

Отчет по выполнению первого этапа индивидуального проекта

Markdown

Агаджанян Артур вячеславович НКА6д-01-23

Цель работы:

Размещение на Github pages заготовки для персонального сайта.

Теоретическое введение

1. Установка программного обеспечения для Hugo

Ubuntu:

```
-sudo apt-get install go hugo
```

Fedora:

```
-sudo dnf install go hugo
```

Windows:

```
- choco install hugo extended
```

Ручная Установка:

Если устанавливаемая в системе версия hugo меньше необходимой, тогда следует установить программу вручную.

Версию Hugo можно посмотреть при помощи команды:

```
- hugo version
```

Следует скачать архив с репозитория:

```
-https://github.com/gohugoio/hugo/releases
```

Основные этапы выполнения работы

1. скачиваем Hugo на компьютер



рис.(1) Скачиваем нужную версию Hugo с github(а именно hugo_extended_0.123.7_Linux-64bit.tar.gz)

2. Создаем папку для корректного использования Hugo

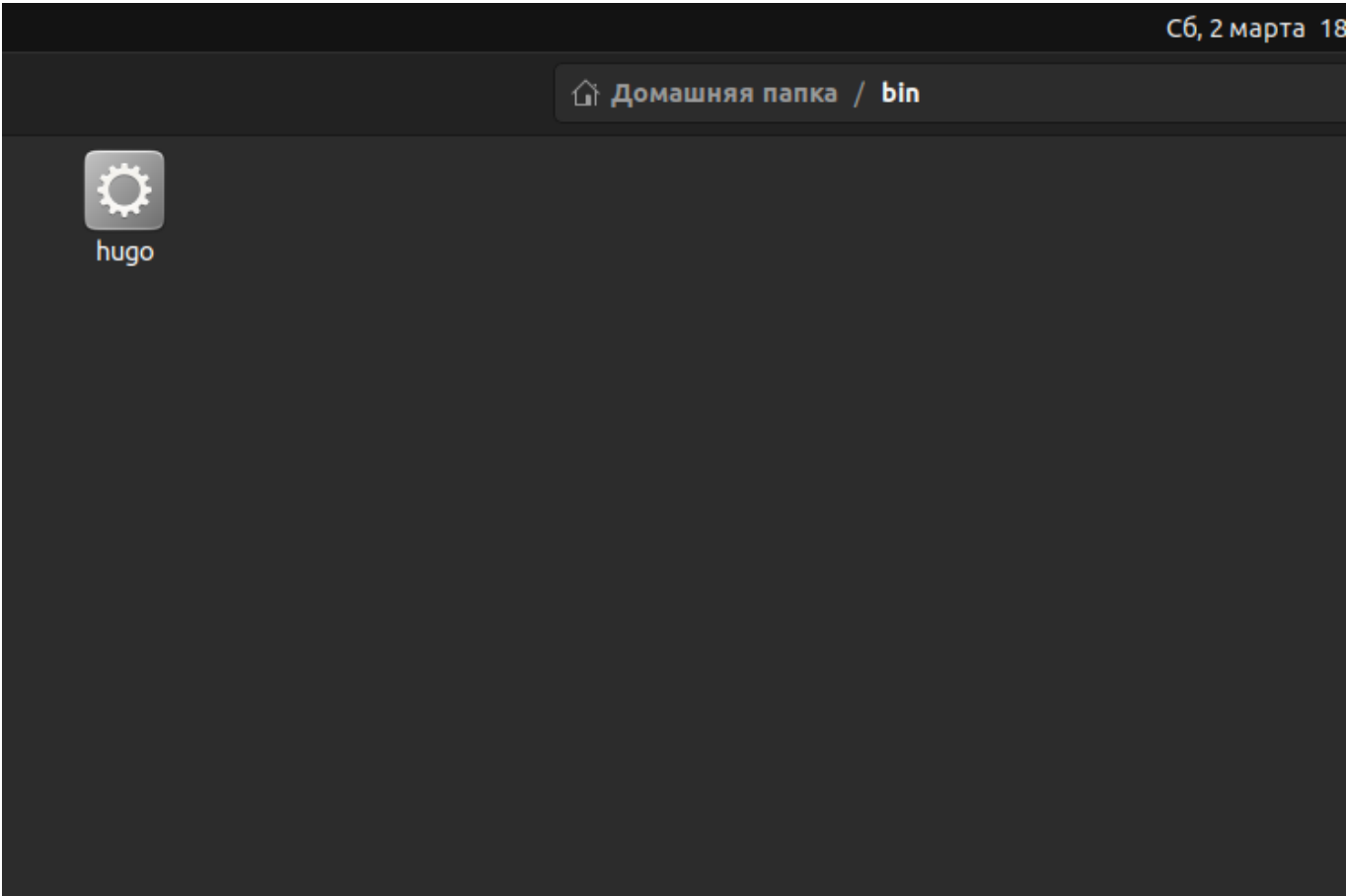
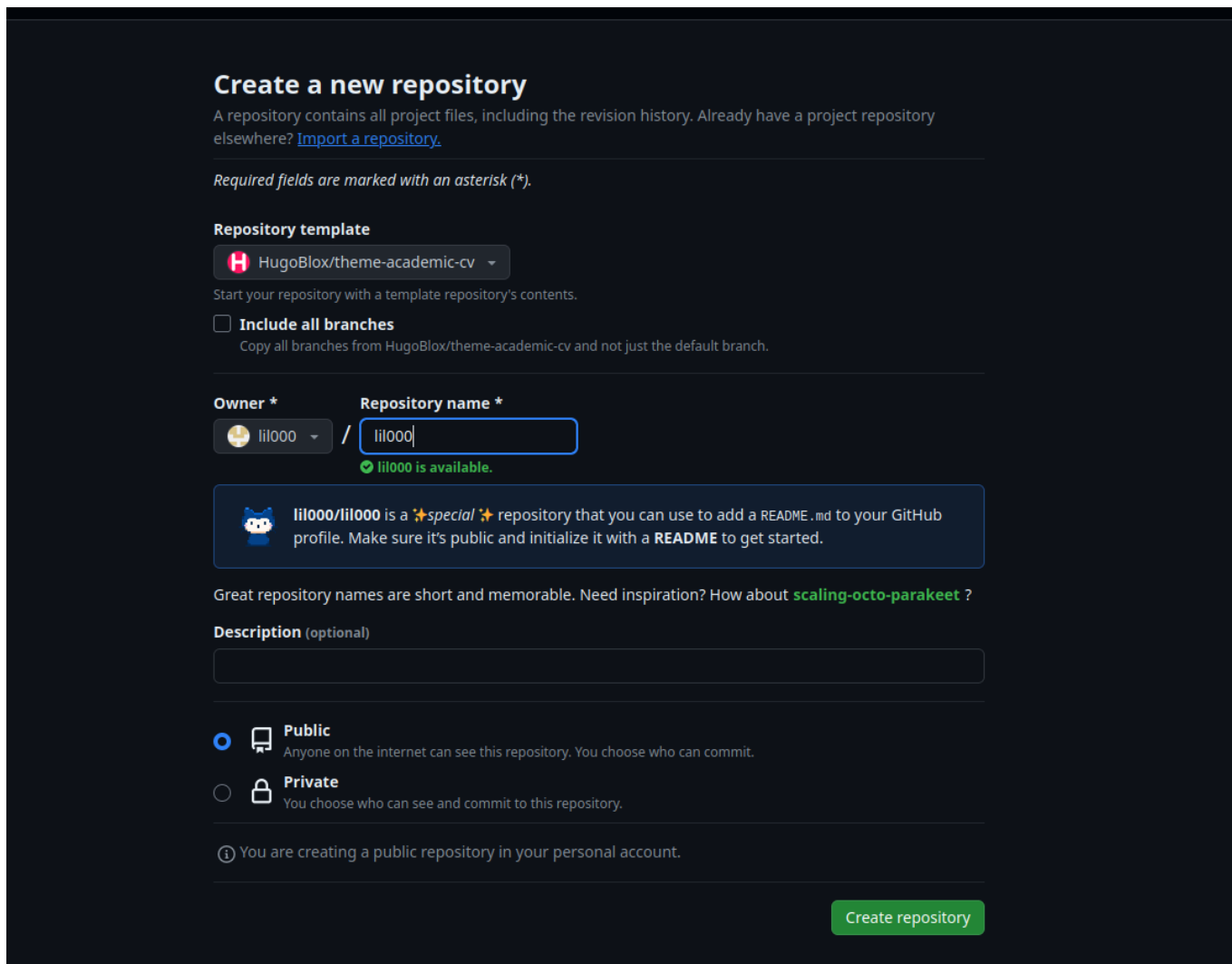


рис.(2)называем папку **bin**

3. С помощью шаблона создаем свой репозиторий на основе его же




Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*


Repository template

 HugoBlox/theme-academic-cv ▾


Start your repository with a template repository's contents.

☐ **Include all branches**
Copy all branches from HugoBlox/theme-academic-cv and not just the default branch.

Owner * **Repository name ***


 lil000 ▾ /


✔ lil000 is available.

 lil000/lil000 is a ✨special ✨ repository that you can use to add a README.md to your GitHub profile. Make sure it's public and initialize it with a **README** to get started.

Great repository names are short and memorable. Need inspiration? How about [scaling-octo-parakeet](#) ?

Description (optional)

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

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

① You are creating a public repository in your personal account.

[Create repository](#)

рис.(3) Создание репозитория

4. Клонировем репозиторий в локальную папку **work**



```
Product-Name:~$ cd work
Product-Name:~/work$ git clone --recursive git@github.com:lil000/blog.git
Cloning into 'blog'...
remote: Enumerating objects: 103, done.
remote: Compressing objects: 100% (103/103), done.
remote: Total 103 (delta 0), reused 83 (delta 2), pack-reused 0
Receiving objects: 100% (103/103), 6.07 МиБ | 6.24 МиБ/с, готово.
100% (3/3), готово.
Product-Name:~/work$
```

рис.(4)с помощью команды `git clone --recursive` клонируем репозиторий.

5. Устанавливаем в локальный репозиторий Hugo

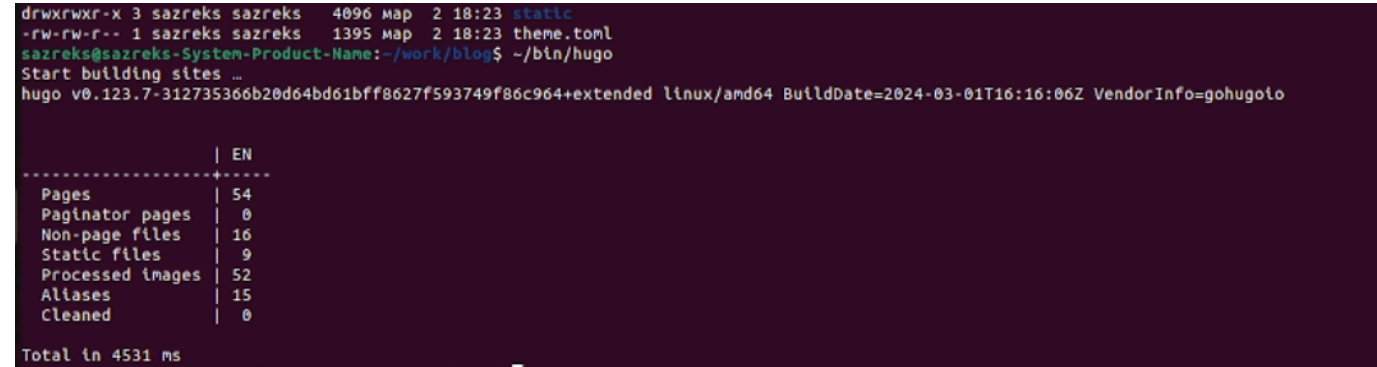


рис.(5) Установка Hugo

6. Создаем локальный сервер с помощью Hugo

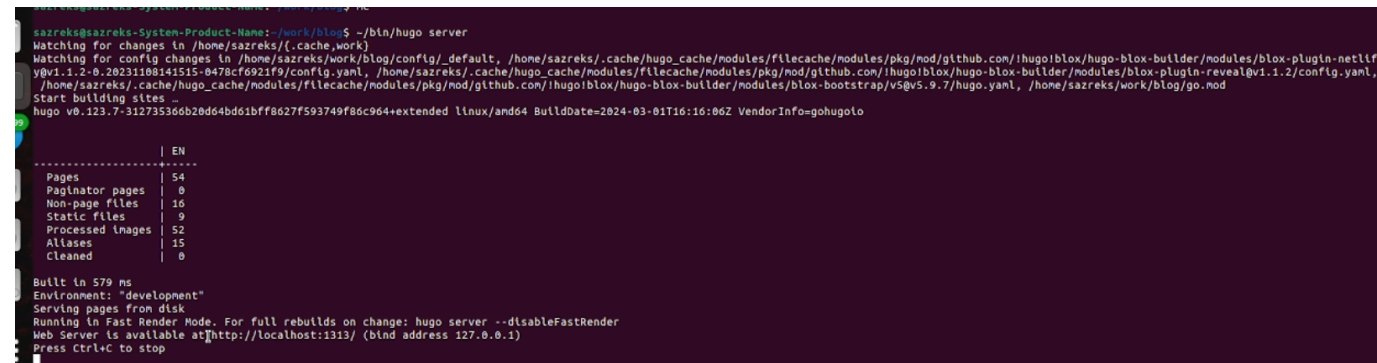


рис.(6) Создание локального сервера

7. Проверяем наш сайт через локальный домен

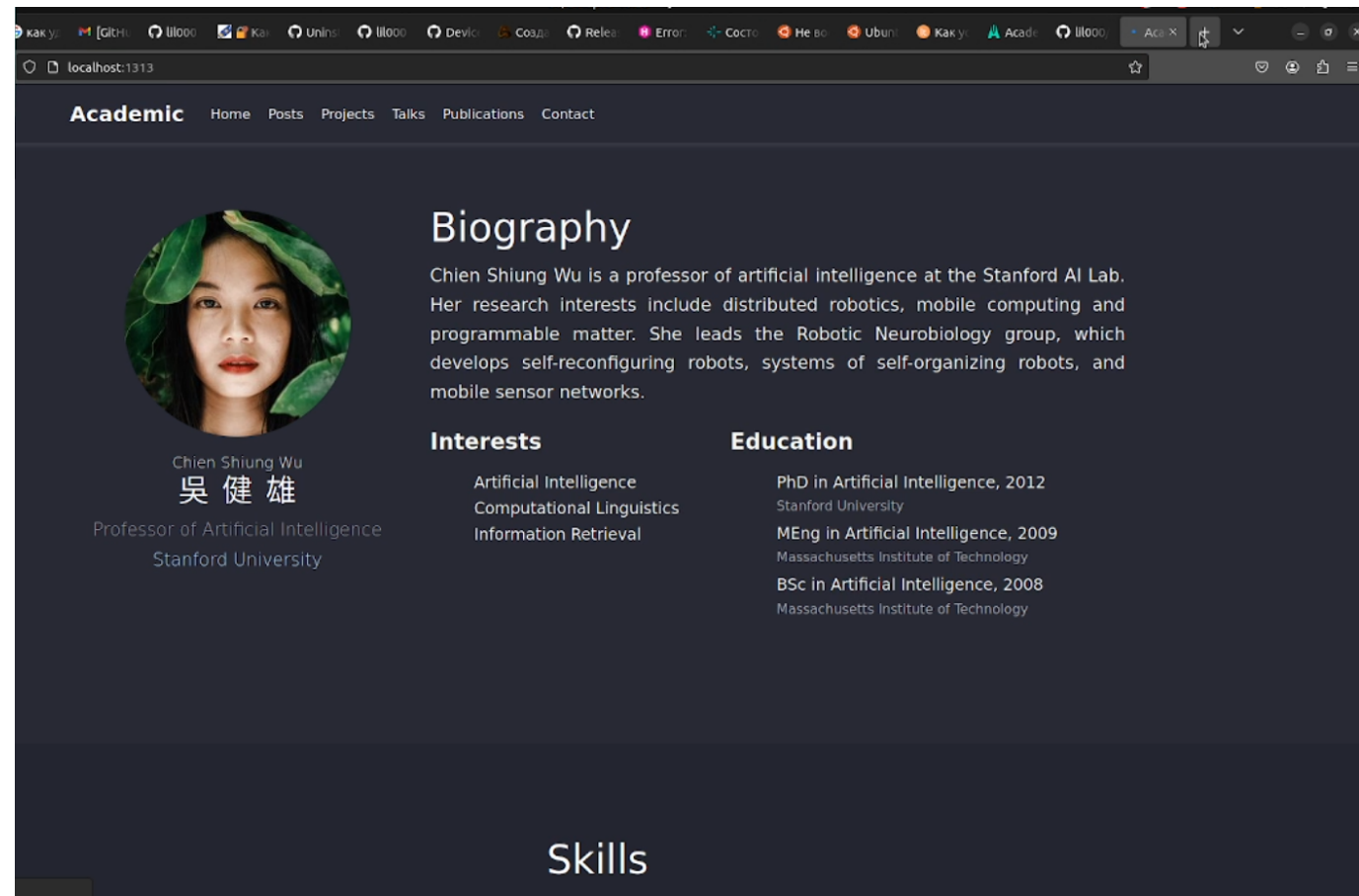
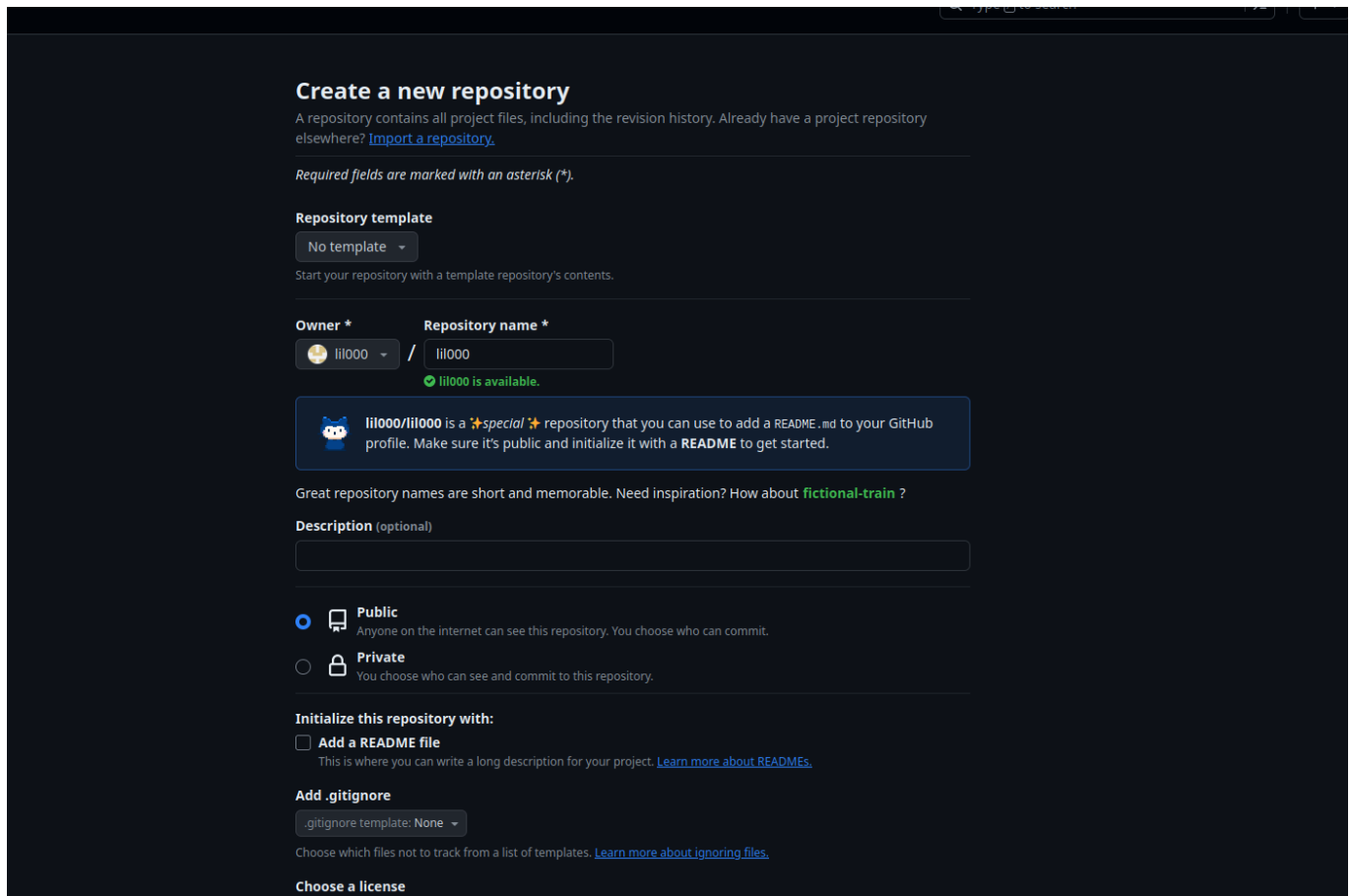


рис.(7) смотрим на наш сайт

8. Создаем новый репозиторий с помощью Github для Github pages



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Repository template

No template

Start your repository with a template repository's contents.

Owner * lil000 / **Repository name *** lil000

lil000 is available.

lil000/lil000 is a *special* repository that you can use to add a README.md to your GitHub profile. Make sure it's public and initialize it with a README to get started.

Great repository names are short and memorable. Need inspiration? How about **fictional-train** ?

Description (optional)

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

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

Initialize this repository with:

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

рис.(8)Создаем новый репозиторий

9. Клонировем новый репозиторий локально в папку **work**

```
того 8
fmxgwxr-x 12 sazeaks sazeaks 4096 map 2 18:25 blog
fmxgwxr-x 1 sazeaks sazeaks 0 map 2 14:03 PISYAPOPA.md
fmxgwxr-x 4 sazeaks sazeaks 4096 map 2 16:08 study
sazeaks@sazeaks-System-Product-Name:~/work$ git clone --recursive git@github.com:lil000/lil000.github.io.git
клонирование в «lil000.github.io»...
warning: Похоже, что вы клонировали пустой репозиторий.
sazeaks@sazeaks-System-Product-Name:~/work$
```

рис.(9) Клонировем новый репозиторий

10. Создаем ветку Main и в ней же файл README.md

```
подсчет объектов: 100% (3/3), готово.
запись объектов: 100% (3/3), 899 байтов | 899.00 КиБ/с, готово.
сего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
o github.com:lil000/lil000.github.io.git
* [new branch] main -> main
sazeaks@sazeaks-System-Product-Name:~/work/lil000.github.io$ cd blog/
bash: cd: blog/: Нет такого файла или каталога
sazeaks@sazeaks-System-Product-Name:~/work/lil000.github.io$ cd blog
bash: cd: blog: Нет такого файла или каталога
sazeaks@sazeaks-System-Product-Name:~/work/lil000.github.io$ cd blog/
bash: cd: blog/: Нет такого файла или каталога
sazeaks@sazeaks-System-Product-Name:~/work/lil000.github.io$ cd ..
sazeaks@sazeaks-System-Product-Name:~/work$ cd blog
sazeaks@sazeaks-System-Product-Name:~/work/blog$ pwd
/home/sazeaks/work/blog
sazeaks@sazeaks-System-Product-Name:~/work/blog$ git submodule add -b main git@github.com:lil000/lil000.github.io.git public
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"
sazeaks@sazeaks-System-Product-Name:~/work/blog$
```

рис.(10)Создаем ветку и файл README.md

11. Подключаем новосозданный репозиторий к папке **public** внутри папки **work/blog**

```

sazreks@fazreks-System-Product-Name:~/work$ cd lil000.github.io
sazreks@fazreks-System-Product-Name:~/work/lil000.github.io$ git checkout -b main
Переключились на новую ветку «main»
sazreks@fazreks-System-Product-Name:~/work/lil000.github.io$ touch README.md
sazreks@fazreks-System-Product-Name:~/work/lil000.github.io$ git add .
sazreks@fazreks-System-Product-Name:~/work/lil000.github.io$ git commit -am "Добавили README.md"
[main (корневой коммит) fa01a07] Добавили README.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
sazreks@fazreks-System-Product-Name:~/work/lil000.github.io$ git push origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 899 байтов | 899.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To github.com:lil000/lil000.github.io.git
 * [new branch]      main -> main

```

рис.(11)Подключаем репозиторий к папке **public**

12. Отключаем игнорирование public в gitignore

Используем команду `mc` для нужного результата

```

подсказка: 'git config advice.addignorefile false'
sazreks@fazreks-System-Product-Name:~/work/blog$ mc

sazreks@fazreks-System-Product-Name:~/work/blog$ cat .gitignore
# IDEs
.idea/

# Hugo
resources/
#public/
jsconfig.json
node_modules/
go.sum
.hugo_build.lock
sazreks@fazreks-System-Product-Name:~/work/blog$

```

рис.(12)Отключили игнорирование,закомментив public - #public

13. Повторяем действия из пункта 11

Подключаем последний созданный репозиторий к папке **public** внутри папки **work/blog**

```

sazreks@fazreks-System-Product-Name:~/work/blog$ git submodule add -b main git@github.com:lil000/lil000.github.io.git public
Клонирование в «/home/sazreks/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.

```

рис.(13)Подключаем репозиторий к папке **public**

14. Используем Hugo для нового репозитория

	EN
Pages	54
Paginator pages	0
Non-page files	16
Static files	9
Processed images	52
Aliases	15
Cleaned	0

рис.(14)Использовали Hugo на новом репозитории

15. Проверяем подключение каталога к репозиторию и добавляем комментарий к коммиту

```
lil000@lil000-System-Product-Name:~/work/blog$ cd public
lil000@lil000-System-Product-Name:~/work/blog/public$ git remote -v
origin  git@github.com:lil000/lil000.github.io.git (fetch)
origin  git@github.com:lil000/lil000.github.io.git (push)
lil000@lil000-System-Product-Name:~/work/blog/public$ git add .
lil000@lil000-System-Product-Name:~/work/blog/public$ git commit -am "Добавили сайт"
```

рис.(15)Проверяем подключение командой **git remote -v** и коммитим с комментарием

16. Проверим наш репозиторий

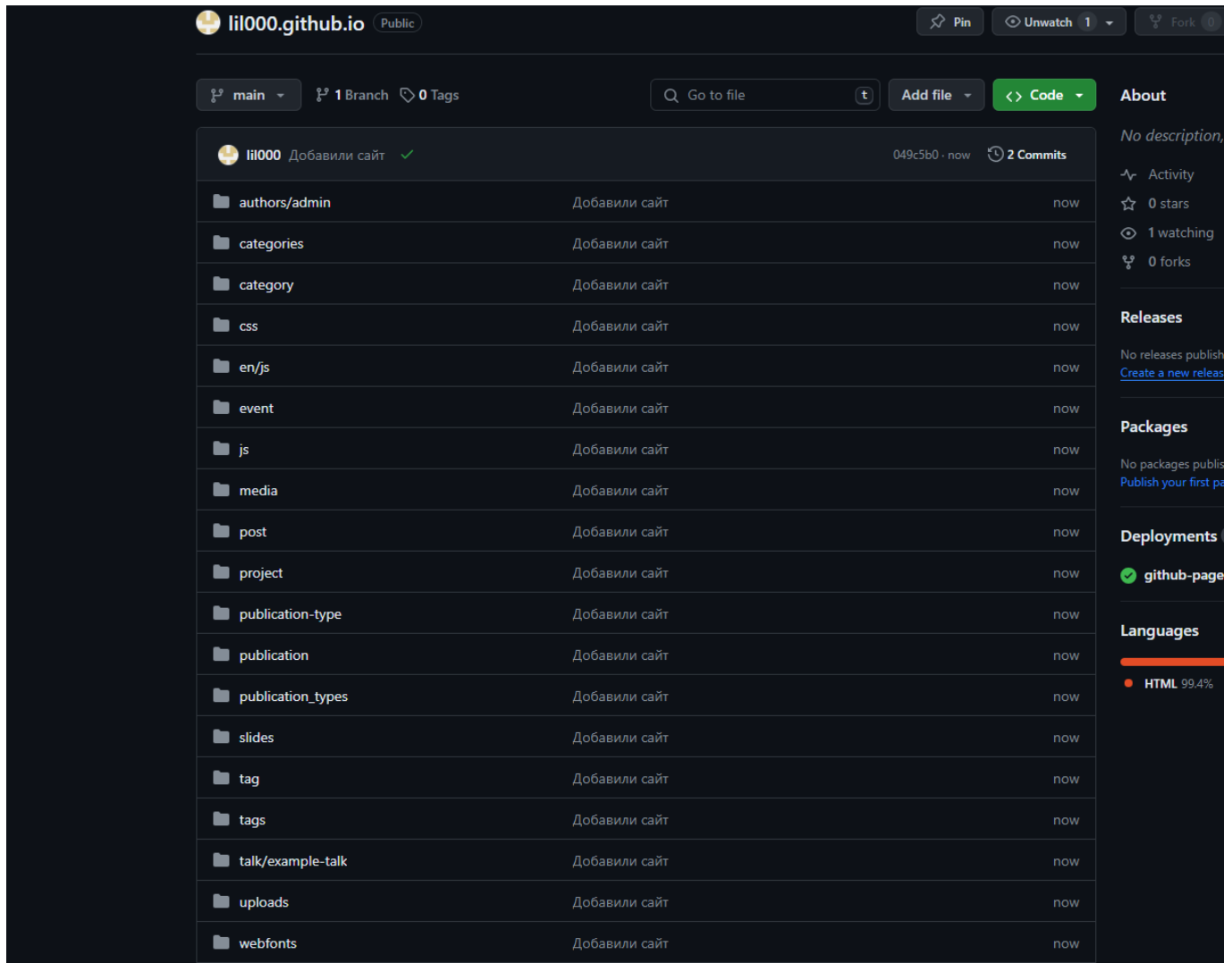


рис.(16)Проверяем все ли загрузилось в репозиторий на github

17. Смотрим на наш сайт через ссылку (<https://lil000.github.io>)

по этой ссылке может пройти кто угодно

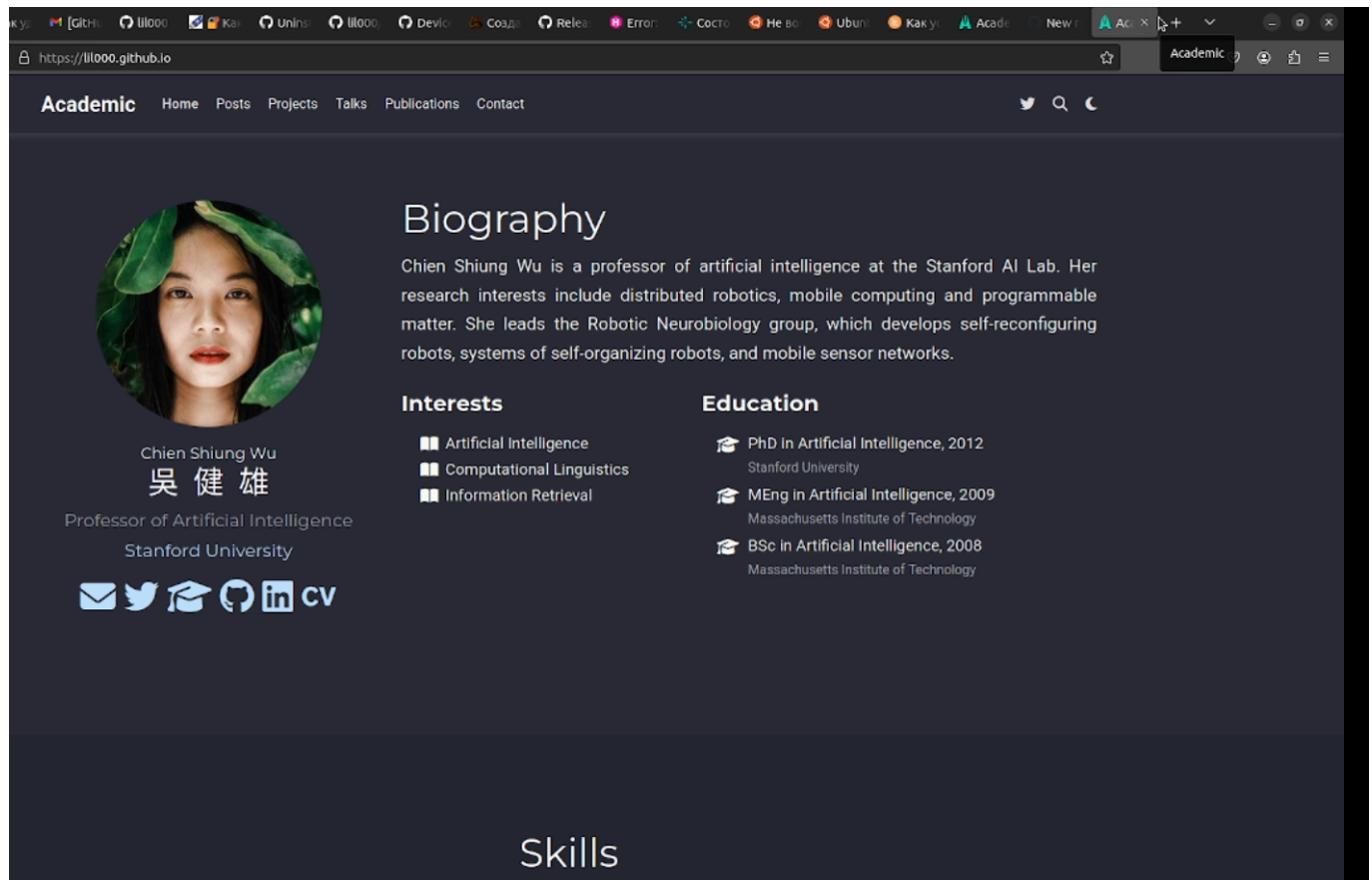


рис.(17) Все корректно

Вывод:

У нас получилось создать сайт с помощью **Hugo** и команд в консоле **Linux** для **Github**, вместе с этим мы разместили заготовку сайта.