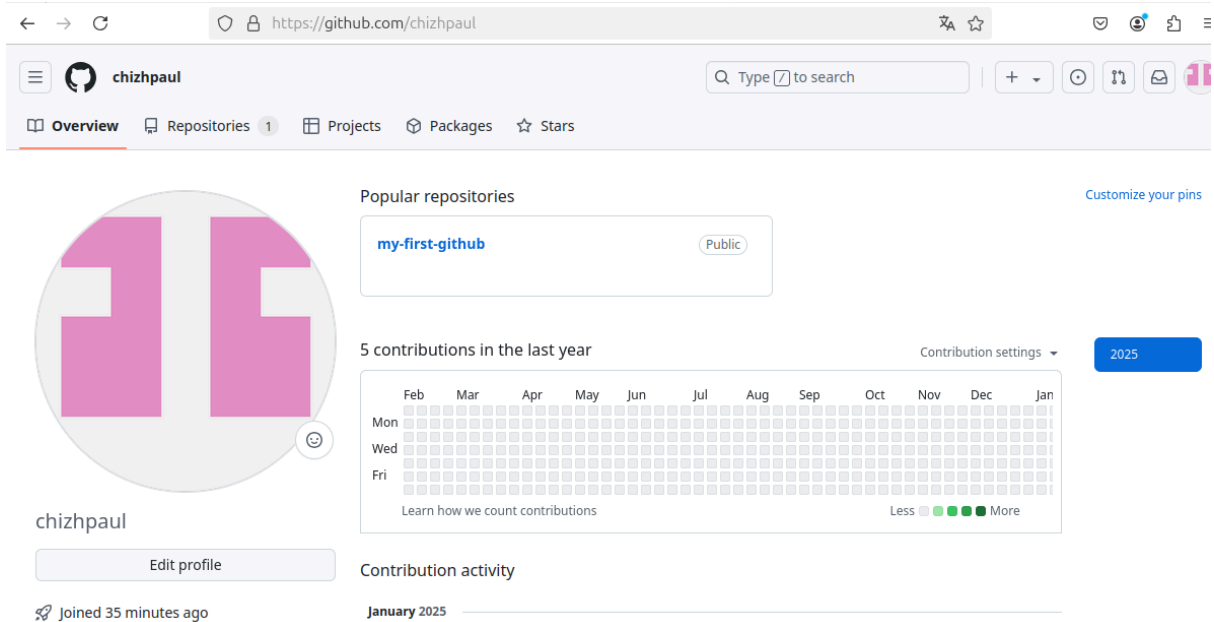


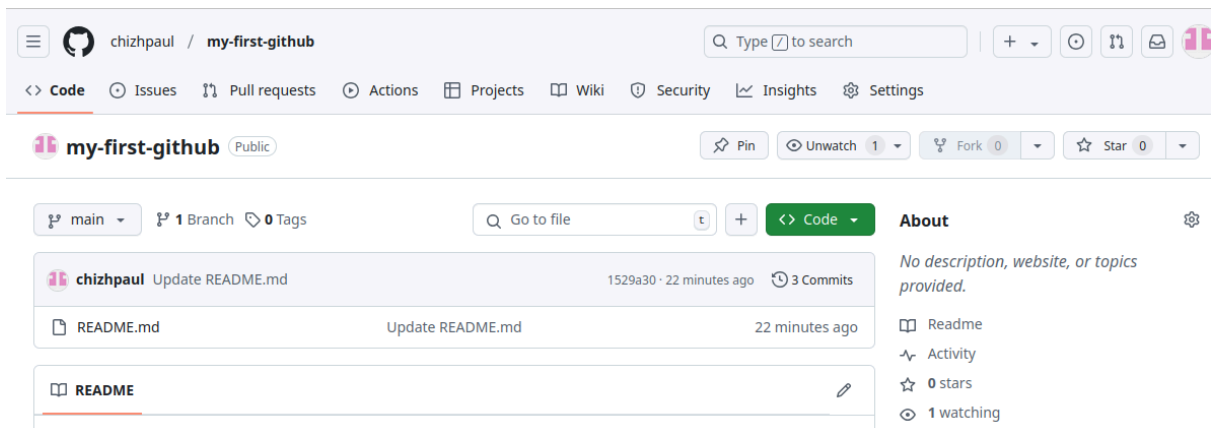
Задание 1

Что нужно сделать:

1. Зарегистрируйте аккаунт на GitHub.



2. Создайте новый отдельный публичный репозиторий. Обязательно поставьте галочку в поле «Initialize this repository with a README».



3. Клонировать репозиторий, используя https протокол `git clone`

```
pavel@pavel-pk:~$ git clone https://github.com/chizhpaul/my-first-github.git
Клонирование в «my-first-github»...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Получение объектов: 100% (9/9), готово.
Определение изменений: 100% (1/1), готово.
```

4. Перейдите в каталог с клоном репозитория.

```
pavel@pavel-pk:~$ cd my-first-github/
pavel@pavel-pk:~/my-first-github$ ls -la
итого 20
drwxrwxr-x  3 pavel pavel 4096 янв 29 13:28 .
drwxr-x--- 16 pavel pavel 4096 янв 29 13:28 ..
drwxrwxr-x  8 pavel pavel 4096 янв 29 13:28 .git
-rw-rw-r--  1 pavel pavel 5743 янв 29 13:28 README.md
```

5. Произведите первоначальную настройку Git, указав своё настоящее имя и email: `git config --global user.name` и `git config --global user.email johndoe@example.com`.

```
pavel@pavel-pk:~/my-first-github$ git config --global user.name chizhpaul
pavel@pavel-pk:~/my-first-github$ git config --global user.email chizhpaul@yandex.ru
```

6. Выполните команду `git status` и запомните результат.

```
pavel@pavel-pk:~/my-first-github$ git status
Текущая ветка: main
Эта ветка соответствует «origin/main».

ничего коммитить, нет изменений в рабочем каталоге
```

7. Отредактируйте файл `README.md` любым удобным способом, переведя файл в состояние `Modified`.

```

[+] pavel@pavel-pk: ~/my-first-github
GNU nano 7.2 README.md
# my-first-github
Hello! This is my first github project!
```

8. Ещё раз выполните `git status` и продолжайте проверять вывод этой команды после каждого следующего шага.

```
pavel@pavel-pk:~/my-first-github$ git status
Текущая ветка: main
Эта ветка соответствует «origin/main».

Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
      изменено:      README.md

индекс пуст (используйте «git add» и/или «git commit -a»)
```

9. Посмотрите изменения в файле `README.md`, выполнив команды `git diff` и `git diff --staged`.

```
pavel@pavel-pk:~/my-first-github$ git diff
diff --git a/README.md b/README.md
index 63e2363..8c74534 100644
--- a/README.md
+++ b/README.md
@@ -1,3 +1,7 @@
+# my-first-github
+Hello! This is my first github project!
+
```

10. Переведите файл в состояние staged или, как говорят, добавьте файл в коммит, командой `git add README.md`.

```
pavel@pavel-pk:~/my-first-github$ git status
Текущая ветка: main
Эта ветка соответствует «origin/main».

Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:      README.md
```

11. Ещё раз выполните команды `git diff` и `git diff --staged`.

```
pavel@pavel-pk:~/my-first-github$ git diff
pavel@pavel-pk:~/my-first-github$ git diff --staged
diff --git a/README.md b/README.md
index 63e2363..8c74534 100644
--- a/README.md
+++ b/README.md
@@ -1,3 +1,7 @@
+# my-first-github
+Hello! This is my first github project!
+
```

12. Теперь можно сделать коммит `git commit -m 'First commit'`.

```
pavel@pavel-pk:~/my-first-github$ git commit -m 'First commit'
[main 6f4ef9c] First commit
 1 file changed, 4 insertions(+)
pavel@pavel-pk:~/my-first-github$ git status
Текущая ветка: main
Ваша ветка опережает «origin/main» на 1 коммит.
  (используйте «git push», чтобы опубликовать ваши локальные коммиты)

ничего коммитить, нет изменений в рабочем каталоге
```

13. Сделайте git push origin master.

```
pavel@pavel-pk:~/my-first-github$ git remote set-url origin https://ghp_YofnpclARID9zB0FIX1VP0dthdHljJ3mJKmF0
github.com/chizhpaul/my-first-github.git
pavel@pavel-pk:~/my-first-github$ git push origin main
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 317 байтов | 317.00 КиБ/с, готово.
Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/chizhpaul/my-first-github.git
 1529a30..6f4ef9c  main -> main
```

The screenshot shows the GitHub web interface for the repository 'my-first-github' by user 'chizhpaul'. The repository is public and has 1 branch (main) and 0 tags. The commit history shows a single commit 'First commit' by 'chizhpaul' with hash '6f4ef9c' 37 minutes ago, containing 4 commits. The README file is displayed, showing the title 'my-first-github' and the content 'Hello! This is my first github project!'. The right sidebar shows repository statistics: 0 stars, 1 watching, and 0 forks. There are no releases published.

```
pavel@pavel-pk:~/my-first-github$ git log -p
commit 6f4ef9c29582143439a3a0de5cc98e33e0677944 (HEAD -> main, origin/main, origin/HEAD)
Author: chizhpaul <chizhpaul@yandex.ru>
Date:   Wed Jan 29 13:53:50 2025 +0900

    First commit

diff --git a/README.md b/README.md
index 63e2363..8c74534 100644
--- a/README.md
+++ b/README.md
@@ -1,3 +1,7 @@
+# my-first-github
+Hello! This is my first github project!
```

В качестве ответа добавьте ссылку на этот коммит в ваш md-файл с решением.
<https://github.com/chizhpaul/my-first-github/commit/6f4ef9c29582143439a3a0de5cc98e33e0677944>

Задание 2

Что нужно сделать:

1. Создайте файл .gitignore (обратите внимание на точку в начале файла) и проверьте его статус сразу после создания.

```
pavel@pavel-pk:~/my-first-github$ nano .gitignore
pavel@pavel-pk:~/my-first-github$ git status
Текущая ветка: main
Эта ветка соответствует «origin/main».

Неотслеживаемые файлы:
(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)
    .gitignore

индекс пуст, но есть неотслеживаемые файлы
(используйте «git add», чтобы проиндексировать их)
```

2. Добавьте файл .gitignore в следующий коммит git add....

```
pavel@pavel-pk:~/my-first-github$ git add .gitignore
pavel@pavel-pk:~/my-first-github$ git status
Текущая ветка: main
Эта ветка соответствует «origin/main».

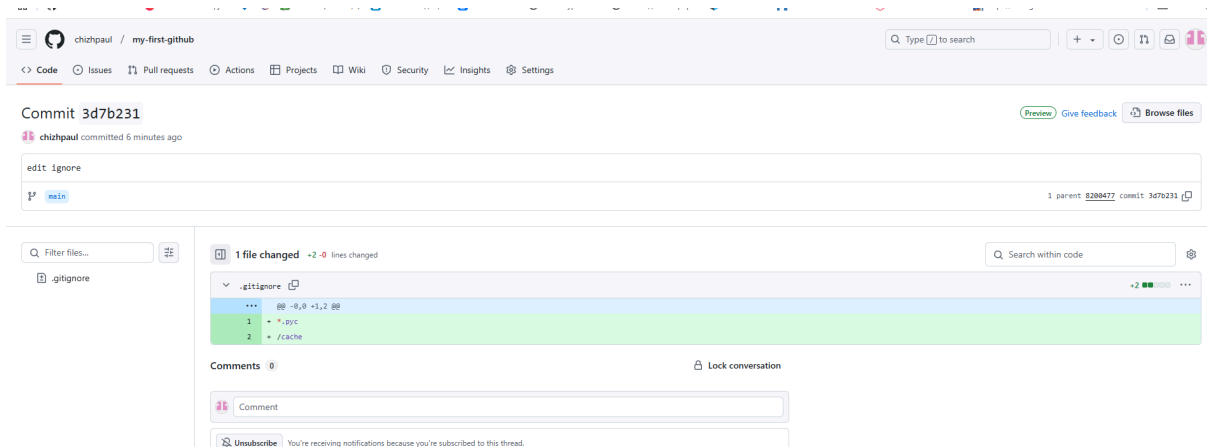
Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    новый файл:    .gitignore
```

3. Напишите правила в этом файле, чтобы игнорировать любые файлы .рус, а также все файлы в директории cache.

```
pavel@pavel-pk: ~/my-first-github
GNU nano 7.2 .gitignore
*.рус
/cache
```

4. Сделайте коммит и пуш.

```
pavel@pavel-pk:~/my-first-github$ git add .gitignore
pavel@pavel-pk:~/my-first-github$ git commit -m "edit ignore"
[main 3d7b231] edit ignore
1 file changed, 2 insertions(+)
pavel@pavel-pk:~/my-first-github$ git push origin main
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 288 байтов | 288.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/chizhpaul/my-first-github.git
8200477..3d7b231 main -> main
```



В качестве ответа добавьте ссылку на этот коммит в ваш md-файл с решением.
<https://github.com/chizhpaul/my-first-github/commit/3d7b2314bdc6a33d9918ae482ca7439fb295178>

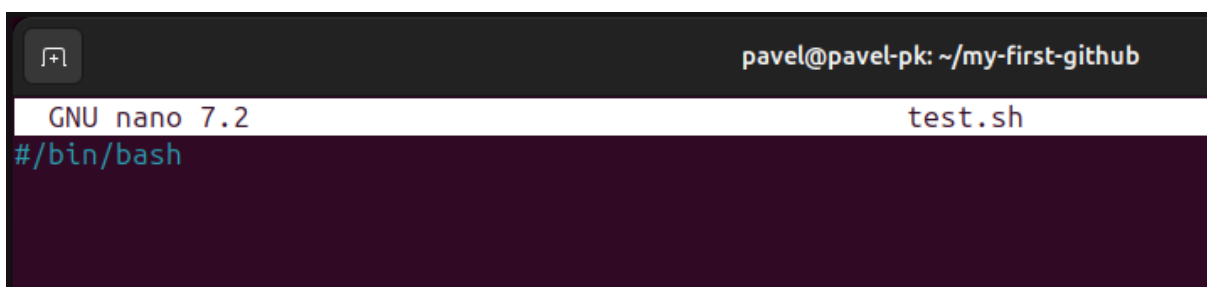
Задание 3

Что нужно сделать:

1. Создайте новую ветку dev и переключитесь на неё.

```
pavel@pavel-pk:~/my-first-github$ git checkout -b dev
Переключились на новую ветку «dev»
pavel@pavel-pk:~/my-first-github$ git branch
* dev
main
```

2. Создайте в ветке dev файл test.sh с произвольным содержимым.



```
pavel@pavel-pk:~/my-first-github$ git add test.sh
pavel@pavel-pk:~/my-first-github$ git status
Текущая ветка: dev
Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
новый файл:   test.sh
```

3. Сделайте несколько коммитов и пушей в ветку dev, имитируя активную работу над файлом в процессе разработки.

```
pavel@pavel-pk:~/my-first-github$ git commit -m "create test.sh"
[dev 830b4a7] create test.sh
 1 file changed, 1 insertion(+)
 create mode 100644 test.sh
pavel@pavel-pk:~/my-first-github$ git push origin dev
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 319 байтов | 319.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote:
remote: Create a pull request for 'dev' on GitHub by visiting:
remote:   https://github.com/chizhpaul/my-first-github/pull/new/dev
remote:
To https://github.com/chizhpaul/my-first-github.git
 * [new branch]      dev -> dev
```

```
pavel@pavel-pk:~/my-first-github$ echo "#add" >> test.sh
pavel@pavel-pk:~/my-first-github$ git add test.sh
pavel@pavel-pk:~/my-first-github$ git status
Текущая ветка: dev
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
        изменено:      test.sh

pavel@pavel-pk:~/my-first-github$ git commit -m "edit test.sh"
[dev e38a8c9] edit test.sh
 1 file changed, 1 insertion(+)
pavel@pavel-pk:~/my-first-github$ git push origin dev
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 264 байта | 264.00 КиБ/с, готово.
Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/chizhpaul/my-first-github.git
 830b4a7..e38a8c9  dev -> dev
```

```
pavel@pavel-pk:~/my-first-github$ echo "#copy" >> test.sh
pavel@pavel-pk:~/my-first-github$ git add test.sh
pavel@pavel-pk:~/my-first-github$ git commit -m "edit2 test.sh"
[dev b64d82e] edit2 test.sh
 1 file changed, 1 insertion(+)
pavel@pavel-pk:~/my-first-github$ git push origin dev
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 271 байт | 271.00 КиБ/с, готово.
Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/chizhpaul/my-first-github.git
 e38a8c9..b64d82e  dev -> dev
```

4. Переключитесь на основную ветку.

```
pavel@pavel-pk:~/my-first-github$ git checkout main
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
```

5. Добавьте файл main.sh в основной ветке с произвольным содержимым, сделайте комит и пуш . Так имитируется продолжение общекомандной

разработки в основной ветке во время разработки отдельного функционала в dev ветке.

```
pavel@pavel-pk:~/my-first-github$ nano main.sh
pavel@pavel-pk:~/my-first-github$ git add main.sh
pavel@pavel-pk:~/my-first-github$ git status
Текущая ветка: main
Ваша ветка опережает «origin/main» на 3 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)

Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    новый файл:    main.sh
```

6. Сделайте мердж dev ветки в основную с помощью `git merge dev`. Напишите осмысленное сообщение в появившееся окно комита.

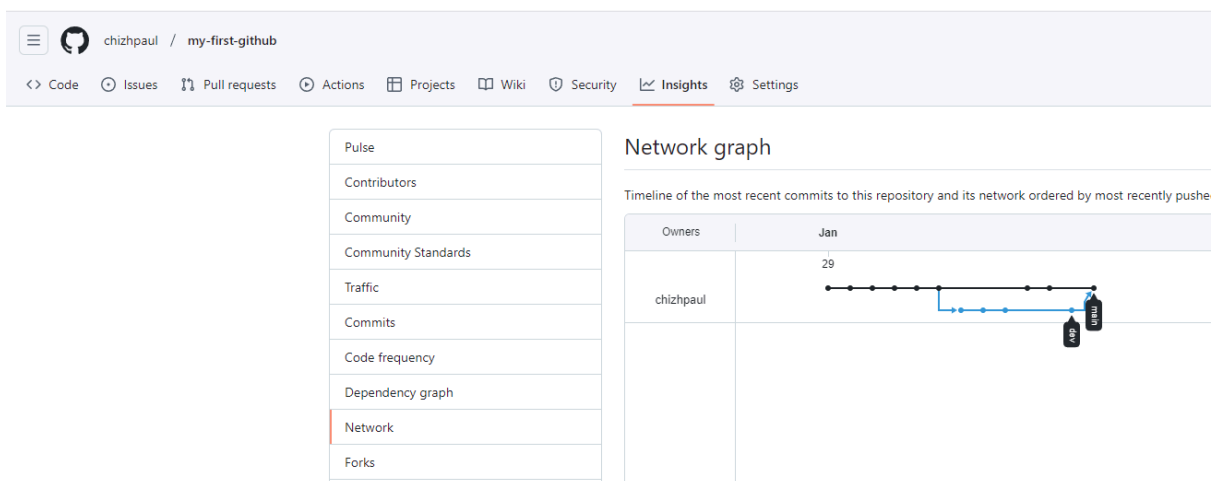
```
pavel@pavel-pk:~/my-first-github$ git merge dev
Обновление 3d7b231..b64d82e
Fast-forward
 test.sh | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 test.sh
```

```
pavel@pavel-pk:~/my-first-github$ git commit -m "merge dev"
[main d4e7f68] merge dev
 1 file changed, 1 insertion(+)
 create mode 100644 main.sh
```

7. Сделайте пуш в основной ветке.
8. Не удаляйте ветку dev.

В качестве ответа прикрепите ссылку на граф коммитов

<https://github.com/ваш-логин/ваш-репозиторий/network> в ваш md-файл с решением.



Ваш граф комитов должен выглядеть аналогично скриншоту:

