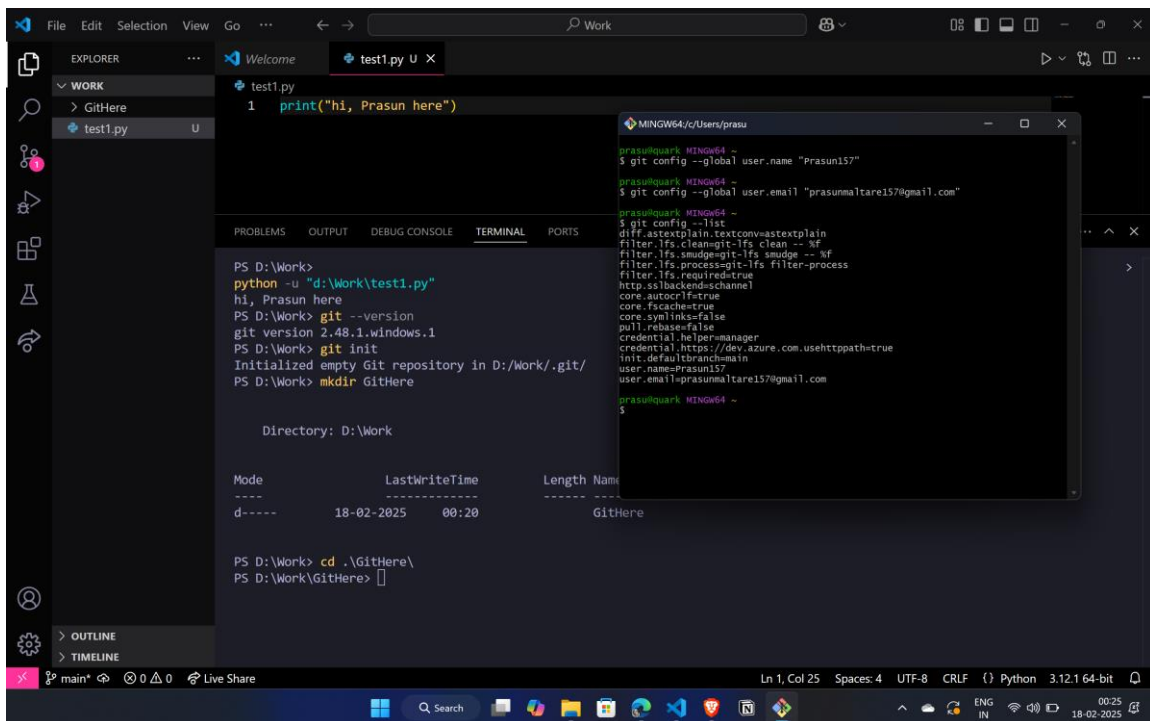


GIT Assignment-1

Objective: The goal of this assignment is to help you get familiar with basic Git commands and workflows, which are commonly used in real-world software development

GitHub Repository
[here](#)

Create a New Repository



The screenshot shows the Visual Studio Code interface with a file explorer on the left, a code editor in the center, and a terminal window at the bottom. The file explorer shows a project named 'WORK' with a subdirectory 'GitHere' and a file 'test1.py'. The code editor shows the content of 'test1.py', which is a single line of Python code: `print("hi, Prasun here")`. The terminal window shows the following commands and output:

```
PS D:\Work> python -u "d:\Work\test1.py"
hi, Prasun here
PS D:\Work> git --version
git version 2.48.1.windows.1
PS D:\Work> git init
Initialized empty Git repository in D:\Work\.git\
PS D:\Work> mkdir GitHere

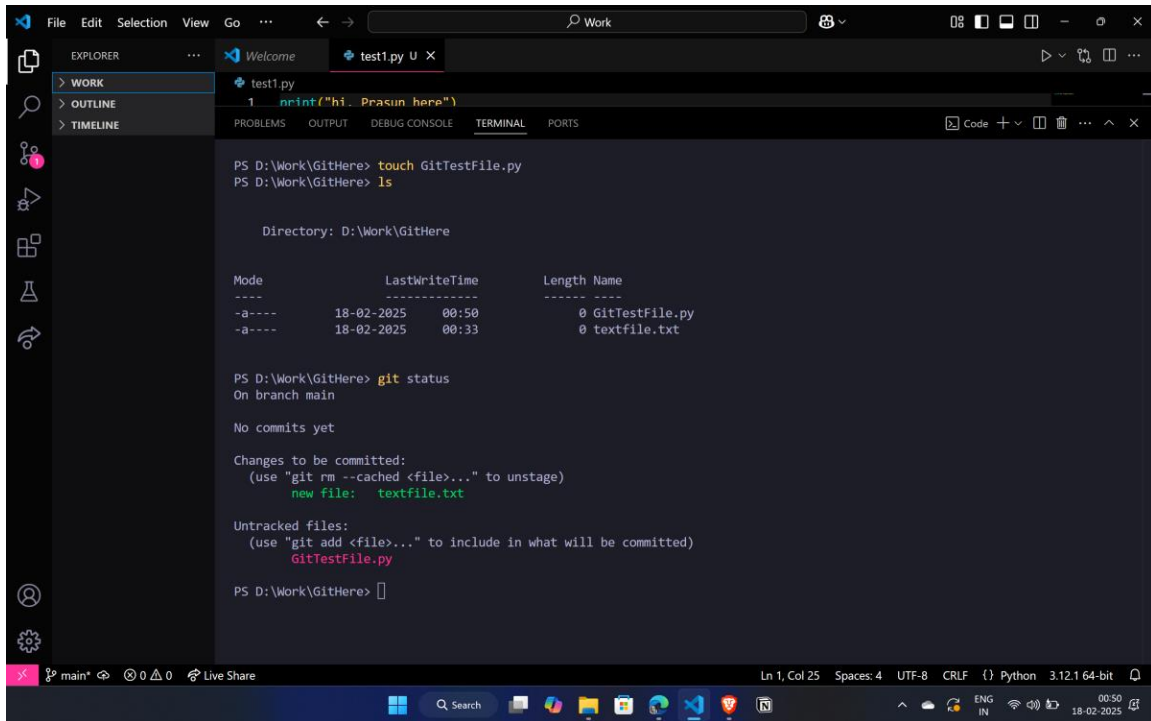
Directory: D:\Work

Mode                LastWriteTime         Length Name
----                -
d-----          18-02-2025     00:20         GitHere

PS D:\Work> cd .\GitHere\
PS D:\Work\GitHere>
```

The terminal window also shows the output of the `git config --list` command, which lists various configuration settings for the user 'Prasun157' and the email 'prasunmaltare157@gmail.com'.

Create a New File



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
PS D:\Work\GitHere> touch GitTestFile.py
PS D:\Work\GitHere> ls
```

Directory: D:\Work\GitHere

Mode	LastWriteTime	Length	Name
-a----	18-02-2025 00:50	0	GitTestFile.py
-a----	18-02-2025 00:33	0	textfile.txt

```
PS D:\Work\GitHere> git status
On branch main

No commits yet

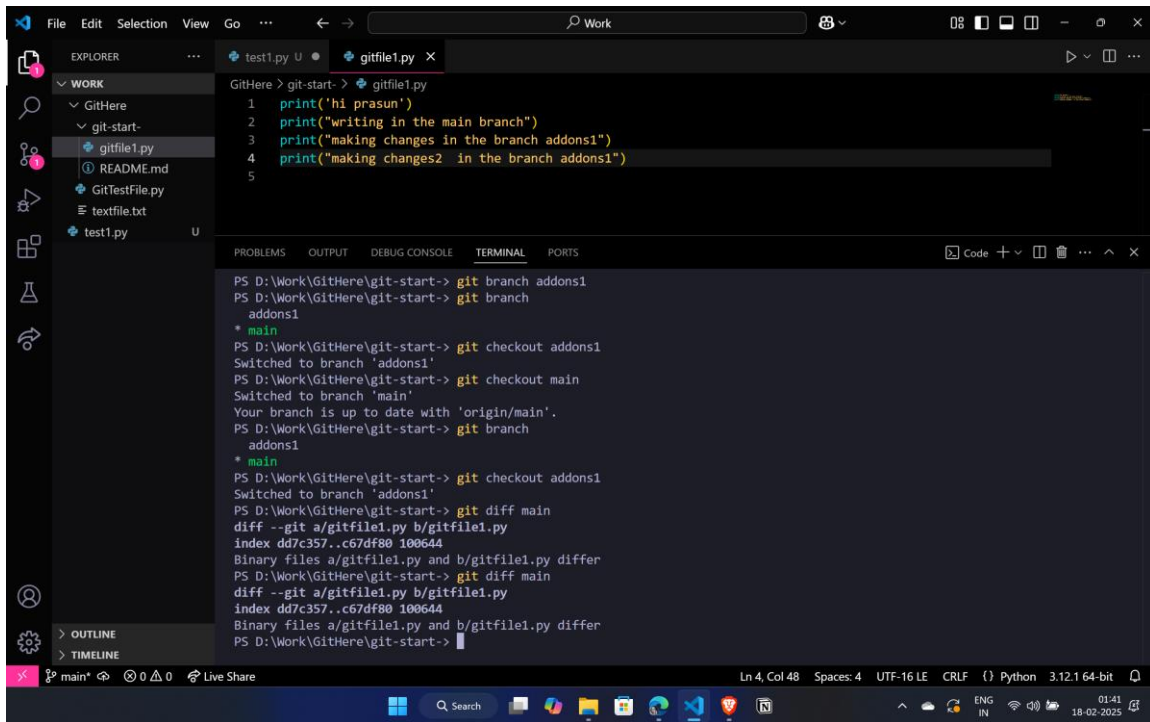
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   textfile.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        GitTestFile.py

PS D:\Work\GitHere>
```

The terminal window is titled "test1.py U X". The Explorer sidebar on the left shows the file structure with "WORK" selected. The status bar at the bottom indicates the current file is "main", the encoding is "UTF-8", and the language is "Python".

Create a Branch and Make Changes in the New Branch

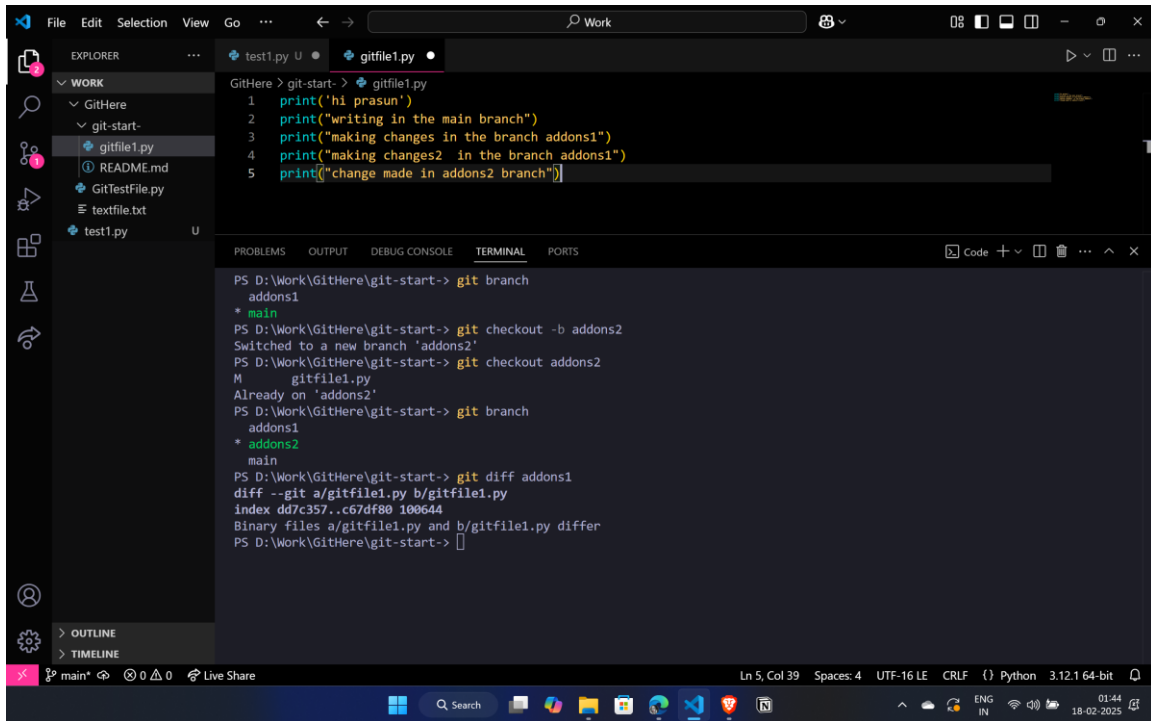


The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the file structure of a project named 'GitHere'. The file 'gitfile1.py' is selected. The main editor area shows the content of 'gitfile1.py', which contains five lines of Python code. The bottom panel shows the TERMINAL tab with the following commands and output:

```
PS D:\Work\GitHere\git-start-> git branch addons1
PS D:\Work\GitHere\git-start-> git branch
addons1
* main
PS D:\Work\GitHere\git-start-> git checkout addons1
Switched to branch 'addons1'
PS D:\Work\GitHere\git-start-> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS D:\Work\GitHere\git-start-> git branch
addons1
* main
PS D:\Work\GitHere\git-start-> git checkout addons1
Switched to branch 'addons1'
PS D:\Work\GitHere\git-start-> git diff main
diff --git a/gitfile1.py b/gitfile1.py
index dd7c357..c67df80 100644
Binary files a/gitfile1.py and b/gitfile1.py differ
PS D:\Work\GitHere\git-start-> git diff main
diff --git a/gitfile1.py b/gitfile1.py
index dd7c357..c67df80 100644
Binary files a/gitfile1.py and b/gitfile1.py differ
PS D:\Work\GitHere\git-start->
```

The status bar at the bottom indicates the current branch is 'main' and the file encoding is UTF-16 LE.

Create Another Branch (Optional)



The screenshot shows the Visual Studio Code interface with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'GitHere' with files 'gitfile1.py', 'README.md', 'GitTestFile.py', 'textfile.txt', and 'test1.py'. The terminal shows the following commands and output:

```
GitHere > git-start- > gitfile1.py
1 print('hi prasun')
2 print("writing in the main branch")
3 print("making changes in the branch addons1")
4 print("making changes2 in the branch addons1")
5 print("change made in addons2 branch")

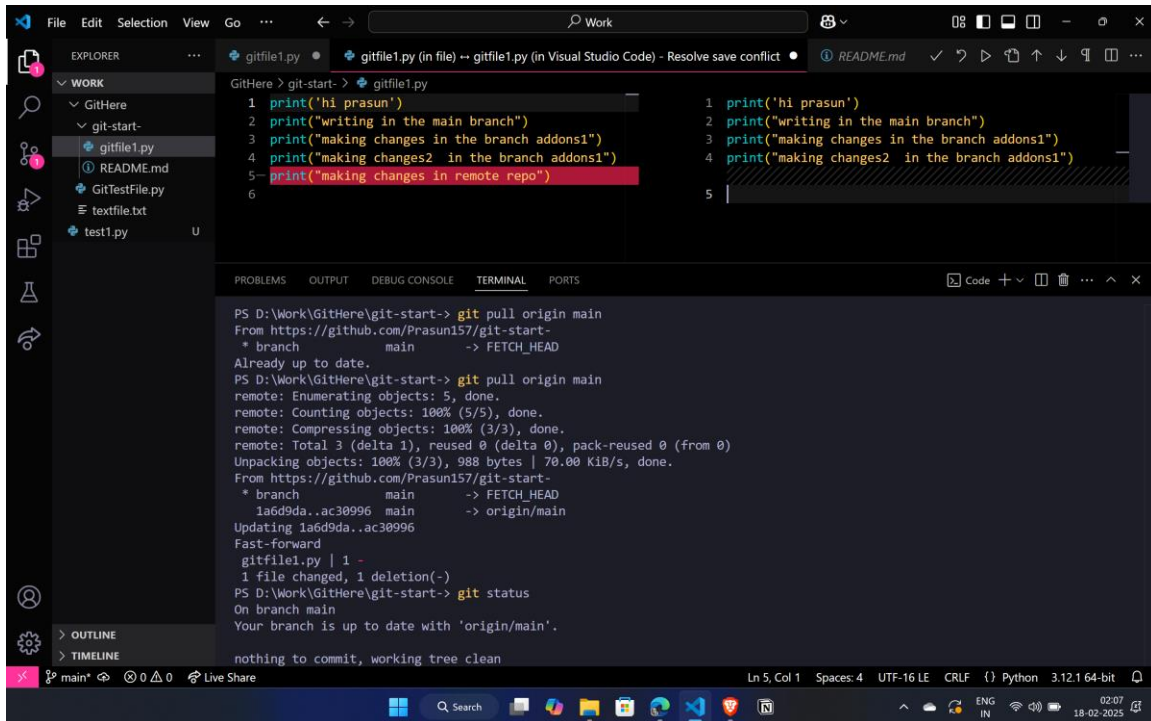
PS D:\Work\GitHere\git-start-> git branch
addons1
* main
PS D:\Work\GitHere\git-start-> git checkout -b addons2
Switched to a new branch 'addons2'
PS D:\Work\GitHere\git-start-> git checkout addons2
M
gitfile1.py
Already on 'addons2'
PS D:\Work\GitHere\git-start-> git branch
addons1
* addons2
main
PS D:\Work\GitHere\git-start-> git diff addons1
diff --git a/gitfile1.py b/gitfile1.py
index dd7c357..c67df80 100644
Binary files a/gitfile1.py and b/gitfile1.py differ
PS D:\Work\GitHere\git-start->
```

The status bar at the bottom indicates the current branch is 'main', the file encoding is UTF-16 LE, and the Python version is 3.12.1 64-bit.

Merge Branches

```
File Edit Selection View Go ... Work
EXPLORER
WORK
  GitHere
  git-start
  gitfile1.py
  README.md
  GitTestFile.py
  textfile.txt
  test1.py
gitfile1.py
1 print('hi prasun')
2 print("writing in the main branch")
3 print("making changes in the branch addons1")
4 print("making changes2 in the branch addons1")
5 print("change made in addons2 branch")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Work\GitHere\git-start-> git branch
* addons1
  addons2
  main
PS D:\Work\GitHere\git-start-> git checkout addons2
M
  gitfile1.py
Switched to branch 'addons2'
PS D:\Work\GitHere\git-start-> git merge addons1
Already up to date.
PS D:\Work\GitHere\git-start-> git merge main
Already up to date.
PS D:\Work\GitHere\git-start->
Ln 5, Col 39 Spaces: 4 UTF-16 LE CRLF () Python 3.12.1 64-bit
main* 0 0 Live Share
```

Pull Changes from Remote



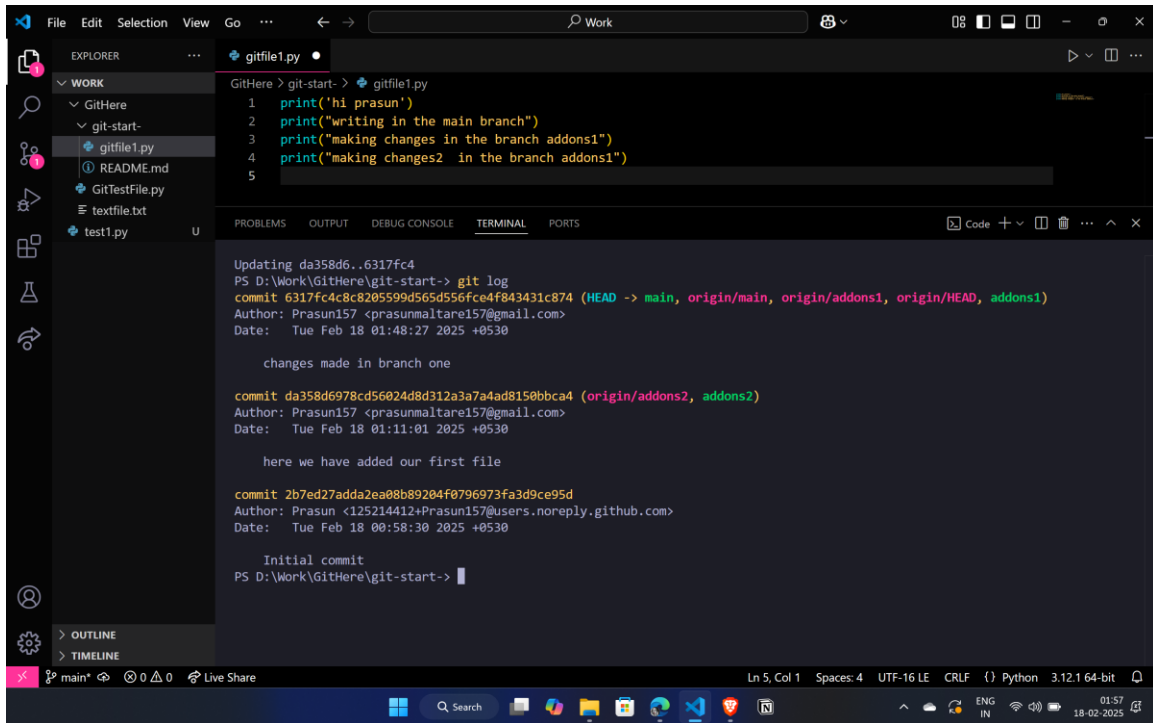
```
gitfile1.py
1 print('hi prasun')
2 print("writing in the main branch")
3 print("making changes in the branch addons1")
4 print("making changes2 in the branch addons1")
5 print("making changes in remote repo")
6

gitfile1.py (in file) ↔ gitfile1.py (in Visual Studio Code) - Resolve save conflict
1 print('hi prasun')
2 print("writing in the main branch")
3 print("making changes in the branch addons1")
4 print("making changes2 in the branch addons1")
5

PS D:\Work\Githere\git-start-> git pull origin main
From https://github.com/Prasun157/git-start-
 * branch      main       -> FETCH_HEAD
Already up to date.
PS D:\Work\Githere\git-start-> git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 988 bytes | 70.00 KiB/s, done.
From https://github.com/Prasun157/git-start-
 * branch      main       -> FETCH_HEAD
   1a6d9da..ac30996  main    -> origin/main
Updating 1a6d9da..ac30996
Fast-forward
 gitfile1.py | 1 -
 1 file changed, 1 deletion(-)
PS D:\Work\Githere\git-start-> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

View Git Log and History



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying a project structure under 'WORK' including 'GitHere', 'git-start', 'gitfile1.py', 'README.md', 'GitTestFile.py', 'testfile.txt', and 'test1.py'. The main editor area shows the content of 'gitfile1.py' with five lines of Python code. The bottom panel is split into 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS' tabs, with the 'TERMINAL' tab active. The terminal displays the output of a 'git log' command, showing the commit history. The status bar at the bottom indicates the current branch is 'main' and the file encoding is 'UTF-16 LE'.

```
GitHere > git-start- > gitfile1.py
1 print('hi prasun')
2 print("writing in the main branch")
3 print("making changes in the branch addons1")
4 print("making changes2 in the branch addons1")
5

Updating da358d6..6317fc4
PS D:\Work\GitHere\git-start-> git log
commit 6317fc4c8c8205599d565d556fce4f843431c874 (HEAD -> main, origin/main, origin/addons1, origin/HEAD, addons1)
Author: Prasun157 <prasunmaltare157@gmail.com>
Date: Tue Feb 18 01:48:27 2025 +0530

    changes made in branch one

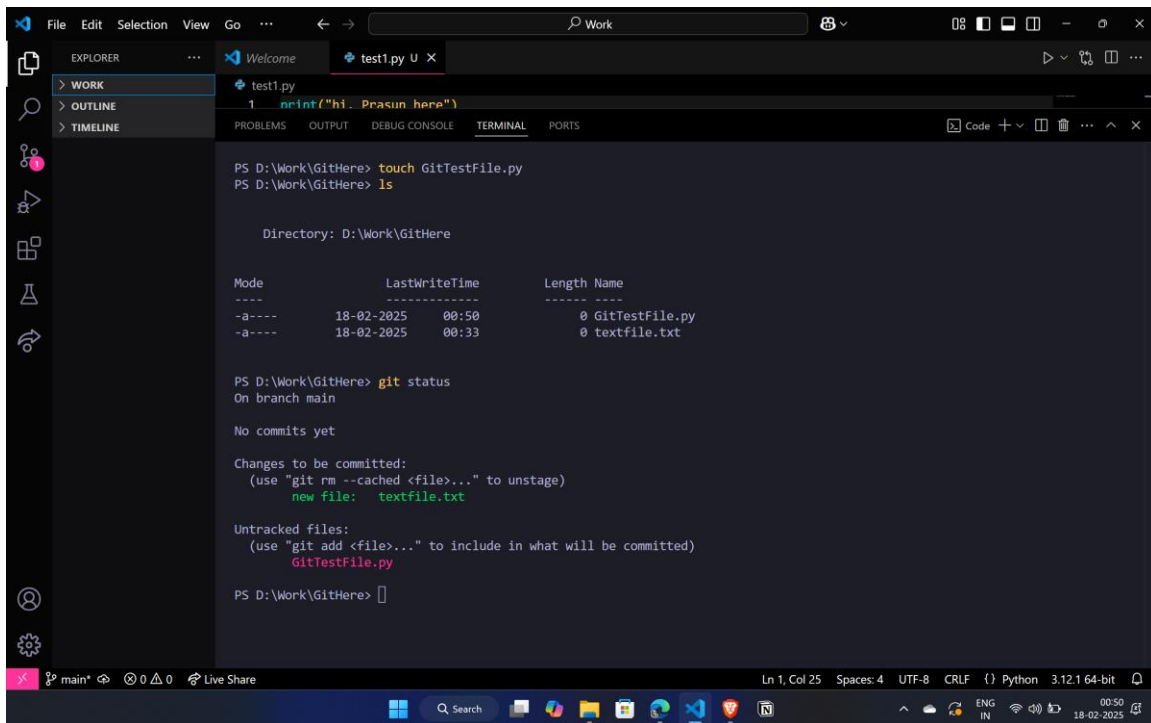
commit da358d6978cd56024d8d312a3a7a4ad8150bbca4 (origin/addons2, addons2)
Author: Prasun157 <prasunmaltare157@gmail.com>
Date: Tue Feb 18 01:11:01 2025 +0530

    here we have added our first file

commit 2b7ed27adda2ea08b89204f0796973fa3d9ce95d
Author: Prasun <125214412@Prasun157@users.noreply.github.com>
Date: Tue Feb 18 00:58:30 2025 +0530

    Initial commit
PS D:\Work\GitHere\git-start->
```

Git Status and Staging



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
PS D:\Work\GitHere> touch GitTestFile.py
PS D:\Work\GitHere> ls

Directory: D:\Work\GitHere

Mode                LastWriteTime         Length Name
----                -
-a----             18-02-2025   00:50             0 GitTestFile.py
-a----             18-02-2025   00:33             0 textfile.txt

PS D:\Work\GitHere> git status
On branch main

No commits yet

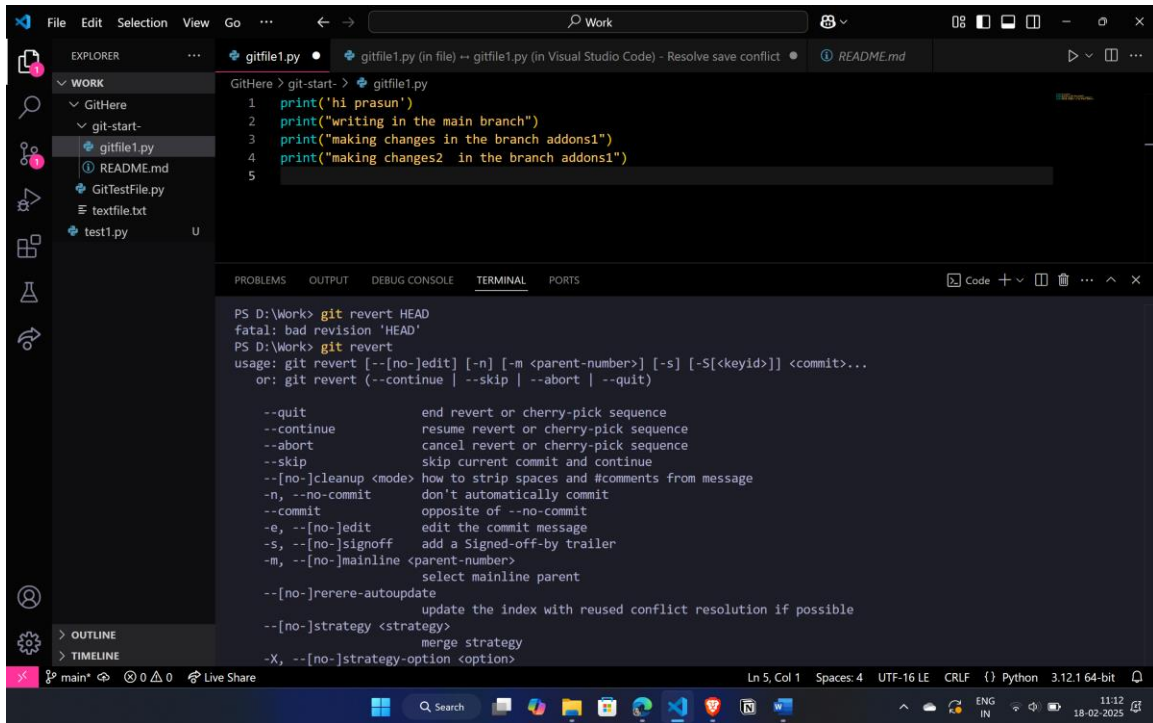
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   textfile.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        GitTestFile.py

PS D:\Work\GitHere>
```

The terminal output shows the directory listing and the results of the `git status` command. The status indicates that there are no commits yet on the `main` branch. There are changes to be committed, including a new file `textfile.txt`. There are also untracked files, including `GitTestFile.py`.

Revert and Reset



The screenshot shows the Visual Studio Code interface with a terminal window open. The Explorer pane on the left shows a project structure with files like README.md, GitTestFile.py, textfile.txt, and test1.py. The main editor area shows a Python file named gitfile1.py with five lines of code. The terminal window displays the output of the command `git revert HEAD`, which results in a fatal error: `fatal: bad revision 'HEAD'`. Below the error, the terminal shows the usage of the `git revert` command, including options like `--quit`, `--continue`, `--abort`, `--skip`, `--[no-]cleanup`, `-n`, `--no-commit`, `--commit`, `-e`, `--[no-]edit`, `-s`, `--[no-]signoff`, `-m`, `--[no-]mainline`, `--[no-]rerere-autoupdate`, `--[no-]strategy`, and `-X`.

```
GitHere > git-start- > gitfile1.py
1 print('hi prasun')
2 print("writing in the main branch")
3 print("making changes in the branch addons1")
4 print("making changes2 in the branch addons1")
5

PS D:\Work> git revert HEAD
fatal: bad revision 'HEAD'
PS D:\Work> git revert
usage: git revert [--[no-]edit] [-n] [-m <parent-number>] [--s] [-S[<keyid>]] <commit>...
or: git revert [--continue | --skip | --abort | --quit)

--quit          end revert or cherry-pick sequence
--continue      resume revert or cherry-pick sequence
--abort         cancel revert or cherry-pick sequence
--skip         skip current commit and continue
--[no-]cleanup <mode> how to strip spaces and #comments from message
-n, --no-commit don't automatically commit
--commit       opposite of --no-commit
-e, --[no-]edit edit the commit message
-s, --[no-]signoff add a Signed-off-by trailer
-m, --[no-]mainline <parent-number>
                select mainline parent
--[no-]rerere-autoupdate
                update the index with reused conflict resolution if possible
--[no-]strategy <strategy>
                merge strategy
-X, --[no-]strategy-option <option>
```