

# **Создание и развертывание персонального сайта с многоязычной поддержкой на GitHub Pages**

**Разработка и деплой персонального сайта на GitHub Pages с  
поддержкой многоязычности**

Майзингер Эллина Сергеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
2.1	Основная задача . . . . .	6
2.2	Функциональные требования . . . . .	6
2.3	Технические требования . . . . .	6
<b>3</b>	<b>Теоретическое введение</b>	<b>8</b>
3.1	GitHub Pages . . . . .	8
3.2	GitHub Actions . . . . .	9
3.3	GitHub Pages и GitHub Actions: Совместное использование . . . . .	9
<b>4</b>	<b>Выполнение работы</b>	<b>11</b>
4.1	Создаем репозиторий на основе шаблона HugoBlox . . . . .	11
4.2	Клонируем репозиторий . . . . .	12
4.3	Настраиваем многоязычность . . . . .	13
4.4	Добавляем контент (посты) . . . . .	13
4.5	Настраиваем меню . . . . .	13
4.6	Фиксируем и отправляем изменения . . . . .	13
4.7	Проверяем результат: . . . . .	14
<b>5</b>	<b>Выводы</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>

## Список иллюстраций

4.1	Копирование репозитория . . . . .	11
4.2	Создание репозитория . . . . .	12
4.3	Клонирование репозитория . . . . .	12
4.4	Рабочий сайт . . . . .	14

## **Список таблиц**

# 1 Цель работы

В рамках данной работы предполагается разработка и запуск личного веб-сайта на базе GitHub Pages. Сайт будет спроектирован как интерактивное портфолио, предназначенное для демонстрации информации об авторе и публикации различного контента. Для обеспечения широкой доступности информации, сайт будет обладать двуязычной поддержкой, позволяющей пользователям переключаться между двумя языковыми версиями. Для реализации этой цели будет реализован удобный механизм переключения между языками. Важным аспектом является независимая локализация интерфейса и контента, что позволит поддерживать актуальность и достоверность информации на обоих языках.

## 2 Задание

### 2.1 Основная задача

Разработать и развернуть персональный сайт на платформе GitHub Pages, удовлетворяющий следующим требованиям:

### 2.2 Функциональные требования

Представление информации об авторе: Сайт должен содержать разделы с информацией об образовании, опыте работы, навыках, проектах и других релевантных деталях, позволяющих сформировать полное представление об авторе. Публикация контента: Предусмотреть возможность публикации статей, записей в блоге, новостей и других материалов, представляющих интерес для автора и целевой аудитории. Многоязычность: Реализовать поддержку двух языков (например, русского и английского) с возможностью удобного переключения между ними. Интерактивность: Сайт должен быть интерактивным, предусматривая возможность взаимодействия с пользователями, например, через форму обратной связи или систему комментариев (опционально).

### 2.3 Технические требования

Развертывание на GitHub Pages: Сайт должен быть размещен и функционировать на платформе GitHub Pages. Поддержка независимого перевода: Реализовать

систему управления контентом, позволяющую независимо переводить элементы интерфейса и текстовое наполнение сайта на разные языки. Адаптивный дизайн: Сайт должен корректно отображаться на различных устройствах (компьютеры, планшеты, смартфоны) с различными разрешениями экрана.

## 3 Теоретическое введение

В современной веб-разработке автоматизация играет ключевую роль в ускорении процессов и повышении эффективности. GitHub, как ведущая платформа для хостинга и совместной разработки проектов, предоставляет мощные инструменты для автоматизации деплоя и хостинга веб-сайтов: GitHub Pages и GitHub Actions.

### 3.1 GitHub Pages

GitHub Pages – это бесплатный сервис хостинга статических веб-сайтов, предлагаемый GitHub. Он позволяет пользователям размещать веб-сайты непосредственно из репозитория GitHub, делая процесс деплоя максимально простым и доступным.

Основные характеристики GitHub Pages: Хостинг статических сайтов: GitHub Pages предназначен для хостинга статических сайтов, состоящих из HTML, CSS, JavaScript, изображений и других статических ресурсов. Он не поддерживает серверный код, такой как PHP или Python. Интеграция с GitHub: GitHub Pages тесно интегрирован с GitHub, что позволяет автоматически публиковать сайт при каждом изменении в репозитории. Простота использования: Развертывание сайта на GitHub Pages не требует специальных знаний или инструментов. Достаточно загрузить статические файлы в репозиторий и настроить GitHub Pages. Бесплатность: GitHub Pages предоставляется бесплатно для личных и организационных проектов. Поддержка пользовательских доменов: GitHub Pages



позволяет использовать собственные доменные имена для веб-сайтов. Поддержка HTTPS: GitHub Pages автоматически обеспечивает шифрование HTTPS для всех сайтов.

## 3.2 GitHub Actions

GitHub Actions – это платформа для автоматизации, интегрированная непосредственно в GitHub. Она позволяет автоматизировать различные задачи в процессе разработки, такие как сборка, тестирование, деплой и многое другое.

Основные характеристики GitHub Actions:

Автоматизация рабочих процессов: GitHub Actions позволяет создавать автоматизированные рабочие процессы (workflows), которые выполняются в ответ на определенные события в репозитории, такие как push, pull request, создание issue и т.д. Основана на YAML: Рабочие процессы описываются с помощью файлов YAML, что делает их легко читаемыми и редактируемыми. Широкий спектр возможностей: GitHub Actions позволяет автоматизировать практически любые задачи, связанные с разработкой и деплоем. Большое количество готовых actions: GitHub Marketplace предлагает множество готовых actions, которые можно использовать в своих рабочих процессах. Кроссплатформенность: GitHub Actions поддерживает выполнение задач на различных операционных системах (Linux, Windows, macOS). Интеграция с GitHub: GitHub Actions тесно интегрирован с GitHub, что позволяет легко получать доступ к репозиторию, переменным окружения и другим ресурсам.

## 3.3 GitHub Pages и GitHub Actions: Совместное

### использование

GitHub Actions может быть использован для автоматизации процесса деплоя статических сайтов на GitHub Pages. Например, можно настроить рабочий про-

цесс, который будет автоматически собирать сайт из исходного кода (например, с использованием статических генераторов сайтов, таких как Jekyll или Hugo), а затем разворачивать его на GitHub Pages при каждом push в определенную ветку репозитория.

Преимущества совместного использования GitHub Pages и GitHub Actions:

Автоматизация деплоя: Автоматический деплой сайта при каждом изменении в репозитории. Улучшение процесса разработки: Автоматизация задач сборки и тестирования позволяет сосредоточиться на разработке, а не на рутинных операциях. Повышение эффективности: Автоматизация сокращает время, необходимое для разворачивания изменений на сайте. Уменьшение количества ошибок: Автоматизированные рабочие процессы снижают вероятность человеческих ошибок при деплое.

## 4 Выполнение работы

### 4.1 Создаем репозиторий на основе шаблона HugoBlox

Мы идем на страницу шаблона: <https://github.com/HugoBlox/theme-blog>.

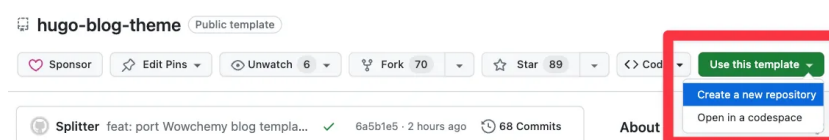


Рис. 4.1: Копирование репозитория

Мы создаем новый репозиторий, используя шаблон, нажав кнопку “Use this template”. Мы даем репозиторию имя (например, your-username.github.io).

## 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-blog ▾

Start your repository with a template repository's contents.

☐ Include all branches

Copy all branches from HugoBlox/theme-blog and not just the default branch.

### Owner \*

 esmayzinger ▾

### Repository name \*

esmayjjzinger.github.io

✔ esmayjjzinger.github.io is available.

Great repository names are short and memorable. Need inspiration? How about [solid-fishstick](#) ?

### 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.

 You are creating a public repository in your personal account.

Create repository

Рис. 4.2: Создание репозитория

## 4.2 Клонирование репозиторий

Теперь мы клонируем репозиторий на свой компьютер, чтобы редактировать его локально. Мы используем команду `git clone https://github.com/esmayzinger/esmayzinger.github.io`

```
ellinamayzinger — -zsh — 80x24
Last login: Sat Mar  8 17:52:00 on ttys000
(base) ellinamayzinger@MacBook-Air-Ellina ~ % git clone https://github.com/esmayzinger/esmayzinger.github.io.git
```

Рис. 4.3: Клонирование репозитория

## 4.3 Настраиваем многоязычность

Мы открываем репозиторий в IDE (VS Code, PyCharm и т. д.). В папке `content` мы находим папку `en` (английский контент). Мы копируем папку `en` и вставляем её в `content`. Мы переименовываем скопированную папку в `ru` (русский контент). В папке `ru` мы редактируем файл `_index.md`, переводя информацию об авторе на русский. Мы добавляем аватар и личную информацию.

## 4.4 Добавляем контент (посты)

В папке `content` мы видим папку `blog` с предустановленными постами. Мы редактируем файл `en/blog/_index.md`, чтобы настроить заголовок страницы блога. Мы создаем новые Markdown-файлы в папках `en/blog` и `ru/blog` для новых постов на английском и русском языках, соответственно. Мы редактируем шаблон `index.md` и добавляем пробный пост.

## 4.5 Настраиваем меню

Мы копируем файл `menus.en.yaml` в `menus.ru.yaml`. Мы редактируем `menus.ru.yaml`, раскомментируя строки для русского языка и переводя пункты меню.

## 4.6 Фиксируем и отправляем изменения

Мы сохраняем все наши изменения. Мы коммитим изменения (`git add .`, `git commit -m "Add multilingual support and content"`). Мы отправляем изменения на GitHub (`git push origin main`).

##Ждем деплоя Мы идем на страницу <https://github.com/esmayzinger/esmayzinger.github.io.git> и ждем, пока сайт задеплоится.

## 4.7 Проверяем результат:

После успешного деплоя мы открываем сайт по адресу <https://esmayzinger.github.io/>. Мы проверяем, что сайт работает, многоязычность настроена, и контент отображается корректно.

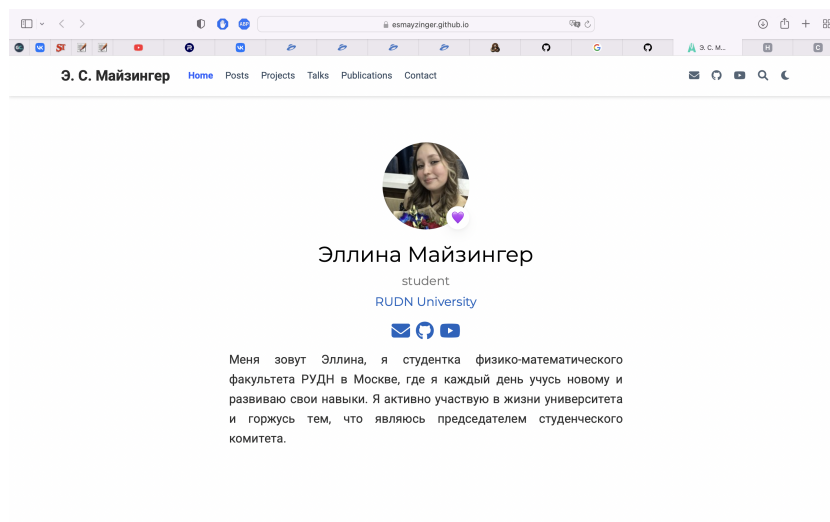


Рис. 4.4: Рабочий сайт

## 5 Выводы

В результате проделанной работы был успешно модифицирован контент веб-страницы, размещенной на платформе GitHub Pages. Процесс включал в себя внесение изменений в существующие текстовые и медиа-материалы, а также, возможно, добавление новых элементов для улучшения представления информации. После завершения этапа редактирования был осуществлен деплой веб-страницы, что позволило опубликовать обновленную версию контента и сделать ее доступной для просмотра в сети Интернет.

# Список литературы

esystem rudn

<https://github.com/yamadharm>

Кристина Александровна Штепа