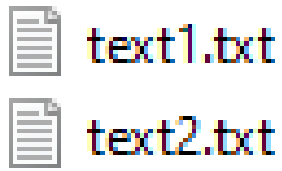
1)Создаем локальный репозиторий

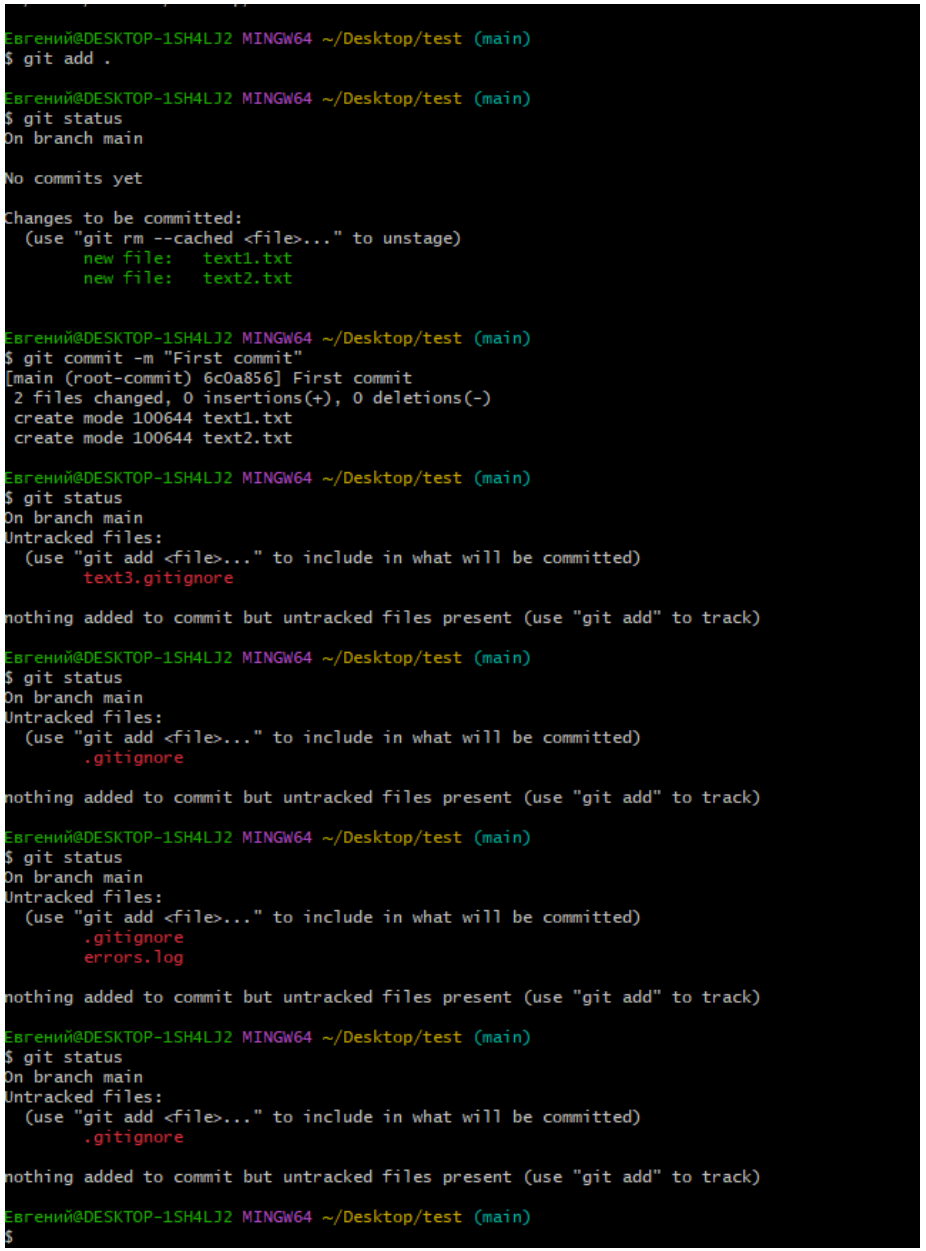
.git	07.09.2025 16:59	Папка с файлами	
.gitignore	07.09.2025 14:40	Текстовый докум...	1 КБ
errors.log	07.09.2025 14:40	Текстовый докум...	0 КБ
README.md	07.09.2025 16:53	Исходный файл ...	1 КБ
text1.txt	07.09.2025 16:05	Текстовый докум...	1 КБ
text2.txt	07.09.2025 15:20	Текстовый докум...	1 КБ

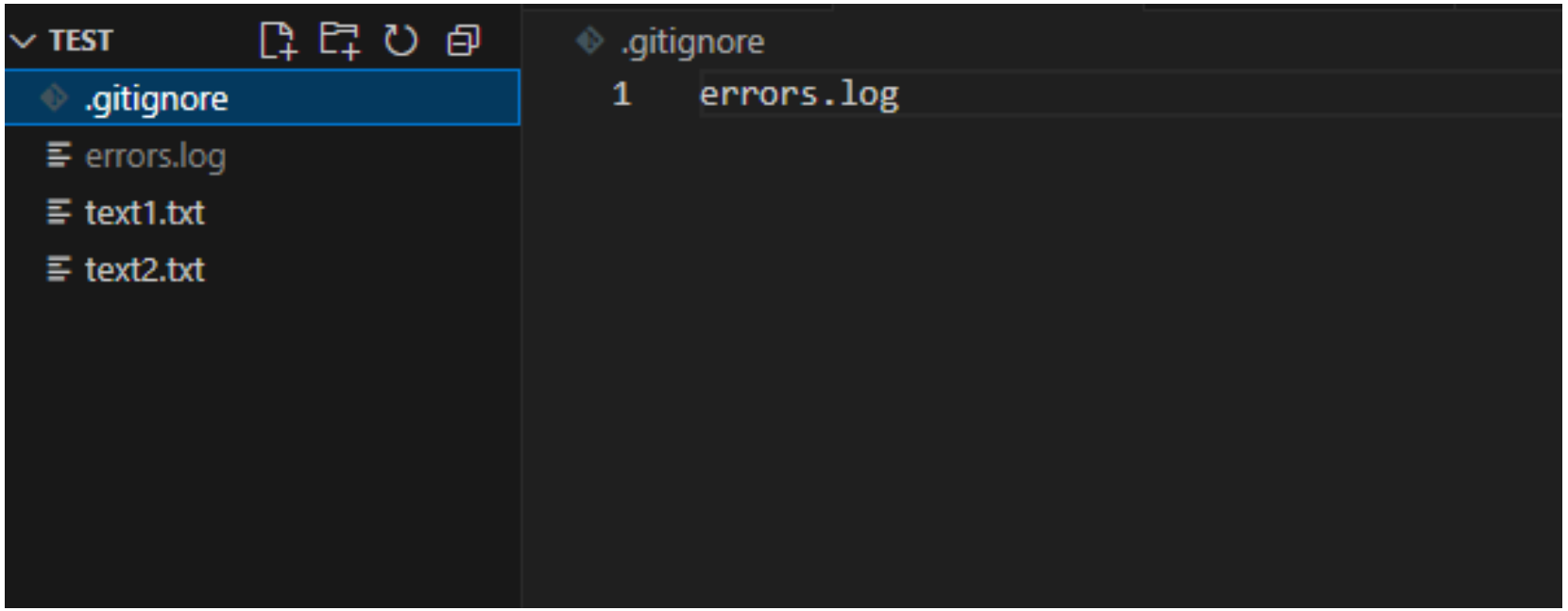
2)Добавляем два текстовых файла в репозиторий и делаем первый коммит



```
Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        text1.txt

nothing added to commit but untracked files present (use "git add" to track)

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git add .

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   text1.txt


Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   text1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        text2.txt


Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
        add

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git add /
fatal: C:/Program Files/Git/: 'C:/Program Files/Git/' is outside repository at '
C:/Users/Евгений/Desktop/test'
```

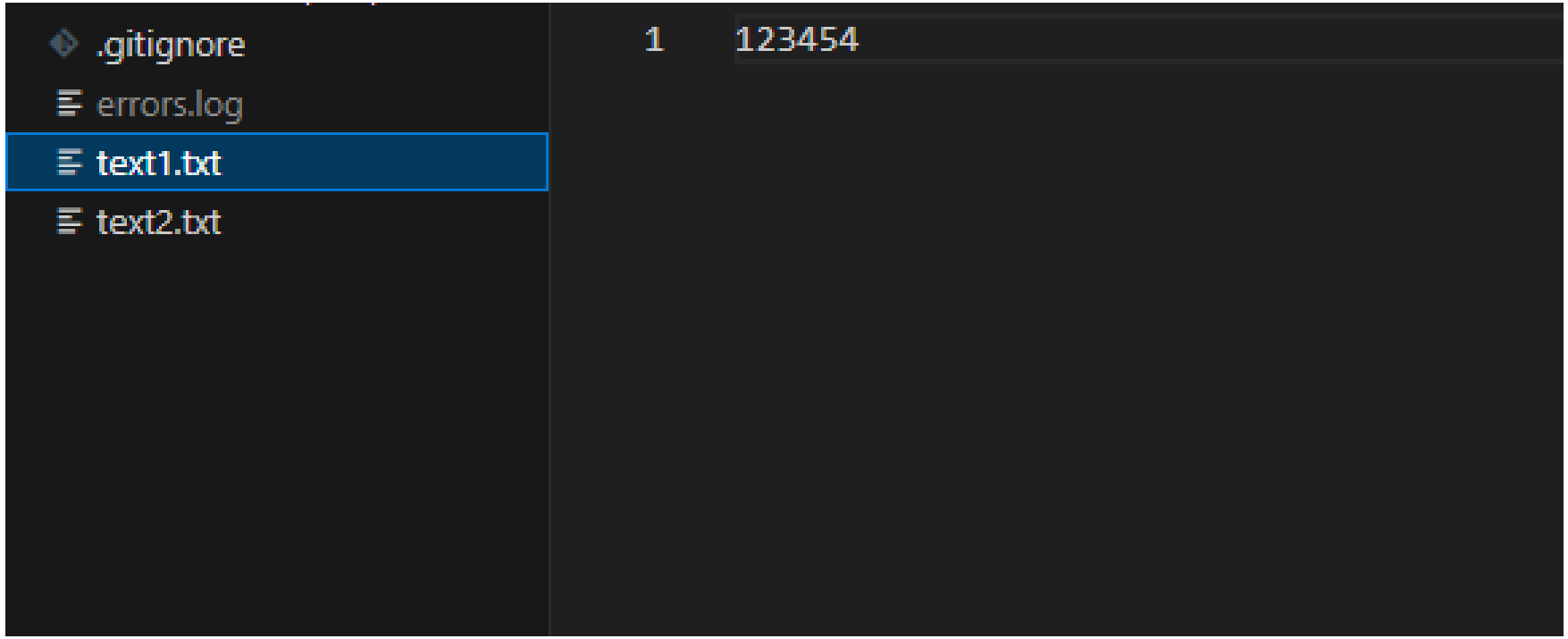
3)Добавьте файл .gitignore, в который прописываем шаблон для игнорирования определенных файлов




TEST
.gitignore
errors.log
text1.txt
text2.txt
.gitignore
1 errors.log

4)Изменяем один из отслеживаемых фалов и делаем второй коммит

```
Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git add .

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .gitignore


Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git commit -m "second commit"
[main 1e0633f] second commit
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$
```


.gitignore
errors.log
text1.txt
text2.txt
1 123454

5)Создаем новую ветку для изменений

```
Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git checkout -b "testing"
Switched to a new branch 'testing'

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (testing)
$ git status
On branch testing
nothing to commit, working tree clean

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (testing)
$ git status
On branch testing
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:   text2.txt

no changes added to commit (use "git add" and/or "git commit -a")

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (testing)
$ git add .

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (testing)
$ git status
On branch testing
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   text2.txt
```

6)Вернулся в основную ветку и сделал слияние.

```
Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (testing)
$ git checkout "main"
Switched to branch 'main'

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git merge "testing"
Updating cef26a0..5ca6f94
Fast-forward
 text2.txt | 1 +
 1 file changed, 1 insertion(+)
```

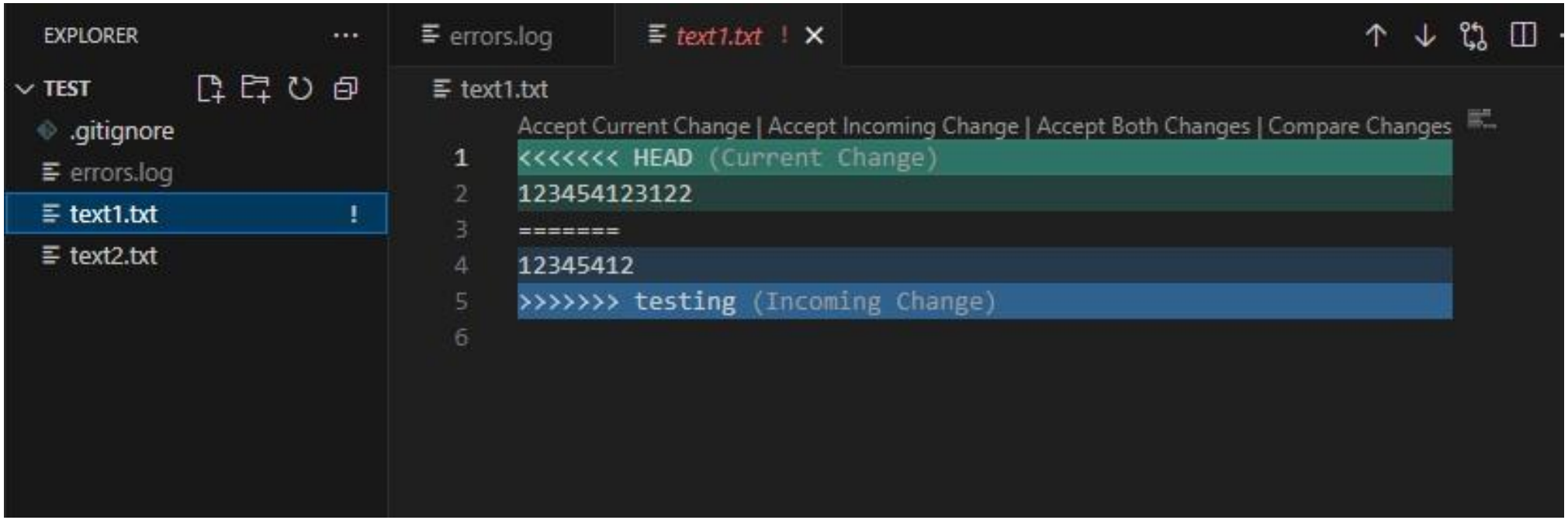
7) Внес изменение в основной ветке и зафиксировал его коммитом

```
Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (testing)
$ git commit -m "third commit"
[testing 5ca6f94] third commit
 1 file changed, 1 insertion(+)

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (testing)
$ git log
commit 5ca6f94c3de7dcbe0671024c08a2ded81e3cb1ff (HEAD -> testing)
Author: Evgeniy <soloviev4466@gmail.com>
Date:   Sun Sep 7 15:16:22 2025 +0300

    third commit

commit cef26a09388f636016f1c869d902f825c4db65c3 (main)
Author: Evgeniy <soloviev4466@gmail.com>
Date:   Sun Sep 7 15:07:39 2025 +0300

    second commit

commit 6c0a8566180b0c68c664e6a684de7aeebdc2dab1
Author: Evgeniy <soloviev4466@gmail.com>
Date:   Sun Sep 7 14:35:27 2025 +0300

    First commit
```

8)Внес изменение в файл и попытался сделать слияние. Произошел конфликт

```
Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (testing)
$ git commit -am "testing change"
[testing 89c2602] testing change
 1 file changed, 1 insertion(+), 1 deletion(-)

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (testing)
$ git checkout main
Switched to branch 'main'

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git commit -am "main change"
[main 009c807] main change
 1 file changed, 1 insertion(+), 1 deletion(-)

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git log --oneline --graph --all
* 009c807 (HEAD -> main) main change
| * 89c2602 (testing) testing change
|/
* 5ca6f94 third commit
* cef26a0 second commit
* 6c0a856 First commit
```

```
Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git merge testing
Auto-merging text1.txt
CONFLICT (content): Merge conflict in text1.txt
Automatic merge failed; fix conflicts and then commit the result.
```


EXPLORER
TEST
.gitignore
errors.log
text1.txt
text2.txt
errors.log
text1.txt
text1.txt
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<<< HEAD (Current Change)
123454123122
=======
12345412
>>>>>>> testing (Incoming Change)

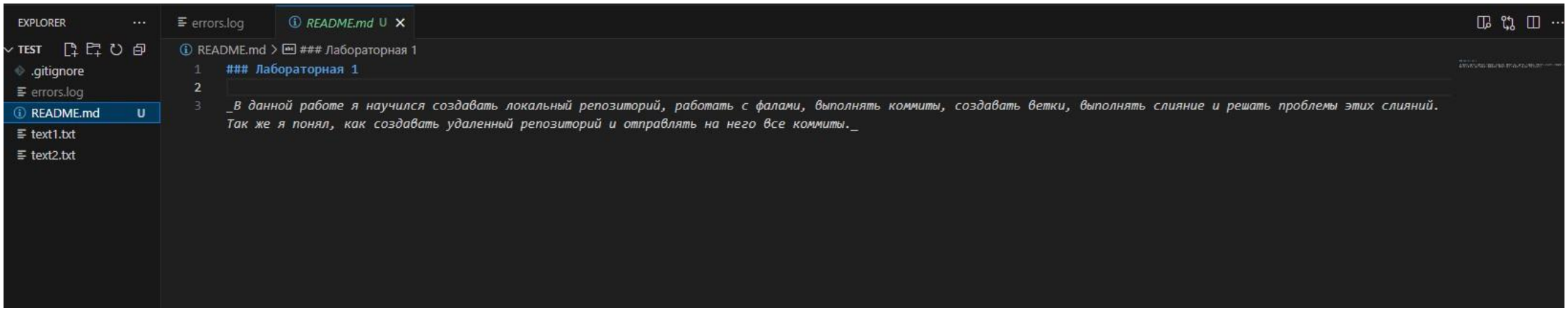
9)Просмотреть историю коммитов с отображением ветвлений

```
Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git log --oneline --graph --all
* 5888ba6 (HEAD -> main) resolve conflict
*   ddb7ef9 resolve conflict
|\
| * 89c2602 (testing) testing change
* | 009c807 main change
|/
* 5ca6f94 third commit
* cef26a0 second commit
* 6c0a856 First commit
```

10)Создаем удаленной репозиторий, связываем его с локальным и отправляем в него коммиты

```
Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git remote add origin https://github.com/Viant11/example

Евгений@DESKTOP-1SH4LJ2 MINGW64 ~/Desktop/test (main)
$ git remote -v
origin  https://github.com/Viant11/example (fetch)
origin  https://github.com/Viant11/example (push)
```

11)Создаем на Github файл README с кратким описанием проекта.



12) В локальном репозитории получите изменения с github с помощью команды pull, чтобы скачать файл README.md

```
 git pull origin main
emote: Enumerating objects: 39, done.
emote: Counting objects: 100% (39/39), done.
emote: Compressing objects: 100% (38/38), done.
emote: Total 38 (delta 19), reused 0 (delta 0), pack-reused 0 (fro
npacking objects: 100% (38/38), 769.84 KiB | 501.00 KiB/s, done.
rom https://github.com/many-wile/-11
* branch            main       -> FETCH_HEAD
  90ba942..605d464  main       -> origin/main
pdating 2e0fcab..605d464
ast-forward
README.md  |    5 +++++
report.pdf | Bin 0 -> 417708 bytes
text1.txt  |    6 +++---
text2.txt  |    5 ++---
4 files changed, 10 insertions(+), 6 deletions(-)
create mode 100644 README.md
create mode 100644 report.pdf
```