

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

Институт информационных технологий и технологического образования
Кафедра информационных технологий и электронного обучения

КУРСОВАЯ РАБОТА

ИСПОЛЬЗОВАНИЕ ГЕНЕРАТОРА СТАТИЧЕСКИХ САЙТОВ HUGO
И ТЕХНОЛОГИИ PYSCRIPT ДЛЯ РАЗРАБОТКИ ВЕБ-ПОРТФОЛИО
СПЕЦИАЛИСТА ПО ДИСЦИПЛИНЕ «ПРОГРАММИРОВАНИЕ»
Направление подготовки: «Информатика и вычислительная техника»

Руководитель:
старший преподаватель
_____ Аксютин П. А.
« » _____ 2022 г.

Автор работы:
студент 4 курса
_____ Мельников Ф. В.
« » _____ 2022 г.

Санкт-Петербург
2022

Оглавление

Введение.....	3
Глава 1. Структура и инструменты разработки веб-портфолио	4
Глава 1.1. Структура и содержание портфолио по дисциплине «Программирование»	4
Глава 1.2. Инструменты для разработки веб-портфолио.....	5
Глава 1.3. Генератор статических сайтов Hugo	6
Глава 1.4. Технология PyScript	7
Глава 2. Реализация веб-портфолио специалиста.....	9
Глава 2.1. Создание шаблона для генератора статических сайтов Hugo	9
Глава 2.2. Размещение заданий в веб-портфолио.....	9
Заключение	14
Список источников	15

Введение

Портфолио – это набор достижений студента, который позволяет оценить результаты обучающегося в ходе учебной деятельности. Оно включает работы, выполненные студентом за весь период обучения.

Портфолио имеет большое значение для оценки профессиональных навыков студента. Оно служит показателем сформированности ключевых профессиональных компетенций.

Кроме того, портфолио студента может быть использовано при трудоустройстве для оценки знаний и навыков будущего сотрудника.

Портфолио может способствовать усилению мотивации студента к учебным и личностным достижениям, совершенствованию профессиональных компетенций, самообразованию.

В связи с распространённостью веб-технологий целесообразным является разработка портфолио в виде веб-сайта. Такой ресурс является структурированным, открытым и гибким.

Веб-портфолио – это веб-базируемый ресурс, используемый для хранения и мониторинга прогресса обучающегося. Важной особенностью такого ресурса является независимость ресурса от платформы.

Разработка портфолио в виде веб-сайта демонстрирует навыки студента в сфере веб-технологий.

Цель работы – разработка веб-портфолио по дисциплине «Программирование» с применением современных веб-технологий.

Задачи:

- определить структуру заданий и портфолио по дисциплине «Программирование»;
- реализовать шаблон веб-портфолио;
- разместить задания в портфолио.

Глава 1. Структура и инструменты разработки веб-портфолио

Глава 1.1. Структура и содержание портфолио по дисциплине «Программирование»

В результате изучения дисциплины «Программирование» в соответствии с рабочей программой дисциплины в РГПУ им. А. И. Герцена по направлению 09.03.01 «Информатика и вычислительная техника» обучающийся должен обладать следующими компетенциями:

- способностью осваивать методики использования программных средств для решения практических задач;
- способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования.

В рамках дисциплины «Программирование» изучаются языки C и Python.

В ходе изучения дисциплины студенты выполняют лабораторные работы, инвариантные и вариативные самостоятельные работы, которые включаются в портфолио.

Основная часть заданий предполагает разработку приложений на языке Python.

Портфолио может включать отдельные разделы для каждого семестра и отдельные разделы для разных типов заданий.

Структура портфолио в рамках одного семестра представлена на рис. 1.

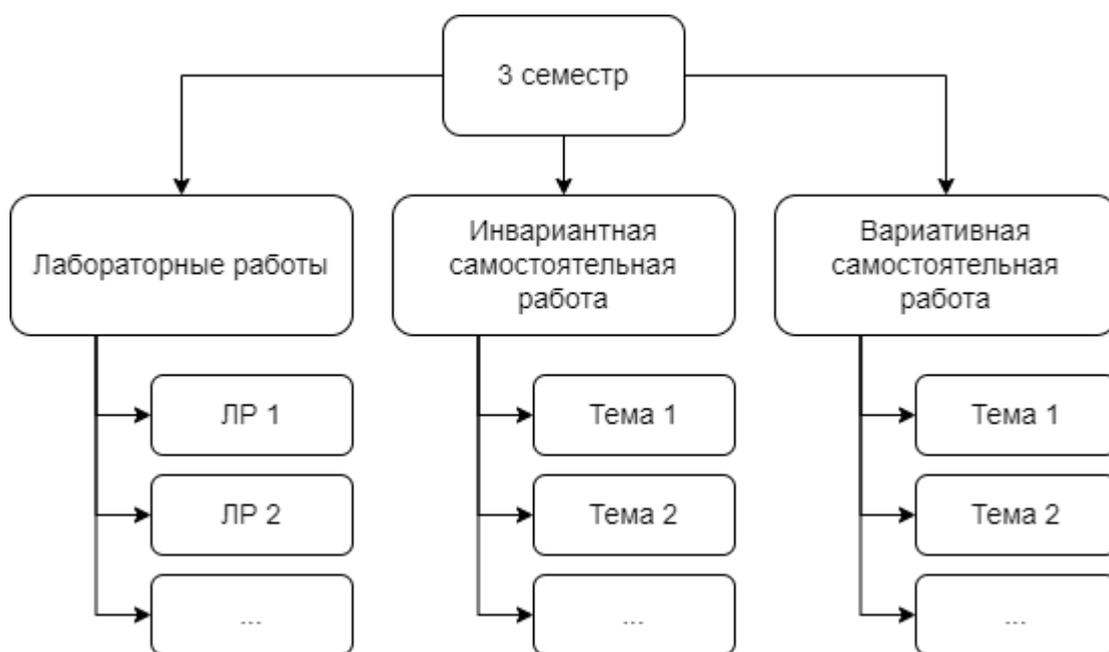


Рисунок 1 – структура портфолио

Глава 1.2. Инструменты для разработки веб-портфолио

Веб-портфолио может быть выполнено в виде динамического или статического веб-сайта.

Динамический сайт состоит из изменяемых страниц. Генерация страниц динамического сайта производится при поступлении запроса от клиента. Для интерпретации страниц на стороне сервера используются такие языки программирования, как PHP, Java, ASP.NET и др. Как правило, при этом производится взаимодействие с базой данных.

Статический сайт состоит из статичных HTML-страниц и других файлов, например, изображений, аудио- и видеофайлов. Генерация страниц производится заранее, сервер отдаёт клиенту готовые файлы.

Статические сайты демонстрируют более высокую скорость загрузки страниц и являются более безопасными по сравнению с динамическими.

Разработка статического сайта «вручную» является трудоёмкой и нецелесообразной. Как правило, в современной разработке статические сайты создаются с помощью генераторов статических сайтов.

Генератор статических сайтов – это программа, которая предназначена для сборки статических сайтов на основе исходных данных и шаблонов.

Существует большое количество генераторов статических сайтов, среди которых можно выделить Hugo, Pelican, Jekyll. Они обладают особенностями, обусловленными экосистемой языка программирования.

При использовании инструментов непрерывной интеграции и непрерывного развёртывания, таких как GitHub Actions или GitLab CI/CD, можно автоматизировать процесс сборки и публикации сайта.

Для разработки веб-портфолио выбран генератор статических сайтов Hugo, поскольку он обеспечивает высокую скорость генерации страниц сайта, а также поддерживает вставку шорткодов для добавления элементов в файлах Markdown. Кроме того, для Hugo существует библиотека готовых тем.

Задания по дисциплине «Программирование» направлены на разработку приложений на языке Python. Некоторые приложения могут быть запущены в среде веб-браузера с помощью технологии PyScript.

Глава 1.3. Генератор статических сайтов Hugo

Hugo – это генератор статических сайтов с открытым исходным кодом, написанный на языке программирования Go.

Hugo использует файловую структуру для организации содержимого. Все файлы, необходимые для сборки сайта, находятся в одной директории.

Hugo обеспечивает высокую скорость генерации страниц. Разработчики заявляют о скорости генерации менее 1 мс на страницу, а средний веб-сайт генерируется за 1 секунду.

Содержимое страниц описывается на языке разметки Markdown. Дополнительные элементы могут быть добавлены на страницы с помощью шорткодов (shortcodes). Также Hugo предоставляет инструменты для работы с файлами в форматах JSON и CSV.

Важной особенностью Hugo является использование синтаксиса шаблонов Go Templates.

Кроме того, разработчики Hugo подготовили отдельный портал с готовыми к использованию темами, где можно найти как простые темы с базовым дизайном и набором функций, так и сложные – с широким выбором шаблонов, комментариями и адаптивностью.

Глава 1.4. Технология PyScript

PyScript – это фреймворк, который позволяет создавать многофункциональные приложения Python, выполняемые в веб-браузере, с использованием HTML. Он использует Pyodide, WebAssembly и другие веб-технологии.

WebAssembly (Wasm) – язык программирования низкого уровня для стековой виртуальной машины, спроектированный как портативная цель компиляции для высокоуровневых языков. Стековая виртуальная машина, исполняющая инструкции бинарного формата wasm, может быть запущена как в среде браузера, так и в серверной среде.

Pyodide – это реализация интерпретатора CPython для WebAssembly.

Библиотека PyScript предоставляет HTML-теги для встраивания и выполнения кода Python в браузерной среде.

PyScript позволяет использовать большое количество пакетов Python, например, numpy, pandas, scikit-learn и др.

Возможно применение PyScript совместно с JavaScript. Обеспечивается связь между объектами и пространствами имён Python и JavaScript.

Также PyScript предоставляет готовые компоненты пользовательского интерфейса: кнопки, контейнеры, текстовые поля.

Применение PyScript позволяет создать интерактивную версию портфолио.

По сравнению с интерпретатором CPython, применение PyScript характеризуется рядом ограничений.

При отправке HTTP-запросов в браузере нельзя использовать модули `urllib.request`, `socket`. Для выполнения запросов в PyScript необходимо модифицировать исходный код и использовать асинхронные функции. Также некоторые запросы могут не быть выполнены из-за ограничений политики безопасности браузера. Это делает нецелесообразным и в некоторых случаях (например, при отправке программой запросов по протоколу HTTP) невозможным размещение интерактивной версии задания в портфолио.

Также невозможно разместить интерактивную версию программ, имеющих графический интерфейс пользователя (например, с использованием библиотеки `tkinter`).

В браузере невозможно создание потоков и процессов, поэтому задания, направленные на разработку программ с использованием модулей `threading`, `multiprocessing` и др. невозможно разместить в веб-портфолио в интерактивной версии.

Глава 2. Реализация веб-портфолио специалиста

Глава 2.1. Создание шаблона для генератора статических сайтов Hugo

Разработан шаблон страниц для Hugo с использованием библиотеки Bootstrap, включающей шаблоны оформления для компонентов веб-интерфейса.

Для добавления заданий были разработаны шорткоды (shortcodes) `rusconfig`, `code` и `button`. Шорткоды в Hugo представляют собой фрагменты, которые позволяют добавлять содержимое шаблона на страницу.

На странице заданий расположен терминал, в котором отображается вывод программ.

Глава 2.2. Размещение заданий в веб-портфолио

Для добавления новой страницы необходимо создать текстовый файл в директории `content`. Содержимое страницы описывается на языке Markdown. Пример разметки страницы приведён на рис. 2.

```
---
title: "Портфолио (demo)"
---

## Программирование (6 семестр)

### Вариативная самостоятельная работа

#### Тема 1

1.2 Разработка скрипта, вычисляющего произведение матриц произвольной
размерности с использованием Numba и замером времени вычисления.
Создание отчета по результатам анализа производительности.

{{< button "Ссылка на репозиторий" "https://github.com/balrundev/programming-tasks/tree/main/6th-semester/var/theme-1" >}}
```

Рисунок 2 – исходный код страницы

Конфигурация PyScript описывается в блоке `rusconfig` (рис. 3). Указываются используемые пакеты Python и файлы, которые используются в программе и должны быть размещены в среде выполнения.

```
{{< pyconfig >}}
packages = ["numpy", "matplotlib"]

[[fetch]]
files = ["data.csv"]
{{< /pyconfig >}}
```

Рисунок 3 – описание конфигурации PyScript

Текст написан с использованием возможностей языка разметки Markdown. Блок code используется для вставки блока кода приложения на страницу. Блок button позволяет добавить кнопку на страницу (рис. 4).

Тема 2

2.1 Написать программу, позволяющую выполнять подсчет слов в тексте, а также вычислять размер (в символах) каждого слова. Используйте для возвращения результатов подсчета механизм генераторов.

```
{{< code filename="theme-2/main.py" >}}
```

```
{{< button "Ссылка на репозиторий" "https://github.com/balrundev/programming-tasks/tree/main/6th-semester/var/theme-2" >}}
```

Рисунок 4 – описание задания

Другие задания добавляются аналогично. Общий вид страницы портфолио представлен на рис. 5.

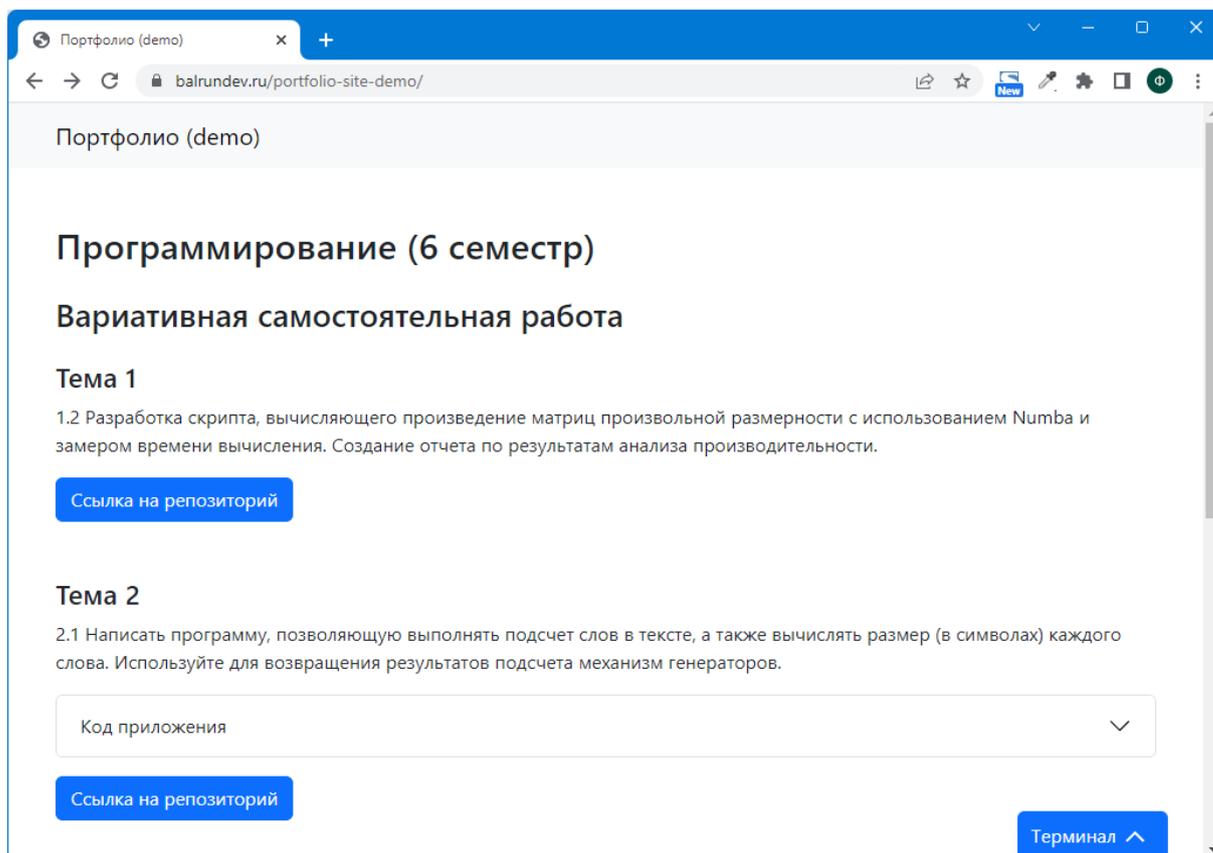


Рисунок 5 – страница веб-портфолио

На странице размещены задания одного семестра: лабораторные работы, задания вариативной и инвариантной самостоятельной работы. Приведён текст задания со ссылками на репозиторий.

При нажатии на кнопку «Ссылка на репозиторий» производится переход на страницу просмотра исходного кода программы в репозитории GitHub.

Возможен просмотр и запуск некоторых программ. На рис. 6 представлен код программы, разработанной в ходе выполнения задания вариативной самостоятельной работы.

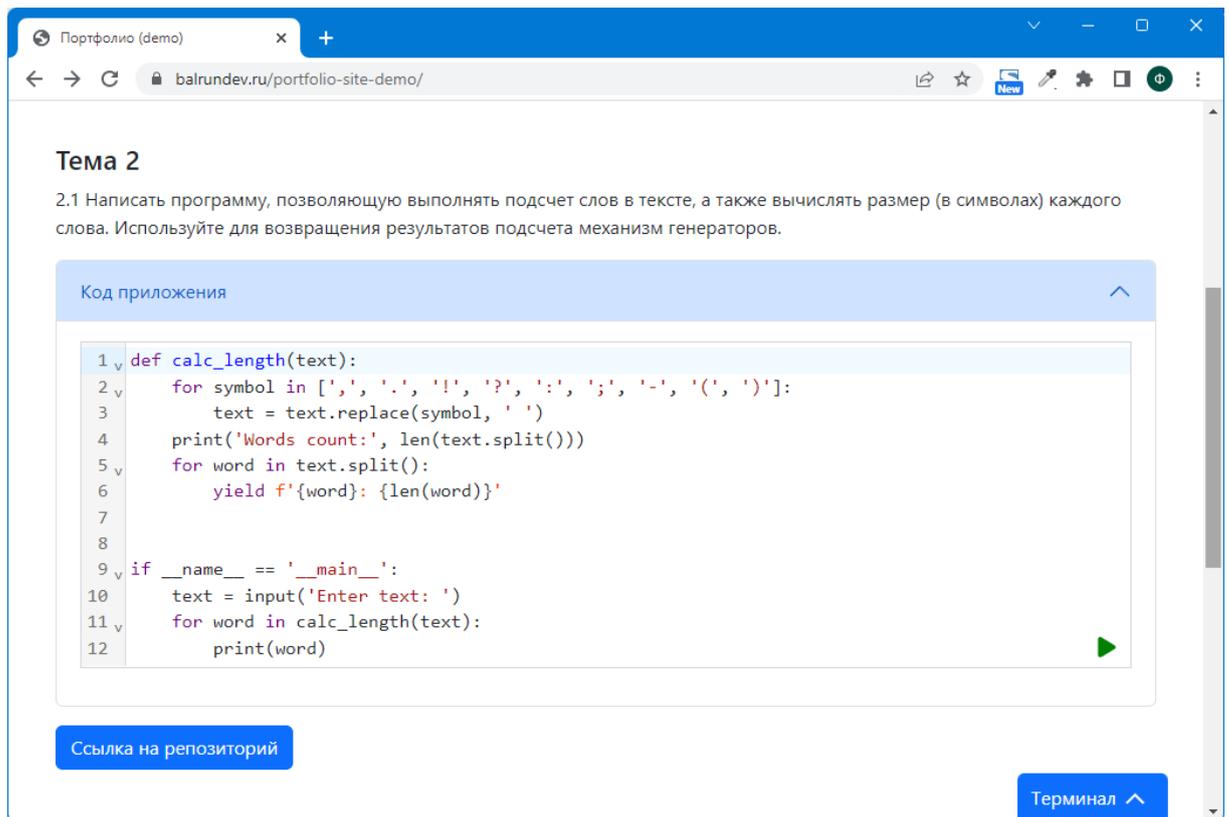


Рисунок 6 – код программы

Если в программе производится ввод данных с клавиатуры, PyScript выводит окно для ввода (рис. 7):

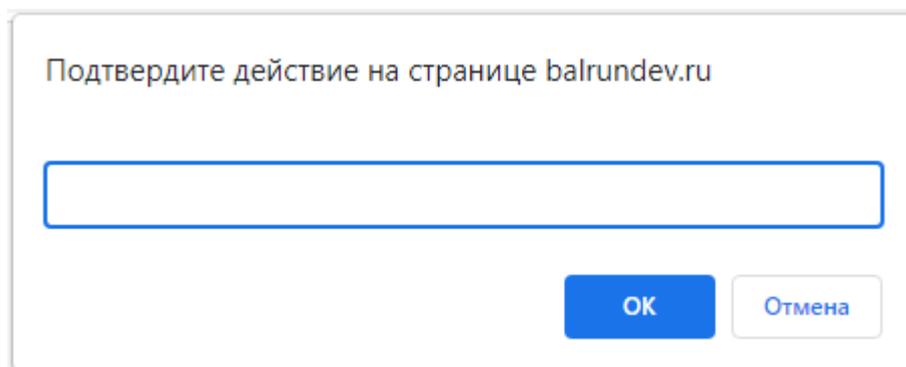
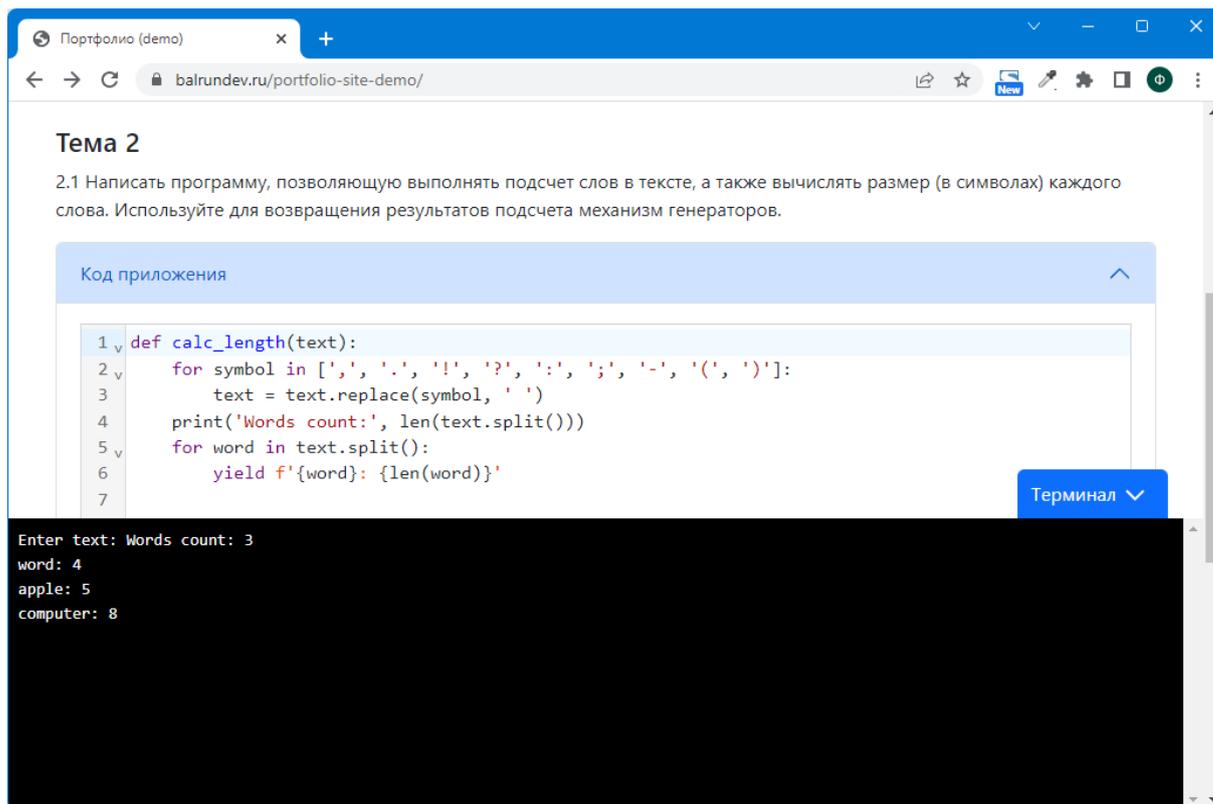


Рисунок 7 – окно ввода данных

Вывод программы отображается в терминале (рис. 8):



The screenshot shows a web browser window with the address bar displaying 'balrundev.ru/portfolio-site-demo/'. The page content includes a heading 'Тема 2' and a task description: '2.1 Написать программу, позволяющую выполнять подсчет слов в тексте, а также вычислять размер (в символах) каждого слова. Используйте для возвращения результатов подсчета механизм генераторов.' Below this is a code editor titled 'Код приложения' containing the following Python code:

```
1 def calc_length(text):
2     for symbol in [',', '.', '!', '?', ':', ';', '-', '(', ')']:
3         text = text.replace(symbol, ' ')
4     print('Words count:', len(text.split()))
5     for word in text.split():
6         yield f'{word}: {len(word)}'
7
```

At the bottom right of the code editor is a 'Терминал' button. Below the code editor is a terminal window with the following output:

```
Enter text: Words count: 3
word: 4
apple: 5
computer: 8
```

Рисунок 8 – вывод программы

После добавления страниц или обновления содержимого необходимо выполнить сборку сайта.

Существует большое количество способов размещения статического сайта. Например, сайт может быть размещён на виртуальном сервере (VPS). Другим способом является размещение на платформе GitHub Pages.

Заключение

В ходе работы было разработано веб-портфолио с использованием генератора статических сайтов Hugo и технологии PyScript.

Генератор статических сайтов позволяет автоматизировать процесс сборки страниц. Технология PyScript позволяет выполнять код Python в среде веб-браузера.

Применение рассмотренных технологий позволяет создать интерактивное веб-портфолио по дисциплине «Программирование».

Список источников

1. Государев И. Б. Введение в веб-разработку на языке JavaScript: Учебное пособие. – СПб.: Издательство «Лань», 2019. – 144 с.
2. Camden R., Rinaldi B. Working with Static Sites. – O'Reilly Media, Inc., 2017. – 195 p.
3. Jain A. Hugo in Action. – Manning Publications, 2022. – 488 p.
4. Никитинская В. М., Сокуренок Ю. А., Государев И. Б. Обзор современных практик создания статических веб-сайтов // Альманах научных работ молодых учёных Университета ИТМО. Том 7. СПб: Университет ИТМО, 2018. С. 229-231.
5. Никитинская В. М., Государев И. Б. Классификация инструментов генерации статических веб-сайтов // Альманах научных работ молодых учёных Университета ИТМО. Том 1. СПб: Университет ИТМО, 2019. – С. 216-219.
6. Гостин А. М., Панюкова С. В. Создание и ведение карьерного веб-портфолио студента // Высшее образование в России. 2014. № 4. С. 126-130.
7. Зеер Э. Ф., Степанова Л. Н. Портфолио как инструментальное средство самооценивания учебно-профессиональных достижений студентов. // Образование и наука. 2018. № 6. С. 139-157.
8. Бурняшов Б. А. Программное обеспечение электронного портфолио студентов российской высшей школы. Открытое образование. 2021. 25(6):24-35.
9. Diaz C. Using Static Site Generators for Scholarly Publications and Open Educational Resources // Code4Lib Journal. Issue 42, 2018. URL: <https://journal.code4lib.org/articles/13861> (дата обращения: 09.12.2022).
10. The world's fastest framework for building websites | Hugo [Электронный ресурс]. URL: <https://gohugo.io/> (дата обращения: 10.12.2022).
11. How to Build a Blazing Fast Static Site With Hugo [Электронный ресурс]. URL: <https://kinsta.com/blog/hugo-static-site/> (дата обращения: 11.12.2022).

12. A Guide to Using Hugo [Электронный ресурс]. URL: <https://strapi.io/blog/guide-to-using-hugo-site-generator> (дата обращения: 11.12.2022).
13. PyScript | Run Python in your HTML [Электронный ресурс]. URL: <https://pyscript.net/> (дата обращения: 15.12.2022).
14. A First Look at PyScript: Python in the Web Browser [Электронный ресурс]. URL: <https://realpython.com/pyscript-python-in-browser/> (дата обращения: 15.12.2022).
15. How to Use PyScript – A Python Frontend Framework [Электронный ресурс]. URL: <https://www.freecodecamp.org/news/pyscript-python-front-end-framework/> (дата обращения: 15.12.2022).