Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации»

**Нижегородский институт управления**

Кафедра Информатики и информационных технологий

ОТЧЕТ

О ПРОДЕЛАННОЙ ЛАБОРАТОРНОЙ РАБОТЕ №1

по предмету «Программная инженерия»

Выполнил: студент группы ИБ-321

Доронова Дарья Валерьевна

Нижний Новгород

2024 г.

Лабораторная работа №1.

Цель: познакомиться с GitHab.

Задачи:

1. Ознакомиться с системой GitHub.
2. Зарегистрироваться на GitHub.
3. Освоить работу GitHub:

* Назначение, возможности, особенности;
* Сравнение (Git, GitHub vs GitLab);
* Российская система GitFlick (возможности);
* Команды системы Git: названия и назначение.

4. Написать отчет.

Оглавление:

[Задание 1. 3](#_Toc159368478)

[Задание 2. 4](#_Toc159368479)

[Задание 3. 5](#_Toc159368480)

[3.1 Назначение, возможности, особенности GitHub. 5](#_Toc159368481)

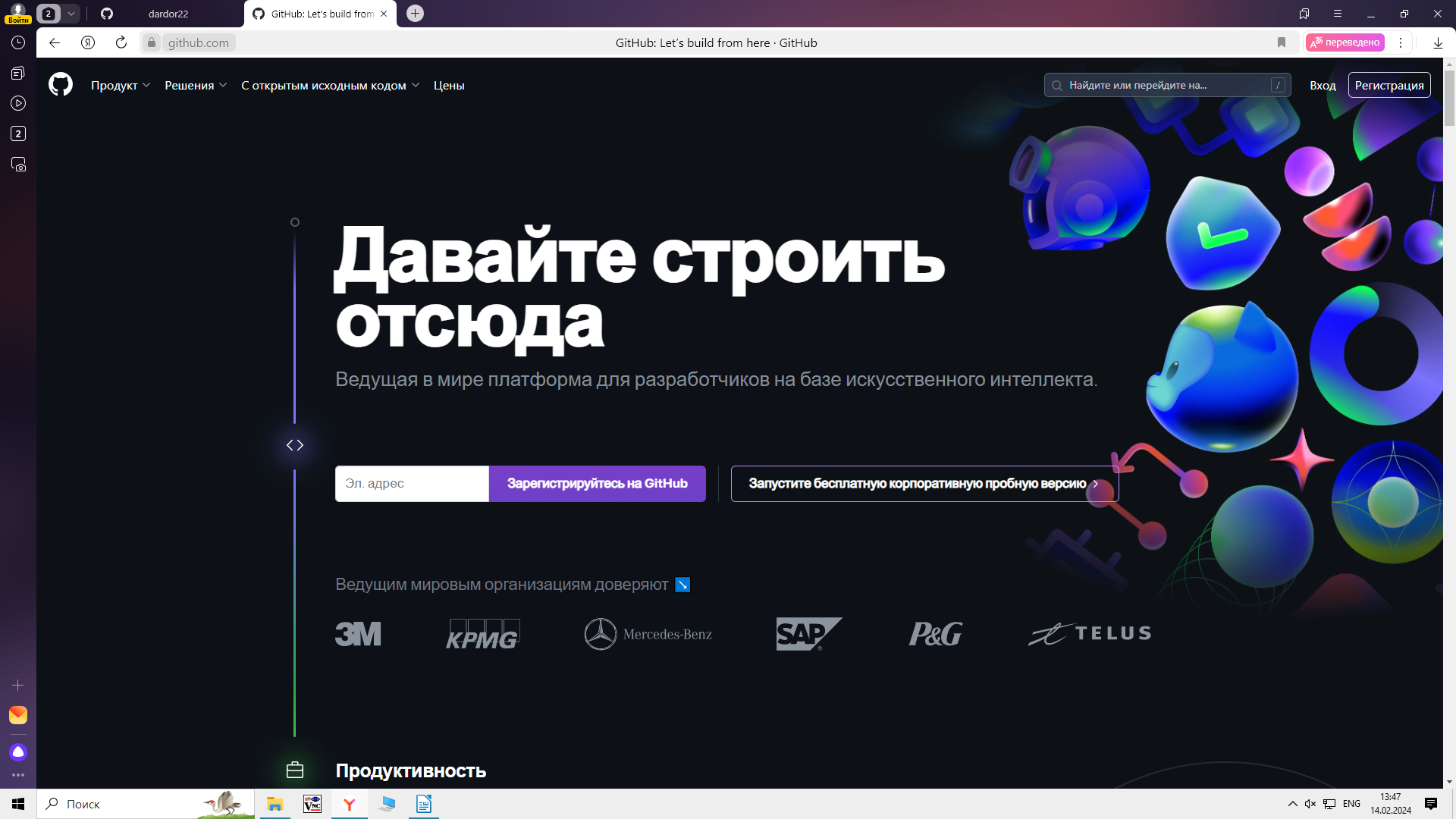
[3.2 Сравнение (Git, GitHub vs GitLab) 6](#_Toc159368482)

[3.3 Российская система GitFlick (возможности) 7](#_Toc159368483)

[3.4 Команды системы Git: названия и назначение. 8](#_Toc159368484)

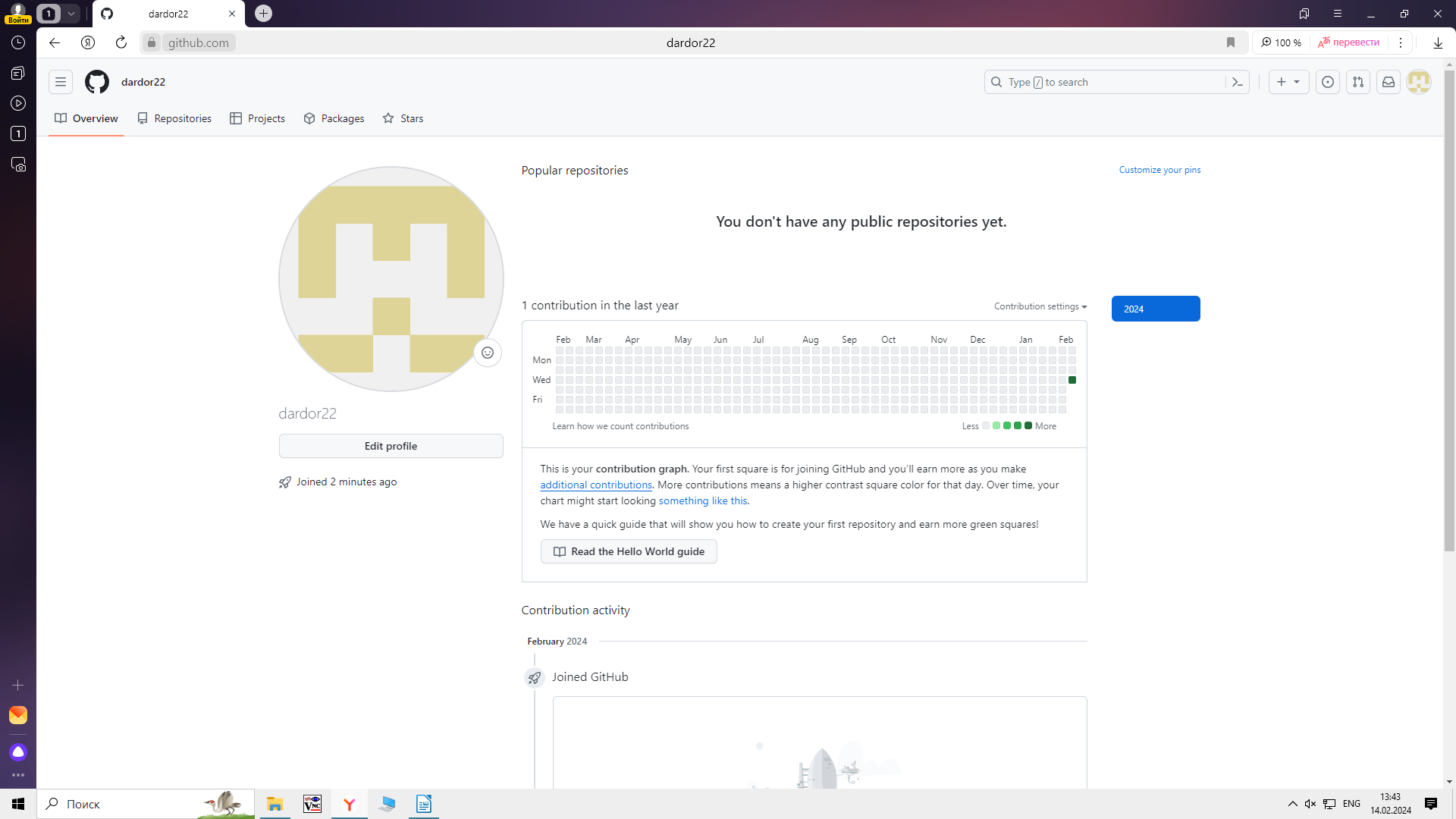
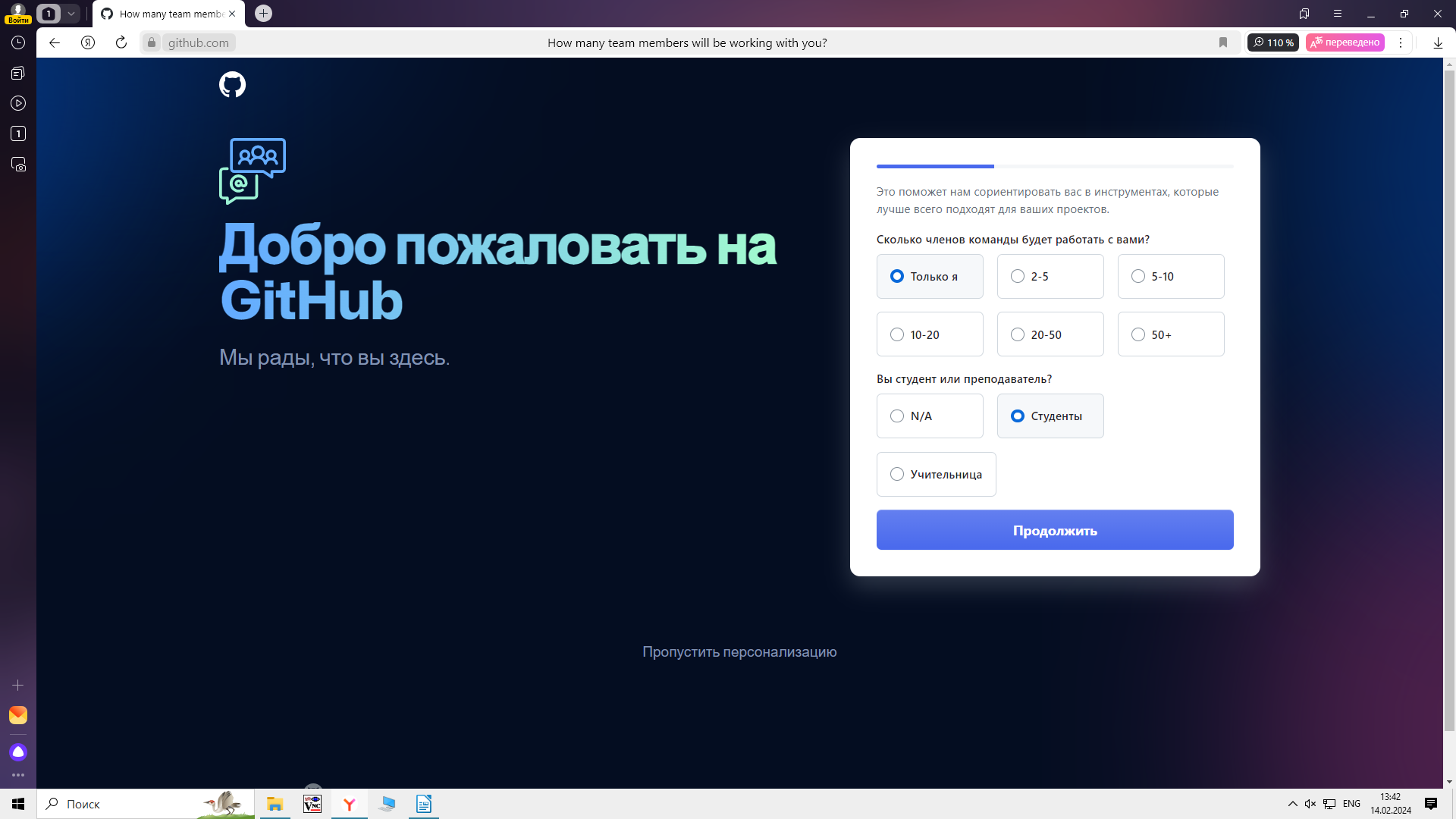
# Задание 1.

Ознакомиться с системой GitHub.



# Задание 2.

Зарегистрироваться на GitHub.



# Задание 3.

## 3.1 Назначение, возможности, особенности GitHub.

GitHub - это платформа для хостинга и совместной работы над проектами с использованием системы контроля версий Git. Назначение GitHub заключается в том, чтобы облегчить совместную разработку программного обеспечения, управление кодом и отслеживание изменений.  
  
*Основные возможности GitHub:*  
  
1. Хостинг репозиториев: GitHub предоставляет возможность хранить и управлять репозиториями Git на удаленных серверах.  
  
2. Коллаборация: GitHub позволяет нескольким разработчикам работать над одним проектом, внося изменения, предлагая исправления и комментируя код.  
  
3. Отслеживание изменений: GitHub позволяет отслеживать изменения в коде, просматривать историю коммитов, а также создавать ветки для разработки новых функций.  
  
4. Управление задачами: GitHub предоставляет инструменты для управления задачами и проектами, такие как Issues и Projects.  
  
5. Сборка и тестирование: GitHub интегрируется с различными инструментами для автоматической сборки, тестирования и развертывания кода.

GitHub имеет множество особенностей, которые делают его популярным среди разработчиков. Некоторые из ключевых особенностей GitHub:  
  
1. Репозитории: GitHub предоставляет возможность хранить репозитории Git на удаленных серверах, делая код доступным для совместной работы и совместного использования.  
  
2. Pull Requests: это механизм, который позволяет разработчикам предложить изменения в коде проекта и запросить их вливание в основную ветку. Pull requests облегчают рецензирование кода и обсуждение изменений.  
  
3. Issues: GitHub позволяет создавать задачи (issues) для отслеживания проблем, улучшений и других задач в проекте. Issues могут быть назначены разработчикам, обсуждаться и закрываться по мере выполнения.  
  
4. Вики-страницы: GitHub позволяет создавать вики-страницы для документации проекта, инструкций по использованию и другой полезной информации.  
  
5. GitHub Pages: С помощью GitHub Pages можно создавать статические сайты и размещать их на GitHub, что удобно для демонстрации проектов, документации или личных сайтов.  
  
6. Actions: GitHub Actions предоставляет возможность автоматизировать процессы сборки, тестирования и развертывания кода, что упрощает жизнь разработчиков.  
  
7. GitHub Desktop: GitHub предлагает удобное приложение для рабочего стола (GitHub Desktop), которое облегчает работу с репозиториями и контроль версий без необходимости использования командной строки.  
  
8. Интеграция с другими сервисами: GitHub интегрируется с множеством популярных сервисов и инструментов разработки, таких как Slack, Jira, Travis CI, Docker и другие.  
  
Эти особенности делают GitHub мощным инструментом для совместной разработки программного обеспечения, управления кодом и автоматизации процессов разработки.

## 3.2 Сравнение (Git, GitHub vs GitLab)

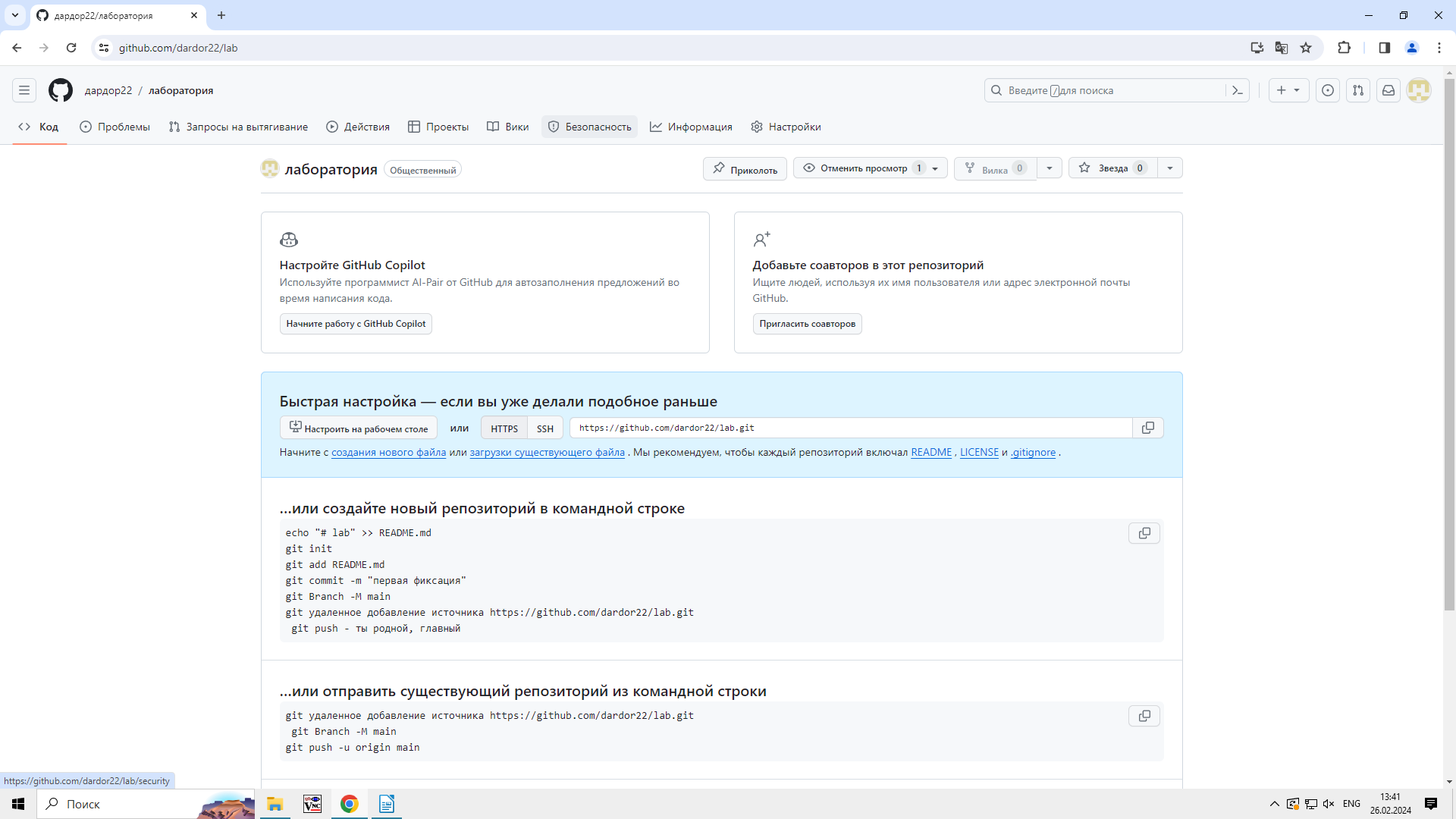
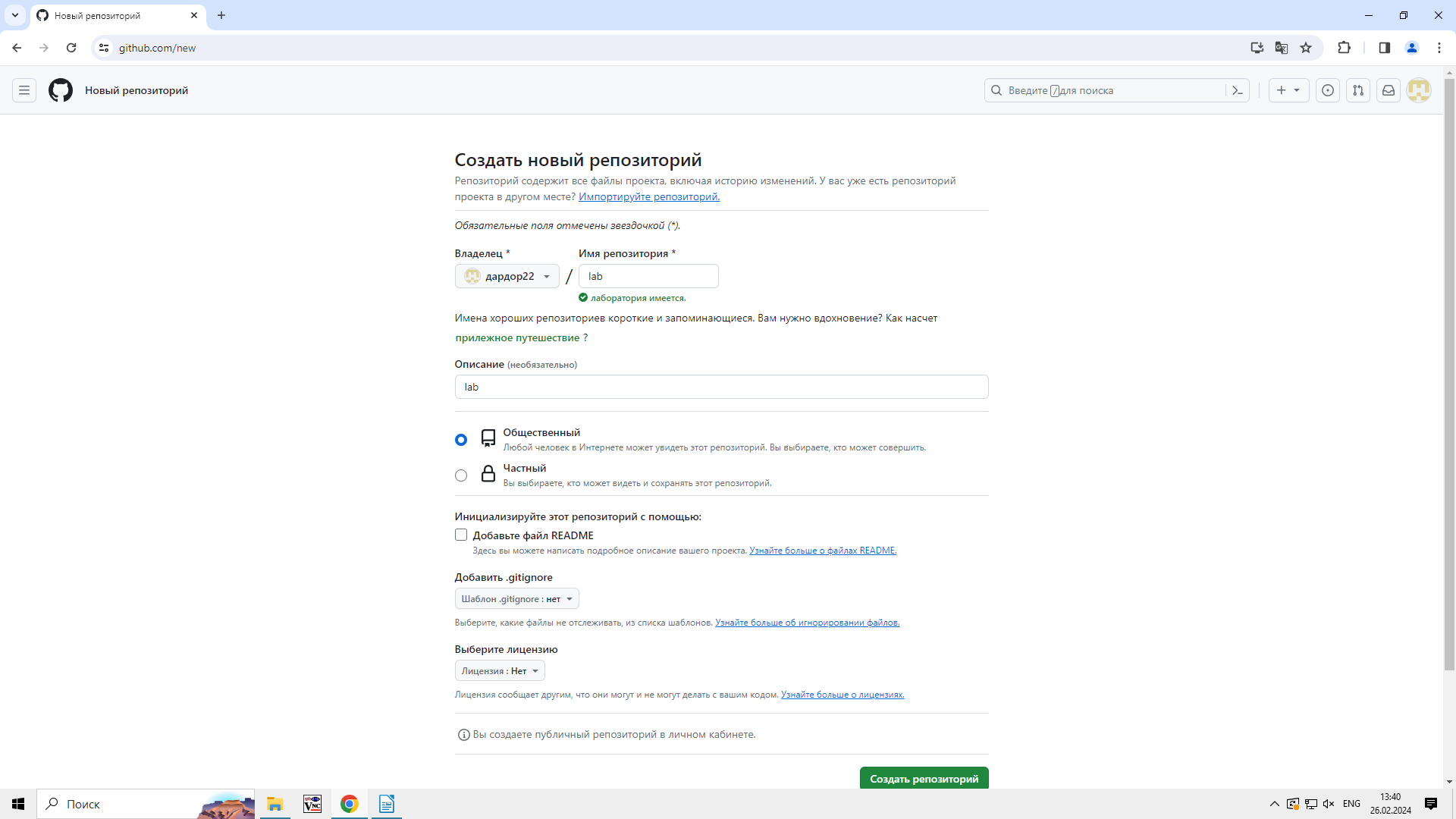
Git, GitHub и GitLab - это три популярных инструмента для управления версиями кода и совместной разработки. Вот некоторые ключевые различия между ними:  
  
1. Git: Git - это распределенная система управления версиями, разработанная Линусом Торвальдсом. Он предоставляет базовые функции для управления кодом, включая команды для создания репозиториев, фиксации изменений, ветвления и слияния.  
  
2. GitHub: GitHub — это веб-платформа для хостинга репозиториев Git и совместной работы над проектами. Он добавляет дополнительные функции, такие как pull requests, issues, вики-страницы, GitHub Pages, Actions и другие. GitHub позволяет разработчикам легко совместно работать над проектами и управлять кодом.  
  
3. GitLab: GitLab - это альтернатива GitHub, предоставляющая аналогичный набор функций для управления репозиториями Git. Однако GitLab также предлагает возможность самостоятельно установить его на собственном сервере (self-hosted), что делает его более гибким вариантом для организаций, которые предпочитают контроль над инфраструктурой.  
  
Некоторые другие различия между GitHub и GitLab:  
  
- Цена: GitHub предлагает бесплатные и платные тарифные планы, в то время как GitLab имеет как облачную версию с бесплатным и платным доступом, так и версию для самостоятельной установки.  
- Функциональность: GitHub и GitLab имеют некоторые различия в функциональности, такие как управление задачами, CI/CD интеграцию, управление приватными репозиториями и другие.  
- Сообщество: GitHub имеет широкое сообщество разработчиков и проектов, что делает его популярным выбором для открытых проектов. GitLab также имеет активное сообщество, особенно среди организаций, предпочитающих самостоятельную установку.  
  
В целом, выбор между GitHub и GitLab зависит от потребностей вашего проекта, предпочтений по функциональности и бюджета. Оба инструмента являются мощными средствами для совместной разработки программного обеспечения и управления кодом.

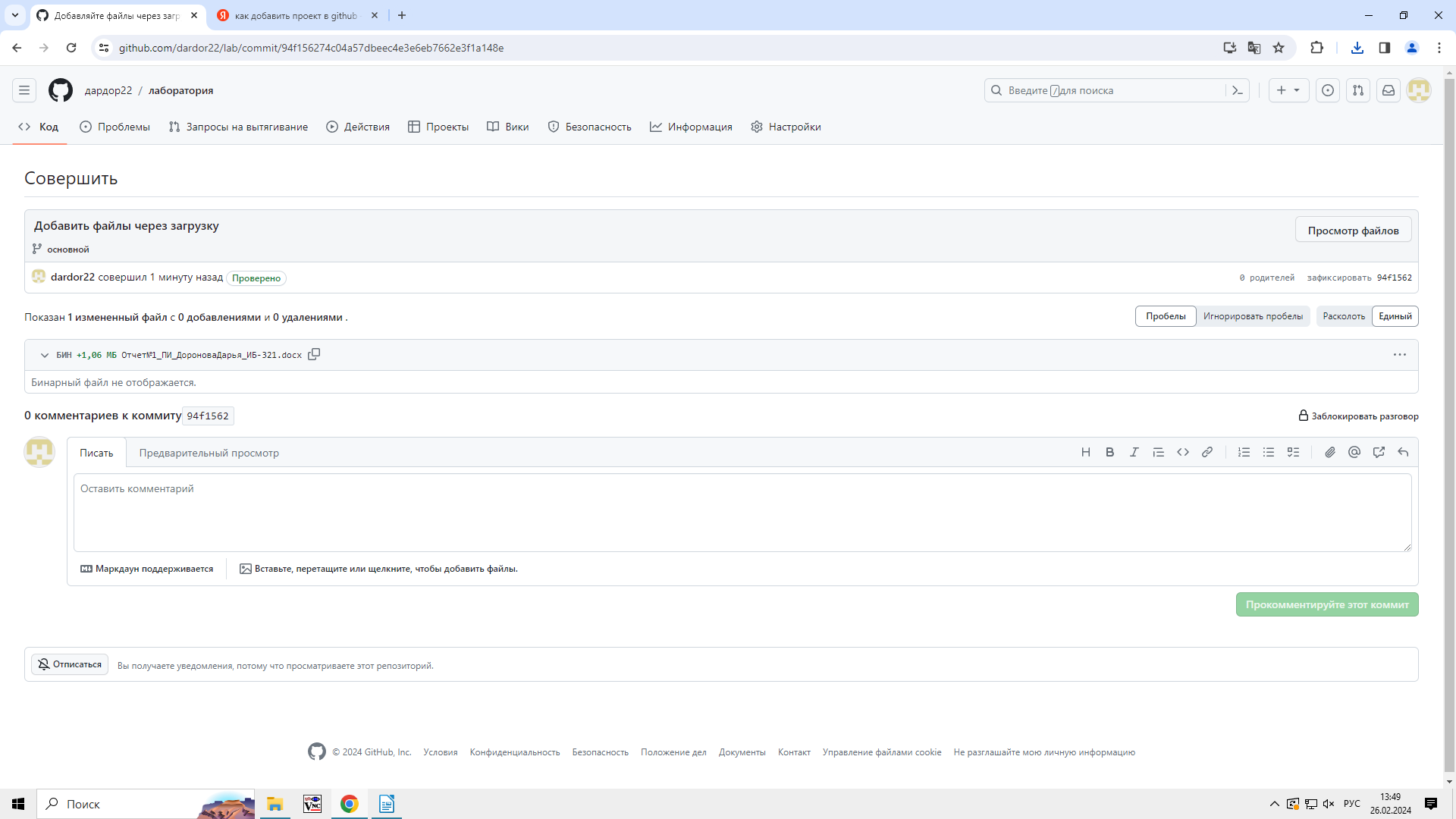
## 3.3 Российская система GitFlick (возможности)

GitFlick — это вымышленная система управления версиями, которая была создана для удовлетворения особых потребностей российских разработчиков. Вот некоторые возможности, которые могли бы быть включены в российскую систему GitFlick:  
  
1. Интеграция с российскими сервисами: GitFlick мог бы интегрироваться с популярными российскими сервисами, такими как Яндекс.Облако, Mail.ru Cloud Solutions, и другими, чтобы обеспечить более удобное хранение и управление кодом.  
  
2. Локализация интерфейса: GitFlick мог бы предоставлять локализованный интерфейс на русском языке для удобства пользователей из России.  
  
3. Учетная запись национального уровня: GitFlick мог бы предоставлять возможность использования учетной записи национального уровня, такой как Госуслуги или ЕСИА, для аутентификации пользователей.  
  
4. Защита данных: GitFlick мог бы предоставлять дополнительные механизмы защиты данных, соответствующие российским законодательным требованиям о хранении данных.  
  
5. Интеграция с российскими системами безопасности: GitFlick мог бы интегрироваться с российскими системами безопасности, такими как ФСТЭК или ФСБ, для обеспечения безопасности при работе с конфиденциальными данными.  
  
6. Поддержка криптографии: GitFlick мог бы предоставлять интеграцию с российскими криптографическими стандартами и алгоритмами для обеспечения безопасности передачи и хранения данных.  
  
7. Поддержка национальных стандартов: GitFlick мог бы поддерживать национальные стандарты разработки программного обеспечения, установленные в России.  
  
Это лишь некоторые возможные функциональности, которые могли бы быть включены в российскую систему GitFlick, чтобы соответствовать особенностям и требованиям российских разработчиков и компаний.

## 3.4 Команды системы Git: названия и назначение.

Git — это распределенная система управления версиями, которая предоставляет множество команд для работы с репозиторием. Вот некоторые из основных команд Git и их назначение:  
  
1. git init: Создает новый локальный репозиторий Git.  
  
2. git clone: Клонирует существующий удаленный репозиторий на локальную машину.  
  
3. git add: Добавляет файлы в индекс для последующего коммита.  
  
4. git commit: Фиксирует изменения в индексированных файлах и создает новый коммит.  
  
5. git push: Отправляет изменения из локального репозитория в удаленный репозиторий.  
  
6. git pull: Получает изменения из удаленного репозитория и объединяет их с локальными изменениями.  
  
7. git status: Показывает состояние рабочего каталога и индекса.  
  
8. git branch: Показывает список веток, создает новую ветку или удаляет существующую.  
  
9. git checkout: Переключается между ветками или восстанавливает файлы из предыдущих коммитов.  
  
10. git merge: Объединяет изменения из одной ветки в другую.  
  
11. git log: Показывает историю коммитов.  
  
12. git reset: Сбрасывает состояние индекса и рабочего каталога до определенного состояния.  
  
Это лишь некоторые из основных команд Git, которые помогают разработчикам эффективно управлять версиями кода и сотрудничать над проектами в распределенной среде.



D

Источник:

* https://ya.ru/