# Персональный проект

Стадия 2

Карпачев Ярослав

# Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы         3.1 Добавляем биографию + фото          3.2 делаем посты          3.3 Загрузка + проверка	. 8
4	Выводы	12
Сп	писок литературы	13

# Список иллюстраций

3.1	мое фото
3.2	моя биография
3.3	мои навыки
3.4	мои курсы
3.5	пост про прощедшую неделю
3.6	пост про git
3.7	hugo
3.8	мои навыки + био
3.9	мои курсы
3.10	мои посты

## Список таблиц

# 1 Цель работы

Продолжить работы со своим сайтом. Редактировать его в соответствии с требованиями. Добавить данные о себе.

### 2 Задание

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

### 3 Выполнение лабораторной работы

### 3.1 Добавляем биографию + фото

1. Добавил фото, перетащив его в нужную папку (рис. 3.1), добавил биографию (рис. 3.2), добавил свои навыки (рис. 3.3), добавил курсы (рис. 3.4).

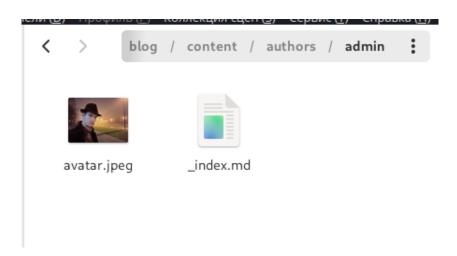


Рис. 3.1: мое фото

```
106 ---
107
108 Yaroslav Karpachev програмист frontend и AI, учился в РУДН, имею опыт работы в саобществе Github, участвовал в развитии проектов - doom emacs, NvChad, pythorch.
109 {style="text-align: justify;"}

Магкdown ▼ Ширина табуляции: 8 ▼ Ln 90, Col 44 NS
```

Рис. 3.2: моя биография

```
46 # Skills
47 # For available icons, see: https://docs.hugoblox.com/getting-started/page-builder/#icons
48 skills:
50
        items:
51
52
            description:
          percent: 60
- name: C++
54
55
56
57
58
             percent: 100
          - name: Rust
          description: ''
percent: 40
- name: fontend (css, html, js/ts)
description: ''
59
60
61
62
             percent: 80
63
64
     - name: Hobbies
        color: '#eeac02
65
         color_border: '#f0bf23'
        items:
67
68
          - name: Stock market
            description:
          percent: 100
- name: Cats
69
70
71
72
             percent: 80
73
74 # Social/Academic Networking
```

Рис. 3.3: мои навыки

```
37 education:
(38 courses:
39 - course: Professional Certificate in computer science for Artificial intelegence (edX)
40 year: 2023
41 - course: Self-driving cars with duckietown (edX)
year: 2023
43 - course: Natural language processing specialization (coursera)
44 year: 2024
```

Рис. 3.4: мои курсы

### 3.2 делаем посты

2. делаем пост про прощедшую неделю (рис. 3.5), делаем пост про Git (рис. 3.6).

```
| 1 --- | 2 title: Как прошла моя неделя | 3 date: '2024-03-15' | 4 summary: 8 марта - продолжали развивать проект на гитхабе | 5 --- | 6 | 7 ![png](1.png) | 8 | 9 ## Кратко о дней | 10 | 11 Так как это был государственный празник, мы с друзьями решили врямя зря не терять и продолжить наш проект оптимизации библиотек питон под срр. | 12 | 13 ## Что успели сделать | 16 | 16 | 17 Также мы полностью оптимизировали библиотеку itertools.
```

Рис. 3.5: пост про прощедшую неделю

```
## Технология Git кратко

7 ![png](I_png)

8 ## Технология Git кратко

10 Управление версиями с помощью Git является необходимым инструментом для разработчиков программного обеспечения. Git позволяет отслеживать изменения в коде, совместно работать над проектами и легко возвращаться к предыдущим версиям файлов.

12 Основными концепциями Git являются репозитории, коммиты и ветки. Репозиторий - это хранилище проекта, коммиты записывают изменения в файлах, а ветки позволяют независимо развивать функциональность, не влияя на основную ветвь разработки.

13 Git предоставляет команды для создания новых веток, слияния их с основной веткой, разрешения конфликтов при слиянии и многое другое. Также Git поддерхивает распределенную модель разработки, что облегчает совместную работу над проектом удаленным командам.

14 Изучение Git является обязательным для любого разработчика, так как это повышает эффективность работы, упрощает отслеживание изменений и предотвращает потерю данных. Git стал стандартом дефакто для управления версиями в индустрии разработки программного обеспечения.

15 | ## Команды для управления гит

17 Вот некоторые стандартные команды Git для управления репозиториями:

18 | . 'git init' - инициализирует новый репозиторий Git в текущей директории.

20 | . 'git clone' - ккомирует узаленный репозиторий на локальную машину.

21 | . 'git init' - инксирует изменения в репозитории с комментарием.

22 | . 'git commit' - фиксирует изменения в репозитории с комментарием.

23 | . 'git tommit' - фиксирует изменения в репозитории и комментарием.

24 | . 'git commit' - фиксирует изменения в репозитории и комментарием.

25 | . 'git theckout' - перечисляет все существующие ветки.

26 | . 'git prid' - получает другую ветку с текущей.

27 | . 'git pull' - получает другую ветку с текущей.

28 | . 'git merge' - объединяет другую ветку с текущей.

28 | . 'git pull' - получает другую ветку с текущей.

29 | . 'git pull' - получает другую ветку с текущей.

29 | . 'git pull' - получает другую ветку с текущей.
```

Рис. 3.6: пост про git

#### 3.3 Загрузка + проверка

3. Запуск hugo (рис. 3.7), загрузка на гитхаб + проверка (рис. 3.8) (рис. 3.9) (рис. ??).

Рис. 3.7: hugo

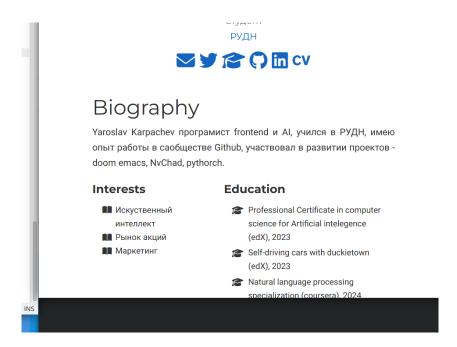


Рис. 3.8: мои навыки + био

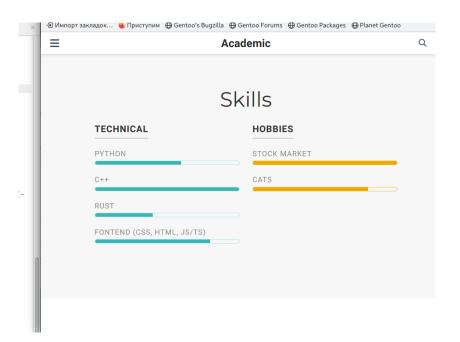


Рис. 3.9: мои курсы

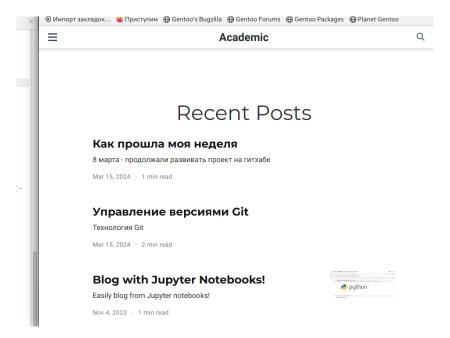


Рис. 3.10: мои посты

# 4 Выводы

Этап 2 пройден (добавленая биография и фото)

# Список литературы