

# **Отчет по персональному проекту версии 1**

Генералов Даниил, НПИ-01-21, 1032212280

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Задание</b>	<b>6</b>
<b>3 Теоретическое введение</b>	<b>7</b>
<b>4 Выполнение лабораторной работы</b>	<b>8</b>
<b>5 Выводы</b>	<b>16</b>

# Список иллюстраций

4.1	Создание репозитория в GitHub . . . . .	8
4.2	Создание шаблона сайта . . . . .	9
4.3	Форум Wowchemy . . . . .	9
4.4	Компиляция шаблона сайта . . . . .	10
4.5	Конфигурация директории сборки . . . . .	10
4.6	Коммит с содержимым сайта . . . . .	11
4.7	Сайт на GitHub Pages без форматирования . . . . .	11
4.8	Ошибка при загрузке CSS . . . . .	12
4.9	Страница создания организации . . . . .	12
4.10	Доменные записи A . . . . .	13
4.11	Ошибка сертификата для домена . . . . .	13
4.12	Создание доменной записи CNAME . . . . .	14
4.13	Создание сертификата . . . . .	14
4.14	Теперь сайт работает . . . . .	15

# **Список таблиц**

# **1 Цель работы**

Целью данной работы является установка генератора статических сайтов Hugo и развертка шаблона сайта на GitHub Pages.

## **2 Задание**

Для выполнения работы требуется установить программу Hugo, создать репозиторий для сайта на GitHub Pages, затем провести развертку шаблона сайта на GitHub Pages.

## 3 Теоретическое введение

Один из удобных способов создания статических сайтов (то есть тех, которые в основном состоят из неизменного контента) – это применить генератор статических сайтов. Это программа, которая принимает на вход документы, написанные на промежуточном языке разметки (например Markdown), и выдают набор HTML-страниц, которые затем можно разместить на хостинг.

Один из таких хостингов – GitHub Pages. С помощью инфраструктуры CI можно сделать так, чтобы содержимое вашего репозитория публиковалось в интернете на определенном доменном имени – без использования своей инфраструктуры и не требуя никакой настройки серверов.

# 4 Выполнение лабораторной работы

Первым делом нужно создать новый репозиторий, внутри которого будут храниться данные сайта. Это можно сделать из командной строки или с помощью веб-интерфейса на рис. 4.1.

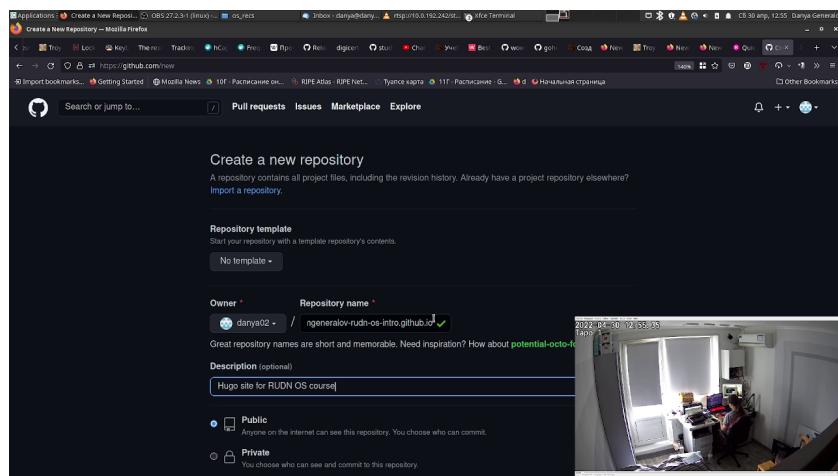


Рис. 4.1: Создание репозитория в GitHub

Дальше я добавил этот репозиторий в основной репозиторий как подмодуль и использовал команду `hugo new site`, чтобы создать шаблон сайта, как показано на рис. 4.2.

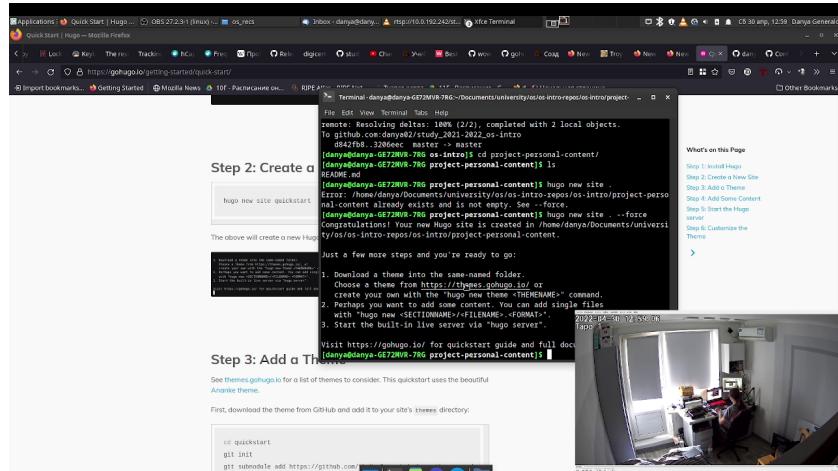


Рис. 4.2: Создание шаблона сайта

После этого я начал искать, как установить тему Academic, и посредством этого поиска узнал, что она теперь называется Wowchemy, и метод установки изменился. Я узнал это на форуме Wowchemy на GitHub, который показан на рис. 4.3.

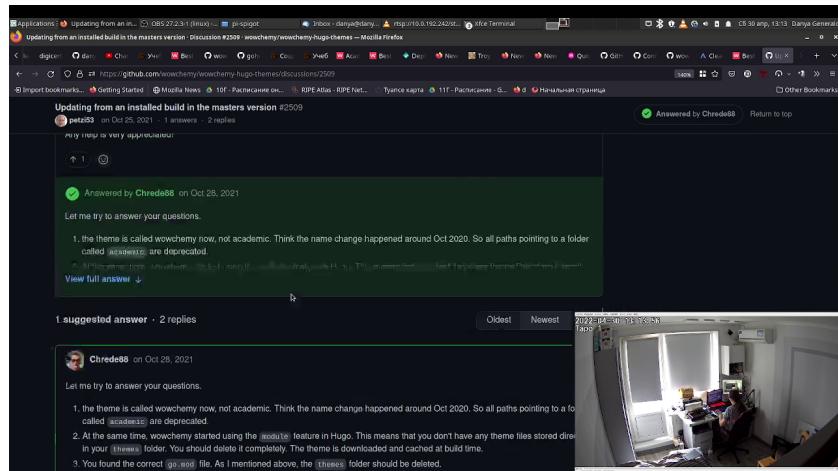


Рис. 4.3: Форум Wowchemy

Как оказалось, нужно включить его как зависимость в определенном файле в корне репозитория, и после этого шаблон темы был успешно собран, как показано на рис. 4.4.

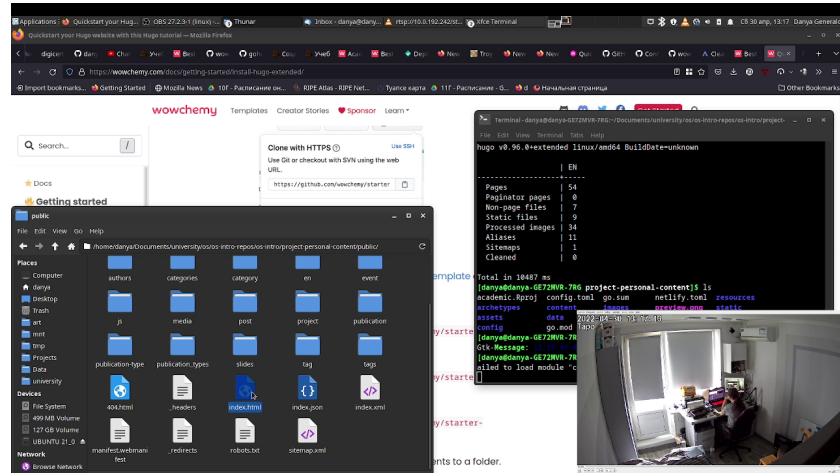


Рис. 4.4: Компиляция шаблона сайта

После этого была задача опубликовать это на GitHub Pages. Есть две опции, где хранятся файлы, подлежащие публикации – в корне репозитория или в папке `docs`. По умолчанию Hugo собирает сайт в папку `public`, поэтому нужно добавить ключ конфигурации, меняющий это поведение – он вносится в файл `config.toml` в корне репозитория, как показано на рис. 4.5.

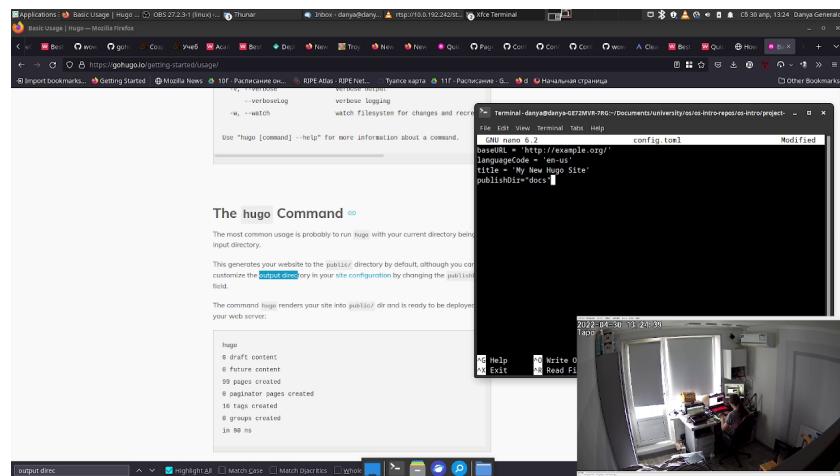


Рис. 4.5: Конфигурация директории сборки

После этого можно сделать коммит с новым сайтом (рис. 4.6) и посмотреть на него на GitHub Pages (рис. 4.7). Но сразу заметно, что что-то не так.

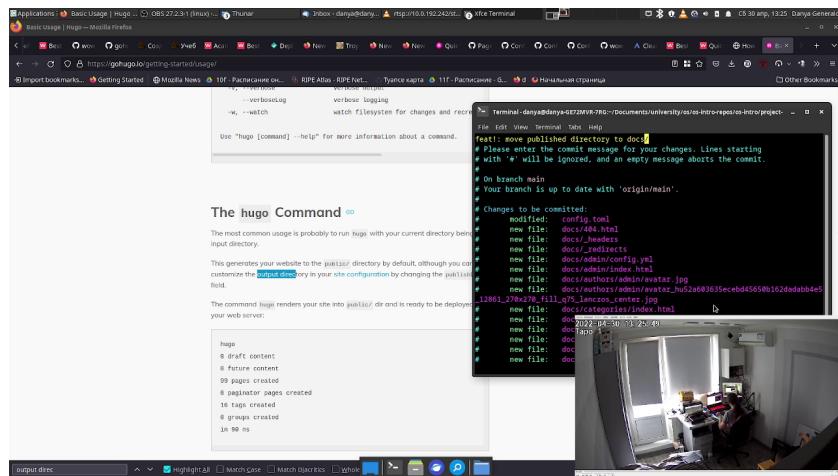


Рис. 4.6: Коммит с содержимым сайта

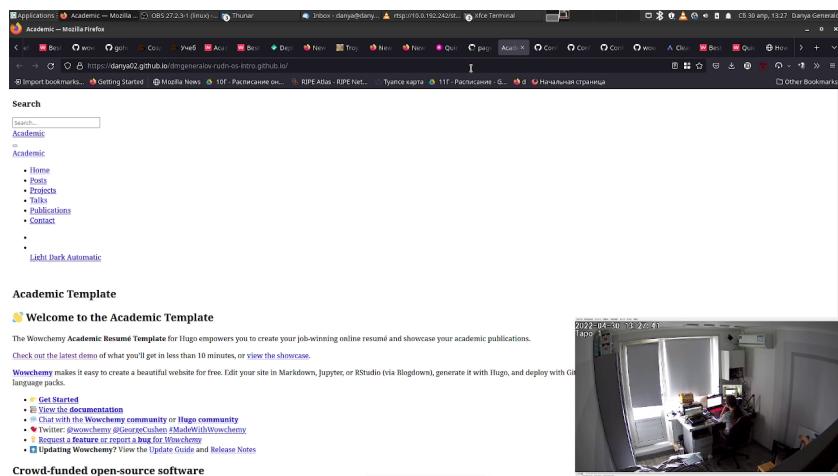


Рис. 4.7: Сайт на GitHub Pages без форматирования

Как нетрудно увидеть, на странице полностью нет форматирования. Если посмотреть в консоли запросов на рис. 4.8, то можно увидеть проблему – все ресурсы сайта выходят с ошибкой 404, хотя в папке сборки присутствуют. Дело в том, что пути к ресурсам начинаются с корня сайта, а GitHub Pages публикует проект в субдиректории, что ломает адресацию.

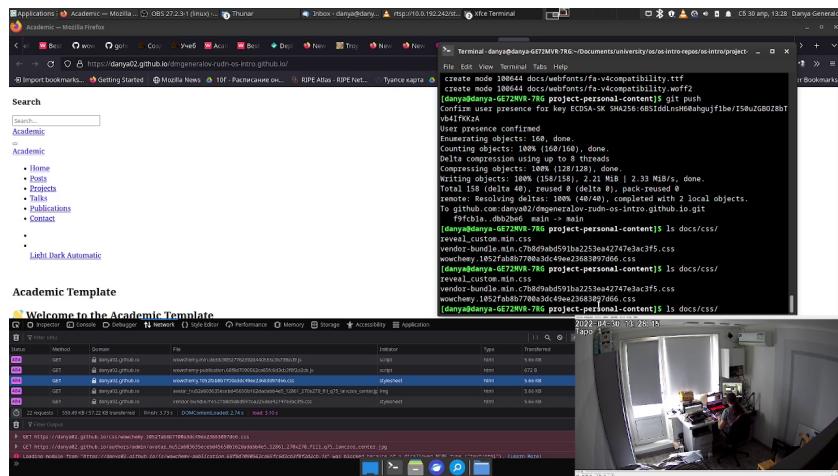


Рис. 4.8: Ошибка при загрузке CSS

Причина того, что GitHub Pages делает так, заключается в том, что страницы проектов являются подстраницами пользователей или организаций. Я не хотел использовать свою страницу пользователя только для этого проекта, поэтому я вместо этого подумал создать организацию. К сожалению, страница “создать организацию” делает впечатление, будто для того, чтобы это сделать, необходимо заплатить – когда я делал работу, я полностью пропустил бесплатную опцию на странице на рис. 4.9.

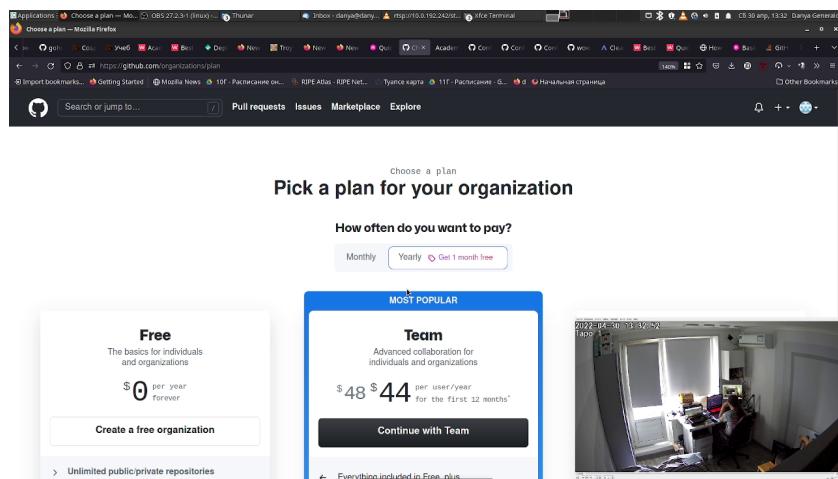


Рис. 4.9: Страница создания организации

Вместо этого я решил пойти другим путём. У меня есть доменное имя, danya02.ru, и я могу направить один из поддоменов на GitHub Pages. Первый

шаг для этого – это прописать DNS-записи в моей записи, чтобы этот поддомен оказался перенаправленным на GitHub Pages. Я добавил для начала A-записи, как показано на рис. 4.10.

The screenshot shows the Cloudflare DNS management interface for the domain `danya02.ru`. On the left sidebar, under the `DNS` section, there are several A records listed:

Type	Name	Content	Proxy status	TTL	Actions
A	dmngenerator.rudn-os-intro	185.199.10.153	Proxied	Auto	<a href="#">Edit</a>
A	dmngenerator.rudn-os-intro	185.199.11.153	Proxied	Auto	<a href="#">Edit</a>
A	dmngenerator.rudn-os-intro	185.199.109.153	Proxied	Auto	<a href="#">Edit</a>
A	dmngenerator.rudn-os-intro	185.199.108.153	Proxied	Auto	<a href="#">Edit</a>
A	*	95.165.8.38	DNS only	Auto	<a href="#">Edit</a>
A	danya02.ru	95.165.8.38	DNS only	Auto	<a href="#">Edit</a>
A	www	95.165.8.38	DNS only	Auto	<a href="#">Edit</a>
MX	danya02.ru	mail.danya02.ru	DNS only	Auto	<a href="#">Edit</a>
TXT	danya02.ru	v=spf1 mx ~all	DNS only	2022-04-30 13:42:37	<a href="#">Edit</a>
TXT	danya02.ru	google-site-verification=loICY...	DNS only	2022-04-30 13:42:37	<a href="#">Edit</a>
TXT	_dmarc	v=DMARC1; p=quarantine;ru...	DNS only	2022-04-30 13:42:37	<a href="#">Edit</a>
TXT	mail_domainkey	v=DKIM1; h=sha256; k=rsa; p...	DNS only	2022-04-30 13:42:37	<a href="#">Edit</a>

Below the table, there is a section titled "Cloudflare Nameservers" with the note: "To use Cloudflare, ensure your authoritative DNS servers, or nameservers have been changed".

Рис. 4.10: Доменные записи A

К сожалению, сделав это, я сделал невозможным для GitHub получить TLS-сертификат на этот домен, и поэтому при открытии этого поддомена появлялось окно, говорящее о неправильном сертификате (рис. 4.11).

The screenshot shows a browser window displaying a warning message: "Warning: Potential Security Risk Ahead". The message states: "Refused to detect a potential security threat and did not continue to dmngenerator.rudn-os-intro.danya02.ru. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details." Below this, it says: "What can you do about it? The issue is most likely with the website, and there is nothing you can do to resolve it. You can notify the website's administrator about the problem." At the bottom, there are two buttons: "Go Back (Recommended)" and "Advanced...".

Below the warning message, there is a detailed error message: "Websites prove their identity via certificates, but Firefox does not trust this site because it uses a certificate that is not valid for the address dmngenerator.rudn-os-intro.danya02.ru. The certificate is only valid for the following names: \*.github.com, www.github.com, github.com, githubusercontent.com, \*.githubusercontent.com". It also mentions an "Error code: SSL\_ERROR\_BAD\_CERT\_DOMAIN".

Рис. 4.11: Ошибка сертификата для домена

Чтобы справиться с этим, как оказалось, нужно указывать не простые A-записи, а CNAME-записи, фактически осуществляющие перенаправление с одного домена

на другой. Я добавил CNAME-запись, указывающую на свой поддомен на GitHub Pages, как показано на рис. 4.12.

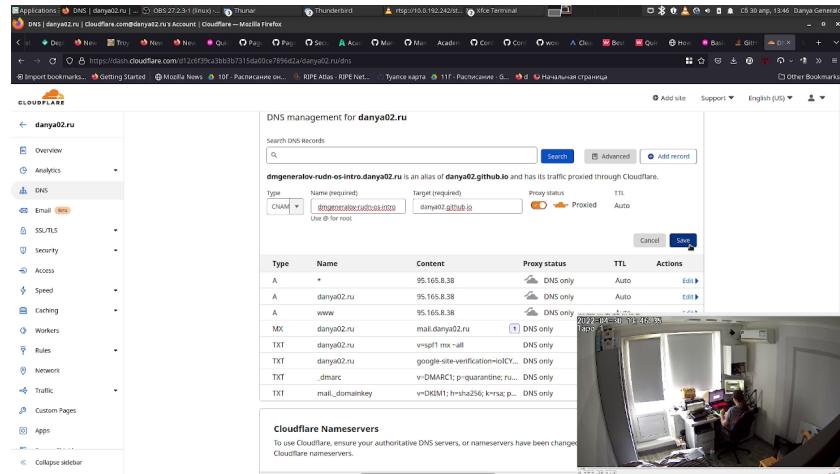


Рис. 4.12: Создание доменной записи CNAME

После этого интерфейс GitHub Pages подтвердил, что процесс создания сертификата начал (рис. 4.13), и после нескольких минут сайт стал открываться без предупреждений (рис. 4.14).

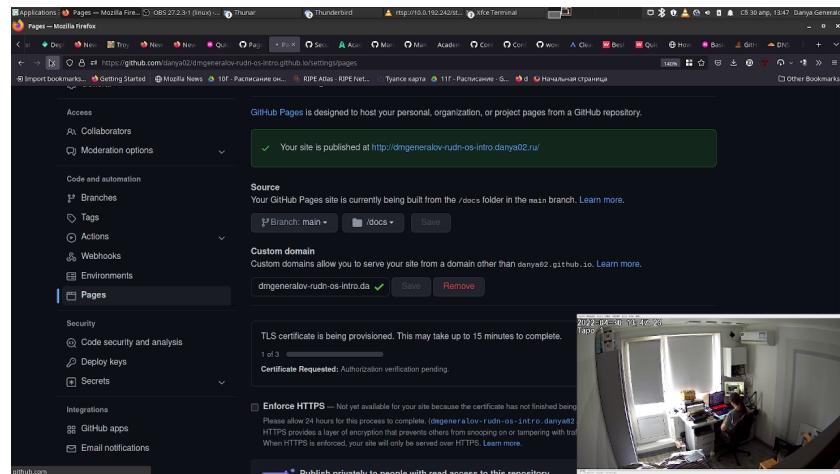


Рис. 4.13: Создание сертификата



Рис. 4.14: Теперь сайт работает

Текущее состояние сайта можно увидеть по адресу <https://dmgeneralov-rudn-os-intro.danya02.ru/>.

## **5 Выводы**

В этой работе мы успешно установили Hugo, сделали шаблон сайта, и опубликовали его на GitHub Pages. Используя эти инструменты, можно быстро создавать сайты без написания кода HTML или развертки собственных серверов. Поэтому такие сайты идеально подходят для проектов, которые не подразумевают высокой сложности пользовательского взаимодействия и в основном служат для публикации контента. В следующих шагах этого проекта мы сможем добавить контента к этому сайту.