

Отчет по 2 этапу индивидуального проекта

Добавление данных о себе

Лушин Артем Андреевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	10
	Список литературы	11

Список иллюстраций

2.1	Данные о себе	6
2.2	Добавление фотографии	7
2.3	Создание поста	7
2.4	Добавление текста	7
2.5	Создание поста	8
2.6	Добавление информации	8
2.7	Проверка на локальном сервере	8
2.8	Готовый онлайн сайт	9

Список таблиц

1 Цель работы

Добавить на сайт свою фотографию, данные о себе и выложить несколько постов.

2 Выполнение лабораторной работы

- 1) Я добавил в папку /home/aalushin/work/blog/content/authors/admin данные о себе, свои интересы и образование.



```
---
# Display name
title: Лушин Артём

# Full name (for SEO)
first_name: Артём
last_name: Лушин

# Status emoji
status:
icon: 🙋

# Is this the primary user of the site?
superuser: true

# Role/position/tagline
role: Ученик Российского университета дружбы народов

# Organizations/Affiliations to show in About widget
organizations:
- name: Российский университет дружбы народов
  url: https://www.rudn.ru/

# Short bio (displayed in user profile at end of posts)
bio: В сфере науки я предпочитаю изучать математику, программирование и нейронные сети.

# Interests to show in About widget
interests:
- Спорт
- Программирование
- Information Retrieval
```

Рис. 2.1: Данные о себе

- 2) Добавил свою фотографию на обложку сайта.

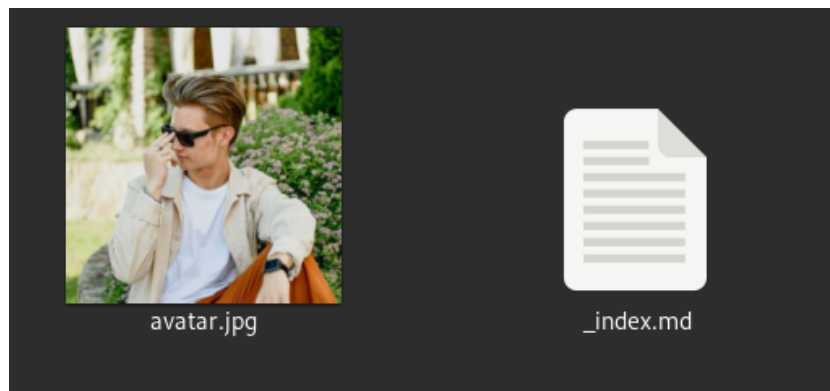


Рис. 2.2: Добавление фотографии

- 3) Я создал пост о прошедшей неделе. Затем заполнил файл и написал как прошла неделя.

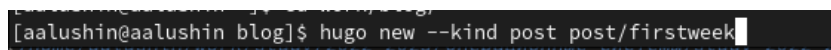


Рис. 2.3: Создание поста

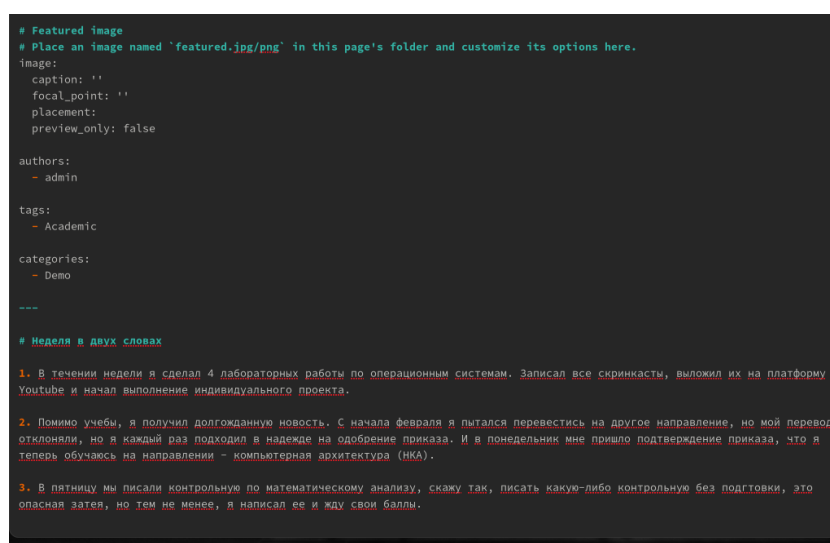


Рис. 2.4: Добавление текста

- 4) Я выбрал “Управление версиями. Git.” и создал пост по этой тема. Нашел в интернете информацию и заполнил файл.

```
[aalushin@aalushin blog]$ hugo new --kind post post/git
```

Рис. 2.5: Создание поста

История

Разработка ядра Linux велась на проприетарной системе BitKeeper, которую автор – Линус Торвалдс, сам разработчик Linux – предоставил проекту по бесплатной лицензии. Разработчики, высококлассные программисты, написали несколько утилит, и для одной Эндрю Триджелл произвёл реверс-инжиниринг формата передачи данных BitKeeper. В ответ Торвалдс обвинил разработчиков в нарушении соглашения и отозвал лицензию, и Торвалдс взялся за новую систему: ни одна из открытых систем не позволяла тысячам программистов кооперировать свои усилия (тот же конфликт привёл к написанию Mercurial). Идеология была проста: взять подход CVS и перевернуть с ног на голову, и заодно добавить надёжности.

Начальная разработка велась меньше, чем неделю: 3 апреля 2005 года разработка началась, и уже 7 апреля код Git управлялся неготовой системой. 16 июня Linux был переведён на Git, а 25 июля Торвалдс отказался от обязанностей ведущего разработчика.

Возможности

Система спроектирована как набор программ, специально разработанных с учётом их использования в сценариях. Это позволяет удобно создавать специализированные системы контроля версий на базе Git или пользовательские интерфейсы. Например, Cogito является именно таким примером оболочки к репозиториям Git, а StGit использует Git для управления коллекцией исправлений (патчей).

Git поддерживает быстрое разделение и слияние версий, включает инструменты для визуализации и навигации по нелинейной истории разработки. Как и Darcs, BitKeeper, Mercurial, Bazaar и Monotone, Git предоставляет каждому разработчику локальную копию всей истории разработки, изменения копируются из одного репозитория в другой.

Удалённый доступ к репозиториям Git обеспечивается git-демоном, SSH- или HTTP-сервером. TCP-сервис git-daemon входит в дистрибутив Git и является наряду с SSH наиболее распространённым и надёжным методом доступа. Метод доступа по HTTP, несмотря на ряд ограничений, очень популярен в контролируемых сетях, потому что позволяет использовать существующие конфигурации сетевых фильтров.

Особенности реализации

Ядро Git представляет собой набор утилит командной строки с параметрами. Все настройки хранятся в текстовых файлах

Рис. 2.6: Добавление информации

- 5) Затем я создал локальный сервер, чтобы проверить правильность заполнения данных.

```
Serving pages from memory
Running in Fast Render Mode. For full rebuilds on change: hugo server --disableFastRender
Web Server is available at http://localhost:1313/ (bind address 127.0.0.1)
Press Ctrl+C to stop
```

Рис. 2.7: Проверка на локальном сервере

- 6) Когда я убедился в правильности заполнения данных, я отправил все данные на систему git и уже проверил сам сайт. По ссылке SkLjT.github.io открывается мой сайт.

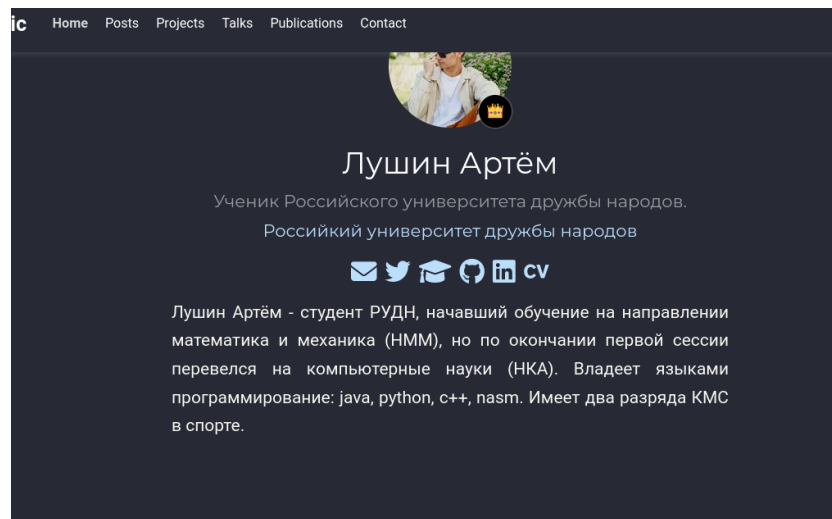


Рис. 2.8: Готовый онлайн сайт

3 Выводы

Я добавил на сайт данные о себе и научился писать посты.

Список литературы