**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА**

**И ГОСУДАРСТВЕННОЙ СЛУЖБЫ**

**ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ – филиал РАНХиГС**

Факультет: управления

Кафедра информатики и информационных технологий

Направление подготовки/специальность: 09.03.03 Прикладная информатика

**ОТЧЕТ ПО ЛЕКЦИИ №1**

**по теме: «Освоение работы в GIT»**

**по курсу: «Программная инженерия»**

**АВТОР**

обучающийся 2 курса группы ИБ-321

очной формы обучения

Кокин Денис Игоревич

Нижний Новгород, 2024 г.

ОГЛАВЛЕНИЕ

[ОСВОЕНИЕ РАБОТЫ В GIT (GIT GUI), GITHUB, GITLAB 3](#__RefHeading___Toc278_2468248565)

[​ Назначение, возможности, особенности: 3](#__RefHeading___Toc280_2468248565)

[​ Сравнение (Git, GitHub vs GitLab): 3](#__RefHeading___Toc282_2468248565)

[​ Российская система GitFlick (возможности): 4](#__RefHeading___Toc284_2468248565)

[​ Команды системы GIT: названия и назначения: 5](#__RefHeading___Toc286_2468248565)

[ПРОЦЕСС РЕГИСТРАЦИИ НА GITHUB 6](#__RefHeading___Toc288_2468248565)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 10](#__RefHeading___Toc599_2626018907)

# **ОСВОЕНИЕ РАБОТЫ В GIT (GIT GUI), GITHUB, GITLAB**

## Назначение, возможности, особенности:

**GIT** — это система контроля версий, используемая в разработке программного обеспечения. Она позволяет отслеживать изменения в файлах и каталогах, а также управлять ими.

**Назначение GIT** заключается в обеспечении совместной работы над проектом, предотвращении конфликтов и обеспечении возможности отката к предыдущим версиям.

**Возможности GIT:**

1. Управление версиями: GIT позволяет сохранять различные версии проекта, что позволяет вернуться к предыдущей версии в случае возникновения проблем.
2. Управление изменениями: GIT отслеживает все изменения, внесенные в проект, и позволяет увидеть, кто и когда внес каждое изменение.
3. Совместная работа: GIT позволяет нескольким разработчикам работать над проектом одновременно, предотвращая конфликты и обеспечивая возможность отката к предыдущей версии при возникновении проблем.
4. Ветвление и слияние: GIT поддерживает создание ветвей для экспериментов или параллельной работы, а затем их слияние обратно в основную ветвь проекта.
5. Отказоустойчивость: GIT сохраняет все изменения в виде коммитов, что позволяет легко восстановить проект из любого момента времени.

**Особенности GIT:**

1. Распределенная система контроля версий: GIT не требует наличия центрального сервера, каждый разработчик имеет свою копию репозитория на своем компьютере.
2. Локальная история изменений: GIT хранит всю историю изменений на компьютере разработчика, что позволяет откатиться к любой версии проекта без необходимости обращения к серверу.
3. Легкость в изучении и использовании: GIT имеет простой и понятный интерфейс, что делает его доступным для начинающих разработчиков.

В целом, GIT является важным инструментом для обеспечения качества разработки программного обеспечения, управления проектами и совместной работы.

## Сравнение (Git, GitHub vs GitLab):

Git, GitHub и GitLab являются важными инструментами для разработчиков программного обеспечения и команд, занимающихся разработкой. Вот некоторые ключевые различия между ними:

**GIT:**

* Система контроля версий для отслеживания изменений в файлах проекта и управления ими.
* Распределенная система, в которой каждый разработчик работает со своей копией репозитория.
* Поддерживает ветви, позволяя разработчикам экспериментировать без риска для основной ветви проекта.
* Сохраняет историю всех изменений, что позволяет легко возвращаться к предыдущим состояниям проекта.

**GitHub:**

* Веб-сервис для хостинга репозиториев Git.
* Предоставляет платформу для совместной работы и общения между разработчиками.
* Имеет широкий набор функций, таких как обсуждение кода, управление задачами и отслеживание ошибок.
* Использует систему учетных записей и токенов для аутентификации и авторизации доступа к репозиториям.
* Предлагает бесплатную и платную версии обслуживания, с возможностью масштабирования по мере роста проекта.

**GitLab:**

* Еще один веб-сервис для хостинга и совместной работы с Git-репозиториями.
* В основном сфокусирован на управлении жизненным циклом кода и управлении задачами.
* Включает в себя функции, такие как отслеживание ошибок, управление доступом и аналитика.
* Также использует систему учетных записей для аутентификации, но вместо токенов использует роли и права доступа.
* Бесплатная версия предлагает практически те же возможности, что и GitHub, но платная версия имеет больше возможностей для масштабирования и интеграции.

## Российская система GitFlick (возможности):

**GitFlick —** это российская система управления версиями, которая предназначена для контроля версий и совместной работы над проектами. Она имеет следующие **возможности:**

* Управление версиями: GitFlick позволяет сохранять разные версии проекта, чтобы можно было вернуться к предыдущей версии, если возникли проблемы.
* Управление изменениями: система отслеживает все сделанные изменения и позволяет видеть, кто и когда их сделал.
* Совместная работа: несколько разработчиков могут работать над проектом одновременно, и GitFlick помогает предотвратить конфликты и вернуться к предыдущей версии.
* Ветвление и слияние: в GitFlick можно создавать ветки для экспериментов или параллельной работы и затем объединять их обратно в основной проект.
* Отказоустойчивость: система сохраняет все изменения как коммиты, что позволяет восстановить проект из любой точки времени.

## Команды системы GIT: названия и назначения:

**git clone:** используется для клонирования репозитория с удаленного сервера на локальный компьютер.

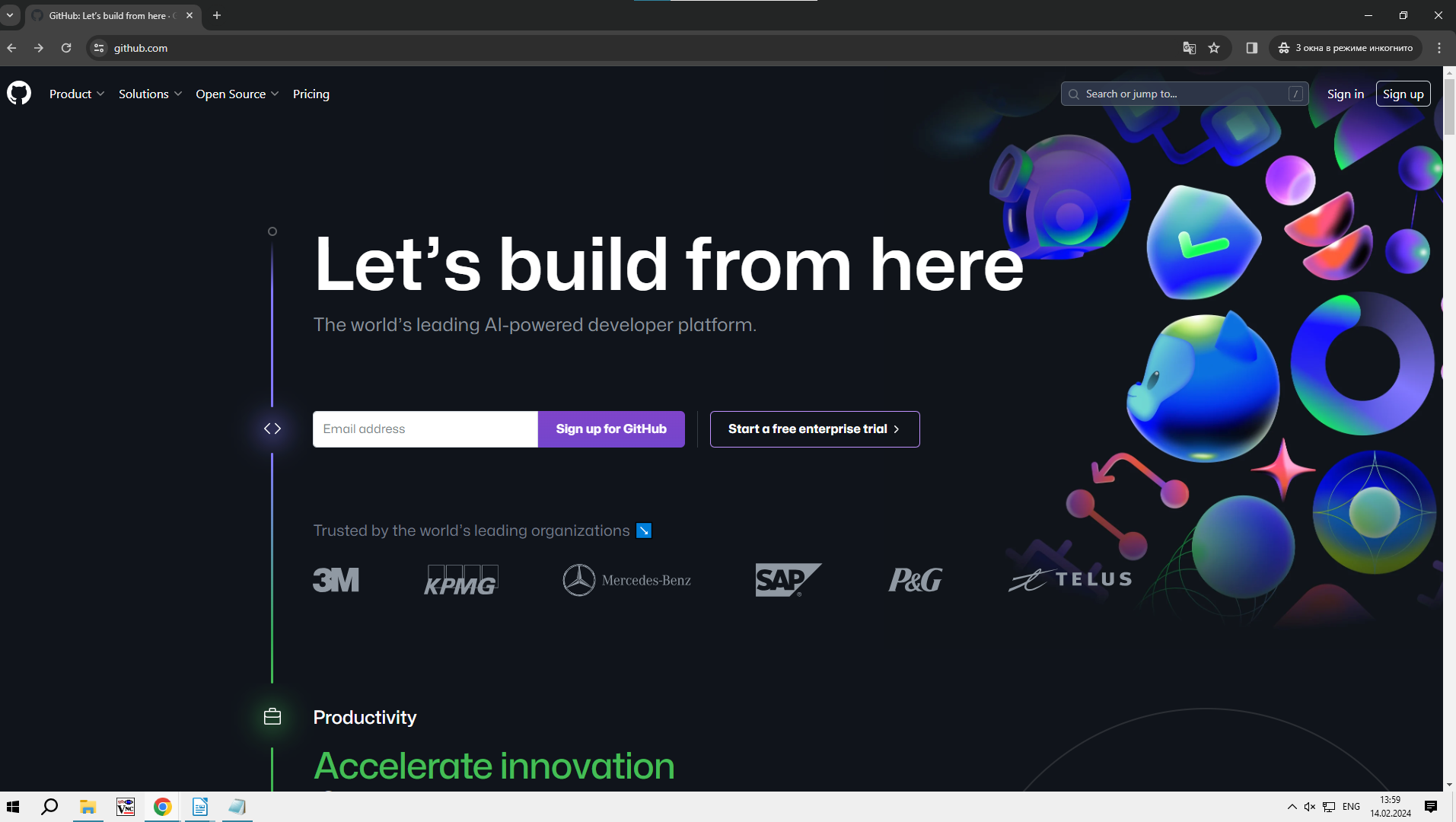
**git add:** добавляет файлы в индекс Git, готовясь к коммиту.

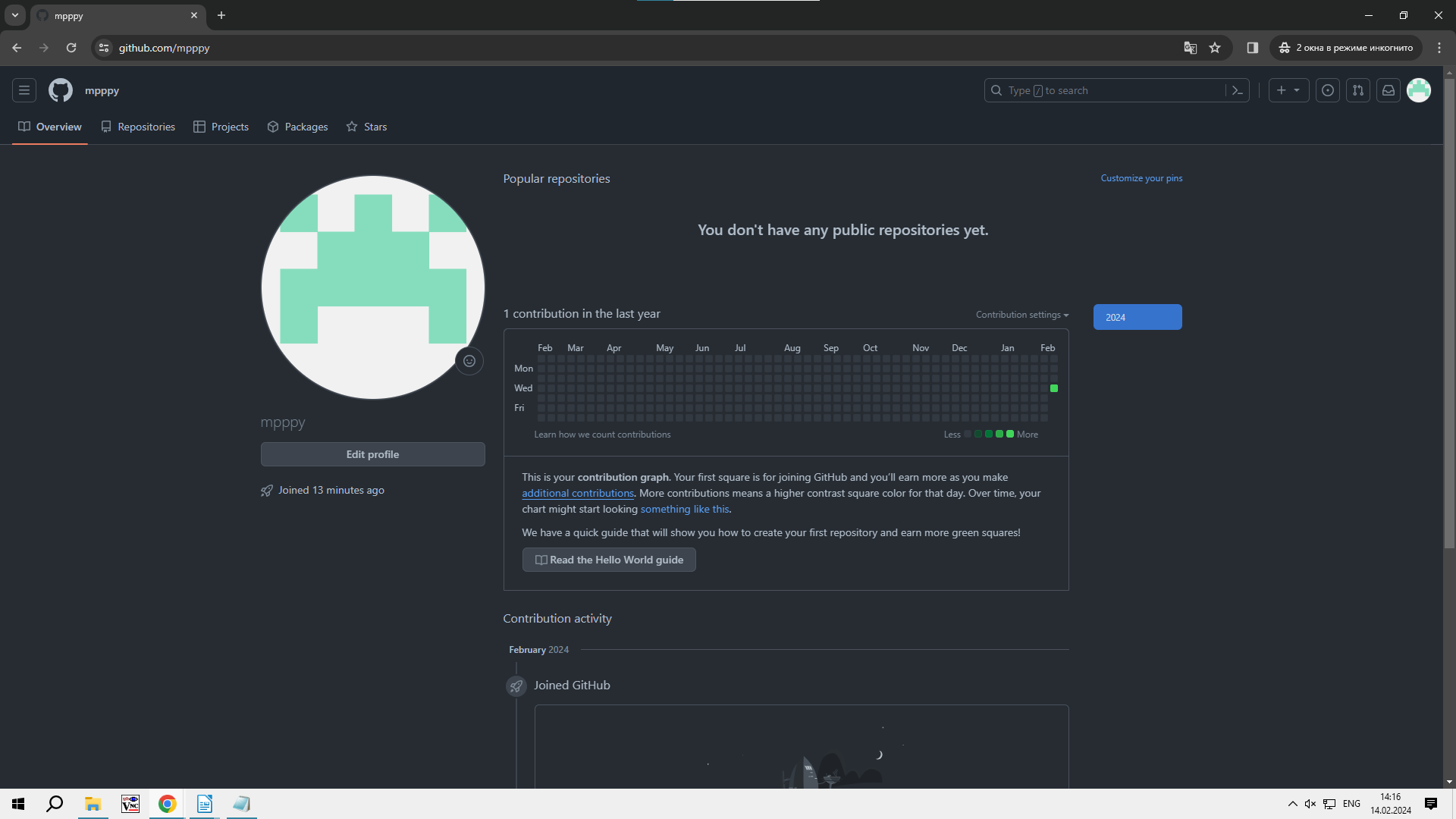
**git commit:** создает новый коммит, сохраняя изменения в репозитории.

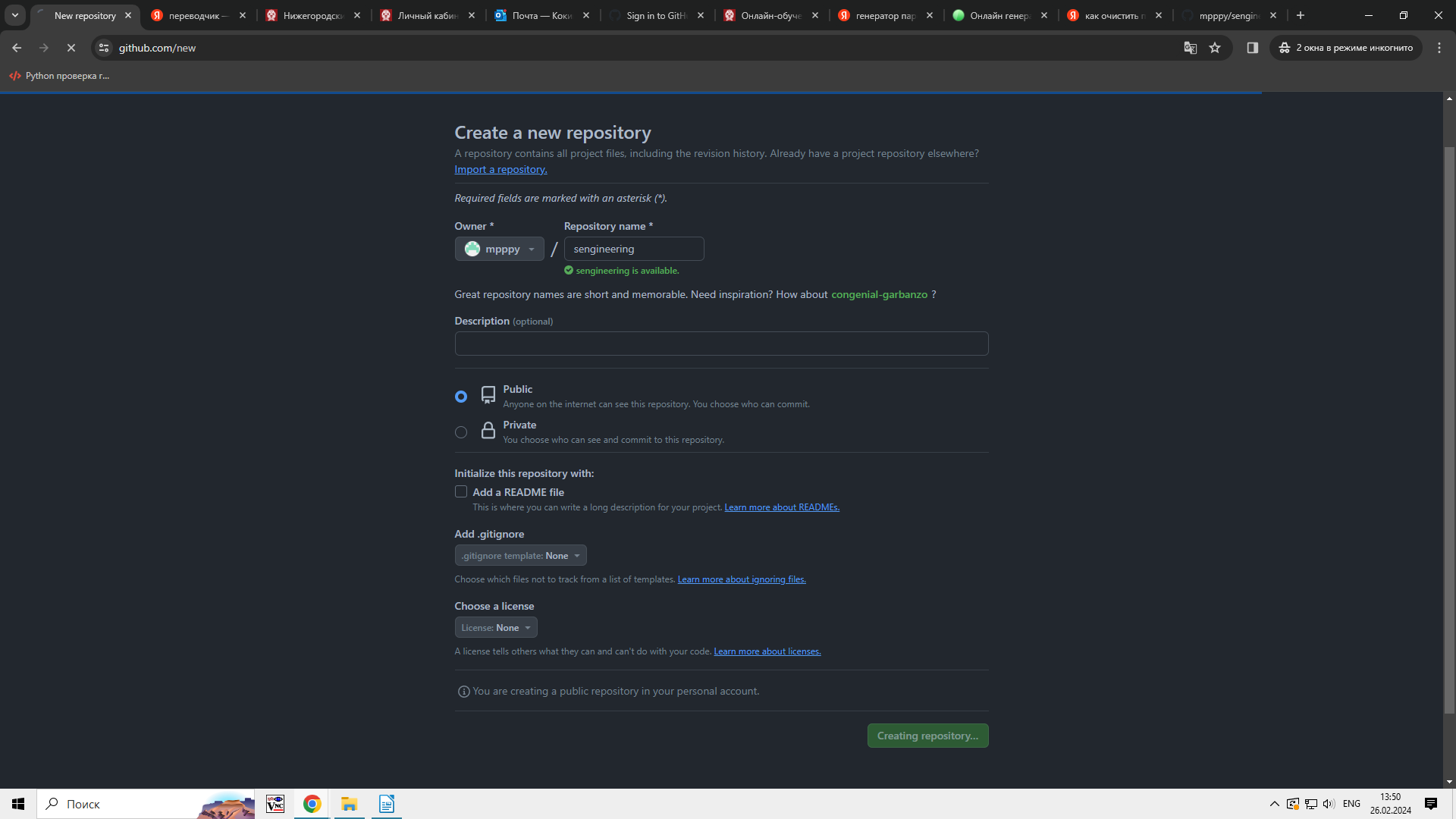
**git push:** отправляет коммиты с локальной машины на удаленный репозиторий.

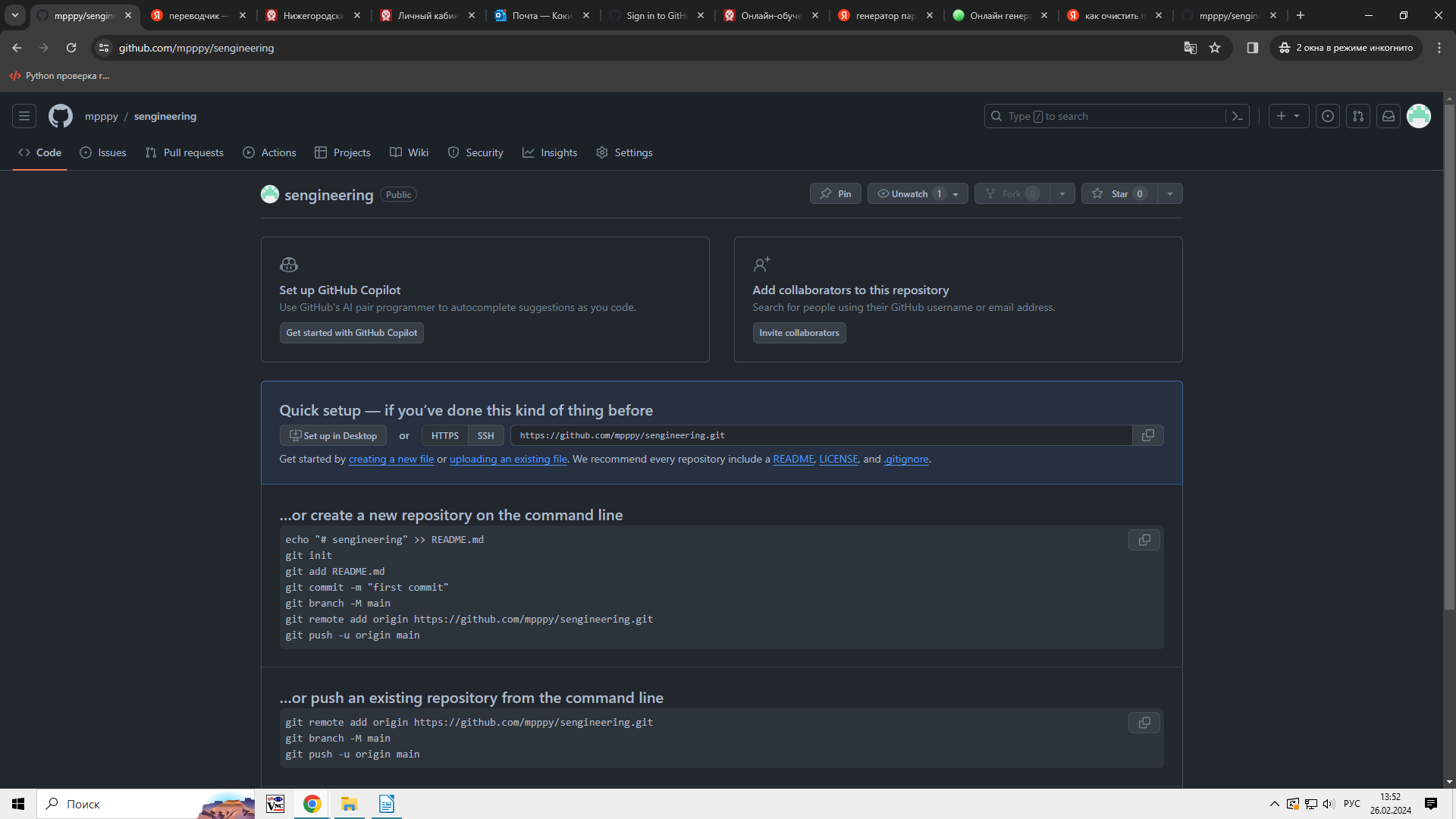
**git pull:** объединяет изменения из удаленного репозитория в локальный репозиторий.

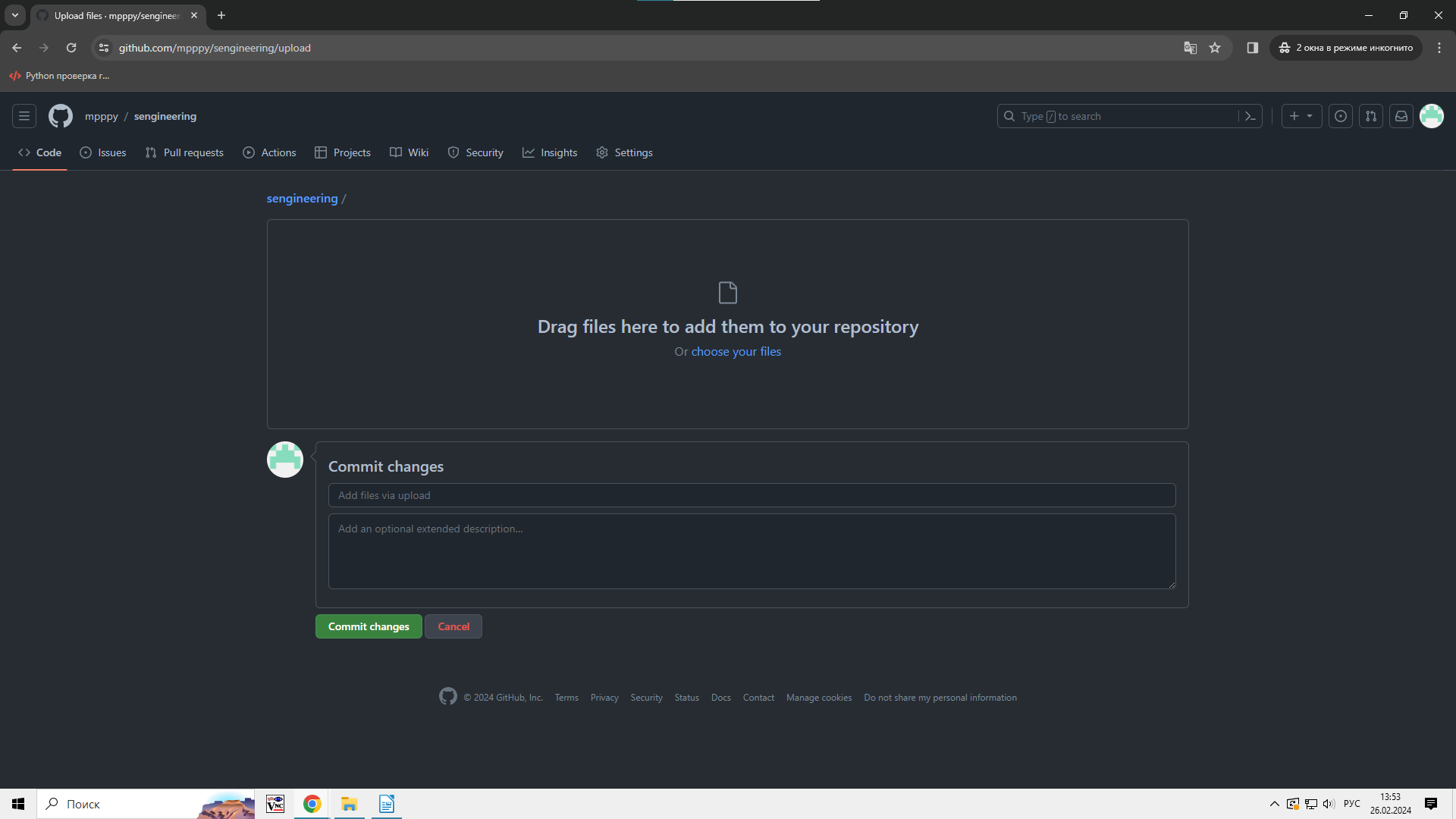
# **ПРОЦЕСС РЕГИСТРАЦИИ НА GITHUB**

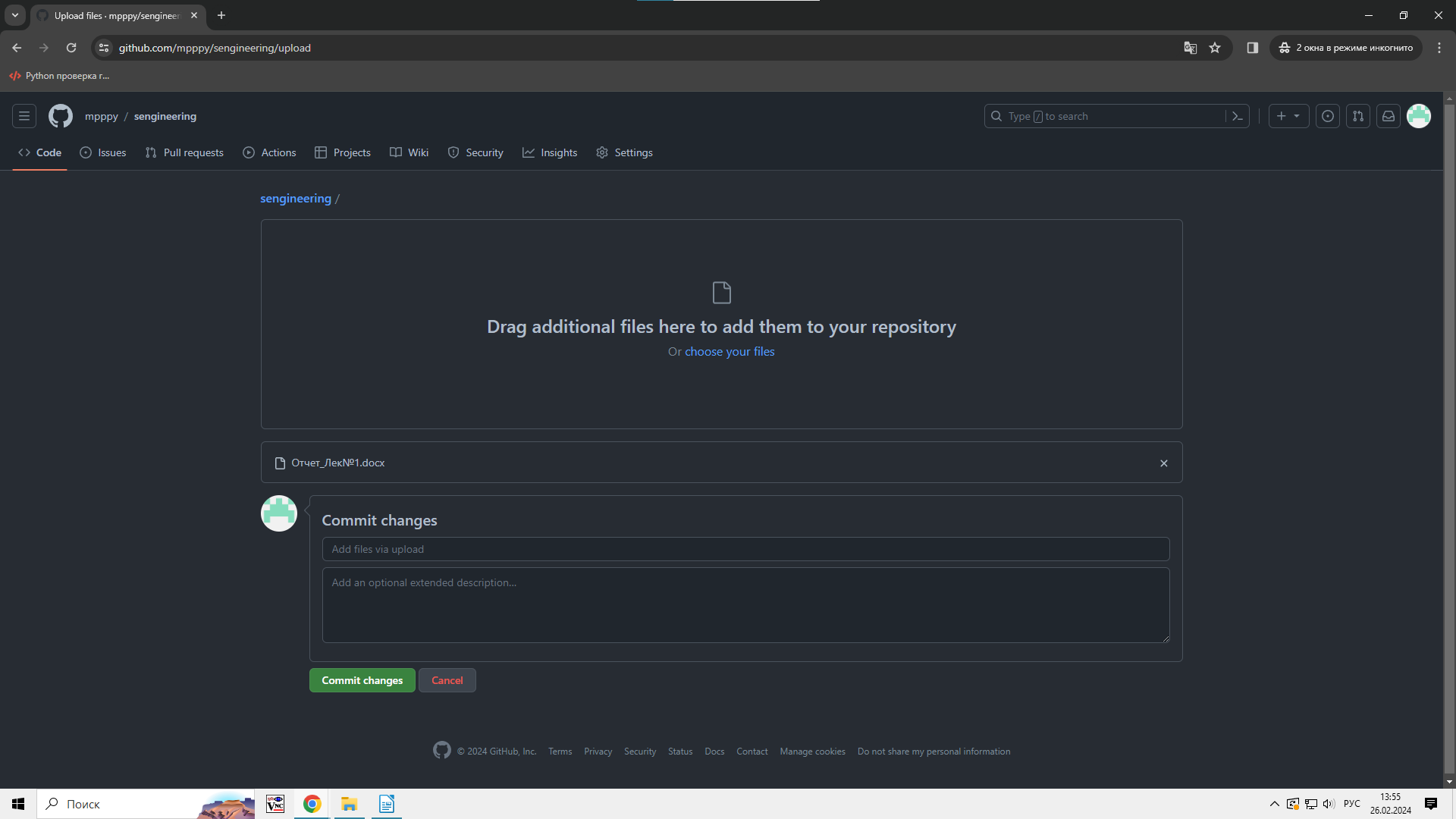
**Рисунок 1.** Главная страница github.com

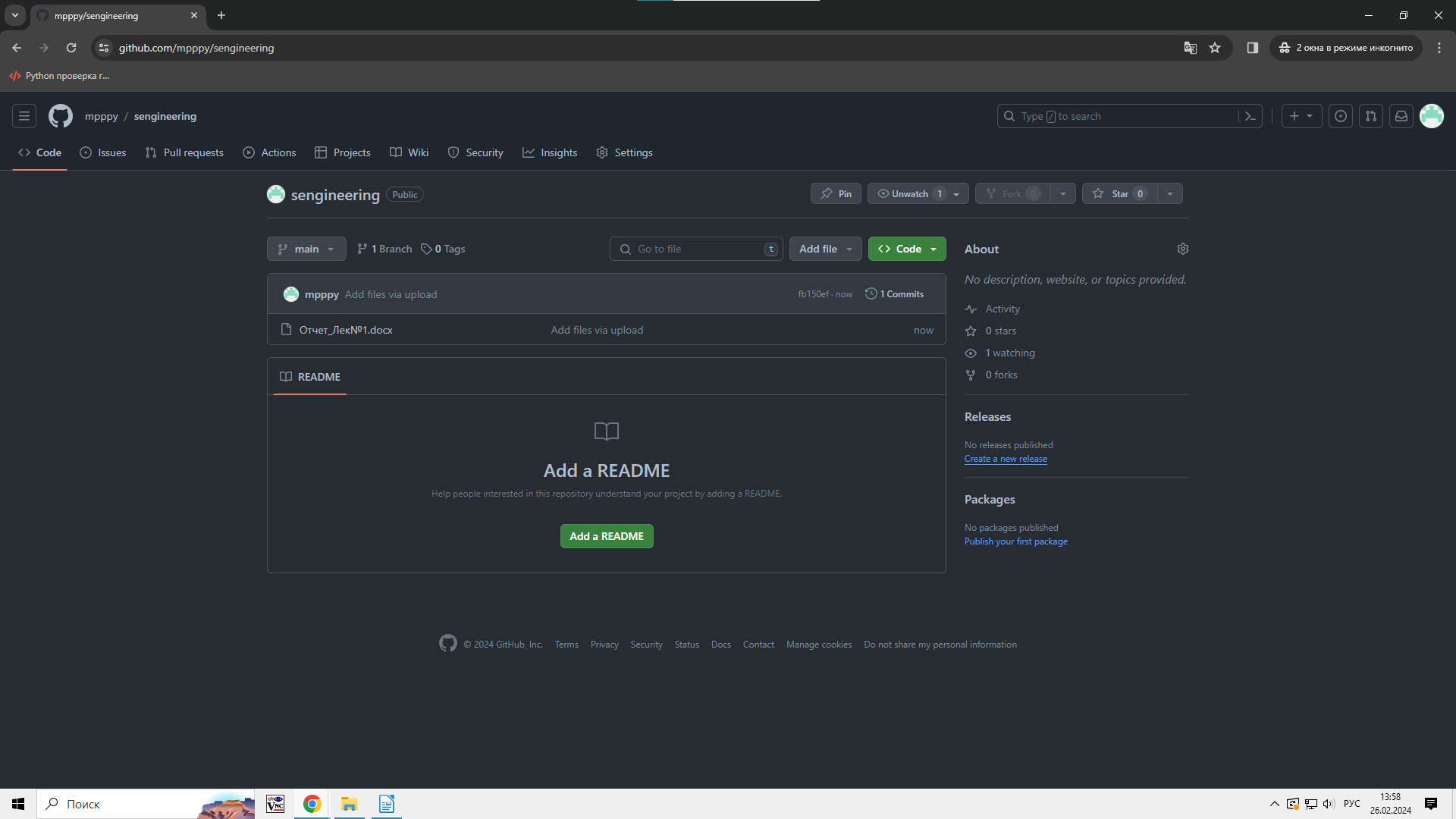
**Рисунок 2.**  Вид зарегистрированного пользователя на сайте github.com

**Рисунок 3.** Создание нового репозитория

**Рисунок 4.** Вид страницы репозитория

**Рисунок 5.** Добавление нового файла в репозиторий

**Рисунок 6.** Вид после прикрепления необходимого файла

Рисунок 7. Вид после добавления файла

# **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. [Яндекс.Практикум](https://practicum.yandex.ru/blog/chto-takoe-git-i-dlya-chego-nuzhen/" \l ":~:text=Git — это специальная программа%2C,выглядит файл в момент сохранения)
2. [Habr](https://habr.com/ru/articles/588801/)
3. [Trends.Rbc](https://trends.rbc.ru/trends/industry/622b8b4f9a7947053add4807)
4. [Book-git](https://book.git-scm.com/book/ru/v2/Приложение-C%3A-Команды-Git-Основные-команды)