

Лабораторная работа №16 (2 ч.)

Публикация проекта на GitHub Pages

Цель работы: научиться публиковать веб-проект с помощью сервиса GitHub Pages, освоить базовые операции с репозиторием на GitHub (создание, загрузка файлов, настройка источника публикации), а также применить практики работы с ветками и историей изменений. Способствовать формированию соответствующих общих компетенций (ОК 01, ОК 02, ОК 03, ОК 04) и профессиональных компетенций (ПК 9.7, ПК 9.9, ПК 9.10).

Средства выполнения работы:

- аппаратные: ПК;
- СЭДО СВФУ (Moodle) - Портал электронного обучения СВФУ 09.02.07 Информационные системы и программирование, Квалификация: Разработчик веб и мультимедийных приложений МДК.09.02 Оптимизация веб-приложений;
- Редакторы кода/среды для веб-разработки (прописать свой, например VSCode);
- Браузер (прописать свой);

I. Теоретическая часть

1. [Документация GitHub](#)
2. Материалы из лекции по теме лабораторной работы.

Что такое GitHub Pages

- Бесплатный хостинг статических сайтов от GitHub.
- Поддерживает HTML, CSS, JavaScript.
- Автоматически публикует содержимое указанной ветки (обычно main или gh-pages).

Основные понятия Git и GitHub

- Репозиторий (repository) — хранилище проекта со всей историей изменений.
- Коммит (commit) — фиксация изменений с описанием.
- Ветка (branch) — параллельная версия кода для разработки без влияния на основную (main) ветку.
- Pull request — запрос на слияние изменений из одной ветки в другую.
- GitHub Pages settings — настройка источника (branch + folder) для публикации.

Процесс публикации сайта на GitHub Pages

- Создание репозитория.
- Загрузка файлов проекта.
- Настройка публикации через Settings → Pages.
- Ожидание сборки и проверка работоспособности по автоматически сгенерированной ссылке.

Преимущества использования GitHub Pages в образовательных и демонстрационных целях

- Бесплатно, просто, интегрировано с Git.
- Позволяет быстро делиться работами, портфолио, учебными проектами.



II. Правила выполнения заданий:

1. Ознакомиться с теоретическим материалом по теме лабораторной работы (*источники с блока теор. части, пояснения преподавателя, материал с лекций*).
2. Выполнить задания из раздела IV. Задания.
3. Оформить отчет. Отчет должен содержать:
 - 1) Титульный лист;
 - 2) Цель лабораторной работы;
 - 3) Задачи;
 - 4) Средства для выполнения работы (аппаратные/программные);
 - 5) Теоретическую сводку (ответы на контр. вопросы (если есть), основные понятия, применяемые в работе/при наличии теор. блок);
 - 6) Описание хода работы со скриншотами, **ссылка на опубликованный сайт и репозиторий;**

- 7) Результаты проделанной работы;
- 8) Выводы по проделанной работе.
4. В СЭДО загрузить текстовый файл отчета.

III. Задание

- 1) Опубликовать созданный в лабораторной работе №15 сайт, используя GitHub Pages.
- 2) Редактировать сайт-пример из прикрепления в СДО в побочной ветке, посмотреть историю версий, провести слияние кода из побочной ветки в основную.

Ход работы:

Задание 1. Публикация сайта на GitHub Pages

1. Пройдите регистрацию на GitHub (если еще нет аккаунта).

Следуйте инструкции из офиц. документации GitHub:

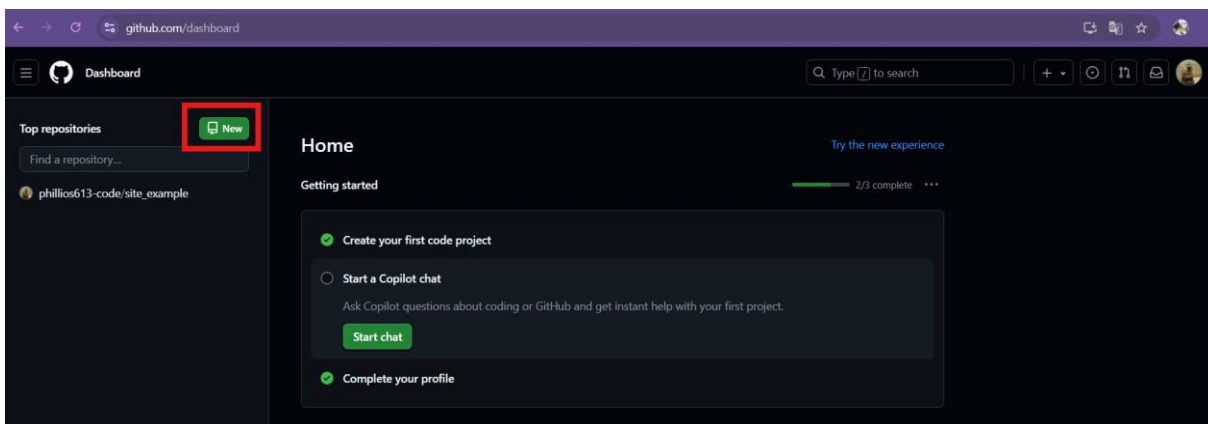
<https://docs.github.com/ru/get-started/start-your-journey/creating-an-account-on-github#signing-up-for-a-new-personal-account>

Официальная документация на английском языке, пользуйтесь встроенным переводчиком в браузере.
Дополнительно: для начального ознакомления можете пройти урок/гайд Hello world!

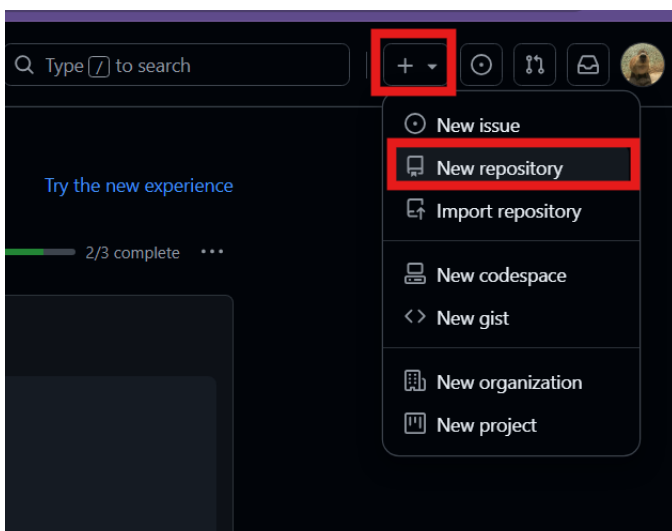
<https://docs.github.com/ru/get-started/start-your-journey/hello-world> - ссылка на урок

2. Создайте новый репозиторий с именем (сайта), например **my-shop**.

Создать репозиторий можно, кликнув на кнопку “New”, в боковой панели в окне главной страницы,



или кликнув на плюсики вверху, выбрать пункт в меню “new repository”.



Заполните данные (название репозитория, краткое описание, настройки приватности, наличие специальных файлов в создаваемом репозитории). После внесения данных нажмите «Create repository».

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

1 General

Owner * phillios613-code / Repository name * git-site-example
git-site-example is available.

Great repository names are short and memorable. How about [glowing-waddle?](#)

Description

29 / 350 characters

2 Configuration

Choose visibility *
Choose who can see and commit to this repository Public

Add README
READMEs can be used as longer descriptions. [About READMEs](#) Off

Add .gitignore
.gitignore tells git which files not to track. [About ignoring files](#) No .gitignore

Add license
Licenses explain how others can use your code. [About licenses](#) No license

Create repository

3. Реализуйте добавление файлов проекта на созданный репозиторий в GitHub. Для этого нажмите выделенную ссылку.

phillios613-code / git-site-example

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

git-site-example Public

Pin Watch 0 Fork 0 Star 0

Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
[Get started with GitHub Copilot](#)

Add collaborators to this repository
Search for people using their GitHub username or email address.
[Invite collaborators](#)

Quick setup — if you've done this kind of thing before

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/phillios613-code/git-site-example.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# git-site-example" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/phillios613-code/git-site-example.git
git push -u origin main
```

Выберите файлы проекта сайта на ПК (медиа сайта, файлы html, css и js). Путем перетаскивания файлов в окошко или выбора “choose your files”.

git-site-example /

Drag files here to add them to your repository
[Or choose your files](#)

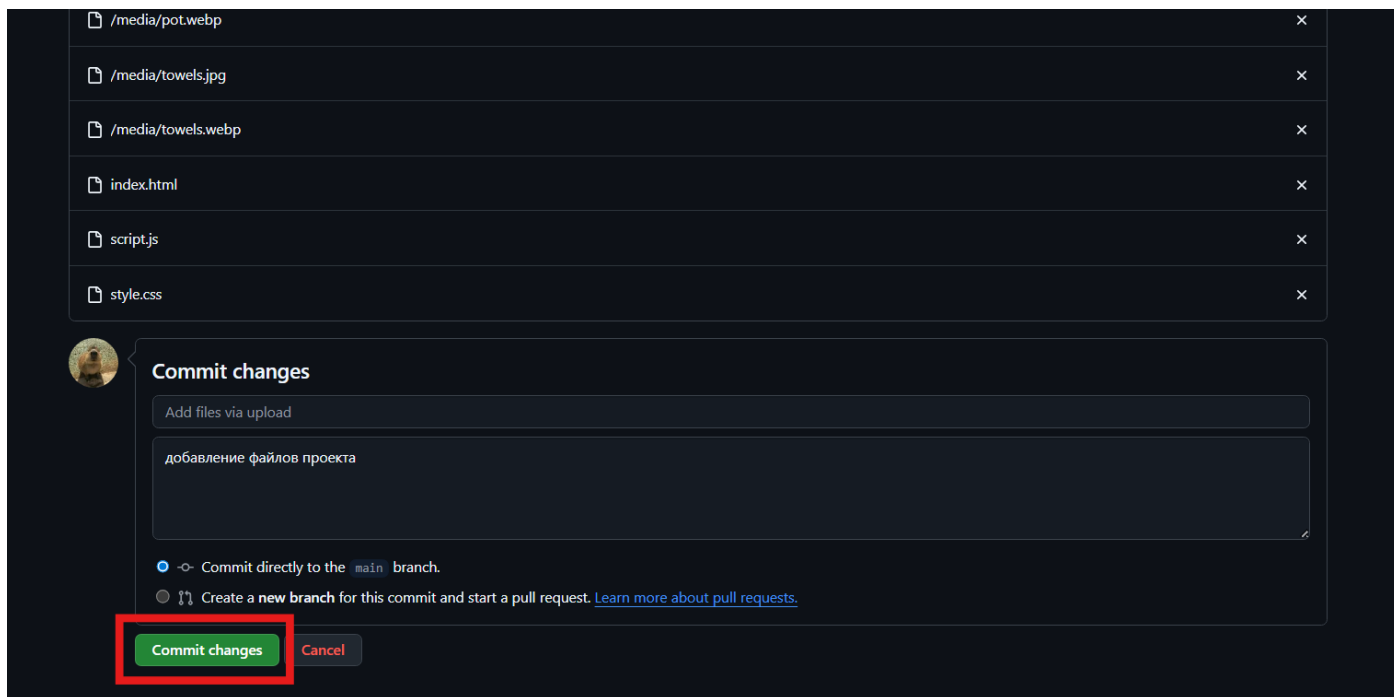
Commit changes

☒ Commit directly to the `main` branch.

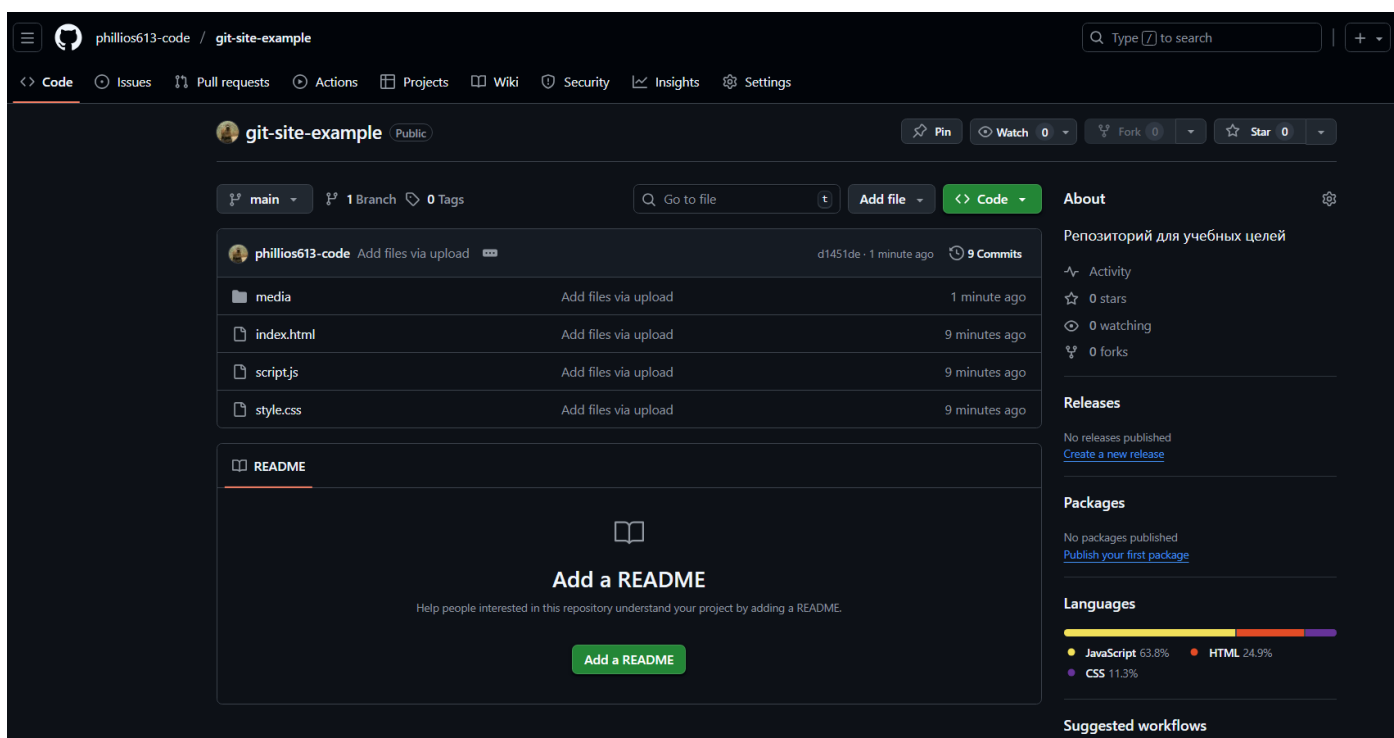
☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests](#).

Commit changes **Cancel**

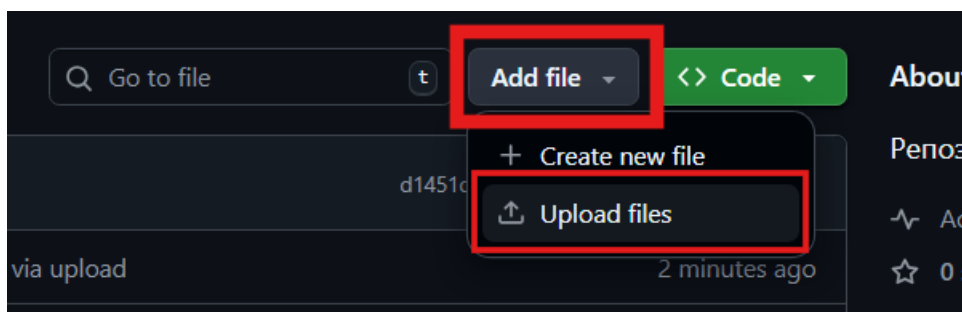
Целые папки можно добавить путем перетаскивания (автоматически будет создана отдельная директория). Когда файлы прогрузятся, можете внести описание в вводимые изменения в репозитории. Далее, делаете подтверждение изменений «Commit changes»



Как будет выглядеть репозиторий после добавления файлов:

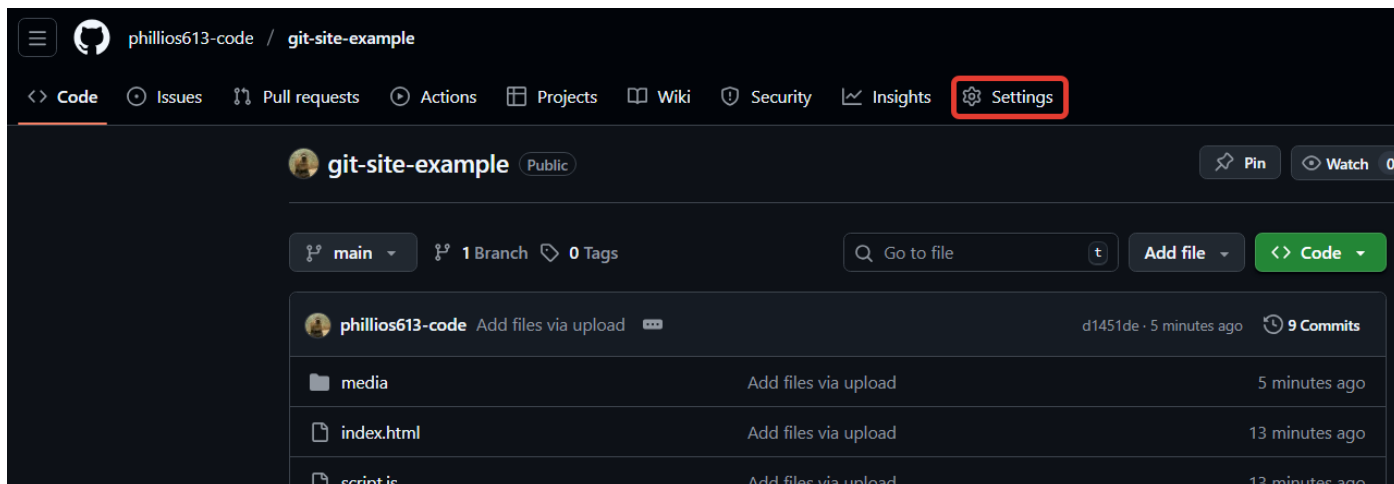


Если нужно добавить что-то дополнительно нажимаете Add File → Upload files

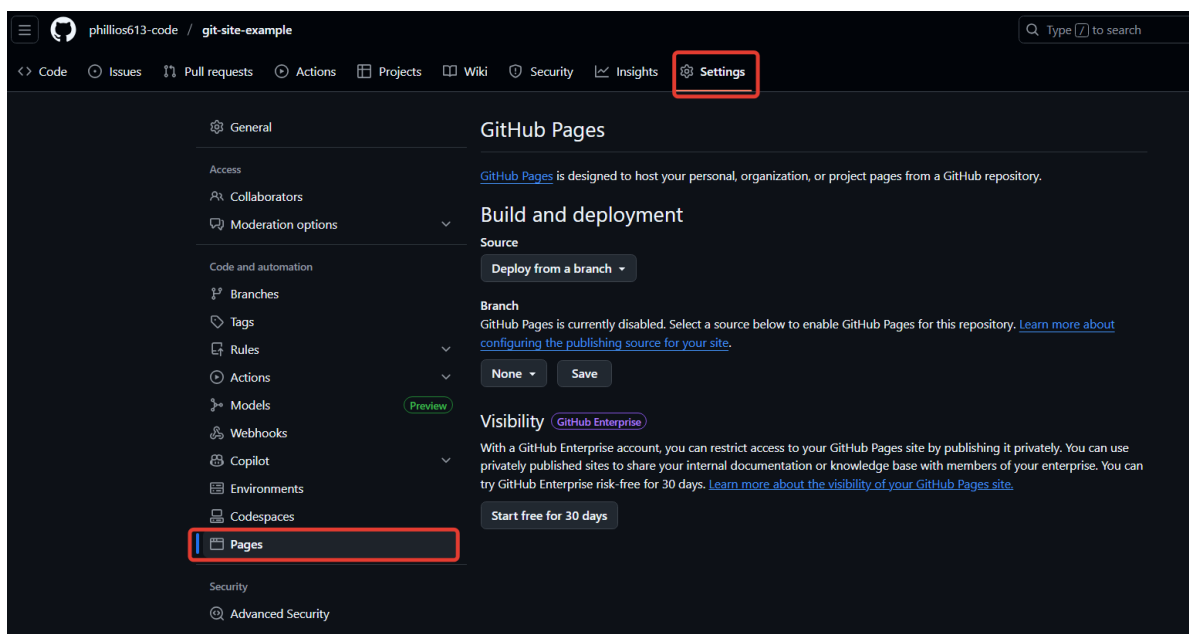


4. Для публикации сайта в настройках репозитория проделайте следующее.

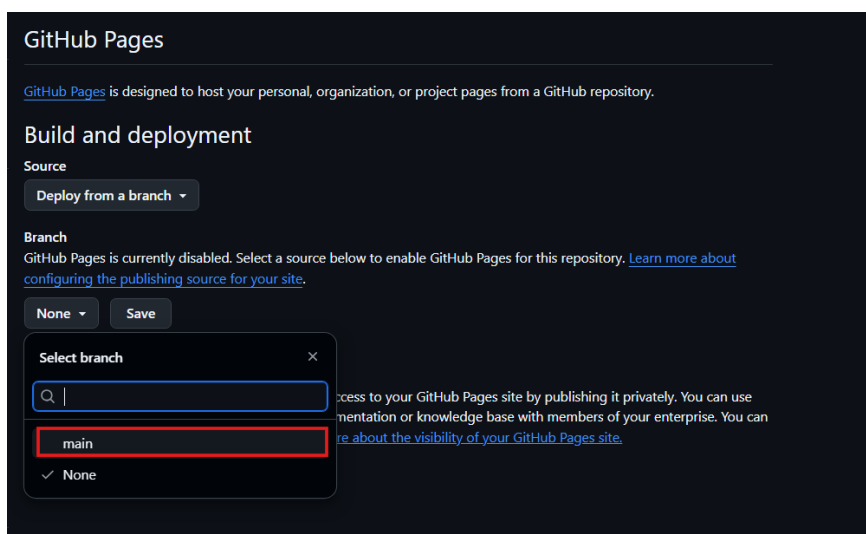
Перейдите в настройки репозитория “Settings”.



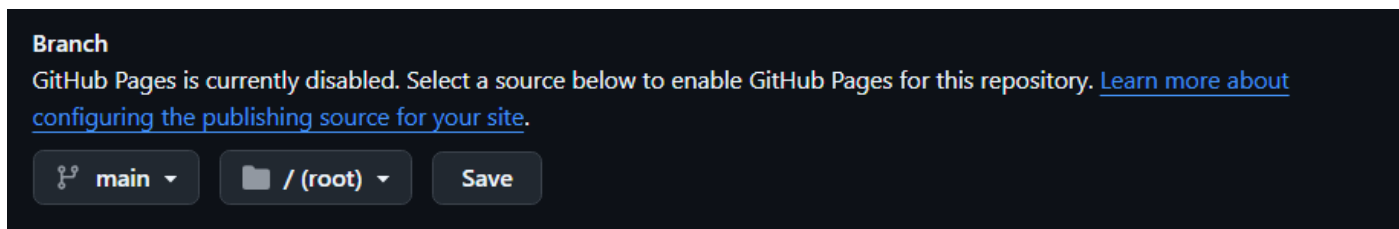
Найдите в списке раздел Pages.



Выберите разветвление - источник откуда GitHub Pages будет брать данные о публикуемом сайте.

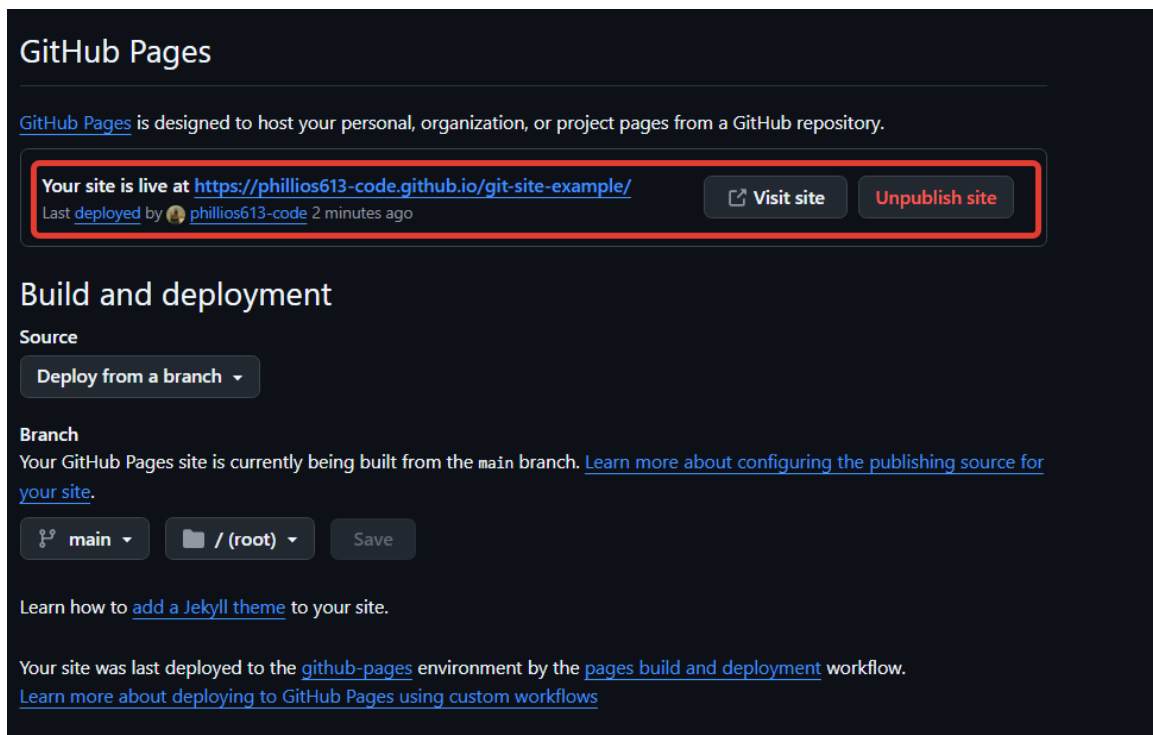


Если у вас html-файл сайта находится не в корневой папке /root, а внутри какой-то директории, выберите свою директорию, если же в корневой – то оставьте как есть.



Сохраните “Save”.

Через некоторое время сайт будет доступен по адресу (нужно подождать 2-3 минуты, обновить страницу).

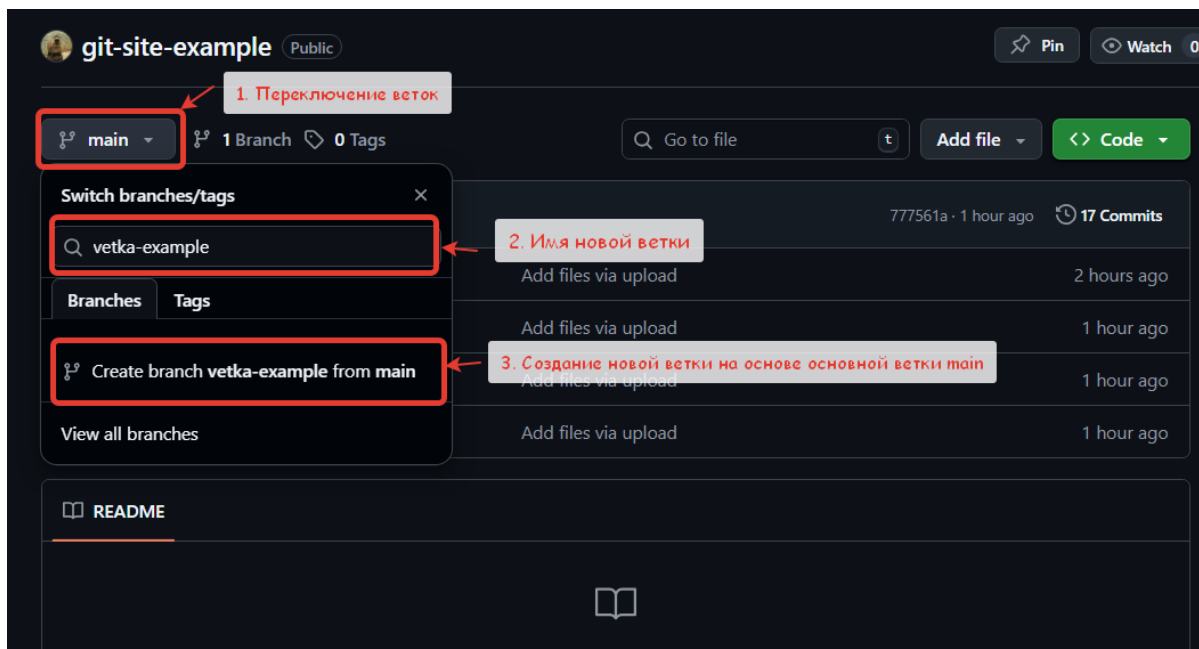


Перейдите по ссылке, проверьте сайт.

Задание 2. Редактирование сайта и работа в альтернативной ветке

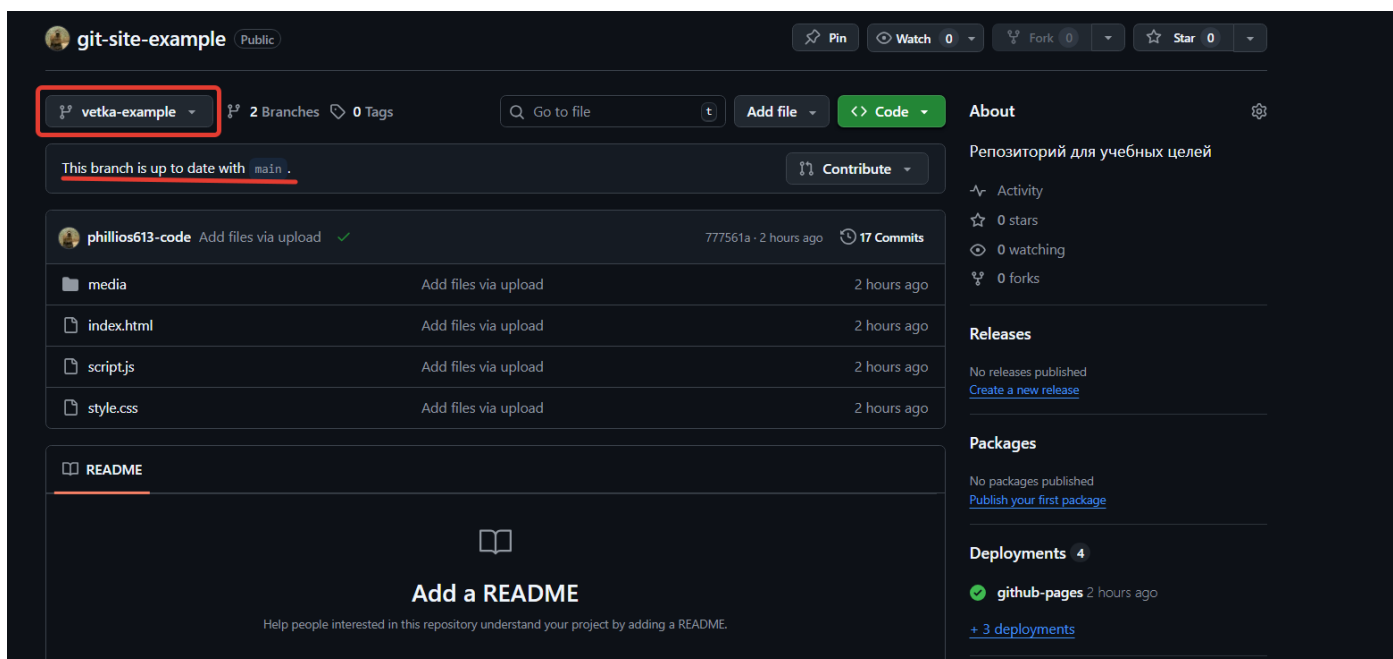
Создайте новый репозиторий. Туда поместите файлы из прикрепленной папки в СДО.

Создайте побочную ветку на основе главной (main). Во вкладке “< >Code”

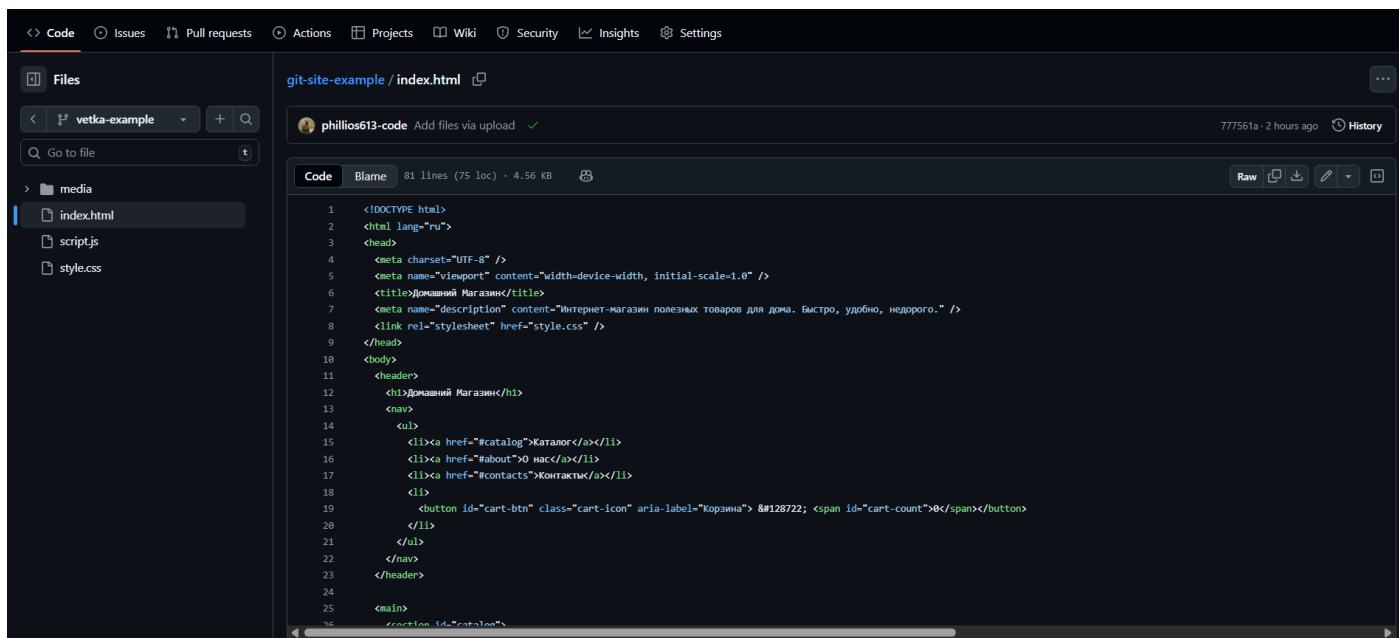


После совершенных действий появится новая ветка – копия от основной, где можно спокойно редактировать файлы, создавать, тестировать новую функциональность, не боясь что это затронет основной код.

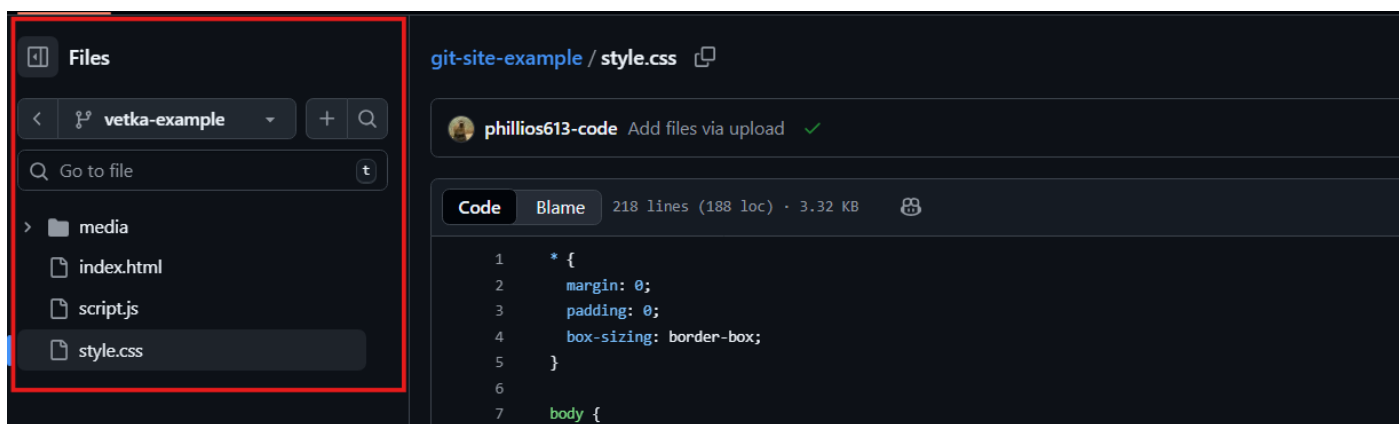
Убедитесь, что у вас работа переключена на побочную ветку.



В самом GitHub встроен свой IDE (среда разработки по типу VScode но чуть попримитивнее), если нажать на файл с кодом, например index.html, отобразится свой редактор кода GitHub:

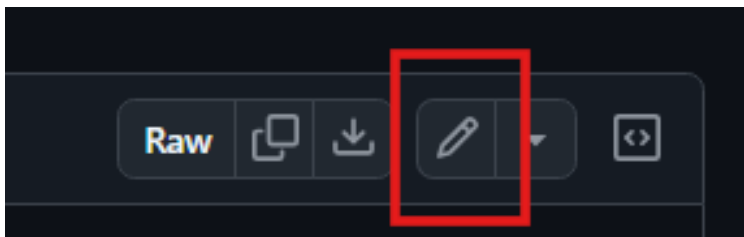


Просмотреть элементы проекта, переключаться между файлами можно через боковую прилегающую панель.

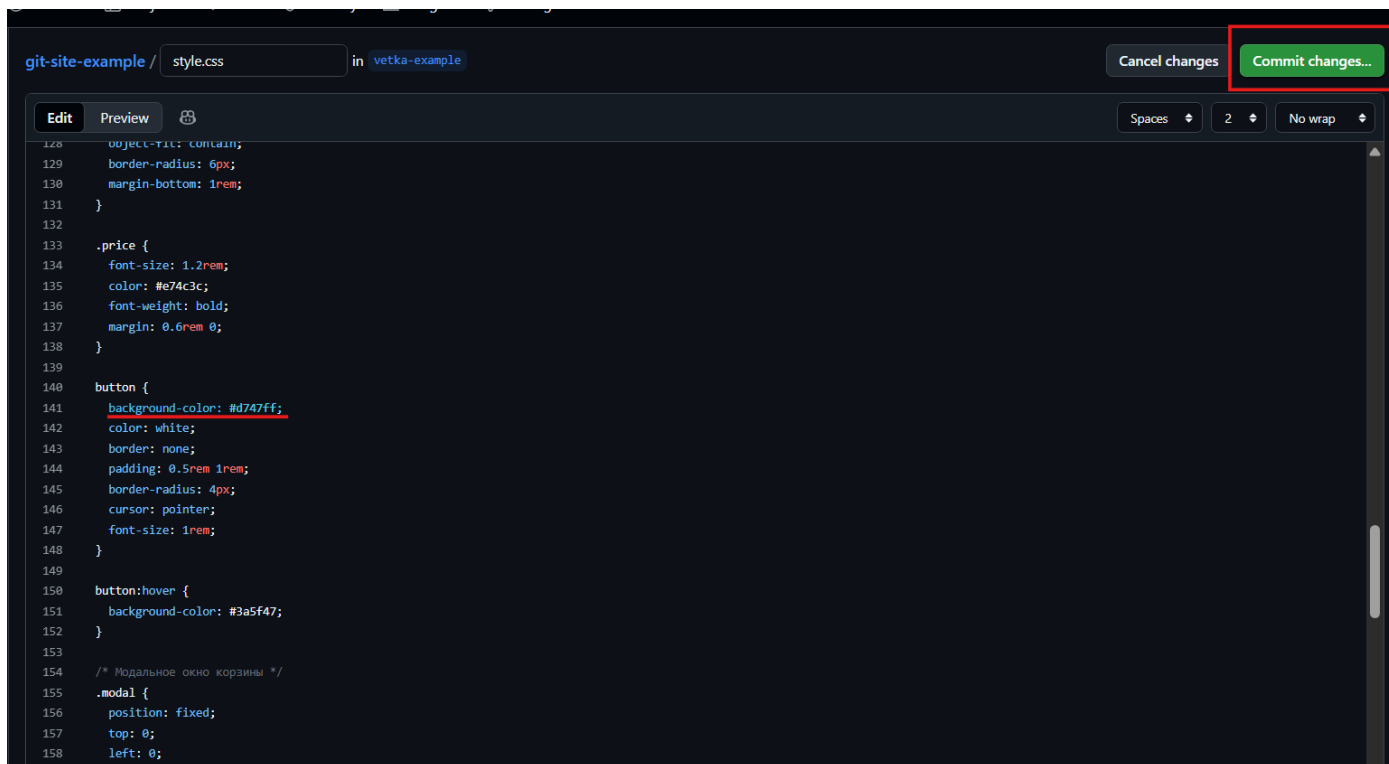


Переключитесь на файл `style.css`. Отредактируйте цвета кнопок и фона карточек товара на оттенки сиреневого.

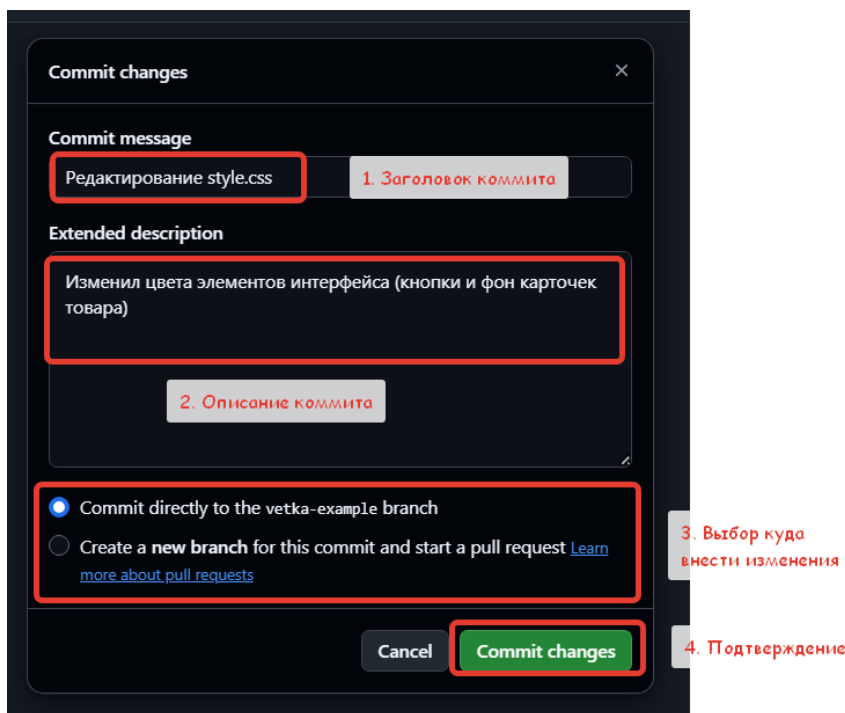
Для редактирования кода, нужно включить режим редактирования путем, нажатия на иконку с карандашом.



После внесения изменений нужно их обязательно подтвердить «Commit changes». **Сделайте коммит.**

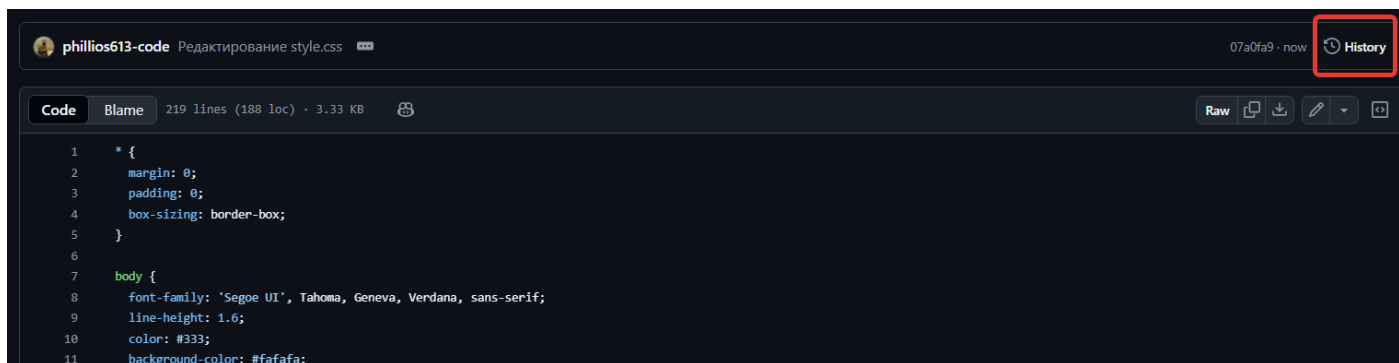


В окне описания совершаемого коммита можете внести данные (о чем коммит, как называется и куда коммитить). Будет полезно потом просматривать/проследивать в истории версий/изменений какое было изменение применено и зачем – как для вас, так и для других разработчиков.

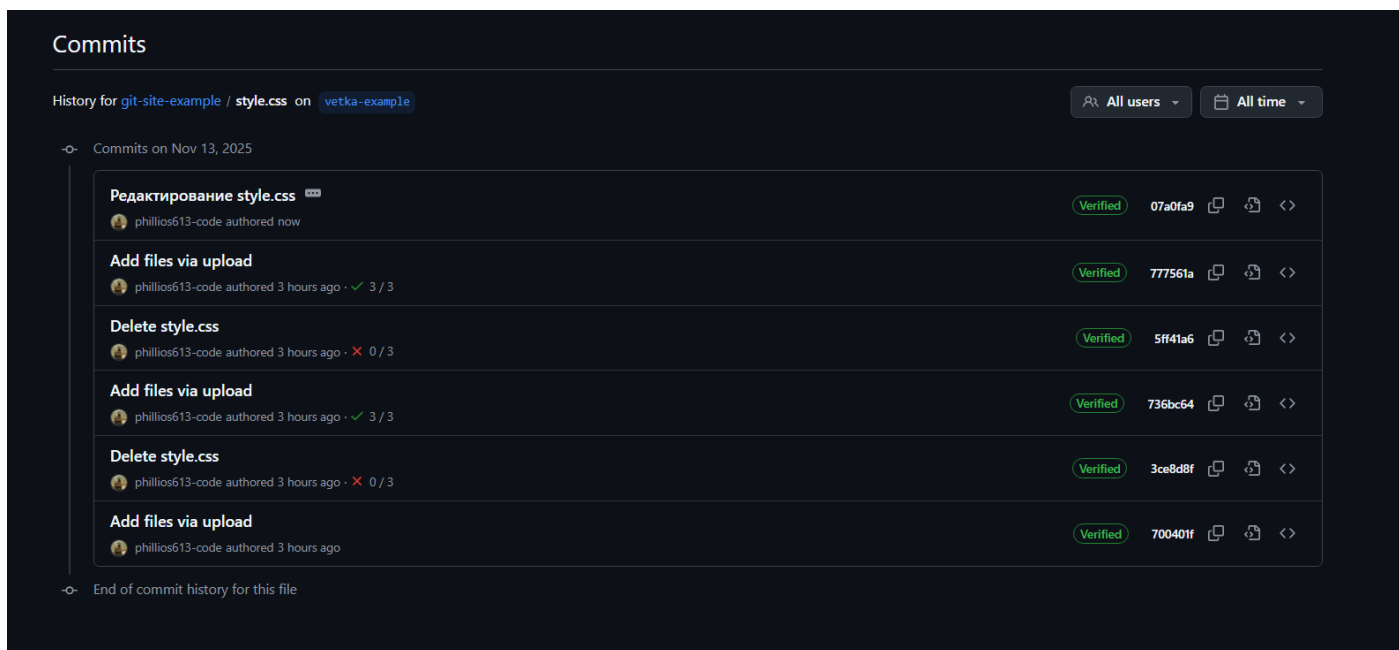


Просмотрите историю версий/коммитов.

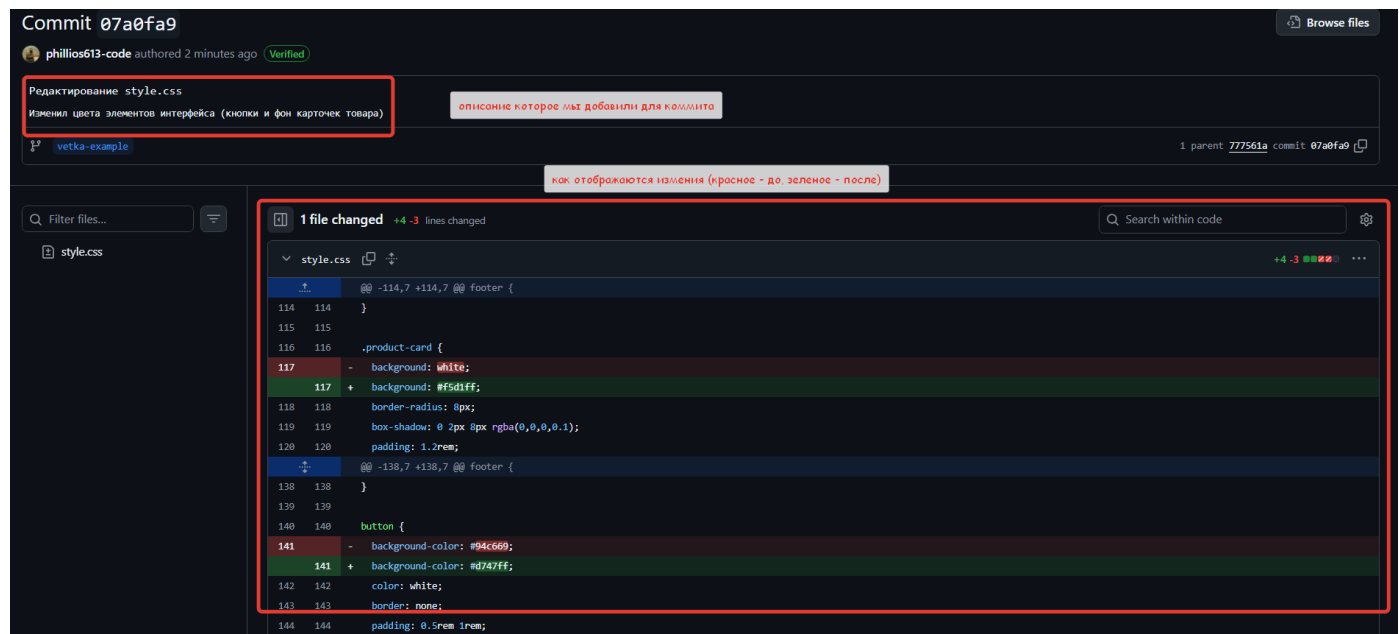
Историю версий можно посмотреть тут:



Там будет отображаться история всех совершенных действий:

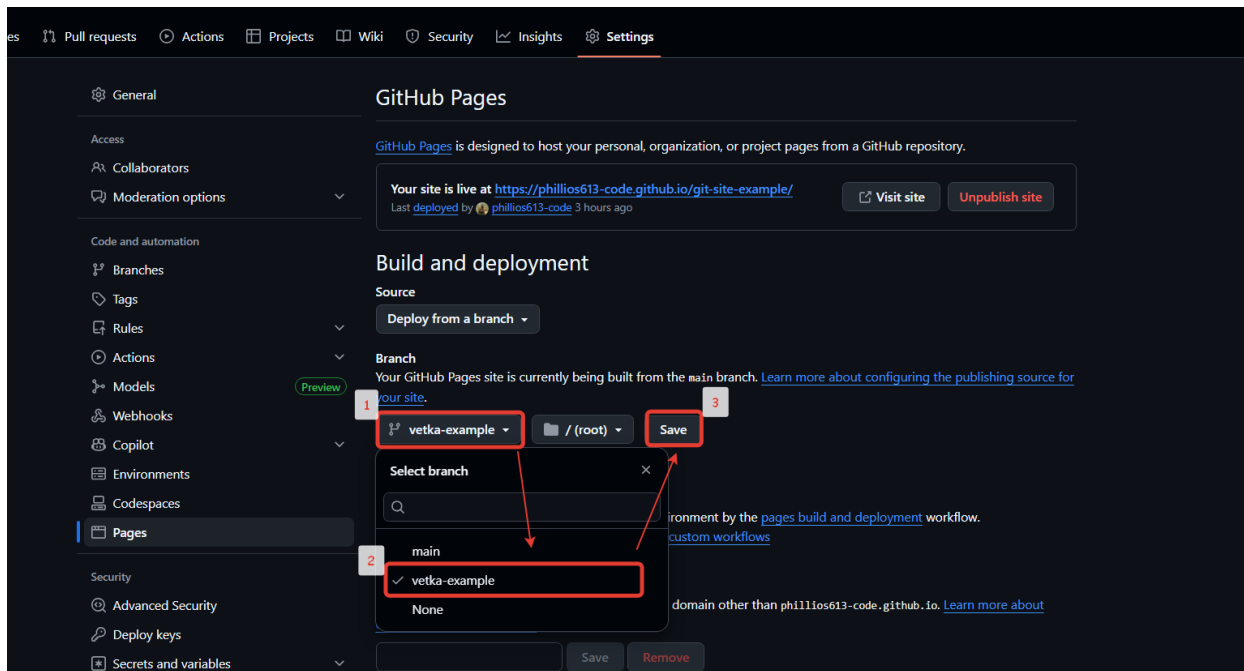


Например, зайдем в запись последнего совершенного нами коммита:



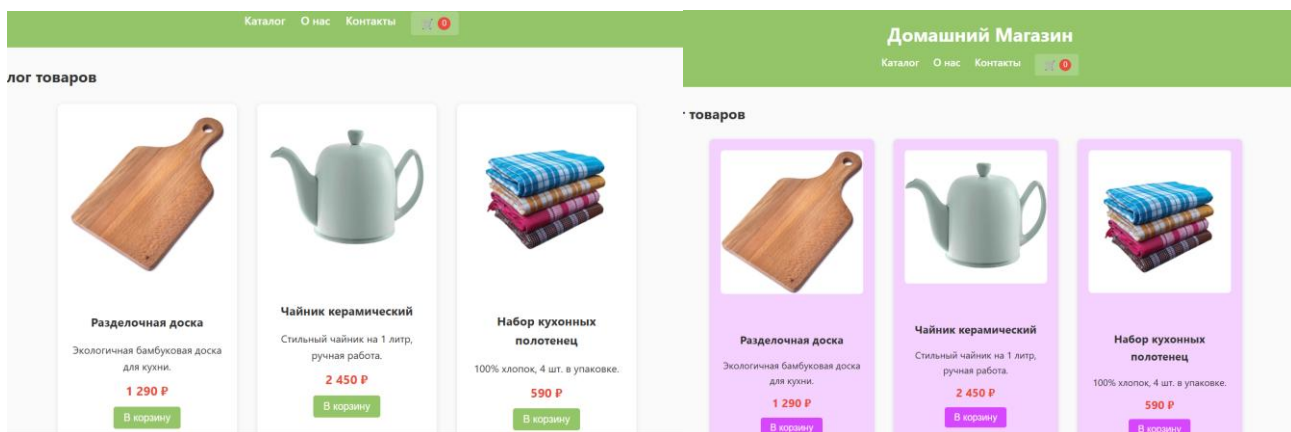
Просмотрите изменения на сайте.

Поскольку мы работаем на побочной ветке, нужно в настройках GitHub Pages, поменять ветку с которой берутся данные, вместо main укажите побочную, подтвердите Save



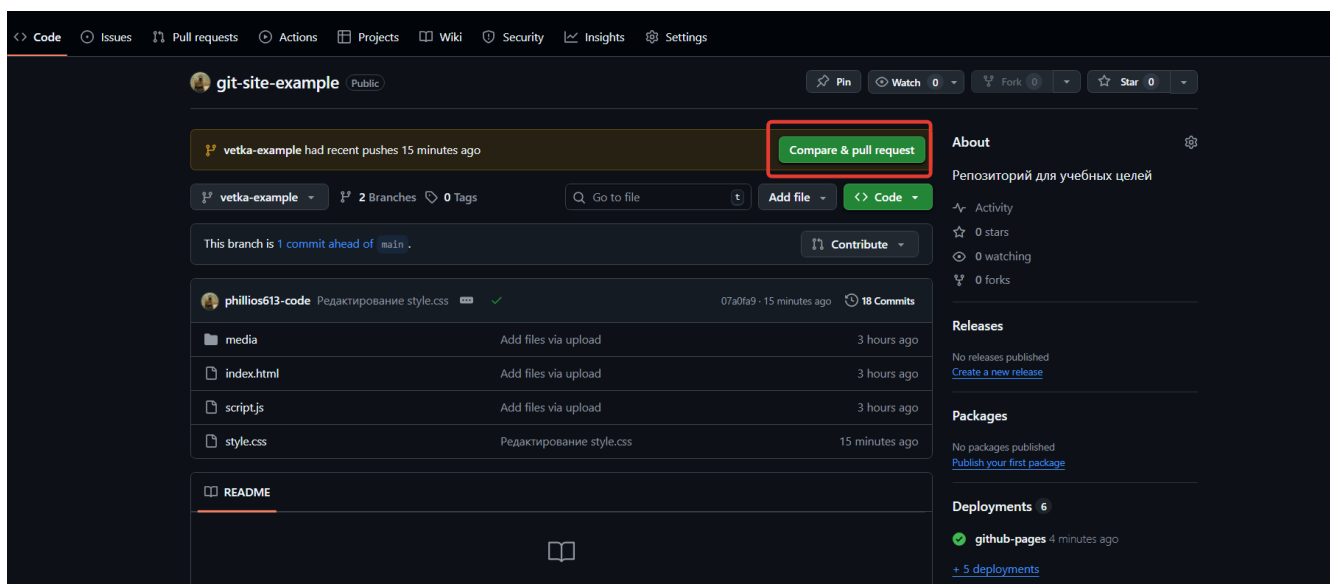
Далее, подождите некоторое время (2-3 минуты). Проверьте сайт по ссылке.

До/после:

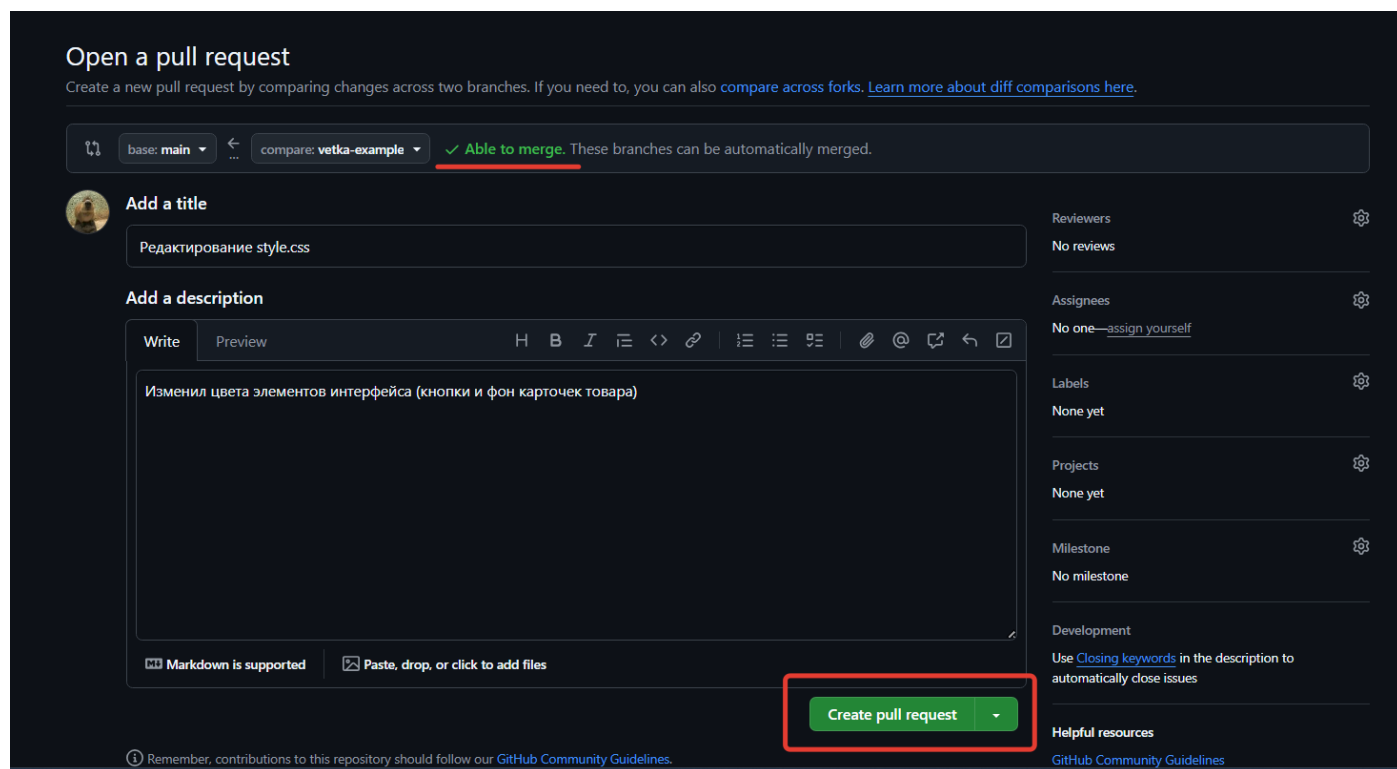


Как видите изменения вступили в силу.

Проведите слияние кода из побочной ветки в основную, нажав «Compare & pull request», чтобы код из побочной ветки применился к основной.



Перед слиянием можно проверить изменения на побочных ветках. Так же сам GitHub проверяет совместимость и возможность применения изменений от побочной ветки к основной. Если все устраивает и стоит плашка “Able to merge”, то вы можете подтвердить слияние.



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#). [Learn more about diff comparisons here](#).

base: main ← compare: vetka-example ✓ Able to merge. These branches can be automatically merged.

Add a title

Редактирование style.css

Add a description

Write Preview

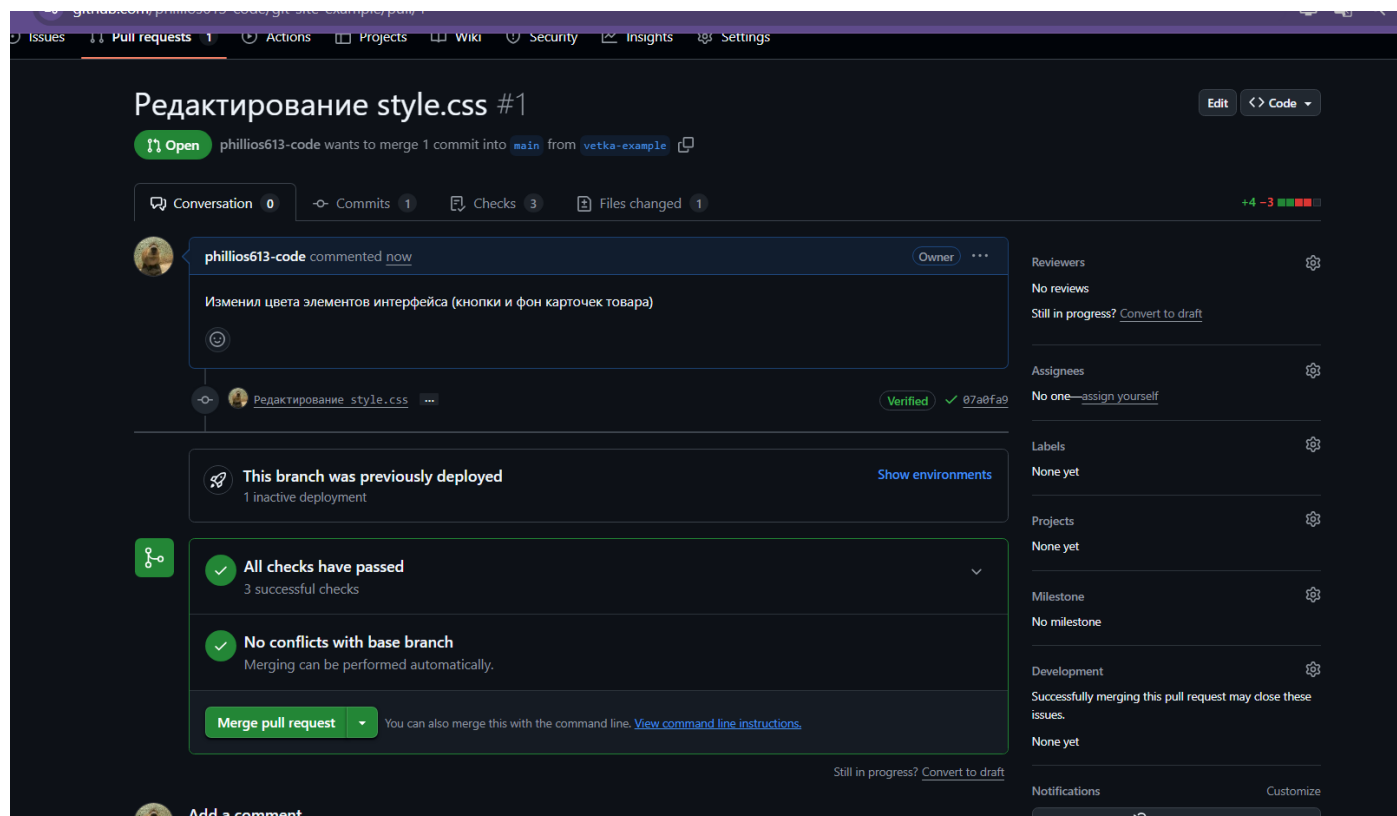
Изменил цвета элементов интерфейса (кнопки и фон карточек товара)

Markdown is supported Paste, drop, or click to add files

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Все подтвердить (Merge pull request):



Редактирование style.css #1

Open phillios613-code wants to merge 1 commit into main from vetka-example

Conversation 0 Commits 1 Checks 3 Files changed 1 +4 -3

phillios613-code commented now

Изменил цвета элементов интерфейса (кнопки и фон карточек товара)

Редактирование style.css Verified ✓ 07a0fa9

This branch was previously deployed 1 inactive deployment Show environments

All checks have passed 3 successful checks

No conflicts with base branch Merging can be performed automatically.

Merge pull request You can also merge this with the command line. [View command line instructions](#)

Still in progress? Convert to draft

Add a comment

Подтвержденные изменения из побочной ветки применяются к основной:

Редактирование style.css #1

Edit <> Code

Merged phillios613-code merged 1 commit into `main` from `vetka-example` now

Conversation 0 Commits 1 Checks 3 Files changed 1 +4 -3

phillios613-code commented [now](#)

Изменил цвета элементов интерфейса (кнопки и фон карточек товара)

Owner

...

...

Редактирование style.css

Verified ✓ 07a0fa9

...

phillios613-code merged commit `bfa0448b` into `main` [now](#)

3 checks passed

View details

Revert

This branch was previously deployed

1 inactive deployment

[Show environments](#)

Pull request successfully merged and closed

You're all set — the `vetka-example` branch can be safely deleted.

[Delete branch](#)

Add a comment

Write

Preview

H

B

I

≡

<>

Convert to draft

Assignees

No one—[assign yourself](#)

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet