

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
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**

по дисциплине «Организация электронной образовательной среды»  
Тема: «Разработка веб-портфолио студента-программиста с помощью  
генератора статических сайтов Pelican»

Обучающегося 4 курса  
Войтенко Игоря Александровича

---

Руководитель:  
Старший преподаватель  
Аксютин Павел Александрович

« \_\_\_\_\_ » \_\_\_\_\_ 2022 г.

Санкт-Петербург  
2022

## Оглавление

<b>ВВЕДЕНИЕ</b> .....	3
<b>ГЛАВА 1</b> .....	5
<b>1.1 Теоретический материал</b> .....	5
<b>1.2 Виды реализации портфолио</b> .....	6
<b>1.3 Генератор статических сайтов «Pelican»</b> .....	6
<b>1.4 GitHub pages</b> .....	8
<b>1.5 Вывод</b> .....	8
<b>ГЛАВА 2</b> .....	9
<b>2.1 Подготовка файлов для портфолио</b> .....	9
<b>2.2 Структура портфолио</b> .....	10
<b>2.3 Установка Pelican и наполнение портфолио</b> .....	10
<b>2.4 Размещение на GitHub pages</b> .....	14
<b>2.5 Вывод</b> .....	15
<b>ЗАКЛЮЧЕНИЕ</b> .....	17
<b>ЛИТЕРАТУРА</b> .....	18

## **ВВЕДЕНИЕ**

Процесс обучения начинается с самого детства человека и продолжается до самой старости, поэтому данный процесс занимает значимое место в нашей жизни. Благодаря обучению люди получают необходимый опыт, который им пригодится в той или иной сфере деятельности. Также в ходе процесса обучения, человек получает необходимые навыки для выполнения определенных задач.

В процессе приобретения знаний, студенты ведут конспекты, выполняют различные задания. В результате такой деятельности возникает потребность в хранении накопленных материалов и работ. Помимо хранения информации, у студента должна быть возможность продемонстрировать свои труды, если есть необходимость в этом. Поэтому портфолио будет лучшим вариантом для хранения, полученных знаний.

Для студента-программиста, портфолио играет существенную роль в дальнейшем трудоустройстве.

**Практическая значимость** данной курсовой работы обуславливается тем, что в связи с накоплением материалов, студентам нужно хранить и структурировать информацию, веб-портфолио поможет им в этом.

**Цель курсовой работы:** разработать веб-портфолио студента-программиста.

### **Задачи курсовой работы:**

- Изучить учебную, научную и справочную литературу по теме исследования.
- Раскрыть основные понятия и термины темы.
- Воспользоваться полученными знаниями для реализации практической задачи.

**Объект исследования:** разработка веб-портфолио студента-программиста.

**Предмет исследования:** использование генератора статических сайтов «Pelican».

Данная работа включает в себя следующие части:

- Теоретическая часть.
- Практическая часть.
- Заключительная часть.
- Список литературы.

Теоретическая часть включает в себя основные понятия и термины, затрагивающие данную тему, а практическая часть реализацию поставленной задачи.

## ГЛАВА 1

В этой части курсовой работы хочу ввести основные понятия, которые будут использоваться в рассмотренной мной теме. А также общие вопросы, которые касаются этой темы.

### 1.1 Теоретический материал

**Портфолио** — это собрание или комплект документов, образцов работ, которые студент собрал за время обучения. Содержание портфолио зависит от вида деятельности, программисты выкладывают в портфолио свои наработки (код), фотографы загружают, сделанные фотографии, и тому подобное.

Главная задача портфолио — показать реальные навыки, наработанный опыт и квалификацию человека.

Портфолио дополняет резюме и показывает на наглядном примере, что умеет специалист. Ценность портфолио состоит в том, что оно нарабатывается годами.

**Веб-портфолио** — это веб-ресурс на котором находятся материалы, которые были получены в процессе обучения: результаты лабораторных работ, проектов, совместной деятельности. Веб-портфолио — то же самое, что и портфолио, только с использованием информационных технологий.

Веб-портфолио является компонентом электронной информационно-образовательной среды образовательного учреждения. В тех случаях, когда среда реализуется на технологической основе LMS, такой как Moodle, веб-портфолио, как правило, реализуется средствами самой LMS.

**Генераторы статических сайтов** — программа, которая обрабатывает файлы: картинки, тексты, шаблоны, а после обработки генерирует статический HTML-сайт.

Генераторы статических сайтов работают автоматически, пользователю нужно только заполнять определенные файлы текстом, в основном это файлы с

Markdown разметкой, при желании можно изменять визуальное представление сайта, но для учебных целей это нам не понадобится.

## 1.2 Виды реализации портфолио

Есть несколько видов реализации портфолио, различия между ними состоит в использовании техник и технологий для их создания.

**Бумажное портфолио** — самый простой и понятный вид портфолио. Работа с бумажным портфолио имеет свои недостатки. Во-первых, материал, который получен во время обучения, нужно собирать и структурировать вручную. Во-вторых, передача данного портфолио становится более затруднительным процессом.

**Электронное портфолио** — более эффективный вид ведения портфолио, чем бумажный, удобное хранение информации, например, на информационных носителях, является большим преимуществом данного вида, также электронное портфолио уже намного легче передавать другим людям.

**Веб-портфолио** — следующий этап развития электронного портфолио. Его суть заключается в том, чтобы разместить на специализированном ресурсе свои электронные документы. Очень часто такие ресурсы публичные, что позволяет любому желающему ознакомиться с вашим личным портфолио. Этот вид портфолио уже не является таким простым как предыдущие, так как здесь потребуются некоторые знания технологий, а в каких-то случаях базовые знания программирования.

## 1.3 Генератор статических сайтов «Pelican»

С помощью генератора можно быстро сделать сайт-визитку, резюме или портфолио на основе любого готового HTML/CSS/JS-шаблона.

Самые популярные генераторы статических сайтов (SSG) написаны на Ruby (Jekyll) и Go (Hugo), но есть и несколько интересных решений на Python:

Pelican, Lektor, Nikola и Hyde. Pelican – наиболее простой и в то же время мощный Python-генератор: с его помощью можно создать собственный сайт или блог в считанные минуты.

Pelican — генератор статических сайтов, написанный на Python, для которого не требуется базы данных или серверной логики.

Как он устроен:

- Пользователь создает контент в виде Markdown файлов. Файлы сохраняются в папке content, при этом посты блога располагаются в поддиректории articles, а страницы – в pages. Картинки для постов и страниц хранятся в общей папке images.
- Движок обрабатывает файлы в соответствии с выбранным шаблоном. Как и другие веб-фреймворки на основе Python, Pelican использует шаблонизатор Jinja2. Язык Jinja2 прост, понятен, и очень похож на Python (на котором и написан сам шаблонизатор).

Преимущества и недостатки Pelican:

- Плюсы:
  - Простота редактирования и обновления контента.
  - Мгновенная загрузка готовых страниц и безопасность сайта.
  - Молниеносная смена дизайна.
  - Простота создания новых тем – если вы знакомы с Jinja2, то легко сможете адаптировать готовый HTML/CSS/JS-шаблон для работы с Pelican.
  - Возможность использовать бесплатный хостинг – сайт можно разместить на GitHub Pages и даже в Dropbox.
  - Предельно простая процедура переноса сайта с одного хостинга на другой.
  - Открытый исходный код.
  - Неплохой набор готовых плагинов (по сравнению с другими SGG на Python).
- Минусы:

- Pelican работает медленнее Hugo. Например, генерация блога, состоящего из сотни страниц, займет не менее 10 секунд.
- Нет панели администрирования – все команды вводятся в CMD.
- Выбор тем и плагинов, по сравнению с популярными динамическими CMS (Wordpress), ограничен.
- Нет WYSIWYG-редактора – для создания постов придется освоить Markdown, а для адаптации готовых шаблонов под работу с Пеликаном нужно понимание языка Jinja2 и базовые познания в HTML и CSS.
- Для подключения комментариев, форм и поиска нужны дополнительные манипуляции. Ниже мы рассмотрим, как подключить к Pelican-сайту поиск, комментарии и контактную форму.

## **1.4 GitHub pages**

GitHub Pages — это бесплатный хостинг для статических файлов.

Так как это бесплатный хостинг, его будем использовать, чтобы загрузить необходимые файлы для отображения веб-портфолио.

## **1.5 Вывод**

В результате разобраны основные термины и этапы работы над портфолио. Рассмотрены основные виды его реализации. Изученная теоретическая часть поможет в разработке индивидуального портфолио.



## ГЛАВА 2

В этой части курсовой работы будет проводится разработка веб-портфолио студента-программиста.

### 2.1 Подготовка файлов для портфолио

Первым делом подготовим файлы для портфолио, создадим на ПК папки для определенных дисциплин и загрузим в них необходимые файлы.

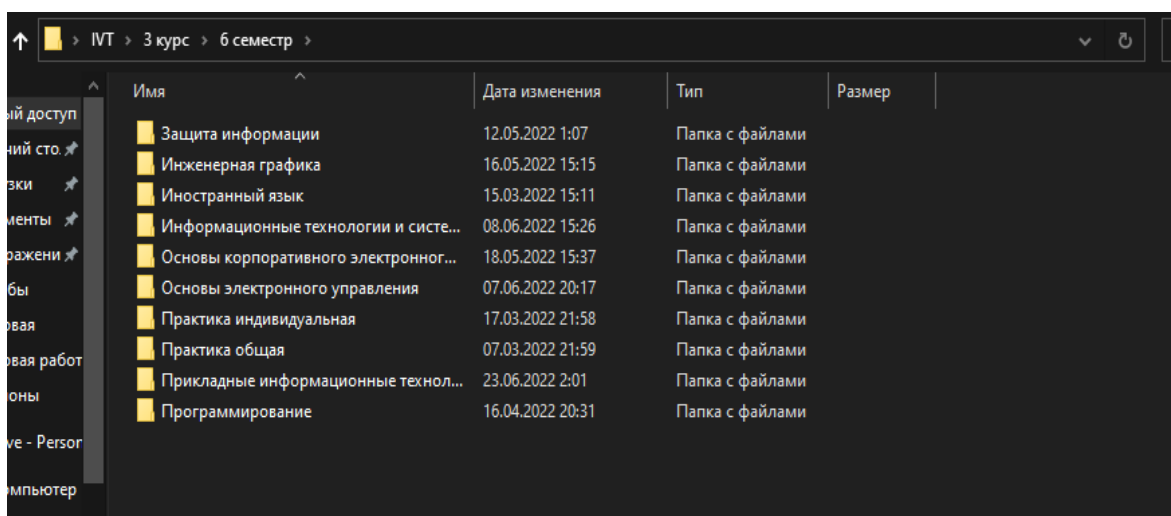


Рисунок 2.1 — пример структуры папок

Следующем шагом будет загрузка файлов в облачное хранилище.

Мой диск > 1 семестр ▾				⌵	ⓘ
Название ↑	Владелец	Последнее изменение	Размер файла		
Информатика	я	4 июл. 2020 г. я	—		
ИТ в математике	я	6 июл. 2020 г. я	—		
ИТ в физике	я	4 июл. 2020 г. я	—		
Курсовая работа_Войтенко	я	4 июл. 2020 г. я	—		
Программирование	я	4 июл. 2020 г. я	—		
Эссе_Карнофф.docx	я	2 дек. 2019 г. я	13 КБ		

Рисунок 2.2 — пример выгрузки файлов в облачное хранилище

Подготовка файлов завершена, выгрузка файлов в облачное хранилище необходимо, чтобы не загружать сами файлы в портфолио, а просто указывать ссылку на файл.

## 2.2 Структура портфолио

После подготовки файлов, разработаем структуру портфолио, то есть навигацию для более удобного и быстрого доступа к файлам портфолио.

В качестве примера я возьму несколько дисциплин: «Серверные веб-технологии», «Компьютерный практикум», «Прикладные ИТ» и «Программирование».



Рисунок 2.3 — структура портфолио

Структура портфолио выглядит просто, в дальнейшем ее можно будет доработать.

Теперь можем перейти к установке Pelican и наполнению портфолио.

## 2.3 Установка Pelican и наполнение портфолио

Установим Pelican при помощи pip:

```
python -m pip install "pelican[markdown]"
```

Рисунок 2.4 — команда установки Pelican

После установки перейдем к созданию проекта

Создадим папку проекта:

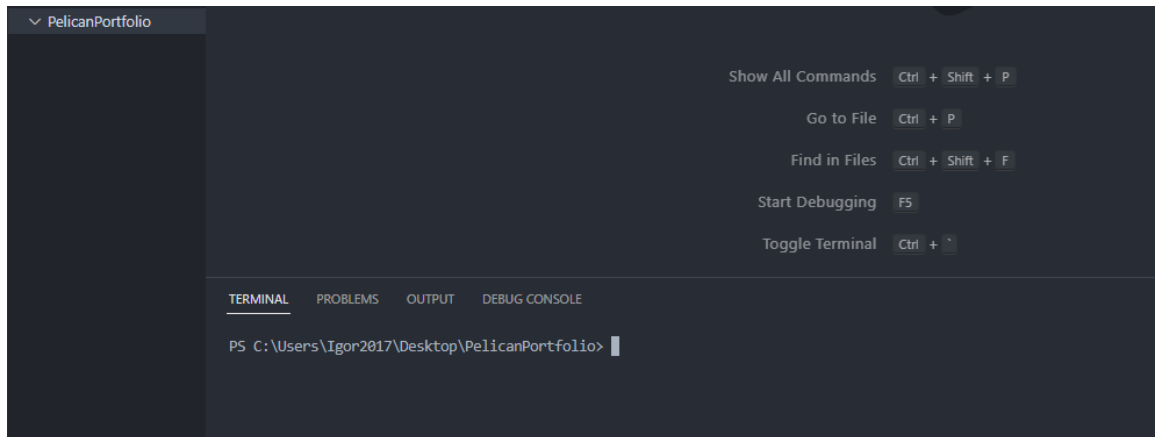


Рисунок 2.5 — созданная папка с проектом

Создадим скелет проекта с помощью pelican-quickstar, которая начинается с вопросов о вашем сайте:

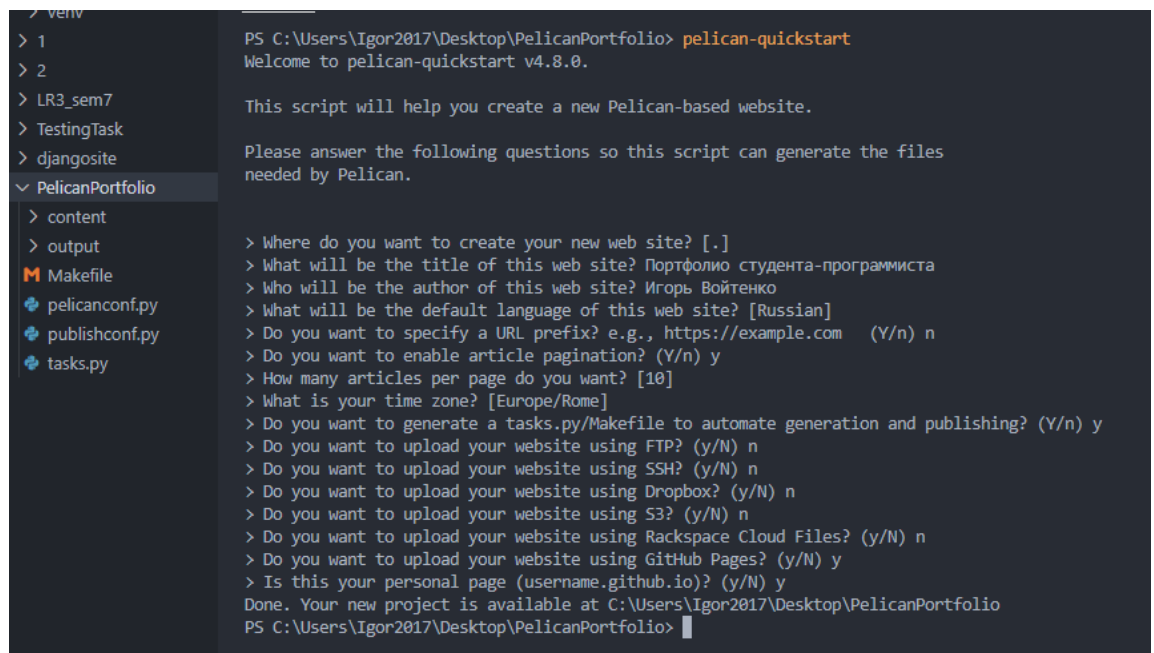


Рисунок 2.6 — создание скелета портфолио после ввода команды

Можем приступить к наполнению портфолио, в директории «content» добавим еще одну директорию «pages», там будут храниться страницы дисциплин в виде Markdown-файлов.

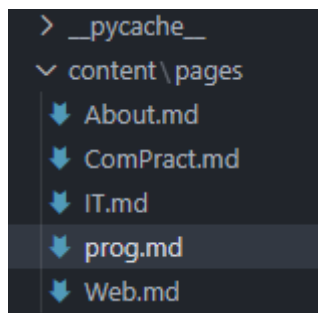


Рисунок 2.7 — md-файлы в директории «pages»

Примерное так будет выглядеть наполнение каждой страницы дисциплины.

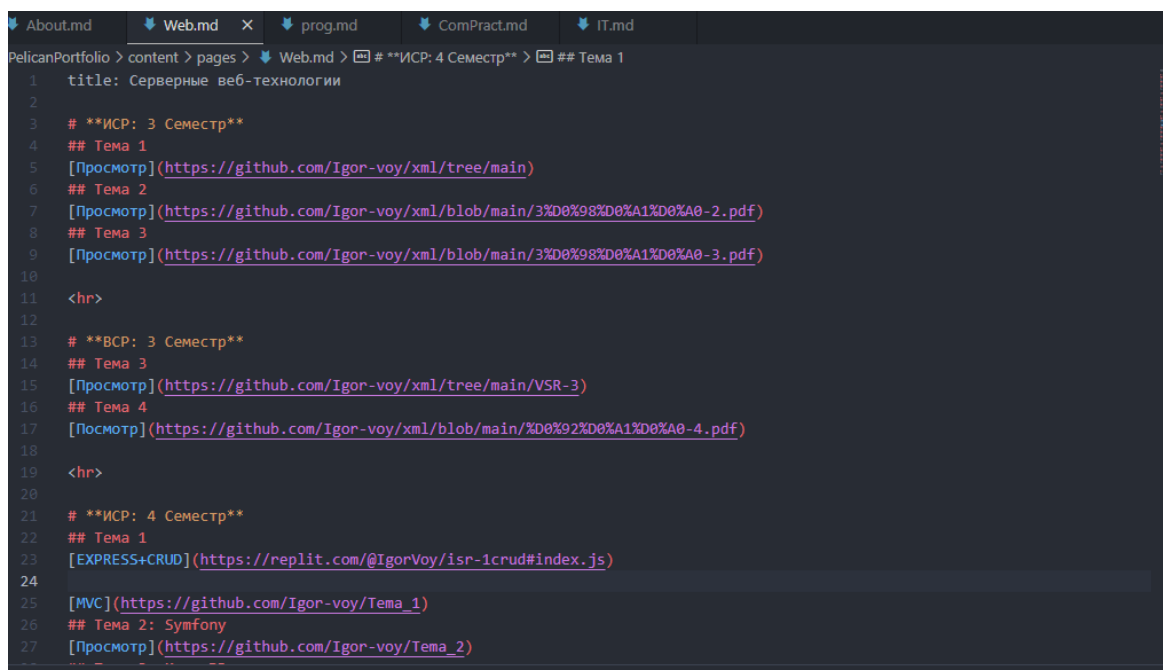


Рисунок 2.8 — страница одной из дисциплин (Markdown разметка)

После наполнения контентом нашего портфолио, нужно сохранить изменения скелета портфолио.

Для того чтобы сохранить изменения контента в командной строке пишем команду:

```
pelican content
```

Рисунок 2.9 — команда создания сайта

После выполнения команды, контент сохранился:

```
PS C:\Users\Igor2017\Desktop\PelicanPortfolio> pelican content
[21:08:41] WARNING Docutils has no localization for 'russian'. Using 'en' instead.
WARNING Watched path does not exist: C:\Users\Igor2017\Desktop\PelicanPortfolio\content\images
Done: Processed 0 articles, 0 drafts, 0 hidden articles, 5 pages, 0 hidden pages and 0 draft pages in 1.40 seconds.
PS C:\Users\Igor2017\Desktop\PelicanPortfolio>
```

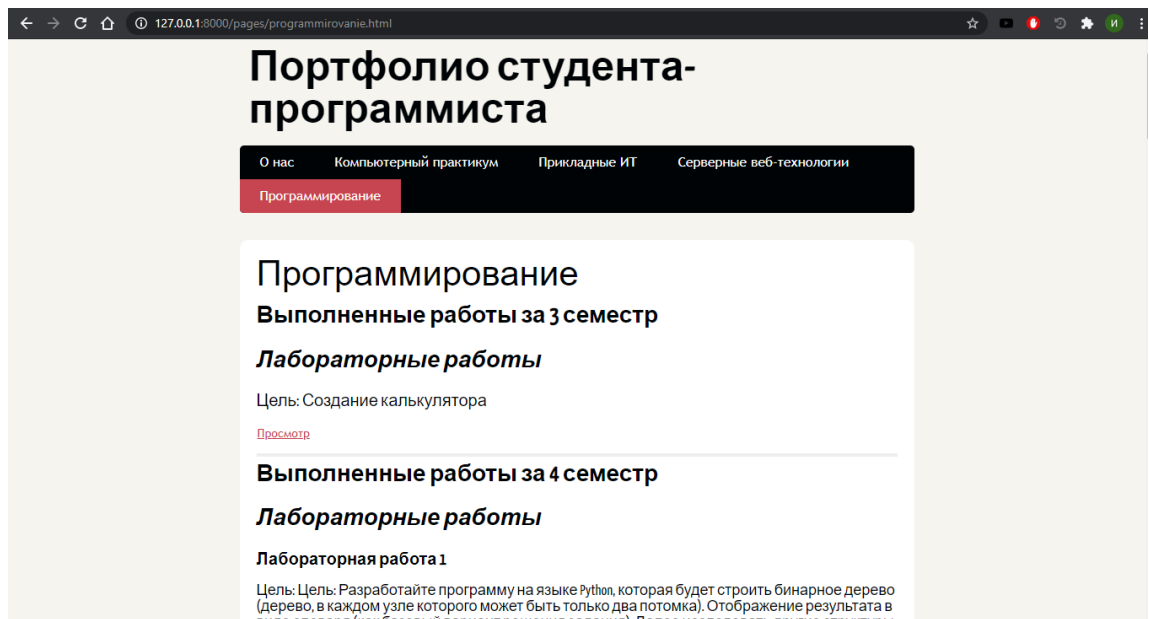
Рисунок 2.10 — создание сайта/сохранение контента

Теперь запустим наше портфолио на localhost, для этого введем команду:

```
pelican --listen
```

Рисунок 2.11 — команда для запуска веб-сервера

Переходим в браузер и смотрим, что у нас получилось после проделанной работы.



С учетом того, что мы заполняли только Markdown-файлы, портфолио выглядит очень хорошо, при желании можно воспользоваться плагинами и другими шаблонами.

Теперь можем приступить к загрузке нашего портфолио на GitHub pages.

## 2.4 Размещение на GitHub pages

Для начала создадим репозиторий, в моем случае назову его IgorVoytenko.github.io.

Можно разместить содержимое директории при помощи командной строки, но мы сделаем проще. Из директории «output» выгрузим все содержимое в созданный репозиторий.

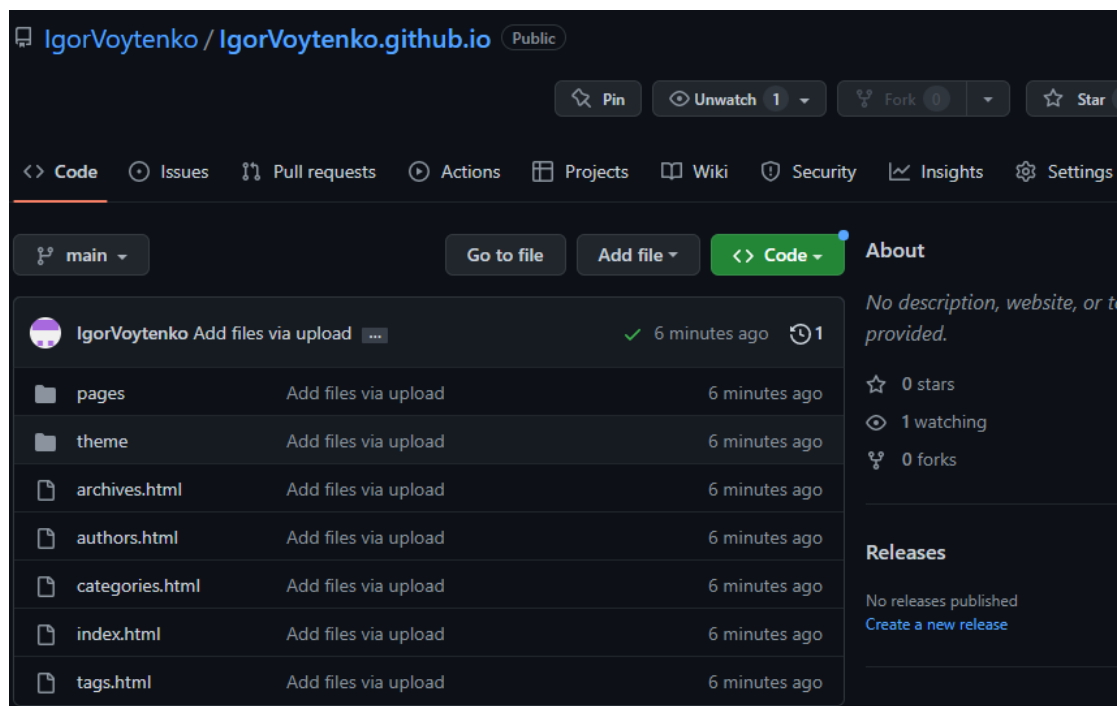


Рисунок 2.13 — загруженные файлы

Все готово, можем перейти по названию нашего репозитория.

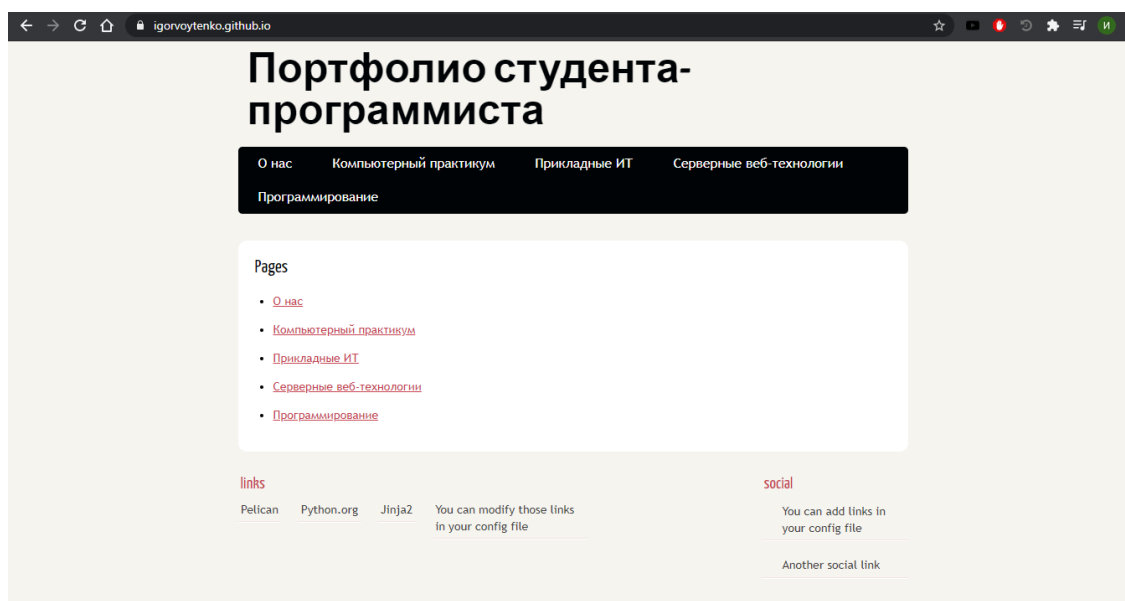


Рисунок 2.14 — портфолио на GitHub pages

## 2.5 Вывод

На основе полученных знаний о средствах и способах создания портфолио была создана электронная интернет версия портфолио студента-программиста.

Его создание включало в себя следующие этапы: сбор документальной базы, сложившейся в период обучения; проектирование структуры портфолио, ее создание и размещение в сети; разработка вспомогательной системы навигации по документам; проведен анализ использованного при этом сервиса.



## **ЗАКЛЮЧЕНИЕ**

В ходе курсовой работы было разработано веб-портфолио студента-программиста, сама разработка поэтапно была продемонстрирована текстом и сопровождающими рисунками. Также был рассмотрен теоретический материал по теме, были выделены несколько видов портфолио и их различия.

Одной из главных задач данной работы было знакомство с генератором статических сайтов «Pelican», хоть он и не является настолько популярным как Hugo, но все равно имеет эффективный функционал. Благодаря данному генератору, человек, который далек от программирования, самостоятельно может создать простое портфолио.

## ЛИТЕРАТУРА

1. Шакон С. Pro Git: учебное пособие —2012. —№ 1.— С. 18-21.
2. Чернявская, А. П., Гречин, Б. С. Современные средства оценивания результатов обучения: учебно-методическое пособие. -Ярославль: Изд-во ЯГПУ. –2008. –98 с.
3. Чакон С., Штрауб Б. Git для профессионального программиста. — Питер, 2017. — 496 с.
4. Фёдоров, Д. Ю. Программирование на языке высокого уровня Python. — Москва: Издательство Юрайт, 2022. — 210 с.
5. Стивен Шафер. HTML, XHTML и CSS. Библия пользователя, 5-е издание = HTML, XHTML, and CSS Bible, 5th Edition. — М.: «Диалектика», 2010. — 656 с.
6. Панюкова С. В., Гостин А. М., Кулиева Г. А, Самохина Н. В. Создание и ведение веб-портфолио преподавателя. Методические рекомендации: учеб. пособие.-Рязань.: «Рязанский государственный радиотехнический университет», 2014 г. –65 с
7. Локк Д. Основы управления проектами: учебное пособие. —2004— № 1.— С. 117-128.
8. Купер А. Об интерфейсе. Основы проектирования взаимодействия. —2009—№ 2.— С. 339-340.
9. Куликов С. Тестирование программного обеспечения. Базовый курс: учебное пособие. —2020—№ 2.— С. 18-25.
- 10.Иванова Г.С. Технология программирования: учебное пособие. — 2016—№ 3.— С 20-22.
- 11.Ехлаков Ю.П. Управление программного проекта: учебное пособие. —2014 —№ 1.— С. 6, 10-12, 17-20, 26-41, 127-128.
- 12.Государев И. Б. Введение в веб-разработку на языке JavaScript: учебное пособие / И. Б. Государев. — Электрон. дан. — Санкт-Петербург: Лань, 2019. — 144 с.

- 13.ГОСТ 15.016-2016 «Система разработки и постановки продукции на производство. Техническое задание. Требования к содержанию и оформлению.» —Пкт. 6.1.
- 14.Беркун С. Искусство управления IT-проектами: учебное пособие. — 2014—№ 1.— С. 37-39.
- 15.Андерсон Д. Канбан: альтернативный путь в Agile: учебное пособие. —2016—№ 1.— С. 9.