



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

«Создание электронного портфолио по дисциплине "Программирование"»

Обучающегося 4 курса
Шардта Максима Александрович

Научный руководитель:
кандидат физико-математических наук,
доцент кафедры ИТиЭО
Жуков Николай Николаевич

Санкт-Петербург
2024

Оглавление

Оглавление	2
Введение	3
1. Теоретическая часть	5
1.1 Понятие электронного портфолио	5
1.2 Подходы к созданию электронного портфолио	6
1.3 Статический генератор сайтов Hugo	7
1.4 Размещение сайта на GitHub Pages	7
2. Практическая часть	9
2.1 Определение структура проекта	9
2.2 Выбор темы	9
2.3 Конфигурация Hugo	10
2.3 Наполнение контентом	11
2.4 Размещение на GitHub Pages	13
Заключение	15
Литература	16

Введение

Современные образовательные и профессиональные процессы требуют от студентов и специалистов эффективного представления своих навыков и достижений. В эпоху цифровой трансформации традиционные резюме дополняются электронными портфолио, которые становятся важным инструментом для демонстрации личных проектов, компетенций и уровня подготовки. Электронное портфолио позволяет структурировать и визуализировать результаты деятельности, делая их доступными для широкой аудитории, включая потенциальных работодателей, наставников и образовательных учреждений.

Актуальность данной темы связана с возрастающими требованиями к прозрачности и доступности профессиональных данных. Это обусловлено не только удобством анализа, но и возможностью оперативно оценить уровень компетенций соискателя. В условиях высокой конкуренции наличие качественного электронного портфолио становится важным преимуществом для успешного трудоустройства. [1]

Цифровизация образовательных процессов также способствует росту популярности портфолио среди студентов. Многие университеты интегрируют их создание в учебные программы, мотивируя учащихся к развитию навыков самопрезентации. В этом контексте создание электронного портфолио по дисциплине "Программирование" отвечает потребностям как образовательной, так и профессиональной среды. [2], [4]

Целью данной работы является создание электронного портфолио, которое демонстрирует навыки программирования автора и служит инструментом для профессионального и образовательного роста. Для её достижения поставлены следующие задачи:

- Изучение концепции электронного портфолио, его значимости и применения в образовательной и профессиональной среде.
- Анализ инструментов для создания электронного портфолио, включая статические генераторы сайтов и платформы для хостинга.
- Проектирование структуры портфолио, настройка и разработка сайта. Наполнение сайта контентом, включающим описания лабораторных работ, ссылки на отчёты и исходный код, а также оформление главной страницы и других разделов.

- Размещение готового сайта с использованием автоматизированных процессов сборки и деплоя.

Объект исследования — процесс создания электронного портфолио. Предмет исследования — применение статического генератора сайтов Hugo и платформы GitHub Pages для разработки и размещения портфолио.

Практическая значимость работы заключается в создании реального инструмента для демонстрации профессиональных навыков. Этот инструмент может использоваться как для образовательных целей, так и для карьерного роста. Работа включает в себя описание ключевых этапов разработки, что делает её полезной для студентов, начинающих изучать веб-технологии.

Таким образом, создание электронного портфолио не только способствует развитию профессиональных навыков, но и помогает выделиться на фоне конкурентов, демонстрируя готовность к работе в современных цифровых условиях. Данная курсовая работа предоставляет уникальную возможность изучить этот процесс и получить ценные практические результаты.

1. Теоретическая часть

1.1 Понятие электронного портфолио

Электронное портфолио представляет собой современный инструмент, позволяющий систематизировать информацию о профессиональных достижениях, навыках и опыте человека. Это многофункциональный цифровой ресурс, который используется для демонстрации компетенций, хранения проектов и достижения образовательных или профессиональных целей. Его уникальность заключается в возможности совмещения текстовой, визуальной и мультимедийной информации для наиболее полного и убедительного представления о пользователе. [1] [3]

Значимость электронного портфолио обусловлена его универсальностью и гибкостью. Оно становится важным элементом в образовании, позволяя студентам не только структурировать свои достижения, но и визуализировать их в удобной и эстетичной форме. Преподаватели могут использовать портфолио для мониторинга прогресса учащихся и оценки их компетенций. В профессиональной среде портфолио играет роль доказательства квалификации, предоставляя потенциальным работодателям наглядные примеры выполненных задач, проектов и навыков.

Качественное электронное портфолио должно соответствовать ряду требований, чтобы быть эффективным инструментом. Одним из таких требований является адаптивность. Портфолио должно корректно отображаться на различных устройствах, включая настольные компьютеры, планшеты и смартфоны. Это особенно важно в условиях повсеместного использования мобильных технологий.

Простота обновления контента — еще одно важное требование. Пользователь должен иметь возможность легко добавлять новые проекты, обновлять достижения и редактировать имеющиеся записи. Это позволяет портфолио оставаться актуальным, что особенно ценно для быстро развивающихся сфер, таких как IT или дизайн.

Эстетическая привлекательность и удобство навигации также играют ключевую роль. Портфолио должно быть оформлено в современном стиле, соответствующем профессиональным стандартам, и обладать интуитивно понятной структурой. Навигация по

разделам должна быть удобной и логичной, чтобы пользователь мог быстро найти интересующую информацию.

Безопасность данных является важным аспектом в разработке электронного портфолио. Оно должно соответствовать нормативным требованиям по защите персональных данных, а также предусматривать возможность ограниченного доступа к конфиденциальной информации. Таким образом, электронное портфолио представляет собой важный инструмент, сочетающий в себе функциональность, эстетичность и безопасность.

1.2 Подходы к созданию электронного портфолио

Создание электронного портфолио может быть реализовано с использованием различных подходов, каждый из которых обладает своими преимуществами и недостатками. Одним из наиболее традиционных методов является использование динамических систем управления контентом (CMS), таких как WordPress или Joomla. Эти платформы предоставляют широкий функционал, включая возможность добавления интерактивных элементов, интеграции с социальными сетями и гибкую настройку дизайна. Однако их использование требует постоянного обновления серверного ПО и может быть сопряжено с рисками безопасности из-за наличия динамических компонентов.

Альтернативным подходом является создание статических HTML-страниц вручную. Этот метод подходит для небольших проектов, где не требуется частое обновление контента. Прямое редактирование HTML-кода позволяет максимально контролировать структуру и оформление сайта, но требует знаний в области веб-разработки и может быть трудоемким.

Ещё одним современным вариантом является использование облачных сервисов, таких как Wix или Squarespace, которые предлагают инструменты для визуального создания сайтов. Эти платформы удобны для начинающих пользователей, так как не требуют навыков программирования. Однако их функциональность ограничена предложенными шаблонами, а также отсутствием полного контроля над кодом и возможностями кастомизации.

Наиболее сбалансированным подходом для создания портфолио является использование статических генераторов сайтов, таких как Hugo, Jekyll или Nexo. Эти инструменты позволяют сочетать преимущества ручного редактирования кода с

автоматизацией процесса генерации страниц. Они обеспечивают высокую скорость работы, безопасность и простоту интеграции с платформами для хостинга.

1.3 Статический генератор сайтов Hugo

Hugo является одним из наиболее популярных статических генераторов сайтов благодаря своей высокой производительности и универсальности. Он способен генерировать тысячи страниц за считанные секунды, что делает его оптимальным выбором для крупных проектов и сложных структур. Этот инструмент поддерживает разметку Markdown, что упрощает написание контента и исключает необходимость использования сложных языков программирования.

Особенностью Hugo является богатая коллекция шаблонов и тем, которые можно легко адаптировать под индивидуальные потребности. Это особенно важно для создания портфолио, так как позволяет разработчику сосредоточиться на содержательной части, а не на дизайне. Инструмент также поддерживает мультиязычность, что делает его подходящим для пользователей, работающих в международной среде.

Ещё одним преимуществом Hugo является его кроссплатформенность. Он поддерживается на операционных системах Windows, macOS и Linux, что делает его доступным для широкой аудитории разработчиков. Интеграция с GitHub Pages и другими платформами для хостинга упрощает процесс публикации и обновления сайтов, обеспечивая стабильность и безопасность.

Hugo выделяется среди других статических генераторов благодаря своей гибкости и простоте использования. Этот инструмент идеально подходит для создания электронного портфолио, отвечающего современным требованиям по производительности, безопасности и эстетичности.

1.4 Размещение сайта на GitHub Pages

GitHub Pages представляет собой платформу для размещения статических сайтов, которая позволяет разработчикам публиковать свои проекты непосредственно из репозитория GitHub. Этот инструмент обладает рядом преимуществ, включая бесплатное использование, интеграцию с системой контроля версий Git и автоматизацию процессов

деплой. Благодаря этим особенностям GitHub Pages стал одним из самых популярных решений для размещения электронных портфолио.

Процесс деплоя на GitHub Pages включает создание репозитория, загрузку в него сгенерированных страниц и активацию функции публикации через настройки репозитория. После этого сайт становится доступным по уникальному URL, что обеспечивает удобство его использования для профессиональной самопрезентации.

Кроме стандартных функций, GitHub Pages поддерживает привязку пользовательских доменов, что позволяет улучшить внешний вид URL и повысить уровень профессионализма сайта. Также платформа автоматически настраивает HTTPS-сертификат для обеспечения безопасности данных.

Аналогами GitHub Pages являются Netlify и Vercel. Netlify предлагает широкий спектр возможностей для автоматизации, включая поддержку CI/CD процессов и интеграцию с системами управления контентом. Vercel, в свою очередь, оптимизирован для работы с современными фреймворками, такими как Next.js, и обеспечивает высокую производительность для сложных веб-приложений. Однако GitHub Pages остаётся предпочтительным выбором для студентов и начинающих разработчиков благодаря своей простоте и тесной интеграции с экосистемой GitHub.

GitHub Pages является надёжным и доступным инструментом для деплоя электронного портфолио. Его использование позволяет создавать профессионально выглядящие сайты, обеспечивая при этом лёгкость управления и высокий уровень безопасности.

2. Практическая часть

2.1 Определение структуры проекта

Создание электронного портфолио начинается с определения его структуры, которая должна быть логичной, удобной и отвечать целям проекта. Основной задачей на этом этапе является формирование архитектуры сайта, включающей такие разделы, как информация о владельце, проекты, достижения, контактные данные и дополнительные элементы, отражающие профессиональную деятельность. Для этого необходимо учитывать специфику аудитории, которая будет использовать данный ресурс, будь то преподаватели, работодатели или коллеги.

Каждый раздел должен иметь чёткую цель и быть оформлен таким образом, чтобы его содержание было легко воспринимаемым. Например, раздел с проектами может включать краткие описания выполненных задач, ссылки на исходный код или демонстрации, а также пояснения, какие навыки были использованы для их реализации. Раздел о владельце должен содержать профессиональное резюме и ключевые компетенции, подчеркивающие квалификацию пользователя.

Структура проекта должна быть продумана таким образом, чтобы в неё можно было легко добавлять новые разделы или элементы. Это особенно важно для тех, кто планирует регулярно обновлять своё портфолио. С помощью заранее продуманной структуры можно избежать хаотичного размещения информации и обеспечить логичное представление всех данных.

2.2 Выбор темы

Выбор темы для портфолио играет ключевую роль в создании визуально привлекательного и функционального сайта. На данном этапе важно учитывать не только эстетические аспекты, но и удобство использования, адаптивность и возможность кастомизации. Большинство статических генераторов сайтов, таких как Hugo, предоставляют широкий выбор готовых шаблонов, которые можно адаптировать под индивидуальные нужды.

Для реализации данного проекта была выбрана тема "no-style-please" по нескольким причинам. Во-первых, её минималистичный дизайн позволяет сосредоточить внимание

пользователя на содержании, а не на отвлекающих визуальных элементах. Это особенно важно для портфолио, где ключевым является представление работ и достижений. Во-вторых, тема обладает высокой адаптивностью и корректно отображается на любых устройствах, включая смартфоны, планшеты и компьютеры. Такое решение обеспечивает удобство для широкой аудитории.

2.3 Конфигурация Hugo

После определения структуры и выбора шаблона можно приступить к конфигурации сайта. Этот этап включает настройку визуальных элементов и интеграцию всех необходимых компонентов. Hugo, как один из самых популярных статических генераторов сайтов, позволяет максимально упростить этот процесс благодаря своей производительности и поддержке Markdown для написания текстов. [5]

Все проекты Hugo должны придерживаться строгой структуры. Каталог archetypes содержит шаблоны для создания новых страниц. Например, если часто создаются страницы для лабораторных работ, можно задать шаблон, чтобы каждый раз не дублировать структуру вручную. content хранит весь контент сайта. Здесь создаются папки и файлы, соответствующие структуре портфолио. Файлы пишутся в формате Markdown, который Hugo преобразует в HTML-страницы. data используется для хранения файлов данных в форматах YAML, TOML или JSON. Эти данные могут быть использованы для динамической генерации контента, например, для создания списков работ. layouts содержит шаблоны страниц. Здесь задаются структура и дизайн различных типов контента. Например, шаблон для лабораторных работ может находиться в layouts/_default/single.html. static хранит статические файлы (изображения, CSS, JavaScript, шрифты и т. д.). Все файлы из этой папки копируются в корневую директорию сгенерированного сайта. themes содержит темы оформления. Hugo поддерживает использование сторонних тем, которые можно скачать или создать самостоятельно. Здесь будет расположена тема, выбранная ранее, как git submodule. Файл config.toml (или config.yaml) является центральным местом настройки проекта. В нем указываются ключевые параметры, такие как: название сайта (title), базовый URL (baseUrl), используемая тема (theme), формат даты (date_format), настройки меню и SEO.

Для настройки темы и отображения сайта необходимо настроить params.theme_config. Данная конфигурация описывает параметры темы сайта, созданного с помощью Hugo, и

задаёт его внешний вид и функциональность. Параметр `appearance = "light"` определяет использование светлой темы оформления, что делает сайт ярким и удобным для чтения при дневном освещении. Кнопка или ссылка для возврата на главную страницу будет отображаться с текстом "Домой", как указано в параметре `back_home_text = "Домой"`. Это обеспечивает удобную навигацию для пользователей, позволяя им быстро вернуться на главную страницу из любого раздела.

Формат отображения даты задаётся через параметр `date_format`. Поскольку он оставлен пустым, будет использоваться формат, заданный по умолчанию в теме. Это позволяет адаптировать отображение дат к региональным стандартам, если такая необходимость возникнет. Настройка `isListGroupByDate = false` отключает группировку контента по датам, то есть все материалы будут выводиться в непрерывном списке без разделения на хронологические категории. Это подходит для сайтов, где даты не играют ключевой роли, например, для портфолио. Наконец, параметр `isShowFooter = false` отключает отображение нижнего колонтитула (footer), что придаёт сайту минималистичный вид, лишённый лишних элементов вроде дополнительных ссылок или контактной информации.

Эта конфигурация создаёт лёгкий, функциональный и минималистичный сайт с акцентом на контент, а не на вспомогательные элементы.

2.3 Наполнение контентом

Процесс наполнения сайта контентом в Hugo начинается с подготовки всех необходимых материалов. Сюда входит подготовка описаний выполненных работ, сбор отчетов и исходного кода для лабораторных работ. После подготовки материалов создаются файлы контента, которые размещаются в папке `content`. Для удобства навигации файлы группируются по разделам. Портфолио разделено на отдельные папки для каждого курса, а в них размещено описание лабораторных работ. Каждый файл в формате становится отдельной страницей на сайте.

Главная страница портфолио представляет собой ключевую точку навигации, которая направляет пользователей к различным разделам сайта. Она начинается с краткой информации об авторе. Это вводное описание сразу даёт посетителям общее представление о создателе портфолио. Далее на странице представлены ссылки на разделы, посвящённые выполненным работам за каждый год обучения, результаты представлены на рисунке 1.

Портфолио

Портфолио работ по дисциплине “Программирование”.

Автор: Шардт М.А., студент 4го курса РГПУ им. А. И. Герцена.

Работы по курсам

- [1 Курс](#)
- [2 Курс](#)
- [3 Курс](#)
- [4 Курс](#)

Рисунок 1. Структура главной страницы сайта.

Каждый раздел организуется как отдельная папка с файлами. Папка posts содержит разделенные по годам разделы портфолио. Каждый файл становится отдельной страницей на сайте, где описываются лабораторные работы, выполненные за указанный год. Для удобства создания списков будет использоваться следующий шаблон, указанный в листинге 1.

```
+++  
title = "Лабораторная работа №1"  
date = "2006-01-20"  
draft = false  
+++
```

Описание лабораторной работы

[\[Отчет\]](#) ([ссылка на отчет](#)) | [\[Исходный код\]](#) ([ссылка на код](#))

Листинг 1. Шаблон создания лабораторной работы.

Таким образом раздел, содержащий работы по каждому семестру, будет выглядеть как показано на рисунке 2.

[Домой](#)

1 Курс

2 Семестр

Лабораторная Работа №1

Математические операции. Переменные и их типы. Операторы. Циклы. Простые условные конструкции. Основы работы со статическими массивами

[Отчет](#) | [Исходный Код](#)

Лабораторная Работа №2

Указатели, арифметика указателей.

[Отчет](#) | [Исходный Код](#)

Рисунок 2. Структура раздела сайта.

2.4 Размещение на GitHub Pages

Для размещения сайта на GitHub Pages существует два основных метода: создание отдельной ветки, в которой будут размещаться статичные файлы сайта, и автоматическое развертывание с помощью GitHub Actions. В данной работе будет продемонстрирован второй метод. Настройка автоматического развертывания сайта начинается с создания workflow в формате yaml в директории .github/workflows. В нем описывается процесс настройки, сборки и развертывания приложения. Полная конфигурация указана в приложении 1, далее же будут рассмотрены основные моменты его работы.

Процесс развертывания начинается с настройки событий, при которых workflow будет активироваться. В частности, он запускается при каждом push в ветки main, а также может быть запущен вручную через вкладку Actions на GitHub, благодаря параметру workflow_dispatch. Это позволяет гибко управлять процессом развертывания и тестирования. В разделе прав доступа настраиваются разрешения для токена GitHub (GITHUB_TOKEN), который будет использоваться в ходе выполнения. Устанавливаются права на чтение содержимого репозитория, запись на GitHub Pages и использование токенов идентификации.

Эти настройки необходимы для правильного функционирования и безопасности процесса деплоя.

В файле предусмотрено ограничение на параллельное выполнение нескольких запусков. С помощью настройки `concurrency` все задачи, связанные с деплоем на GitHub Pages, выполняются в одной группе, при этом новые запуски не отменяют те, что находятся в процессе. Это особенно важно для обеспечения корректного завершения текущих операций перед началом новых. Далее идет настройка среды выполнения с использованием Bash, что является стандартом для большинства команд в данном workflow. Для выполнения самого процесса развертывания определены два основных этапа — сборка и деплой.

На этапе сборки происходит установка необходимых инструментов, таких как Hugo и Dart Sass. Затем выполняется клонирование репозитория, настройка GitHub Pages и установка зависимостей, если они требуются для проекта. В процессе сборки используется Hugo с флагом минимизации, а также задается правильный базовый URL для сайта. По завершении этого этапа все результаты сохраняются в артефакты, которые будут использованы на следующем шаге.

На этапе деплоя результат сборки загружается на GitHub Pages с помощью действия `actions/deploy-pages`. Этот шаг зависит от успешного завершения предыдущего этапа сборки, что обеспечивает целостность процесса. Таким образом, весь workflow автоматизирует процесс создания и развертывания статического сайта, минимизируя необходимость в ручных вмешательствах и ускоряя цикл разработки и публикации.

Этот workflow позволяет эффективно автоматизировать процесс разработки и публикации статического сайта. Благодаря интеграции с GitHub Actions, процесс сборки и деплоя становится максимально простым и быстрым, что значительно ускоряет цикл разработки и снижает вероятность ошибок, связанных с ручными действиями.

Заключение

В ходе выполнения данной курсовой работы была достигнута основная цель — разработано электронное портфолио, демонстрирующее навыки программирования автора. Работа охватила как теоретические аспекты, так и практическую реализацию, что позволило обеспечить комплексное понимание процесса создания портфолио. Получившееся портфолио доступно по URL: <https://mxschardt.github.io/uni-portfolio-site/>.

На этапе теоретического исследования изучено понятие электронного портфолио, его значимость в образовательной и профессиональной среде, а также инструменты, применяемые для его разработки. В частности, рассмотрены возможности статического генератора сайтов Hugo и платформы GitHub Pages. Эти технологии были выбраны за их высокую производительность, простоту использования и соответствие современным требованиям к разработке сайтов.

Практическая часть включала проектирование структуры портфолио, настройку шаблонов и параметров сайта, а также создание контента. Логичная организация разделов позволила эффективно представить результаты учебной деятельности, включая лабораторные работы, проекты и достижения. Сайт был наполнен текстовыми и мультимедийными материалами, обеспечивающими его информативность и визуальную привлекательность. Размещение готового проекта на платформе GitHub Pages с использованием автоматизированных процессов сборки и деплоя продемонстрировало возможность быстрой и удобной публикации сайта в интернете.

Практическая значимость работы заключается в создании инструмента, который может использоваться для образовательных целей и карьерного роста. Электронное портфолио способствует развитию навыков самопрезентации, позволяет эффективно демонстрировать компетенции и выделяться среди конкурентов.

Подводя итоги, можно утверждать, что выполненная работа достигла поставленных целей, а её результаты имеют как образовательную, так и профессиональную ценность. Созданное портфолио представляет собой практический пример применения современных технологий в контексте цифровизации образовательных и профессиональных процессов.

Литература

1. Павленко И. И. Электронный портфолио в вузе // Прикладная информатика. 2008. №6. URL: <https://cyberleninka.ru/article/n/elektronnyy-portfolio-v-vuze> (дата обращения: 18.12.2024).
2. Ильина Елена Александровна, Гладышева Мария Михайловна, Дьяконов Никита Александрович, Арефьева Дарья Яковлевна, Кольба Юрий Юрьевич Анализ учебной деятельности в электронном портфолио // АНИ: педагогика и психология. 2019. №1 (26). URL: <https://cyberleninka.ru/article/n/analiz-uchebnoy-deyatelnosti-v-elektronnom-portfolio> (дата обращения: 20.12.2024).
3. Нам Т. А. Электронное портфолио студентов как инновационная технология в высшем профессиональном образовании // Гуманитарные и социальные науки. 2020. №2. URL: <https://cyberleninka.ru/article/n/elektronnoe-portfolio-studentov-kak-innovatsionnaya-tehnologiya-v-vysshem-professionalnom-obrazovanii> (дата обращения: 20.12.2024).
4. Марголис А.А., Сафронова М.А., Дробязько А.А., Куравский Л.С., Голованова И.А., Поминов Д.А. Электронное портфолио как средство формирования рефлексивного отношения студентов к учебе [Электронный ресурс] // Психолого-педагогические исследования. 2021. Том 13. № 2. С. 3–16. DOI: 10.17759/psyedu.2021130201 URL: https://psyjournals.ru/journals/psyedu/archive/2021_n2/Margolis_et_al (дата обращения: 20.12.2024).
5. Документация Hugo // GoHugo URL: <https://gohugo.io/documentation/> (дата обращения: 18.12.2024).