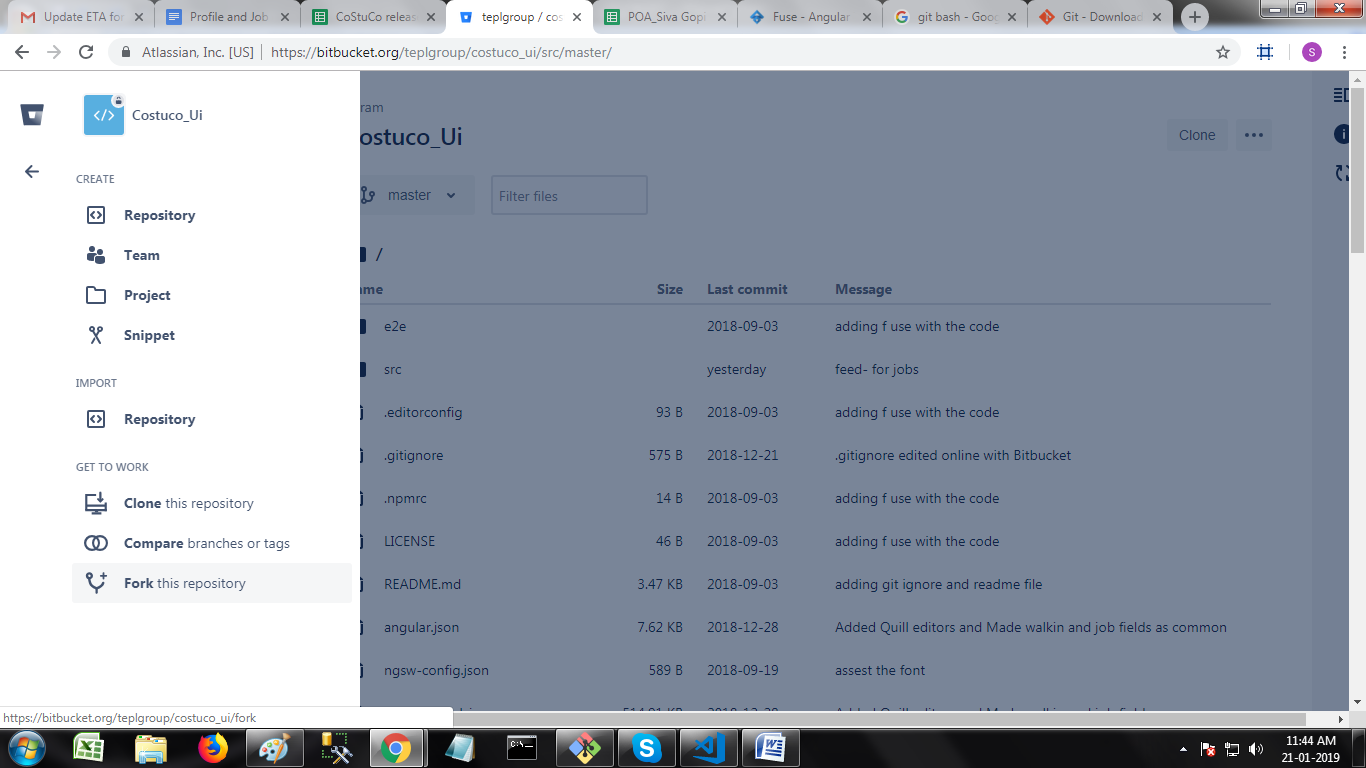


2. Select the repository you have to use from repositories list.



3 . Select the repository and fork the repository.

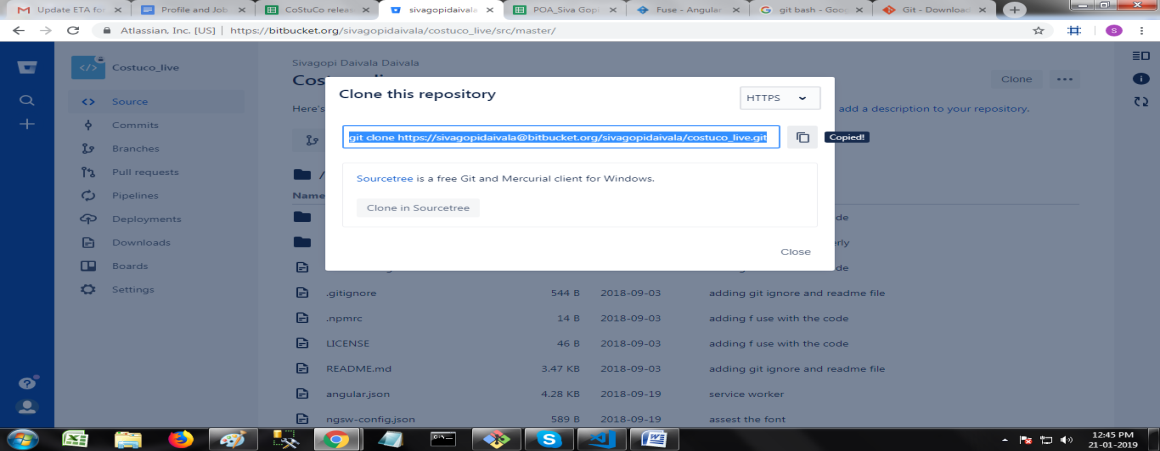
4. click the + icon on left top corner below search icon, click **Fork this repository.**

****

4. Name the repository, The name of the new repository should not match your own old repositories and click **Fork repository** button. After forking click the bucket icon on top left corner.



5. select your repository in repositories list. Click clone button on right top corner and click copy in popup box.



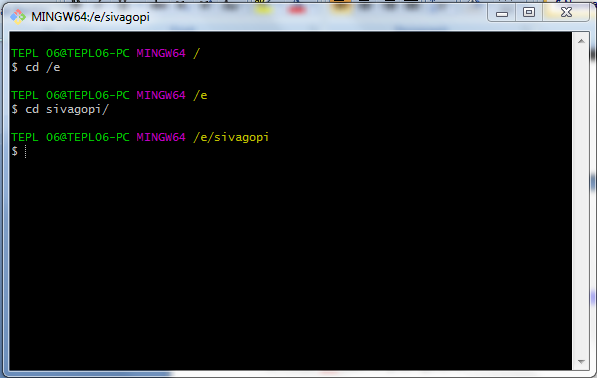
6. Download Git Bash from <https://git-scm.com/downloads>

7. Select the OS type and download.

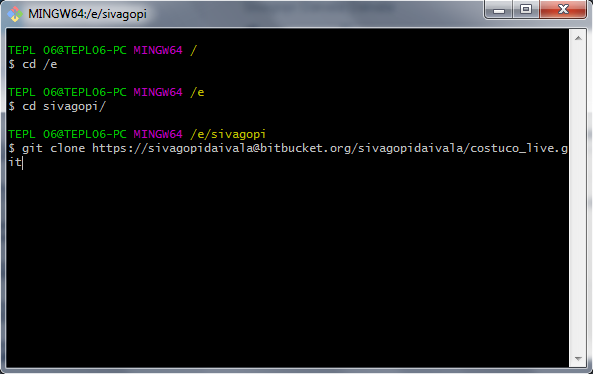
8. Install Git Bash in your system.

9. Open Start menu, search Git Bash and open.

10. Go to specific folder in local disc to clone the bit bucket repository.



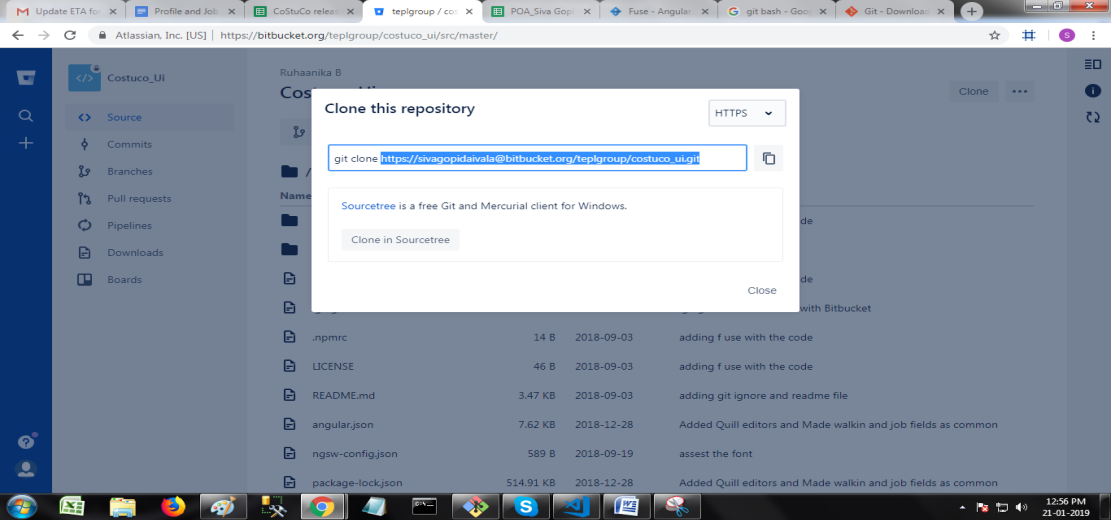
11. Paste the cloned line form bit bucket.



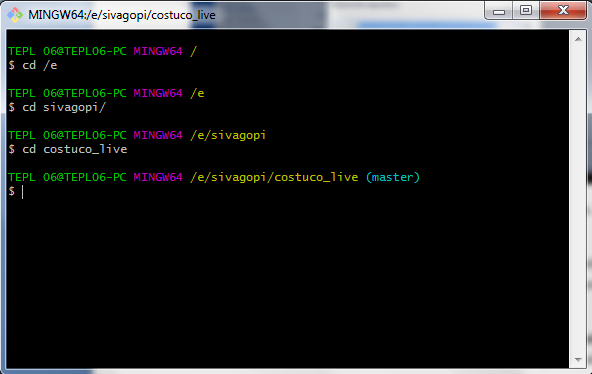
12. Type the command **git remote –v** to check it is cloned correctly or not. Your repository should be the origin and **don’t close the Git Bash**.

13. Open bit bucket, select the repository which one you have forked.

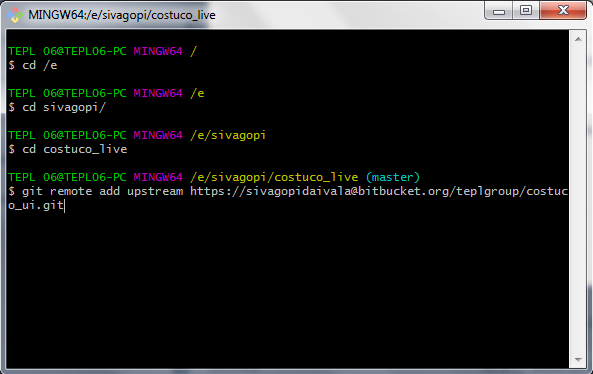
14. Click clone button and copy the url only, starting from https://



15. Open Git Bash and type **cd <your repository name>** or **cd + Tab.** You should get **(master)** after folder name. i.e., you are in master branch **and don’t close the Git Bash**.

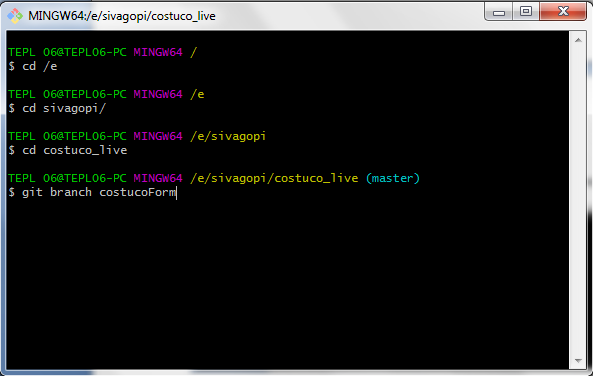


16. Enter the command **git remote add upstream <copied url from bit bucket>**. After executing, enter command **git remote –v.** Your will see **upstream** with forked person repository and **origin** with tour repository name.

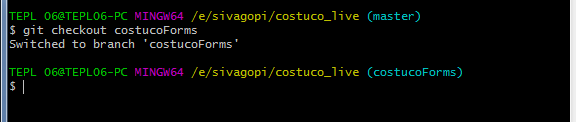


17. Now, The entire setup is ready. Open Git Bash, now you are **master** branch. It is not good process to work on **master** branch. Create other branch with name of your task without spaces ex: **imageUpload.**

18. Use command **git branch <yourBranchName>** will create a branch.



19. Use command **git checkout <yourBranchName>** to open the branch, now you will see **(yourBranchName)** next to your folder name.



20. Now you can work on the folder.

Daily using commands:

1. Daily before beginning work on your folder pull the code from the bit bucket to take the changes from the forked person repository.

2. To pull the code from the bit bucket enter command **git pull - -rebase upstream <yourForkedPersonSuggestedBranchToPull>**

3. After completing the task push the code to bit bucket from your branch for merge request.

4. To push the code from your branch enter command **git push -f origin <yourBranchName>.**

5. Open bit bucket in browser and search for latest updated repository with your name, open the repository and create a pull request.

6. Change Title with full explanation of your task and same in description. **Note: Title will be highlighted to your reviewers, so write full explanation in Title**.

7. Add the reviewers without neglect. Note: If you are creating pull request for first time from your branch, mention your reviewers carefully without missing.

8. Below the **create pull request button** you can see two tabs with name **Diff** and **commits**.

9. Click on Diff tab, it will explain about the previous changes and your changes with color difference.

10. Click on commits, It will explain you about how many times you have raised pull requests from same branch, what changes you have done in each commit.

Checking statuses of code before creating push and pull request:

1. Before pushing the code if you want to check the status, type the command **git status.** It will show the folders what you have done changes.

2. If you want to check the difference between previous code and your changes in particular file, type the command **git diff <filePath>**

3. After creating pull request if you get conflicts. Check the conflicts in the folder and resolve them, after resolving conflicts type the commands mentioned below

**git add <conflictedFilespath> 🡪 Add all conflicted file paths with space between them.**

**git rebase - -continue 🡪 This command is to take the conflict resolved changes.**

**git rebase - -skip 🡪 If you still see (REBASE 2/1) without any conlicts. Type this command.**

**git rebase –abort 🡪 If you want to get back from pull request and don’t want to pull. Type this command.**