

Clinic Management System

Team No: 04
Team Members: Rashmi Nayak S
Malavika K Nair
Nithin Peter

Git Guidelines Manual

Create branches under the main branch as per your requirements to keep the work sorted in the repository. In our project it is as follows.

1. Development/RestAPI - Backend- (Asp.Net)
2. Development/UI - frontend (angular)
3. Documents - contains documentations related to the project

Steps:

1. Initializing git repository

Step 1: create a remote repository on github.

Step 2: create a project on a local device.

Step 3: initialise project folder as git folder by using 'git init' command

Step 4: make initial commit by using following command:

- Git add . (Stage changes)
- Git commit -m "developer message"

Step 5: connect local project folder to remote git repository by following command:

- Git add remote

Eg: git add remote origin <https://github.com/team4/sample.git>

Step 6: make new branches if needed

- Git checkout -b Example: git checkout -b development/ui

Step 7: push committed changes to remote repository

- Git push -u -f Example: git push -u -f origin development/ui

Note: The contents in the local repository will now be uploaded to the remote repository from which other team members can pull and work with their respective tasks.

2. Using the existing git repository by Cloning

Step1 : use following command to clone existing remote git repository and use it in our local system.

- Git clone - -branch

Example: git clone <https://github.com/team4/sample.git> - - branch development/ui

Step 2: after making any changes and want to upload it to a remote repository use the following commands.

- Git add or git add -A
- Git commit -m "developer message"
- Git push Example: git push origin development/ui

Note: Every time when a developer/team member wants to upload the changes that they have made, one should perform a 'pull' from the remote repository.

- Git checkout Example:git checkout development/ui
- Git pull

3. Resolving Conflicts

Note: When you perform "Pull", most likely it will result in conflicts in the files so we have to check where the conflict is and resolve the conflict using three options.

They are :

- Accept incoming changes
- Accept current changes
- Accept both changes

It depends on the type of operation leading to that conflict.

In your case, a merge, where:

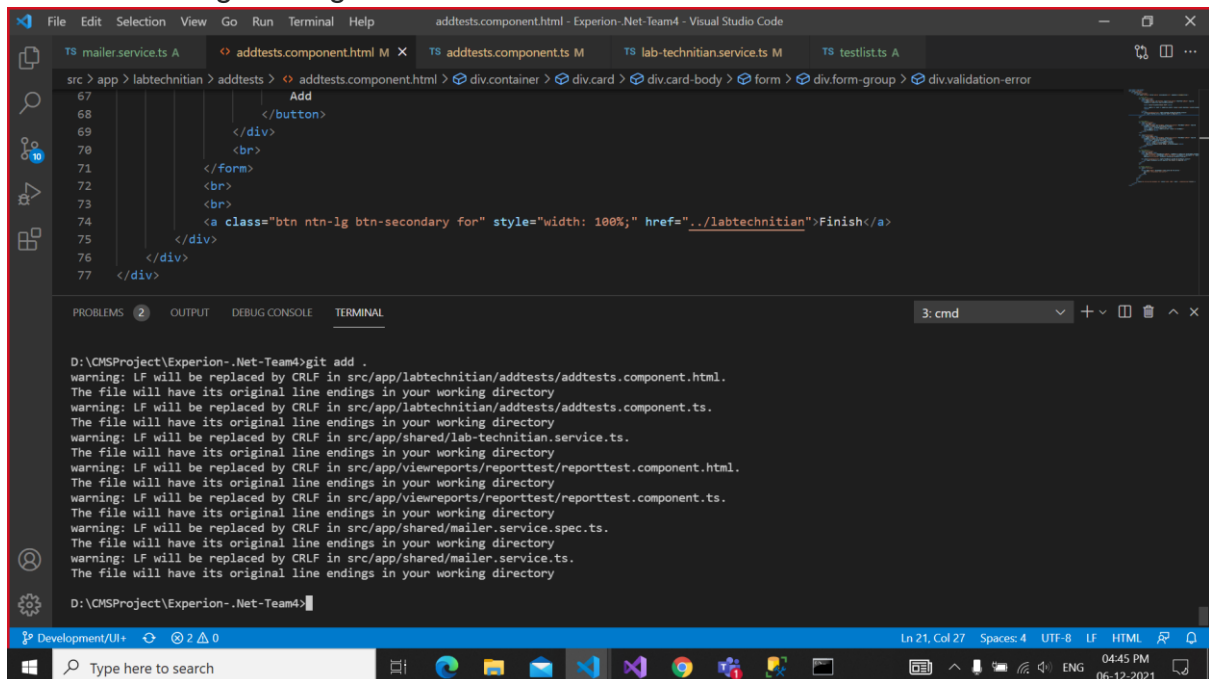
- current change represents what you have (the destination of the merge)
- incoming change represents what you merge (the source of the merge)

Then:

- Option 1 ("Accept Incoming changes") would ignore completely what you had, and keep what you merge.
- Option 2 ("Accept current changes") would ignore completely what you merge, and keep what you had.
- Option 3 ("Accepts both incoming and current changes")

After choosing the right option and resolving the conflict click on stage changes.

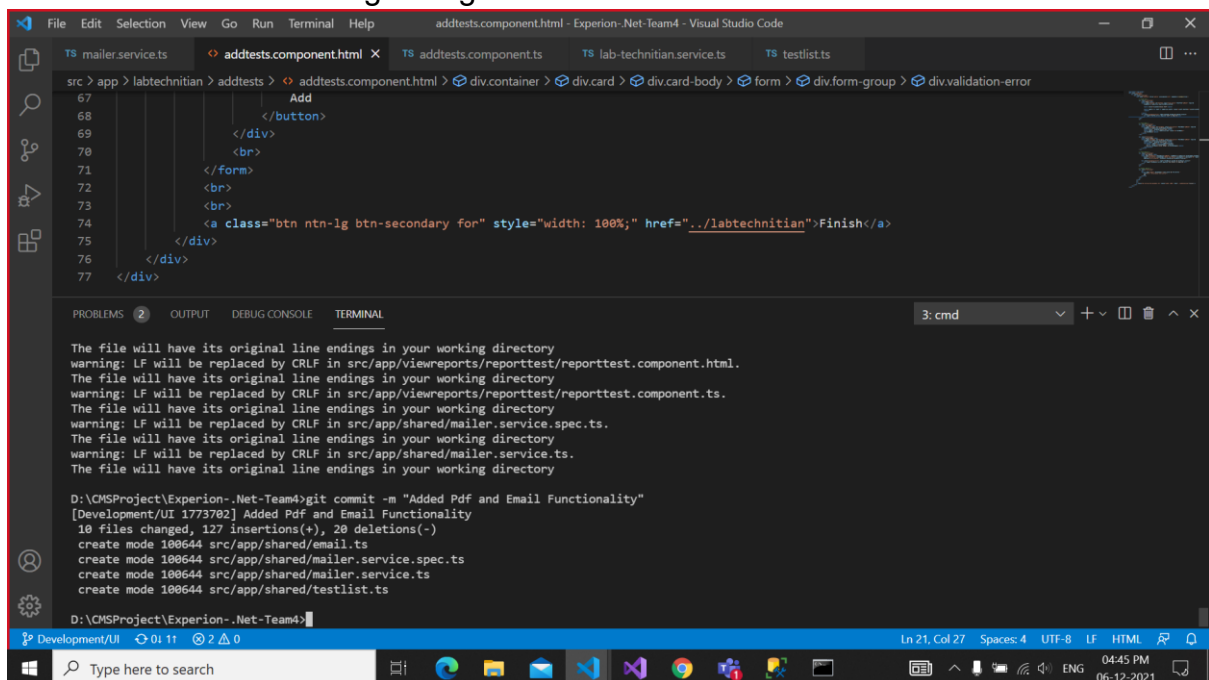
Screenshots showing some operations in git using gui and cmd.
Git add . to stage changes



The screenshot shows the Visual Studio Code interface with a file explorer on the left displaying a project structure. The main editor shows a file named 'addtests.component.html' with HTML code. The terminal at the bottom displays the output of the 'git add .' command, showing warnings about line endings (LF vs CRLF) for several files in the project.

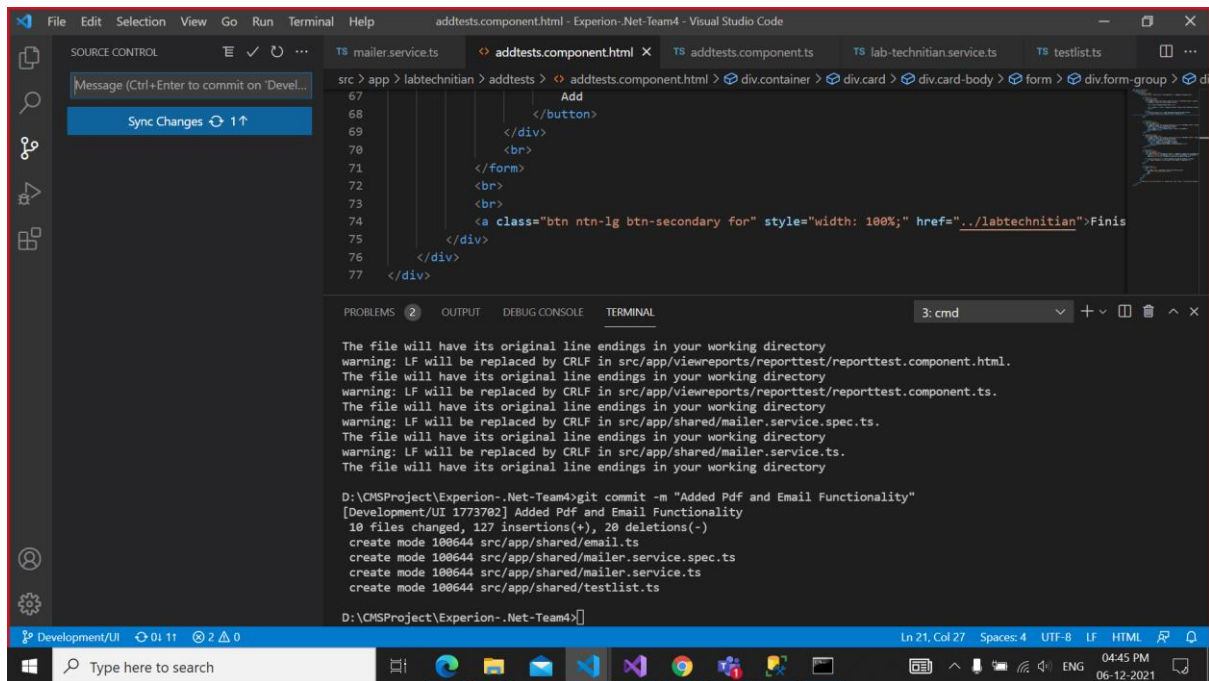
```
D:\CMSProject\Experion-.Net-Team4>git add .
warning: LF will be replaced by CRLF in src/app/labtechnitian/addtests/addtests.component.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/labtechnitian/addtests/addtests.component.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/shared/lab-technitian.service.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/viewreports/reporttest/reporttest.component.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/viewreports/reporttest/reporttest.component.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/shared/mailler.service.spec.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/shared/mailler.service.ts.
The file will have its original line endings in your working directory
D:\CMSProject\Experion-.Net-Team4>
```

Git commit -m "committing changes"

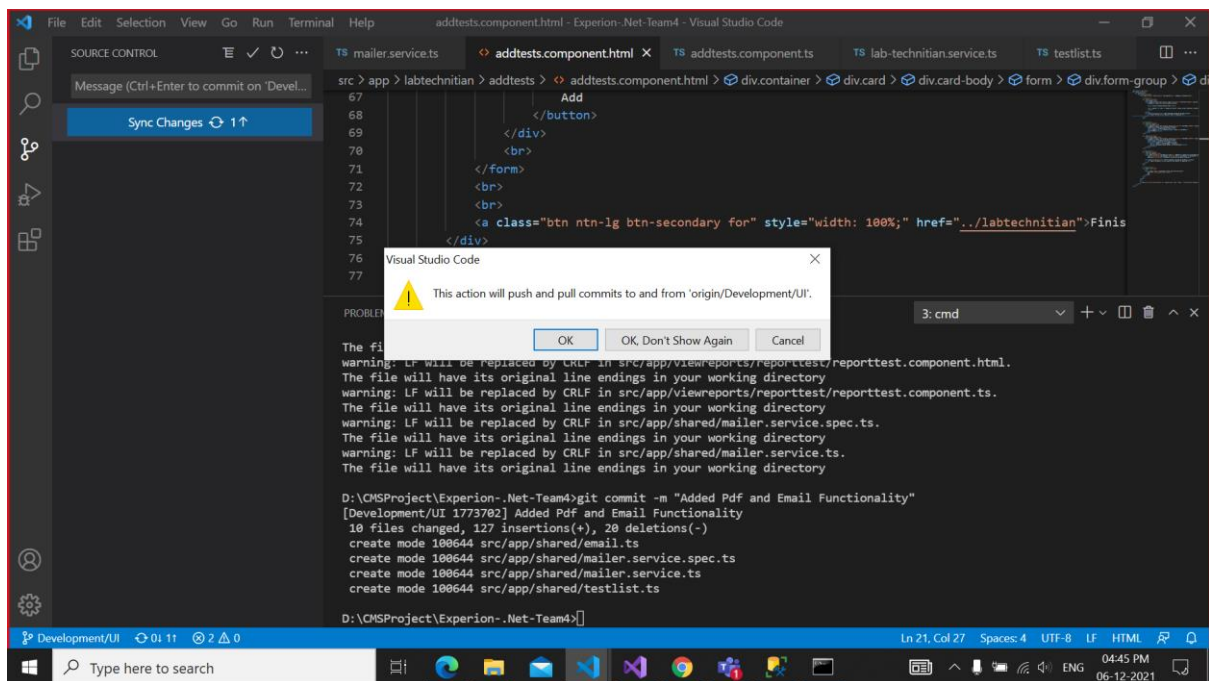


The screenshot shows the Visual Studio Code interface with the same file explorer and editor as the previous screenshot. The terminal at the bottom displays the output of the 'git commit -m "Added Pdf and Email Functionality"' command, showing the commit message and the files that were committed.

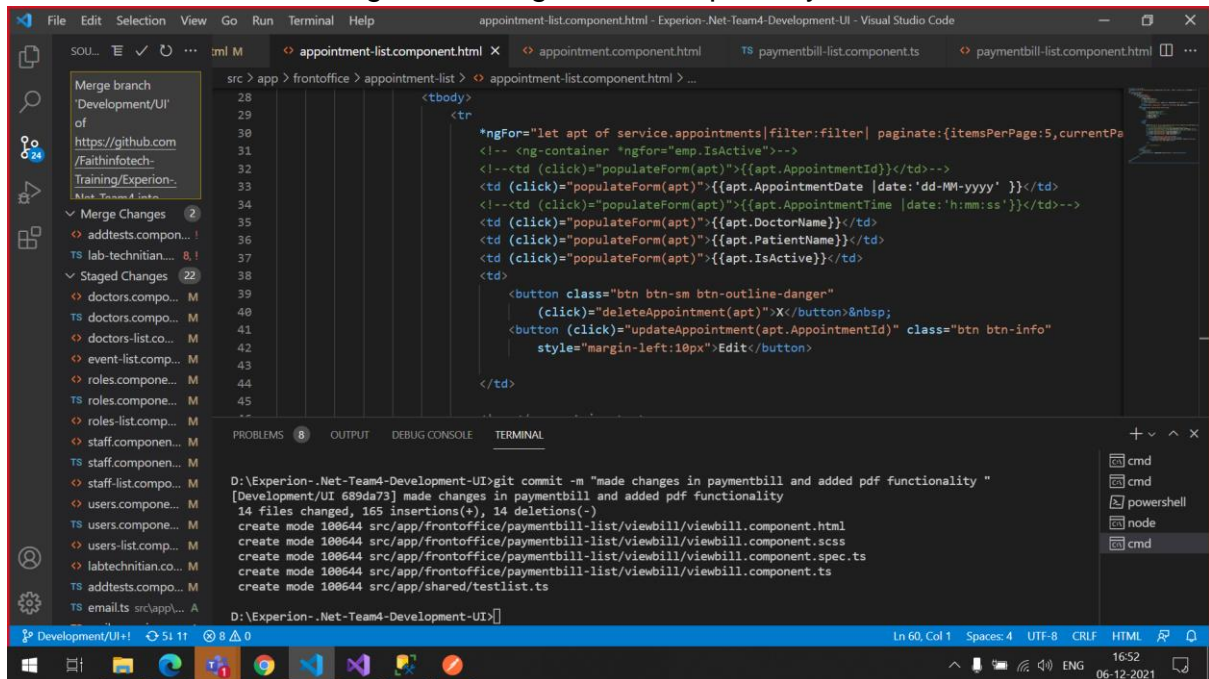
```
D:\CMSProject\Experion-.Net-Team4>git commit -m "Added Pdf and Email Functionality"
[Development/UI 1773702] Added Pdf and Email Functionality
10 files changed, 127 insertions(+), 20 deletions(-)
create mode 100644 src/app/shared/email.ts
create mode 100644 src/app/shared/mailler.service.spec.ts
create mode 100644 src/app/shared/mailler.service.ts
create mode 100644 src/app/shared/testlist.ts
D:\CMSProject\Experion-.Net-Team4>
```



After committing changes click sync the changes. This action will push and pull commits from the branch.

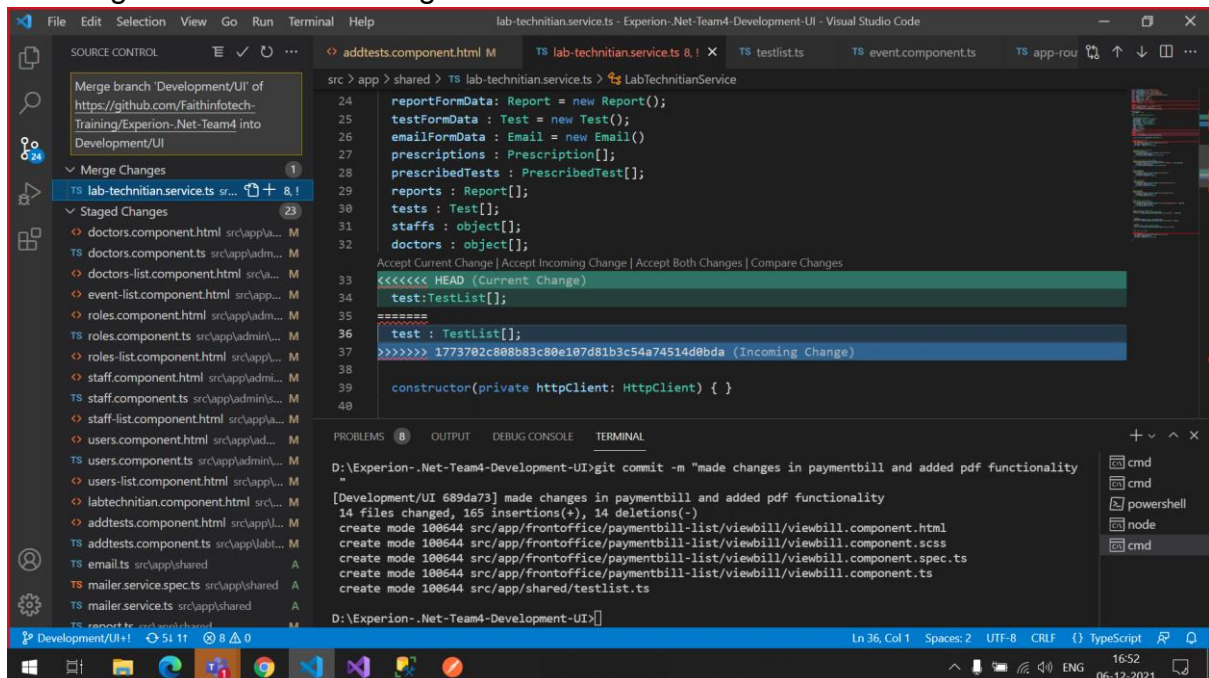


Another author committing their changes to the repository



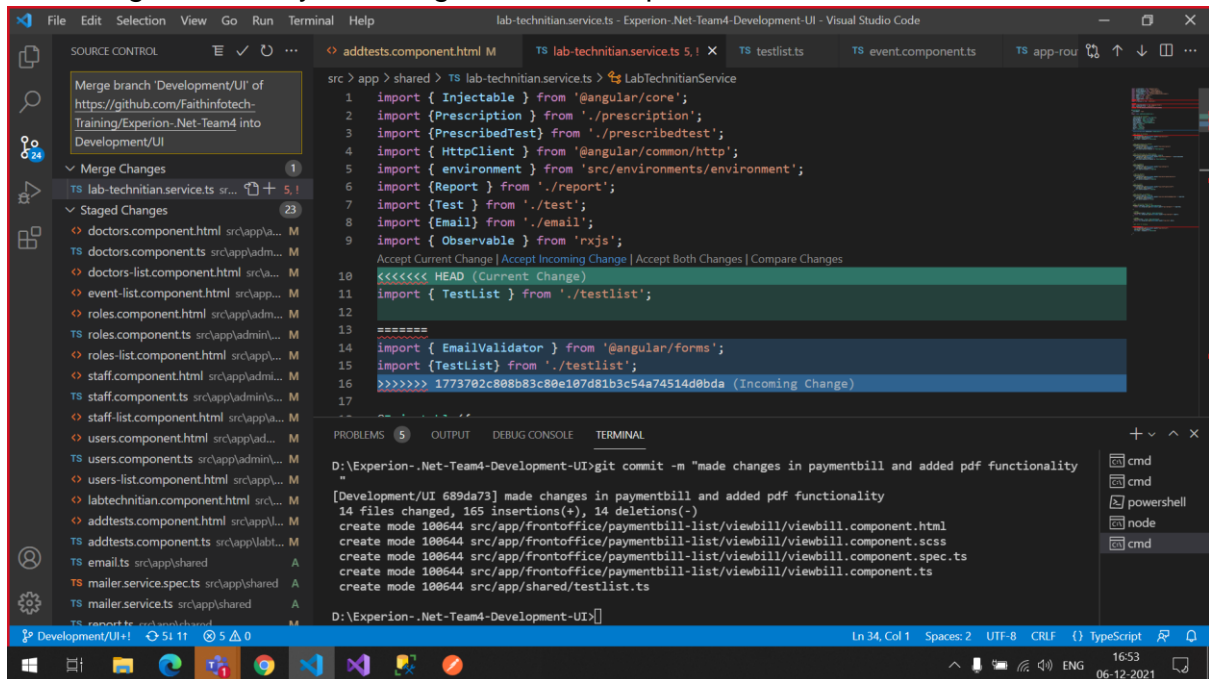
The screenshot shows the Visual Studio Code interface with the 'appointment-list.component.html' file open. The left sidebar displays the 'Source Control' view, showing a list of files and folders. The main editor area shows the code for 'appointment-list.component.html'. The bottom status bar indicates the file is at line 60, column 1, with 4 spaces, UTF-8 encoding, and CRLF line endings. The terminal at the bottom shows the command 'D:\Experi...-Net-Team4-Development-UI>git commit -m "made changes in paymentbill and added pdf functionality"' and the output of the commit process.

Showing the conflicts occurring.



The screenshot shows the Visual Studio Code interface with the 'lab-technician.service.ts' file open. The left sidebar displays the 'Source Control' view, showing a list of files and folders. The main editor area shows the code for 'lab-technician.service.ts'. The bottom status bar indicates the file is at line 36, column 1, with 2 spaces, UTF-8 encoding, and CRLF line endings. The terminal at the bottom shows the command 'D:\Experi...-Net-Team4-Development-UI>git commit -m "made changes in paymentbill and added pdf functionality"' and the output of the commit process. A conflict is visible in the code, with a green bar indicating the 'HEAD (Current Change)' and a blue bar indicating the 'Incoming Change'.

Resolving conflicts by choosing the desired options.



After resolving conflicts, commit--->commit staged

