



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 веб-портфолио  
по дисциплине "Программирование"»

Обучающегося 4 курса  
Стецук Максима Николаевича

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

Санкт-Петербург  
2024

# ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	3
Теоретический блок.....	5
1.1 Веб-портфолио. Требования к веб-портфолио студента .....	5
1.2 Инструменты для разработки веб-портфолио .....	8
1.2.1 Инструменты для создания сайтов.....	8
1.2.1.1 Платформы для создания сайтов .....	8
1.2.1.2 Базовые инструменты веб-разработки .....	9
1.2.1.3 Фреймворки и библиотеки .....	10
1.2.1.4 Генераторы статических сайтов.....	11
1.2.2 Инструменты оформления программного кода .....	11
Практический блок.....	13
2.1 Реализация веб-портфолио на основе Bootstrap .....	13
2.1.1 Подготовка репозитория для публикации веб-портфолио.....	13
2.1.2 Установка и настройка Lektor.....	14
2.1.3 Подключение Bootstrap и настройка шаблонов для реализации структуры .....	15
2.1.4 Наполнение веб-портфолио работами по различным дисциплинам .....	17
2.1.5 Публикация сайта на GitHub.....	18
2.2 Публикация работ по программированию и машинному обучению.....	19
ЗАКЛЮЧЕНИЕ.....	22
ПРИЛОЖЕНИЕ 1 .....	24
ПРИЛОЖЕНИЕ 2 .....	26

## ВВЕДЕНИЕ

В настоящее время сфера информационных технологий стремительно развивается, в результате чего все большее количество компаний ищет специалистов, способных работать в этой области. Однако стремительный рост и развитие инструментов, необходимых в сфере информационных технологий, приводят к необходимости тщательной оценки кандидатов. Таким образом, каждая компания стремится найти специалиста, который будет обладать именно теми навыками, которые необходимы для развития конкретного продукта или выполнения определенного проекта.

Кроме того, все больше людей стремится освоить минимальный набор навыков, которые они смогут указать в резюме, чтобы найти работу в сфере информационных технологий. Этот фактор приводит к увеличению числа младших разработчиков, аналитиков данных, инженеров по тестированию и дизайнеров на рынке труда. Это создает серьезные проблемы при поиске первой работы для начинающих специалистов, ведь кроме них на конкретную позицию может претендовать огромное количество людей с аналогичными навыками, указанными в резюме.

В связи с этим, помимо написания резюме, одной из главных задач любого специалиста в сфере информационных технологий является создание личного портфолио, которое служит основным способом презентации собственных профессиональных навыков и достижений, таких как учебные проекты или работы, выполненные на заказ. Такой способ самопрезентации позволяет значительно повысить ценность специалиста на фоне других соискателей.

Личное портфолио — важный инструмент для любого специалиста, однако основная сложность в его создании возникает из-за необходимости четкой структуризации и качественного представления собственных работ. Даже хорошо выполненная работа, представленная без должного качества, не будет восприниматься доверительно потенциальным работодателем.

Для студентов, обучающихся по направлениям, связанным с информационными технологиями, и стремящихся стать разработчиками, наиболее эффективным способом представления собственных работ является их публикация в виде веб-портфолио. В настоящее время для создания такого портфолио существует множество инструментов.

Актуальность данной работы заключается в необходимости качественной самопрезентации собственных навыков программирования каждым начинающим разработчиком, включая выпускников вузов, которые хотят заниматься разработкой как своей профессиональной деятельностью.

**Цель работы:** Исследование инструментов и разработка личного веб-портфолио студента по дисциплине "Программирование".

**Задачи работы:**

- Исследование современных способов и инструментов для создания веб-портфолио;
- Анализ требований и подходов к публикации программных продуктов;
- Разработка структуры собственного веб-портфолио и его реализация с помощью выбранных инструментов;
- Подготовка и публикация работ по программированию в личном портфолио.

**Объект исследования:** Процесс разработки и публикации персонального веб-портфолио студента по программированию.

**Предмет исследования:** Современные способы и инструменты для разработки статических сайтов и публикации кода программных продуктов.

В первой главе работы рассматриваются основные теоретические аспекты создания личного портфолио программиста и рекомендации для качественного представления собственных работ по программированию. Также анализируются готовые программные решения и возможности их применения для создания личного портфолио. Во второй главе рассматривается практический процесс реализации веб-портфолио студента по дисциплине "Программирование".

## **Теоретический блок**

### **1.1 Веб-портфолио. Требования к веб-портфолио студента**

Веб-портфолио стало важным инструментом для самопрезентации студентов, обучающихся в области информационных технологий и стремящихся продемонстрировать свои практические навыки и достижения, а также технических специалистов, стремящихся к профессиональному развитию и карьерному росту. В данной части работы мной будут рассмотрены ключевые понятия, связанные с темой данной работы, а также требования к веб-портфолио студента. В частности, данная часть работы содержит описание структуры и особенностей дизайна, которые необходимо учесть при реализации собственного веб-портфолио по программированию.

Веб-портфолио - это интернет-ресурс, который чаще всего представляет собой персональный сайт, на котором студент или технический специалист демонстрирует свои профессиональные достижения, учебные или рабочие проекты, выполненные работы, а также описывает собственные навыки и освоенные во время обучения компетенции. Веб-портфолио является важным инструментом самопрезентации, который позволяет потенциальным работодателям, при трудоустройстве, или научным руководителям, при оценке студента на государственной аттестации, ознакомиться с результатами работы студента в удобной и наглядной форме.

Таким образом, рассматривая веб-портфолио студента, как инструмент, необходимый для успешного завершения обучения и начала построения профессиональной карьеры в сфере информационных технологий, выделяют следующие преимущества, которые оно дает студенту:

- позволяет показать не только теоретическую подготовку, но и практический опыт работы с различными инструментами и технологиями, в том числе с использованными при разработке портфолио;
- помогает систематизировать и выделить из работ, выполненных в ходе обучения те, которые студент хотел бы продемонстрировать преподавателям, потенциальным работодателям или соискателям, которые, как и он стремятся к саморазвитию в конкретной сфере;
- в случае хорошей организации и качественного оформления, в едином стиле, веб-портфолио позволяет привлечь внимание к себе, как к специалисту, а также оставить хорошее первое впечатление.

Портфолио является лицом студента или технического специалиста, поэтому оно обязательно должно содержать в себе информацию об авторе данного интернет-ресурса, в

частности краткую биографию, общую информацию, цели и карьерные устремления, затем, веб-портфолио должно содержать описание навыков и компетенций его автора, которое включает в себя, как soft skills, так и hard skills, которые могут быть представлены как в виде списка на сайте или отдельно реализованного резюме. Основная часть любого портфолио студента технического вуза - его проекты и работы, а также образование и личные достижения, подтвержденные соответствующими документами. А для возможности связи с автором веб-портфолио, оно обязательно должно содержать контактную информацию, в том числе и ссылки на профили на профессиональных платформах по найму сотрудников.

Как было сказано ранее, на качество восприятия влияет не только наполнение портфолио, но и его внешняя составляющая - дизайн и качество пользовательского интерфейса. Дизайн веб-портфолио должен быть чистым, современным и удобным для пользователя. Важно, чтобы структура сайта была интуитивно понятной, а навигация - простой и логичной. При реализации веб-портфолио рекомендуется помнить об адаптивности, которая позволяет сайтам корректно отображаться на различных устройствах.

Исходя из этого можно выделить основные требования к дизайну личного веб-портфолио:

- Интуитивно понятный интерфейс, с знакомыми пользователям элементами управления, например боковым бургер меню;
- Логичная и последовательная структура, способствующая простоте восприятия портфолио;
- Адаптивный дизайн и лаконичный дизайн, корректно отображающийся на различных устройствах и экранах с различным расширением;
- Использование простых анимаций, для ускорения работы веб-портфолио и упрощения его восприятия.

Однако при реализации собственного веб-портфолио можно столкнуться с рядом серьезных трудностей, таких как отсутствие достаточного количества опыта работы с профессиональными инструментами, сложностью выбора инструмента для реализации собственного сайта-портфолио, а также недостатком времени на качественную реализацию всех тех идей, которые могли бы быть реализованы. В данном случае важно помнить о том, что основная цель веб-портфолио - демонстрация работ и достижений студента, а не навыков в веб-разработке, вследствие чего при недостатке времени можно пожертвовать какими-то необычными функциями, в пользу общего качества конечно сайта.

Таким образом, веб-портфолио является важным инструментом для студентов, стремящихся продемонстрировать свои навыки и достижения в сфере информационных технологий. Это не только хороший способ хранения и систематизации собственных работ,

но и способ повышения личной профессиональной значимости, которая важна как начинающим специалистам, так и тем, кто имеет реальный опыт. Однако важно помнить и о том, что веб-портфолио с плохим дизайном и пользовательским интерфейсом может оттолкнуть незнакомых людей, тем самым испортив первое впечатление, учебных руководителей или потенциальных работодателей, о студенте, как о начинающем специалисте.

Подводя итог данной части работы, можно сделать вывод, что грамотно составленное и презентованное портфолио, значительно увеличивает шансы студента, сразу после завершения обучения, на трудоустройство, ведь оно предоставляет работодателю не только информацию об образовании и абстрактных навыках автора, но и подкрепляет их конкретными учебными заданиями, программными проектами и официальными сертификатами, которые способны продемонстрировать его подход к решению задач и реальный уровень профессиональной подготовки.

## **1.2 Инструменты для разработки веб-портфолио**

После изучения основных рекомендаций по созданию веб-портфолио, необходимо определиться с инструментами, которые будут использоваться при разработке веб-портфолио по программированию. Для этого необходимо определить, какие инструменты будут использоваться для разработки самого сайта-портфолио и каким образом будет оформляться код, написанный в ходе выполнения лабораторных работ, связанных с программированием, в частности тех, что связаны с машинным обучением.

### **1.2.1 Инструменты для создания сайтов**

Для создания сайта могут использоваться различные инструменты, начиная от простых конструкторов сайтов до самостоятельной разработки с использованием стандартных веб-технологий, таких как HTML, CSS и JavaScript. В данной части работы мной будут рассмотрены различные подходы и инструменты для создания веб-портфолио, начиная от интуитивно понятных платформ, таких как Google Sites, заканчивая более сложными вариантом, таким как самостоятельная верстка с использованием HTML, CSS, JavaScript и вспомогательных инструментов.

Рассматривая инструменты и методы создания веб-портфолио, можно разделить их на четыре основных группы:

- Платформы для создания сайтов (Google Sites или Wix);
- Базовые инструменты веб-разработки (HTML, CSS, JavaScript);
- Фреймворки и библиотеки (Bootstrap или Tailwind CSS);
- Генераторы статических сайтов (Lektor).

#### **1.2.1.1 Платформы для создания сайтов**

На данный момент наиболее известными и популярными платформами для создания личных сайтов являются Google Sites, Wix и GitHub Pages. Данные онлайн платформы позволяют легко и быстро развернуть собственный сайт.

1. Google Sites - это бесплатный и простой в использовании конструктор сайтов, который предоставляет возможность создавать веб-страницы без необходимости знаний в области программирования или веб-разработки. Google Sites позволяет собирать и структурировать информацию, не тратя время на изучение сложных технологий, а также предоставляет гибкость для настройки внешнего вида и функционала сайта.

Одним из главных преимуществ данного сервиса является его простота и доступность. Все действия выполняются через интуитивно понятный интерфейс с drag-and-



drop функционалом (перетаскивание объектов), что позволяет быстро и легко добавлять различные элементы на страницы. Кроме того, Google Sites интегрируется с другими популярными сервисами Google, такими как Google Docs и Google Drive. Это даёт возможность легко добавлять документы, таблицы, презентации, видео и изображения прямо на страницы сайта.

2. Wix - это популярная платформа для создания сайтов, которая предлагает удобный и мощный конструктор, позволяющий пользователям создавать веб-страницы без необходимости в глубоких знаниях в области программирования. Этот сервис идеально подходит для людей, которые хотят быстро и легко создать собственный сайт, в частности личное веб-портфолио.

Одним из главных достоинств Wix является его универсальность и широкие возможности для кастомизации. Платформа предлагает огромный выбор. Эти шаблоны можно легко адаптировать с помощью drag-and-drop конструктора, что позволяет без особых усилий перемещать элементы на страницах сайта.

Одним из самых привлекательных аспектов Wix является наличие искусственного интеллекта - Wix ADI (Artificial Design Intelligence). Этот инструмент позволяет за несколько минут создать сайт на основе ответов пользователя на несколько простых вопросов.

3. GitHub Pages - это бесплатный сервис для хостинга статических веб-страниц, предоставляемый платформой GitHub. Он подходит для разработчиков, студентов, технических специалистов и всех тех, кто хочет создать сайт, используя минимальные ресурсы и максимально интегрировать его с процессом разработки. GitHub Pages позволяет хостить сайт прямо из репозитория на GitHub, что упрощает обновление и развертывание контента, что значительно упрощает процесс актуализации информации на сайте.

К основным преимуществам GitHub Pages можно отнести возможность бесплатного хостинга для статических сайтов и простоту интеграции с GitHub. Так же к преимуществам GitHub можно отнести возможность использования Jekyll (популярный генератор статических сайтов, встроенный в GitHub) для генерации сайтов, гибкость настройки, управление через Git, а так же возможность интеграции GitHub Actions.

### **1.2.1.2 Базовые инструменты веб-разработки**

Для тех, кто хочет своим портфолио продемонстрировать не только собственные работы, но и базовые навыки в веб-разработке, а также создать более персонализированное портфолио следует разрабатывать собственный сайт-портфолио с нуля. Для его реализации следует использовать стандартный стек технологий веб-разработки - HTML, CSS и

JavaScript. Данный метод предоставляет максимальную гибкость и контроль над процессом создания веб-портфолио, но также требует больше времени и усилий, ведь для создания качественного портфолио необходимо изучить все 3 базовых инструмента, а также взаимодействие между ними.

1. HTML (HyperText Markup Language) - это язык разметки, который используется для создания структуры веб-страниц. С помощью HTML можно создавать различные элементы на странице, такие как заголовки, параграфы, изображения, ссылки и таблицы. Это основа любого сайта, и для создания портфолио необходимо освоить основные теги и их использование.

2. CSS (Cascading Style Sheets) - это язык стилей, который используется для оформления веб-страниц. С помощью CSS можно настроить внешний вид элементов HTML, например, задать шрифты, цвета, отступы и расположение элементов на странице.

3. JavaScript - это язык программирования, который позволяет добавить интерактивности на веб-страницу. Например, с помощью JavaScript можно реализовать динамическую загрузку контента, анимации и другие элементы, которые сделают портфолио более привлекательным.

Таким образом, для реализации веб-портфолио с помощью данного метода необходим значительный объем знаний, времени и усилий, однако он предоставляет существенные преимущества над конструкторами сайтов. К преимуществам данного метода можно отнести возможность полной персонализации, отсутствие ограничений в функционале и возможность демонстрации собственных навыков в веб-разработке.

### **1.2.1.3 Фреймворки и библиотеки**

Для упрощения создания сайтов с помощью базовых инструментов веб-разработки были созданы специализированные фреймворки и библиотеки, которые позволяют использовать готовые шаблоны на уровне кода, а также упрощают процесс их кастомизации и интеграции в общий код проекта, что позволяет существенно сократить время, затрачиваемое на разработку. Наиболее популярными инструментами на данный момент являются Bootstrap и Tailwind CSS.

1. Bootstrap - это популярный фронтенд-фреймворк, который предоставляет набор готовых компонентов для создания адаптивных сайтов. Bootstrap включает в себя стилизованные элементы интерфейса, такие как кнопки, формы, навигационные панели, модальные окна, таблицы и многое другое. Все эти компоненты способны автоматически адаптироваться под различные устройства и экраны, в зависимости от потребностей

разработчика. К его преимуществам можно отнести готовые шаблоны и компоненты, а также простоту использования, в сравнении с базовыми инструментами веб-разработки.

2. Tailwind CSS - это утилитарный CSS-фреймворк, который позволяет разрабатывать страницы с высокой гибкостью, предоставляя множество классов для прямого управления стилями элементов. В отличие от Bootstrap, который предоставляет готовые компоненты, Tailwind предлагает заготовки элементов, с которыми можно работать в декларативном стиле. Данный фреймворк можно сравнить с современными фреймворками, которые используются в нативной мобильной разработке, такими как SwiftUI для разработки под iOS или Jetpack Compose для разработки под Android. К преимуществам Tailwind можно отнести высокую гибкость кастомизации, простоту, адаптивность под мобильные устройства и минимизацию кода.

#### **1.2.1.4 Генераторы статических сайтов**

Продолжая рассматривать инструменты разработки веб-портфолио можно так же обратить внимание на автоматизацию развертывания и обновления собственного сайта, а также на упрощение создания страниц и обновления информации на них. Для этого были разработаны специализированные инструменты - генераторы статических сайтов, позволяющие создавать новые страницы и вносить правки в расположенную на них информацию с помощью отдельных панелей администрирования. К таким инструментам относится Lektor Bot.

Lektor Bot - это инструмент для генерации статических сайтов. Он позволяет создавать сайты, генерируя страницы на основе заранее подготовленных данных.

К основным преимуществам данного инструмента можно отнести простоту в использовании, ведь он не требует специфичной настройки и развертывания, а также способен использовать в качестве источников данных заранее подготовленные файлы в формате Markdown или текстовые документы, размещая информацию из них в специальных шаблонах. В частности Lektor может использоваться для публикации кода в репозиторий через удобный графический интерфейс. Lektor предоставляет гибкость и возможность кастомизации, возможность быстрой генерации статического контента, а также легкость в хостинге, так как статические сайты могут развертываться на GitHub.

#### **1.2.2 Инструменты оформления программного кода**

Кроме создания самого веб-портфолио, необходимо так же определиться с инструментами, которые будут использоваться для оформления программного кода,

написанного в ходе выполнения работ по программирования. Наиболее популярными инструментами, доступными студенту и начинающим специалистам являются - README файлы, в каждом из которых описывается основной функционал конкретной работы, документация кода с помощью DocString, которая в дальнейшем упростит понимание кода, а также оформление работ по машинному обучению с помощью Jupyter Notebooks.

1. README файл - это один из первых файлов, который обычно просматривают пользователи или разработчики при знакомстве с проектом. Этот файл является важным инструментом для предоставления информации о проекте, его целях, установке и использовании. Правильное оформление README файла помогает ускорить понимание сути работы и облегчить ее восприятие. Пример проекта с README файлом представлен в приложении 1.

2. В ходе обучения, для выполнения работ по программированию и машинному обучению, чаще всего мной использовался язык программирования Python, основным инструментом для документации кода в котором является DocString. Это “строка” документации, которая используется в Python для описания функциональности модулей, классов и функций. Она является неотъемлемой частью хорошей практики оформления кода, так как позволяет другим разработчикам понять, что делает тот или иной элемент программы, без необходимости чтения всего кода целиком. Стоит отметить, что использование DocString позволяет в дальнейшем создать документацию к проекту с помощью специализированных генераторов документации, например Sphinx. Пример такой документации представлен в приложении 1.

3. Jupyter Notebooks - это инструмент для создания документов, которые содержат как код, так и визуализацию данных. Он широко используется в проектах, связанных с машинным обучением, анализом данных и научными исследованиями. Jupyter Notebooks позволяет интегрировать код Python, графику, математические уравнения, текст и визуализации в одном документе, что делает его идеальным инструментом для презентации и демонстрации работ. Пример такой визуализации данных представлен в приложении 1.

Таким образом, разработка веб-портфолио предполагает широкий выбор инструментов, от базовых технологий (HTML, CSS, JavaScript) до готовых решений, таких как сторонние сервисы или генераторы статических сайтов. Для студентов и начинающих специалистов, занимающихся программированием, важно не только создать визуально привлекательное портфолио, но и продемонстрировать свои профессиональные навыки, которые включают в себя написание и документирование кода, а также использование современных технологий.

## Практический блок

### 2.1 Реализация веб-портфолио на основе Bootstrap

После проведения анализа основных требований к веб-портфолио в первой части работы, основных принципов и инструментов, были выбраны те, которые будут использоваться при реализации веб-портфолио по программированию.

В данной части работы рассматривается процесс создания веб-портфолио с использованием основных инструментов веб-разработки - HTML, CSS, JavaScript, для упрощения работы с которыми используется фреймворк Bootstrap 5, а также статический генератор Lektor, использовавшийся для упрощения публикации сайта в GitHub.

Процесс разработки личного веб-портфолио можно разделить на 5 основных этапов:

1. Подготовка репозитория для публикации веб-портфолио;
2. Установка и настройка Lektor;
3. Подключение Bootstrap и настройка шаблонов для реализации структуры;
4. Наполнение веб-портфолио работами по учебным дисциплинам;
5. Публикация сайта на GitHub.

#### 2.1.1 Подготовка репозитория для публикации веб-портфолио

Перед началом разработки веб-портфолио, были созданы два репозитория на GitHub. Первый для публикации и хостинга веб-портфолио, второй для публикации работ, выполненных в ходе обучения, в том числе по программированию и машинному обучению. На рисунке 1 представлен пример репозитория для публикации статического сайта с помощью Lektor. А на рисунке 2 представлен репозиторий для хранения работ по дисциплинам. Ссылки на данные репозитории представлены в приложении 1.

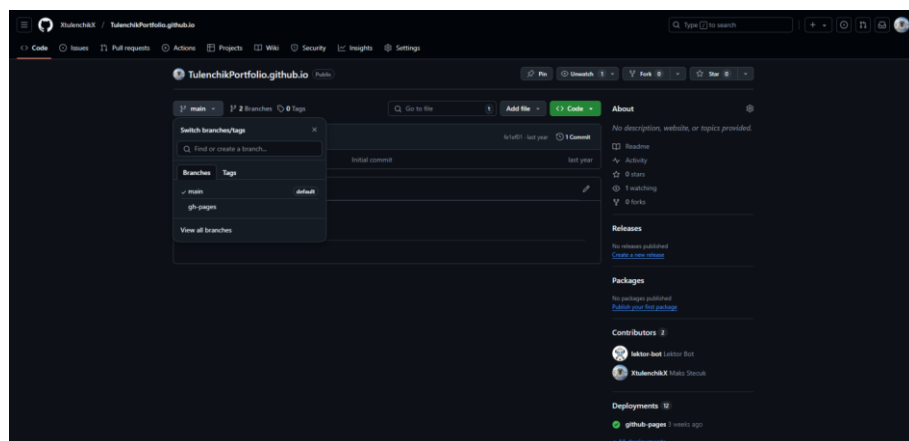


Рис. 1: Репозиторий для публикации веб-портфолио

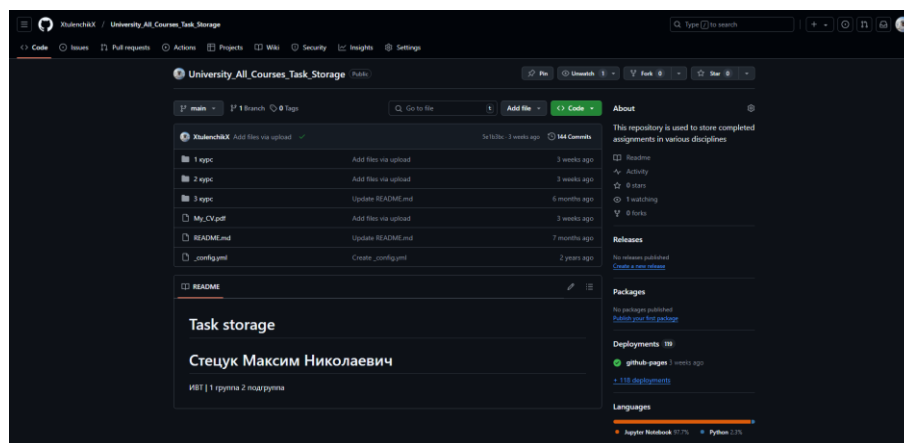


Рис. 2: Репозиторий для публикации работ по учебным дисциплинам

## 2.1.2 Установка и настройка Lektor

После подготовки репозитория, была выполнена установка и настройка Lektor. Для работы с программным кодом использовался редактор - VisualStudio Code, с установленными расширениями для работы с базовыми инструментами веб-разработки, такими как HTML, CSS и JavaScript, данные расширения были необходимы для улучшения наглядности, читаемости, структуризации и компиляции кода, так как основной процесс развертывания и публикации был реализован с помощью Lektor. После подготовки редактора к работе с кодом, был создан каталог проекта для размещения веб-портфолио.

После того, как подготовительные шаги были выполнены, была выполнена установка и настройка Lektor. Для загрузки Lektor использовался менеджер пакетов pip, после чего с помощью команды он был инициализирован в ранее созданном каталоге. После прохождения шагов в консоли и подключения репозитория GitHub, с использованием ghq таргета, была сгенерирована базовая структура проекта, представленная на рисунке 3. После этого был запущен локальный хостинг, на рисунке 4 представлена базовая страница настройки статично сайта, при его создании с помощью Lektor. Команды использованные на данных шагах представлены на листинге 1.

```
// загрузка Lektor с помощью утилиты pip
pip install lektor

// инициализация Lektor в проекте
lektor quickstart

// запуск локального хостинга
Lektor server
```

Листинг 1: Установка и запуск Lektor

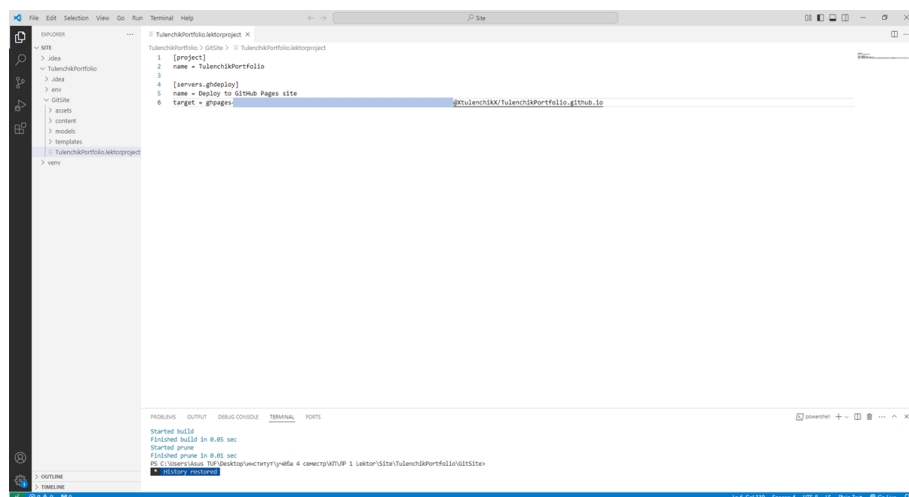


Рис. 3: Базовая структура проекта Lektor

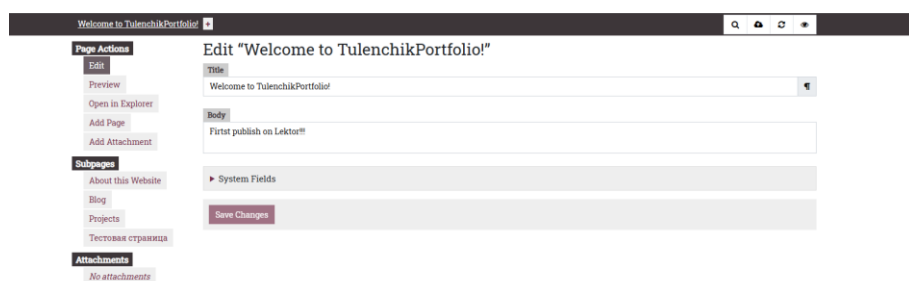


Рис. 4: Страница настройки сайта

В данной работе не были использованы все возможности Lektor, он использовался только для упрощения запуска локального хостинга, создания структуры проекта и простой публикации изменений в репозиторий с помощью специального пользовательского интерфейса.

### 2.1.3 Подключение Bootstrap и настройка шаблонов для реализации структуры

После того как была подготовлен пустой проект, началось формирование списка разделов, которые необходимо реализовать в портфолио, основанное на требованиях рассмотренных в первой главе данной работы. На данном шаге были выделены 3 основных блока, которые будут представлены в веб-портфолио:

- Блок с информацией об авторе;
- Блок с работами по дисциплинам, распределенных по курсам;
- Блок с контактной информацией.

После этого был выбран шаблон сайта, написанный с помощью HTML, CSS, JS и фреймворка Bootstrap 5. Выбор шаблонов основывался на следующих параметрах: минималистичный дизайн, количество доступных к использованию элементов, возможность их кастомизации, а также актуальность применения данного шаблона для сайта-портфолио и возможность реализации адаптивности под мобильные устройства.

После проведения анализа, выделения основных частей портфолио и определения с дизайном, шаблон кода был размещен в проекте, а также существенно доработан, чтобы соответствовать требованиям веб-портфолио. Пример фрагмента кода и файла с шаблоном представлен на рисунке 5.

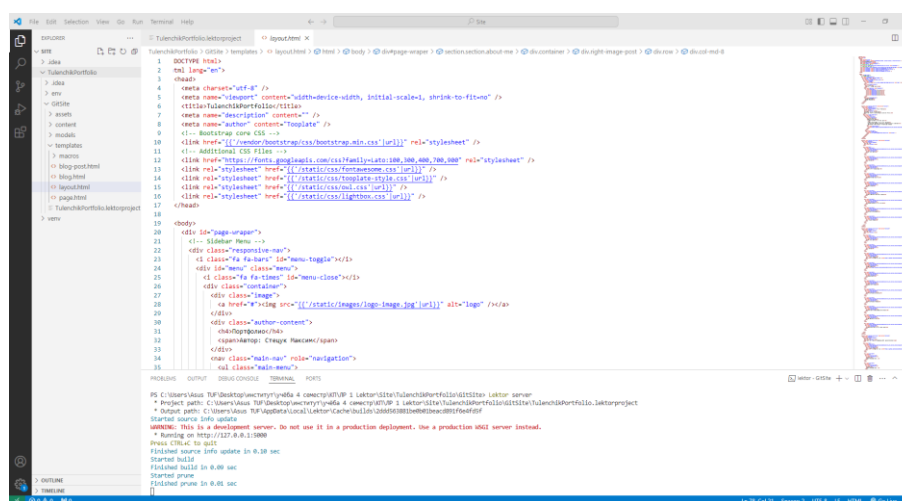


Рис. 5: Основной файл с разметкой сайта

После того, как были внесены все необходимые правки в шаблон, были добавлены все требуемые блоки и заполнена информация об авторе контакты. Затем были подготовлены заготовки элементов, которые будут заполнены при наполнении портфолио работами по дисциплинам, пример кода заготовки представлен на листинге 2. Реализованная структура портфолио представлена на рисунке 6.

```
<div class="col-md-6">
  <a href="">
    <div class="service-item">
      <i class="discipline"></i>
      <h4>Название дисциплины</h4>
    </div>
  </a>
</div>
```

Листинг 2: Заготовка элемента под дисциплину



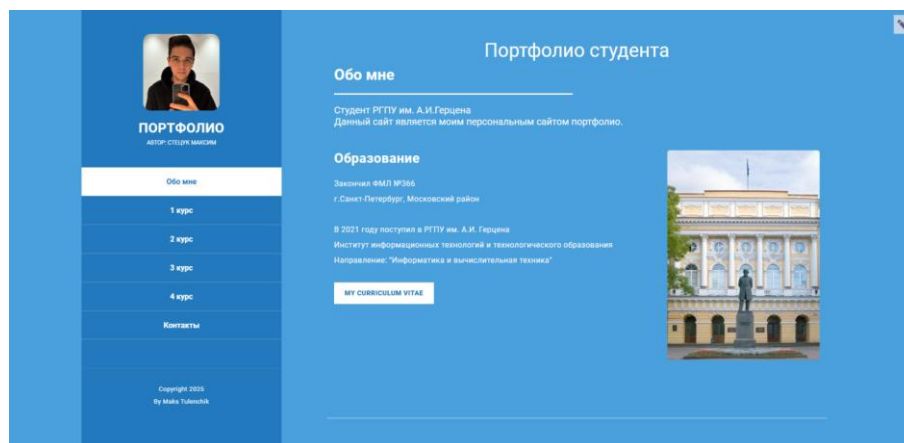


Рис. 6: Локально развернутое портфолио

### 2.1.4 Наполнение веб-портфолио работами по различным дисциплинам

После того как, шаблон был адаптирован под веб-портфолио и все необходимые заготовки элементов были добавлены, началось его наполнение работами по дисциплинам, в том числе и работами по программированию. Для этого использовались заготовки, представленные на листинге 2. Были заполнены названия дисциплин, указаны ссылки на каталоги в репозитории, подготовленном на первом шаге, а также для дисциплины “Программирование” был расширен шаблон и добавлен подзаголовок, который использован для разделения работ по семестрам, так как работы по программированию в каждом семестре являлись развитием предыдущих. Данный подход позволил лучше структурировать лабораторные работы и представить их наиболее доступным образом. На рисунке 7 представлены примеры заполненных элементов для различных дисциплин.

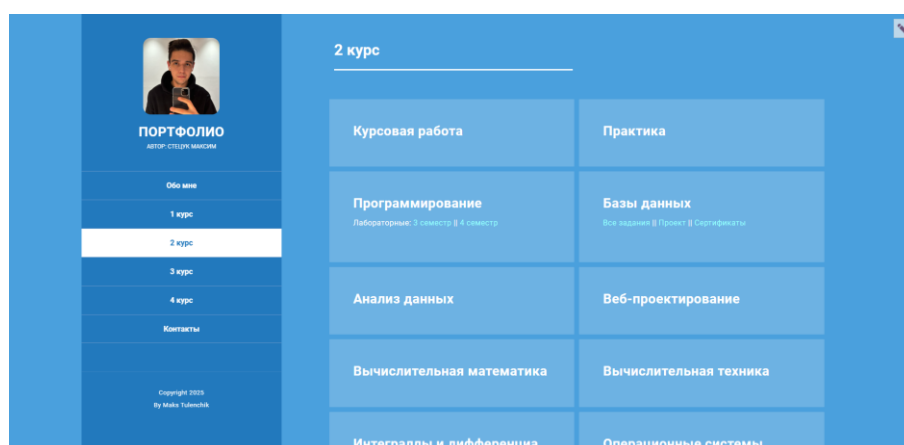


Рис. 7: Отображение дисциплин в веб-портфолио

### 2.1.5 Публикация сайта на GitHub

На данном шаге были внесены финальные правки в оформление портфолио, а также в настройках CSS стилей было добавлено адаптивное отображения веб-портфолио на различных устройствах. После чего с помощью интерфейса Lektor, представленного на рисунке 4, сайт был опубликован в GitHub. После успешной публикации, в настройках репозитория был реализован хостинг с помощью GitHub Pages. Ссылка на репозиторий с полным кодом опубликованного портфолио представлена в приложении 1, а ссылка на само веб-портфолио представлена в приложении 2. На рисунке 8 представлено адаптивное отображение сайта-портфолио на мобильном устройстве.

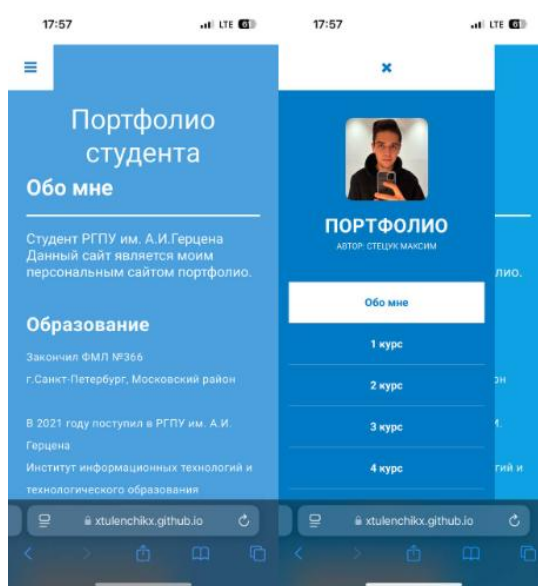


Рис. 8: Отображение портфолио на мобильном устройстве  
с двумя состояниями бокового меню

Подводя итоги, можно сказать, что реализация веб-портфолио с использованием Lektor и Bootstrap является отличным способом создать статичный, адаптивный и красивый сайт, с возможностью простой публикации изменений через пользовательский интерфейс. А также стоит отметить, что хостинг сайта через GitHub Pages позволяет не только обеспечить доступность портфолио через интернет, без оплаты отдельного сервера, но и демонстрирует знания и навыки в области работы с Git и GitHub, которые необходимы любому специалисту в области информационных технологий, хотя бы на базовом уровне.

## 2.2 Публикация работ по программированию и машинному обучению

Как ранее было сказано, оформление кода играет ключевую роль при создании портфолио разработчика. В данной части работы рассматривается процесс формирования отчетов к выполненным работам по дисциплине “Программирование” с использованием README файлов и языка разметки Markdown, а также документирование кода с помощью DocString и визуализация выполнения этапов с помощью Jupyter Notebook.

Для каждой работы по программированию, результатом которой является некоторый программный продукт, был добавлен README файл, представляющий собой простейший вариант документации к программному продукту, с описанием основного функционала и информацией об авторе. Пример такого файла представлен на рисунке 9.

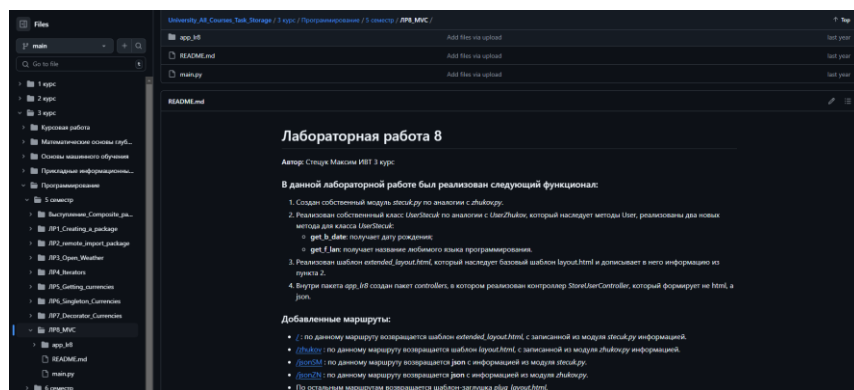


Рис. 9: Пример README файла

Кроме добавления файла README файлов, были добавлены комментарии к коду, с помощью DocString, в тех местах, в которых они не были написаны в процессе разработки, что позволило существенно упростить читаемость кода и предоставить возможность использовать его в дальнейшем, без необходимости повторного чтения всего кода. Пример документации функции с помощью DocString представлен на рисунке 10.

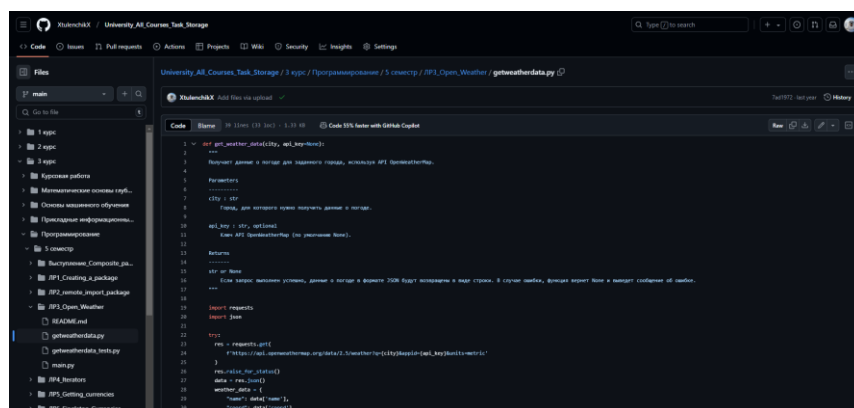


Рис. 10: Пример документации функции с помощью DocString

Такой способ документации кода позволяет не только улучшить его читаемость, но и генерировать документацию к проектам в виде статических сайтов, ссылка на пример сгенерированной документации к одному из проектов представлена в приложении 1. На рисунке 11 представлен пример сгенерированной документации в виде статического сайта с помощью Sphinx.

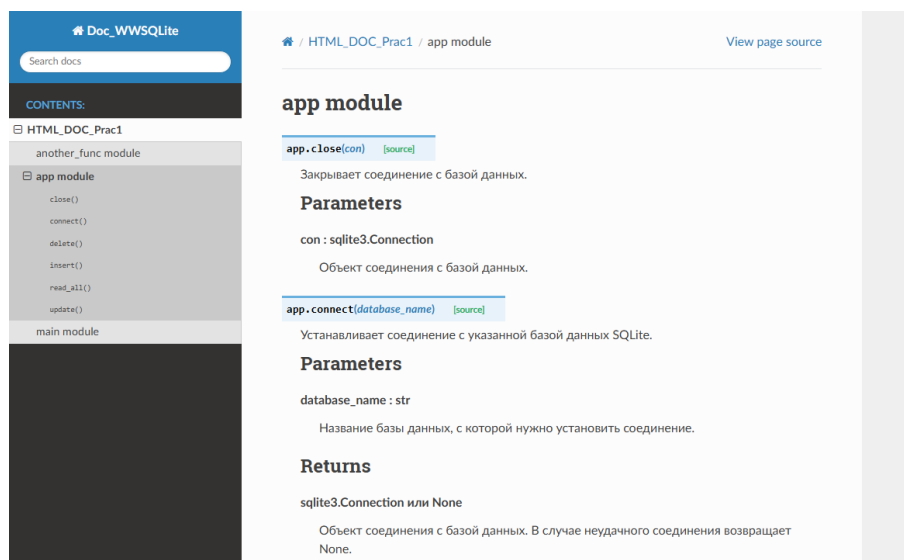


Рис. 11: Пример документация в виде статического сайта

Однако не для любой работы по программированию подойдет вариант с использованием лишь README и DocString. В работах по машинному обучению необходимо четко визуализировать результаты на промежуточных этапах. Для этого подходит такой вариант оформления работ, как Jupyter Notebooks. Данный инструмент позволяет отображать промежуточные результаты прямо в коде, а также добавлять собственные комментарии к ним. Пример использования данного инструмента в лабораторной работе представлен на рисунке 12, а ссылка на репозиторий с данной работой представлена в приложении 1.

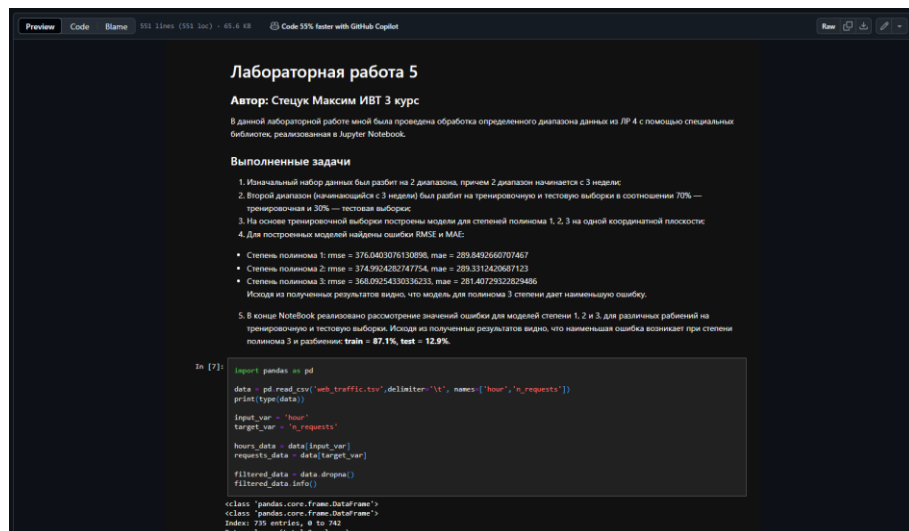


Рис. 12: Пример оформления работы по машинному обучению

Подводя итоги, важно отметить, что правильное оформление и публикация кода на таких платформах, как GitHub, с использованием Docstring, Markdown и Jupyter Notebooks, позволяет не только организовать и задокументировать собственные программные проекты, но и продемонстрировать высокий уровень профессионализма, что существенно усиливает вашу ценность, как начинающего специалиста.

## ЗАКЛЮЧЕНИЕ

При написании курсовой работы по представленной теме были изучены профессиональные статьи и литература, включавшие справочные материалы по принципам создания и требованиям к веб-портфолио технического специалиста, а также ресурсы сети Интернет посвященные качественному оформлению кода и инструментам, использующимся при создании сайтов-портфолио специалистами различных сфер деятельности.

В теоретической части курсовой работы рассмотрены современных способы и инструменты для создания веб-портфолио, с указанием конкретных примеров и их преимуществ. Также в теоретической части рассмотрены основные требования к веб-портфолио студента, которые необходимо соблюдать для успешной реализации качественного личного сайта-портфолио. В практической части работы были рассмотрены основные шаги, выполненные при реализации личного сайта портфолио с использованием Bootstrap и Lektor, начиная проектирования структуры сайта и создания пустого проекта, заканчивая реализацией адаптивности моего личного сайта под различные устройства и публикацией в GitHub. Помимо разработки веб-портфолио, практическая часть работы включала в себя оформление работ по программированию, с помощью добавления README файлов, документации функций с помощью DocString и структурирования работ по машинному обучению с помощью Jupyter Notebook.

Таким образом, было проведено исследование материалов по теме курсовой работы, изучены инструменты и реализовано личное веб-портфолио по дисциплине “Программирование”. Реализованное в ходе выполнения курсовой работы веб-портфолио является важным инструментом, который будет применяться мной, как при сдаче Государственной итоговой аттестации, так и в начале построения карьеры сфере информационных технологий.

## ЛИТЕРАТУРА

1. Pibernik J., Kanižaj B., Frljak L. Conceptual models for better quality of portfolio web site experience //Proceedings ELMAR-2012. – IEEE, 2012. – С. 161-164.
2. Kikuchi K., Garcia G. J., Manio J. Development of a Web-Based Student Curriculum Guide and Portfolio //2023 21st International Conference on ICT and Knowledge Engineering (ICT&KE). – IEEE, 2023. – Т. 21. – С. 1-5.
3. Bräuer G., Ziegelbauer C. The Electronic Portfolio: Self-Regulation and Reflective Practice //Digital Writing Technologies in Higher Education: Theory, Research, and Practice. – Cham : Springer International Publishing, 2023. – С. 245-259.
4. Панюкова С. В., Гостин А. М., Самохина Н. В. Интерактивное веб-портфолио студента: структура и содержание //Информатика и образование. – 2013. – №. 10. – С. 83-87.
5. Шакирова З. Х. Анализ функциональных возможностей веб-портфолио студента //Современная техника и технологии. – 2016. – №. 2. – С. 59-62.
6. Petersen H. From static and dynamic websites to static site generators //university of TARTU, Institute of Computer Science. – 2016.
7. Spurlock J. Bootstrap: responsive web development. – " O'Reilly Media, Inc.", 2013.
8. Павленко И. И. Электронный портфолио в вузе //Прикладная информатика. – 2008. – №. 6. – С. 3-5.
9. McMillan C. et al. Portfolio: finding relevant functions and their usage //Proceedings of the 33rd International Conference on Software Engineering. – 2011. – С. 111-120.

## ПРИЛОЖЕНИЕ 1



Ссылка на репозиторий с кодом портфолио: <https://clck.ru/3FNQXW>



Ссылка на репозиторий с выполненными работами: <https://clck.ru/3FNQV7>



Ссылка на пример README файла: <https://clck.ru/3FNQQm>



Пример документирования функции с помощью DocString: <https://clck.ru/3FNQeK>





Пример Jupyter Notebook: <https://clck.ru/3FNQho>



Пример документации сгенерированной с помощью Sphinx: <https://clck.ru/3FNQkS>

## ПРИЛОЖЕНИЕ 2



Ссылка на веб-портфолио: <https://clck.ru/3FNQa3>