Dang Bao Nghi Nguyen

COURSE: SOFTWARE DEVELOPMENT PROJECT

ASSIGNMENT 3

GIT REPOSITORY

2020



**CONTENTS**

1. **Procedure 3**
   1. **Create the Project 3**
   2. **Adding Project to Git Source Control 4**
   3. **Modify the Project 4**
   4. **Commit to Git 5**
   5. **Modify the Project Again 5**
   6. **Commit to Git 6**
   7. **Add new Branch 7**
   8. **Modify new Branch 7**
   9. **Merge to Master 8**

**1.10. Publish to GitHub 8**

**1.11. Synchronize Changes to GitHub 9**

**1.12. Clone from GitHub 9**

**1.13. Conflict 10**

**1.14. Solve the Conflict 13**

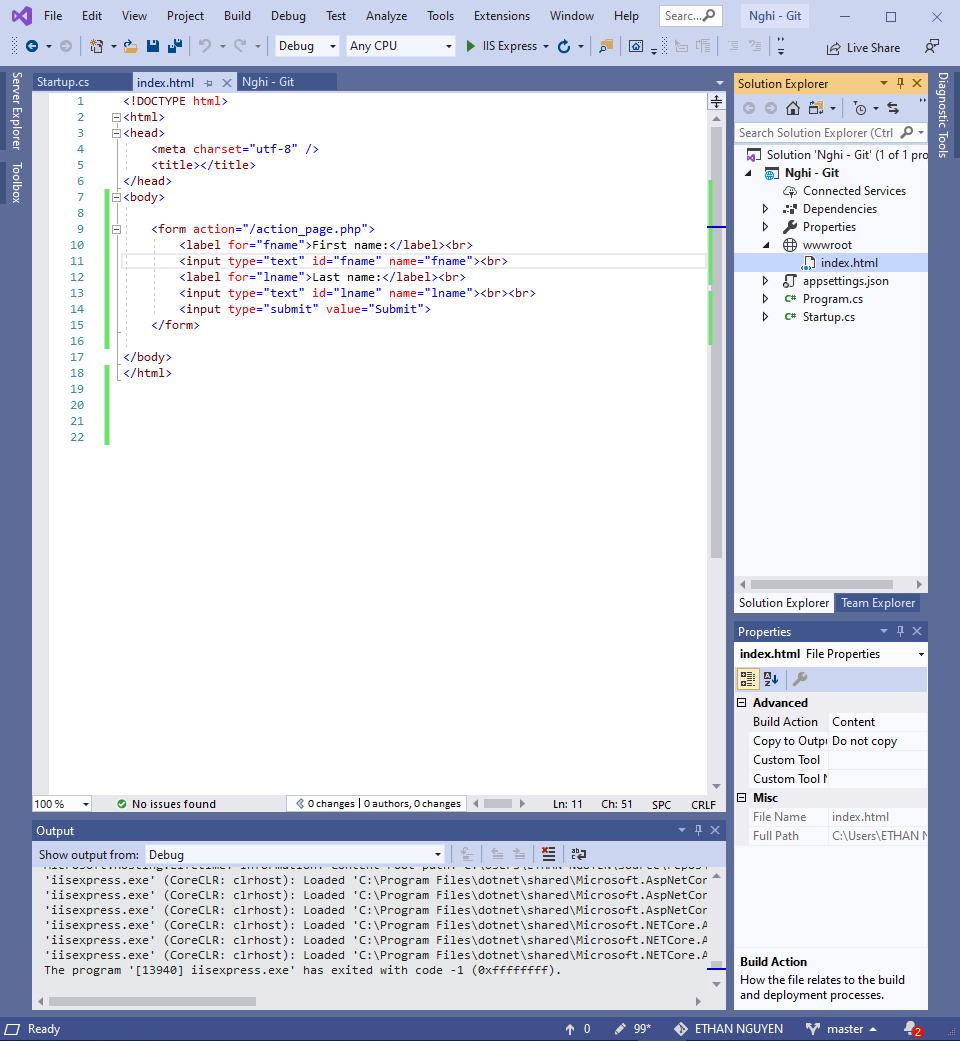
**1.15. Synchronize with GitHub 14**

1. **Procedure**

Based on the previous assignment 2, we will continue using the same project, but modify the contents to make the assignment 3 demonstrate possible, using **Command UI**. The assignment 3 starts from **1.10** afterward.

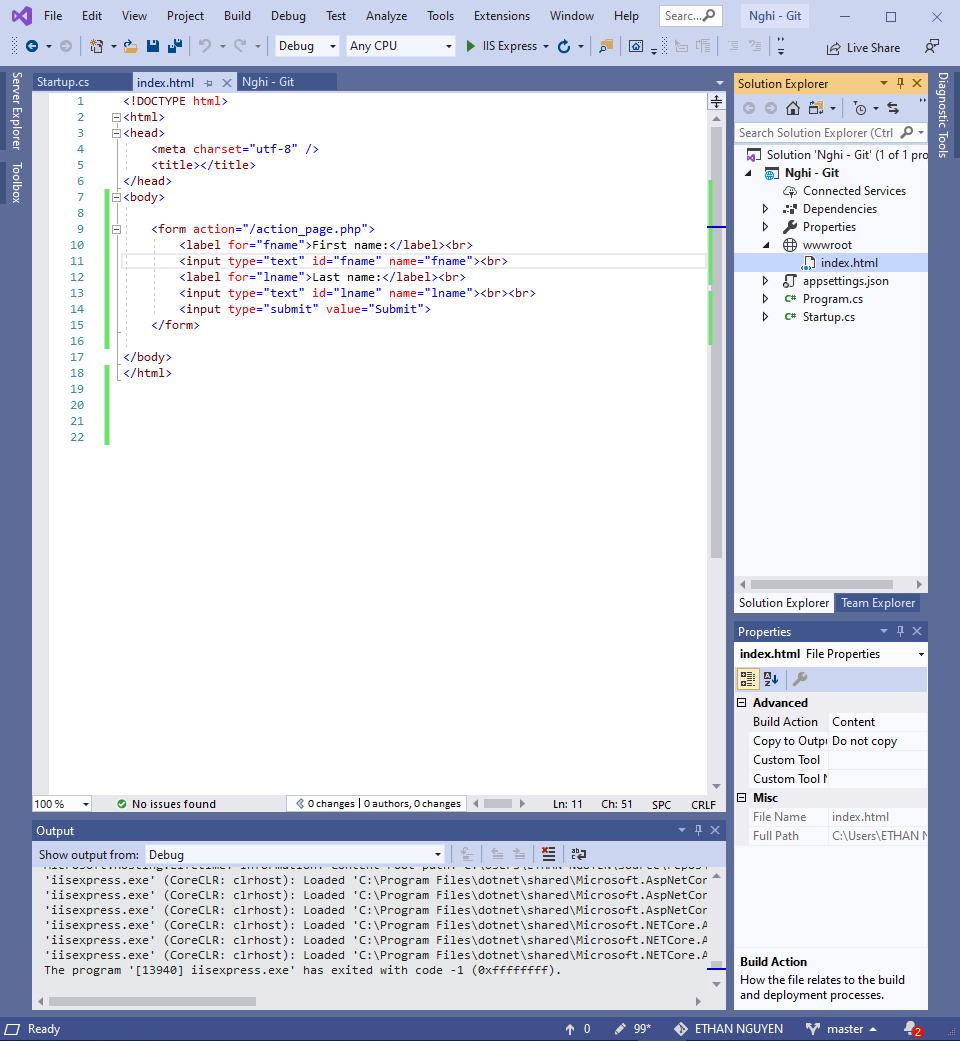
* 1. **Create the Project**

There are various ways of demonstrating **Git Repository**, in this case we will be using **Visual Studio 2019**. We first open **Visual Studio 2019** and create any type of web application we want. Something like this will do:



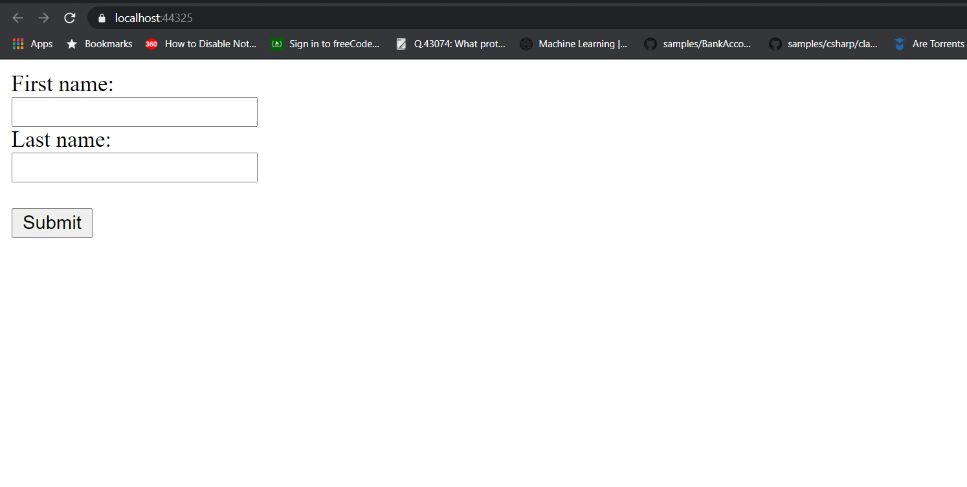
* 1. **Adding Project to Git Source Control**

After we have successfully created and modify the project into our desire, we can now add the project to **Git Source.** To be able to add the project into the **Git Source**, look at the bottom right corner, we can easily connect our project with **Git Source.**



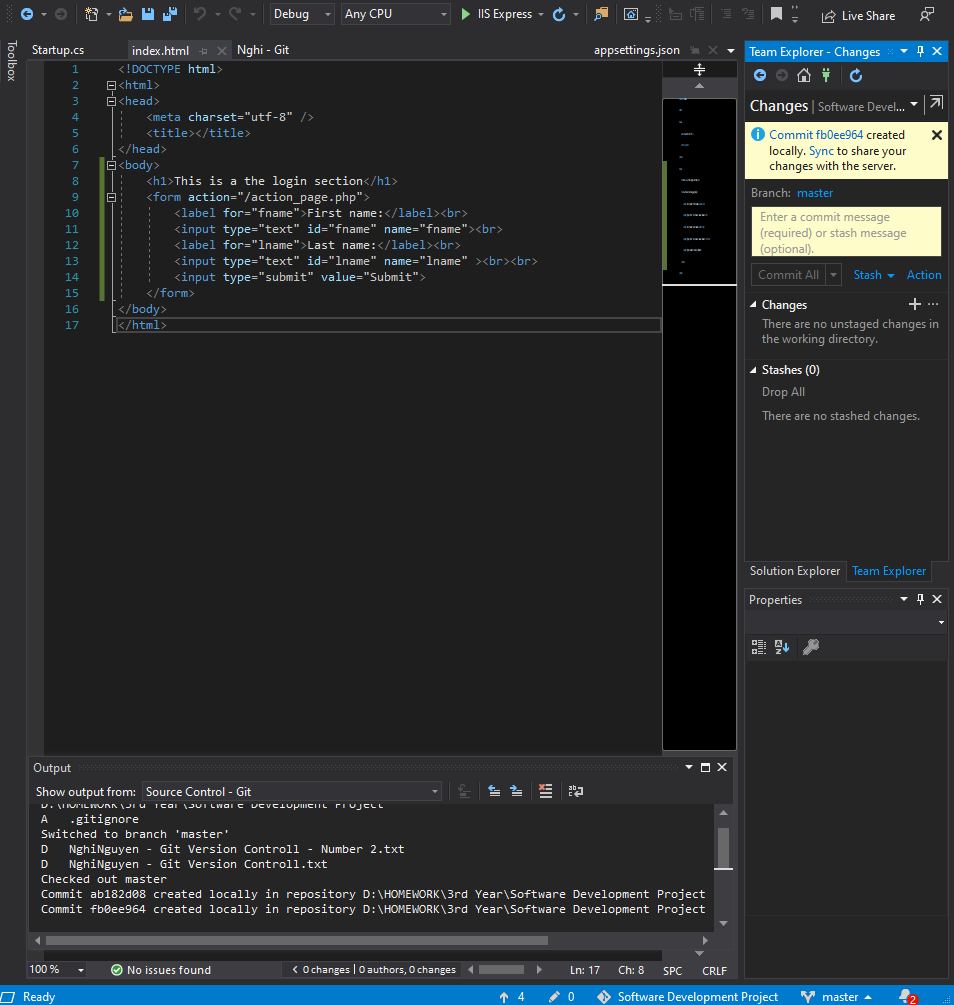
* 1. **Modify the Project**

Let us test out our new project to see if its work or not, the result should be expected to be working properly.



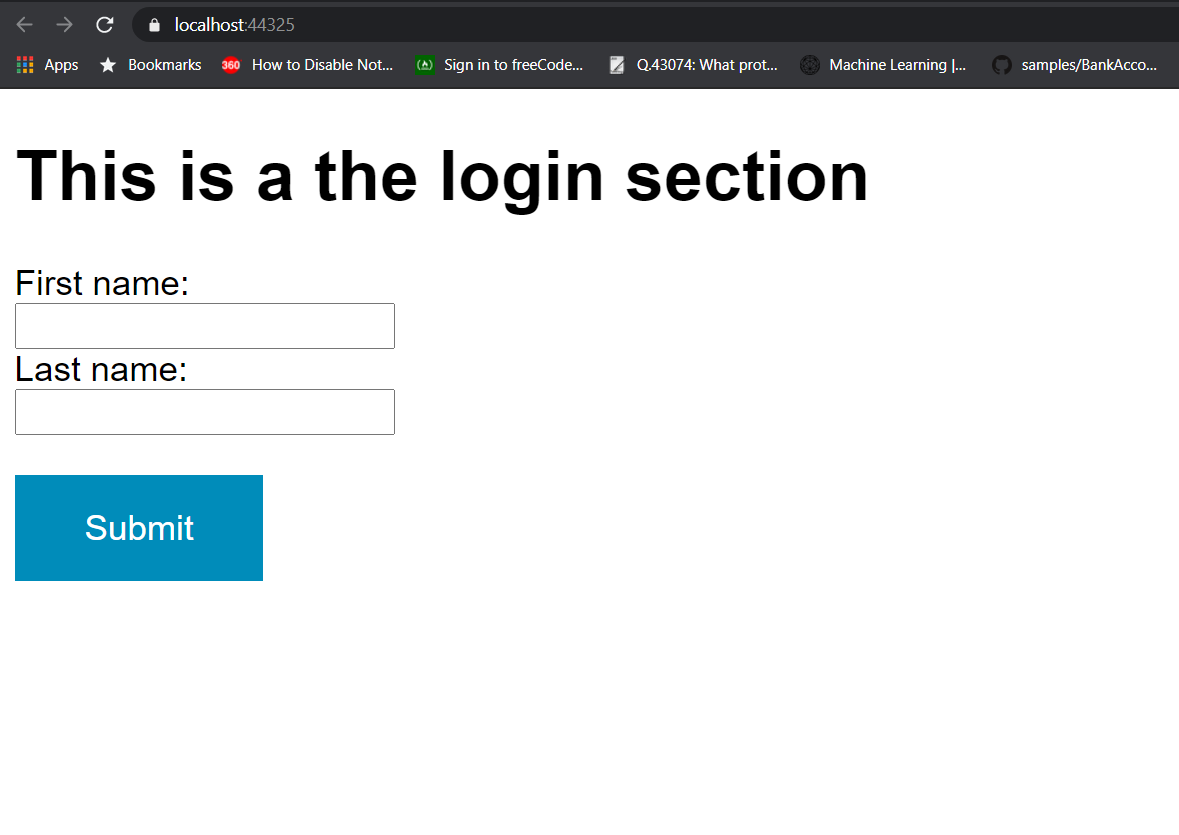
* 1. **Commit to Git**

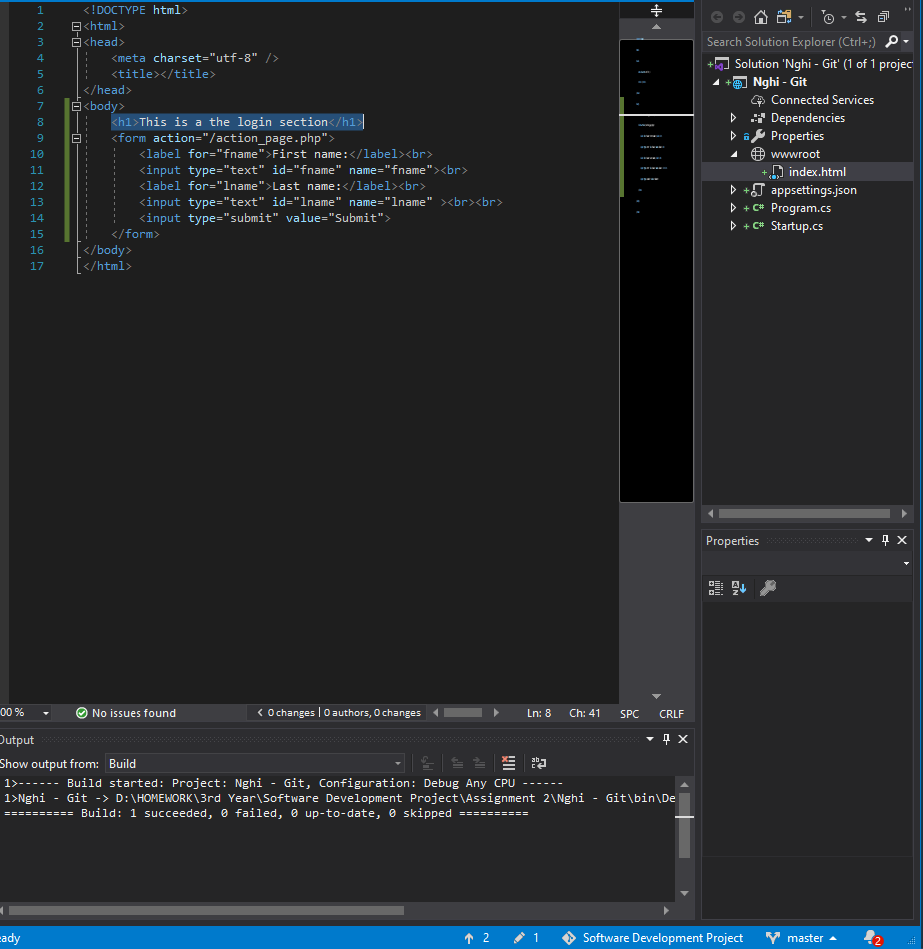
Now we do a **Git commit** to apply the project into **Git Repository**.



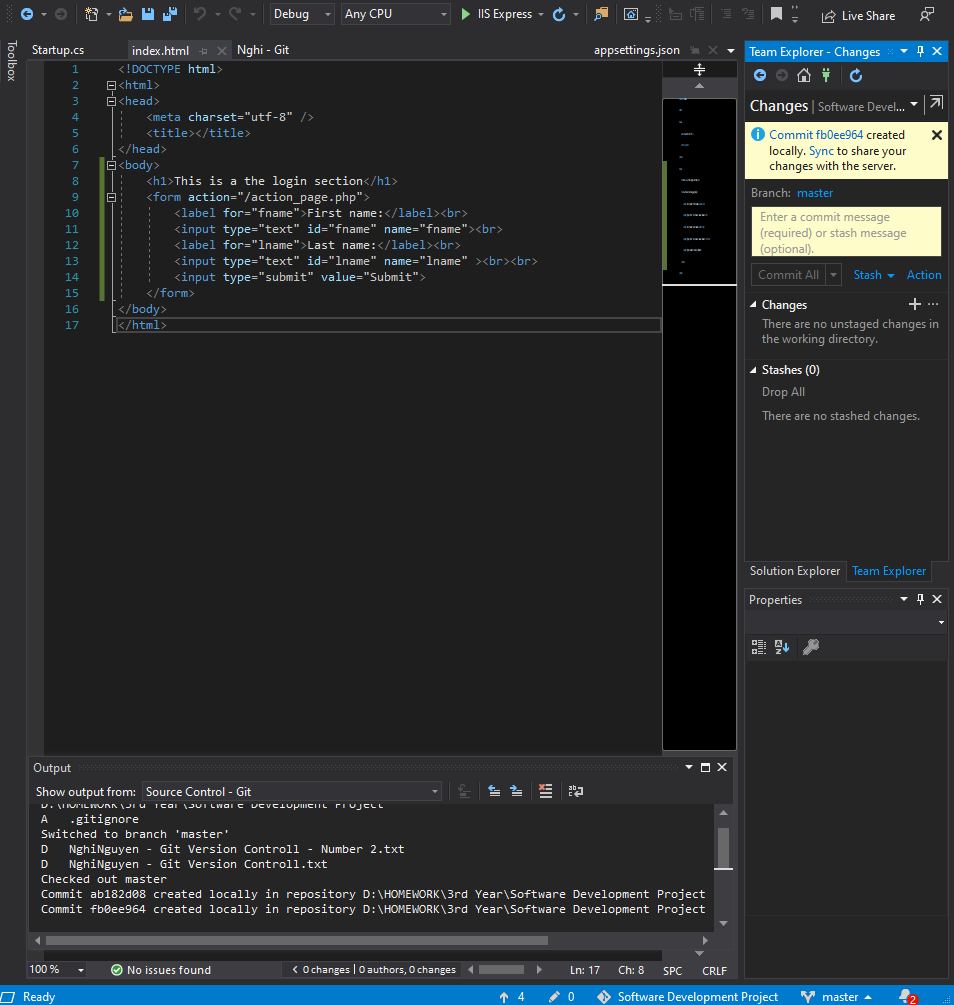
* 1. **Modify the Project Again**

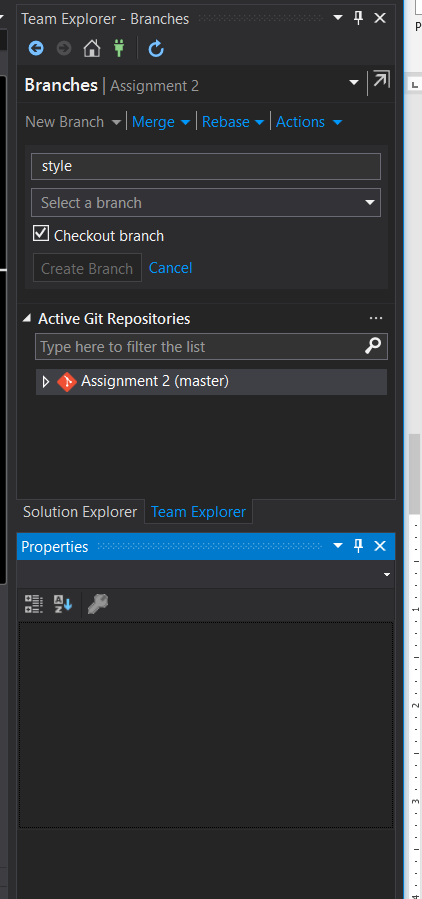
Now we try to modify the project by adding some new context, something like, in this case will be a new header, says “**This is the login section”**. The result should be expected like this:

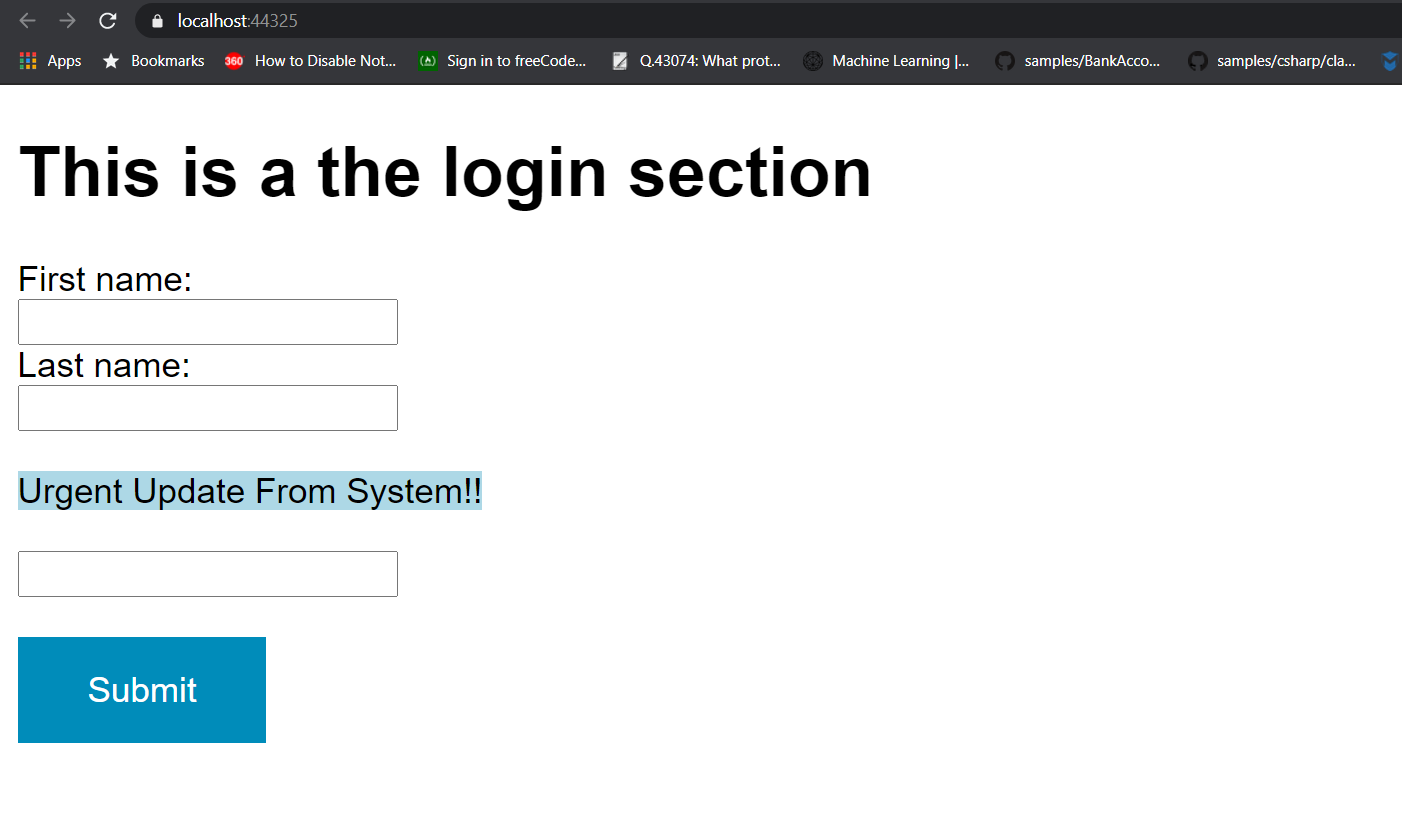




* 1. **Commit to Git**

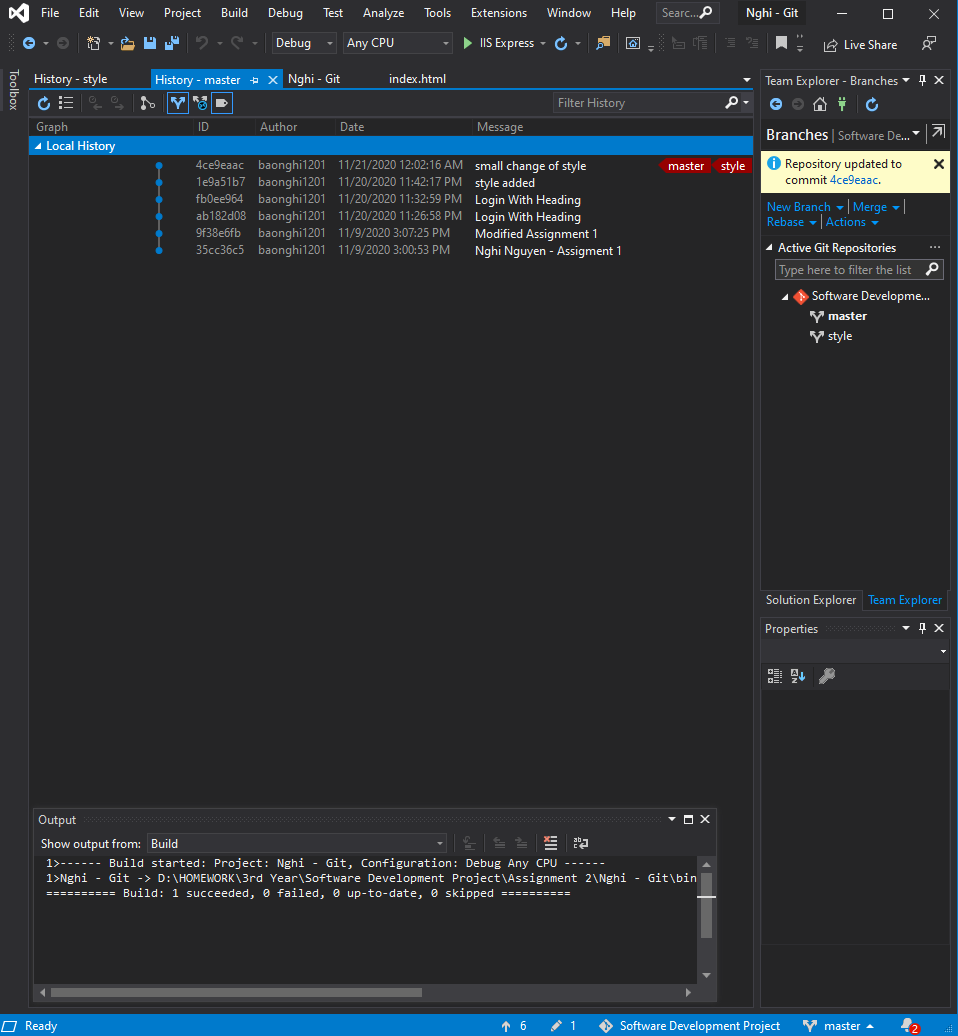
Then again, we commit this modified section to **Git Repository.**

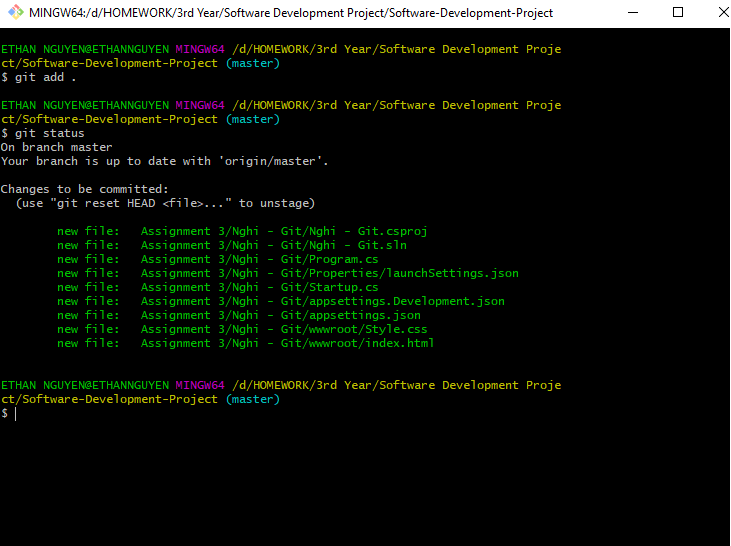
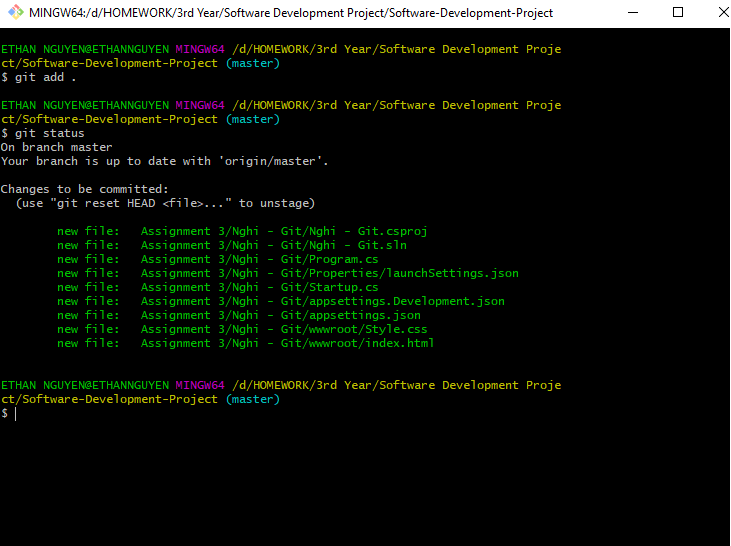
* 1.  **Add New Branch**
  2. **Modify new Branch**

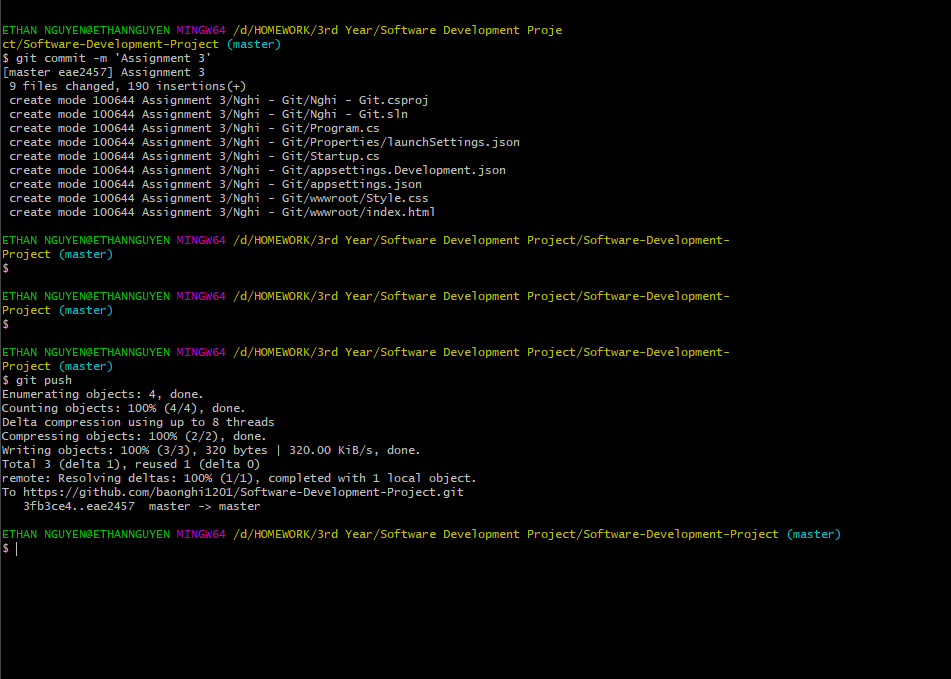
By modifying this new branch, is actually changing some of the context in the project. But the only difference here, is that we will first need to select this new branch we just created called, in this case **“style”** from there, we can now modify what ever we want. Then after that commit it to the **“style”** branch, in this case the changes will be adding another header to the project.

* 1. **Merge to Master**

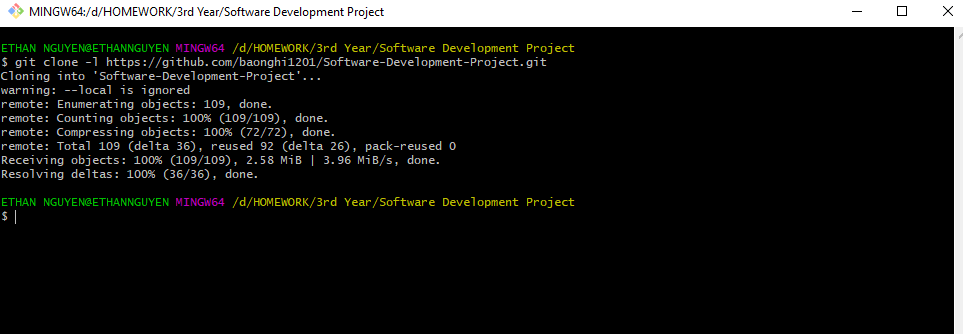
After we done playing around with our project, let us merge these branches together into one **master** branch.



* 1. **Publish to GitHub**
  2. **Synchronize Changes to GitHub**

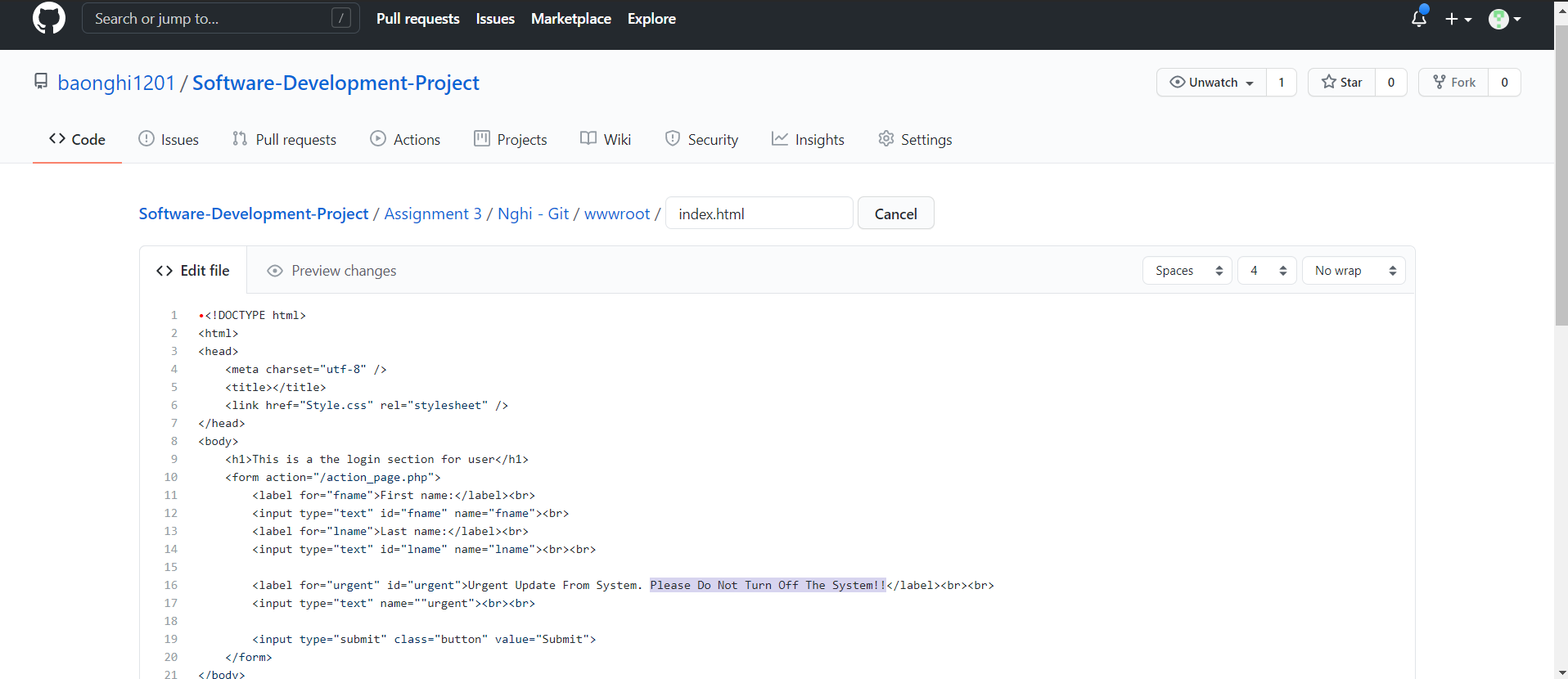
****The purpose of making synchronize changes and push that local changes to GitHub, is to ensure the connect between the local and remote site.

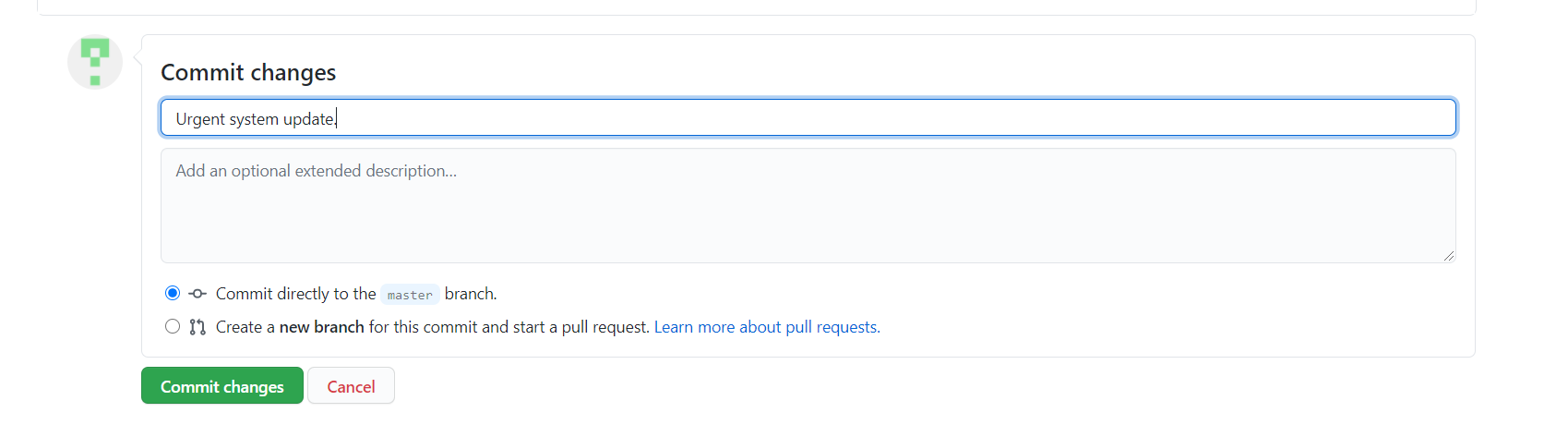
* 1. **Clone Project from GitHub**

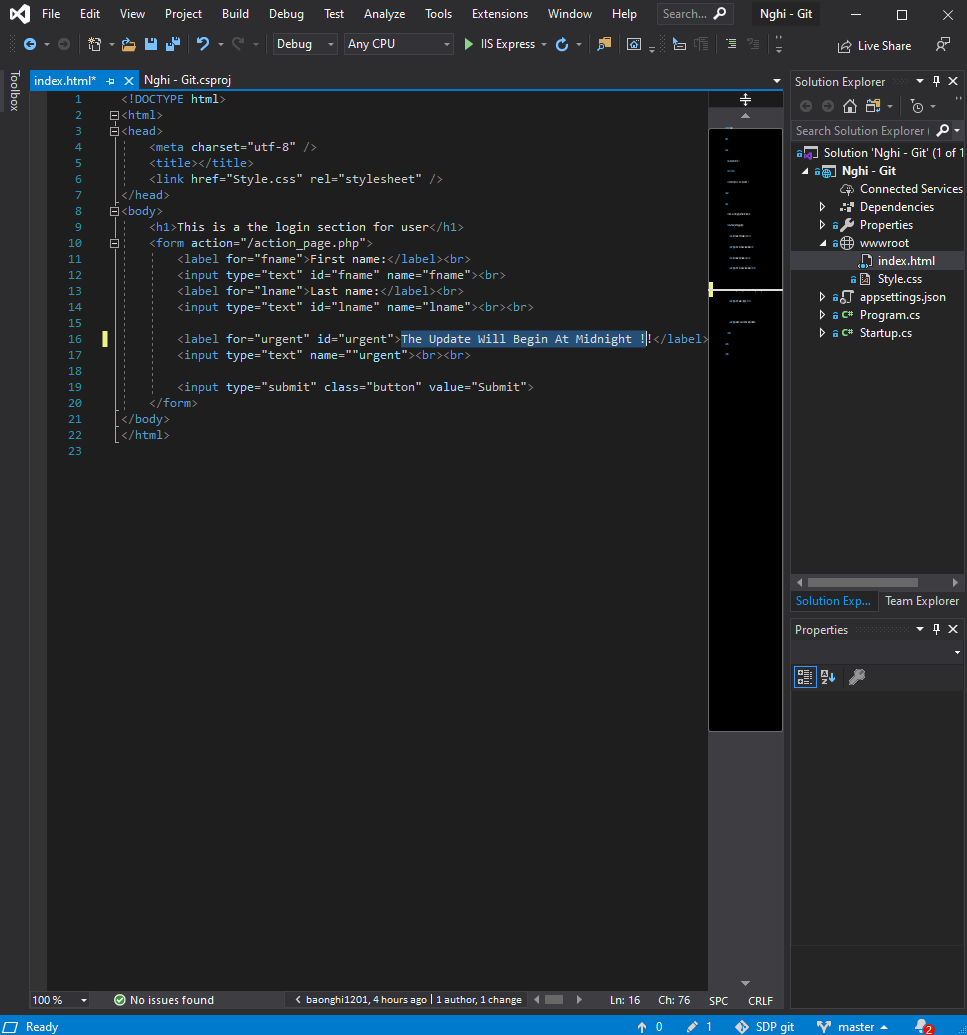
Let have one friend clone the project from remote GitHub to their local machine.

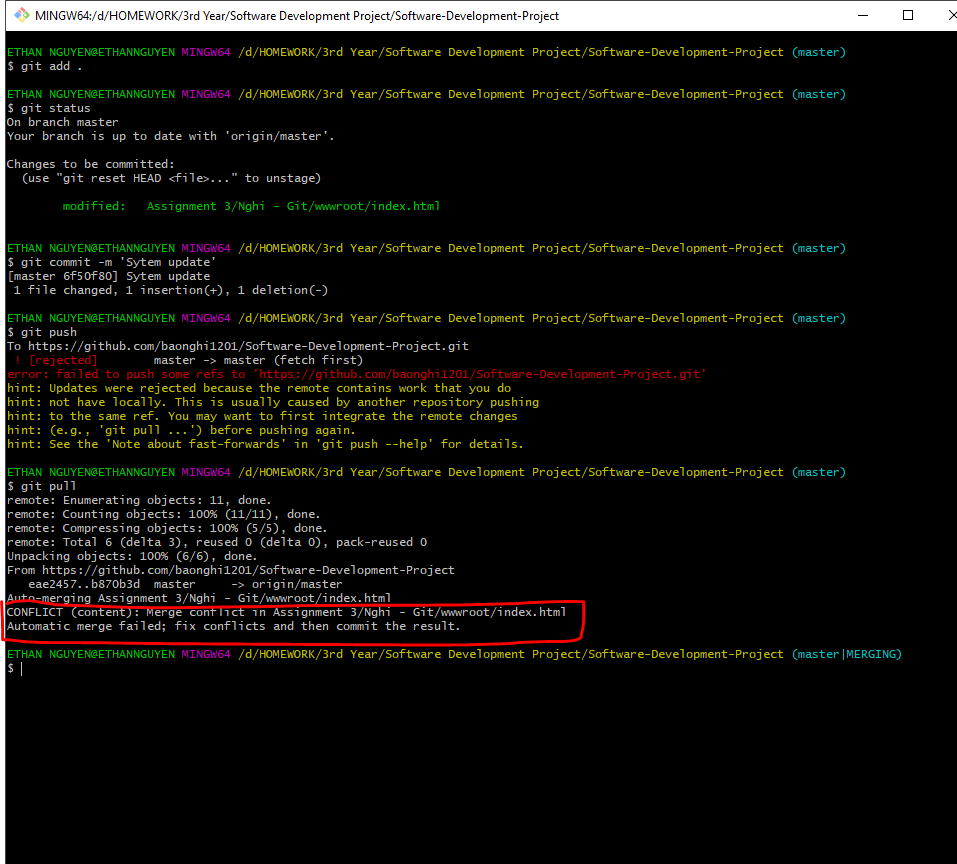
* 1. **Git Conflict**

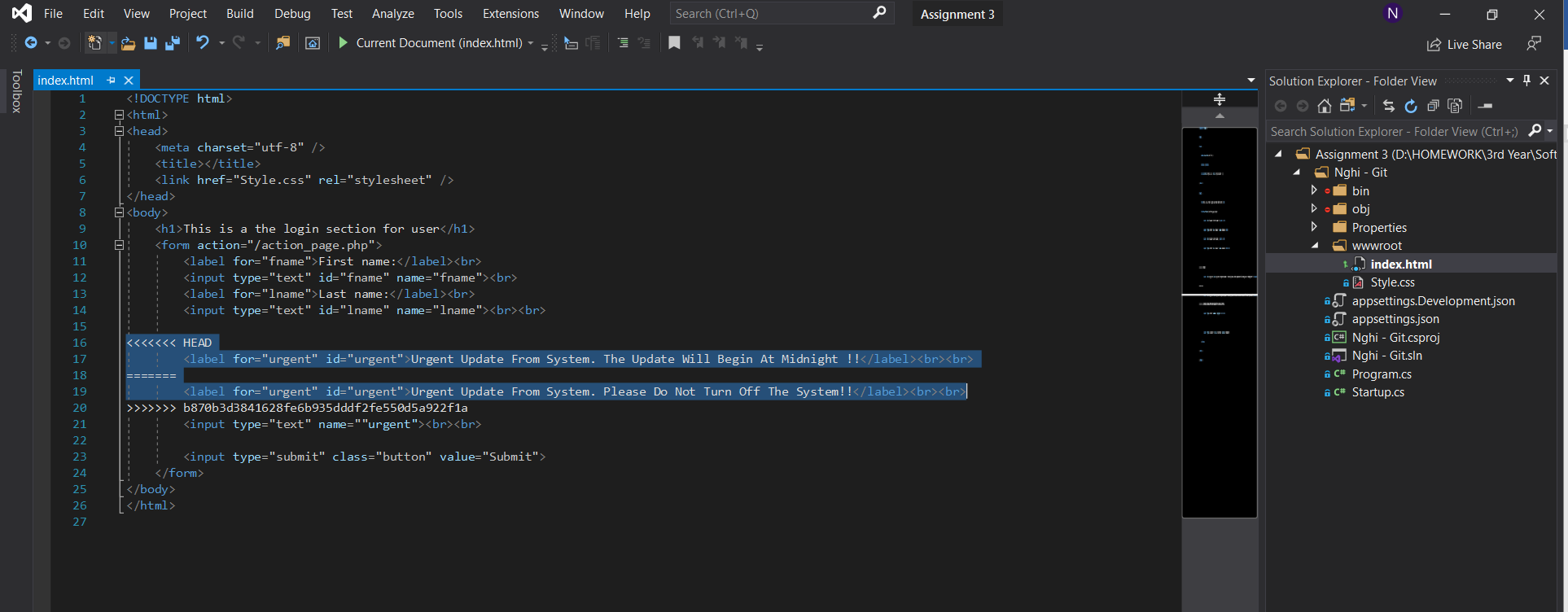
The definition of **Git Conflict** happens when the two persons, at the same modify the same file, but from different source. For instance, one person is current modifying the project on the local machine, while the other is modifying on **Git Repository**. So, when the they finished modifying their file and submitted, the conflict will instantly occur, due to the asynchronization between them. First, let us try to modify the file on the remote site and commit it. GitHub allows the user to modify their file directly on remote site for convenient:



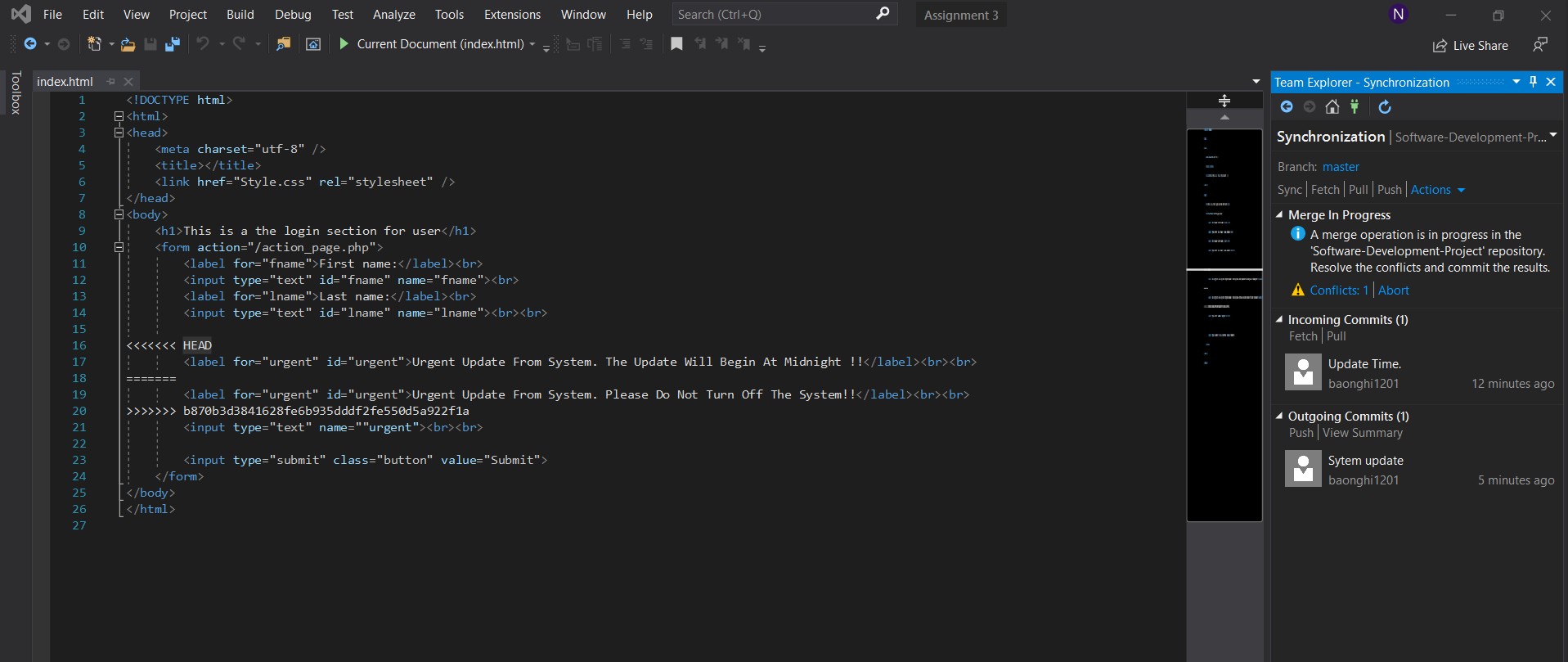


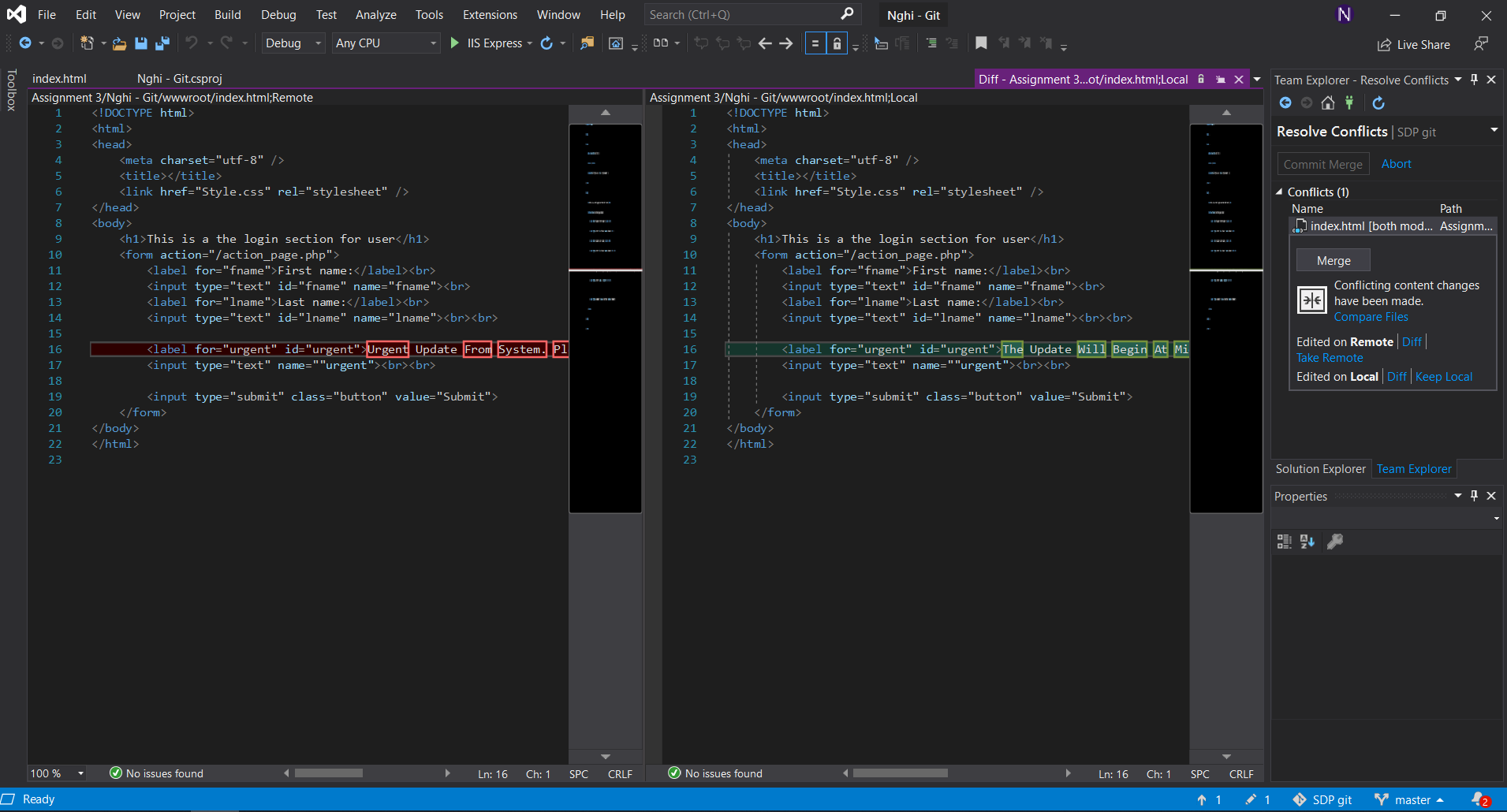
Now, turn to the local machine where the same file will be modified locally and try to commit it:

This is an example of the conflict message would appear:

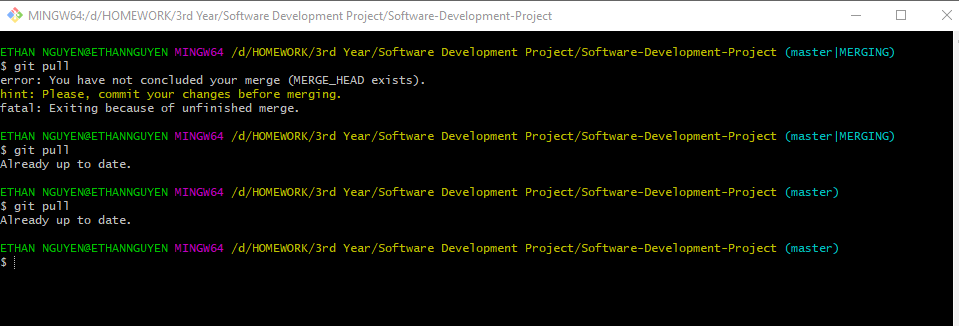
On Visual Studio 2019 we can also see the conflict message indicated:

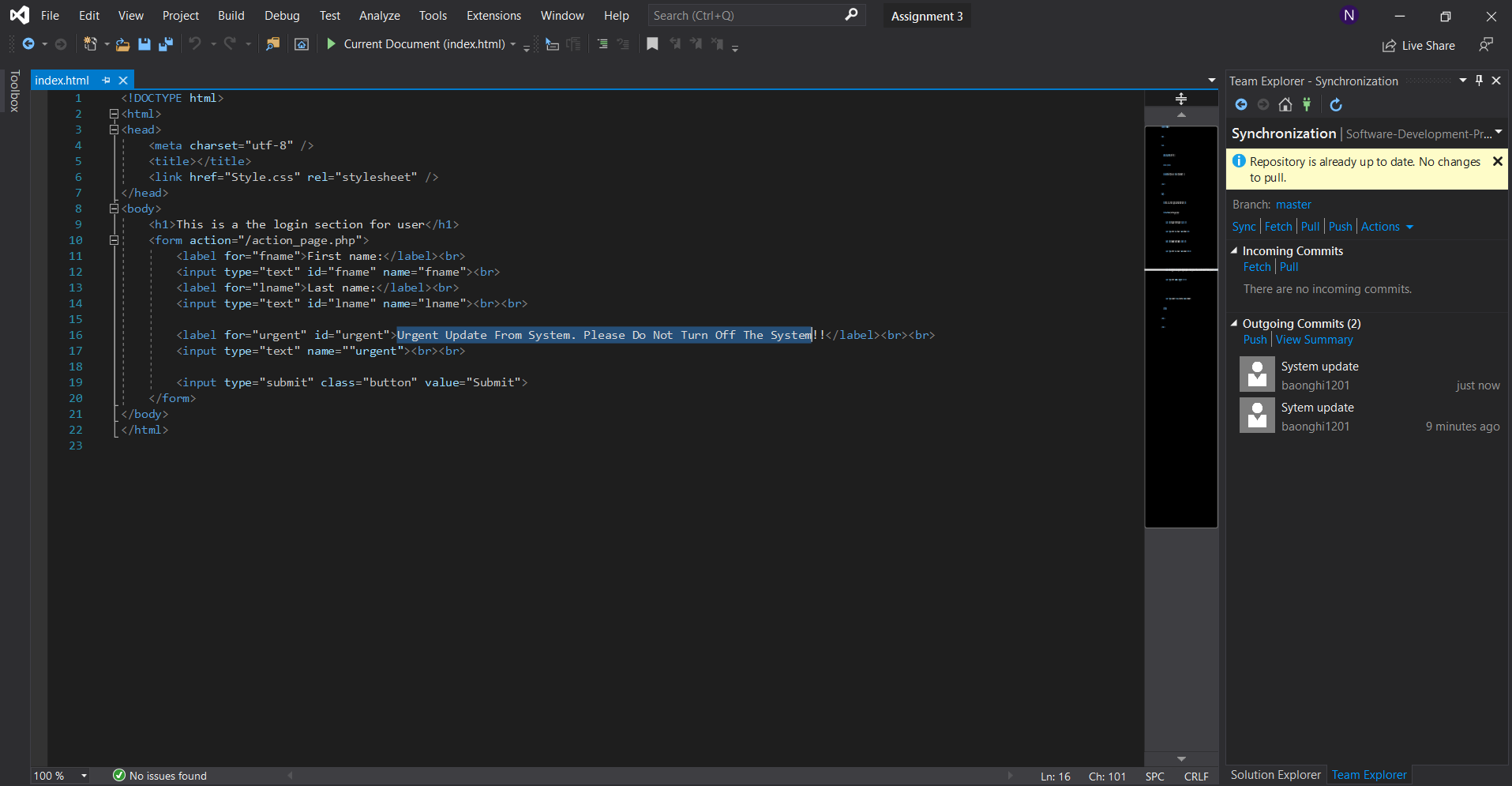
* 1. **Solve Conflict**

To solve out the conflict, the solution for solving the conflict was quite easy, the two person that specifically modify the file must decide which modified source should be applied and subtract the other. Something like this:

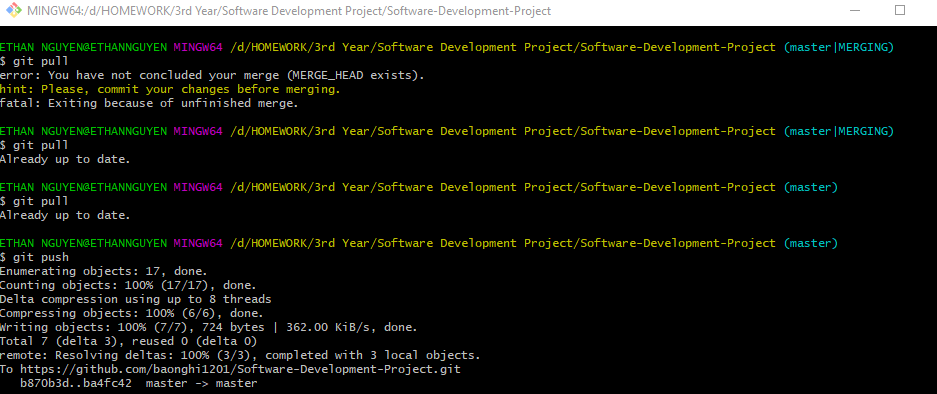


Having decided which one to take and which one to give up, so we decided to take the remote file from the GitHub the change:

The modified line in that file is now taking the change from GitHub system, instead of the local edited line:



* 1. **Synchronize with GitHub**

In doing so, we can now safely synchronize the local file to GitHub remote system.