**Case Study - Git : Merge Conflicts**

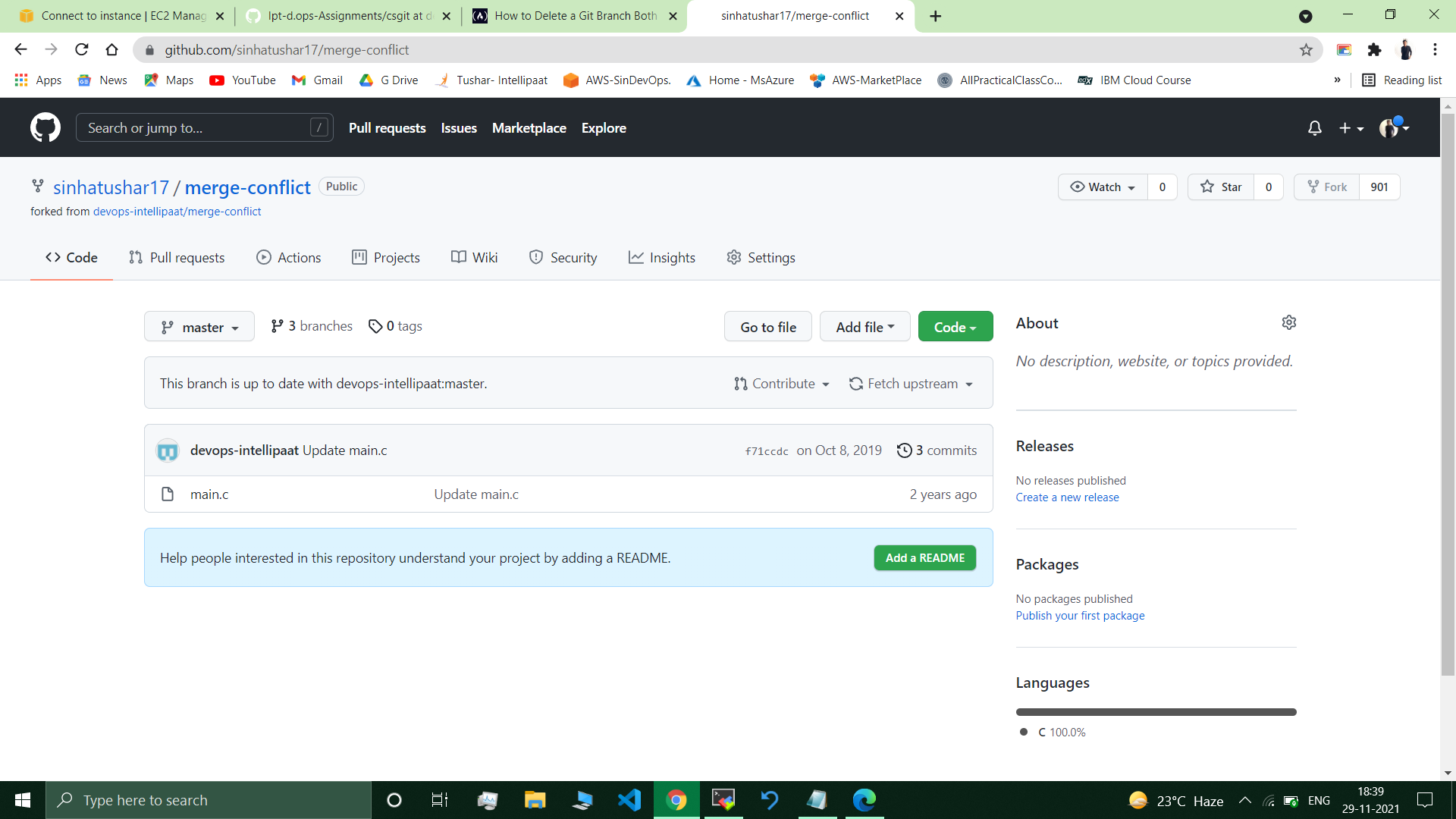
**To Do :**

**Problem Statement :**

You work for Zendrix Software & Co. You have been assigned the task of updating the Master branch of their Git repository with all the features from the feature branches. Following is the GitHub account, https://github.com/devops-intellipaat/merge-conflict.git Consider, • Feature1 branch to be a public branch • Feature2 branch to be a private branch The company relies on a monolithic architecture, and for now all the code resides in one file “main.c”. The respective features have been added in the feature branches for main.c. Meanwhile, a security patch was made to the master branch, and now feature1 and feature2 branches are behind from master by 1 commit. Following tasks have to be done: 1. Update Feature1 and Feature2 branch with the Security Patch 2. Apply changes of Feature1 and Feature2 branch on master 3. Finally push all the branches to GitHub For Solving this, please fork the repository to your Github account and then work.

**Solution :**

We want all the codes in a same file i.e pseudo code,security and feature1 & feature2 code together in a file.

Visited the GitHub url where project files were stored, Forked it.

git remote set-url origin git@github.com:sinhatushar17/merge-conflict.git

//git remote -v : to view remote repository.

mkdir csmc

cd csmc

git clone git@github.com/devops-intellipaat/merge-conflict /cloned the repo

ls //merge-conflict dir available

cd merge-conflict

//on branch master

git checkout feature1

git checkout feature2

git branch // feature1,\* feature2, master

git checkout feature1

ls //main.c available

cat main.c //feature1 part of code present.

//main.c - main code file,, main.c.orig - raw file (it contains unnecessary lines likr: <<<<<<< HEAD,=====).

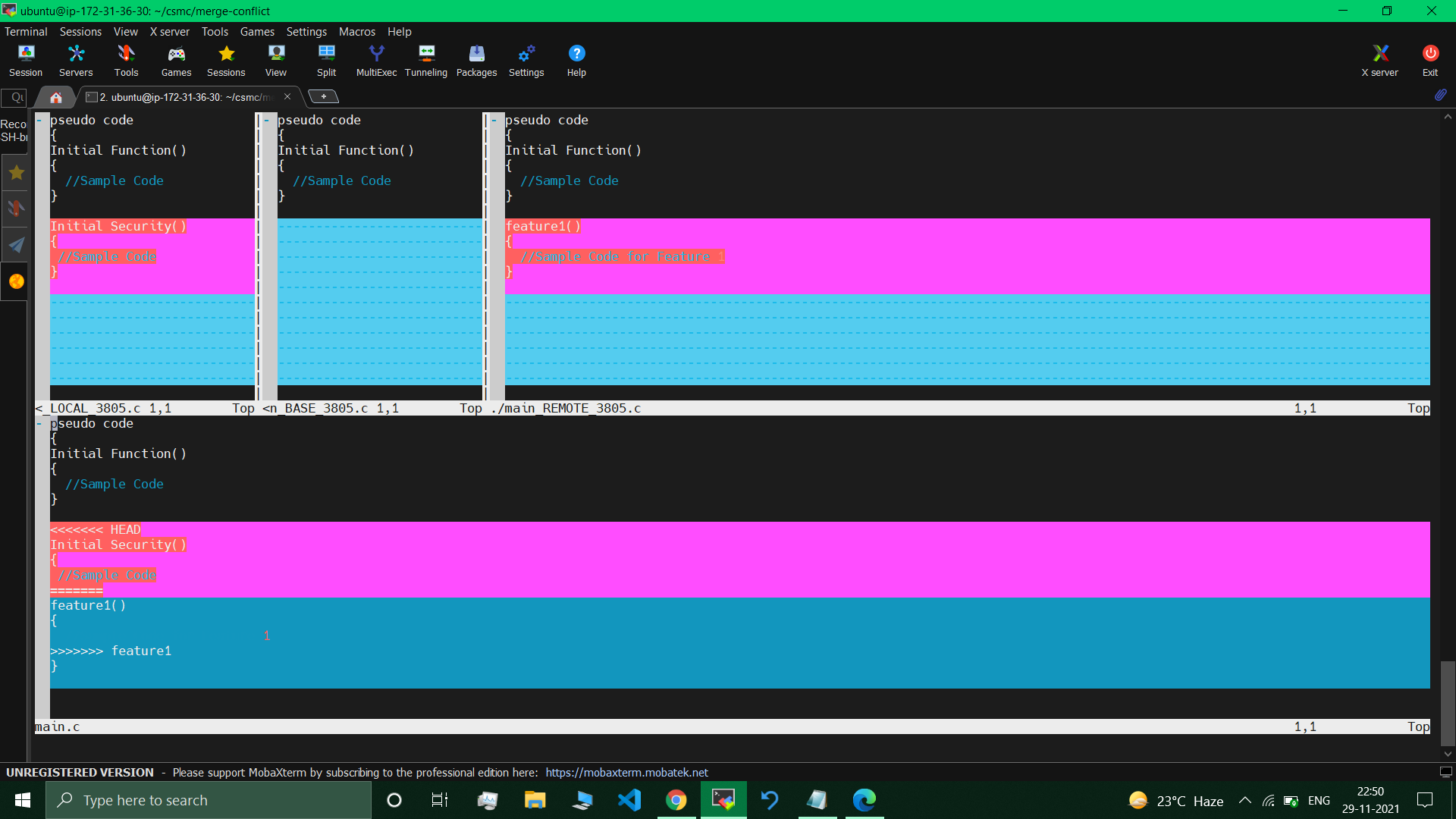
git checkout master

git merge feature1 //Merge conflict in main.c Automatic merge failed.

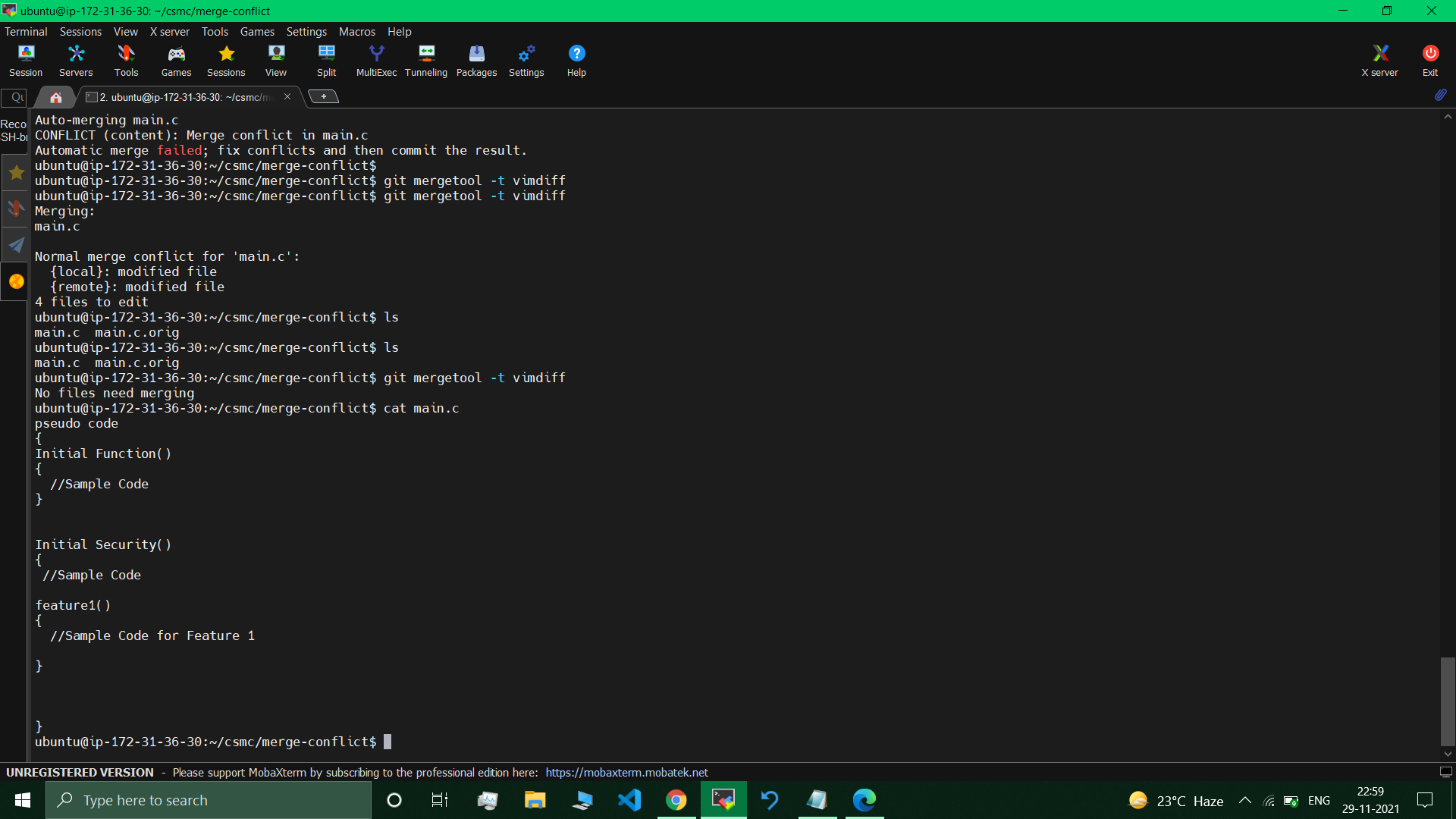
//We have to fix conflicts and then commit the result.

Resolving Conflicts :

git mergetool -t vimdiff //removed unnecessary lines from main.c./(orig) //merged feature1



cat main.c //unnecessary lines werre removed so not present here.

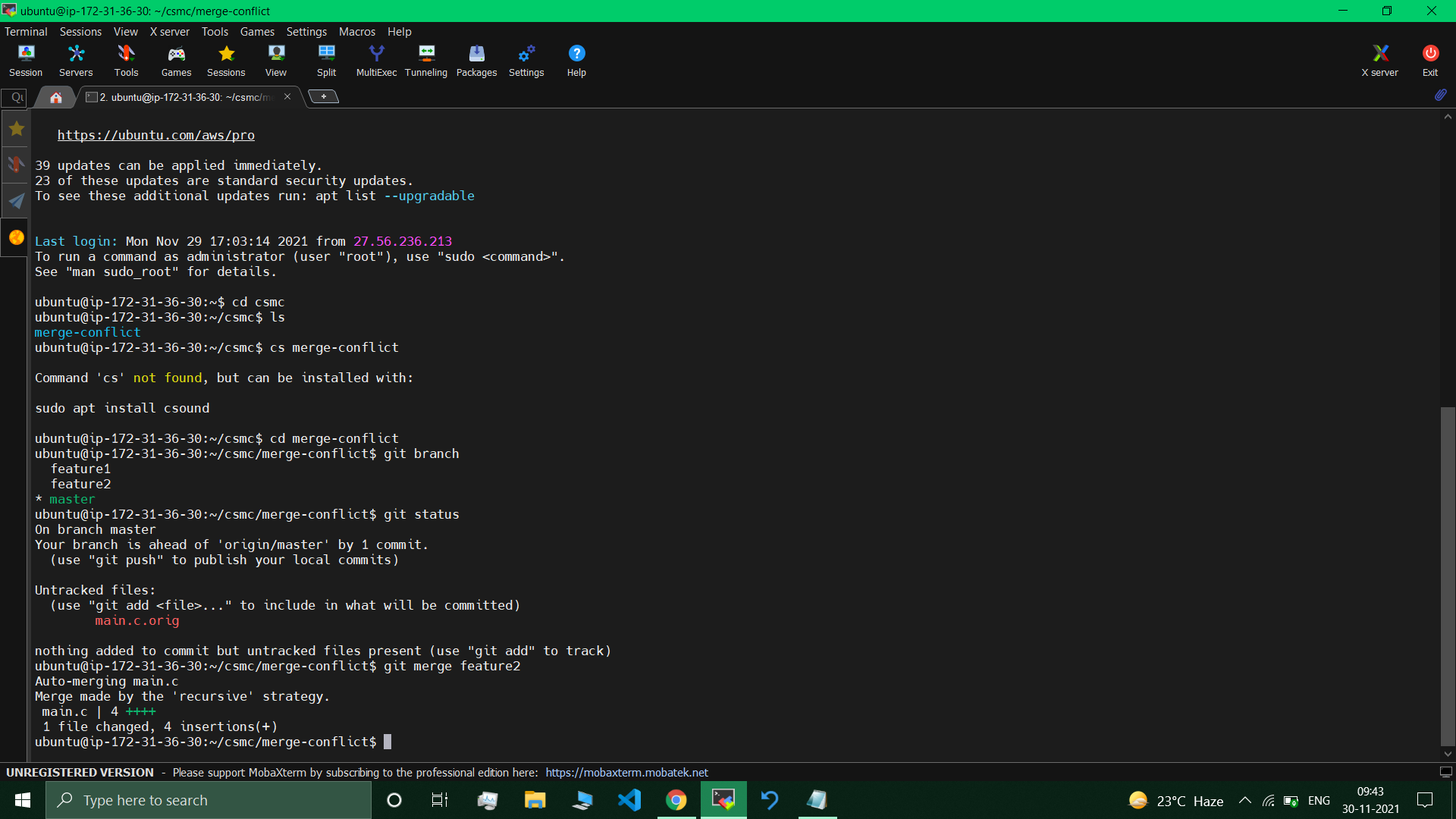


git add main.c

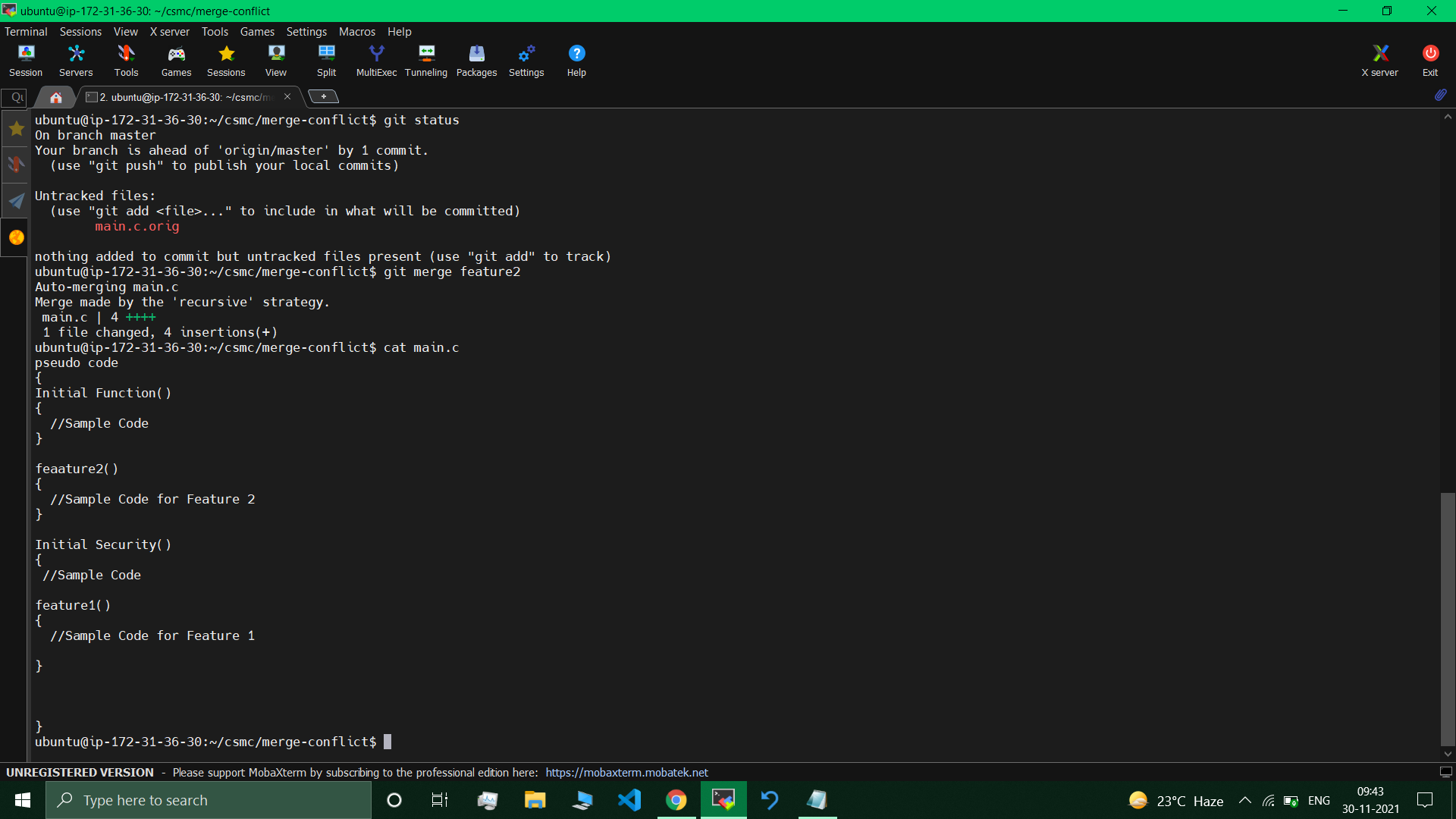
git commit -m "msg"

git commit -m "m"

git merge feature2

 //feature2 merged

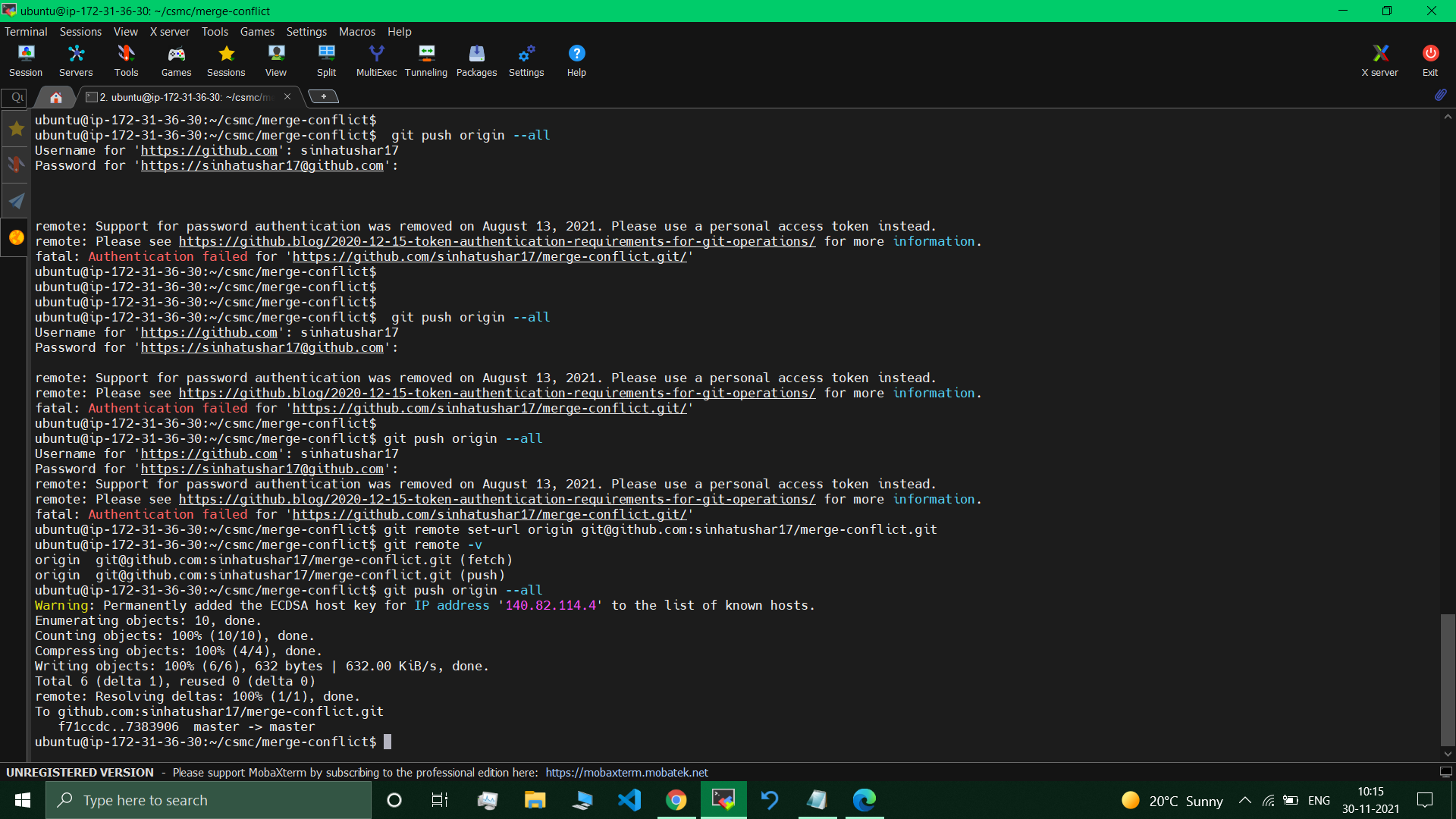
cat main.c //codes from feature2 & f1 + security code all are present.



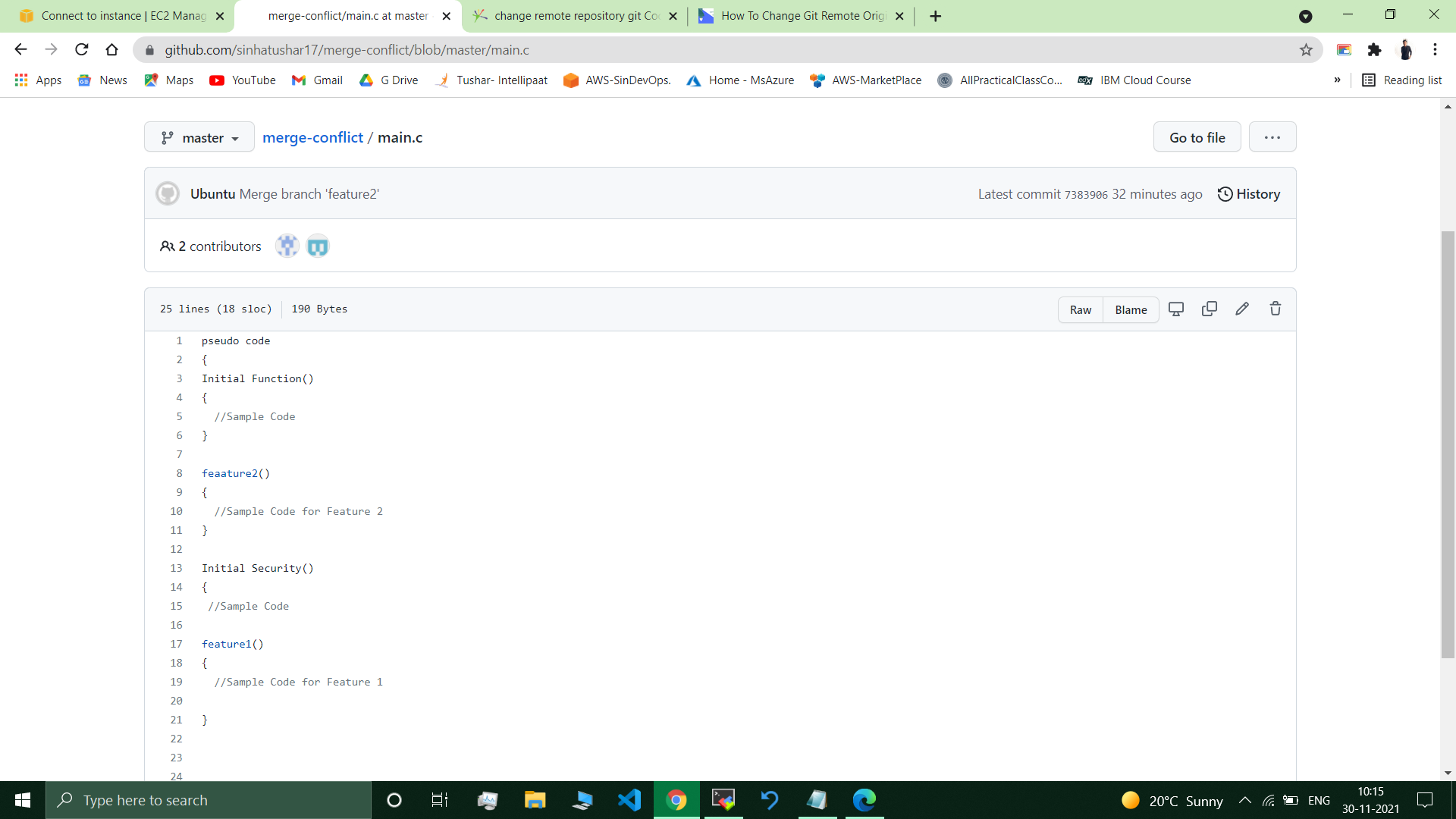
git remote add origin git@github.com/sinhatushar17/merge-conflict.git

//Now we can make a push.

git push origin –all



Visiting & checking on github repo --> main.c : changes applied , f1 f1 + security codes available in main.c file.

 //Available.

---------------------------------------------------------------------------------------------------------------------

**Submitted By : Tushar Sinha**

**Email ID :** [**sinhatushar17@gmail.com**](mailto:sinhatushar17@gmail.com)

**Enrolled in the “DevOps Architect Master’s Training Program”**

,