Отчёт по первому этапу проекта

Операционные системы

Сячинова Ксения

Содержание

# 1 Цель работы

Размещение на Github pages заготовки для персонального сайта.

# 2 Задание

* Установить необходимое программное обеспечение.
* Скачать шаблон темы сайта.
* Разместить его на хостинге git.
* Установить параметр для URLs сайта.
* Разместить заготовку сайта на Github pages.

# 3 Выполнение лабораторной работы

1. Скачиваем исполняемый файл Hugo для генерации страниц сайта.Переходим по ссылке и скачиваем последнюю версию сайта. Для работы в компьютерном классе на нашу операционную истему мы вибираем файл hugo\_extended\_0.98.0\_Linux-64bit.tar.gz.(рис. 1)(рис. 2)

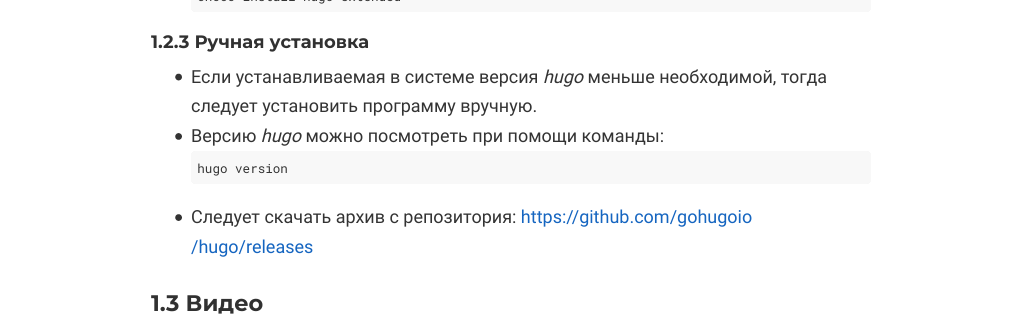


Рис. 1: Скачивание файла

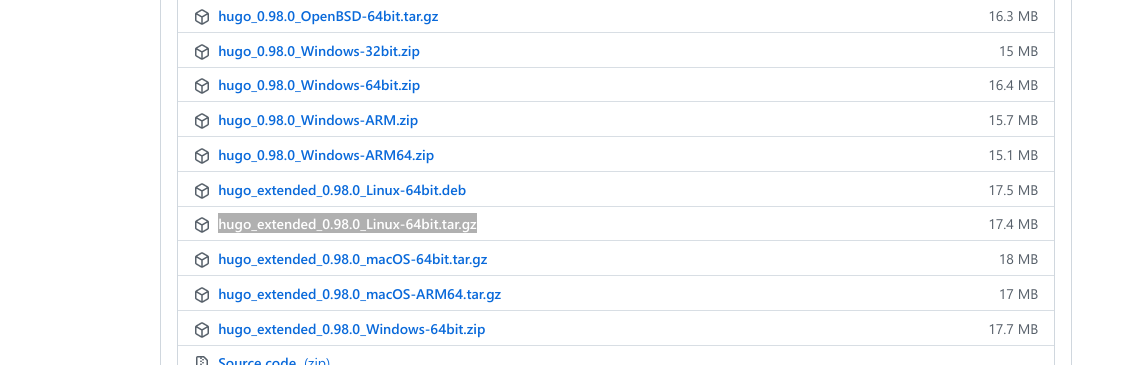


Рис. 2: Выбор для операционной системы

1. Затем извлекаем текущую папку из архива. Вырезаем исполняемый файл hugo в папку bin, предварительно создав её в домашней папке. Затем переходим в Konsole.
2. Далее сделаем клон репозитория. Для этого сначала создаём его, я назвала мой репозиторий “Project”. После этого копируем ссылку и с помощью команды git clone --recursive создаём копию.(рис. 3)

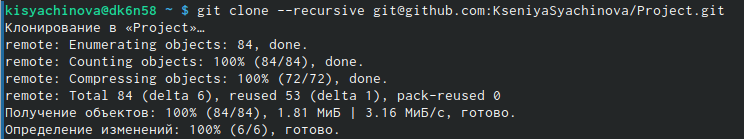


Рис. 3: Копия репозитория

1. Переходим в “Prijects”. С помощью команды ls -l видим наши файлы. После этого выполняем команду ~/big/hugo обязательно ставим знак тильды, так как мы переходим в каталог пользователя. Удалим папку public для упрощения работы.(рис. 4)

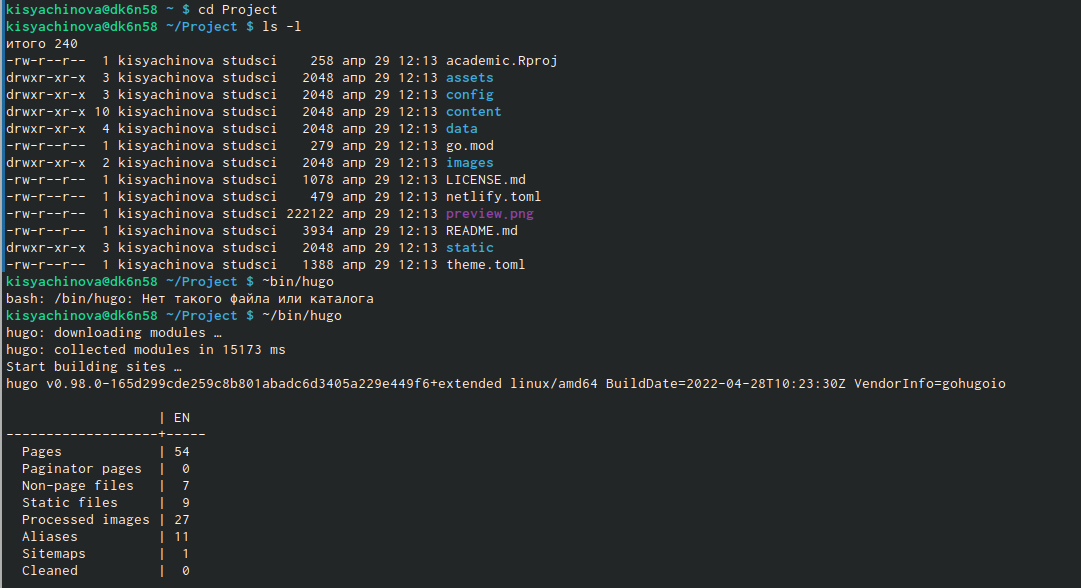


Рис. 4: Переход в папку

1. Используем команду ~/bin/hugo server и при правильном выполнении мы видим ссылку на наш сайт. Копируем и вставляем в браузер. При выполнении данной команды мы автоматически создали файлы. Этот сайт виден только нам. (рис. 5)

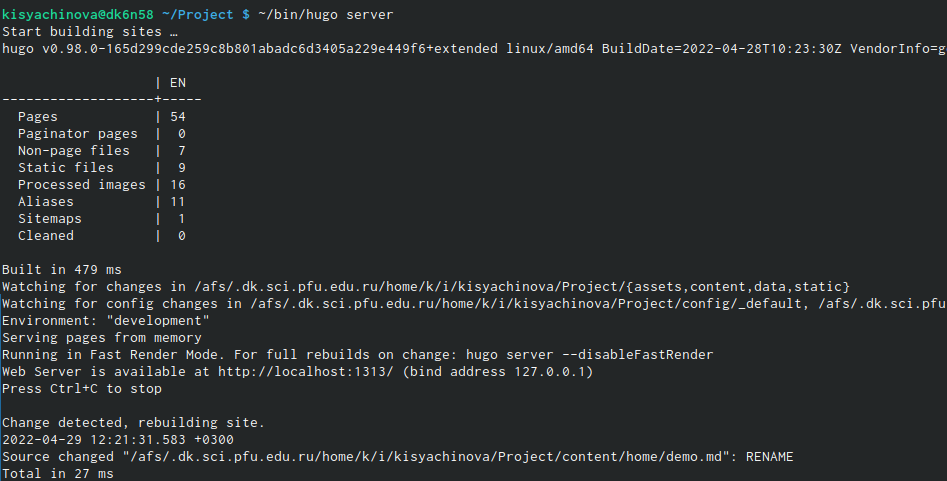


Рис. 5: Создания сайта

1. Уберём зелёное поле, которое сейчас находится на нашем сайте. Для этого нужно удалить одну папку. Пусть к этой папке прописан на сайте: content/home/demo.mg. После удаления видим исчёзновение зеленого фона. (рис. 6), (рис. 7)

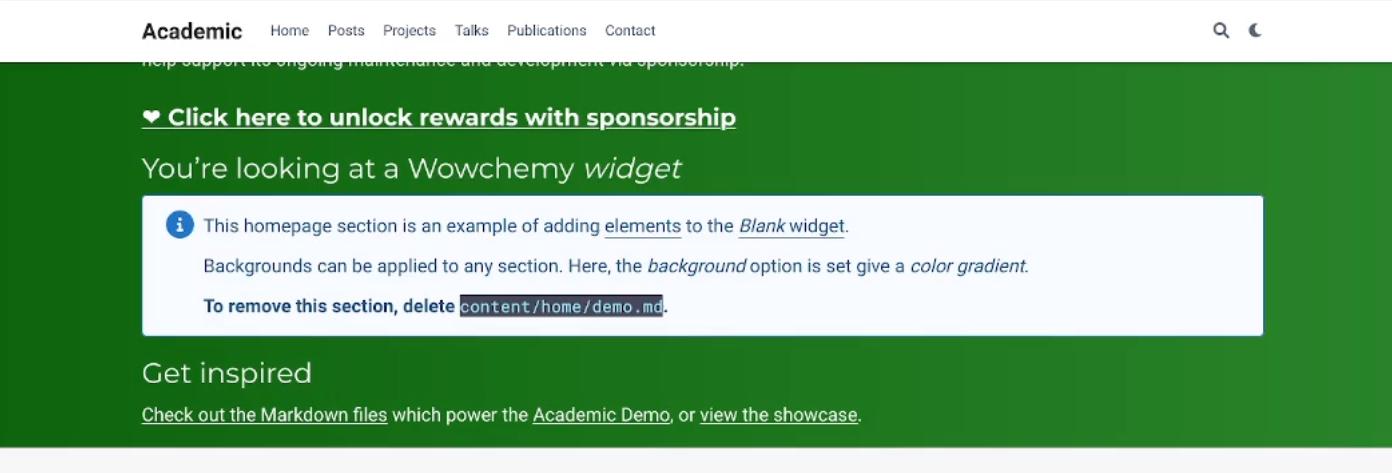


Рис. 6: Удаление файла

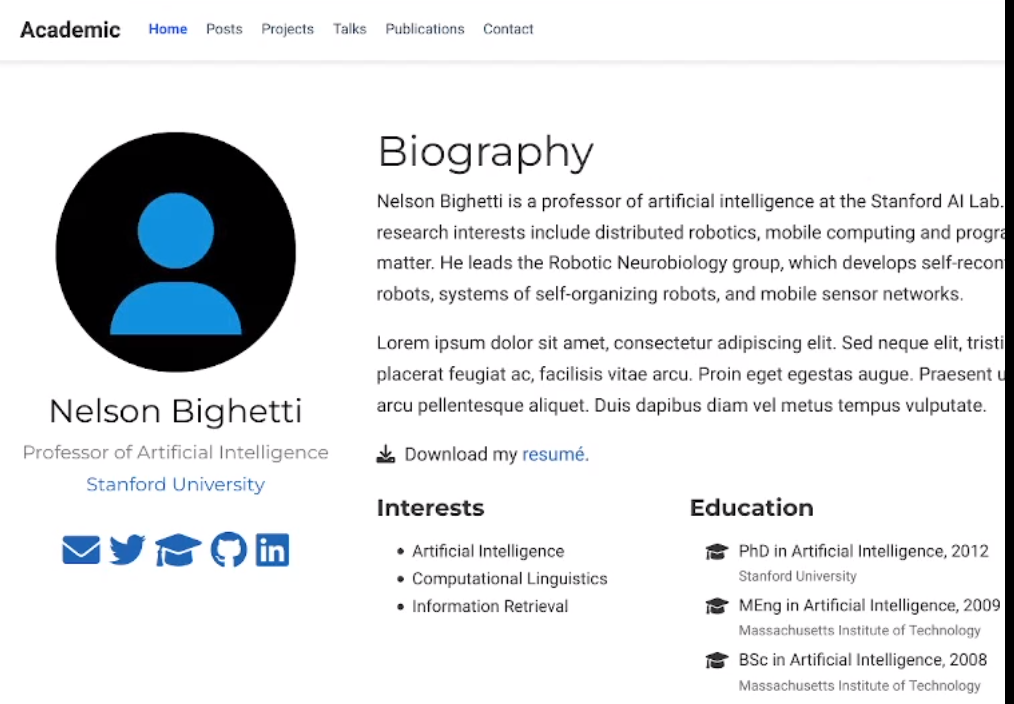


Рис. 7: Итог

1. Теперь переносим сайт на репозиторий. Для этого создаём ещё один репозиторий с определённым именем. В моём случае это KseniyaSyachinova.github.io (рис. 8)

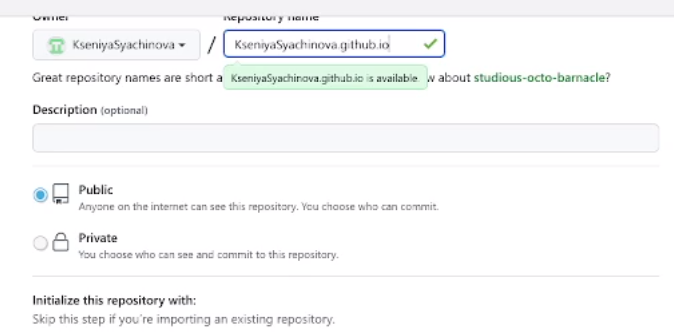


Рис. 8: Создание репозитория

1. После создания репозитория переходим в консоль, поднимаемся на уровень выше и рядом с ним клонируем наш созданный репозиторий. Проверяем с помощью команды ls.(рис. 9)

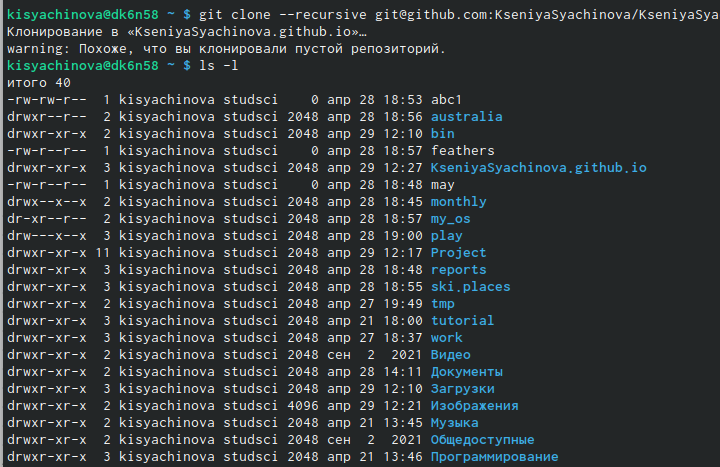


Рис. 9: Клонирование нового репозитория

1. Переходим в репозиторий, создаёт ветку main с помощью команды git checkout -b main, создаём пустой файл для активации репозитория, добовляем его с помощью знакомых действий.(рис. 10)

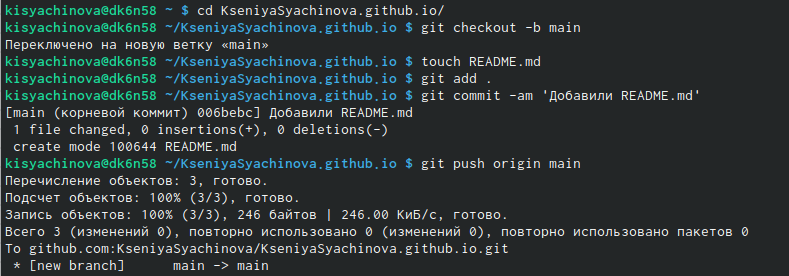


Рис. 10: Создание ветки main

1. Переходим в наш Project и теперь подключаем репозиторий к папке public внутри нашего блока для эффективной регенирации страниц нашего сайта. Для этого используем команду git submodule add -b main git@github.com:KseniyaSyachinova/KseniyaSyachinova.github.io.git public. Подключаем обратно наш publiс. Комментируем его #public и проверяем. Опять используем нашу команду и видим изменения. рис. 11)

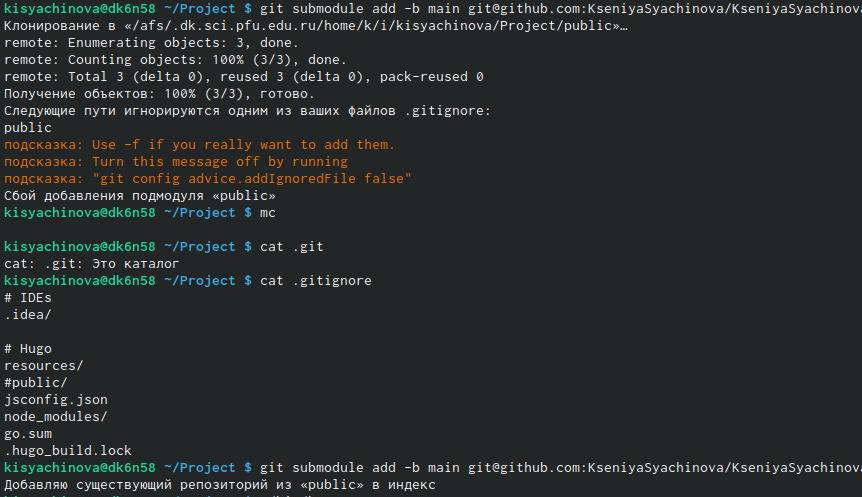


Рис. 11: Поключение к public

1. С помощью ~/bin/hugo мы автоматически добавляем файлы в нашу папку, которые в будущем будут являтся файлами нашего сайта.(рис. 12)

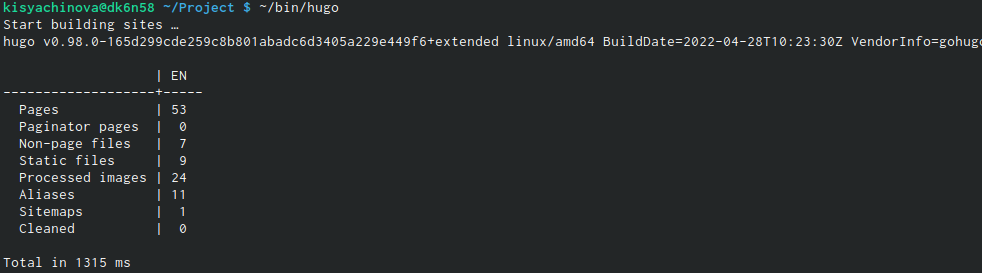


Рис. 12: Автоматическое подключение

1. Синхронизируем эти файлы с репозиторием. Возвращаемся в public, для проверки делаем git remote -v. (рис. 13), (рис. 14),

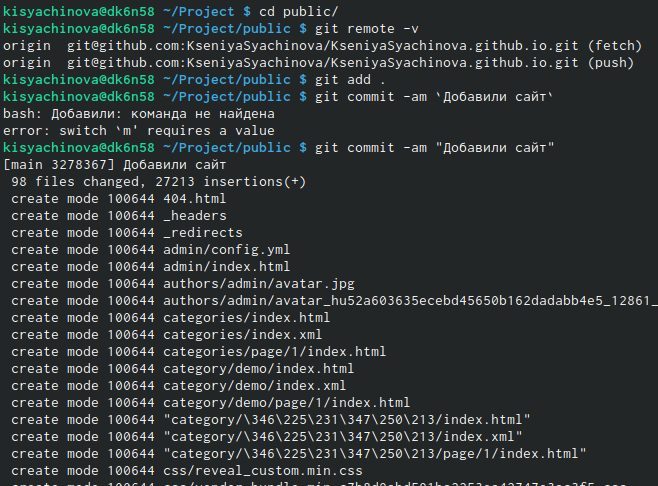


Рис. 13: Синхронизация

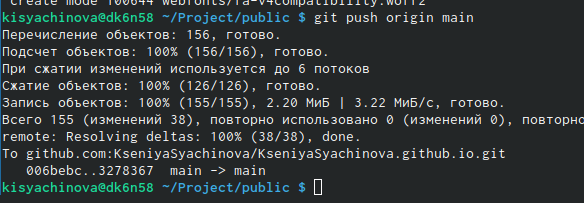


Рис. 14: Синхронизация

1. Обовляем наш репозиторий и видим все файлы. Копируем ссылку, вставляем в браузер и видим наш новый сайт. На это первый этап закончен. (рис. 15), (рис. 16),

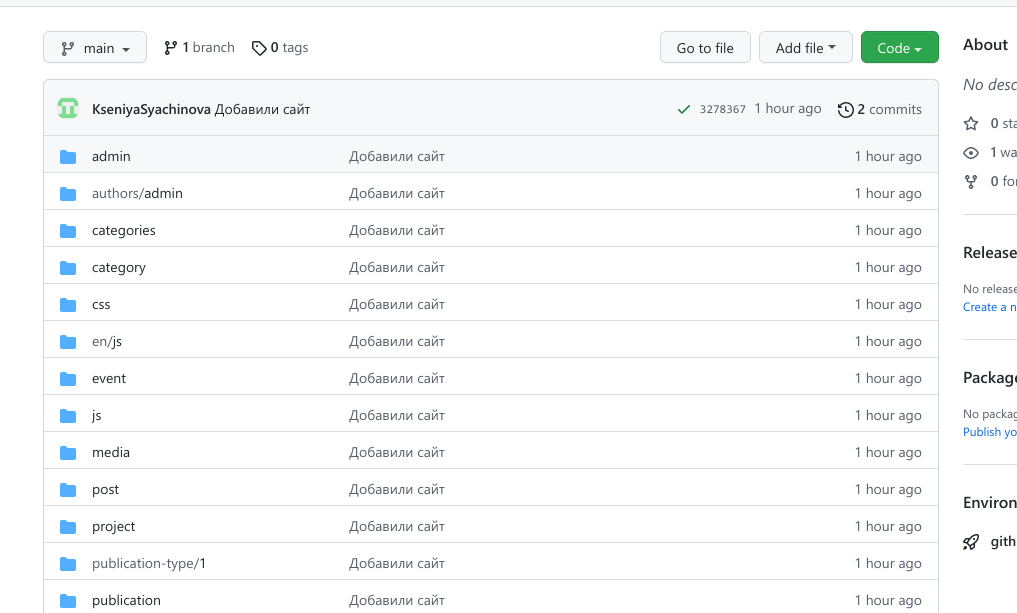


Рис. 15: Синхронизация

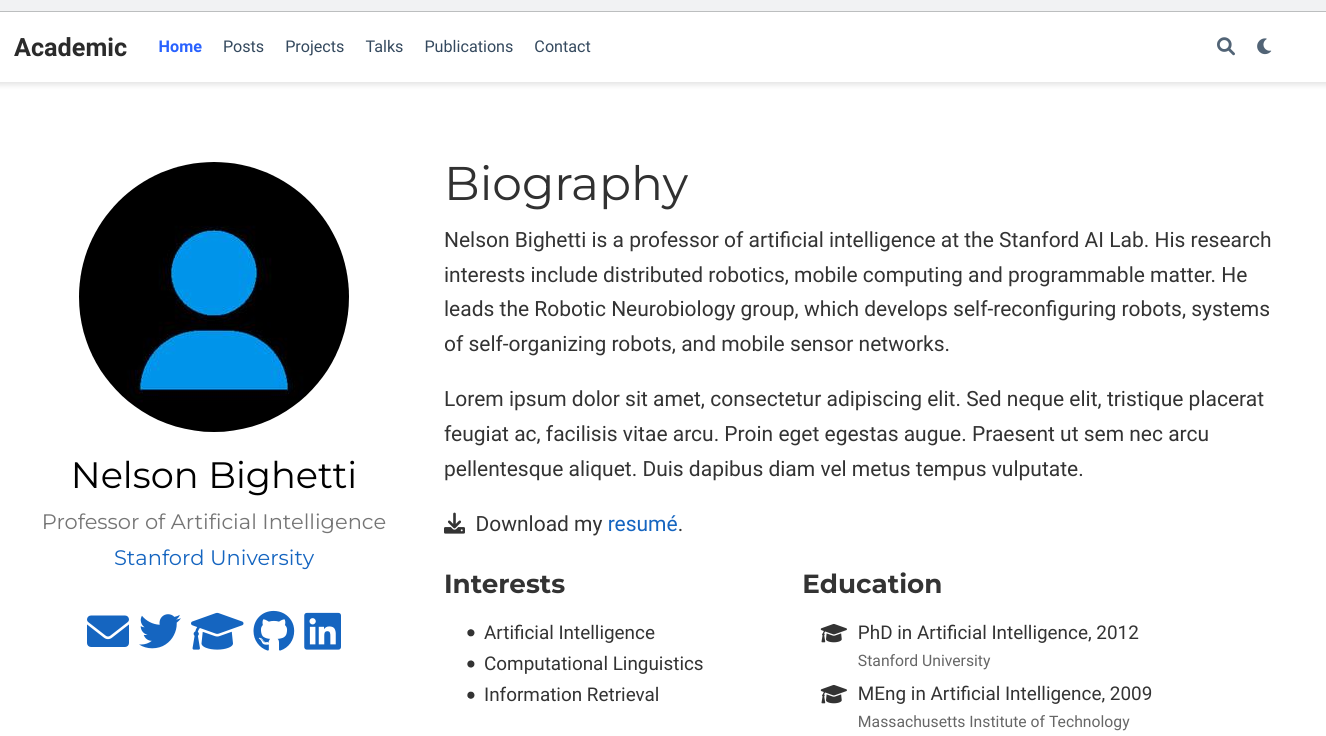


Рис. 16: Синхронизация

# 4 Выводы

После выполнения первго этапа проекта я научилась:

* Устанавливать необходимое программное обеспечение.
* Скачивать шаблон темы сайта.
* Размещать его на хостинге git.
* Установливать параметр для URLs сайта.
* Размещать заготовку сайта на Github pages.