МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



**ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И   
ТЕХНОЛОГИЧЕСКОГО ОБРАЗОВАНИЯ**

**Кафедра информационных технологий и электронного обучения**

Основная профессиональная образовательная программа

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) «Технологии разработки программного обеспечения»

форма обучения – очная

**Курсовая работа**

«Проектирование и разработка электронного портфолио по дисциплине “Управление программными проектами”»

Обучающегося 4 курса

Величко Арсения Александровича

Научный руководитель:

кандидат физико-математических наук,   
доцент кафедры ИТиЭО

Власов Дмитрий Викторович

Санкт-Петербург

2025

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_9facf1f5zbz5)

[1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_t8clj6bhoda7)

[1.1 Требования к портфолио 5](#_t3wo55lu52x1)

[1.2 Выбор инструментов реализации электронного портфолио 7](#_46g5mqfswbx8)

[2 ПРАКТИЧЕСКАЯ ЧАСТЬ 10](#_8abigvdrylfx)

[2.1 Подготовка исходных материалов 10](#_ed1afiewg6rg)

[2.2 Добавление метаинформации (front matter) 11](#_beqntwy7clau)

[2.3 Создание и структура репозитория GitHub 12](#_tlyg6t4obs7v)

[2.4 Настройка Jekyll через \_config.yml 12](#_9zfs59ynaoen)

[2.5 Автоматизация публикации (CI/CD) 15](#_m9ho8cdbxstd)

[2.6 Проверка выполнения 18](#_qk9wy4y8i46e)

[2.7 Актуализация контента 18](#_b6wpwreiykii)

[2.8 Результаты и перспективы 19](#_urw7et2vcga3)

[ЗАКЛЮЧЕНИЕ 20](#_n8sy9nwhs2eh)

[ЛИТЕРАТУРА 21](#_53yzn1r9v5l9)

# 

# ВВЕДЕНИЕ

В условиях активного внедрения цифровых технологий в образовательный процесс востребованность электронного портфолио, которое фиксирует достижения и компетенции обучающегося, подтверждается исследованиями, демонстрирующими его эффективность как инструмента самооценки, внутренней мотивации и внешней презентации результатов обучения [1; 3; 4]. В сфере «Управление программными проектами» портфолио обеспечивает прозрачность прогресса, создаёт доказательную базу освоения методологий и способствует формированию профессиональной репутации выпускника ИТ‑специальности. Бесплатные сервисы GitHub Pages и статический генератор Jekyll, признанные сообществом разработчиков [9; 10], позволяют развернуть надёжное, поисково‑оптимизированное и легко обновляемое веб‑портфолио без затрат на хостинг, что усиливает актуальность заявленной темы для студентов.

Цель курсовой работы заключается в проектировании и разработке электронного портфолио студента по дисциплине «Управление программными проектами» с применением GitHub Pages и Jekyll. Для достижения цели необходимо выполнить анализ научных публикаций и нормативных требований к студент‑ориентированному портфолио, определить функциональные и технические характеристики будущего веб‑ресурса, спроектировать его информационную архитектуру и визуальный стиль, реализовать прототип на выбранных технологиях, провести испытания, сформулировать рекомендации по дальнейшему расширению и сопровождению.

Объектом исследования выступает процесс создания и практического использования студенческих электронных портфолио в высшей школе. Предмет исследования — методические и технологические средства, обеспечивающие разработку, опубликование и поддержку портфолио средствами GitHub Pages и Jekyll.

Структура работы включает введение, в котором раскрываются актуальность, цель, задачи, объект и предмет исследования. Первая глава содержит теоретический обзор понятия электронного портфолио, а также анализ существующих подходов к его внедрению в образовательные программы. Вторая глава посвящена обоснованию выбора инструментов, проектированию структуры, описанию этапов разработки и развертыванию готового решения. Заключение подводит итоги, оценивает выполненные задачи и намечает направления дальнейших исследований.

# 

# 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Требования к портфолио

Электронное портфолио понимается как динамическая совокупность артефактов, фиксирующих академические и практические достижения обучающегося. Исследования показывают, что регулярная работа с портфолио усиливает внутреннюю мотивацию и способствует более осмысленному планированию собственных образовательных траекторий [1; 4]. Подготовка и анализ материалов формируют культуру самооценки, повышают ответственность за результат обучения и позволяют студенту увидеть развитие компетенций во времени.

В учебной среде портфолио играет роль альтернативного средства оценивания. Преподаватель получает объективные доказательства освоения дисциплины, а студент — наглядное подтверждение своего прогресса [3]. Рефлексивная составляющая также критична: короткие пояснения к каждому артефакту помогают осмыслить применимость изученных методов управления проектами и выявить зоны, требующие дальнейшего внимания [8]. Такой подход согласуется с современными педагогическими моделями, где приоритет отдан формированию метакомпетенций, а не исключительно знаниевых результатов.

На рынке труда электронное портфолио выполняет функцию публичного резюме, подкреплённого фактами. Работодатели получают доступ к реальным примерам кода, отчётам и проектным артефактам, что облегчает оценку уровня подготовленности кандидата [7]. Портфолио подтверждает способности вести документацию, соблюдать сроки и применять методологии разработки, тем самым укрепляя доверие и повышая шансы на успешное трудоустройство [6]. Дополнительную ценность создаёт возможность быстро актуализировать материалы по мере появления новых проектов, что поддерживает позитивный профессиональный образ выпускника.

Переход к сетевым технологиям усиливает преимущества портфолио. Размещение в открытом доступе даёт студенту и наставнику круглосуточный доступ к материалам, исключает риск потери данных и расширяет аудиторию, готовую предоставить обратную связь [5]. Интеграция мультимедийных форматов делает представление достижений насыщенным и наглядным, что особенно важно для дисциплины «Управление программными проектами», ориентированной на комплексное демонстрирование результатов командной работы.

Эффективность портфолио зависит от соответствия ряду требований. Структура сайта должна оставаться логичной и последовательной. Разделы рекомендуется располагать так, чтобы посетитель без затруднений переходил от общей информации к конкретным проектным случаям, раскрывающим применение методик управления программными проектами [3]. Навигационные элементы должны быть однозначными, а содержание — разделённым на краткие тематические блоки с чёткими заголовками.

Адаптивное отображение страниц на устройствах с различным размером экрана необходимо, поскольку студенты и работодатели часто просматривают материалы с мобильных телефонов. Высокая скорость загрузки достигается благодаря использованию статической генерации и оптимизации медиаконтента, что подтверждают рекомендации разработчиков GitHub Pages и Jekyll [9; 10]. Статический характер ресурса снижает нагрузку на сервер, минимизирует вероятность отказа и улучшает общую доступность.

Безопасность и простота сопровождения также входят в число обязательных критериев. Размещение исходного кода в системе контроля версий GitHub фиксирует все изменения, облегчает коллективную работу и обеспечивает прозрачность истории проекта [9]. Автоматизация через GitHub Actions исключает ручные ошибки при обновлении, ускоряет выпуск новых версий портфолио и демонстрирует владение практикой непрерывной поставки, что соответствует требованиям профессионального стандарта ИТ‑специалиста.

Таким образом, портфолио выполняет одновременно образовательную и карьерную функции, а строгие требования к структуре, адаптивности, производительности и безопасности обеспечивают его долговременную полезность. Соблюдение указанных критериев с применением GitHub Pages, GitHub Actions и Jekyll создаёт надёжную основу для демонстрации индивидуальных достижений в области управления программными проектами и подтверждения сформированных компетенций.

## 1.2 Выбор инструментов реализации электронного портфолио

Выбор технологического набора для разработки электронного портфолио определяется совокупностью требований: бесплатное размещение, надёжность, простота обновления, отсутствие необходимости обслуживания серверного окружения и возможность демонстрации навыков разработки, что особенно важно для студентов‑разработчиков [1]. Этим критериям в полной мере соответствует связка GitHub Pages, GitHub Actions и Jekyll.

GitHub Pages — это сервис, предоставляемый платформой GitHub для публикации статических веб‑страниц из репозитория без дополнительной настройки серверов. Решение интегрируется с системой контроля версий, что облегчает сопровождение проекта и документирует каждое изменение, повышая прозрачность учебного процесса [9]. В рамках дисциплины «Управление программными проектами» использование GitHub Pages дополнительно демонстрирует освоение методик коллективной разработки, так как вся история правок и распределение ролей фиксируются автоматически.

Jekyll — статический генератор сайтов, рекомендованный разработчиками GitHub Pages в качестве основного средства подготовки разметки и шаблонов [10]. Он преобразует исходные файлы Markdown и разметки Liquid в готовые HTML‑страницы, что упрощает реализацию адаптивного дизайна и разделение содержимого и представления. Благодаря возможности подключать темы и расширения Jekyll удовлетворяет эстетическим и функциональным требованиям к портфолио при сохранении высокой производительности, поскольку итоговый сайт состоит исключительно из статических файлов.

GitHub Actions дополняет эту связку, обеспечивая автоматизацию процессов сборки и развертывания. Каждый раз при изменении репозитория GitHub Actions запускает сценарий, который пересобирает проект Jekyll и публикует обновлённые файлы на GitHub Pages. Такой конвейер полностью исключает ручные операции, сокращает вероятность ошибок и иллюстрирует студенту практику непрерывной интеграции и поставки, что соответствует компетенциям, заложенным в программе дисциплины.

Для полноты анализа рассмотрим две альтернативы. WordPress представляет собой систему управления содержимым, популярную за счёт большого числа готовых расширений. Однако WordPress требует динамической среды исполнения, регулярных обновлений и баз данных, что усложняет обслуживание и повышает риски уязвимостей. Кроме того, в студенческом проекте лишняя функциональность WordPress оказывается избыточной, а производительность портфолио замедляется из‑за дополнительных запросов к серверу.

Tilda — облачный конструктор сайтов с визуальным редактором. Он позволяет быстро создавать страницы без знания разметки, но накладывает ограничения на гибкость шаблонов и подключение внешних библиотек. Бесплатная версия Tilda ограничивает объём проекта и брендирует сайт, что снижает академическую ценность решения. Дополнительной проблемой становится ограниченный контроль над исходным кодом: студент не может продемонстрировать владение системами контроля версий и автоматизированными процессами поставки, тогда как это является значимой частью компетенций в области управления программными проектами.

По сравнению с перечисленными альтернативами выбранная связка GitHub Pages — GitHub Actions — Jekyll обеспечивает:

1. бесплатное размещение и отсутствие скрытых платежей;
2. полный контроль над исходным кодом и историей изменений;
3. высокую скорость загрузки благодаря статической природе страниц;
4. автоматическое развертывание при каждом изменении содержимого;
5. расширяемость через открытую экосистему тем и плагинов Jekyll;
6. формирование навыков работы с современными практиками DevOps, что особенно важно в контексте дисциплины «Управление программными проектами» [3].

Таким образом, выбор GitHub Pages, GitHub Actions и Jekyll обоснован их соответствием функциональным, образовательным и организационным требованиям проекта, тогда как WordPress и Tilda, несмотря на популярность, оказываются менее подходящими по критериям лёгкости сопровождения, открытости кода и демонстрационной ценности.

# 

# 2 ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1 Подготовка исходных материалов

Работа начинается с перевода отчётов по лабораторным и самостоятельным работам из формата DOCX в Markdown. Для автоматизации применён универсальный конвертер Pandoc[11]. Команда выполняется в терминале macOS/Linux и формирует файл с тем же именем и расширением .md:

Листинг 1. Пример команды для конвертации документа из формата DOCX в формат Markdown

pandoc LR‑1.docx -o lr‑1.md --from=docx --to=gfm --standalone

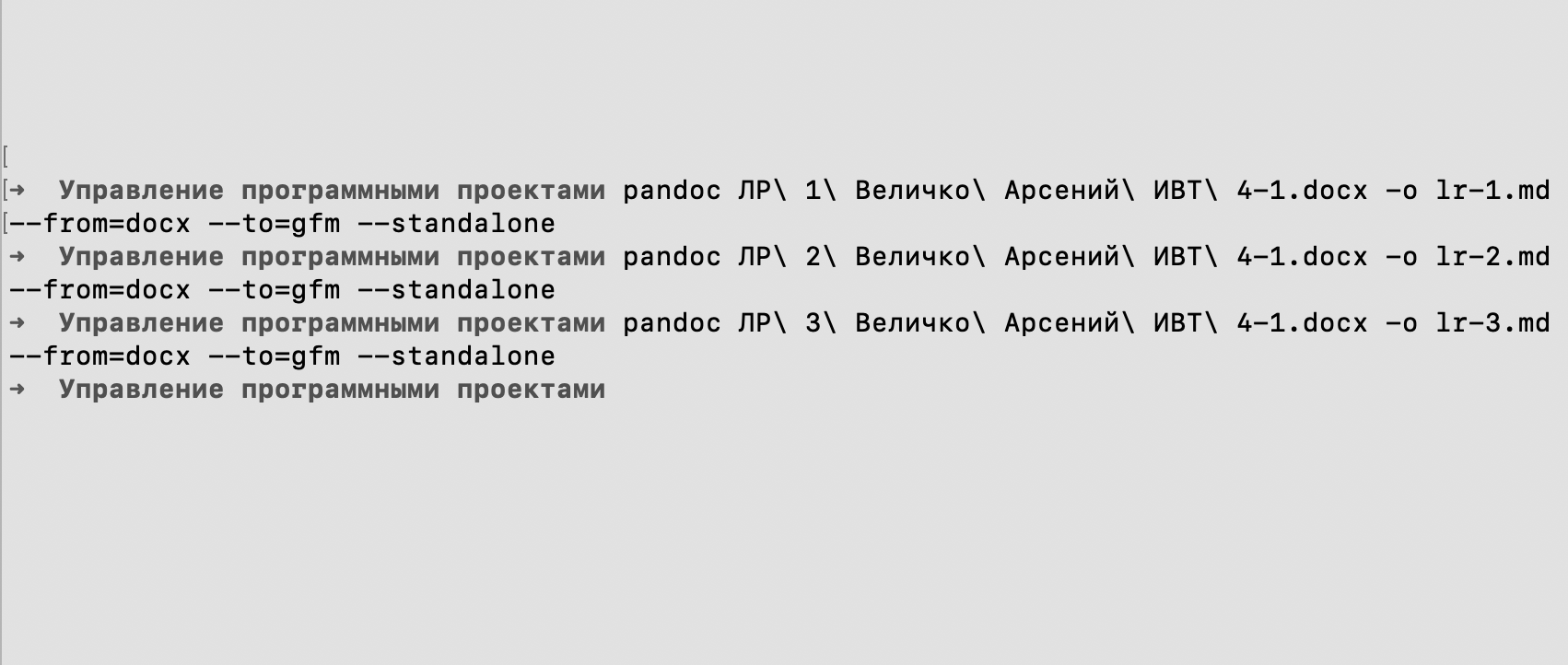


Рисунок 1. Конвертация отчёта из DOCX в Markdown с помощью Pandoc

После генерации Markdown‑файл просматривается в редакторе кода, проверяются корректность таблиц, рисунков и русских абзацев. При необходимости заголовки правятся вручную, чтобы сохранить семантический уровень #, ##, ### и т. д.

## 2.2 Добавление метаинформации (front matter)

Каждый материал Jekyll интерпретирует как страницу благодаря фронт‑маттеру — блоку YAML, помещённому в начале файла между тройными дефисами. Пример для первой лабораторной работы:

Листинг 2. Пример фронт‑маттера для отчета по лабораторной работе №1

---

layout: default

title: "Лабораторная работа 1"

permalink: /lab-reports/lr-1/

id: lr-1

---

Наличие поля title позволяет настраивать отображаемое имя статьи в боковом меню, а permalink формирует человекочитаемый адрес.

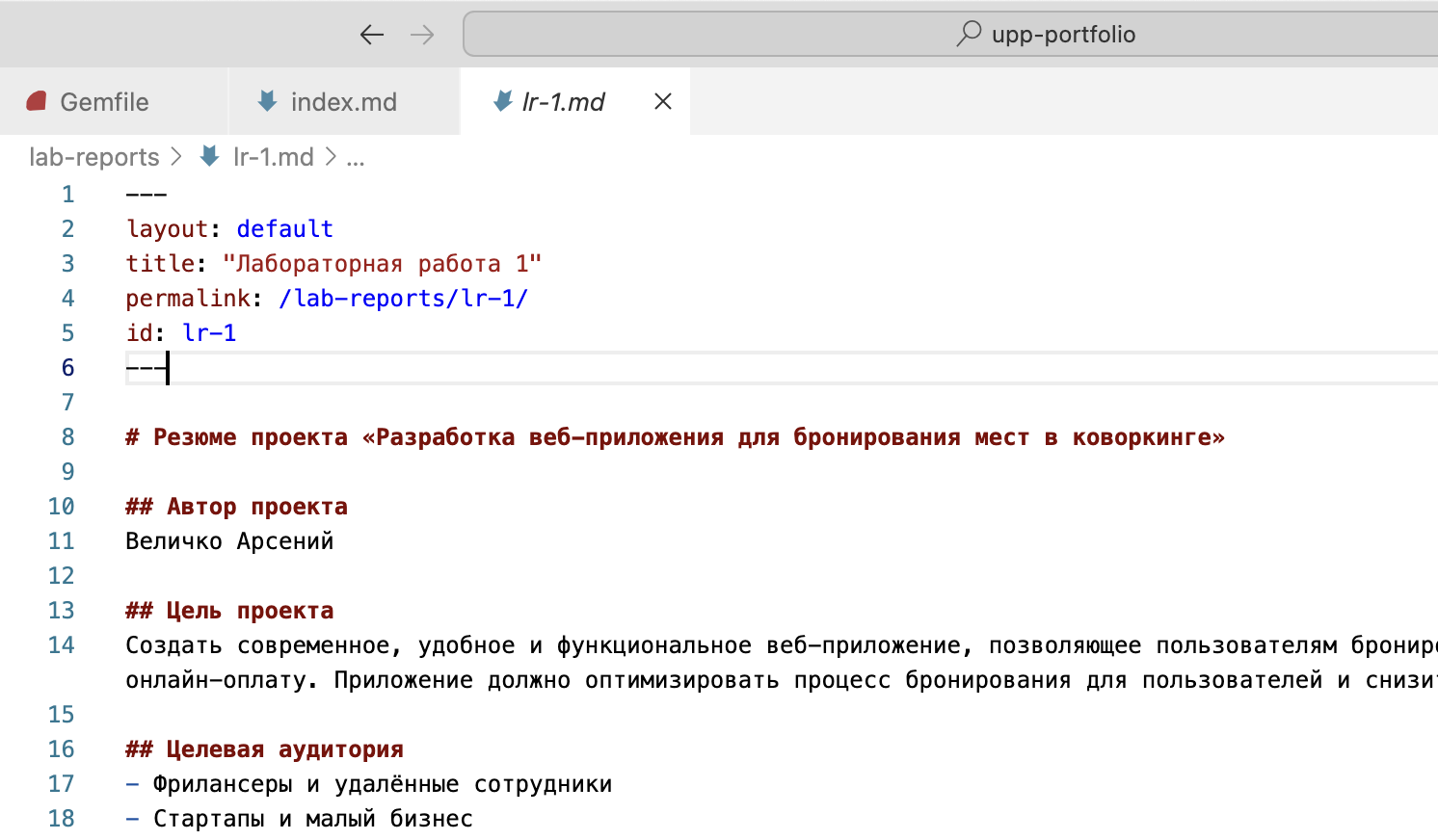


Рисунок 2. Файл lr‑1.md после добавления фронт‑маттера в редакторе VS Code

## 2.3 Создание и структура репозитория GitHub

В аккаунте GitHub создаётся открытый репозиторий “upp‑portfolio”. Корневая структура проекта после начального коммита выглядит следующим образом:

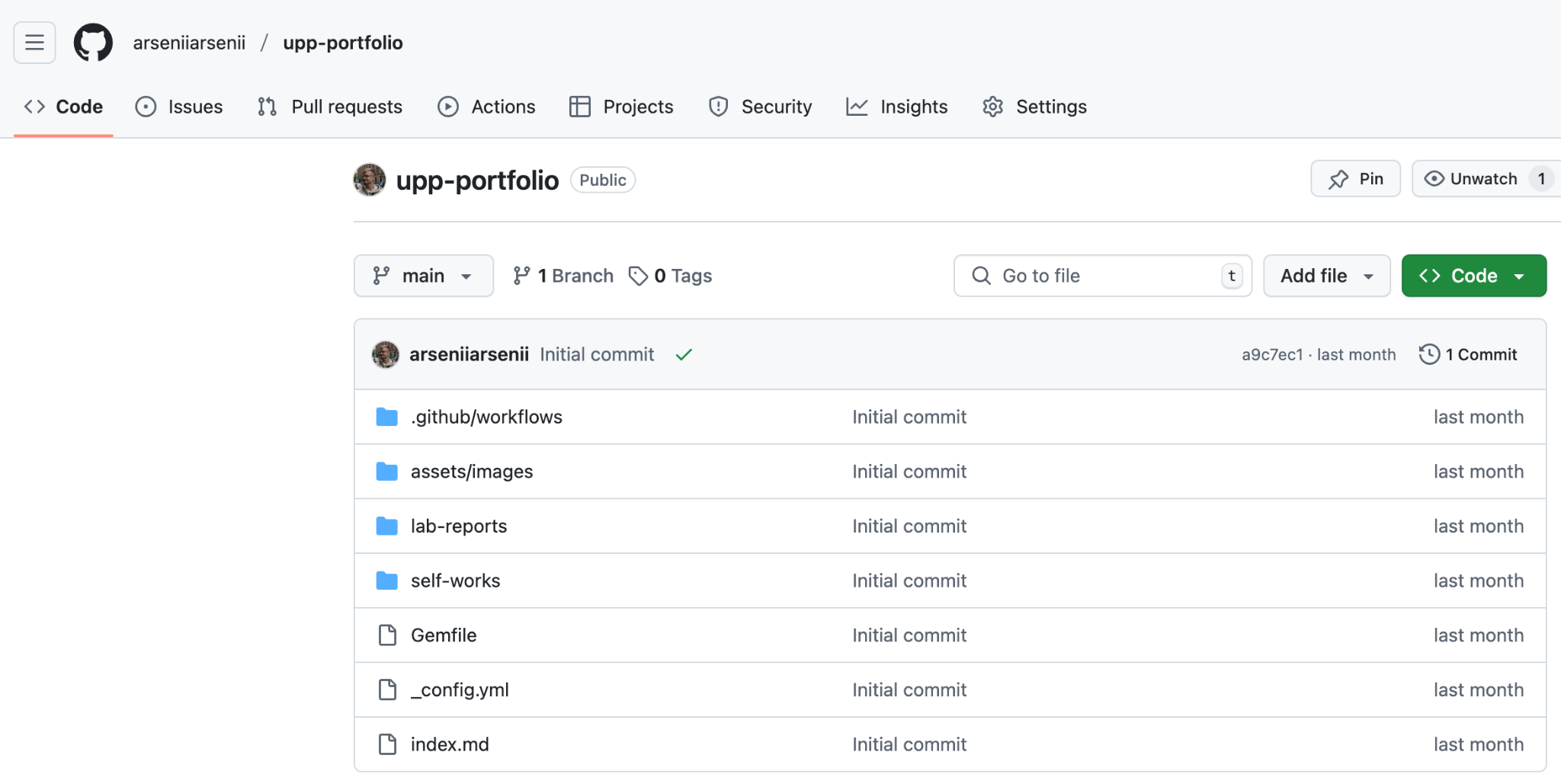


Рисунок 3. Структура репозитория веб‑портфолио в GitHub Web

## 2.4 Настройка Jekyll через \_config.yml

Файл конфигурации Jekyll полностью определяет поведение сайта. Итоговое содержимое представлено в листинге 3.

Листинг 3. Файл \_config.yml проекта

title: Управление программными проектами

description: Портфолио по дисциплине "Управление программными проектами"

author: Величко Арсений

baseurl: "/upp-portfolio"

url: "https://github.com/arseniiarsenii"

remote\_theme: zendesk/jekyll-theme-zendesk-garden@main

markdown: GFM

plugins:

- jekyll-relative-links

relative\_links:

enabled: true

collections: true

include:

- "\*.md"

- "\*\*/\*.md"

defaults:

-

scope:

path: "" # all files

values:

layout: "default"

sidebar:

- label: Лабораторные работы

children:

- lr-1

- lr-2

- lr-3

- lr-4

- lr-5

- label: Самостоятельные работы

children:

- sr-1

- sr-2

- sr-3

Файл \_config.yml задаёт глобальные параметры, определяющие работу Jekyll‑сайта, который публикуется через GitHub Pages. Он записан в формате YAML; каждая пара «ключ — значение» считывается движком Jekyll при сборке и влияет на генерацию HTML‑страниц.

Первая тройка строк (title, description, author) формирует метаданные сайта. Значение title выводится в элементе <title> браузера и участвует в формировании либо хлебных крошек, либо заголовка главной страницы в зависимости от выбранной темы. Параметр description используется для тега <meta name="description">, что повышает корректность сниппета в поисковой выдаче. Поле author хранит имя владельца портфолио и может подставляться в подписи к страницам или блоке «О сайте».

Связка baseurl и url описывает адрес размещения ресурса. url содержит корневой домен публикации — здесь это профиль GitHub пользователя. baseurl задаёт подкаталог, в котором будет развёрнут скомпилированный сайт; GitHub Pages использует именно такую схему для проектов, отличных от персонального сайта. Jekyll автоматически подставляет оба значения при генерации абсолютных ссылок и ссылок‑каноников.

Параметр remote\_theme указывает на удалённую тему оформления. В примере подключается репозиторий zendesk/jekyll-theme-zendesk-garden с веткой main, что освобождает автора от необходимости хранить файлы темы в собственном репозитории. Jekyll при сборке подтягивает шаблоны и стили из внешнего источника через Ruby‑gem‑механизм.

Опция markdown: GFM заставляет Jekyll обрабатывать исходные .md‑файлы по диалекту GitHub Flavored Markdown[12], поддерживающему расширенные таблицы, подстроки, зачёркнутый текст и другие элементы, принятые в экосистеме GitHub.

Блок plugins включает расширения, которые нужно загрузить до начала сборки. Подключён модуль jekyll‑relative‑links, корректирующий относительные ссылки, чтобы они оставались рабочими как в GitHub‑репозитории, так и в собранном сайте. Дополнительная секция relative\_links активирует плагин и уточняет, что он должен обрабатывать не только обычные страницы, но и документы из коллекций.

Раздел include перечисляет шаблоны файлов, которые Jekyll обязан скопировать в выходную директорию без дополнительной фильтрации. Здесь явно указаны одиночные и вложенные Markdown‑файлы, что гарантирует отсутствие потерь контента при генерации.

Секция defaults задаёт значения по умолчанию для всех обрабатываемых файлов. В примере определён один объект: его область (scope) охватывает весь путь проекта, а все элементы получают макет layout: default, определённый темой. Это избавляет от необходимости вносить аналогичную строчку в каждый фронт‑маттер.

Наконец, объект sidebar описывает структуру бокового меню. Он содержит две группы: «Лабораторные работы» и «Самостоятельные работы». В каждой группе перечислены слаги файлов без расширения; тема Garden рендерит из этого списка многоуровневое меню навигации. Когда автор добавляет новый отчёт, достаточно внести его идентификатор в соответствующий массив, и ссылка появится в интерфейсе автоматически.

Таким образом, \_config.yml концентрирует настройки адресации, внешнего вида, обработки контента и логики меню; вся дальнейшая работа над портфолио сводится к добавлению Markdown‑страниц и небольшому обновлению массивов sidebar без изменения программного кода.

## 2.5 Автоматизация публикации (CI/CD)

Для автоматизации публикации портфолио был описан (CI/CD) конвейер, в формате, специфичном для платформы GitHub Actions. Полный исходный код конвейера приведен в листинге 4.

Листинг 4. CI‑сценарий сборки и развёртывания

name: Deploy Jekyll site to Pages

on:

push:

branches: ["main"] # Запуск при каждом пуше в основную ветку

workflow\_dispatch: # Возможность ручного запуска из интерфейса

permissions:

contents: read

pages: write

id-token: write # Требуется для публикации через OIDC

concurrency:

group: "pages"

cancel-in-progress: true

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Checkout

uses: actions/checkout@v4 # Получение кода

- name: Setup Ruby

uses: ruby/setup-ruby@v1 # Установка Ruby и Bundler

with:

ruby-version: '3.2'

bundler-cache: true

- name: Setup Pages

id: pages

uses: actions/configure-pages@v4

- name: Build site

run: |

gem install bundler # Установка Bundler

bundle install # Загрузка зависимостей

bundle exec jekyll build # Генерация статического сайта

- name: Upload artifact

uses: actions/upload-pages-artifact@v3

with:

path: ./\_site # Публикуемый каталог

deploy:

environment:

name: github-pages

url: ${{ steps.deployment.outputs.page\_url }}

runs-on: ubuntu-latest

needs: build

steps:

- name: Deploy to GitHub Pages

id: deployment

uses: actions/deploy-pages@v4

Файл представляет собой сценарий GitHub Actions, автоматизирующий сборку и выкладку статического сайта, созданного на Jekyll, в хостинг GitHub Pages. В строке name задаётся понятное обозначение рабочего процесса, чтобы его было легко идентифицировать во вкладке Actions. Раздел on содержит два способа запуска: событие push, ограниченное веткой main, гарантирует, что конвейер стартует при каждом обновлении основной версии проекта, а параметр workflow\_dispatch разрешает принудительный запуск вручную из веб‑интерфейса. Блок permissions определяет минимально необходимый набор прав для токена, выдаваемого при исполнении задачи: чтение содержимого репозитория, запись артефактов в раздел Pages и возможность подписывать деплой через механизм OpenID Connect. Секция concurrency предотвращает параллельные публикации: при одновременных пушах новая сборка отменяет предыдущую, сохраняя консистентность конечного сайта.

В разделе jobs описаны два задания. Первое, build, выполняется на виртуальной машине Ubuntu‑latest. Оно включает последовательные шаги: извлечение исходного кода, установку среды Ruby с кешированием Bundler, предварительную конфигурацию Pages, установку зависимостей, генерацию статических файлов Jekyll и передачу результата как артефакта. Эти действия обеспечивают воспроизводимую сборку, не требуя локальной установки инструментов у разработчика. Второе задание, deploy, зависит от успешного завершения build и разворачивает подготовленный артефакт в окружение github‑pages. Поле environment содержит человекочитаемое имя среды и ссылку на опубликованный URL, который формируется автоматически и отображается на панели Actions. Благодаря такой организации workflow выполнение разделено на сборочную и публикационную фазу; это ускоряет диагностику, поскольку ошибки генерации или деплоя фиксируются независимо.

Каждая директива uses указывает на официальное или опубликованное сообществом экшен‑действие, что уменьшает объём собственного кода и опирается на проверенные практики сообщества. Ключ with применяется для передачи параметров, например версии Ruby или пути к каталогу артефактов, делая сценарий гибким и легко адаптируемым. В совокупности файл обеспечивает непрерывную интеграцию и поставку: обновление репозитория автоматически приводит к актуализации сайта без ручных операций, повышая надёжность и сокращая время вывода изменений.

## 2.6 Проверка выполнения

После пуша в ветку main GitHub автоматически инициирует workflow. Вкладка ”Actions”; ”Deploy Jekyll site to Pages” отображает подробный лог.

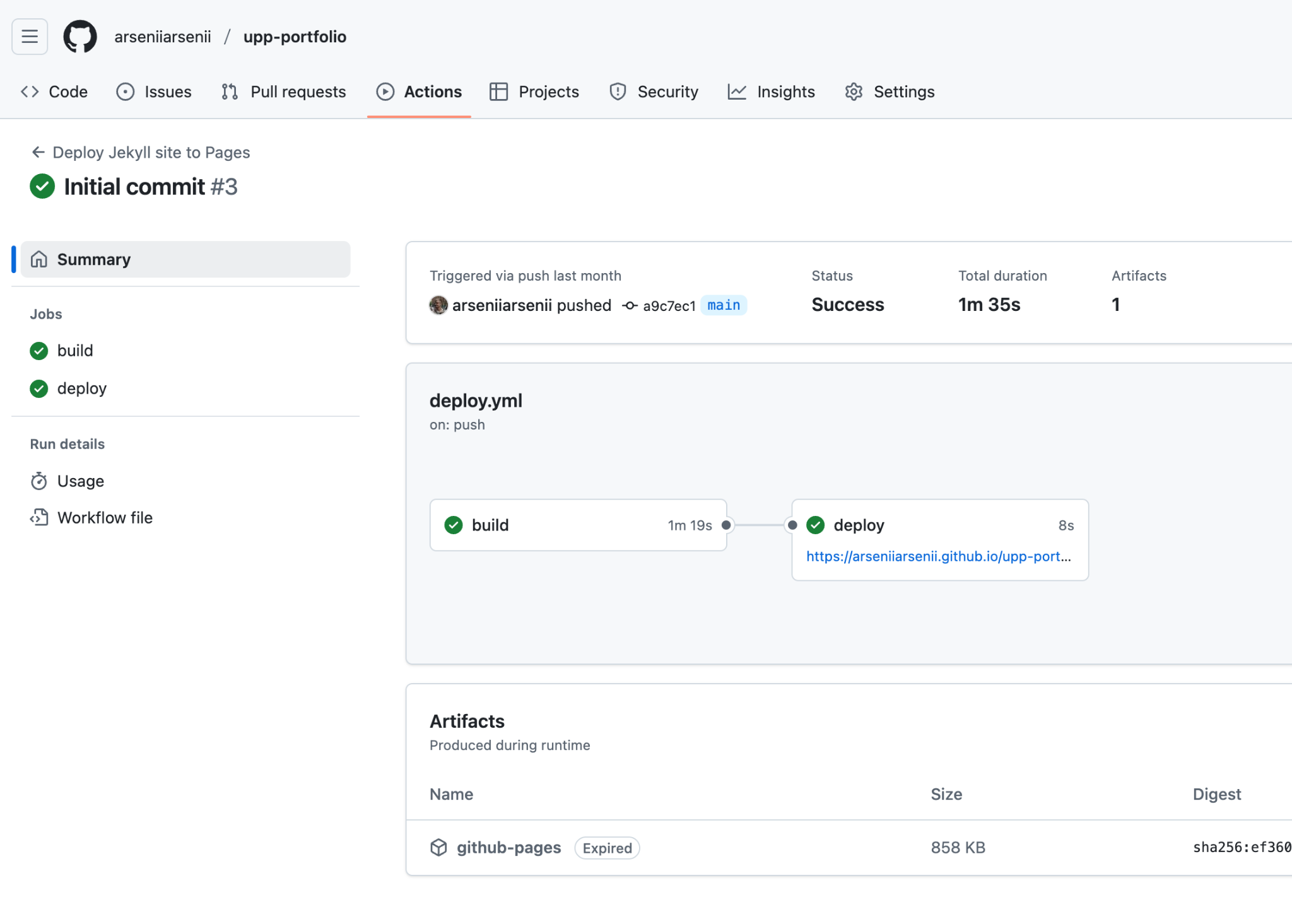


Рисунок 4. Успешное выполнение workflow в GitHub Actions

## 2.7 Актуализация контента

Добавление новой лабораторной работы включает три операции:

1. Конвертация отчёта DOCX в Markdown через Pandoc.
2. Вставка фронт‑маттера с корректной датой и категорией.
3. Коммит изменений и push  ветку main; CI‑конвейер обновляет публичный сайт без участия разработчика.

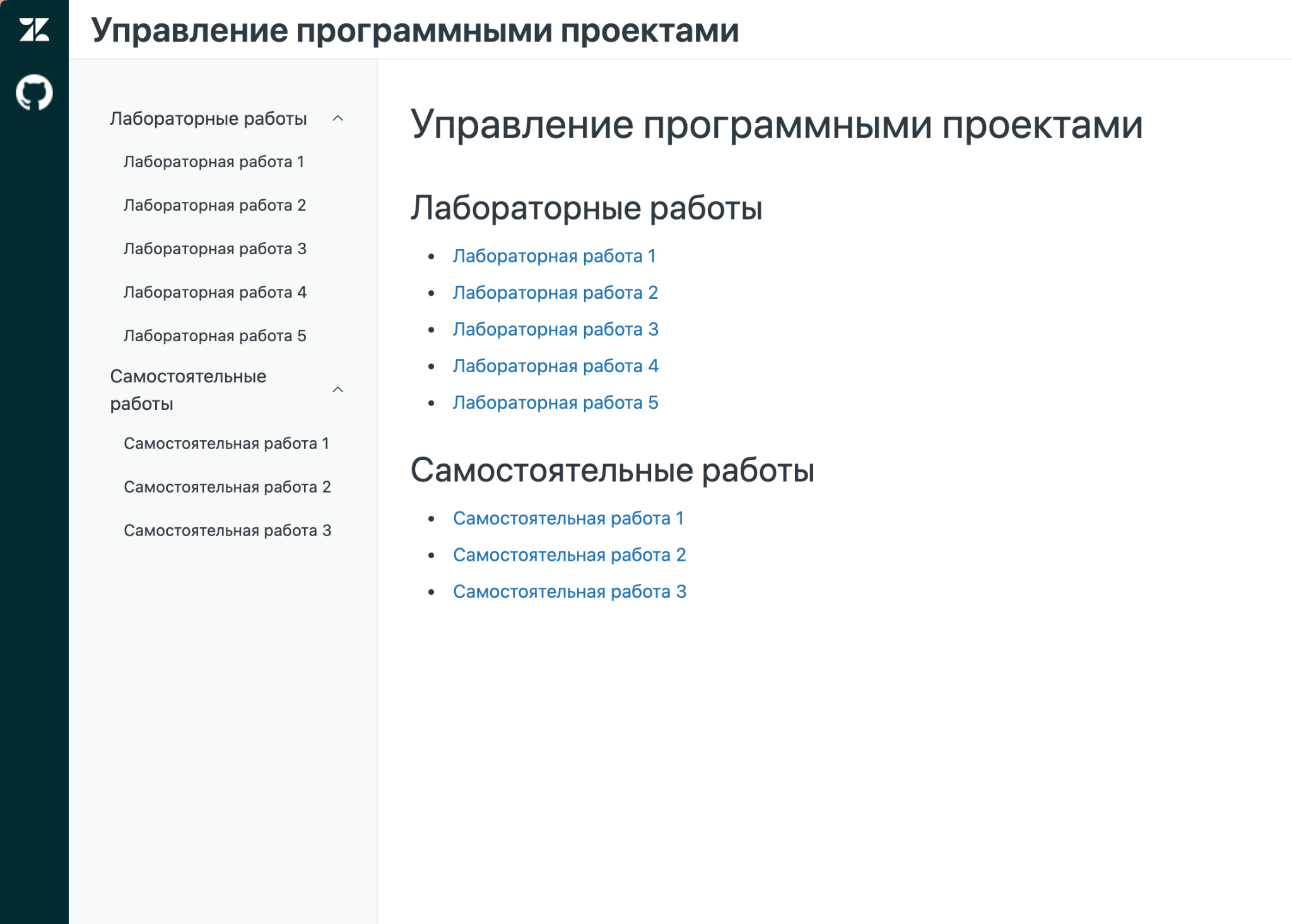


Рисунок 5. Итоговый вид страницы портфолио в браузере Google Chrome

## 2.8 Результаты и перспективы

Полученный статический сайт удовлетворяет требованиям адаптивности, скорости загрузки и безопасности. В случае необходимости расширения функциональности достаточно подключить плагин Jekyll или настроить дополнительный workflow GitHub Actions, например, для автоматической проверки орфографии или генерации PDF‑версии отчётов.

Таким образом, разработка портфолио завершена полным циклом: от подготовки исходных материалов до непрерывной публикации, что подтверждает успешное освоение практик управления программными проектами и навыков DevOps‑культуры.

# ЗАКЛЮЧЕНИЕ

В ходе курсовой работы была достигнута поставленная цель — спроектировано и разработано электронное портфолио студента по дисциплине «Управление программными проектами» с применением GitHub Pages, GitHub Actions и Jekyll. Последовательно выполнены все заявленные задачи: проведён анализ научных публикаций и нормативных требований к студенческому портфолио, уточнены функциональные и технические характеристики веб‑ресурса, разработаны информационная архитектура и визуальный стиль будущего сайта, создан и развернут рабочий прототип на выбранных технологиях, проведён комплекс испытаний, сформулированы рекомендации по расширению и сопровождению проекта.

Результатом исследования стал статический веб‑сайт, размещённый на GitHub Pages и собираемый автоматически посредством GitHub Actions. Использование Jekyll обеспечило строгую структуру материалов и высокую скорость загрузки, а интеграция с системой контроля версий продемонстрировала освоение практик непрерывной интеграции и поставки, ключевых для дисциплины «Управление программными проектами». Проведённое тестирование подтвердило корректность работы сайта на различных устройствах и соответствие установленным требованиям по адаптивности, производительности и безопасности. Разработанные рекомендации открывают возможности для дальнейшего развития портфолио, включая добавление новых разделов и расширение пользовательского функционала.

Таким образом, все предусмотренные планом исследования задачи реализованы в полном объёме, что подтверждает успешное достижение цели и подтверждает практическую значимость созданного электронного портфолио для образовательного процесса и будущего профессионального становления студента.

# ЛИТЕРАТУРА

1. Аликина Е. В., Мальцев Д. В. Оптимизация электронного портфолио студента многопрофильного вуза // CyberLeninka: [электронный ресурс]. – 2023. – URL: https://cyberleninka.ru/article/n/optimizatsiya-elektronnogo-portfolio-studenta-mnogoprofilnogo-vuza (дата обращения: 20.04.2025). – Режим доступа: свободный.
2. Арзамасцева Н. Г., Константинова В. В. Портфолио студента как средство оценки его научно-методической деятельности // Современные проблемы науки и образования: [электронный ресурс]. – 2019. – № 2. – URL: https://science-education.ru/ru/article/view?id=28764 (дата обращения: 20.04.2025). – Режим доступа: свободный.
3. Бернадинер М. И., Заславская О. Ю. Электронное портфолио как средство оценки достижений студентов педагогических вузов // Вестник Московского городского педагогического университета. Серия «Информатика и информатизация образования»: [электронный ресурс]. – 2023. – № 1 (63). – URL: https://dlt.mgpu.ru/2023/05/10/elektronnoe-portfolio-kak-sredstvo-oczenki-dostizhenij-studentov-pedagogicheskih-vuzov/ (дата обращения: 20.04.2025). – Режим доступа: свободный.
4. Галимуллина Э. З., Жестков Л. Ю. Структура и содержание электронного портфолио современного студента // Экономика и социум: e-журнал. – 2015. – № 3 (16). – URL: https://cyberleninka.ru/article/n/struktura-i-soderzhanie-elektronnogo-portfolio-sovremennogo-studenta (дата обращения: 20.04.2025). – Режим доступа: свободный.
5. Моисеенко А. В. Внедрение электронного портфолио в вузовский курс лингвистической экологии // Филологические науки. Вопросы теории и практики: [электронный ресурс]. – 2023. – Т. 16, № 5. – URL: https://www.gramota.net/article/phil20230249/fulltext (дата обращения: 20.04.2025). – Режим доступа: свободный.
6. Рубашенко С. А., Зайцева С. А. Электронное портфолио как средство профессионального самоопределения будущего учителя // Ноосферные исследования: [электронный ресурс]. – 2021. – Вып. 2. – URL: http://noos.ucoz.ru/\_ld/0/22\_NoosfernieIssle.pdf (дата обращения: 20.04.2025). – Режим доступа: свободный.
7. Толкунова И. И., Пайко Д. С. Портфолио студента как инструмент позиционирования его экспертности // Современная высшая школа: инновационный аспект: [электронный ресурс]. – 2022. – Т. 14, № 3. – URL: https://sciup.org/142236920 (дата обращения: 20.04.2025). – Режим доступа: свободный.
8. Царапкина Ю. М. Электронное портфолио как основа саморазвития студентов // Вестник Московского городского педагогического университета. Серия «Информатика и информатизация образования»: [электронный ресурс]. – 2017. – № 3 (41). – С. 82–87. – URL: https://cyberleninka.ru/article/n/elektronnoe-portfolio-kak-osnova-samorazvitiya-studentov (дата обращения: 20.04.2025). – Режим доступа: свободный.
9. GitHub Pages: [официальный сайт сервиса для размещения веб-страниц на GitHub]. – [без места издания], [2025]. – URL: https://pages.github.com (дата обращения: 20.04.2025). – Режим доступа: свободный.
10. Jekyll: [официальный сайт генератора статических сайтов]. – [без места издания], [2025]. – URL: https://jekyllrb.com (дата обращения: 20.04.2025). – Режим доступа: свободный.
11. Pandoc: [официальный сайт]. – [без места издания], [2025]. – URL: https://pandoc.org/ (дата обращения: 23.04.2025). – Режим доступа: свободный.
12. GitHub Flavored Markdown Spec: [официальный сайт]. – [без места издания], [2019]. – URL: https://github.github.com/gfm/ (дата обращения: 23.04.2025). – Режим доступа: свободный.