

Лабораторна робота № 8.1

РОБОТА З САЙТОМ GITHUB.COM

Мета: Отримати уявлення та основні навички роботи з сайтом GITHUB.COM

Необхідно мати наступні знання, вміння та навички:

1. Знати теоретичний матеріал про сайт GITHUB.COM
2. Вміти знаходити необхідну інформацію про принципи роботи сайту GITHUB.COM в мережі Інтернет.
3. Володіти навичками використання сайту GITHUB.COM в особистих цілях і колективної діяльності.

1.1. Сутність поняття git

Git — це розподілена система керування версіями файлів та спільної роботи. Проект створив Лінус Торвальдс для управління розробкою ядра Linux, сьогодні проект підтримується Джуніо Хамано.

Git є однією з найефективніших, надійних і високопродуктивних систем керування версіями, яка надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні і злитті гілок. Для забезпечення цілісності історії та стійкості до змін заднім числом використовуються криптографічні методи, можлива прив'язка цифрових підписів розробників до тегів і комітів.

Система спроектована як набір програм, спеціально розроблених з урахуванням їхнього використання у скриптах. Це дозволяє зручно створювати спеціалізовані системи управління версіями на базі Git. Наприклад, Cogito є саме таким прикладом фронтенда до репозиторіїв Git. А StGit використовує Git для управління колекцією патчів.

Система має ряд інтерфейсів: наприклад, gitk та git-gui.

Віддалений доступ до репозиторіїв Git забезпечується git-демоном, SSH або HTTP сервером. TCP-сервіс git-daemon входить у дистрибутив Git і є разом з SSH найпоширенішим та надійним методом доступу. Метод доступу HTTP, незважаючи на ряд обмежень, дуже популярний у контрольованих мережах, тому що дозволяє використання конфігурацій мережевих фільтрів, що існують.

Збереження файлів

Git, на відміну від Subversion і подібних до неї систем, не зберігає інформацію як список змін (патчів) для файлів. Замість цього, Git зберігає дані набором зліпків.

Кожного разу при фіксації поточної версії проекту Git зберігає зліпок того, як виглядають всі файли проекту. Але, якщо файл не змінювався, то дається посилання на раніше збережений файл. Git схожий на своєрідну файлову систему з інструментами, які працюють поверх неї.

Уся база даних Git зберігається в теці з назвою `.git`. Для кожного файлу, що відстежується, Git зберігає розмір, час створення і час останньої зміни. Ці дані зберігаються у файлі `index`.

У Git файли можуть знаходитися в одному із 3-х станів: **зафіксованому** (файл уже збережено в локальній базі даних), **зміненому** (файл було змінено, але зміни не зафіксовано) і **підготовленому** (файли було змінено і відмічено для фіксації).

Локальні операції

На відміну від Subversion, де без підключення до мережі Інтернет можна лише редагувати файли, але зберегти зміни у вашу базу даних неможливо (оскільки вона відключена від репозиторію), у Git більшість дій можна виконувати без використання Інтернет-підключення на локальній файловій системі. Будь-який коміт спочатку робиться локально, потім вивантажується у віддалений репозиторій. Уся історія змін локально зберігається і, за необхідності, вивантажується у віддалений репозиторій.

Цілісність даних

У своїй базі Git зберігає дані по хешах файлів. Як хешувальна функція використовується SHA-1. Перед кожним збереженням файлів Git обчислює SHA-1 хеш файлу, і отриманий хеш стає індексом файлу в Git. Використовуючи хеш, Git легко відслідковує зміни в файлах.

Галуження (гілки)

Галуження — це розмежування від основної лінії розробки. Git дозволяє створити декілька гілок і перемикатися між ними.

Гілка в Git просто являє собою вказівник на одну із фіксацій. Під час кожної нової фіксації гілка в Git рухається автоматично (тобто перемикається на фіксацію). Гілка є простим файлом, який містить 40 символів контрольної суми SHA-1 фіксації. Створення нової гілки дуже швидке, оскільки це дорівнює запису в файл 41 байта (40 символів + символ нового рядка).

Галуження корисне, бо дозволяє працювати декільком розробникам над своїм функціоналом, не заважаючи іншим і не псуючи основну гілку.

За замовчуванням Git створює гілку з назвою `master`.

Зливання та перебазування даних

Git підтримує два способи для інтеграції змін з гілки в гілку: ***merge*** (зливання) та ***rebase*** (перебазування). Основна різниця полягає в тому, що `rebase` запам'ятовує фіксації у вигляді патчів, перемотує гілку і застосовує патчі у вигляді фіксацій, `merge` - зливає дві гілки в одну.

Завдання на оцінку "задовільно":

1.2. Реєстрація на сайті github.com

Для того щоб почати працювати з системою Git, потрібно зареєструватися на сайті github.com та створити репозиторій.

Репозиторій - це контейнер для зберігання ваших файлів.

Розглянемо процес реєстрації на сайті github.com:

У браузері перейдіть до сайту ***github.com***. Для цього у адресному рядку введіть ***https://github.com/***. Відкриється сторінка з формою реєстрації.

Заповніть форму (рис. 1) та натисніть на кнопку *Sign up for GitHub*.

Рис.1. Форма реєстрації на сайті *github.com*

1.3. Створення репозиторію

Після успішної реєстрації та входу на сайт *github.com* під своїм обліковим записом, натисніть на кнопку «+» та з меню, що випадає, оберіть *New repository* (рис. 2).

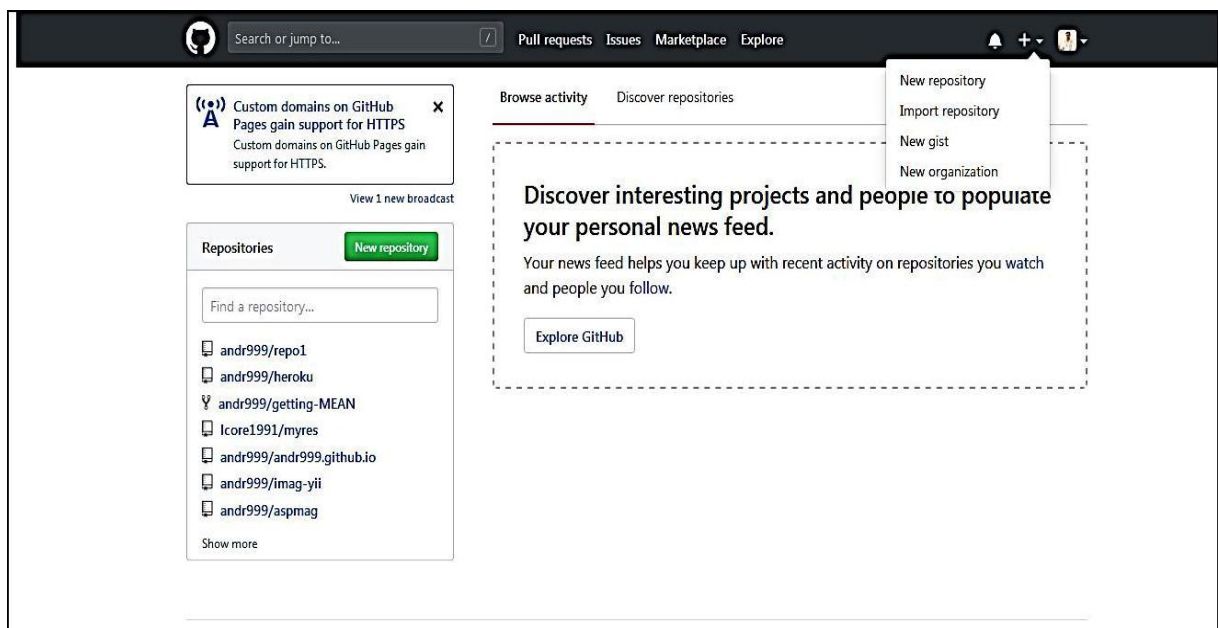


Рис. 2. Створення нового репозиторію

На сторінці створення репозиторію треба задати його ім'я, задати опис та ініціалізувати репозиторій README файлом. Потім натиснути на кнопку *Create repository*.

Після створення репозиторію ви потрапите на сторінку репозиторію. Поки що тут є один файл `Redme.md`, котрий було створено разом з репозиторієм (рис. 3).

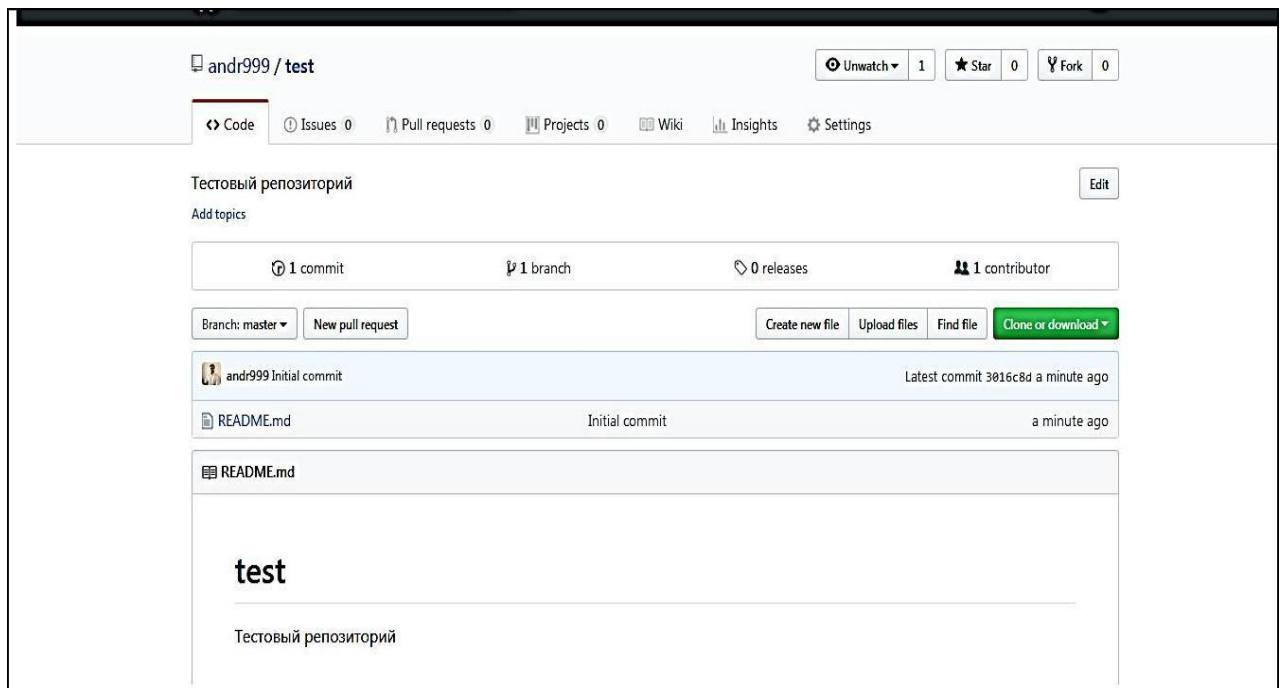


Рис. 3. Сторінка репозиторію

РОБОТА З РЕПОЗИТОРІЄМ

2.1. Створення нового файлу

Для додавання нового файлу у репозиторій натисніть кнопку *Create new file* (рис. 4).

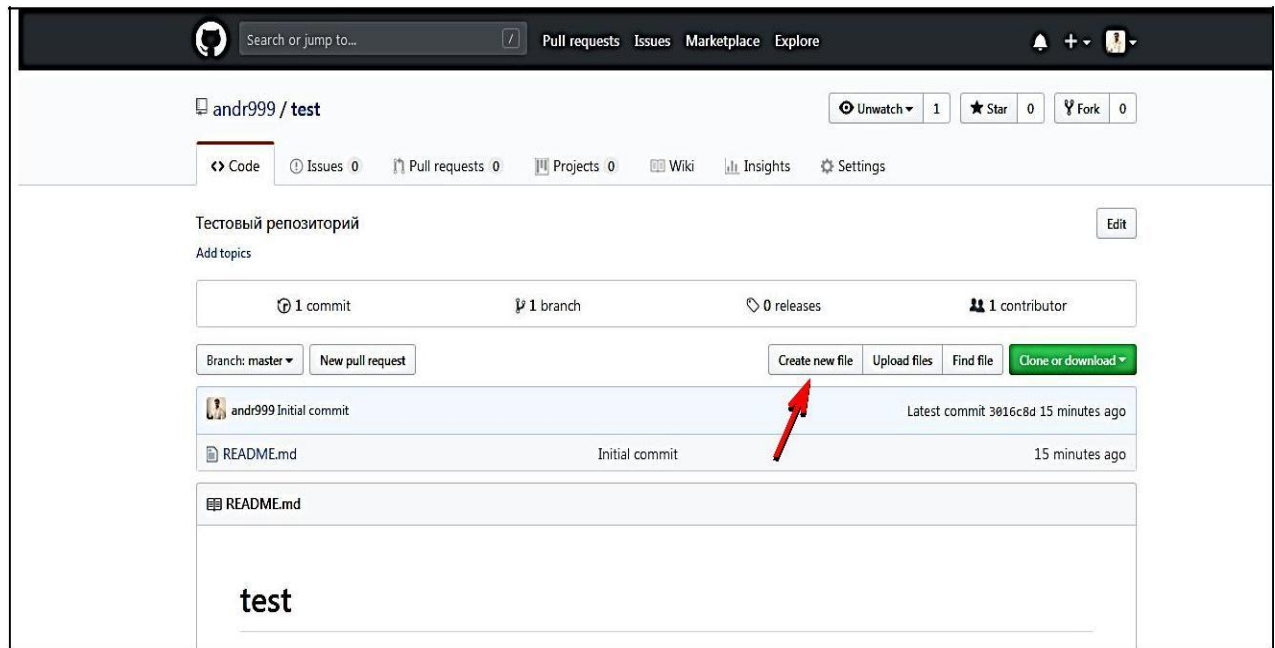


Рис. 4. Створення нового файла у репозиторії

На сторінці створення файла введіть назву файла та відредагуйте його зміст (рис. 5).

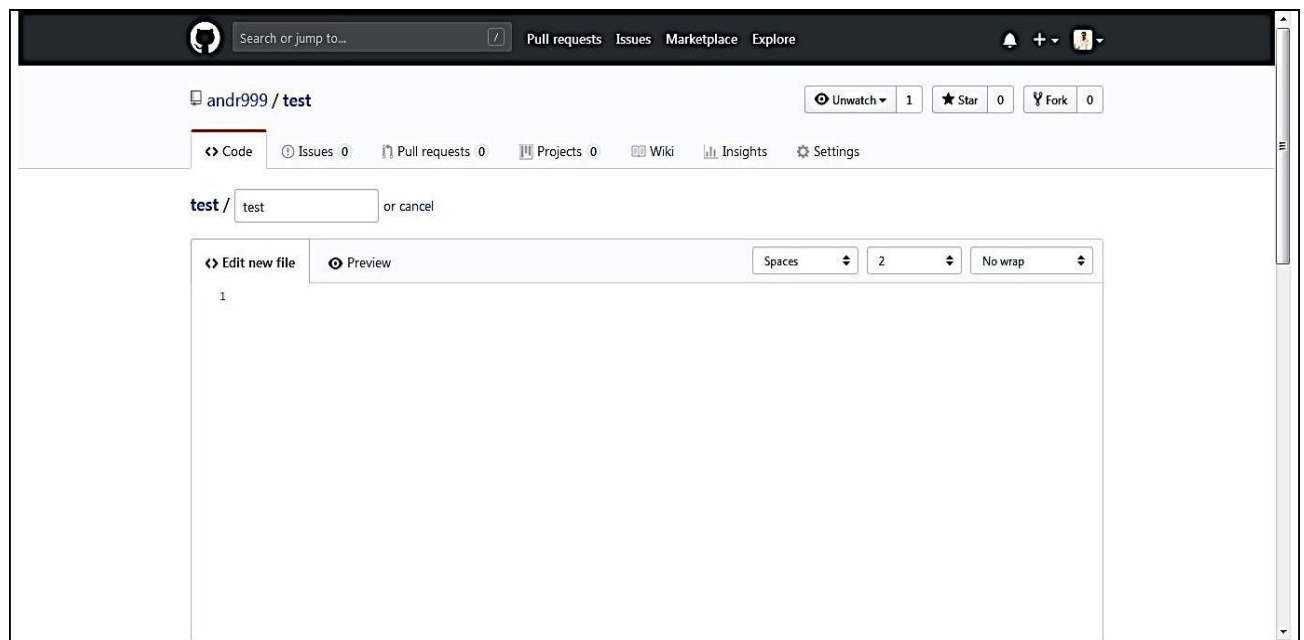


Рис. 5. Редагування файлу

Після чого натисніть на кнопку *Commit new file*, при цьому новий файл буде створено (рис. 6).

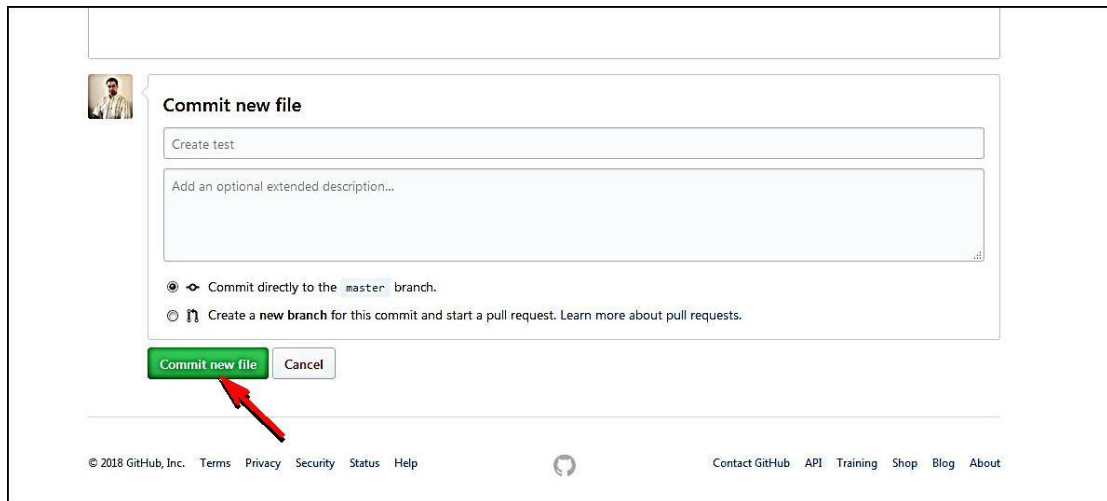


Рис. 6. Підтвердження змін у файлі

2.2. Завантаження файлів до репозиторію

Для завантаження файлів до репозиторію потрібно натиснути кнопку *Upload files* (рис. 7).

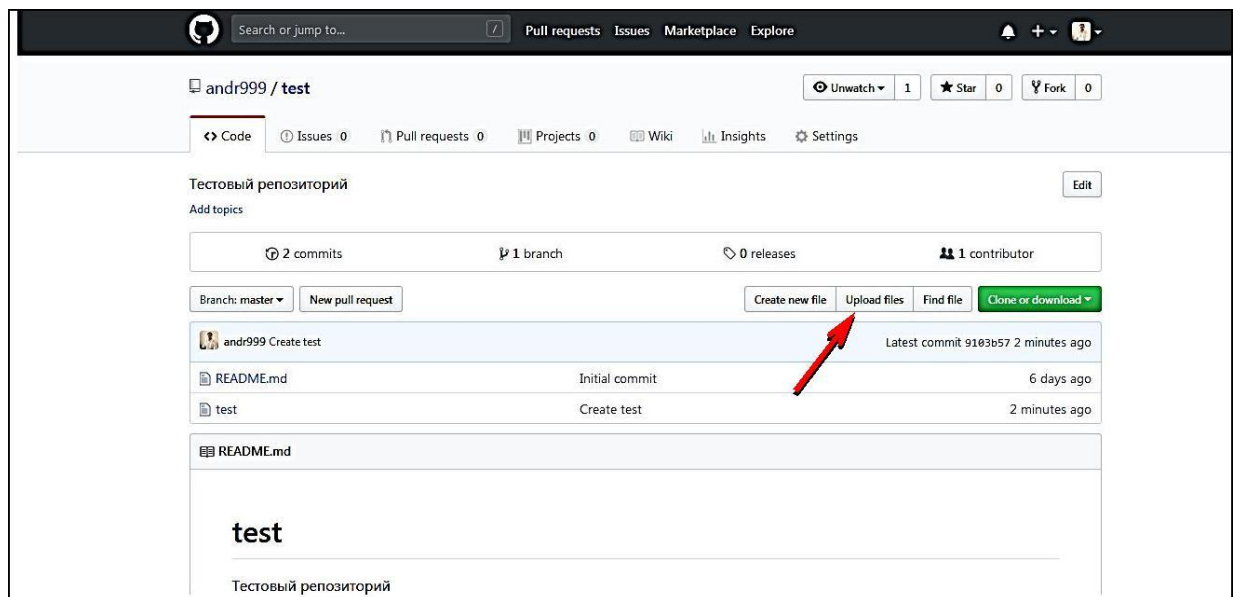


Рис. 7. Завантаження файлів до репозиторію

На наступній сторінці натисніть на посилання *Choose your files* (рис. 8).

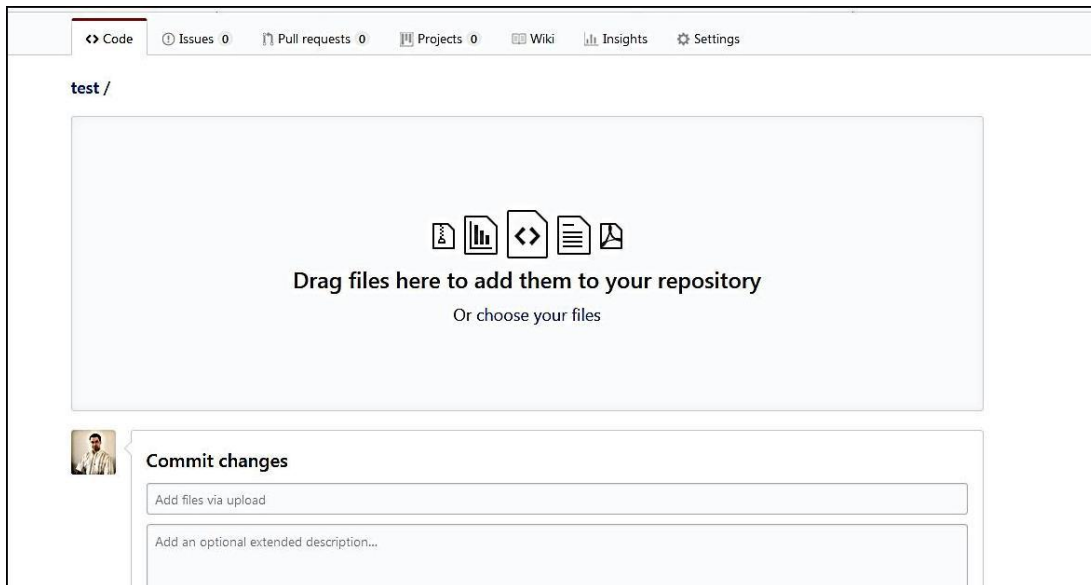


Рис. 8. Обрання файлів для завантаження

У вікні, що відкриється, оберіть файли для завантаження на Github і натисніть на кнопку *Відкрити*, при цьому файли завантажаться до репозиторію.

2.3. Створення нової гілки

Гілки необхідні для створення різних версій проекту. Таким чином, за допомогою гілки Ви, нібито заморожуєте проект на якійсь стадії, і можете над ним працювати окремо, не втручаючись в основну гілку, котра називається master-гілка. Таких гілок можна створювати скільки завгодно, обмежень немає.

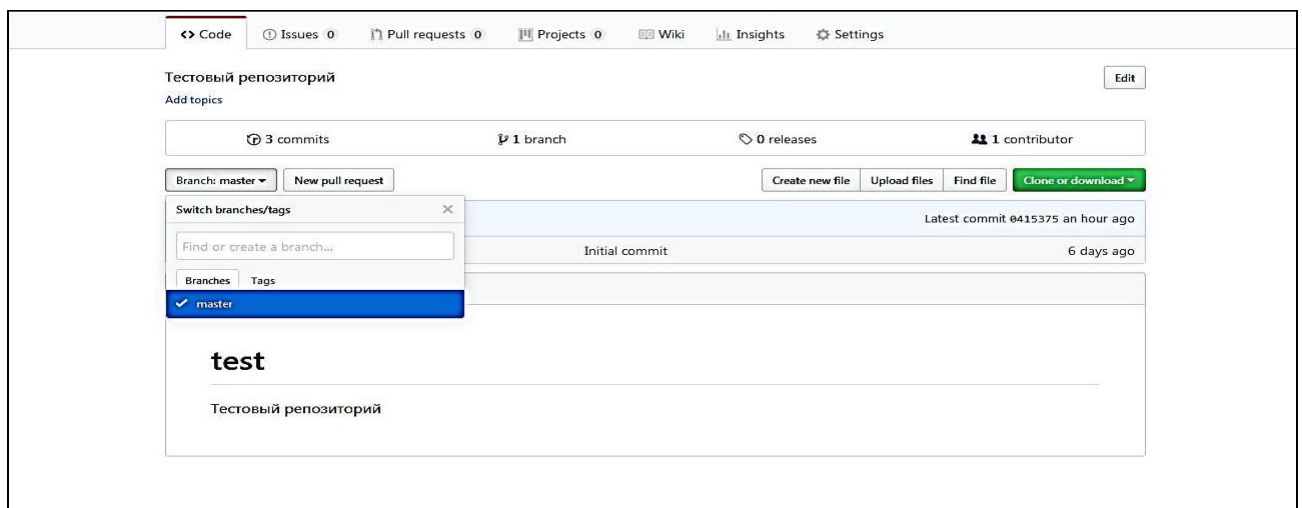


Рис. 9. Створення нової гілки

Для створення гілки потрібно натиснути на кнопку *Branch master* та у списку, що випадає, ввести ім'я нової гілки (рис. 9).

За такої умови ви отримаєте копію головної гілки і будете працювати вже у цій гілці. Отже, будь-які зміни, що ви зробите, не вплинуть на основну гілку доти, поки ви не розпочнете злиття гілок з основною гілкою.

2.4. Злиття гілок

Для того щоб зміни, зроблені у будь-якій гілці, синхронізувались з основною, потрібно синхронізувати або злити ці гілки. Для цього натисніть на кнопку *New pull request* (рис. 10).

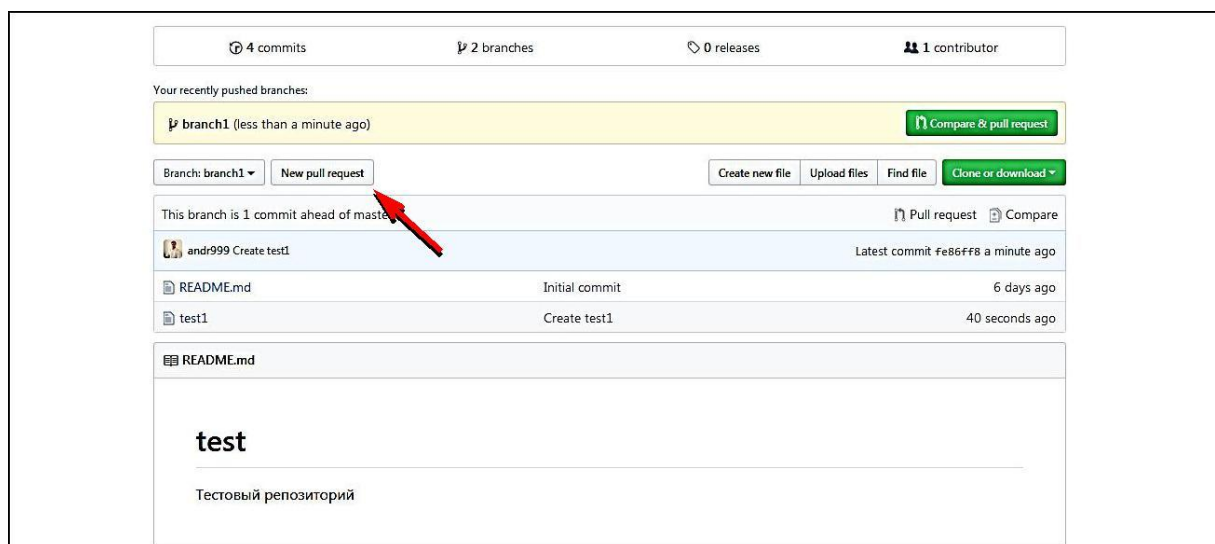


Рис. 10. Створення запиту на злиття гілок

На сторінці запиту на злиття буде спочатку перевірено можливість злиття гілок. Після перевірки та можливості злиття натисніть на кнопку *Create pull request* (рис. 11).

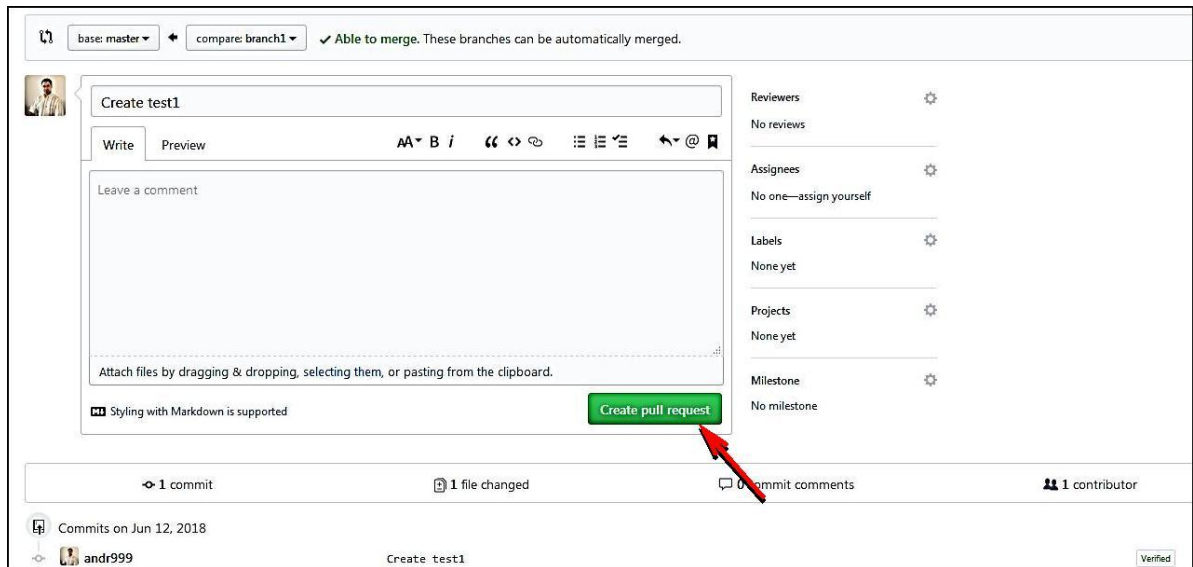


Рис. 11. Злиття гілок

На сторінці підтвердження злиття гілок буде також задіяно перевірку можливості злиття гілок і, якщо перевірка успішна, натисніть на кнопку *Merge pull request* (рис. 12).

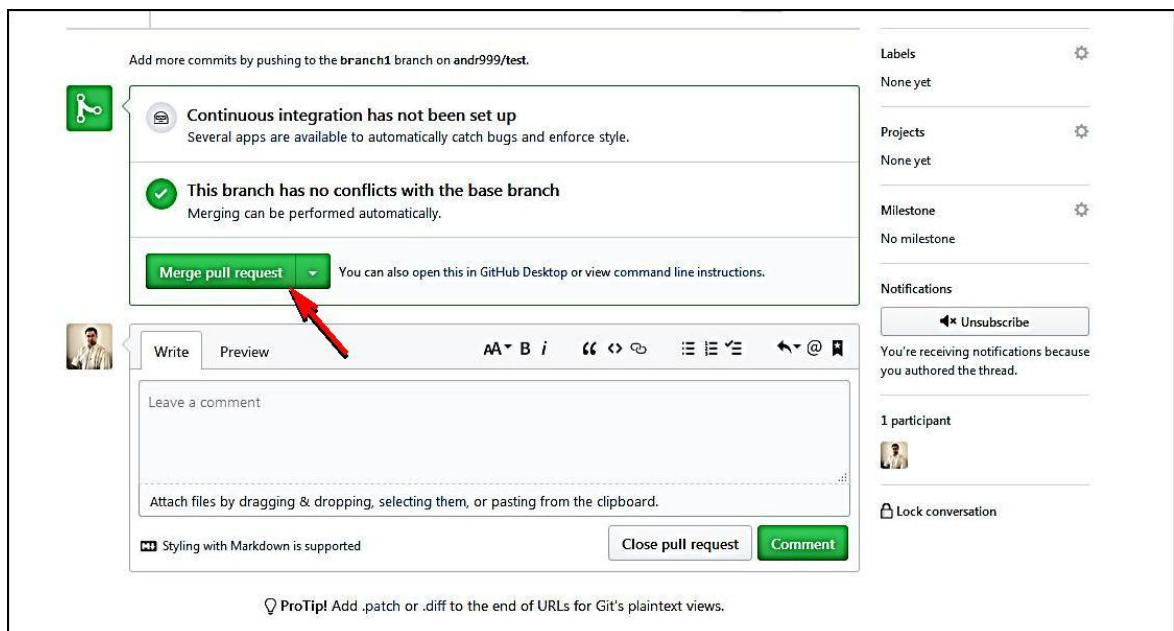


Рис. 12. Підтвердження злиття гілок

На наступній сторінці натисніть на кнопку *Confirm merge*. Після цього розпочнеться механізм злиття гілок. Після закінчення злиття необхідно видалити або залишити гілку, якщо вона ще потрібна (рис. 13).

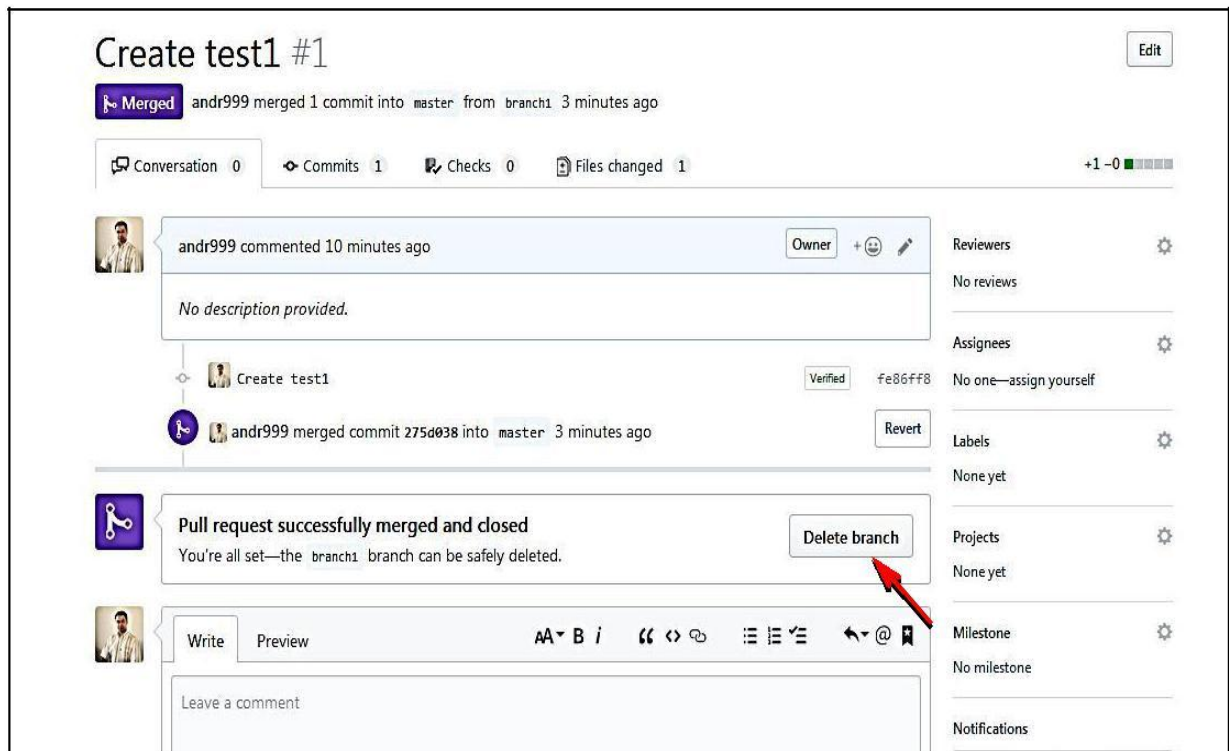


Рис. 13. Видалення гілки

2.5. Завантаження репозиторію

Для завантаження репозиторію на комп'ютер необхідно натиснути на кнопку *Clone or Download*, потім *Download ZIP* (рис. 14).

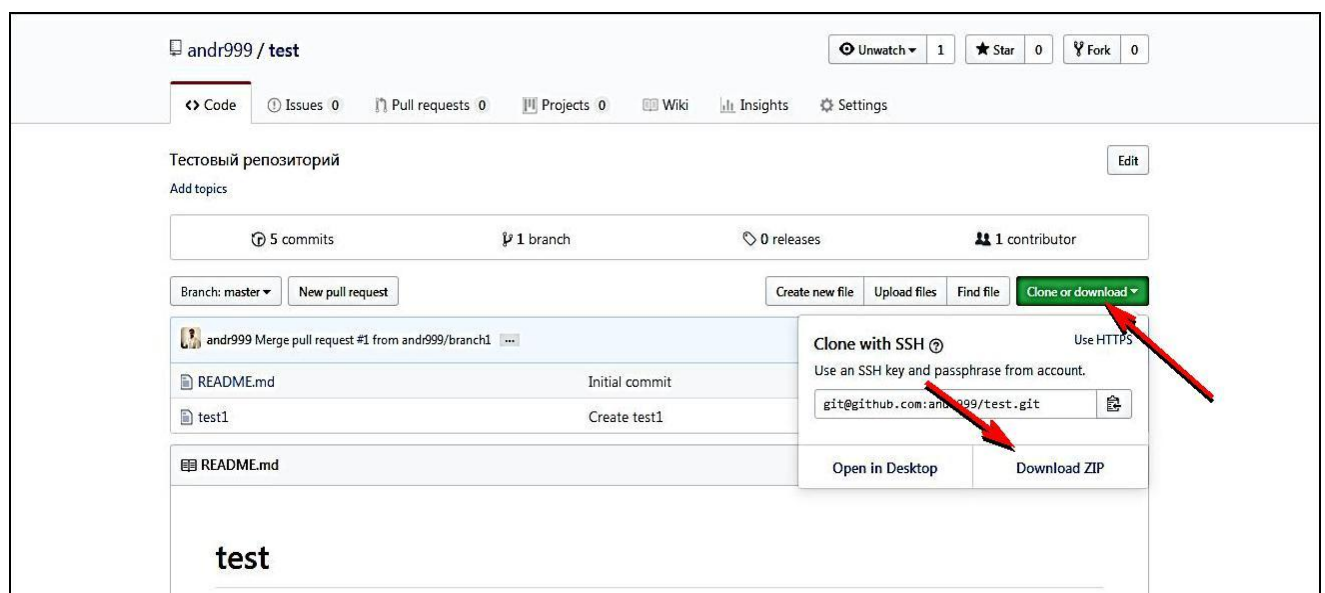


Рис. 14. Завантаження репозиторію

Водночас завантажиться архів у форматі Zip, який необхідно буде розархівувати.

Завдання на оцінку "добре":

У будь-якому текстовому редакторі створити звіт з відповідями на питання, додати цей файл до свого репозиторію:

- 1. Як створити новий репозиторій?*
- 2. Для чого призначен репозиторій?*
- 3. Як створити нову гілку у репозиторії?*
- 4. Як злити гілки з основною гілкою master?*
- 5. Як завантажити репозиторій до себе на комп'ютер?*

Завдання на оцінку "відмінно":

- 1. Зареєструйтеся на сайті github.com.*
- 2. Створіть репозиторій. Дайте назву репозиторію (наприклад My repository).*
- 3. У новому репозиторії створіть новий файл.*
- 4. Створіть нову гілку.*
- 5. У новій гілці створіть файл.*
- 6. Синхронізуйте нову гілку з основною гілкою master.*
- 7. Завантажте репозиторій на комп'ютер.*
- 8. Покажіть виконанні завдання викладачеві.*

КОНТРОЛЬНІ ПИТАННЯ:

- 1. Як створити новий репозиторій?*
- 2. Для чого призначен репозиторій?*
- 3. Як створити нову гілку у репозиторії?*
- 4. Як злити гілки з основною гілкою master?*
- 5. Як завантажити репозиторій до себе на комп'ютер?*

РЕКОМЕНДОВАНА ЛІТЕРАТУРА:

- 1 Білас О. Якість програмного забезпечення та тестування: навч. посіб. / О. Білас. – Львів : вид-во НУ «Львівська політехніка», 2011.– 216 с.
- 2 Лавріщева К.М. Програмна інженерія. –Підручник.–К.: Академперіодика, 2008.–415с.
- 3 Левус Є. В., Мельник Н. Б. Вступ до інженерії програмного забезпечення: Навчальний посібник. Львів: Видавництво Львівської політехніки, 2018. 248 с.
- 4 Основы инженерии качества программных систем / Ф.И.Андон, Г.И.Коваль, Т.М. Коротун, Е.М.Лаврищева, В.Ю. Суслов - К.: Академперіодика-2007.-678 с.
- 5 Соммервилл И. Инженерия программного обеспечения. 6-е изд. / Соммервилл И. – М.: "Вильямс", 2006. – 624 с.

Лабораторна робота №8.2

СТВОРЕННЯ БЕЗКОШТОВНОГО САЙТУ НА GITHUB PAGES

Мета: Формувати практичні вміння та навички зі створення безкоштовного сайту за допомогою використання інтернет-ресурсу.

Необхідно мати наступні знання, вміння та навички:

1. Знати теоретичний матеріал про призначення, основні можливості Інтернет-ресурсів з створення безкоштовних сайтів в Інтернет
2. Вміти знаходити необхідну навчальну інформацію в мережі Інтернет.
3. Володіти навичками збереження та обробки створеного сайту.

Сайт - це ваш спосіб комунікації зі світом. Існує кілька способів по створенню простих і безкоштовних сайтів для початківців. В основному такі сайти робляться на базі GitHub і WordPress.

WordPress - це відмінний варіант для новачків, які не розбираються в сайтобудування. На жаль, тут є одне «але». Якщо створювати безкоштовний сайт на WordPress, то всі побачать, що ви користуєтеся цією платформою, оскільки адреса сайту буде закінчуватися на `wordpress.com`, а внизу кожної сторінки з'явиться логотип WordPress.

Знайомлячись з миром технологій, ви швидко помітите, що вміння працювати з GitHub дасть вам явна перевага. Ну, а якщо ви вже міцно влаштувалися в ІТ-всесвіті, то, швидше за все, встигли відзначитися в якомусь GitHub-репозиторії.

При розміщенні проектів на GitHub люди бачать не тільки ваш код, але і те, що ви робите і як. GitHub - це справжній круговорот ідей. Так чому б не створити свій сайт на GitHub і не розмістити його в самому репозиторії? До того ж так ви додасте кілька коммітов до свого профілю!

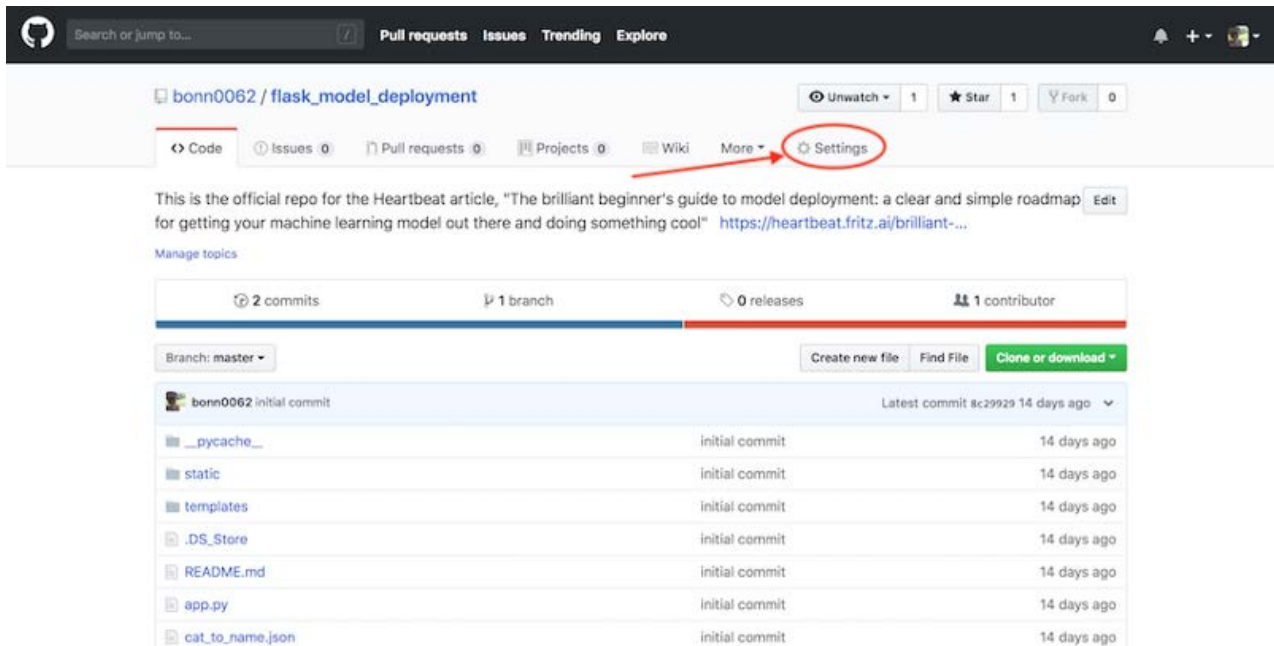
Існує цілих два можливих сценарії для створення сайтів. Ви або створюєте сайт з нуля, або вже маєте якийсь прототип (або готовий сайт) та безкоштовно розміщуєте його на GitHub.

ЗАВДАННЯ

На оцінку «задовільно»:

Якщо у вас є файл `index.html`, то GitHub відразу зрозуміє, що від нього хочуть.

Тепер саме час переключитися на GitHub Pages. Зайдіть в свій GitHub-репозиторій і натисніть Settings.



Прокрутіть до GitHub pages. Тут ви побачите наступне:

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source
GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

None

Theme Chooser
Select a theme to publish your site with a Jekyll theme using the master branch. [Learn more.](#)

Choose a theme

Danger Zone

Make this repository private
Hide this repository from the public. [Make private](#)

Transfer ownership
Transfer this repository to another user or to an organization where you have the ability to create repositories. [Transfer](#)

Тепер в випадаючому меню Source виберіть master branch або master branch / docs folder. Різниця ось у чому: якщо ви будете розміщувати проект з папки docs, то ця папка (docs) обов'язково має бути присутня в гілці master, з якої запускається сайт!

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source
GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

None ▾

Select source

- master branch**
Use the master branch for GitHub Pages.
- master branch /docs folder
Use only the /docs folder for GitHub Pages.

☒ **None**
Disable GitHub Pages.

Make this repository private
Hide this repository from the public. [Make private](#)

Transfer ownership
Transfer this repository to another user or to an organization where you have the ability to create repositories. [Transfer](#)

Master branch - репозиторій буде розміщений «як є». Далі слід повідомлення, що сайт готовий до розміщення.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at https://bonn0062.github.io/flask_model_deployment/.

Source
Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

master branch ▾

Theme Chooser
Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

[Choose a theme](#)

Custom domain
Custom domains allow you to serve your site from a domain other than bonn0062.github.io. [Learn more.](#)

[Save](#)

☒ **Enforce HTTPS**
— Required for your site because you are using the default domain (bonn0062.github.io)

Зачекайте хвилину-другу, а потім оновіть сторінку або спробуйте відкрити потрібну вам посилання. Як тільки сайт буде розміщений, ви побачите наступне:

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at https://bonn0062.github.io/flask_model_deployment/

Source

Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

master branch ▾

Theme Chooser

Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

Choose a theme

Custom domain










Custom domains allow you to serve your site from a domain other than bonn0062.github.io. [Learn more.](#)

Save

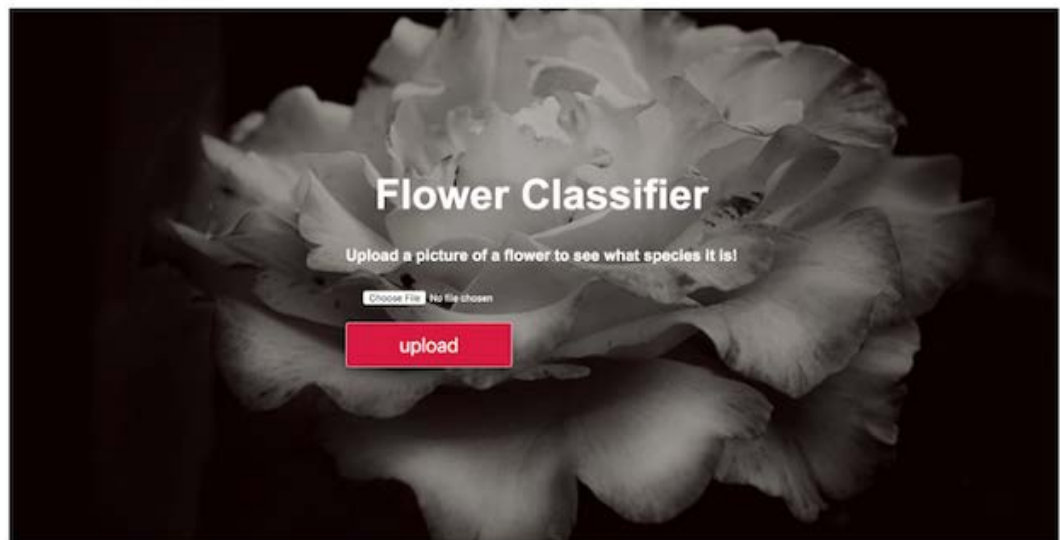
☒ Enforce HTTPS

— Required for your site because you are using the default domain (bonn0062.github.io)

Клацніть по цьому посиланню. Результат – ваш сайт-сторінка готовий!

← → ↻ https://bonn0062.github.io/flask_model_deployment/ ☆         

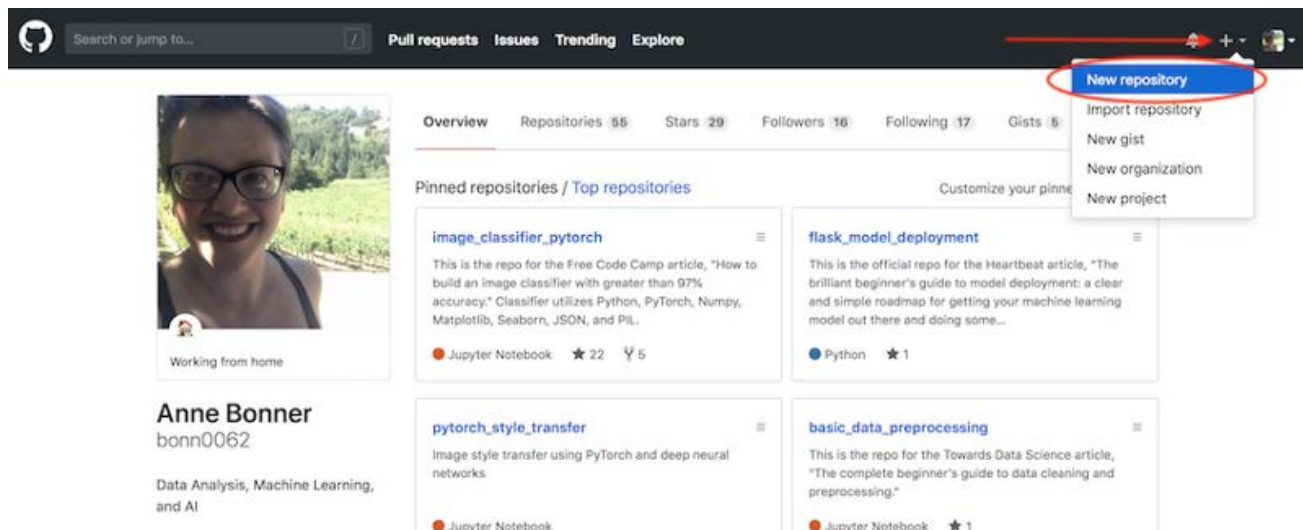
article, [How to show off your machine learning model: A clear and simple roadmap for deploying your machine learning model and getting it to do something everyone can understand](#)



You built this amazing machine learning model, but now what? How do you take your model and turn it into something that you can display on the web? How do you turn it into something that other people can interact with? How do you make it useful?

На оцінку «добре»:

Створення сайту «з нуля» в GitHub: Для початку створимо новий репозиторій.



Вкажіть назву сховища, його короткий опис, відзначте галочкою Initialize this repository with a README і натисніть Create repository.

Create a new repository

A repository contains all project files, including the revision history.

Owner: bonn0062 ▾ / Repository name: ✓

Great repository names are short and memorable. Need inspiration? How about **fluffy-eureka**?

Description (optional):

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾ | Add a license: **None** ▾ ⓘ

Create repository

Тепер йдемо в Settings (правий верхній кут) і прокручуємо до розділу GitHub Pages. Відкриваємо меню, що випадає, міняємо його значення з None на master branch.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source
GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

None ▾

Select source

- master branch**
Use the master branch for GitHub Pages.
- master branch /docs folder
Use only the /docs folder for GitHub Pages.

☒ **None**
Disable GitHub Pages.

Make this repository private
Hide this repository from the public. [Make private](#)

Transfer ownership
Transfer this repository to another user or to an organization where you have the ability to create repositories. [Transfer](#)

Спочатку ви побачите ось це:
GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at https://bonn0062.github.io/beginner_website/.

Source
Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

master branch ▾

Theme Chooser
Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

[Choose a theme](#)

Custom domain
Custom domains allow you to serve your site from a domain other than bonn0062.github.io. [Learn more.](#)

[Save](#)

☒ **Enforce HTTPS**
— Required for your site because you are using the default domain (bonn0062.github.io)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)

Потім:

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at https://bonn0062.github.io/beginner_website/

Source
Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

master branch ▾

Theme Chooser
Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

Choose a theme

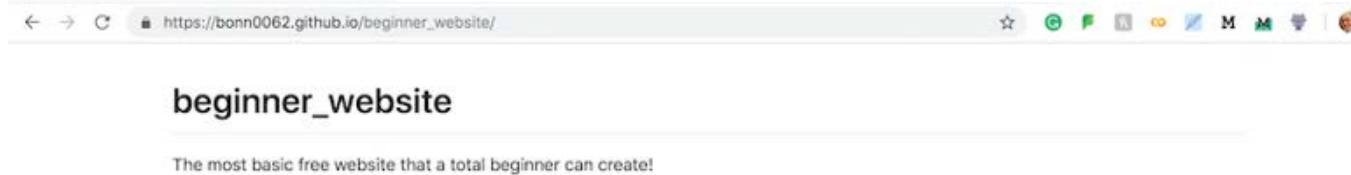
Custom domain
Custom domains allow you to serve your site from a domain other than bonn0062.github.io. [Learn more.](#)

Save

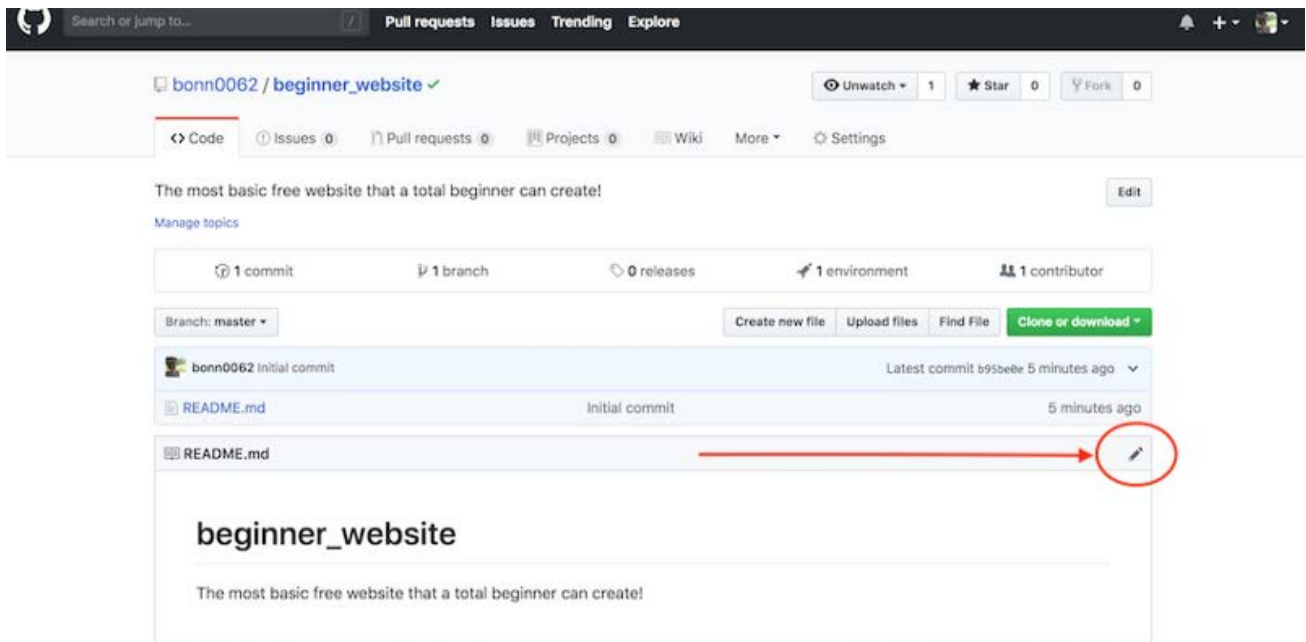
☒ **Enforce HTTPS**
— Required for your site because you are using the default domain (bonn0062.github.io)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)

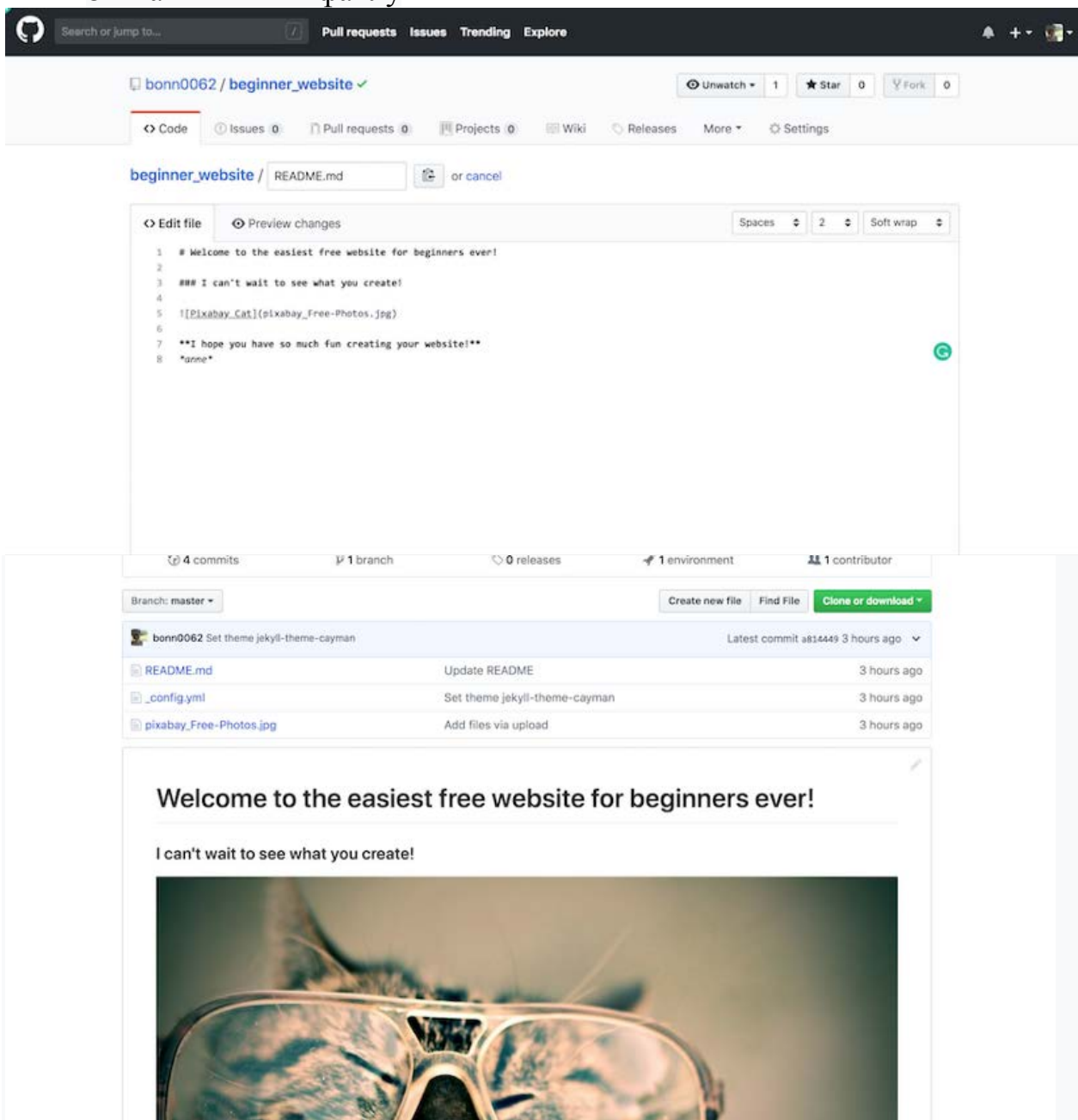
Клацніть по посиланню.



Для внесення змін відкоригуйте README-файл і додайте туди все, що вважаєте за потрібне. Поверніться до свого репозиторій, клікніть по іконці з олівцем і почніть змінювати README.

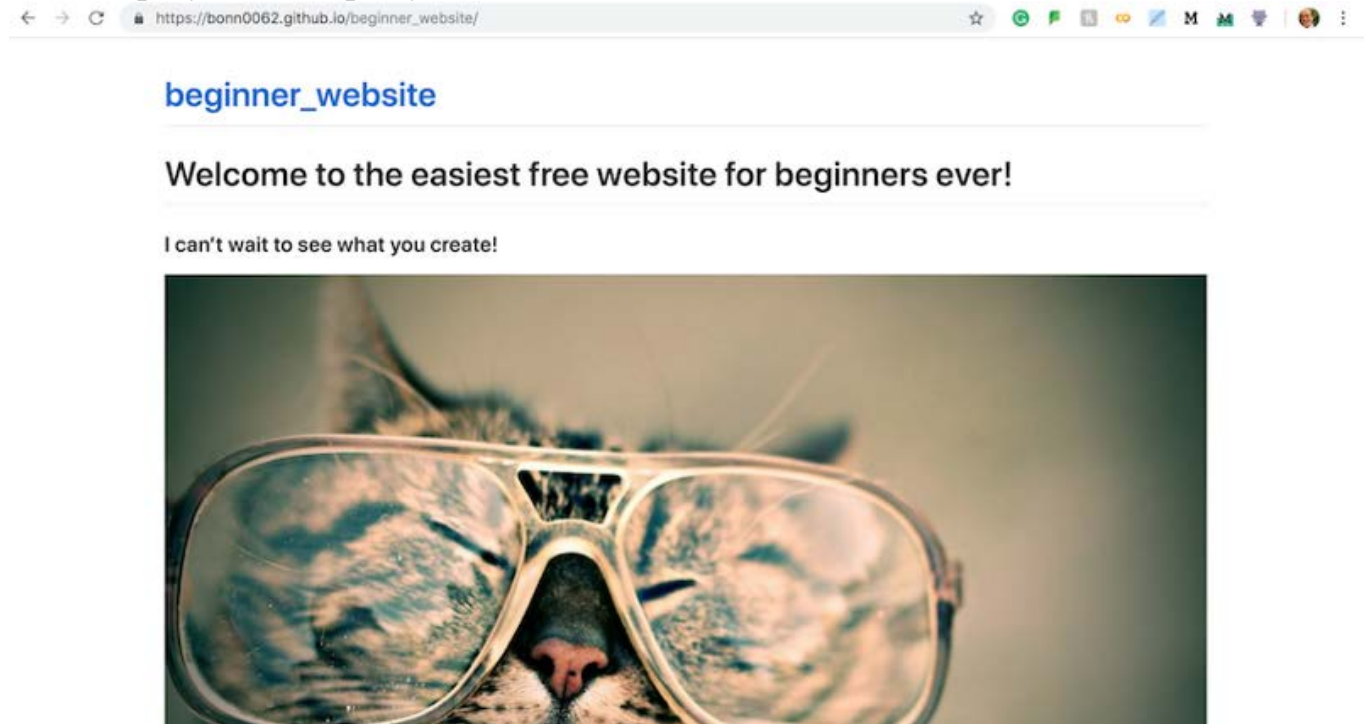


Зміна README-файлу



При додаванні зображень в README, потрібно обов'язково завантажувати їх в репозиторій. Інакше GitHub не зрозуміє, про що мова!

В результаті отримуємо:



Досить часто для вступу в силу змін потрібно почекати кілька хвилин. Якщо раптом ваш браузер постійно відкочується до попередньої версії сайту, то просто почистить історію в браузері за останню добу.

Давайте повернемося до GitHub Pages в Settings і натиснемо Choose a theme.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

None ▾

Theme Chooser

Select a theme to publish your site with a Jekyll theme using the master branch. [Learn more.](#)

Choose a theme

Danger Zone

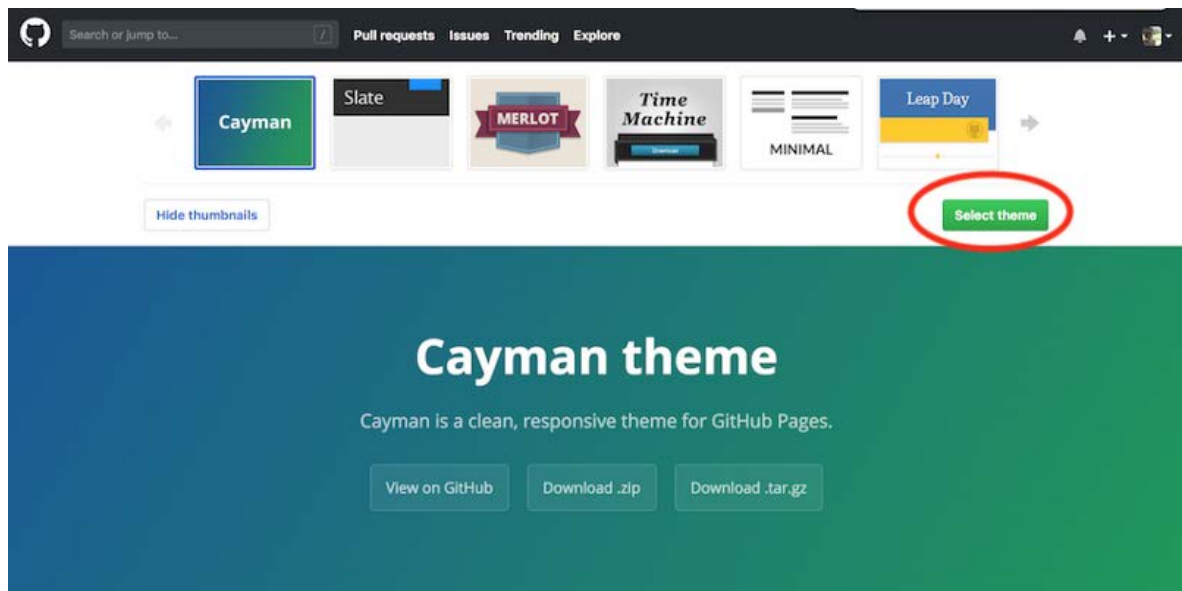
Make this repository private
Hide this repository from the public.

Make private

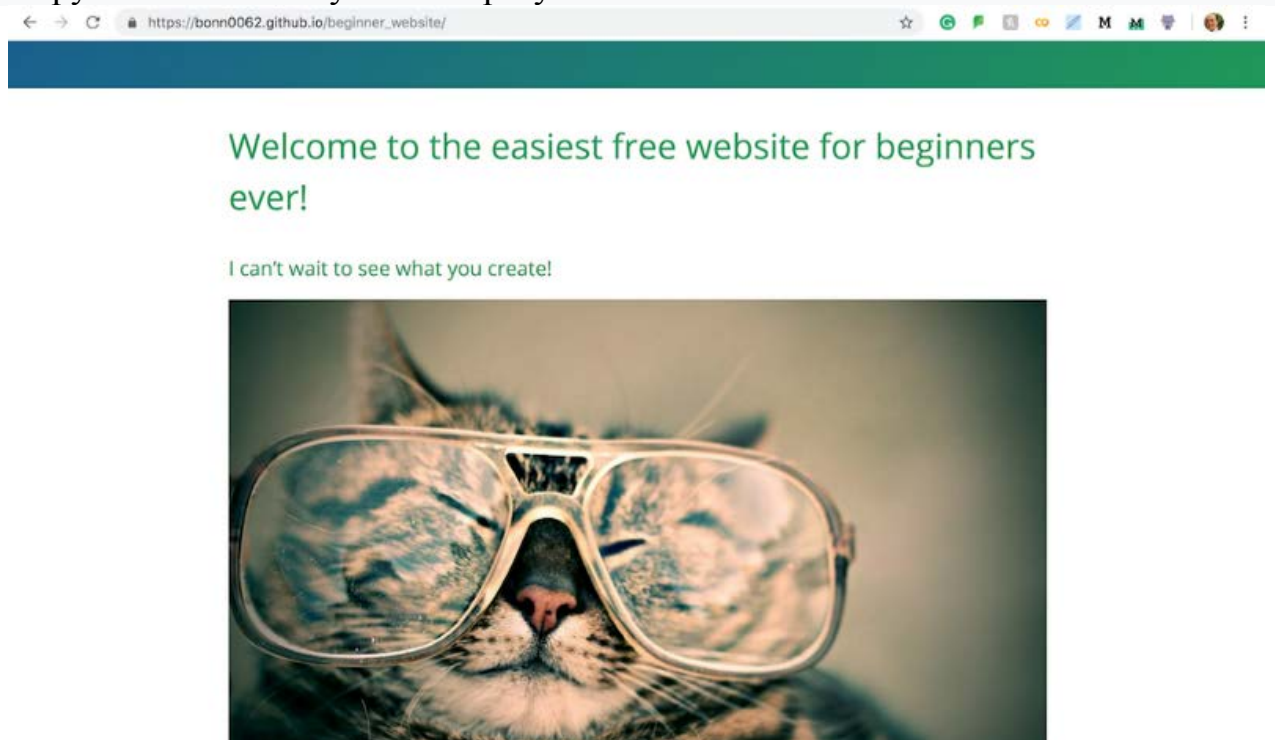
Transfer ownership
Transfer this repository to another user or to an organization where you have the ability to create repositories.

Transfer

Перевіримо, як зміниться сайт при виборі першої з доступних тем. Для цього натисніть зелену кнопку Select theme, трохи почекайте і ще раз відкрийте свій сайт.



Пару хвилин по тому бачимо результат:

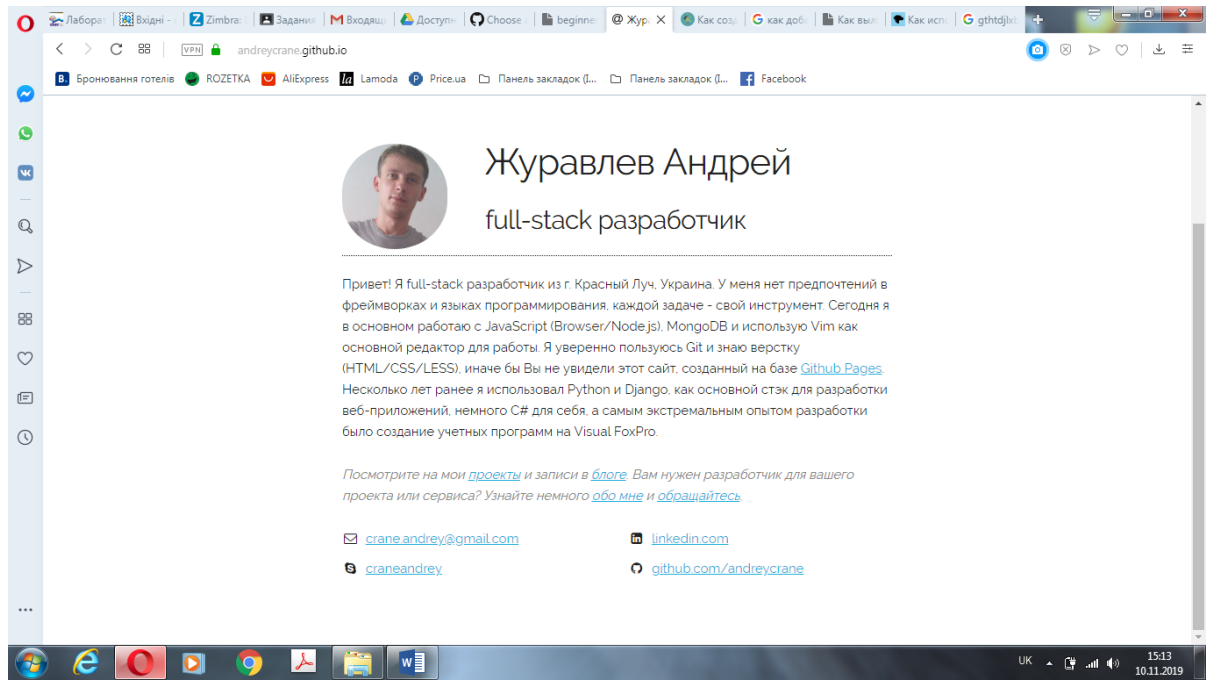


Якщо раптом вам перестала подобатися поточна тема, то її можна за просто видалити. У репозиторії є файл `_config.yml`, в якому розміщується інформація про тему. Видаліть цей файл, і ваша тема відійде разом з ним!

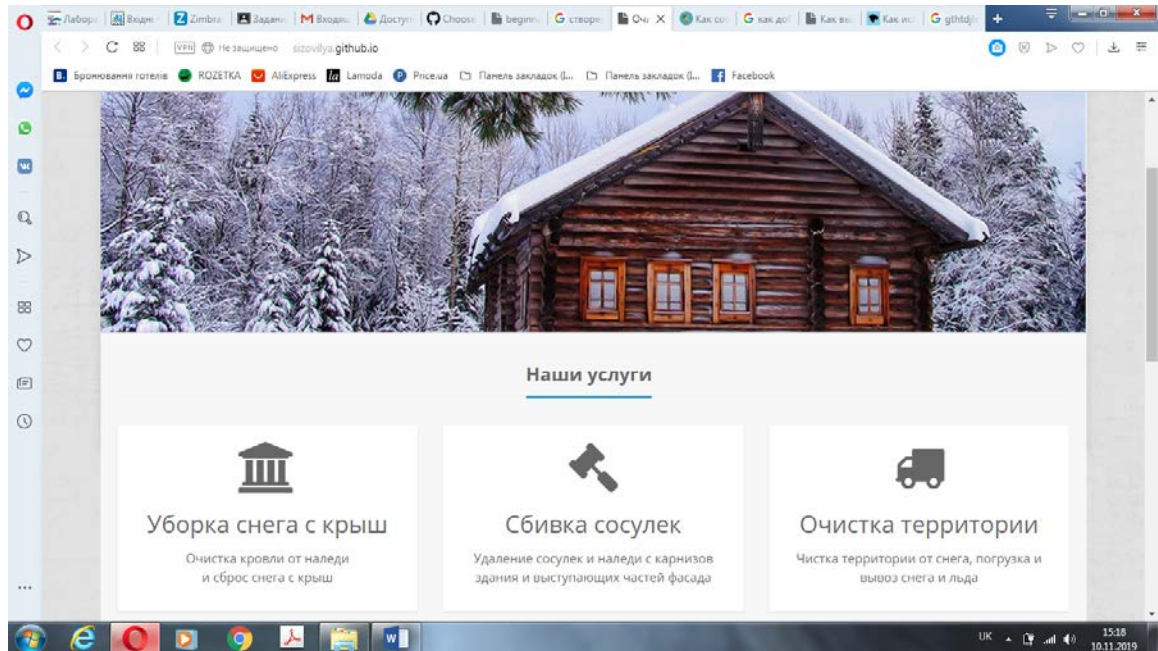
Якщо ви хочете змінити тему або щось в неї додати, то знову ж таки шукайте файл `_config.yml` і міняйте там все, що захочете.

На оцінку «відмінно»:

Створити особисту сайт-візитку за зразком:



або



КОНТРОЛЬНІ ПИТАННЯ:

1. Що таке репозиторій?
2. Навіщо потрібні розподілені системи?
3. Що таке оновлення робочої копії проекту?
4. Що таке фіксація змін проекту?
5. Що таке модифікація проекту?
6. Для чого використовується розгалуження в СКВ?
7. Що таке «Злиття версій»?
8. Що розуміють під конфліктом при роботі СКВ?
9. За яких умов злиття версій проходить автоматично і без конфліктів?
10. Чи призводить видалення та зміна одного і того ж файлу або каталогу у різних версіях до конфлікту при їх злитті?

РЕКОМЕНДОВАНА ЛІТЕРАТУРА:

- 6 Білас О. Якість програмного забезпечення та тестування: навч. посіб. / О. Білас. – Львів : вид-во НУ «Львівська політехніка», 2011.– 216 с.
- 7 Лавріщева К.М. Програмна інженерія. –Підручник.–К.: Академперіодика, 2008.– 415с.
- 8 Левус Є. В., Мельник Н. Б. Вступ до інженерії програмного забезпечення: Навчальний посібник. Львів: Видавництво Львівської політехніки, 2018. 248 с.
- 9 Основы инженерии качества программных систем / Ф.И.Андон, Г.И.Коваль, Т.М. Коротун, Е.М.Лаврищева, В.Ю. Суслов - К.: Академперіодика-2007.-678 с.
- 10Соммервилл И. Инженерия программного обеспечения. 6-е изд. / Соммервилл И. – М.: "Вильямс", 2006. – 624 с.