**ВВЕДЕНИЕ**

Существуя в период высокой конкуренции приложения и веб-сервисов за аудиторию, мне стало интересно попробовать свои силы в создании собственного проекта. Целью моего проекта является создание самостоятельно-существующего сайта. Главной функцией его будет возможность отправки файлов различных типов (txt, docx, xlsx, cvs и другие) между пользователями ресурса.

**ОСНОВНАЯ ЧАСТЬ**

**ТЕХНОЛОГИЧЕСКАЯ ЗАДАЧА**

Наиболее интересной задачей мне кажется идея с отказом от какой-либо регистрации пользователей ресурса. Связь между отправителем и получателем планирую реализовать через случайно генерируемый код. Он создается при отправке файла и участвует в получении.

Минимальная архитектура ресурса представляет из себя набор следующих веб-страниц:

1. Главная (стартовая) страница. От сюда начинается ветвление проекта на 2 части в зависимости от цели человека, зашедшего сюда.

2.1 Страница отправления файла. Здесь реализуется загрузка файла, генерация кода и отправка файла.

2.2 Страница получения файла. Здесь пользователь использует известный ему код, который вводится в поле.

3.1 Вторая страница отправления файла. Здесь появляется информация об успешной отправлении файла.

3.2 Вторая страница получения файла. Здесь после использования правильного кода в пункте 2.2 появляется появляется файл с возможность скачивания на устройство получателя.

* 1. **Список решаемых задач**

В проекте ставятся задачи:

* по созданию нескольких связанных друг с другом веб-страниц,
* по созданию функции с генерацией кода,
* по связи отправленного файла с полученным
  1. Подключаемые модули, функции, платформы

Работа над проектом будет происходить на языке программирования Python в среде разработки Pycharm.

Для работы с поставленной задачей первого пункта будут использованы файлы в расширении HTML и CSS. HTML файлы будут содержать в себе все элементы находящиеся на той или иной веб-странице, а CSS файлы, в свою очередь, будут содержать информацию о стилях элементов из HTML файла.

Для работы над проектом были дополнительно установлены модули Flask и Django.

Также планировалось использовать модули speedtest и speedtest-cli для предоставления пользователям возможности узнать скорость их интернета на главной странице сайта. Однако от реализации и внедрения пришлось отказаться.

**РЕАЛИЗУЕМАЯ АРХЕТИКТУРА**

1. **Модули программы, функции, структура данных**

Расположение файлов выведено на рисунке 1.

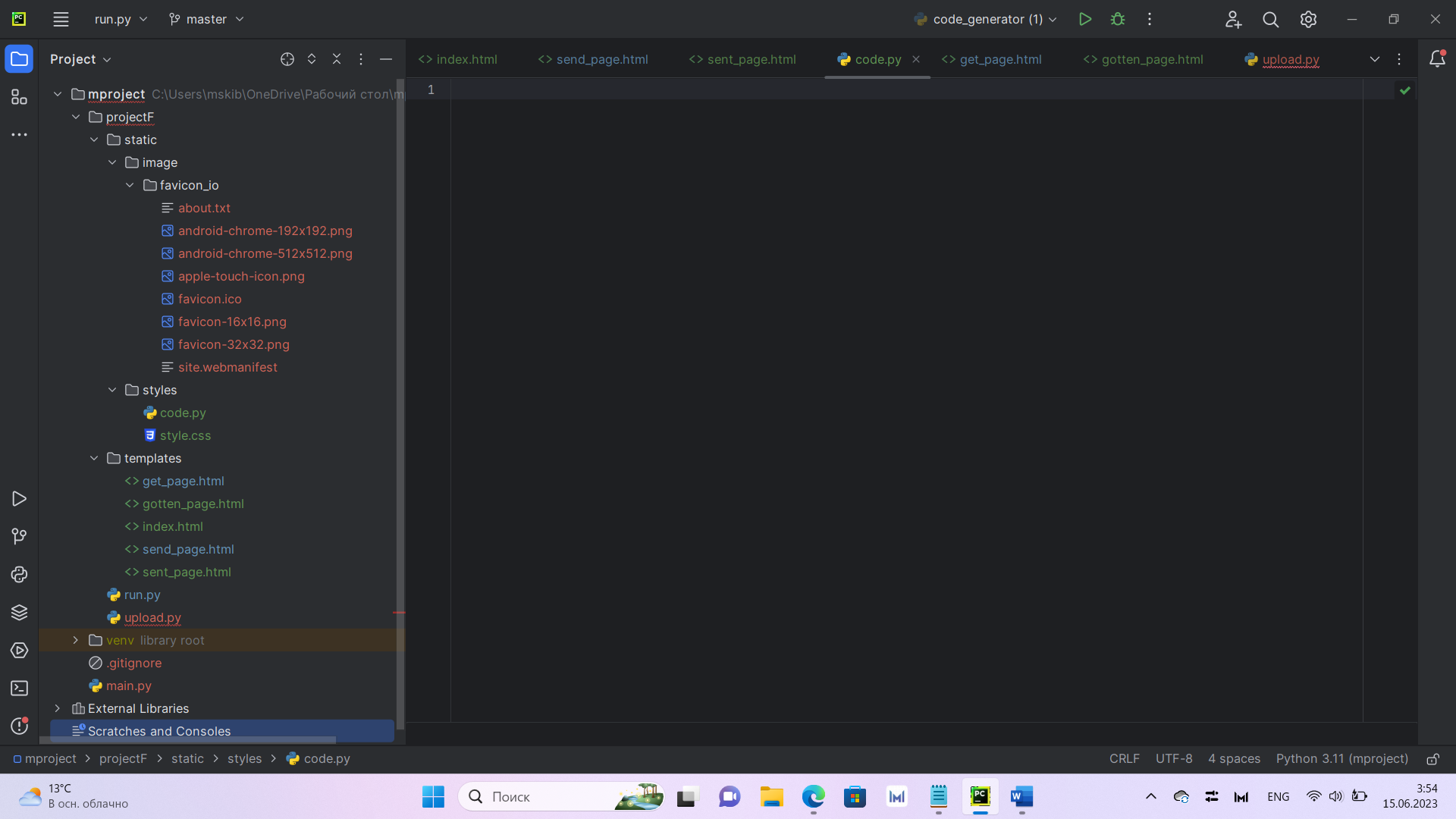


Рис 1. Расположение файлов в структуре проекта.

Все HTML файлы с созданными веб страницами существуют в одном пространстве на одном уровне templates. Сделано это для простаты ссылки одних страниц на другие.

В static хранятся директории image и styles. Первая хранит в себе набор изображений в разных размерах, параметров и ссылку на онлайн ресурс favicon.io, с помощью которого была создана иконка для сайта. Она используется на всех страницах.

В styles находится style.css и code.py. Первый хранит в себе стили для использующихся элементов всех файлов HTML. Второй служит буфером обмена для закрепления сгенерированного кода за отправленным файлом.



Рис. 2. Иконка сайта.

Ресурсом для хранения кода онлайн был выбран GITHUB за его возможность сразу получить собственный домен, сделать сайт публичным и открытым для доступа с любых устройств.

Фрагмент написанного кода из body в файле index.py, который отвечает за начальную веб-страницу:

<img src="https://i.gifer.com/QNBK.gif">  
  
<div class="hello">WELCOME TO THE SITE !</div>  
  
<div class="creator">Development by Mikhail Skibitskiy. Support: mskib@mail.ru</div>  
  
<div class="what">What do you need ?</div>  
  
<a href="send\_page.html" style="display: inline-block; margin: 10px;">  
 <button class="bsend">  
 SEND  
 </button>  
</a>  
  
<a href="get\_page.html" style="display: inline-block; margin: 10px;">  
 <button class="bget">  
 GET  
 </button>  
</a>  
  
<div>  
<button class="speedtest" style=""  
 onclick="multiply()">Run speedtest</button>

Результат выполнения кода с подключенными параметрами из style.css

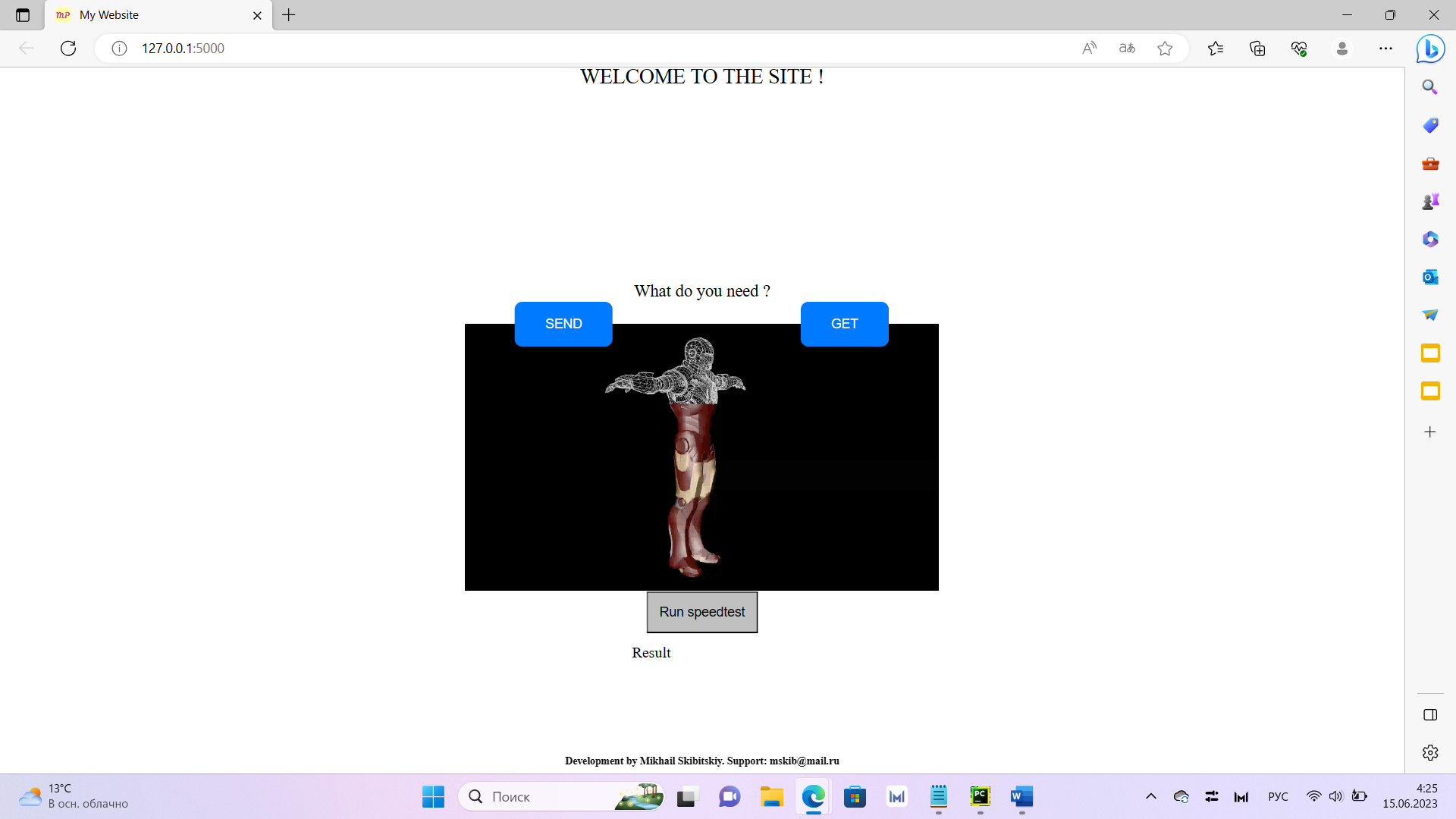


Рис. 3. Стартовая страница.

Так выглядит скрипт для генерации кода проверки. Состоит из латинских букв обоих регистров и десяти цифр. Длинна составляет 9 знаков. Количество уникальных значений составляет **20286591270**

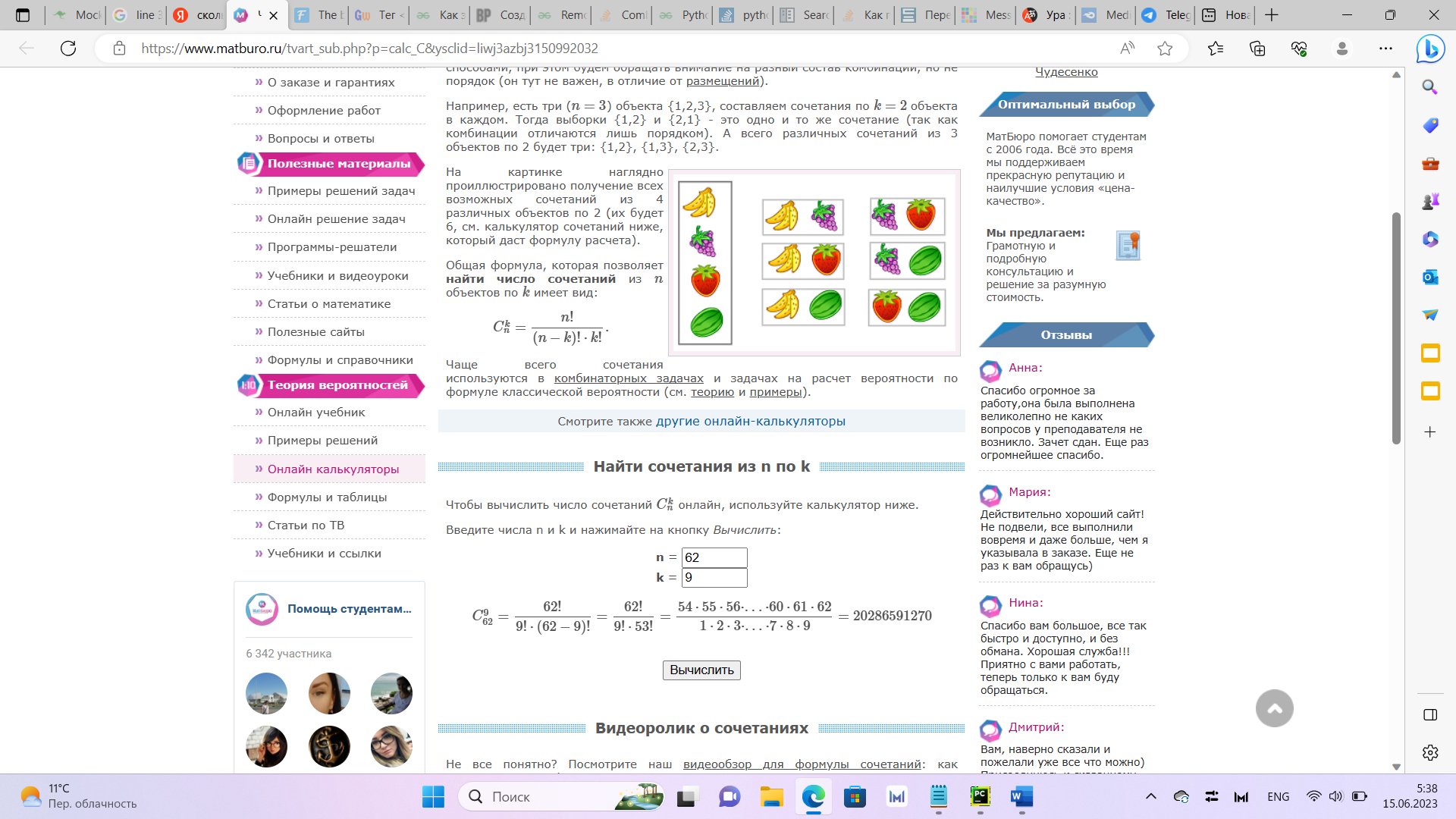


Рис. 4. Расчет количества уникальных комбинаций кода.

<script>  
function generateCode() {  
 var small\_s = "qwertyuiopasdfghjklzxcvbnm";  
 var big\_s = "QWERTYUIOPASDFGHJKLZXCVBNM";  
 var number\_s = "0123456789";  
 var use\_for = small\_s + big\_s + number\_s;  
 var length\_code = 9;  
 var code = "";  
 for (var i = 0; i < length\_code; i++) {  
 code += use\_for.charAt(Math.floor(Math.random() \* use\_for.length));  
 }  
 document.getElementById("codeInput").value = code;  
}  
</script>

**ОПИСАНИЕ РАБОТЫ**

Зайдя на стартовую страницу(index.html), пользователю доступен переход на страницу отправки файлов(send\_page.html) и страницу получения файлов (get\_page.html). Также в нижней части страницы находится кнопка для запуска теста скорости интернета пользователя.

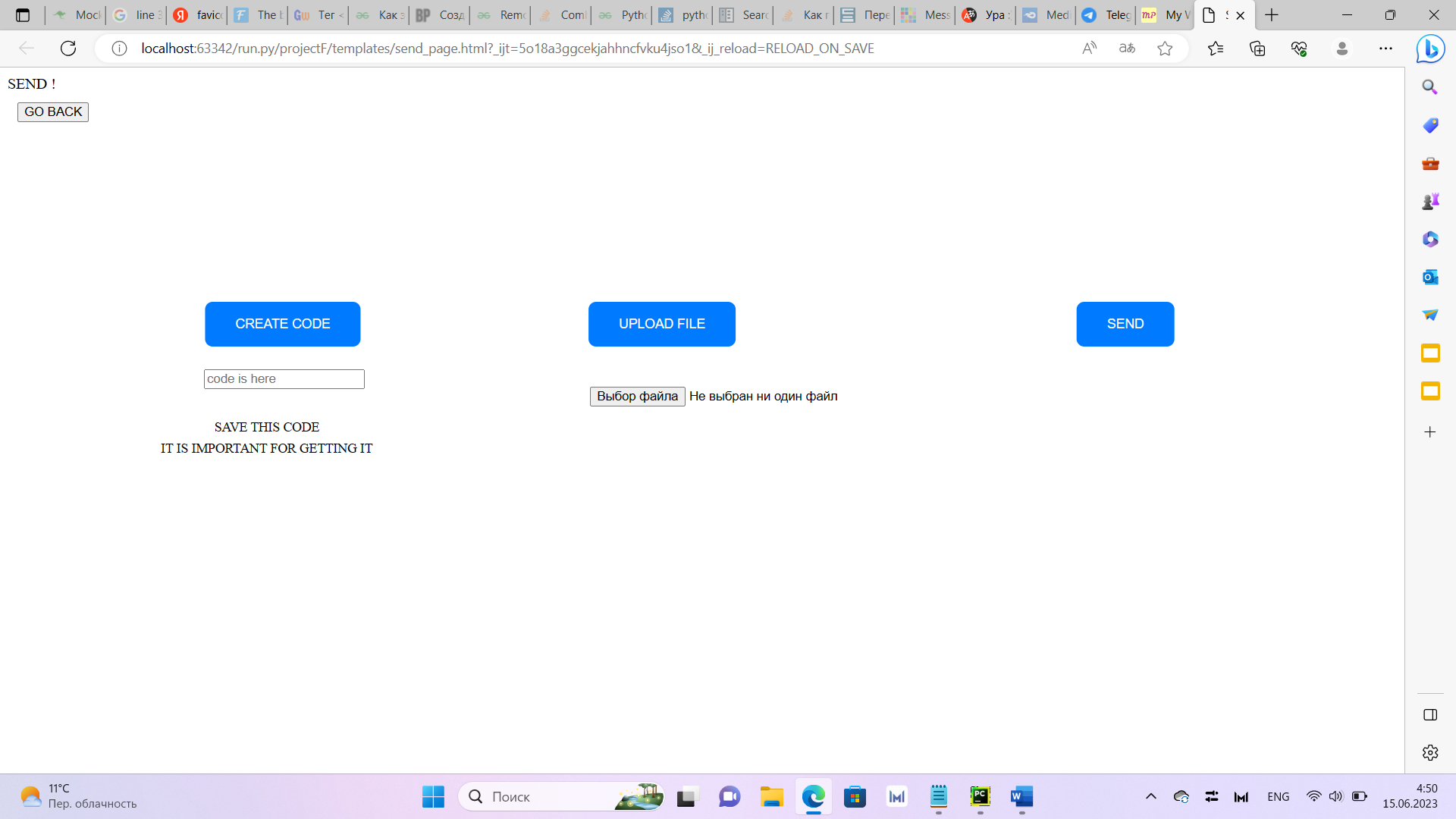


Рис. 4. Страница отправляющая файлы из файла send\_page.html

Совершив 1 нажание левой кнопкой мыши по кнопке SEND(в переводе на рус. «отправить», если включен перевод в браузере), происходит переход на страницу отправки файлов. Здесь доступны возможности для создания уникального девятизначного кода, выбора и загрузки файла, и отправки файла. Также имеется кнопка GO BACK(в переводе на рус. «возвратиться», если включен перевод в браузере) для удобного перехода на стартовую страницу.

Если же на старотовой странице выполнить 1 нажание левой кнопкой мыши по кнопке GET пользователь будет перенеправлен на страницу, предназначеную для получения файлов и их скачивания на свое устройство.

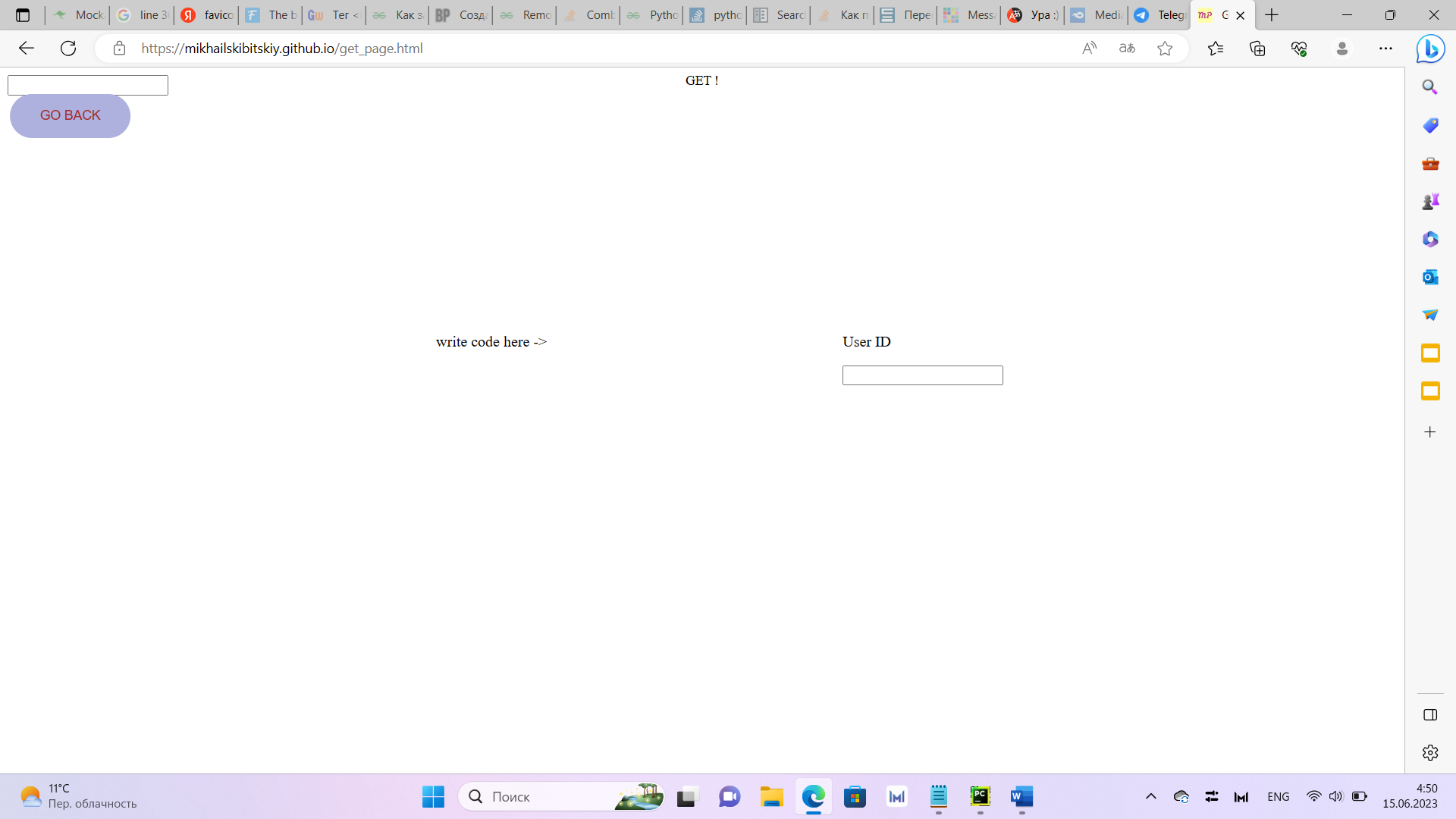


Рис. 5. Страница получающая файлы из файла get\_page.html

Страница также имеет в своем интерфейсе кнопку GO BACK для возврата на начальную страницу.

**ВЫВОД**

Выбранный проект поставил передо мной ряд новых, ранее не встречавшихся задачь различного уровня трудности. Начало работы протекало в теми или иными ошибками в написаном програмном коде. Однако используя дополнительные информационные ресурсы из сети Интернет, обучающие видео и запросы на форумах прошлых лет от людей со схожими целями мне удалось найти ряд различных решений для устарнения препятствующих ошибок.

В проекте удалось реализовать собственый сайт, который работает до сих пор по URL адрессу <https://mikhailskibitskiy.github.io/>.

Удалось реализовать свое видение в системе храниения директорий.

Удалось разработать собственный дизайн веб-страниц и находящихся на них элементах интерфейса.

Удалось воспользоваться ресурсом для создания иконок страниц и применить их в проекте.

Удалось реализовать скрипт по созданию и присвоению случайного девятизначного кода.

Учитывая количество уникальных комбинаций по созданию кода проверки, считаю вопрос получения чужого файла не актуальным. Но готов заметь, что можно добавить проверку на не повторяемость. Чтобы исключить возможность присвоения одного кода разным файлам.

Главной трудность остается создание и подключения к проекту пространства для хранения файлов без финансовых вложений.

**РЕСУРСЫ**

Видео урок по HTML и CSS с примером работы - <https://youtu.be/Bmtu5eNnjK8>.

Видео урок по HTML – <https://youtu.be/6K83dgjkQNw>.

Онлайн форум с вопросами и ответами - [Stack Overflow - Where Developers Learn, Share, & Build Careers](https://stackoverflow.com/).