

# **Отчет по созданию проекта**

**1 этап проекта**

Гибшер Кирилл Владимирович

# Содержание

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>1</b> | <b>Цель работы</b>                    | <b>5</b>  |
| <b>2</b> | <b>Задание</b>                        | <b>6</b>  |
| <b>3</b> | <b>Теоретическое введение</b>         | <b>7</b>  |
| <b>4</b> | <b>Выполнение лабораторной работы</b> | <b>8</b>  |
| <b>5</b> | <b>Выводы</b>                         | <b>18</b> |
|          | <b>Список литературы</b>              | <b>19</b> |

## Список иллюстраций

|      |  |    |
|------|--|----|
| 4.1  | Установка hugo-extended . . . . .                      | 8  |
| 4.2  | Перемещение исполняемого файла hugo . . . . .          | 9  |
| 4.3  | Создание нового репозитория . . . . .                  | 9  |
| 4.4  | Клонирование репозитория . . . . .                     | 10 |
| 4.5  | Проверка файлов . . . . .                              | 10 |
| 4.6  | Запуск hugo . . . . .                                  | 11 |
| 4.7  | Удаление каталога public . . . . .                     | 11 |
| 4.8  | Hugo server . . . . .                                  | 12 |
| 4.9  | Переход на сайт по сгенерированной ссылке . . . . .    | 12 |
| 4.10 | Создание репозитория github.io . . . . .               | 13 |
| 4.11 | Клонирование репозитория github.io . . . . .           | 13 |
| 4.12 | Активация репозитория . . . . .                        | 14 |
| 4.13 | Git add submodule public . . . . .                     | 15 |
| 4.14 | Исправление файла gitignore и повтор команды . . . . . | 15 |
| 4.15 | Запуск файла hugo в blog . . . . .                     | 16 |
| 4.16 | Выгрузка сайта на github . . . . .                     | 16 |
| 4.17 | Проверка загрузки . . . . .                            | 17 |
| 4.18 | Проверка работоспособности сайта . . . . .             | 17 |

## **Список таблиц**

# 1 Цель работы

- Адаптировать виртуальную машину под дальнейшую работу с сайтом путем создания новых каталогов и внедрения генератора статических сайтов hugo.

## 2 Задание

1. Скачать расширение для работы с сайтом.
2. Используя шаблоны с github создать необходимые каталоги и настроить их работу.
3. Проверить работоспособность сайта после завершения всех настроек.

### 3 Теоретическое введение

**Сайт** - одна или несколько логически связанных между собой веб-страниц; также место расположения контента сервера. Обычно сайт в Интернете представляет собой массив связанных данных, имеющий уникальный адрес и воспринимаемый пользователями как единое целое.

## 4 Выполнение лабораторной работы

Устанавливаем архив hugo-extended на нашу ОС и затем распаковываем этот архив. (рис. [4.1]).

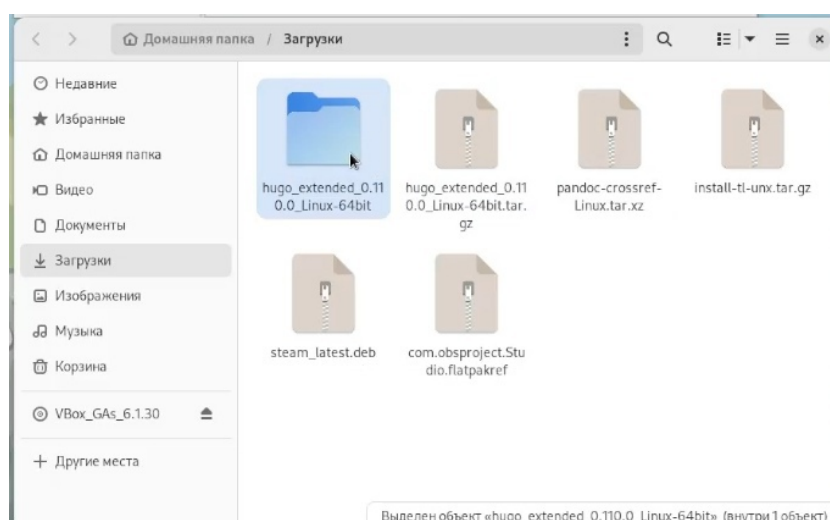


Рис. 4.1: Установка hugo-extended

Создаем в домашнем каталоге пользователя папку bin. Из исходного распакованного архива hugo-extended забираем только лишь исполняемый файл hugo и перемещаем в созданную папку bin. (рис. [4.2]).



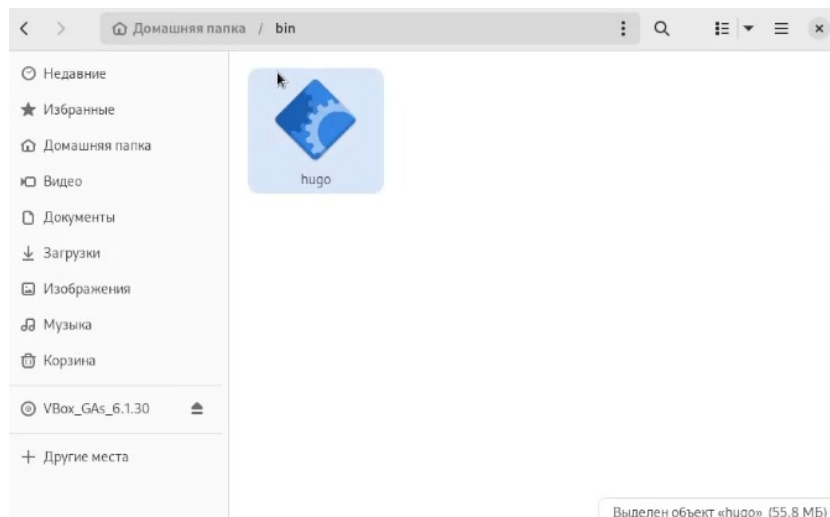


Рис. 4.2: Перемещение исполняемого файла hugo

Используя шаблон с github создаем новый репозиторий blog. (рис. [4.3]).

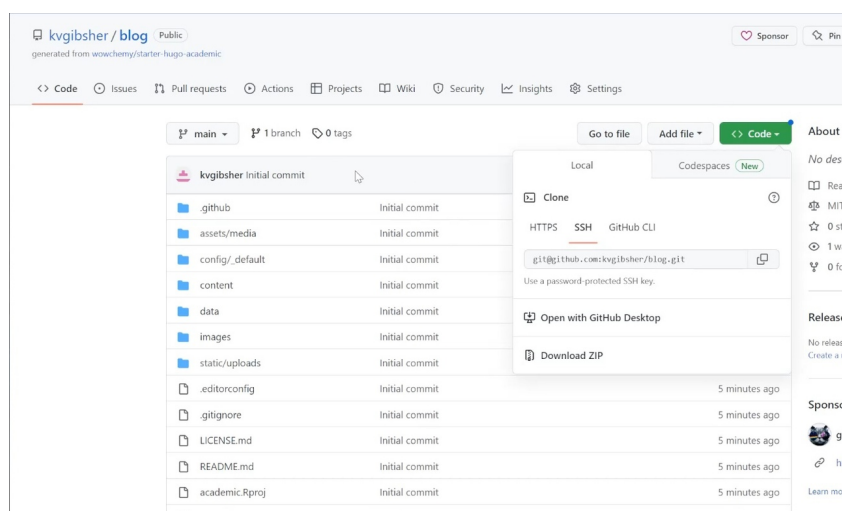


Рис. 4.3: Создание нового репозитория

С помощью команды `git clone` клонируем репозиторий blog в нашу ОС с последующим созданием каталога соответственно. (рис. [4.4]).

```
kvgibsher@kvgibsher:~ — git clone --recursive git@github.co...
[kvgibsher@kvgibsher ~]$ git clone --recursive git@github.com:kvgibsher/blog.git
Клонирование в «blog»...
remote: Enumerating objects: 103, done.
remote: Counting objects: 100% (103/103), done.
remote: Compressing objects: 100% (91/91), done.
Получение объектов: 41% (43/103), 1012.00 КиБ | 1.96 МиБ/с
```

Рис. 4.4: Клонирование репозитория

Проверяем успешность загрузки файлов с github с помощью команды ls (рис. [4.5]).

```
kvgibsher@kvgibsher:~/blog
[kvgibsher@kvgibsher ~]$ git clone --recursive git@github.com:kvgibsher/blog.git
Клонирование в «blog»...
remote: Enumerating objects: 103, done.
remote: Counting objects: 100% (103/103), done.
remote: Compressing objects: 100% (91/91), done.
remote: Total 103 (delta 3), reused 80 (delta 0), pack-reused 0
Получение объектов: 100% (103/103), 5.88 МиБ | 1.82 МиБ/с, готово.
Определение изменений: 100% (3/3), готово.
[kvgibsher@kvgibsher ~]$ cd /blog
bash: cd: /blog: Нет такого файла или каталога
[kvgibsher@kvgibsher ~]$ cd blog/
[kvgibsher@kvgibsher blog]$ ls
academic.Rproj  content  images  preview.png  theme.toml
assets          data    LICENSE.md  README.md
config          go.mod  netlify.toml  static
[kvgibsher@kvgibsher blog]$ ~bin/hugo
```

Рис. 4.5: Проверка файлов

Запускаем исполняемый файл hugo ранее перемещенный в созданную нами

в домашнем каталоге пользователя папку bin. (рис. [4.6]).

```
kygibsher@kygibsher blog1: ~/bin/hugo
hugo: downloading modules -
hugo: collected modules in 14399 ms
start building sites _
hugo v0.110.0-e32a493b7826d0763c3b79623952e025402b166+extended linux/amd64 BuildDate=2023-01-17T12:16:09Z VendorInfo=gohugoio

-----| EN
Pages | 55
Paginator pages | 0
Non-page files | 16
Static files | 9
Processed images | 51
Aliases | 15
Sitemaps | 1
Cleaned | 0

otal in 19013 ms
kygibsher@kygibsher blog1: ~
```

Рис. 4.6: Запуск hugo

Удаляем каталог public находящийся в blog ввиду его ненужности. (рис. [4.7]).

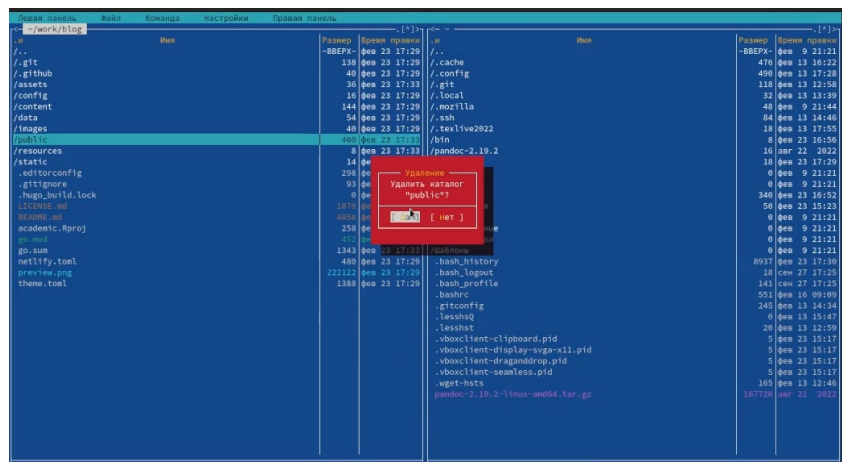


Рис. 4.7: Удаление каталога public

Находясь в каталоге blog прописываем команду hugo server , которая созда-  
ет локальный хост на сайт и выдает ссылку рабочую лишь в той ОС и на там  
устройстве, в котором произошла ее генерация (рис. [4.8]).



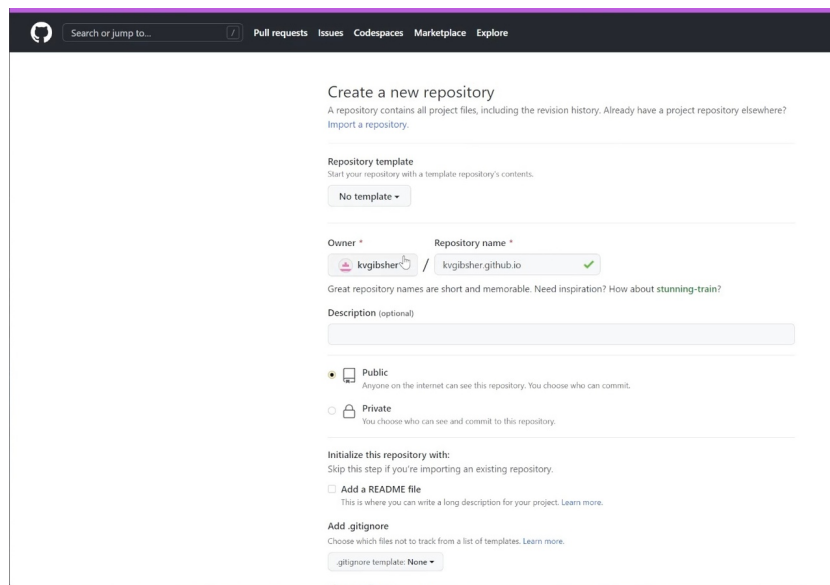


Рис. 4.10: Создание репозитория github.io

Клонируем только что созданный нами репозиторий в нашу ОС и с помощью команды `ls` проверяем, что все успешно получилось. Каталог склонировался, но пустой.(рис. [4.11]).

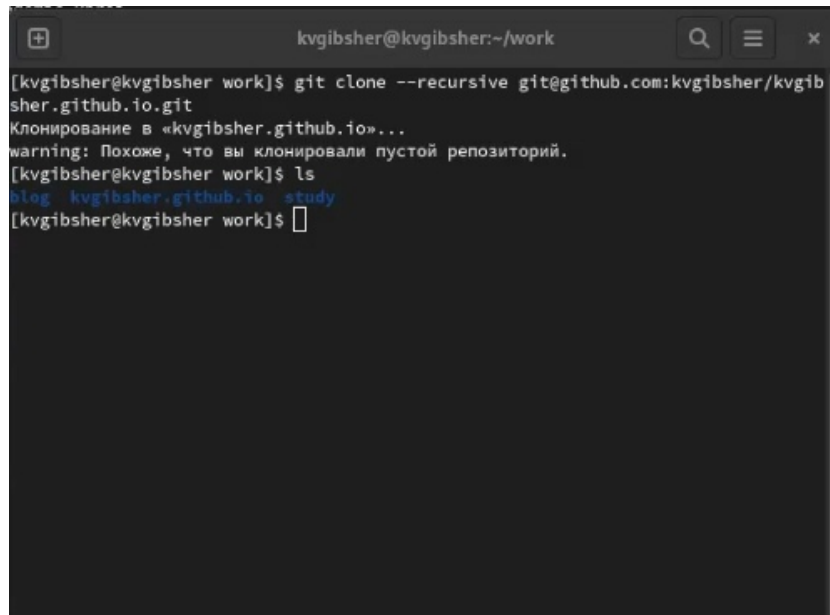
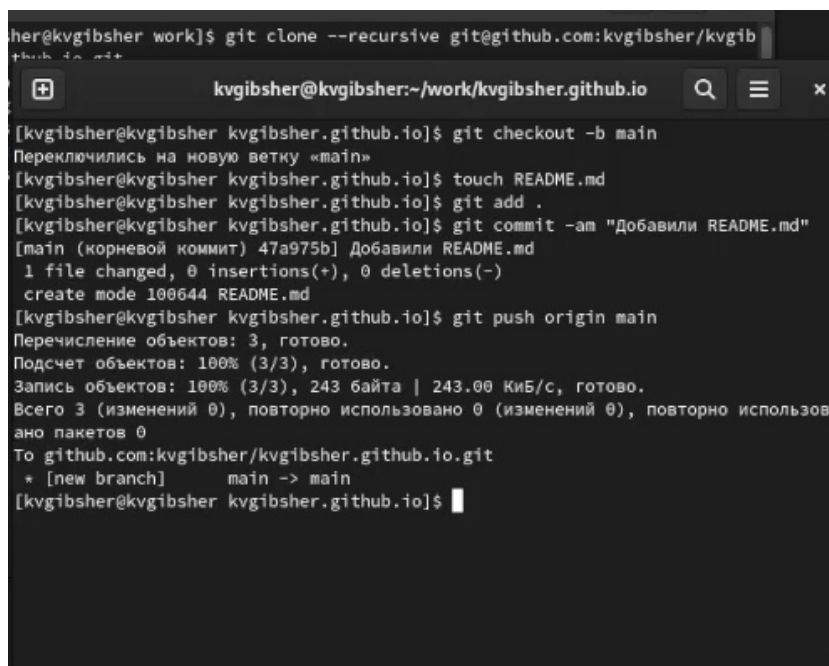


Рис. 4.11: Клонирование репозитория github.io

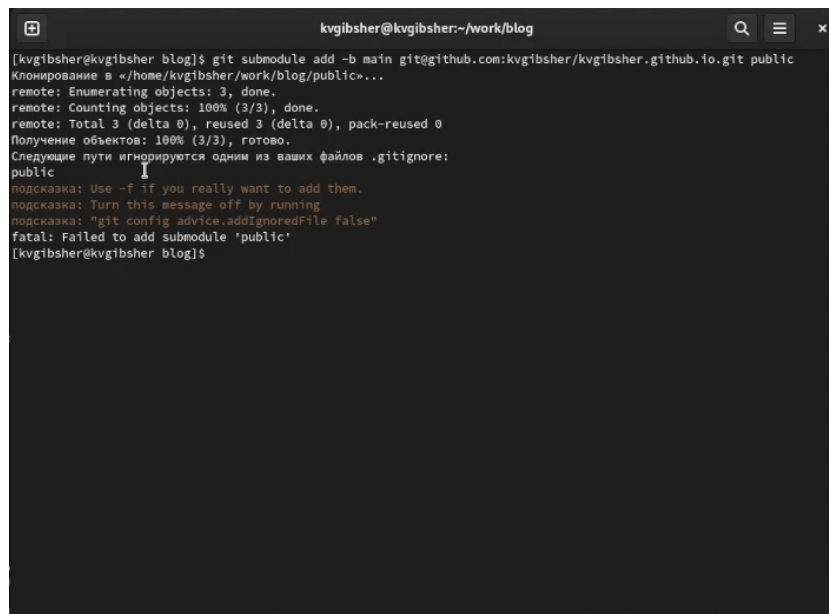
Создаем главную ветку в каталоге и называем ее main. Затем, чтобы активировать репозиторий создаем пустой файл README.md и прописываем git push origin main для выгрузки на github. (рис. [4.12]).



```
her@kvgibsher work]$ git clone --recursive git@github.com:kvgibsher/kvgibsher.github.io
kvgibsher@kvgibsher:~/work/kvgibsher.github.io$ git checkout -b main
Перешли на новую ветку «main»
kvgibsher@kvgibsher kvgibsher.github.io$ touch README.md
kvgibsher@kvgibsher kvgibsher.github.io$ git add .
kvgibsher@kvgibsher kvgibsher.github.io$ git commit -am "Добавили README.md"
[main (корневой коммит) 47a975b] Добавили README.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
kvgibsher@kvgibsher kvgibsher.github.io$ git push origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 243 байта | 243.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To github.com:kvgibsher/kvgibsher.github.io.git
 * [new branch]      main -> main
kvgibsher@kvgibsher kvgibsher.github.io$
```

Рис. 4.12: Активация репозитория

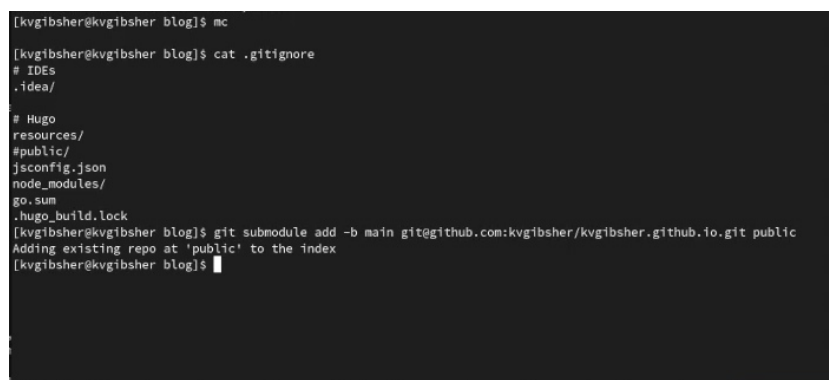
Прописываем git submodule add -b main находясь в каталоге blog. Но выполнение этой команды не очень успешно, т.к. некоторые пути игнорируются файлом gitignore. Потому попытка добавить submodule public проваливается. (рис. [4.13]).



```
kvgibsher@kvgibsher:~/work/blog
[kvgibsher@kvgibsher blog]$ git submodule add -b main git@github.com:kvgibsher/kvgibsher.github.io.git public
Клонирование в ~/home/kvgibsher/work/blog/public...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Получение объектов: 100% (3/3), готово.
Следующие пути игнорируются одним из ваших файлов .gitignore:
public
подсказка: Use -f if you really want to add them.
подсказка: Turn this message off by running
подсказка: "git config advice.addIgnoredFile false"
fatal: Failed to add submodule 'public'
[kvgibsher@kvgibsher blog]$
```

Рис. 4.13: Git add submodule public

С помощью MightCommander переходим в файл gitignore и коммитим с помощью знака # необходимую строку. Затем повторяем команду из прошлого пункта и на этот раз все получается и автоматически у нас создается каталог public в blog. (рис. [4.14]).



```
[kvgibsher@kvgibsher blog]$ mc
[kvgibsher@kvgibsher blog]$ cat .gitignore
# IDEs
.idea/

# Hugo
resources/
#public/
jsconfig.json
node_modules/
go.sum
.hugo_build.lock
[kvgibsher@kvgibsher blog]$ git submodule add -b main git@github.com:kvgibsher/kvgibsher.github.io.git public
Adding existing repo at 'public' to the index
[kvgibsher@kvgibsher blog]$
```

Рис. 4.14: Исправление файла gitignore и повтор команды

Запускаем вновь исполняемый файл hugo в каталоге blog и все содержимое автоматически выгружается в каталог public. (рис. [4.15]).

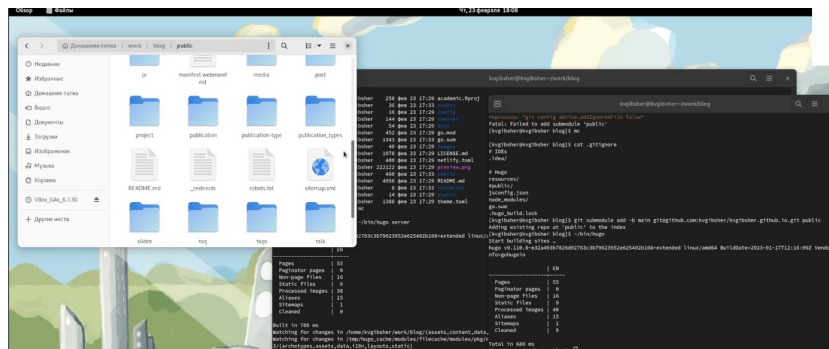


Рис. 4.15: Запуск файла hugo в blog

С помощью команды `git remote -v` проверяем точно ли необходимый для сайта репозиторий подключен к нашей ОС. Затем с помощью уже знакомых команд добавляем сайт и все сопутствующие файлы на гитхаб. (рис. [4.16]).

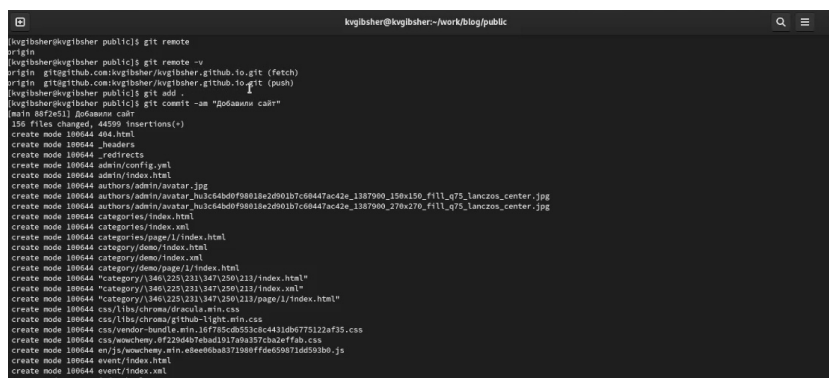


Рис. 4.16: Выгрузка сайта на github

Заходим на гитхаб и убеждаемся в том, что все файлы сайта успешно загружены. (рис. [4.17]).



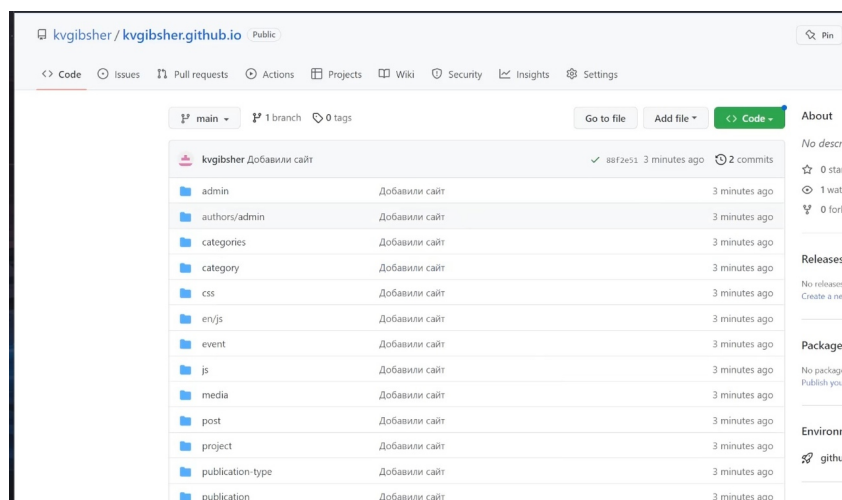


Рис. 4.17: Проверка загрузки

Копируем часть ссылки на одноименный репозиторий и проверяем работоспособность нашего личного сайта. (рис. [4.18]).

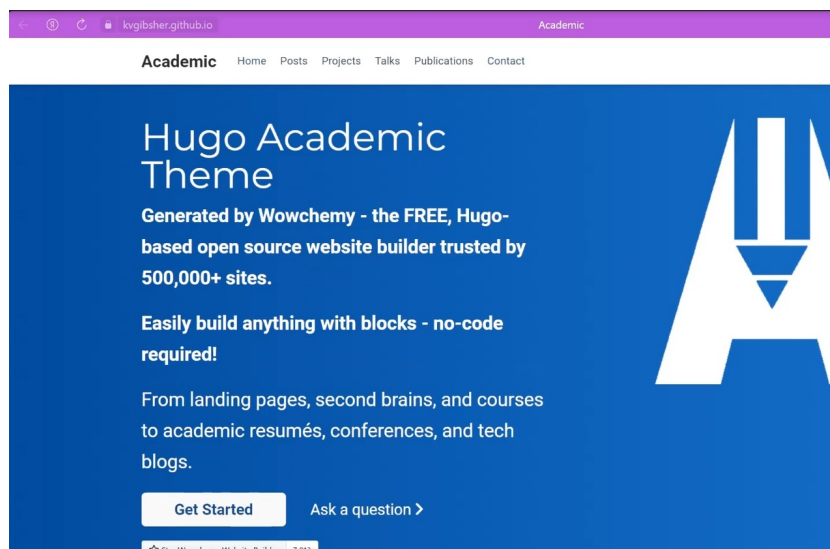


Рис. 4.18: Проверка работоспособности сайта

## 5 Выводы

Мы установили генератор статичных сайтов hugo , создали необходимые для сайта репозитории , связали их с нашей ОС и проверили работоспособность первоначальной версии нашего сайта.

## Список литературы

1. Первый этап индивидуального проекта by Кулябов Д.С. [Электронный ресурс] - Режим доступа:<https://www.youtube.com/watch?v=OpsSv0RE3C4>