GITHUB ASSIGNMENT - 5

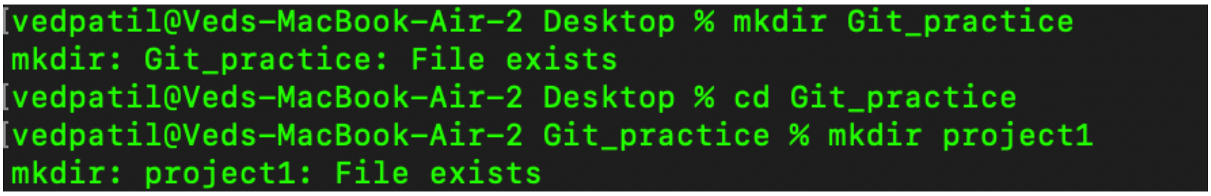
NAME - VED PATIL

ROLL NO -

BATCH - LARRY PAGE

DATE - 11 / 11 / 25

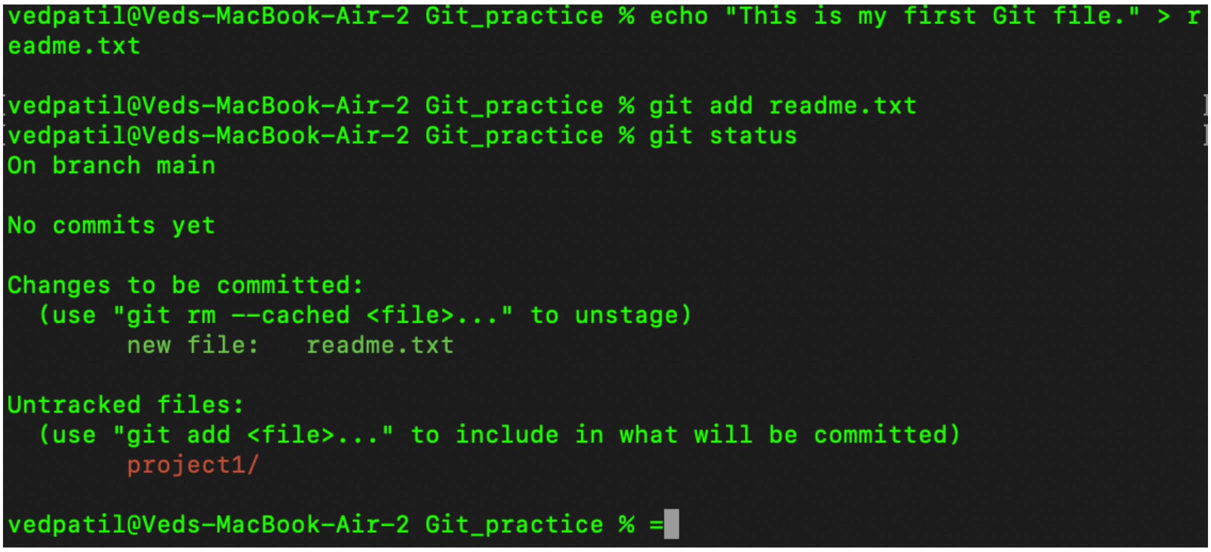
Task 1: Initialize Your First Git Repository



```
[vedpatil@Veds-MacBook-Air-2 Desktop % mkdir Git_practice
mkdir: Git_practice: File exists
[vedpatil@Veds-MacBook-Air-2 Desktop % cd Git_practice
[vedpatil@Veds-MacBook-Air-2 Git_practice % mkdir project1
mkdir: project1: File exists
```

1. Create a new folder “project1” and navigate into it.
2. Run “git init” to initialize a git repository.
3. Verify “.git” folder is created and terminal shows “Initialized empty Git repository.

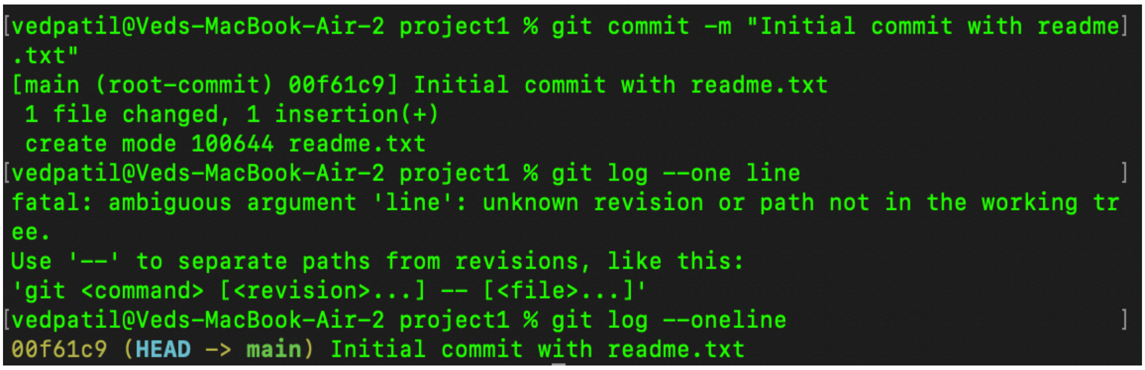
Task 2: Create and Stage Your First File



```
vedpatil@Veds-MacBook-Air-2 Git_practice % echo "This is my first Git file." > r
eadme.txt

[vedpatil@Veds-MacBook-Air-2 Git_practice % git add readme.txt                   ]
[vedpatil@Veds-MacBook-Air-2 Git_practice % git status                           ]
On branch main

No commits yet

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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        project1/

vedpatil@Veds-MacBook-Air-2 Git_practice % =
```

1. Make a file called “readme.txt and write something in it.
2. Use “git add readme.txt to stage the file.
3. Check that the file is ready to be committed with no errors.

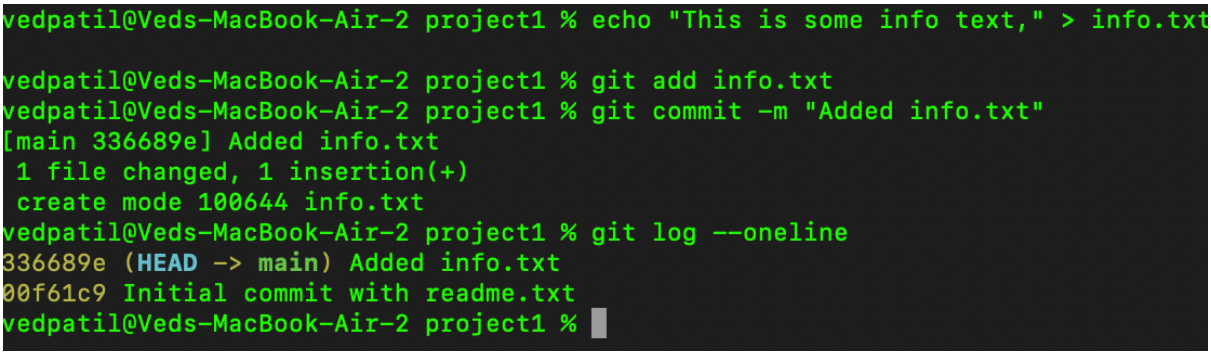
Task 3: Make the First Commit



```
[vedpatil@Veds-MacBook-Air-2 project1 % git commit -m "Initial commit with readme]
.txt"
[main (root-commit) 00f61c9] Initial commit with readme.txt
 1 file changed, 1 insertion(+)
 create mode 100644 readme.txt
[vedpatil@Veds-MacBook-Air-2 project1 % git log --one line                       ]
fatal: ambiguous argument 'line': unknown revision or path not in the working tr
ee.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
[vedpatil@Veds-MacBook-Air-2 project1 % git log --oneline                        ]
00f61c9 (HEAD -> main) Initial commit with readme.txt
```

1. Initialize the Git repository using “git init”.
2. Stage the file with “git add readme.txt”.
3. Commit the file using “git commit -m "Initial commit with readme.txt" and verify with “git log --oneline”.

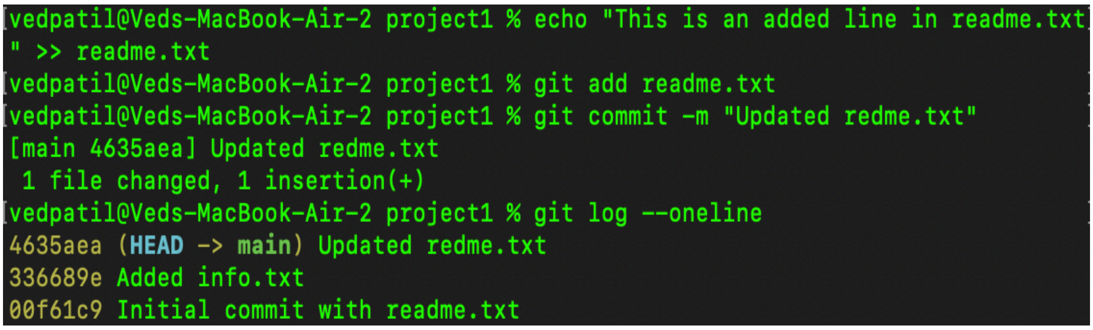
Task 4: Add Another File to the Project



```
vedpatil@Veds-MacBook-Air-2 project1 % echo "This is some info text," > info.txt

vedpatil@Veds-MacBook-Air-2 project1 % git add info.txt
vedpatil@Veds-MacBook-Air-2 project1 % git commit -m "Added info.txt"
[main 336689e] Added info.txt
 1 file changed, 1 insertion(+)
 create mode 100644 info.txt
vedpatil@Veds-MacBook-Air-2 project1 % git log --oneline
336689e (HEAD -> main) Added info.txt
00f61c9 Initial commit with readme.txt
vedpatil@Veds-MacBook-Air-2 project1 %
```

1. Create a new file “info.txt” and add some text to it.
2. Stage the file using “git add info.txt”.
3. Commit the file with “git commit -m “Added info.txt” and verify using “git log –oneline”.

Task 5: Modify and Commit Changes

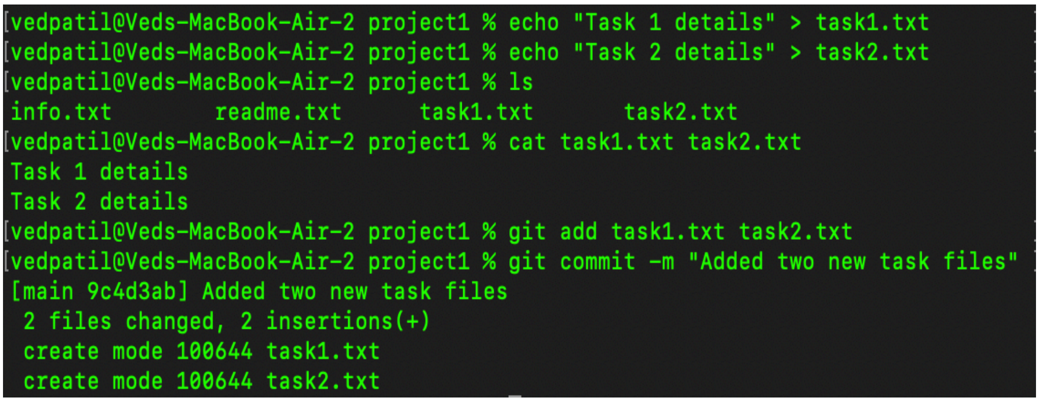


```
[vedpatil@Veds-MacBook-Air-2 project1 % echo "This is an added line in readme.txt
" >> readme.txt
[vedpatil@Veds-MacBook-Air-2 project1 % git add readme.txt
[vedpatil@Veds-MacBook-Air-2 project1 % git commit -m "Updated redme.txt"
[main 4635aea] Updated redme.txt
 1 file changed, 1 insertion(+)
[vedpatil@Veds-MacBook-Air-2 project1 % git log --oneline
4635aea (HEAD -> main) Updated redme.txt
336689e Added info.txt
00f61c9 Initial commit with readme.txt
```

1. Add a new file to “readme.txt”.
2. Staged the updated file with “git add readme.txt”.

3. Committed the change using git commit -m "Updated readme.txt" and verified with “git log --oneline”.

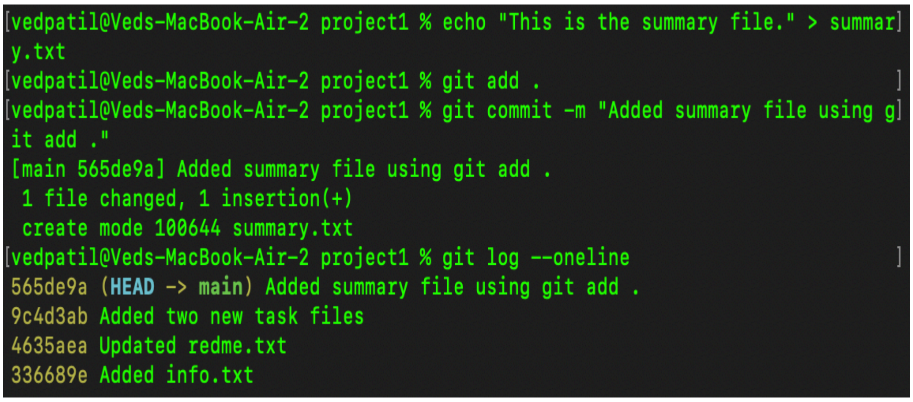
Task 6: Add Multiple Files at Once\



```
[vedpatil@Veds-MacBook-Air-2 project1 % echo "Task 1 details" > task1.txt
[vedpatil@Veds-MacBook-Air-2 project1 % echo "Task 2 details" > task2.txt
[vedpatil@Veds-MacBook-Air-2 project1 % ls
info.txt        readme.txt      task1.txt       task2.txt
[vedpatil@Veds-MacBook-Air-2 project1 % cat task1.txt task2.txt
Task 1 details
Task 2 details
[vedpatil@Veds-MacBook-Air-2 project1 % git add task1.txt task2.txt
[vedpatil@Veds-MacBook-Air-2 project1 % git commit -m "Added two new task files"
[main 9c4d3ab] Added two new task files
 2 files changed, 2 insertions(+)
 create mode 100644 task1.txt
 create mode 100644 task2.txt
```

1. Create two new files “task1.txt” and “task2.txt” with sample text.
2. Stage both files together using “git add task1.txt task2.txt”.
3. Committed the change using git commit -m “Added two new task files”.\

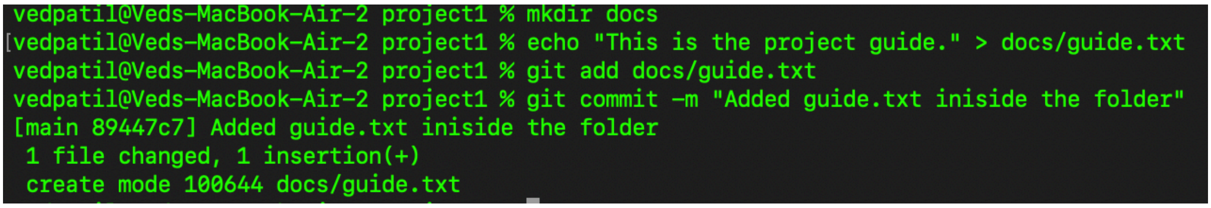
Task7: Use git add . to Add Everything



```
[vedpatil@Veds-MacBook-Air-2 project1 % echo "This is the summary file." > summar]
y.txt
[vedpatil@Veds-MacBook-Air-2 project1 % git add .                                ]
[vedpatil@Veds-MacBook-Air-2 project1 % git commit -m "Added summary file using g]
it add ."
[main 565de9a] Added summary file using git add .
 1 file changed, 1 insertion(+)
 create mode 100644 summary.txt
[vedpatil@Veds-MacBook-Air-2 project1 % git log --oneline                        ]
565de9a (HEAD -> main) Added summary file using git add .
9c4d3ab Added two new task files
4635aea Updated redme.txt
336689e Added info.txt
```

1. Create a new file "summary.txt" and add some text.
2. Staged all changes at once using "git add .".
3. Committed the changes by git commit -m "Added summary file using git add." and you can verify it with "git log –oneline".

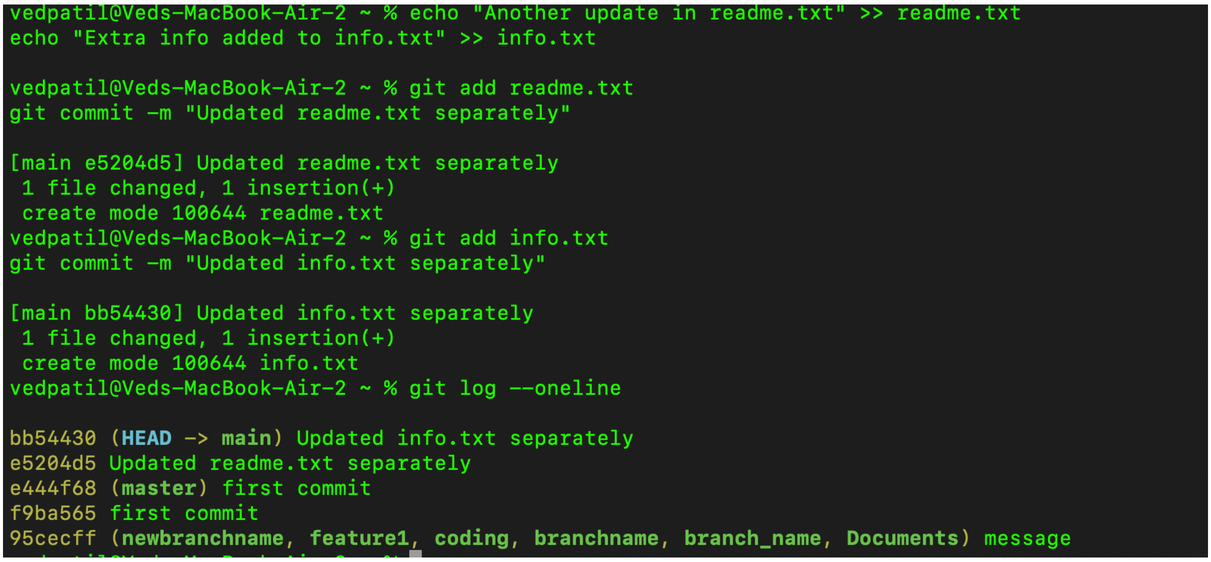
Task 8: Add a File in a Subfolder



```
 vedpatil@Veds-MacBook-Air-2 project1 % mkdir docs
[vedpatil@Veds-MacBook-Air-2 project1 % echo "This is the project guide." > docs/guide.txt
 vedpatil@Veds-MacBook-Air-2 project1 % git add docs/guide.txt
 vedpatil@Veds-MacBook-Air-2 project1 % git commit -m "Added guide.txt iniside the folder"
 [main 89447c7] Added guide.txt iniside the folder
  1 file changed, 1 insertion(+)
  create mode 100644 docs/guide.txt
```

1. Create a new folder named docs and add a file "guide.txt" in it.
2. Stage the file using "git add docs/guide.txt".
3. Commit the file git commit -m "Added guide.txt inside the folder" and you can verify it with "git log –oneline".

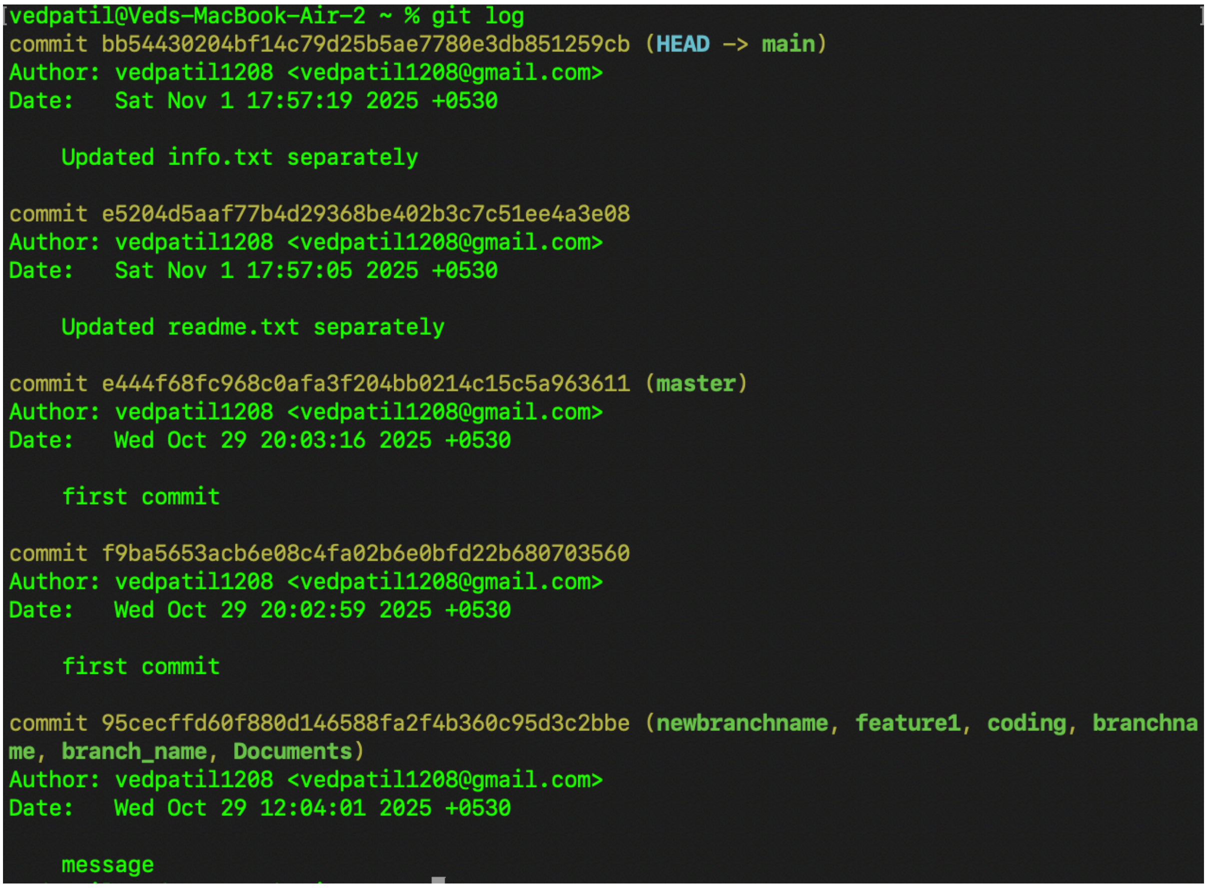
Task 9: Stage and Commit Files Separately



```
vedpatil@Veds-MacBook-Air-2 ~ % echo "Another update in readme.txt" >> readme.txt
echo "Extra info added to info.txt" >> info.txt

vedpatil@Veds-MacBook-Air-2 ~ % git add readme.txt
git commit -m "Updated readme.txt separately"

[main e5204d5] Updated readme.txt separately
 1 file changed, 1 insertion(+)
 create mode 100644 readme.txt
vedpatil@Veds-MacBook-Air-2 ~ % git add info.txt
git commit -m "Updated info.txt separately"

[main bb54430] Updated info.txt separately
 1 file changed, 1 insertion(+)
 create mode 100644 info.txt
vedpatil@Veds-MacBook-Air-2 ~ % git log --oneline

bb54430 (HEAD -> main) Updated info.txt separately
e5204d5 Updated readme.txt separately
e444f68 (master) first commit
f9ba565 first commit
95cecff (newbranchname, feature1, coding, branchname, branch_name, Documents) message
```

1. Modify both “readme.txt” and “info.txt” by adding new lines.
2. Stage and commit “readme.txt” with a separate message.
3. Stage and commit “info.txt” separately.

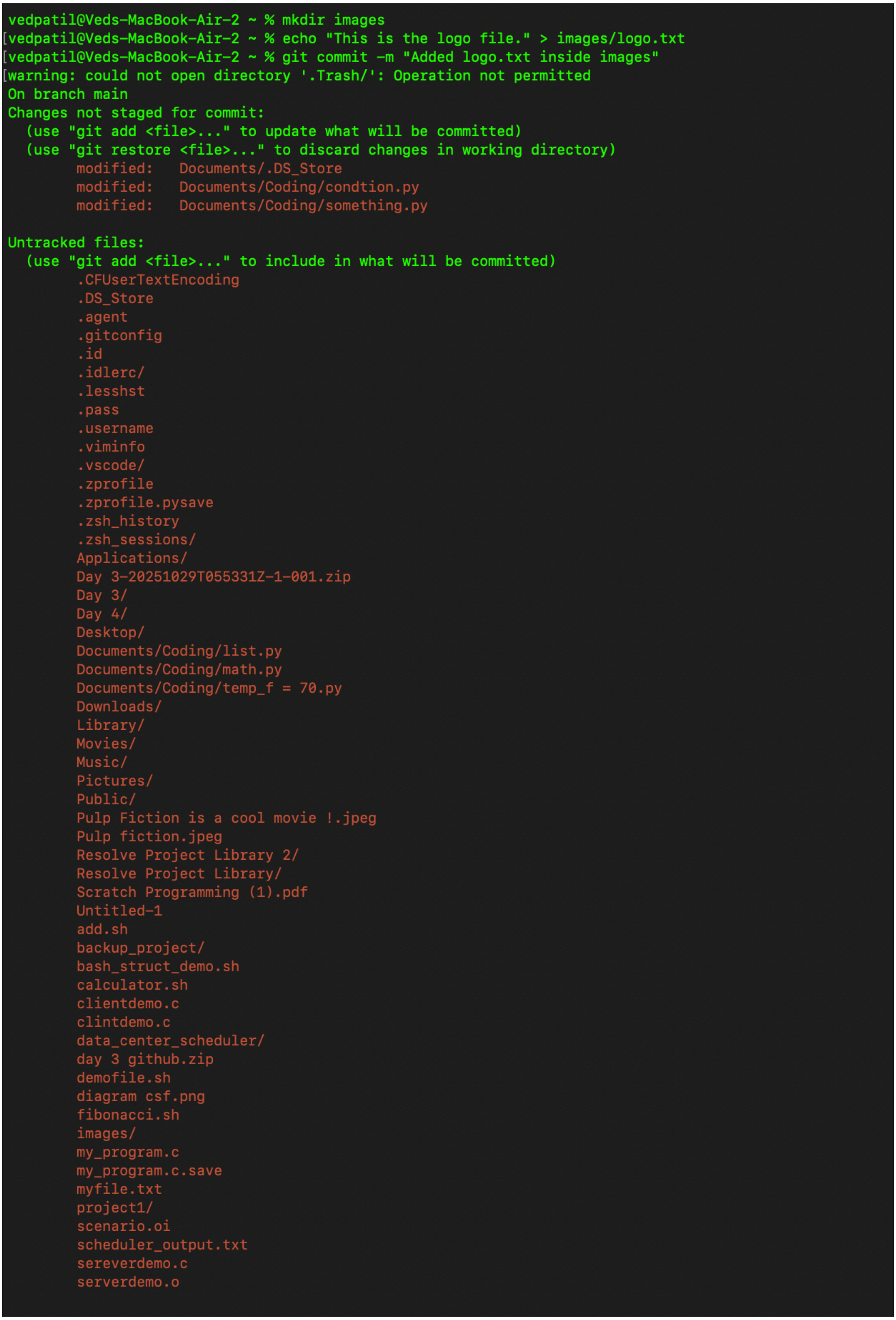
Task 10: View Commit History



```
vedpatil@Veds-MacBook-Air-2 ~ % git log
commit bb54430204bf14c79d25b5ae7780e3db851259cb (HEAD -> main)
Author: vedpatil1208 <vedpatil1208@gmail.com>
Date:   Sat Nov 1 17:57:19 2025 +0530

    Updated info.txt separately

commit e5204d5aaf77b4d29368be402b3c7c51ee4a3e08
Author: vedpatil1208 <vedpatil1208@gmail.com>
Date:   Sat Nov 1 17:57:05 2025 +0530

    Updated readme.txt separately

commit e444f68fc968c0afa3f204bb0214c15c5a963611 (master)
Author: vedpatil1208 <vedpatil1208@gmail.com>
Date:   Wed Oct 29 20:03:16 2025 +0530

    first commit

commit f9ba5653acb6e08c4fa02b6e0bfd22b680703560
Author: vedpatil1208 <vedpatil1208@gmail.com>
Date:   Wed Oct 29 20:02:59 2025 +0530

    first commit

commit 95cecffd60f880d146588fa2f4b360c95d3c2bbe (newbranchname, feature1, coding, branchna
me, branch_name, Documents)
Author: vedpatil1208 <vedpatil1208@gmail.com>
Date:   Wed Oct 29 12:04:01 2025 +0530

    message
```

1. Executed “git log” to display the complete commit history of the repository.
2. Reviewed each commit message, author, and timestamp for verification.

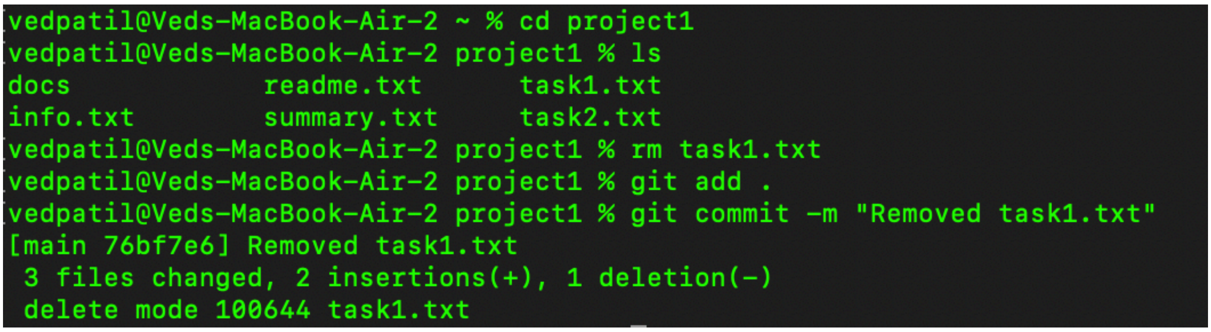
Task 11: Add a folder with a file



```
vedpatil@Veds-MacBook-Air-2 ~ % mkdir images
[vedpatil@Veds-MacBook-Air-2 ~ % echo "This is the logo file." > images/logo.txt
[vedpatil@Veds-MacBook-Air-2 ~ % git commit -m "Added logo.txt inside images"
[warning: could not open directory '.Trash/': Operation not permitted
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Documents/.DS_Store
        modified:   Documents/Coding/condtion.py
        modified:   Documents/Coding/something.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .CFUserTextEncoding
        .DS_Store
        .agent
        .gitconfig
        .id
        .idlerc/
        .lesshst
        .pass
        .username
        .viminfo
        .vscode/
        .zprofile
        .zprofile.pysave
        .zsh_history
        .zsh_sessions/
        Applications/
        Day 3-20251029T055331Z-1-001.zip
        Day 3/
        Day 4/
        Desktop/
        Documents/Coding/list.py
        Documents/Coding/math.py
        Documents/Coding/temp_f = 70.py
        Downloads/
        Library/
        Movies/
        Music/
        Pictures/
        Public/
        Pulp Fiction is a cool movie !.jpeg
        Pulp fiction.jpeg
        Resolve Project Library 2/
        Resolve Project Library/
        Scratch Programming (1).pdf
        Untitled-1
        add.sh
        backup_project/
        bash_struct_demo.sh
        calculator.sh
        clientdemo.c
        clintdemo.c
        data_center_scheduler/
        day 3 github.zip
        demofile.sh
        diagram csf.png
        fibonacci.sh
        images/
        my_program.c
        my_program.c.save
        myfile.txt
        project1/
        scenario.oi
        scheduler_output.txt
        sereverdemo.c
        serverdemo.o
```

1. Create a new folder named images and add a file “logo.txt” in it.
2. Stage the file using “git add images/ .txt’.
3. Committed the files with commit -m “Added logo.txt inside images”.

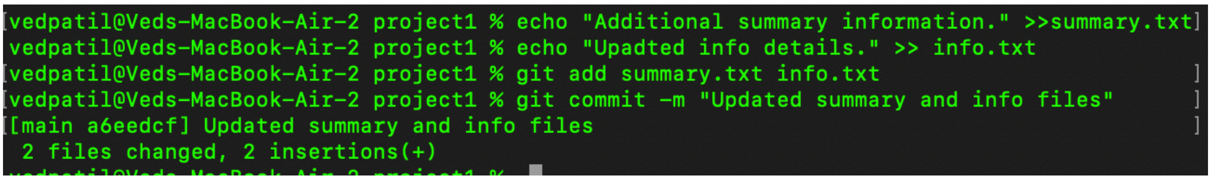
Task 12 : Remove a File and Commit the Change



```
vedpatil@Veds-MacBook-Air-2 ~ % cd project1
vedpatil@Veds-MacBook-Air-2 project1 % ls
docs            readme.txt      task1.txt
info.txt        summary.txt     task2.txt
vedpatil@Veds-MacBook-Air-2 project1 % rm task1.txt
vedpatil@Veds-MacBook-Air-2 project1 % git add .
vedpatil@Veds-MacBook-Air-2 project1 % git commit -m "Removed task1.txt"
[main 76bf7e6] Removed task1.txt
 3 files changed, 2 insertions(+), 1 deletion(-)
 delete mode 100644 task1.txt
```

1. Navigated to the project folder using “cd project1.
2. Deleted the file “task1.txt” and staged the change with “git add ..”
3. Committed the deletion using git commit -m "Removed task1.txt".

Task 13: Modify and Commit Multiple Files Together

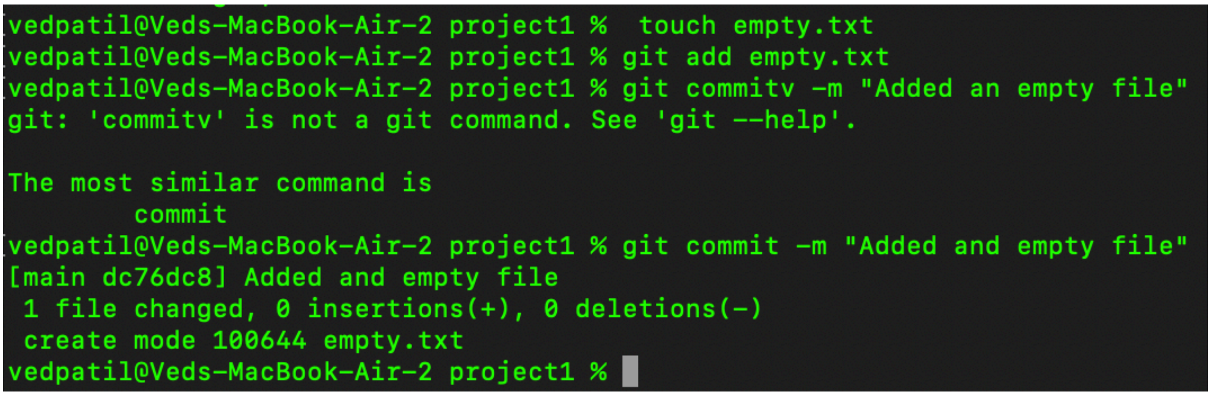


```
vedpatil@Veds-MacBook-Air-2 project1 % echo "Additional summary information." >>summary.txt
vedpatil@Veds-MacBook-Air-2 project1 % echo "Upadted info details." >> info.txt
vedpatil@Veds-MacBook-Air-2 project1 % git add summary.txt info.txt
vedpatil@Veds-MacBook-Air-2 project1 % git commit -m "Updated summary and info files"
[main a6eedcf] Updated summary and info files
 2 files changed, 2 insertions(+)
vedpatil@Veds-MacBook-Air-2 project1 %
```

1. Modify both summary.txt and info.txt and by adding new files.

2. Stage both files together using “git add summary.txt info.txt .
3. Committed the changes with git commit -m "Updated summary and info files".

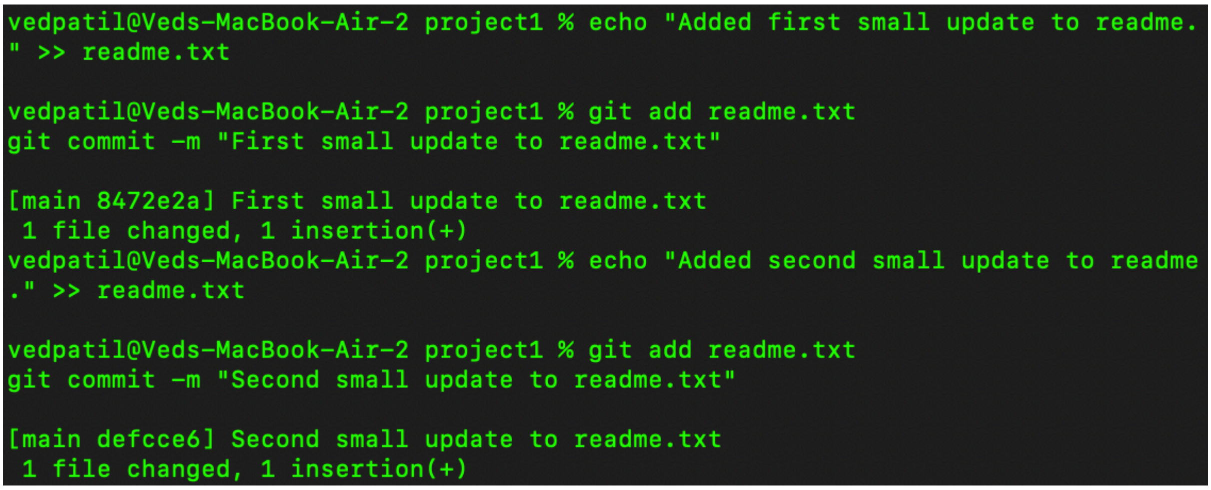
Task 14: Create and Commit an Empty File



```
vedpatil@Veds-MacBook-Air-2 project1 %  touch empty.txt
vedpatil@Veds-MacBook-Air-2 project1 % git add empty.txt
vedpatil@Veds-MacBook-Air-2 project1 % git commitv -m "Added an empty file"
git: 'commitv' is not a git command. See 'git --help'.

The most similar command is
        commit
vedpatil@Veds-MacBook-Air-2 project1 % git commit -m "Added and empty file"
[main dc76dc8] Added and empty file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 empty.txt
vedpatil@Veds-MacBook-Air-2 project1 %
```

1. Create an empty file name empty.txt using command “touch”.
2. Stage the file with git add.txt.
3. Committed the file using git commit -m "Added an empty file".

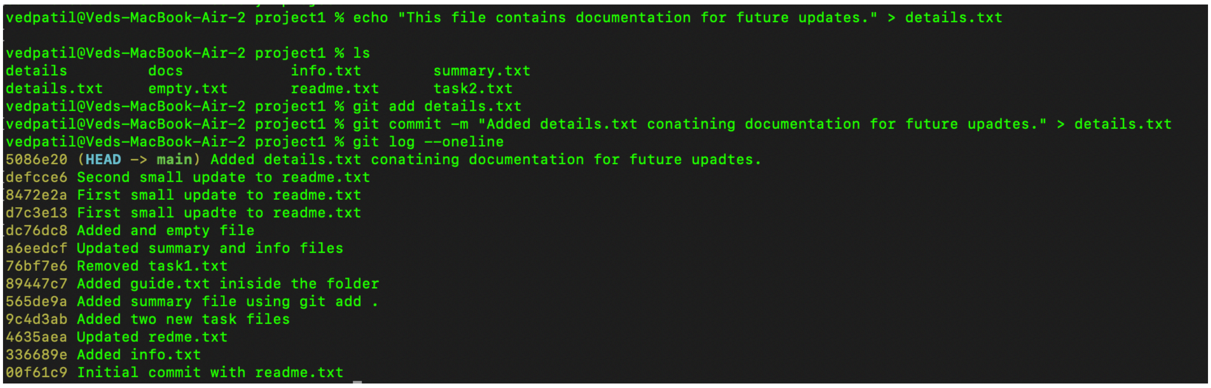
Task 15: Make Consecutive Small Commits



```
vedpatil@Veds-MacBook-Air-2 project1 % echo "Added first small update to readme.
" >> readme.txt

vedpatil@Veds-MacBook-Air-2 project1 % git add readme.txt
git commit -m "First small update to readme.txt"

[main 8472e2a] First small update to readme.txt
 1 file changed, 1 insertion(+)
vedpatil@Veds-MacBook-Air-2 project1 % echo "Added second small update to readme
." >> readme.txt

vedpatil@Veds-MacBook-Air-2 project1 % git add readme.txt
git commit -m "Second small update to readme.txt"

[main defcce6] Second small update to readme.txt
 1 file changed, 1 insertion(+)
```

1. Added a new line to readme.txt and made the first small commit.
2. Added another line to readme.txt and made the second small commit.

Task 16: Write a Detailed Commit Message

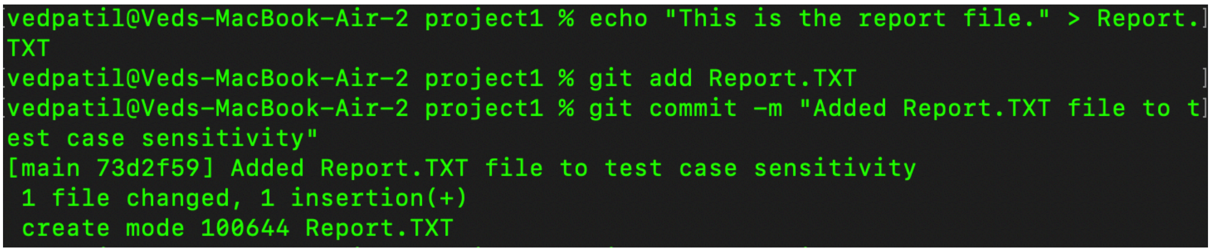


```
vedpatil@Veds-MacBook-Air-2 project1 % echo "This file contains documentation for future updates." > details.txt

vedpatil@Veds-MacBook-Air-2 project1 % ls
details         docs            info.txt        summary.txt
details.txt     empty.txt       readme.txt      task2.txt
vedpatil@Veds-MacBook-Air-2 project1 % git add details.txt
vedpatil@Veds-MacBook-Air-2 project1 % git commit -m "Added details.txt conatining documentation for future upadtes." > details.txt
vedpatil@Veds-MacBook-Air-2 project1 % git log --oneline
5086e20 (HEAD -> main) Added details.txt conatining documentation for future upadtes.
defcce6 Second small update to readme.txt
8472e2a First small update to readme.txt
d7c3e13 First small upadte to readme.txt
dc76dc8 Added and empty file
a6eedcf Updated summary and info files
76bf7e6 Removed task1.txt
89447c7 Added guide.txt iniside the folder
565de9a Added summary file using git add .
9c4d3ab Added two new task files
4635aea Updated redme.txt
336689e Added info.txt
00f61c9 Initial commit with readme.txt
```

1. Created a new file details.txt containing documentation for future updates.
2. Staged the file using git add details.txt.

3. Committed the file with a detailed message using git commit -m "Added details.txt containing documentation for future updates".

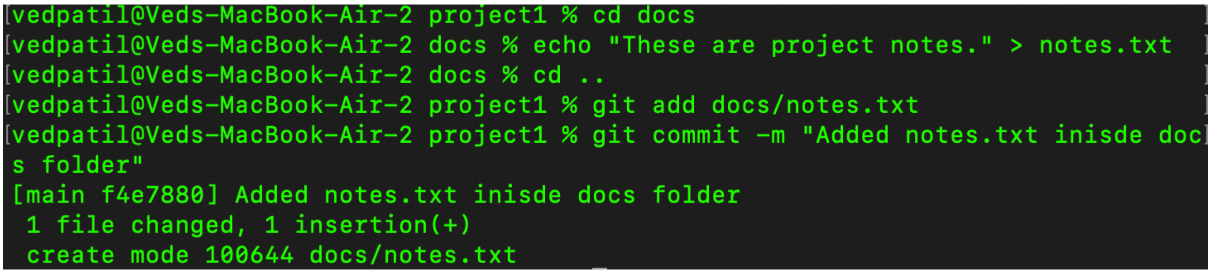
Task 17: Commit a File with Uppercase Name



```
vedpatil@Veds-MacBook-Air-2 project1 % echo "This is the report file." > Report.
TXT
vedpatil@Veds-MacBook-Air-2 project1 % git add Report.TXT
vedpatil@Veds-MacBook-Air-2 project1 % git commit -m "Added Report.TXT file to t
est case sensitivity"
[main 73d2f59] Added Report.TXT file to test case sensitivity
 1 file changed, 1 insertion(+)
 create mode 100644 Report.TXT
```

1. Create a new file “Report.TXT” with sample text.
2. Stage the file using git add Report.TXT .
3. Committed the file with git commit -m "Added Report.TXT file to test case sensitivity".

Task 18: Stage and Commit a File Inside a Subfolder



```
vedpatil@Veds-MacBook-Air-2 project1 % cd docs
vedpatil@Veds-MacBook-Air-2 docs % echo "These are project notes." > notes.txt
vedpatil@Veds-MacBook-Air-2 docs % cd ..
vedpatil@Veds-MacBook-Air-2 project1 % git add docs/notes.txt
vedpatil@Veds-MacBook-Air-2 project1 % git commit -m "Added notes.txt inisde doc
s folder"
[main f4e7880] Added notes.txt inisde docs folder
 1 file changed, 1 insertion(+)
 create mode 100644 docs/notes.txt
```

1. Create a new file notes.txt inside the docs with some text.
2. Stage the file using “git add docs/notes.txt” .

3. Commit the file with git commit -m "Added notes.txt inside docs folder" and you can verify with git log --oneline.