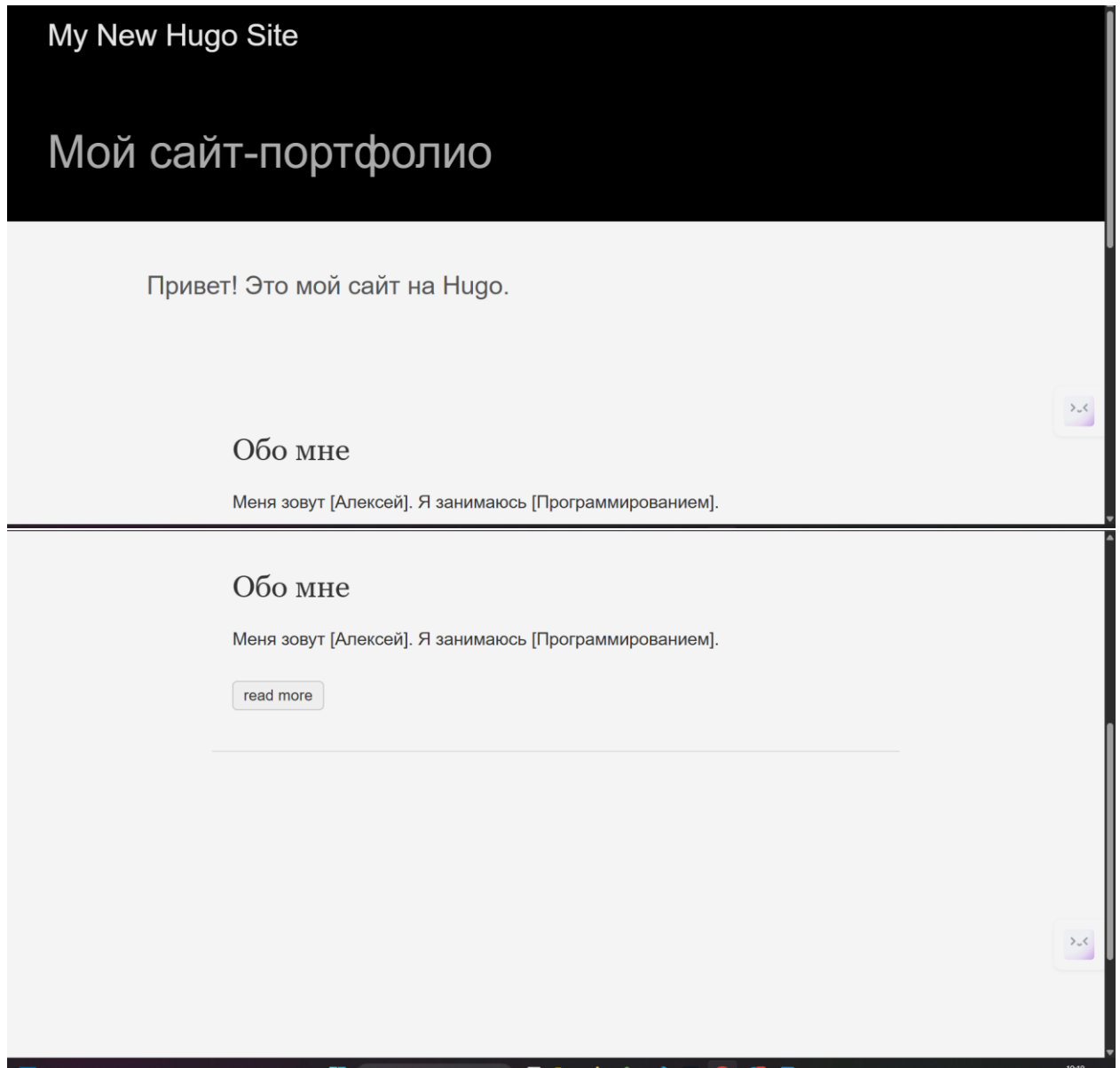


В рамках этого задания была развернута CMS Hugo на локальном компьютере, создано несколько страниц портфолио, сгенерирован статический сайт и размещен на GitHub Pages. Исходный код сайта был размещен в отдельном репозитории GitHub, не получилось настроить автоматическое развертывание с использованием GitHub Actions.



[Ссылка на гитхаб](#)

Скрипт для развертывания Hugo с помощью GitHub Actions
(.github/workflows/deploy.yml)

name: Deploy Hugo

on:

push:

branches: [main] # Или master, если используете эту ветку

jobs:

deploy:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

with:

submodules: true # Это важно для загрузки темы

fetch-depth: 0 # Для корректной работы дат в Hugo

- name: Setup Hugo

uses: peaceiris/actions-hugo@v2

with:

hugo-version: "latest"

extended: true # Для поддержки SCSS

- name: Build

run: hugo --minify --baseURL "https://\${{ github.repository_owner }}.github.io/My-portfolio/"

- name: Deploy

uses: peaceiris/actions-gh-pages@v3

with:

github_token: \${{ secrets.GITHUB_TOKEN }}

publish_dir: ./public

keep_files: false # Очищает папку перед деплоем

Структура скрипта (deploy.yml):

- name: Deploy Hugo: Определяет имя workflow, которое будет отображаться в списке workflow в GitHub Actions.
- on:: Указывает триггеры, при которых workflow должен запускаться.
- push:: Workflow запускается при каждом push в ветку, указанную в branches.

* `branches: [main]`: Workflow запускается только при push в ветку main (или master, если вы используете эту ветку в своем репозитории).

- `jobs::` Определяет одну или несколько задач, которые должны быть выполнены в рамках workflow.

- `deploy::` Определяет задачу с именем deploy, которая содержит шаги для сборки и развертывания Hugo сайта.

* `runs-on: ubuntu-latest`: Указывает, что задача должна выполняться на виртуальной машине с Ubuntu (последней версии).

* `steps::` Определяет последовательность шагов, которые должны быть выполнены в рамках задачи deploy.

Описание каждого шага (steps):

1. `uses: actions/checkout@v4`: Использует действие `actions/checkout@v4` для извлечения (checkout) исходного кода из репозитория.

- `with::` Конфигурирует действие checkout.

* `submodules: true`: Важно! Указывает, что нужно загрузить подмодули (submodules). Это необходимо, если вы используете тему Hugo, которая установлена как подмодуль Git.

* `fetch-depth: 0`: Критически важно! Указывает, что нужно получить полную историю Git. Это необходимо для корректной работы функций Hugo, которые зависят от истории Git, например, для отображения дат последнего изменения файлов.

2. `name: Setup Hugo`: Задаёт имя для шага "Setup Hugo" (отображается в логах GitHub Actions).

- `uses: peaceiris/actions-hugo@v2`: Использует действие `peaceiris/actions-hugo@v2` для установки Hugo. Это значительно упрощает процесс установки Hugo в окружении GitHub Actions.

- `with::` Конфигурирует действие hugo.

* `hugo-version: "latest"`: Указывает, что нужно установить последнюю версию Hugo.

* `extended: true`: Указывает, что нужно установить расширенную версию Hugo. Расширенная версия Hugo включает поддержку SCSS, которая часто используется для стилизации веб-сайтов.

3. `name: Build`: Задаёт имя для шага "Build" (отображается в логах GitHub Actions).

- `run: hugo --minify --baseURL "https://${{ github.repository_owner }}.github.io/My-portfolio/"`: Выполняет команду hugo для сборки статического сайта.

* `--minify`: Указывает Hugo минифицировать HTML, CSS и JavaScript файлы, уменьшая их размер и улучшая производительность сайта.

* `--baseUrl "https://${{ github.repository_owner }}.github.io/My-portfolio/"`: Задаёт базовый URL для сайта. `${{ github.repository_owner }}` заменяется именем пользователя GitHub, а "My-portfolio" - именем вашего репозитория. Важно: Измените "My-portfolio", если имя вашего репозитория отличается! Этот URL используется для создания абсолютных ссылок на вашем сайте.

4. `name: Deploy`: Задаёт имя для шага "Deploy" (отображается в логах GitHub Actions).

- `uses: peaceiris/actions-gh-pages@v3`: Использует действие `peaceiris/actions-gh-pages@v3` для развертывания сайта на GitHub Pages.

- `with::` Конфигурирует действие `gh-pages`.

- `github_token: ${{ secrets.GITHUB_TOKEN }}`: Предоставляет действие токен GitHub, необходимый для записи в репозиторий GitHub Pages. `${{ secrets.GITHUB_TOKEN }}` – это секрет, автоматически созданный GitHub Actions, который даёт право `workflow` записывать в репозиторий.

- `publish_dir: ./public`: Указывает каталог, содержащий файлы, которые нужно развернуть. В данном случае, это каталог `public`, куда Hugo помещает сгенерированные файлы сайта.

- `keep_files: false`: Важно! Указывает, что перед развертыванием нужно очистить существующую папку GitHub Pages. Это гарантирует, что на сайте не останутся старые файлы.