**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: 09.03.03 Прикладная информатика

Направленность (профиль): корпоративные информационные системы управления

**Лабораторная работа № 1**

|  |  |
| --- | --- |
| по дисциплине: | Программная инженерия |
| на тему: | «Освоение работы в Git, GitHab, GitLab» |
|  |  |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **АВТОР** | | | |
|  |  | Обучающийся 2 курса группы Ик-723 | | | |
|  |  | заочной формы обучения | | | |
|  |  |  | | | |
|  |  |  |  | | С.А. Кутырев |
|  |  | *(подпись) (фамилия, инициалы)* | | | |
|  |  |  | | | |
|  |  | **РУКОВОДИТЕЛЬ** | | | |
|  |  | кандидат технических наук, доцент | | | |
|  |  |  | | | |
|  |  | оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | |
|  |  | «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ г. | | | |
|  |  | *(дата защиты)* | | | |
|  |  |  | |  | В.И. Окулич |
|  |  | *(подпись) (фамилия, инициалы)* | | | |

Нижний Новгород, 2025 г.

**Целью данной лабораторной работы является** освоение работы в Git, GitHab, GitLab: сравнение ПО, изучение возможностей и особенностей.

**Теоретическая часть**

**Освоение работы в Git, GitHab, GitLab. Назначение, возможности, особенности**

**Git** – Распределенная система контроля версий (VCS), позволяющая отслеживать изменения в коде, управлять историей правок и работать в команде 56.

**Возможности:**

* Ветвление (branching) и слияние (merging).
* Откат изменений (revert, reset).
* Работа с локальными и удаленными репозиториями.

**Особенности:**

* Не зависит от центрального сервера (каждый разработчик имеет полную копию репозитория).
* Высокая скорость работы даже с большими проектами.

**GitHub** - Облачная платформа для хостинга Git-репозиториев, совместной разработки и управления проектами.

**Возможности:**

* Pull Requests (запросы на слияние).
* GitHub Actions (CI/CD).
* GitHub Pages (хостинг статических сайтов).

**Особенности:**

* Огромное open-source сообщество.
* Интеграция с Jira, Slack, Docker Hub.

**GitLab –** Полноценная DevOps-платформа с Git-хостингом, CI/CD, управлением задачами и безопасностью.

**Возможности:**

* Встроенный CI/CD (без необходимости сторонних инструментов).
* Kubernetes-интеграция.
* Гибкие настройки доступа (ролевая модель).

**Особенности:**

* Поддержка приватных репозиториев даже в бесплатной версии.
* Возможность self-hosting (развертывание на своих серверах

**Сравнение (Git, GitHub vs GitLab)**

| **Критерий** | **Git (локальный)** | **GitHub (облачный)** | **GitLab (облачный/self-hosted)** |
| --- | --- | --- | --- |
| **Основное назначение** | Контроль версий | Хостинг + соц. сеть для разработчиков | Полный DevOps-цикл (от кода до деплоя) |
| **CI/CD** | Нет | GitHub Actions (интеграция) | Встроенный GitLab CI/CD |
| **Приватные репозитории** | Да (локально) | Бесплатно (ограничения) | Бесплатно (полный доступ) |
| **Self-hosting** | Нет | GitHub Enterprise (платно) | Бесплатно (Community Edition) |
| **Сообщество** |  | Огромное (open-source) | Меньше, но растет |
| **Интеграции** |  | Jira, Slack, Docker | Kubernetes, Prometheus, Grafana |

**Вывод:**

* Git подходит для личных проектов, когда нет необходимости обмениваться обновлениями с другими разработчиками.
* GitHub лучше для open-source и портфолио.
* GitLab удобнее для корпоративных проектов и CI/CD

**Российская система GitFlic**

**GitFlic –** Российский аналог GitHub/GitLab для хостинга Git-репозиториев.

**Возможности:**

* Бесплатные приватные репозитории.
* Поддержка SSH-ключей (аналогично GitHub).
* Интеграция с российскими сервисами (например, СБИС).

**Особенности:**

* Меньше пользователей по сравнению с GitHub/GitLab.
* Акцент на соответствие российскому законодательству.

**Команды системы Git: названия и назначение**

**Базовые команды**

* git init – создает новый репозиторий 5.
* git clone <url> – клонирует удаленный репозиторий.
* git add <file> – добавляет файлы в staging area.
* git commit -m "message" – фиксирует изменения.

**Работа с ветками**

* git branch <name> – создает ветку.
* git checkout <branch> – переключается на ветку.
* git merge <branch> – сливает ветки.

**Синхронизация с удаленным репозиторием**

* git pull – забирает изменения с сервера.
* git push – отправляет изменения на сервер.

**Отмена изменений**

* git reset --hard – откатывает все изменения.
* git revert <commit> – создает новый коммит, отменяющий старый.

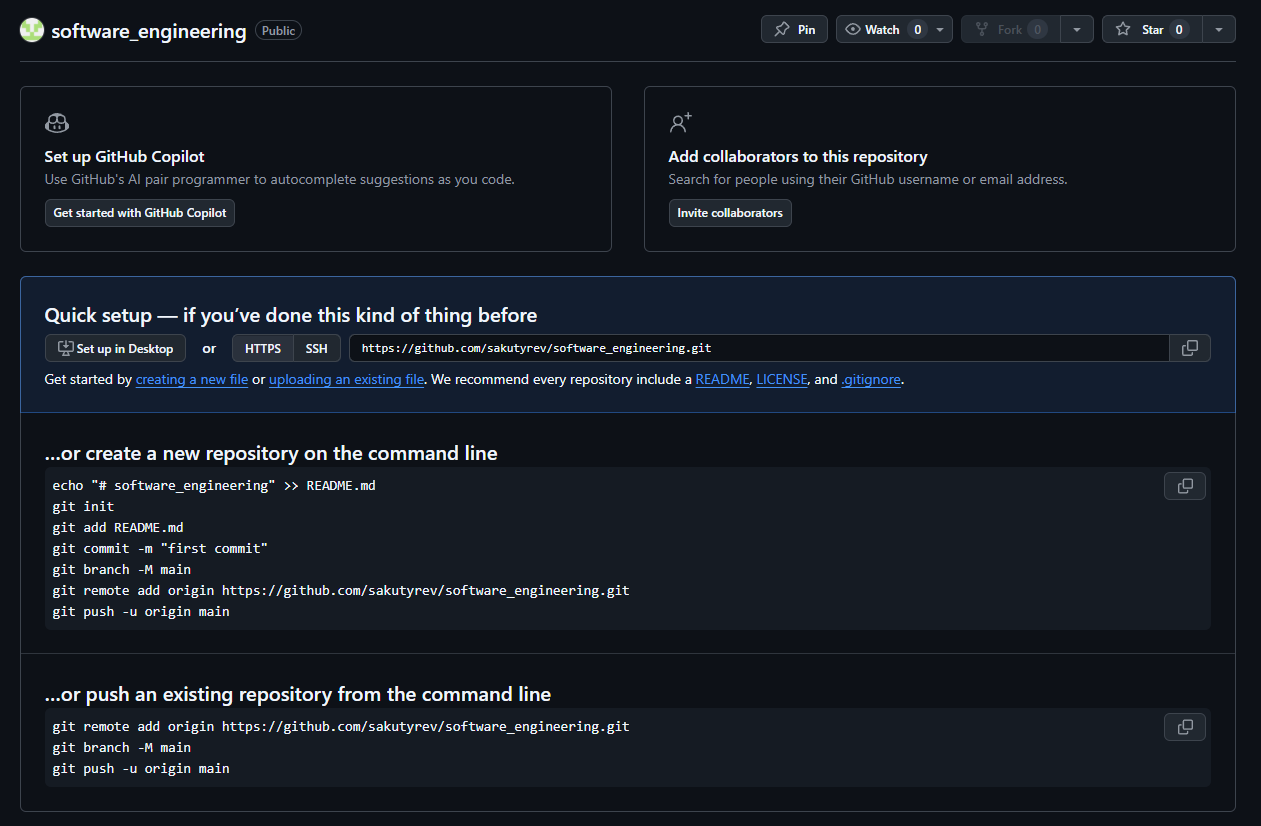
**Просмотр истории**

* git log – показывает историю коммитов.
* git diff – показывает изменения между коммитами

**Практическая часть**

**Зарегистрироваться в одном из репозиториев и записать в них отчёты по лабораторным работам**

Был создан публичный репозиторий в GitHub. Достуный по ссылке <https://github.com/sakutyrev/software_engineering> . На рисунке 1 представлен интерфейс репозиторияю



**Рис. 1. Интерфейс GitHub**

Для того, чтобы загрузить какие-либо файлы в репозиторий необходимо сначала создать локальный git репозиторий в выбранной директории.

Для создания локального репозитория, а затем подключения его к удаленному, необходимо выполнить следующие команды:

git init

git add README.md

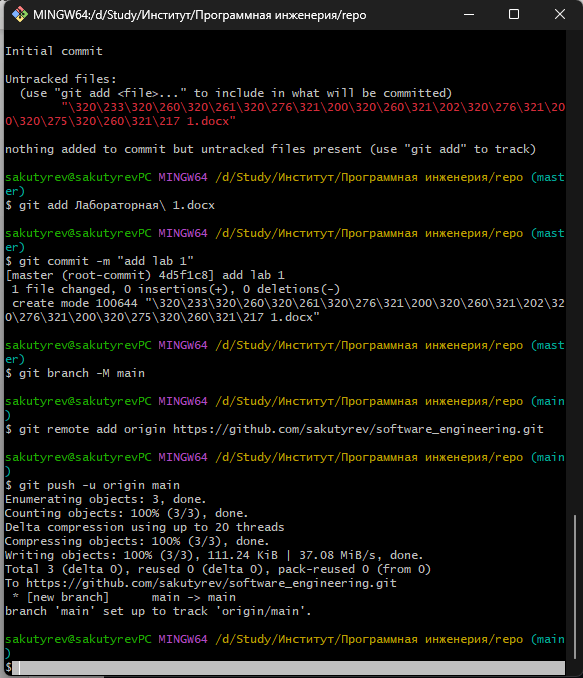
git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/sakutyrev/software\_engineering.git

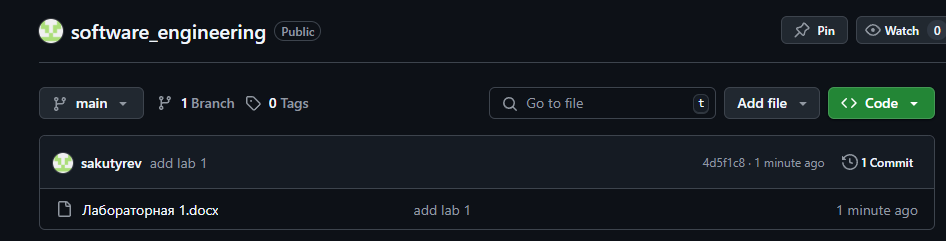
git push -u origin main

На рисунке 2 показана часть выполненных команд.



**Рис. 2. Процесс создания репозитория**

В итоге выполнения команд отчет был добавлен в репозиторий. На рисунке 3 можно видеть добавленный в репозиторий отчет.



**Рис 3. Содержимое GitHub**