A Project report

on

"REVAMP"

with

Source Code Management

(CS181)

Submitted by

Team Member 1 Ansh Bhasin   Roll No. 210992018

Team Member 2 Ishank Kapur Roll No. 210992086

Team Member 3 Ayush Vats   Roll No. 2110992080

Team Member 4 Manav Shandilya   Roll No. 2110992080

Department of Computer Science & Engineering

Chitkara University Institute of Engineering and Technology, Punjab

Jan- June
(2021-22)

Institute/School	**Chitkara University Institute of Engineering and Technology**
Name	**Ansh Bhasin**
Department Name	**Department of Computer Science & Engineering**
Programme Name	**Bachelor of Engineering (B.E.), Computer Science & Engineering**

Course Name	**Source Code Management**	Session	**2021-22**
Course Code	**CS181**	Semester/Batch	**2nd/2021**
Vertical Name	**Zeta**	Group No	G27

Course Coordinator	**Dr. Neeraj Singla**
Faculty Name	**Dr. Anuj Jain**

Submission

Name: Ansh Bhasin
Signature: Ansh

Date: 23/05/2022

CHITKARA UNIVERSITY
Table of Content

1. Create a distributed Repository and add members in project team

Creation of a distributed repository means creating a platform where each team member can work on their own workspace, to make that workspace a reality, we use branches.
To make people work on those branches we add them as collaborators.

Collaborators are people joining us to create the project we can add them in github repository settings

Step-1: To create a distributed repository, we first need to add collaborators in the git repository

Manage access

Add people

Select all   Type ▼

🔍 Find a collaborator...

ayushvats1227
Collaborator   Remove

It-Is-Ishank
Collaborator   Remove

Manav Shandilya
Manav-Shandilya • Collaborator   Remove

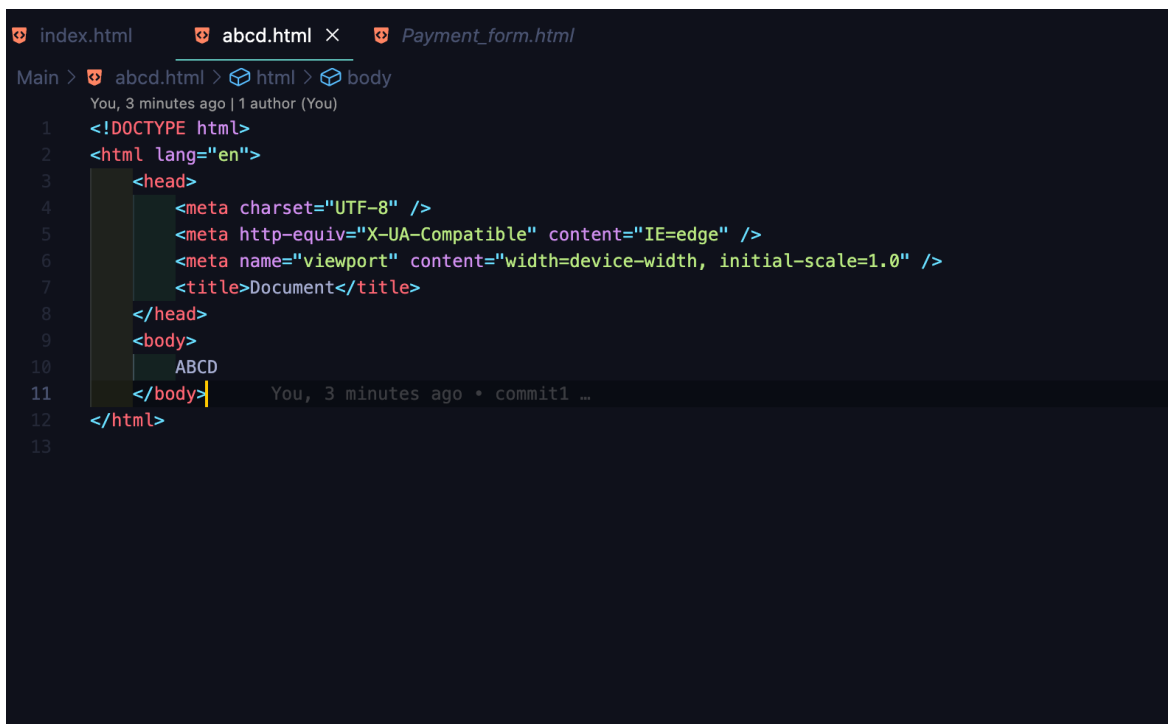## Step-2: Create branches for all and publish them to the remote git repository

```
[(base) anshbhasin@Anshs-MacBook-Pro Revamp % git branch
 * main
[(base) anshbhasin@Anshs-MacBook-Pro Revamp % git branch ishank
[(base) anshbhasin@Anshs-MacBook-Pro Revamp % git branch ayush
[(base) anshbhasin@Anshs-MacBook-Pro Revamp % git branch manav
[(base) anshbhasin@Anshs-MacBook-Pro Revamp % git push -u origin ishank
 Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
 remote:
 remote: Create a pull request for 'ishank' on GitHub by visiting:
 remote:      https://github.com/ItzAnsh/Revamp/pull/new/ishank
 remote:
 To https://github.com/ItzAnsh/Revamp.git
  * [new branch]      ishank -> ishank
 branch 'ishank' set up to track 'origin/ishank'.
[(base) anshbhasin@Anshs-MacBook-Pro Revamp % git push -u origin ayush
 Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
 remote:
 remote: Create a pull request for 'ayush' on GitHub by visiting:
 remote:      https://github.com/ItzAnsh/Revamp/pull/new/ayush
 remote:
 To https://github.com/ItzAnsh/Revamp.git
  * [new branch]      ayush -> ayush
 branch 'ayush' set up to track 'origin/ayush'.
[(base) anshbhasin@Anshs-MacBook-Pro Revamp % git push -u origin manav
 Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
 remote:
 remote: Create a pull request for 'manav' on GitHub by visiting:
 remote:      https://github.com/ItzAnsh/Revamp/pull/new/manav
 remote:
 To https://github.com/ItzAnsh/Revamp.git
  * [new branch]      manav -> manav
 branch 'manav' set up to track 'origin/manav'.
 (base) anshbhasin@Anshs-MacBook-Pro Revamp %
```

2. Open and close a pull request

A pull request is a term we use in git to check changes between any 2 branches and merge them if required, it helps us with comparison between the two files

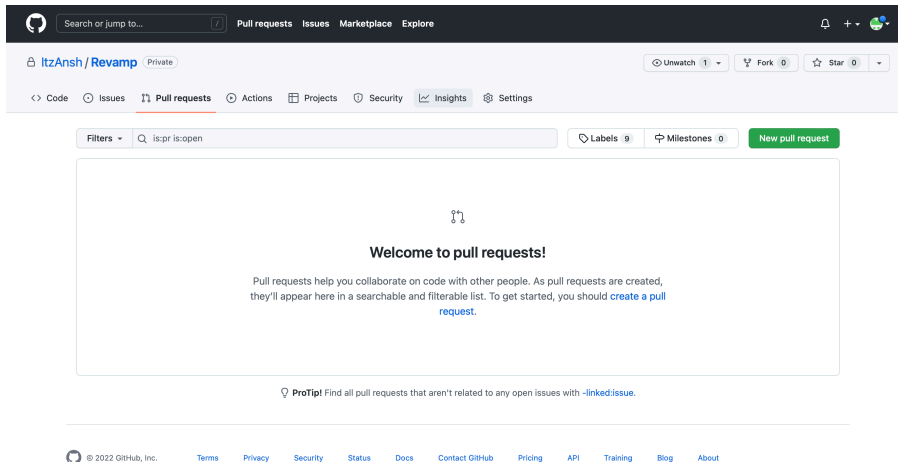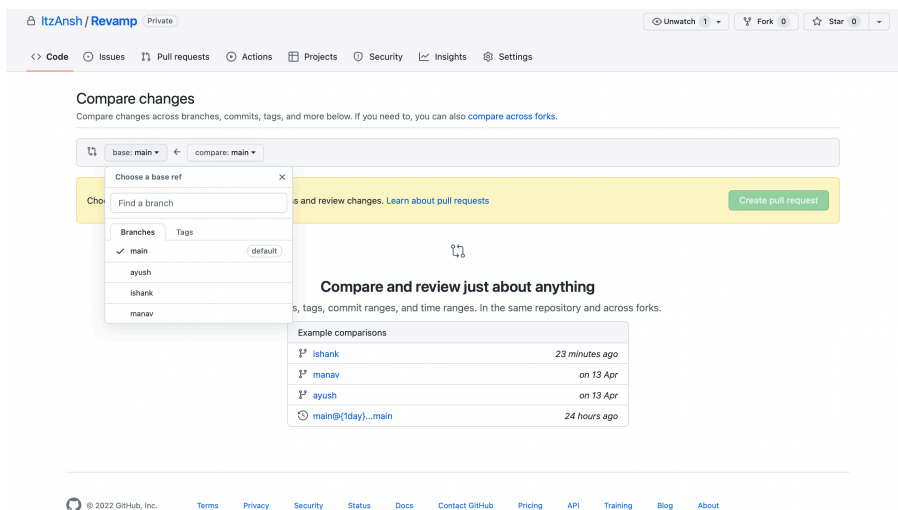### Step-1: To create a pull request do some changes in your file

Step-2: In order to check changes, we need to compare 2 branches so,

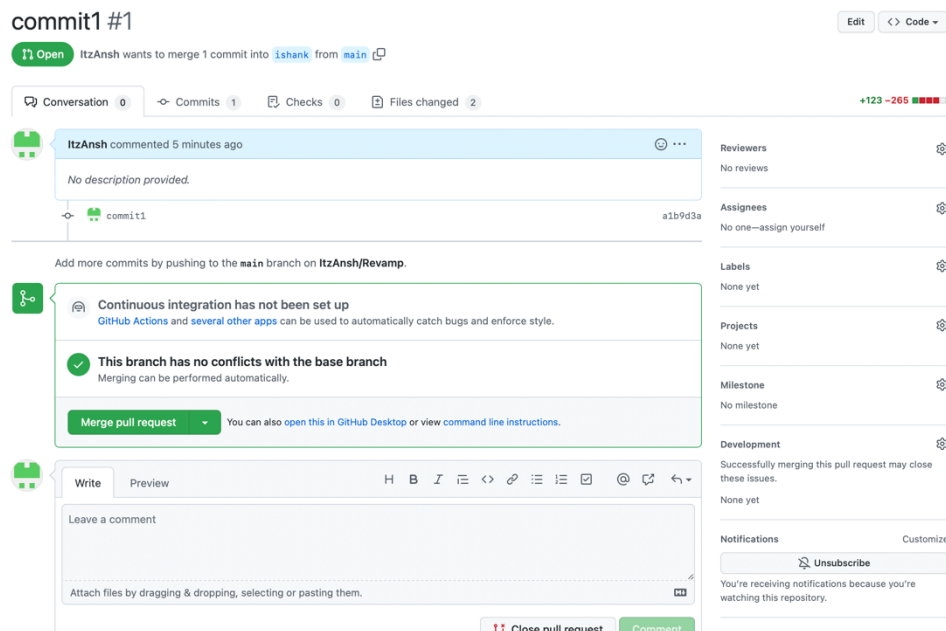- Go to your GitHub repository

- **Go to pull request section**

- **Click on New pull request and select 2 branches to compare**

- **After selecting to branches to compare the changes u want according to your need and click on create pull request**
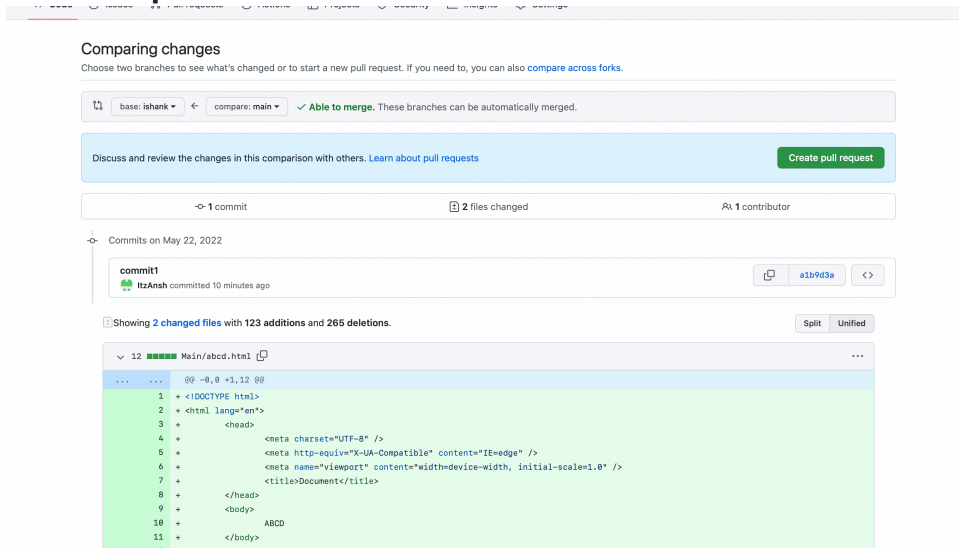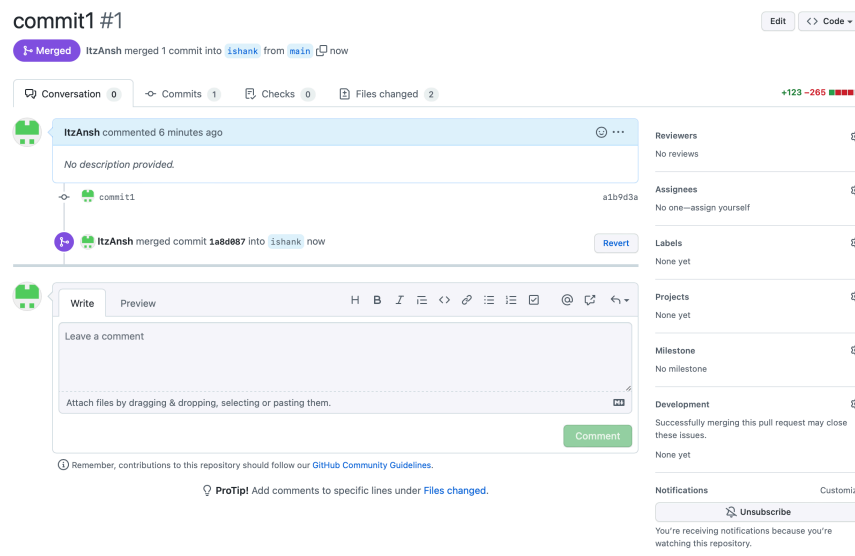

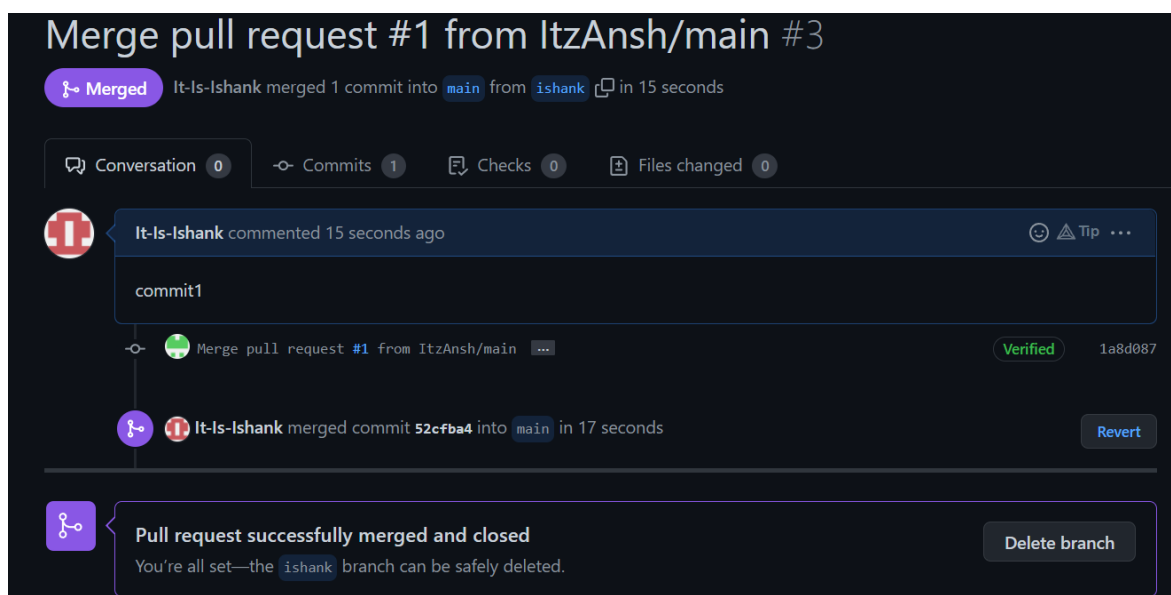- **Check for all the conflicts, resolve them and merge to close the pull request**

3. Each project member shall create a pull request on a team members repo and close pull requests generated by team members on own Repo as a maintainer.

Ansh:

Ishank:

Manav:

Ayush:

4. Network Graphs

The network graph is one of the useful features for developers on GitHub. It is **used to display the branch history of the entire repository network, including branches of the root repository and branches of forks that contain commits unique to the network**.

```
[(base) anshbhasin@Anshs-MacBook-Pro Revamp % git log --all --decorate --oneline --graph
*   c4c57d2 (origin/ayush, ayush) Merge pull request #4 from ItzAnsh/main
|\
| *   52cfba4 (origin/main) Merge pull request #3 from ItzAnsh/ishank
| |\
| | *   1a8d087 (origin/ishank, ishank) Merge pull request #1 from ItzAnsh/main
| | |\
| | |/
| |/|
| | | * 4d15525 (origin/manav, manav) Merge pull request #2 from ItzAnsh/main
| |_|/|
|/| |/
| |/|
| * | a1b9d3a (HEAD -> main) commit1
| |/
| * 5b9a8a8 psuh
| * 2e498d7 abcd
|/
* 47e8217 Revert "ishank"
* 535d87c ishank
* da1a6af Create settings.json
* a576064 Update index.html
* 54694d2 c
(base) anshbhasin@Anshs-MacBook-Pro Revamp %
```