

Отчёт по этапу №2 индивидуального проекта

Дисциплина: Операционные системы

Елизавета Андреевна Алмазова

Содержание

Цель работы	1
Задание	1
Теоретическое введение	1
Реализация этапа проекта.....	2
Выводы	3

Цель работы

Цель данного индивидуального проекта - поэтапное создание и заполнение сайта, представляющего меня как научного работника.

Задание

Добавить к сайту данные о себе:

- Разместить фотографию владельца сайта.
- Разместить краткое описание владельца сайта (Biography).
- Добавить информацию об интересах (Interests).
- Добавить информацию об образовании (Education).
- Сделать пост по прошедшей неделе.
- Добавить пост на тему по выбору:
 - Управление версиями. Git.
 - Непрерывная интеграция и непрерывное развертывание (CI/CD).

Теоретическое введение

Статический генератор сайта — программа, которая из различных исходных файлов (картинок, шаблонов в разных форматах, текстовых файлов и т.п) генерирует статический HTML-сайт. Один из ярких представителей — Hugo. 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. Установить фреймворк можно практически на любую современную платформу в несколько команд.

Фреймворк имеет очень хорошую документацию. Она доступна только на английском языке. Информация очень хорошо структурирована, что позволяет освоить данную технологию, за несколько дней. Для лучшего восприятия, практически в каждой главе есть обучающее видео от разработчиков. Все это позволяет очень быстро приступить к созданию собственных сайтов.

Основной контент или содержимое сайта хранится в формате .md в папке content. В роли контента могут выступать статьи, новости, продукты интернет магазина и прочее. Каталог «data» используется для хранения файлов конфигурации, которые Hugo может использовать при создании веб-сайта. Можно записать эти файлы в формате YAML, JSON или TOML.

Hugo имеет около 200 готовых тем, которые можно легко установить и использовать. Это значительно ускоряет процесс разработки. Темы включают в себя не только отличные дизайнерские, но и хорошие программные решения. Wowchemy предлагает удобный шаблон для создания сайта научного работника и не только, узнать больше можно на их официальном сайте.

Реализация этапа проекта

1. Разместила фотографию владельца сайта, заменив файл avatar.jpg в папке ~/work/blog/content/authors/admin на нужную картинку.
2. Изменила в этой же папке файл "index.md", разместив краткое описание владельца сайта (Biography), добавила информацию об интересах (Interests), добавила информацию об образовании (Education) (рис. 1).

```
# Display name
title: Алмазова Елизавета

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

# Role/position/tagline
role: Студентка РУДН

# Organizations/Affiliations to show in About widget
organizations:
  - name: РУДН
    url: https://www.rudn.ru

# Short bio (displayed in user profile at end of posts)
bio: Я студентка РУДН по направлению "Прикладная математика и информатика"

# Interests to show in About widget
interests:
  - Математика
  - Программирование
  -
```

Рисунок 1 - Заполнение файла с информацией о владельце сайта

3. Сделала пост по прошедшей неделе (2-6 мая, 2022 г.), создав соответствующую папку в ~/work/blog/content/post и в ней файл index.md, заполнила его (рис. 2).

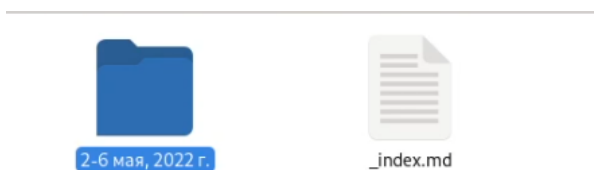


Рисунок 2 - Создание папки “2-6 мая, 2022 г.”

4. Сделала пост на тему “Управление версиями. Git.”, создав соответствующую папку в `~/work/blog/content/post` и в ней файл `index.md`, заполнила его (рис. 3).

```
37
38 ## Системы контроля версий
39
40 Системы контроля версий – это программное обеспечение для облегчения работы над изменяющейся информацией,
41 которое часто применяют при работе нескольких человек над одним проектом. Они позволяют фиксировать
42 изменения в файлах и кто их внес, отслеживать и разрешать возникающие конфликты: производить откат,
43 совмещать изменения, сделанные разными участниками, заблокировать файлы для изменения и вручную выбрать
44 версию файлов. Также они могут иметь и дополнительные функции, например, поддержка работы с несколькими
45 версиями путем сохранения общей истории изменений до точки ветвления и собственные истории каждой ветви.
46 Обычно основное дерево проекта хранится в локальном или удаленном репозитории, к которому настроен доступ
47 для участников проекта. В классических системах используется централизованная модель, т.е. файлы хранятся в
48 едином репозитории. Большинство функций по управлению версиями выполняется специальным сервером.
49 Пользователь перед началом работы с помощью команд получает нужную версию файлов. После внесения изменений
50 он размещает новую версию в хранилище. При этом предыдущие версии удалять не будут, к ним можно вернуться в
51 любой момент. Сервер также может сохранять не полную версию измененных файлов, а производить дельта-
52 компрессию, т.е. сохранять только изменения между последовательными версиями, уменьшая объем хранимых
53 данных. В распределенных системах центральный репозиторий не является обязательным.
54
55 ## Задачи, решаемые VCS
56
57 - Обеспечение командной и распределенной работы над проектом без специального инструментария,
58 предоставление возможности изменять файлы, не мешая работе других пользователей;
59 - Сохранение исходного кода, хранение всех предыдущих версий или, для уменьшения объема, изменений между
60 последовательными версиями;
61 - Ведение журнала изменений: кто из участников, когда и какие изменения внес, причем доступ к нему можно
62 ограничить;
63 - Отслеживание и разрешение конфликтов, возникающих при работе нескольких человек над одним проектом;
64 - Автоматическое или ручное объединение (совмещение) изменений;
65 - Ручной выбор нужной версии;
66 - Откат изменений;
67   • Настраиваемая блокировка файлов для изменения, предоставление привилегированного доступа только одному
68   пользователю, работающему с файлом;
69   • Поддержка работы с несколькими версиями одного файла, сохранение общей истории изменений до точки
70   ветвления версий и собственные изменения каждой ветви.
71
72 ## Основные команды git
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Рисунок 3 - Редактирование поста об управлении версиями.

Выводы

В ходе реализации второго этапа проекта я добавила к сайту данные о себе, в т.ч. числе фотографию, биографию, интересы и образование, а также добавила два поста: по прошедшей неделе и теме “Управление версиями. Git”.