Отчет по выполнению первого этапа индивидуального проекта

Дисциплина: Операционные системы

Анастасия Александровна Астраханцева

Содержание

# 1 Цель работы

Создать сайт на Hugo.

# 2 Задание

1. Установить необходимое программное обеспечение.
2. Создать репозиторий на основе шаблона.
3. Подключить его к git.
4. Разместить заготовку сайта на Githyb pages.

# 3 Теоретическое введение

Сайт – это информационный, коммерческий или развлекательный ресурс в интернете, который состоит из страничек с разнообразным контентом. Чтобы «завести» веб-сайт необходимо зарегистрировать доменное имя, привязать его к хостингу – сервису, который размещает страницы в сети.

Hugo — один из самых популярных генераторов статических сайтов с открытым исходным кодом, написан на языке Go. Благодаря своей удивительной скорости и гибкости, Hugo делает создание веб-сайтов увлекательным.

Он подойдет как для создания лендингов, «многостраничников» — так и для создания более сложных продуктов: блог, новостной портал и даже интернет-магазин.

# 4 Выполнение лабораторной работы

Для начала нам нужно установить необходимое ПО, а именно hugo. Выбираем версию extended и новейшее обновление (у меня 0.110.0) (рис. [1](#fig:001)).

Figure 1: Установка hugo

Figure 1: Установка hugo

Мы устанавиваем пакет, далее нам необходимо разархивировать его и исполняющий файл hugo перенести в каталог ~/bin, который мы создаем с помощтю коамнды mkdir (рис. [2](#fig:002) - [3](#fig:003)).

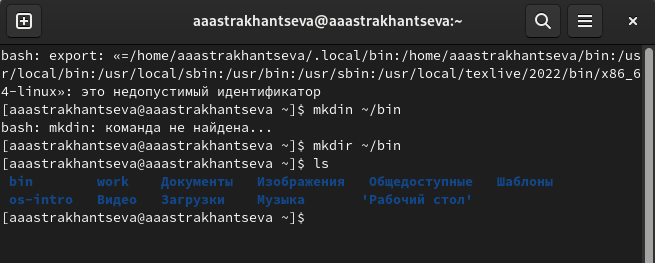


Figure 2: Создание папки bin

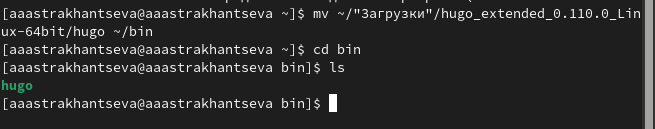


Figure 3: Перенос hugo в каталог bin

У нас есть шаблон репозитория, на основе которго мы будем создавать сайт. Нам необходимо добавить данный репозиторий к себе, назвать его можно как “blog” (рис. [4](#fig:004)).

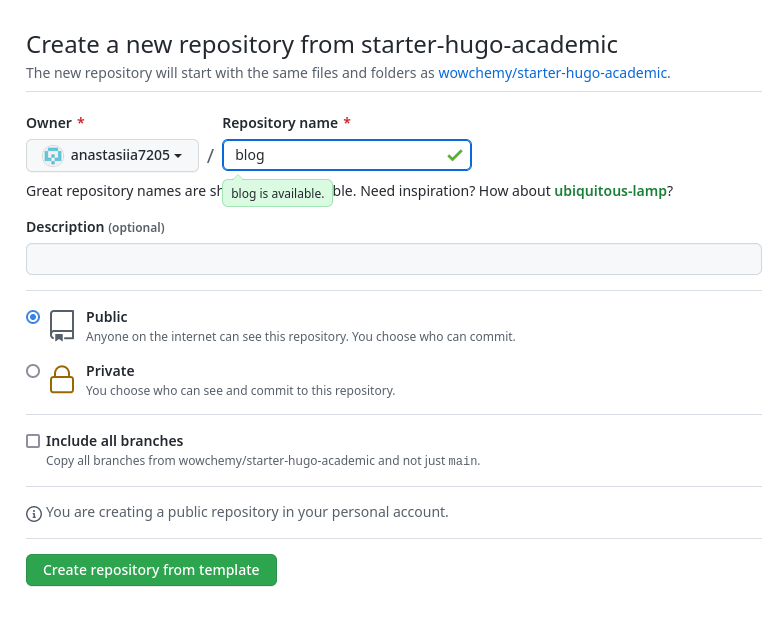


Figure 4: Создание репозитория

Далее нам необходимо клонировать созданный ранее репозиторий (рис. [5](#fig:005)).

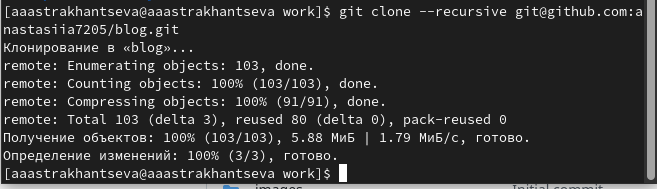


Figure 5: Клонирование репозитория

После этого переходим в каталог “blog”, который был создан при клонировании репозитория. Проверяем наличие всех файлов в данном каталоге (рис. [6](#fig:006)).

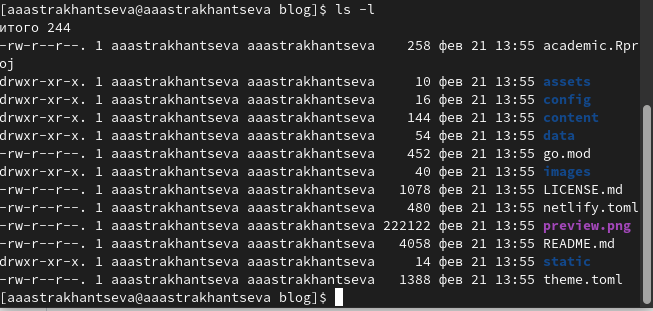


Figure 6: Проверка наличия файлов в каталоге “blog”

Запускаем hugo в каталоге “blog”. После этого проверяем содержимое каталога: у нас появился каталог “public”, который нам нужно удалить. Сделаем мы это с помощью mc. В дальнейшем мы создадим его снова (рис. [7](#fig:007) - [9](#fig:009)).

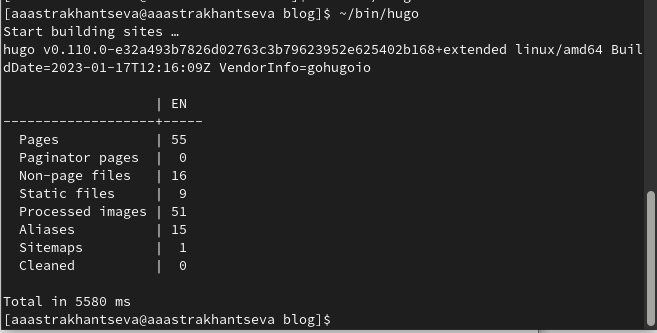


Figure 7: Запускаем hugo в каталоге “blog”

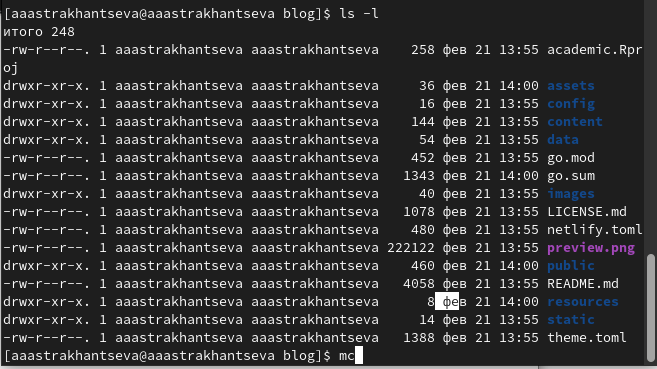


Figure 8: Проверка создания каталога “public” в каталоге “blog”

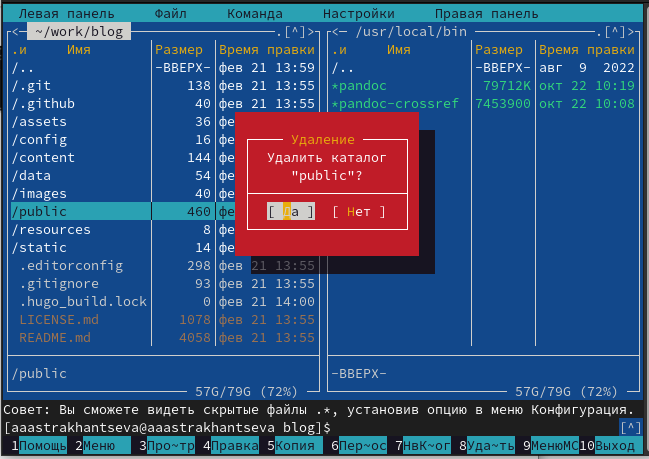


Figure 9: Удаление “public”

Далее нужно запросить ссылку на наш сайт с помощью команды ~/bin/hugo server. По данной ссылке мы переходим на наш сайт. Этот сайт пока что виден только на моем компьютере (рис. [10](#fig:010) - [12](#fig:012)).

![Figure 10: Получение ссылки с помощью hugo server](data:application/octet-stream;base64,)

Figure 10: Получение ссылки с помощью hugo server

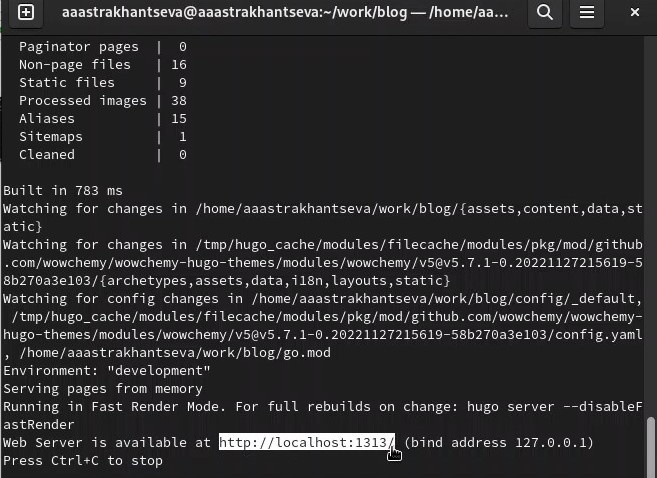


Figure 11: Ссылка на наш сайт

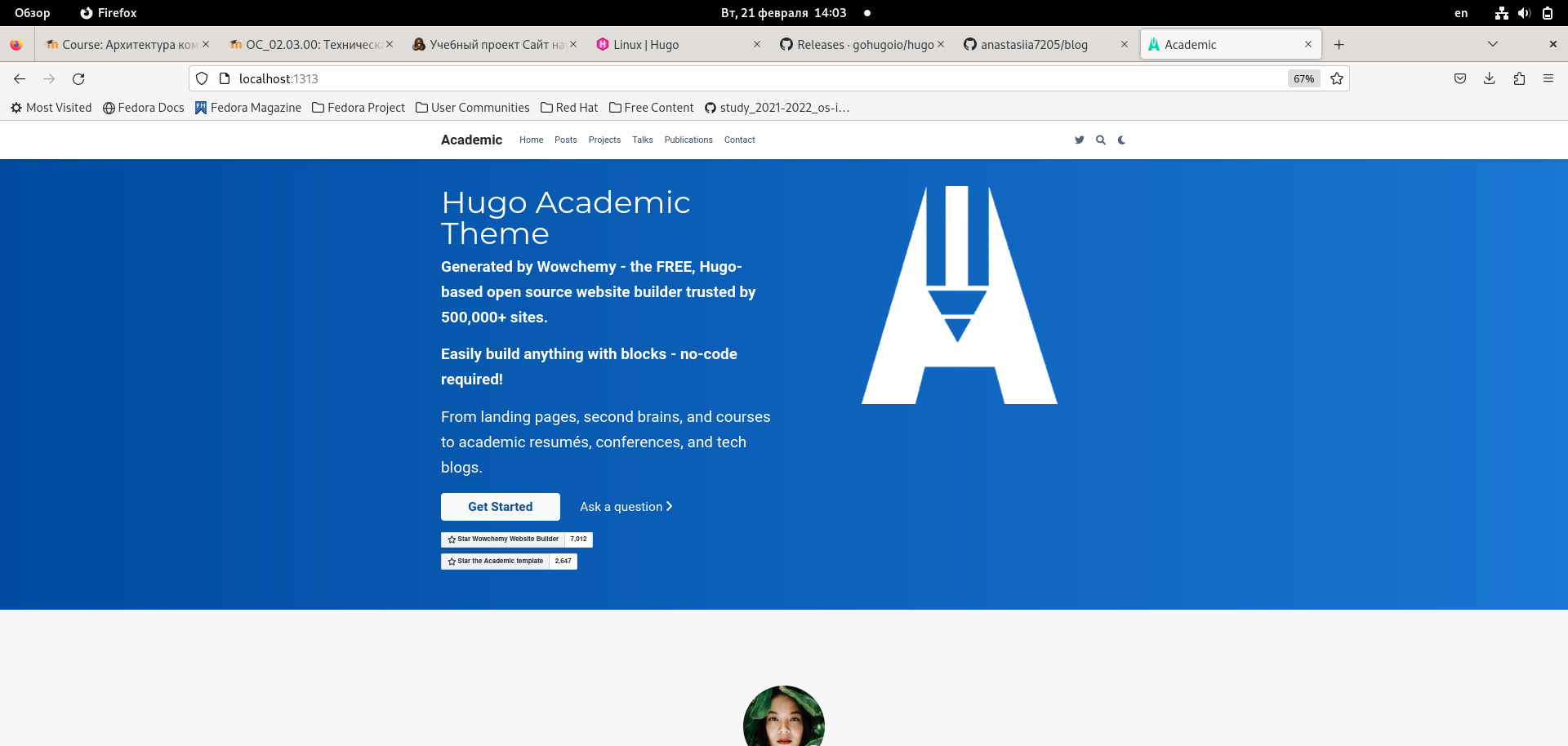


Figure 12: Сайт, домашняя страница

Далее на сайте Github создаем новый пустой репозиторий. Имя репозитория должно быть в таком формате: .github.io (рис. [13](#fig:013)).

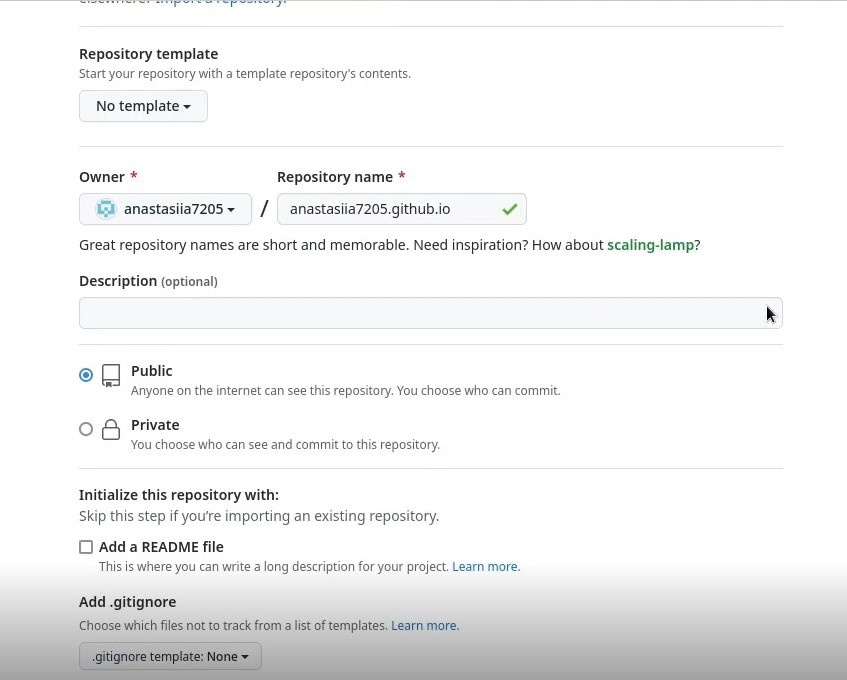


Figure 13: Создание репозитория

После того, как мы создали репозиторий, нам нужно скопировать ссылку на него (рис. [14](#fig:014)).

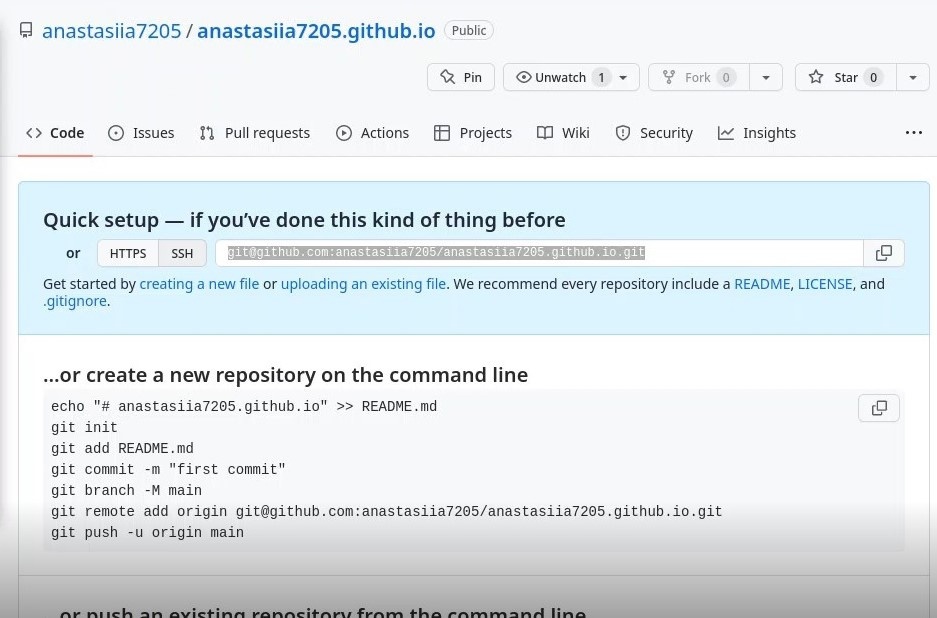


Figure 14: Ссылка на репозиторий

Далее клонируем созданный репозиторий с помощью ссылки, которую мы скопировали ранее. Проверяем то, что наш каталог с репозиторием появился с помощью команды ls -l. Переходим в созданный каталог и создаем ветку “main”. Создаем файл “README.md” (рис. [15](#fig:015)).

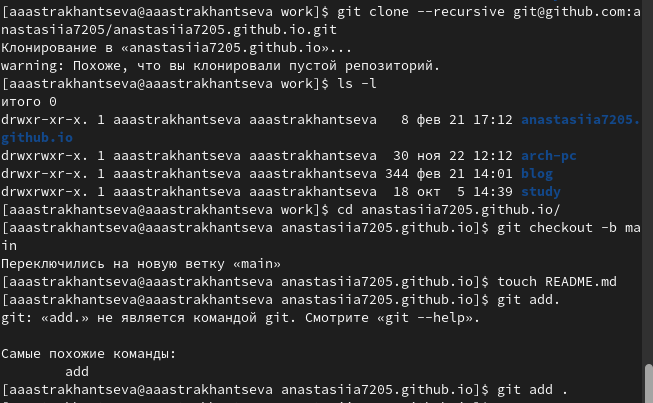


Figure 15: Клонировние репозитория, создание ветки “main” и файла “README.md”

Далее отправляем изменения на git (рис. [16](#fig:016)).

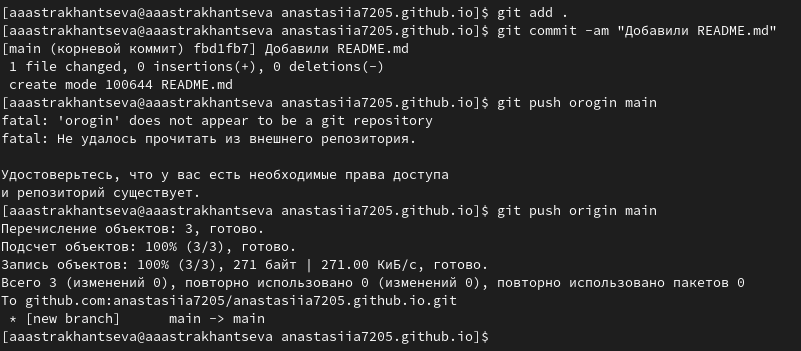


Figure 16: Отправка на git

Переходим в каталог “blog”. Привязыываем наш сайт к каталогу “public”. Все изменения, которые мы бедм производить в папке “public” будут отображаться на нашем сайте. С первого раза у меня не получилось это сделать, потому что “public” игнорируеся. Чтобы исправить это нужно в файле “.gitignore” закоментировать “public”. Я сделала это с помощью mc (рис. [17](#fig:017) - [18](#fig:018)).

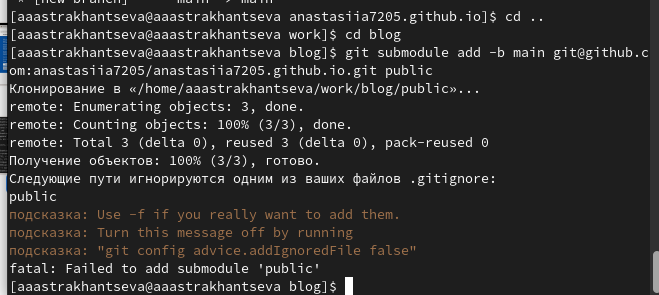


Figure 17: Игнорирование public

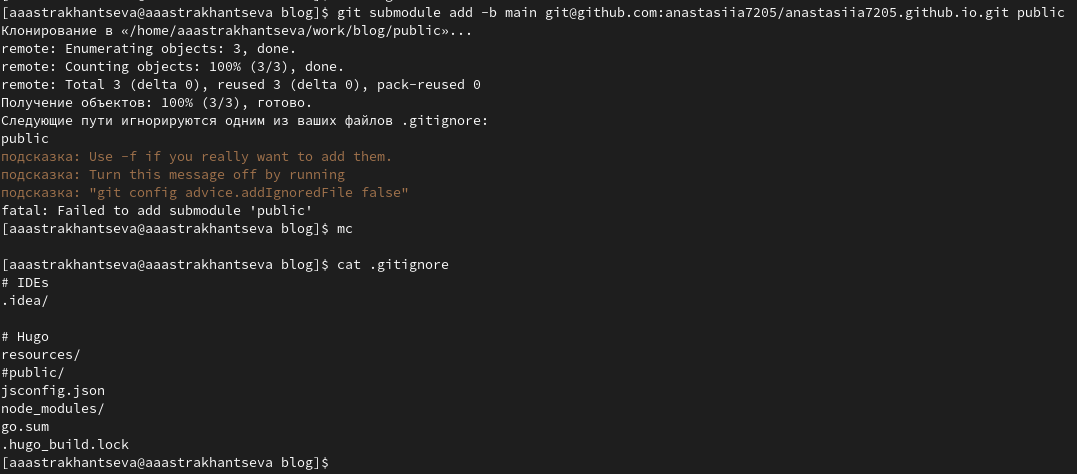


Figure 18: Изменение “.gitignore” (закоментировала public)

После того, как мы закоментрвали public, повторяем команду. Далее запускаем hugo и видим, что все файлы для редактированиия нашего сайта сохраняются в каталог public (рис. [19](#fig:019)).

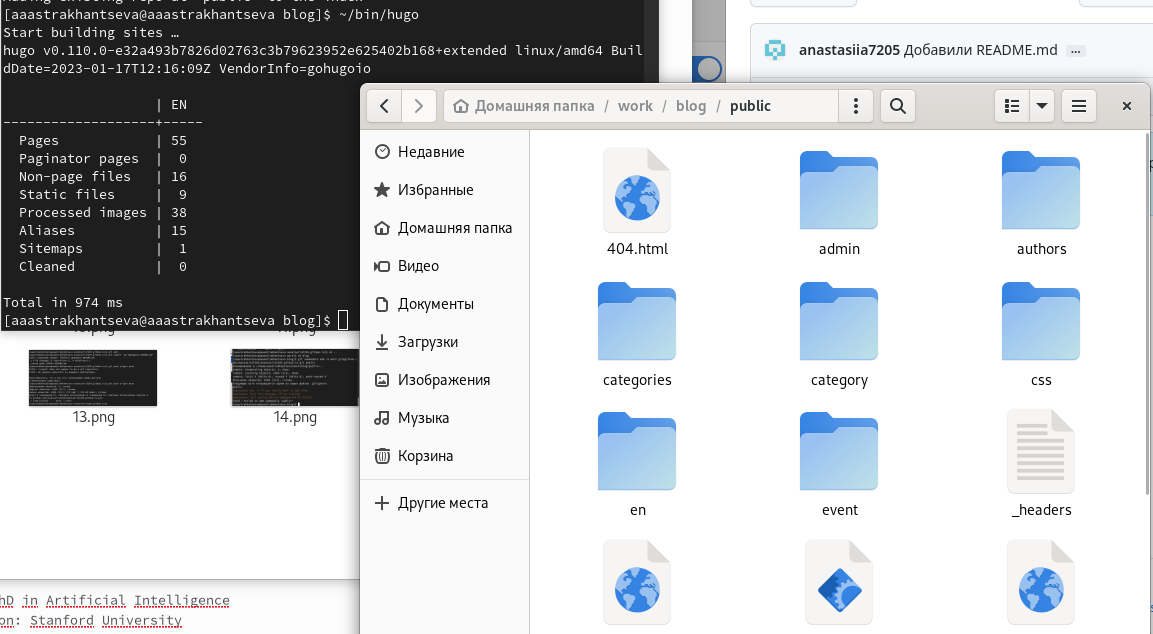


Figure 19: Сохранение каталогов для редактированиия нашего сайта в каталоге public

Переходим в каталог public и проверяем подключенние. После этого отправляем все изменения на Git (рис. [20](#fig:020) - [21](#fig:021)).

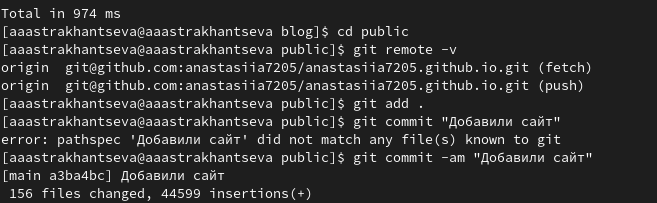


Figure 20: Проверка подключения и отправка файлов на сайт

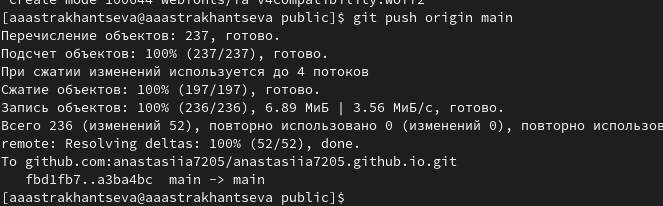


Figure 21: Отправка изменений на Git

Переходим на сайт Github в репозиторий для нашего сайта. Копируем из основной ссылки часть ссылки “.github.io”. С помощью данной ссылки мы можем попасть на наш сайт. Теперь он доступен не только на нашем компьютере, но и абсолютно любому пользователю (рис. [22](#fig:022)).

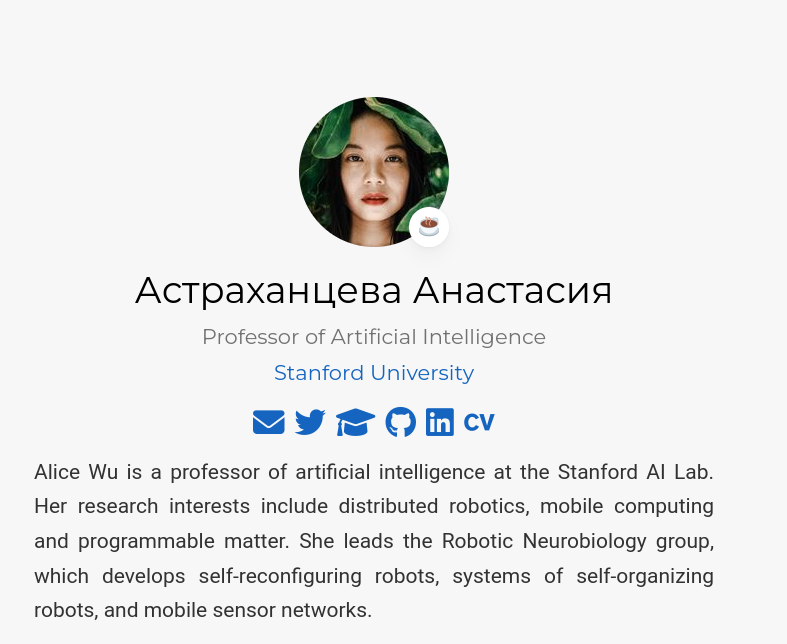


Figure 22: Мой сайт

# 5 Выводы

В ходе выполения первого этапа индивидуального проекта мною было установлено необходимое ПО. Выполены необходимые действия для создания сайта на Hugo.

# Список литературы

1. Сайт [электронный ресурс] - Режим доступа: https://www.calltouch.ru/blog/glossary/sajt/
2. Статический генератор сайтов Hugo. Фронтенд реалии [электронный ресурс] - Режим доступа: https://habr.com/ru/post/532738/