

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

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



Основная профессиональная образовательная программа  
Направление подготовки 09.03.01 Информатика и вычислительная техника  
Направленность (профиль) «Технологии разработки программного обеспечения»  
форма обучения – очная

## **Курсовая работа**

по дисциплине «Организация электронной образовательной среды»  
«Технологии разработки веб-портфолио студента-программиста на  
примере оформления дисциплины “Практикум по  
программированию”»

Обучающейся 4 курса  
Елкиной Галины Александровны



Научный руководитель:  
старший преподаватель  
Аксютин Павел Александрович

Санкт-Петербург  
2021

# Содержание

|  |    |
|--|----|
| ВВЕДЕНИЕ .....   | 3  |
| ГЛАВА 1. АНАЛИЗ СЕРВИСОВ И ИНСТРУМЕНТОВ ДЛЯ СОЗДАНИЯ ВЕБ-ПОРТФОЛИО ..... | 5  |
| 1.1 Анализ сервисов .....  | 5  |
| 1.2 Анализ инструментов.....   | 12 |
| ГЛАВА 2. РАЗРАБОТКА ВЕБ-ПОРТФОЛИО ПО ВЫБРАННОЙ ДИСЦИПЛИНЕ .....          | 14 |
| 2.1 Проектирование структуры портфолио .....                             | 14 |
| 2.2 Реализация выбранной структуры портфолио .....                       | 16 |
| ЗАКЛЮЧЕНИЕ.....  | 20 |
| ЛИТЕРАТУРА .....   | 21 |

# ВВЕДЕНИЕ

Портфолио — это созданный и оформленный список проектов. Портфолио своих работ создают представители множества профессий: веб-разработчики, фотографы, дизайнеры, художники, журналисты, модели, юристы, PR-менеджеры, программисты и представители других профессий.

Портфолио используют для представления своих навыков и умений, знаний и компетенций опытные представители профессий, новички и студенты, в общем, те, кто хочет показать свои возможности кому-либо. Чаще всего портфолио создают для прикрепления его к резюме, но также есть случаи, когда это делается для демонстрации результатов, например, студентами ВУЗов.

Портфолио можно представлять двумя способами: в бумажном и электронном форматах. Для представителей IT-профессий наиболее удобным будет электронный формат, так как множество вещей, созданных, например, программистами, хранятся именно в электронном формате и чаще всего на просторах Интернета. Поэтому свое портфолио программисты – и опытные, и студенты, новички, – создают на сервисах, позволяющих хранить большие объемы данных, а также позволяющих представить их в более выгодном свете.

Для создания веб-портфолио могут использоваться различные сервисы, например, Notion, GitHub Pages, Behance и Dribbble, и инструменты, такие как система контроля версий Git, VS Code, Notepad+.

Целью данной курсовой работы является исследовать технологии создания веб-портфолио: сервисы и инструменты – и, найдя оптимальный вариант, создать портфолио студента программиста для выбранной дисциплины.

Исходя из цели курсовой работы, были поставлены следующие задачи:

1. Составить и проанализировать список сервисов, которые предоставляют возможность создания веб-портфолио.

2. Выбрать наиболее подходящий сервис для создания портфолио по дисциплине «Практикум по программированию».
3. Проанализировать инструменты, которые могут понадобиться для составления веб-портфолио с использованием выбранного сервиса.
4. Выбрать наиболее подходящие инструменты для поставленной цели.
5. Создать веб-портфолио по выбранной дисциплине с использованием выбранных сервиса и инструментов.
6. Дать оценку проделанной работе.

Также отмечу, что данная курсовая работа позволит приобрести навык создания портфолио для будущего резюме, ведь работодателям часто важно знать, что именно умеет соискатель, а портфолио – отличная возможность показать это, особенно для студента.

# ГЛАВА 1. АНАЛИЗ СЕРВИСОВ И ИНСТРУМЕНТОВ ДЛЯ СОЗДАНИЯ ВЕБ-ПОРТФОЛИО

## 1.1 Анализ сервисов

Для создания портфолио существует множество сервисов, предоставляющих разные возможности. Наиболее распространенными сервисами, где программисты составляют свои портфолио, являются сервисы Notion, GitHub Pages и Behance.

Но перед тем, как приступать к анализу самих сервисов, нужно определиться с критериями, по которым можно будет оценить тот или иной сервис, а затем определить наиболее подходящий для выбранной дисциплины.

Так как работы по выбранной мной дисциплине представляют собой страницы сайтов и макетов сайтов, а также программный код, на сервисе должны быть доступны следующие возможности:

- представление веб-страниц сайта в формате «as-is»;
- вставка ссылок, ведущих на конкретные файлы;
- представление программного кода с подсветкой синтаксиса выбранного языка программирования;
- вставка изображений на страницу портфолио;
- создание отдельных страниц для отдельных работ.

Также для меня важно иметь возможность создать свой собственный дизайн страницы портфолио, что может выражаться в создании дизайна с нуля при помощи веб-технологий или создании макета страницы из готовых блоков, которые также можно настроить «под себя».

Чтобы провести качественный анализ сервисов, которые могут подойти мне для реализации веб-портфолио в рамках этой курсовой работы, я составила таблицу 1, в которой отразила все характеристики каждого сервиса по каждому из определенных мной критериев, нужных для создания веб-портфолио.

Таблица 1. Сравнительный анализ сервисов

| Создание отдельных страниц для отдельных работ |   |
|--|---|
| <b>Notion</b>                                  | <p>На данном сервисе можно создавать отдельные страницы несколькими способами.</p> <ol style="list-style-type: none"> <li>1. Создать страницу со ссылкой на нее на текущей странице. Тогда созданная страница будет находится внутри текущей страницы.</li> <li>2. Создать страницу независимую от текущей и прикрепить ссылку на нее отдельно.</li> </ol> <p>Такая ссылка в обоих случаях будет выглядеть, как гипертекст. Также на самой странице можно добавить опознавательный значок-иконку для каждой страницы. Иконку можно создать с помощью эмоджи или загруженной картинки буквально в два-три клика.</p> <p>Для перехода между страницами используется строка навигации вверху страницы.</p>           |
| <b>GitHub Pages</b>                            | <p>В GitHub Pages отдельные страницы можно создавать с помощью отдельных папок. Также можно отображать отдельные файлы в виде страниц. Реализовать отображение отдельных страниц можно двумя способами:</p> <ol style="list-style-type: none"> <li>1) с помощью веб-технологий. Нужно создать отдельные папки для каждой страницы. В каждую из папок нужно положить пакет файлов, реализующих веб-интерфейс, то есть это будут файлы форматов HTML+CSS, иногда также JS-файлы.</li> <li>2) с помощью разметки Markdown. Для этого нужно реализовать также некоторую систему из папок и/или страниц, которые будут связаны между собой. Если это будут папки, то в каждой из папок должен находится MD-</li> </ol> |

|                            |  |
|----------------------------|--|
|                            | <p>файл, который сервис сможет преобразовать к веб-формату.</p> <p>Для реализации перехода с одной страницы на другую нужно реализовать связь между страницами по средствам ссылок/кнопок.</p>   |
| <b>Behance</b>             | Отдельные страницы можно создать по средствам создания карточек с проектами. Перемещение между такими карточками может осуществляться только через переход обратно на страницу профиля пользователя.   |
| <b>Вставка ссылок</b>      |  |
| <b>Notion</b>              | <p>Можно вставить различные типы ссылок:</p> <ul style="list-style-type: none"> <li>• обычная текстовая ссылка – гипертекст;</li> <li>• ссылка на страницу с иконкой;</li> <li>• вставка через ссылку различных файлов: <ul style="list-style-type: none"> <li>○ PDF,</li> <li>○ видео,</li> <li>○ аудио;</li> </ul> </li> <li>• ссылки на страницы/файлы из разных сервисов, например, Google Диск, Twitter.</li> </ul> |
| <b>GitHub Pages</b>        | <p>Вставка ссылок возможна через файлы Markdown или через веб-технологии. В первом случае это будет текстовая ссылка, во втором:</p> <ul style="list-style-type: none"> <li>• текстовая ссылка,</li> <li>• кнопка со ссылкой,</li> <li>• иконка со ссылкой,</li> <li>• вставка чего-либо через тег embed.</li> </ul>   |
| <b>Behance</b>             | Создавать ссылки невозможно.   |
| <b>Вставка изображений</b> |  |
| <b>Notion</b>              | Изображения можно добавить несколькими способами:  |

|  |   |
|--|---|
|  | <ul style="list-style-type: none"> <li>• через базу данных вида «Галерея»;</li> <li>• просто на страницу через функцию вставки изображения;</li> <li>• через ссылку.</li> </ul>   |
| <b>GitHub Pages</b>                    | <p>Изображения через Markdown можно вставить, только если перед этим картинка была загружена в репозиторий или по ссылке на источник изображения.</p> <p>Также с помощью тега <code>&lt;img&gt;</code> можно добавить изображение как в файл Markdown, так и на html-страницу из папки репозитория.</p>   |
| <b>Behance</b>                         | Изображения помещаются на страницу проекта только со своего компьютера и только в формате галереи: когда изображения идут друг за другом в колонке.   |
| <b>Представление программного кода</b> |   |
| <b>Notion</b>                          | <p>Для этого в Notion есть специальный блок, в котором можно написать код программы. Если данный язык программирования, на котором написана ваша программа, будет в списке языков программирования этого блока, – а в данном блоке достаточно большое количество поддерживаемых языков программирования, – то синтаксис программы даже будет подсвечен. Если же такого языка не оказалось, то Notion просто сделает ваш код моноширинным шрифтом, что тоже неплохо.</p>                                   |
| <b>GitHub Pages</b>                    | <p>При выборе стека GitHub Pages и Markdown вы можете воспользоваться специальным синтаксисом, который позволит вам написать код моноширинным шрифтом, выделив его в отдельный блок и подсветив синтаксис программы, если это один из распространенных языков программирования.</p> <p>При выборе стека GitHub Pages и веб-технологий программный код можно писать с помощью тегов <code>code</code> и <code>pre</code>. Однако в этом случае не будет реализовано подсветки синтаксиса. Но это можно</p> |



|  |  |
|--|--|
|  | исправить, подключив специальную библиотеку по средствам CSS и Javascript.   |
| <b>Behance</b>                                     | Вставка кода невозможна. Единственное, что можно сделать, это вставить код по средствам изображения, то есть сделать скриншот кода из редактора кода и выложить его на страницу проекта.   |
| <b>Представление веб-страниц в формате «as-is»</b> |  |
| <b>Notion</b>                                      | Данная функция в Notion доступна только в формате вставки веб-страницы на страницу Notion. То есть если веб-страница существует на каком-либо сайте, то ее можно встроить в страницу Notion с помощью функции embed и настроить размеры вставки данного поля.  |
| <b>GitHub Pages</b>                                | Данный сервис может самостоятельно из готового стека HTML+CSS+JS, организованного в отдельной папке со своей структурой, создать веб-страницу и отобразить ее в браузере в формате полноценной страницы.   |
| <b>Behance</b>                                     | Данная функция на сервисе не доступна. Веб-страницу сайта можно загрузить только как изображение.  |
| <b>Пользовательский дизайн страниц</b>             |  |
| <b>Notion</b>                                      | <p>Чтобы сделать дизайн страницы в Notion «под себя», сервис предоставляет некоторые возможности, которые перечислены ниже.</p> <ul style="list-style-type: none"> <li>• Персонализация шапки страницы. В Notion шапка страницы называется обложкой. Обложку можно добавить или убрать. Добавить ее можно, выбрав подходящую картинку у себя на компьютере или используя ссылку на картинку из Интернета. Также можно выбрать градиентный фон из коллекции Notion или понравившуюся картинку из предоставленной коллекции Unsplash.</li> </ul> |

|                     |  |
|---------------------|--|
|                     | <ul style="list-style-type: none"> <li>• Персонализация цветовой гаммы блоков. В Notion нельзя изменить фоновый цвет страницы, но можно менять фоновый цвет отдельных блоков. Также можно изменять цвет текста на странице, но это доступно только в наполнении страницы, а не в ее заголовке.</li> <li>• Организация блоков страницы. Блоки текстовой информации можно организовать несколькими способами. Первый – их можно выстроить в несколько колонок, в которых могут быть также несколько блоков. Второй – Notion предоставляет возможность представить информацию с помощью нескольких видов баз данных, в том числе с помощью Галереи, что тоже является способом организации информации и персонализации дизайна страницы.</li> </ul>   |
| <b>GitHub Pages</b> | <p>Данный сервис обладает огромными возможностями для персонализации дизайна страниц.</p> <p>Если вы создали страницу с помощью разметки Markdown, то у вас по умолчанию действует некоторый «дизайн» - простой текст, увеличенный шрифт для различных заголовков, форматирование строк таблиц в бело-серую «полоску».</p> <p>Если же вы выбрали такой способ создания страницы, но при этом хотите ее «оживить», то можно воспользоваться готовыми темами для оформления страниц от GitHub Pages в рамках одного репозитория. Это позволит сделать дизайн страниц более ярким, но это будет шаблонный дизайн, подстроенный исключительно под типовые стили.</p> <p>Однако, если вы хорошо владеете веб-технологиями для создания статического сайта, вы можете создать абсолютно любой дизайн, который будет вам по вкусу. Для этого вы верстаете статический</p> |

|                |  |
|----------------|--|
|                | <p>сайт на html+css, если это необходимо, то подключаете js-файлы для анимации или динамической загрузки данных. Затем помещаете файлы в репозиторий на GitHub по принципу «одна страница – одна папка – один файл с названием “index.html”». И после деплоя GitHub Pages наслаждаетесь собственным дизайном каждой страницы.</p> <p>Также можно смешивать описанные подходы, что иногда оказывается очень удобно.</p> |
| <b>Behance</b> | <p>Из персонализации дизайна страницы пользователь может поменять только свой аватар и выбрать шапку для главной страницы. Все остальное изменять нельзя.</p>  |

После детального анализа сервисов по критериям можно утверждать, что Behance абсолютно не подходит для цели данной курсовой работы – создания веб-портфолио программиста, так как большинство критериев, выдвинутых мной, не были соблюдены.

Notion гораздо больше подходит для моих задач, так как многие критерии соблюдаются, и по некоторым из них Notion даже лучше остальных сервисов. Однако у него есть несколько проигрышных сторон:

- 1) персонализация страниц очень ограничена;
- 2) чтобы представить работу по веб-дизайну, например, нужно сначала загрузить исходные файлы на сервис, где они создадут единую веб-страницу, и только после этого можно будет загружать эту страницу в Notion.

Таким образом, самым лучшим вариантом для создания веб-портфолио по выбранной мной дисциплине является сервис GitHub Pages, так как он удовлетворяет всем критериям отбора сервиса.

## 1.2 Анализ инструментов

Для создания веб-портфолио через сервис GitHub Pages мне необходимо использовать несколько инструментов для различных целей:

- редактор кода для создания персонализированных страниц портфолио;
- система контроля версий для загрузки файлов в репозиторий портфолио;
- редактор Markdown-разметки для создания некоторых простых страниц портфолио.

Так как я выбрала GitHub Pages в качестве сервиса, куда буду загружать свое портфолио, то с системой контроля версий определяться не нужно. Я буду использовать Git для этой цели, так как это система контроля версий по умолчанию для GitHub-сервиса. Git – удобная и простая утилита для компьютера, которая позволяет отправлять локальные файлы в удаленный репозиторий. Данная система вполне подходит для достижения цели данной курсовой работы.

Однако выбор инструмента, позволяющего редактировать код и Markdown-файлы, не так очевиден. Для этих задач я выбрала Visual Studio Code (VS Code) по нескольким причинам.

1. Данное приложение достаточно легковесное в отличие от других известных мне подобных приложений.
2. VS Code обладает простым и понятным интерфейсом с русской адаптацией.
3. Данный редактор умеет работать со всеми типами файлов, которые нужны для реализации веб-портфолио, а именно с файлами типов HTML, CSS, JS, MD.
4. Также данный редактор подсвечивает синтаксис любого из вышеперечисленных типов файлов, что упрощает работу с ними.

5. В VS Code есть возможность предварительного просмотра результата обработки разметки Markdown, что значительно увеличивает скорость работы с данным типом файлов.
6. Также можно управлять расположением и некоторыми свойствами файлов и папок прямо через интерфейс данного редактора, что делает его более удобным с точки зрения работы с архитектурой проекта, в данном случае, веб-портфолио.
7. Не мало важным является то, что VS Code имеет свой собственный терминал, через который можно воспользоваться системой контроля версий Git. Для менее опытных пользователей у данного редактора есть GUI-интерфейс, который поможет работать с Git при помощи кликов мыши.

Определившись с сервисом, с помощью которого будет размещено портфолио, и инструментами, с помощью которых будет реализовано портфолио, я приступила к созданию веб-портфолио по дисциплине «Практикум по программированию».

## ГЛАВА 2. РАЗРАБОТКА ВЕБ-ПОРТФОЛИО ПО ВЫБРАННОЙ ДИСЦИПЛИНЕ

### 2.1 Проектирование структуры портфолио

Для того чтобы выбрать оптимальную структуру для портфолио, нужно понять, что именно будет представлять из себя портфолио.

По выбранной мной дисциплине было выполнено шесть работ:

- две из этих работ представляют собой страницы макетов сайта, которые требуют публикации на отдельной странице портфолио;
- остальные четыре представляют собой программный код, который лучше всего публиковать частично, но при этом обязательно оставить в портфолио ссылку на полный код программы.

Исходя из этого, я составила макет главной страницы портфолио, который представлен на рисунке 1.

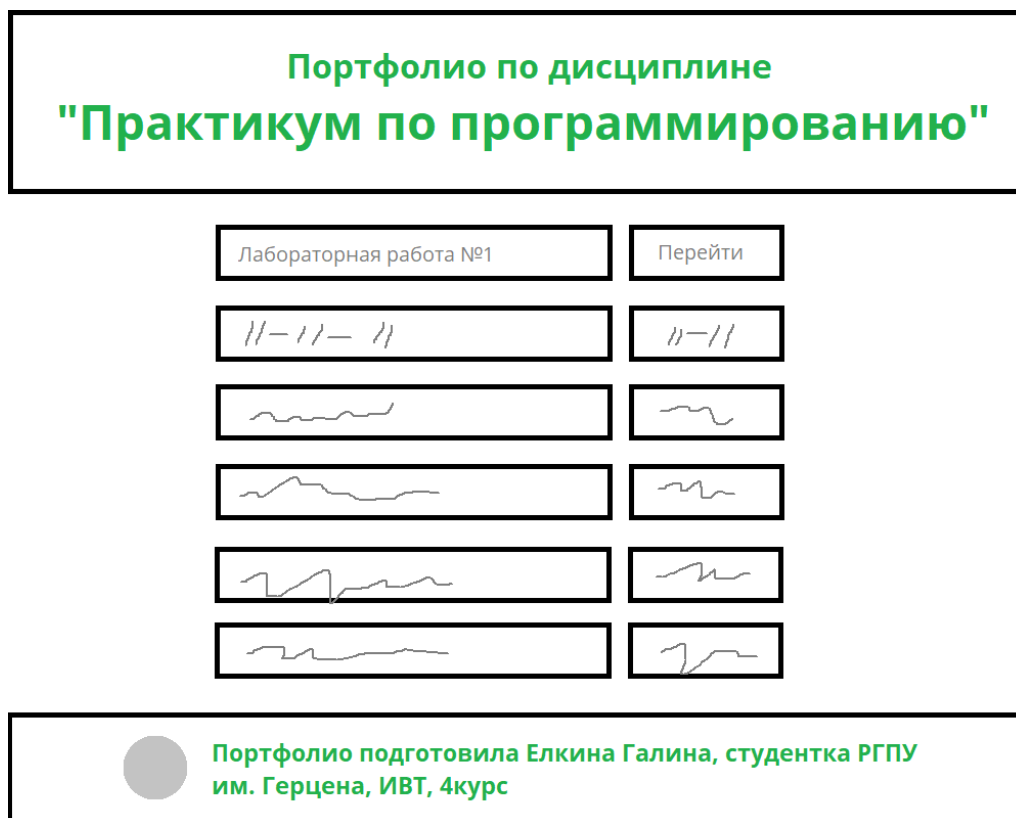


Рисунок 1. Макет главной страницы портфолио

Также не мало важно было составить макет страниц с заданиями, включающими в себя работу с программным кодом. Этот макет представлен на рисунке 2.

|   |
|---|
| <b>Название работы</b>  |
| Описание работы   |
| <div>Кусок программного кода</div> <div>////////////////////////////////////</div> <div>-----</div> <div>////////////////////////////////////</div> <div>-----</div> <div>////////////////////////////////////</div> <div>-----</div> <div>////////////////////////////////////</div> |
| Ссылка на полный код программы  |

*Рисунок 2. Макет страницы, с включенным в нее программным кодом*

Главную страницу портфолио было решено создать с применением веб-технологий, чтобы отразить свою индивидуальность, что было бы воспринято хорошо, если бы я создавала данное портфолио для резюме. Страницы с включенным в них кодом было решено создать с помощью разметки Markdown, что позволит быстрее создать данные страницы и лучше отобразить программный код, так как Markdown подсвечивает синтаксис выбранного языка программирования, что улучшает восприятие кода.

## 2.2 Реализация выбранной структуры портфолио

Для реализации портфолио я создала новую папку на своем компьютере, в которой у меня будет вестись работа. В этой папке я также создала шесть папок для размещения в них выполненных заданий с именами по шаблону  $lr\{N\}$ , где  $\{N\}$  – это номер работы. И создала одну папку с названием *static* для размещения в ней файлов типа CSS и логотипа ВУЗа.

В папки  $lr\{N\}$  я поместила выполненные задания и приступила к описанию каждой работы. Для этого я создавала в каждой из нужных папок файл с ключевым названием *README.md*. Это файл с разметкой Markdown, который воспринимается сервисом GitHub Pages также, как файл *index.html*, то есть файлом, отвечающим за интерфейс страницы. В итоге получилась структура портфолио следующего содержания:

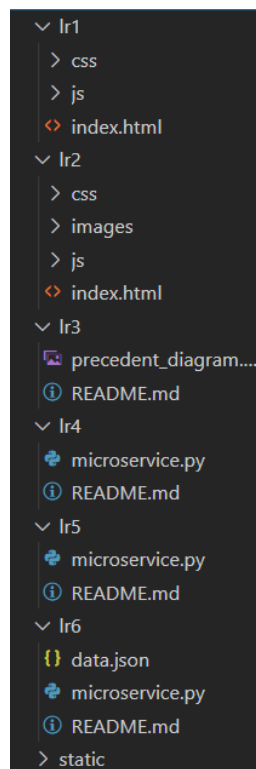


Рисунок 3. Структура папок портфолио, представленная через редактор VS Code

Затем каждый из файлов *README.md* я наполняла по макету, составленному в прошлом шаге (см. рис. 2). В редакторе VS Code это выглядело так, как представлено на рисунке 4.



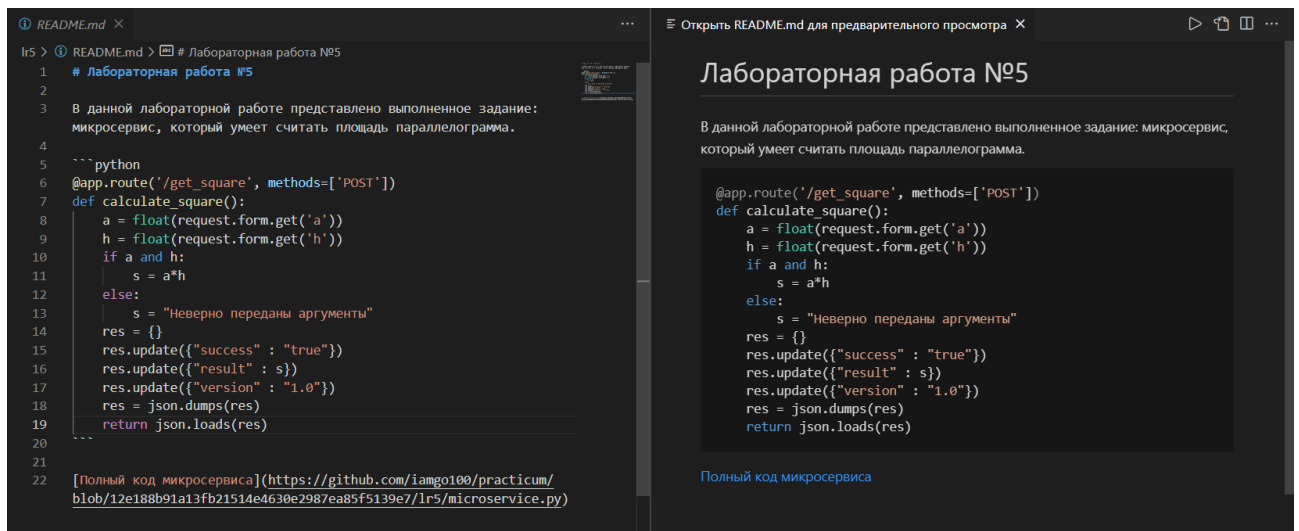


Рисунок 4. README-файл в интерфейсе редактора VS Code

После того как все папки с заданиями и описаниями к ним были готовы, я приступила верстке главной страницы портфолио. Для этого я создала в корневой папке файл *index.html*, а в папке *static* создала файл *index.css* для стилей, а также в папку *static* я скопировала логотип ВУЗа, чтобы реализовать макет, представленный на рисунке 1.

Далее, следуя созданному макету, я сверстала HTML-файл и написала CSS-файл для персонализации дизайна главной страницы; также я подключила CSS-файл к HTML-файлу. В итоге структура файлов в редакторе VS Code выглядела так (см. рис. 5):

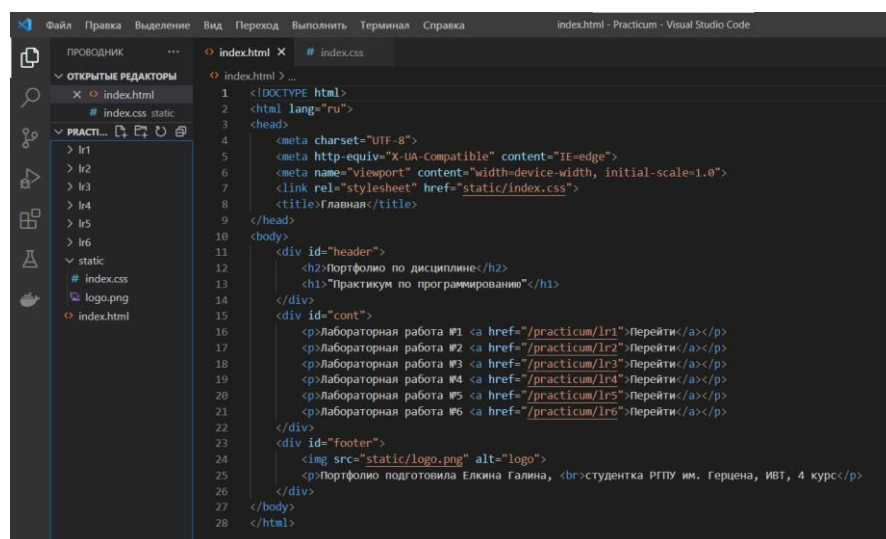
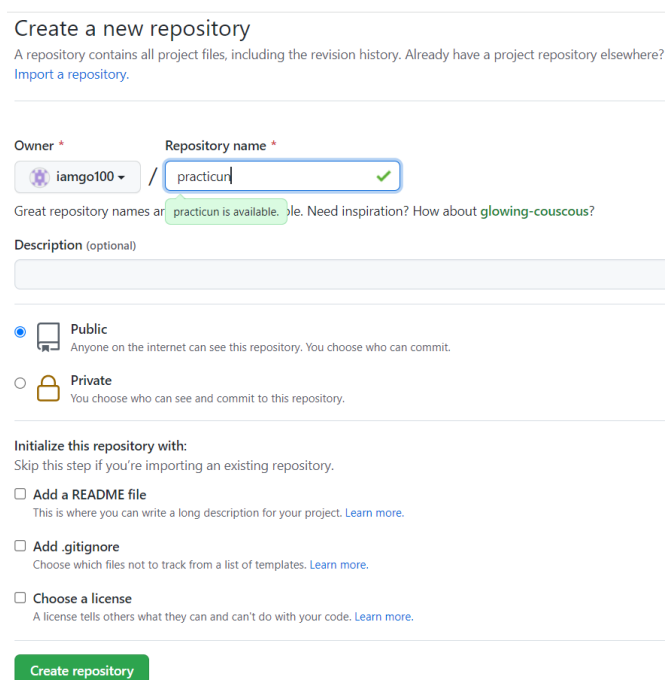


Рисунок 5. HTML-файл и структура папок портфолио

Затем, проверяя результат в браузере через локальный путь к файлам, я добивалась нужного расположения элементов на странице, подбирала цвета, а также исправляла ошибки верстки, чтобы главная страница портфолио выглядела нужным мне образом.

Когда я добилась искомого результата, мне нужно было залить все полученные материалы на выбранный сервис.

Чтобы реализовать портфолио через сервис GitHub Pages, нужно иметь аккаунт на сервисе GitHub. Он у меня есть, поэтому создавать его не нужно было. На аккаунте должен быть свободный репозиторий, в который можно залить все имеющиеся файлы. Для этого я создала новый репозиторий (см. рис. 6):



*Рисунок 6. Создание репозитория для портфолио*

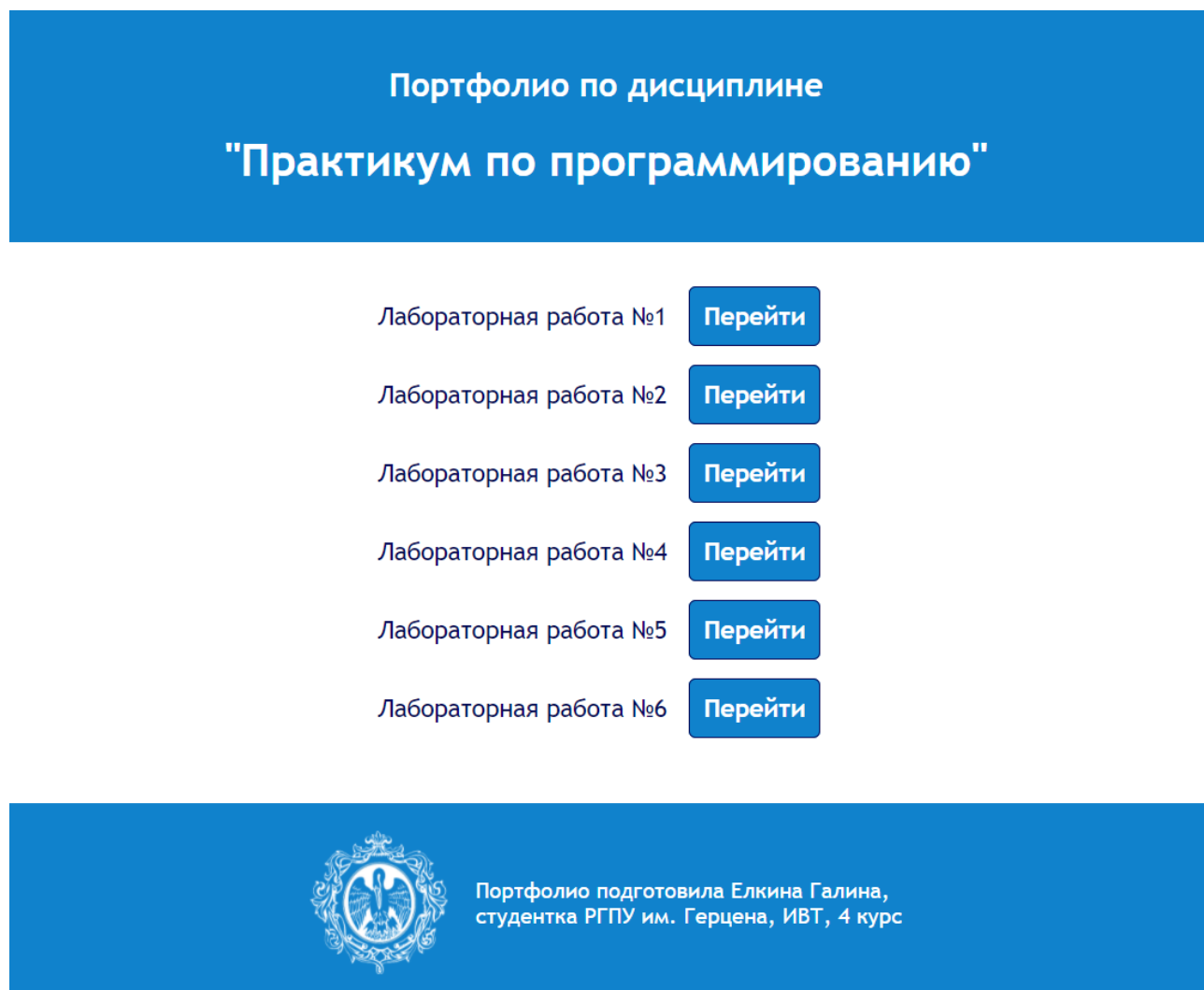
в который с помощью терминала VS Code положила все необходимые файлы (см. рис. 7).

```
PS C:\MyCodes\Study\Practicum> git add .
PS C:\MyCodes\Study\Practicum> git commit -m "create portfolio"
PS C:\MyCodes\Study\Practicum> git push origin main
```

*Рисунок 7. Команды Git в терминале VS Code*

После нескольких коммитов, которые помогли усовершенствовать портфолио, я получила результат, который можно наблюдать по следующей ссылке: <https://iamgo100.github.io/practicum/>

А также ниже, на рисунке 8, можно увидеть получившийся интерфейс главной страницы портфолио.



*Рисунок 8. Главная страница портфолио*

Чтобы разместить свое портфолио по вышеуказанной ссылке, я нашла в настройках (Settings) репозитория вкладку опции (Options), на которой в разделе Pages установила в качестве источника файлов для страниц портфолио ветку *main* репозитория, после чего получила ссылку на опубликованное портфолио.

## **ЗАКЛЮЧЕНИЕ**

В ходе данной работы были тщательным образом проанализированы на предмет нужных функций и составляющих три сервиса для создания веб-портфолио и несколько инструментов, один из которых был выбран и описан в данной курсовой работе. Для выполнения поставленных задач были использованы выбранные мной инструменты и сервисы.

В результате были выполнены все поставленные задачи, и цель курсовой работы была достигнута, что было продемонстрировано выше, в конце главы 2.

## ЛИТЕРАТУРА

1. Как грамотно формировать портфолио? // vc.ru URL: <https://vc.ru/u/95559-roman-stepanov/155000-kak-gramotno-formirovat-portfolio> (дата обращения: 15.12.2021).
2. Руководство по созданию выдающегося портфолио // Хабр URL: <https://habr.com/ru/post/327600/> (дата обращения: 15.12.2021).
3. Успешное портфолио на Behance. Подробное руководство // SkillBox Media | Дизайн URL: <https://skillbox.ru/media/design/behance-super-cases/> (дата обращения: 25.12.2021).
4. Цвет (подсветка синтаксиса) внутри тега HTML `<code>` // CodeRoad URL: <https://coderoad.wiki/1647724/%D0%A6%D0%B2%D0%B5%D1%82-%D0%BF%D0%BE%D0%B4%D1%81%D0%B2%D0%B5%D1%82%D0%BA%D0%B0-%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81%D0%B0-%D0%B2%D0%BD%D1%83%D1%82%D1%80%D0%B8-%D1%82%D0%B5%D0%B3%D0%B0-HTML-code> (дата обращения: 25.12.2021).
5. `<code>`: The Inline Code element // MDN Web Docs URL: <https://developer.mozilla.org/ru/docs/Web/HTML/Element/code> (дата обращения: 25.12.2021).