МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации

**

**ОТЧЁТ**

**по практическая работа №3**

«Клиент-серверная архитектура ПО**»**

**по дисциплине: «***Технологии и методы программирования***»**

|  |  |
| --- | --- |
| Выполнил: | Проверил: |
| Студент гр. «АБс-123», «АВТФ» | Ассистент кафедры ЗИ |
| Номоконов Д.С. | Медведев М.А. |
| «30» мая 2023 г. | «\_\_\_» \_\_\_\_\_\_ 2023 г. |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (подпись) | (подпись) |

Новосибирск 2023

**Цель работы**: развертывание собственного VDS в инфраструктуре НГТУ. Деплой веб-приложения.

**Задачи:**

1. Авторизация на сайте cloud.nstu.ru;
2. Знакомство с устройством хостинга;
3. Создание виртуального сервера:
   * 1. Выбор ОС;
     2. Установка времени работы;
     3. Выбор кол-ва виртуальных ядер, оперативной памяти;
     4. Задание квоты дискового пространства.
4. Установка необходимых компонентов рабочего окружения и развертывание веб – сервиса:
   * 1. Собственный VPN сервер;
     2. Веб – сайт;
     3. Приложение.
5. Настройка внешнего доступа.

Варианты предметной области:

Школьное образование

**Ход работы:**

1. Создание виртуального сервера:
   1. Выбор ОС;
   2. Установка времени работы;
   3. Выбор кол-ва виртуальных ядер, оперативной памяти;
   4. Задание квоты дискового пространства.

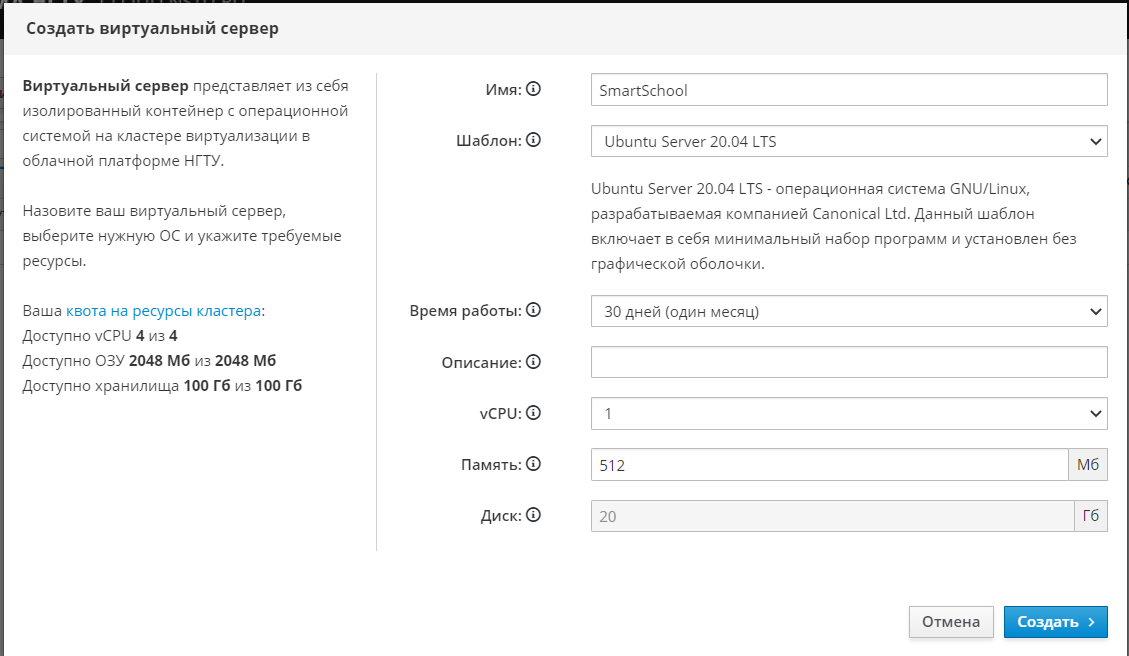


Рисунок 1 – Создание виртуального сервера

На рисунке 1 представлено развёртывание виртуального сервера с памятью 512Мб и ОС Ubuntu Server 20.04 LTS.

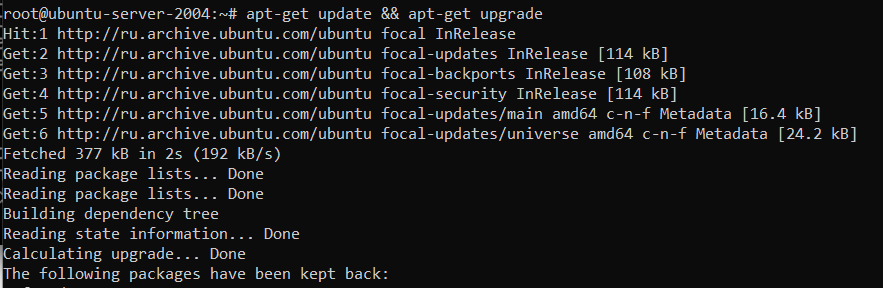


Рисунок 2 - Обновление хранилища пакетного менеджера apt

На рисунке 2 представлено обновленное хранилище пакетного менеджера apt

– программы для установки, обновления и удаления программных пакетов в операционных системах Debian и основанных на них.



Рисунок 3 – Размещение SSH

На рисунке 3 представлено размещение SSH.

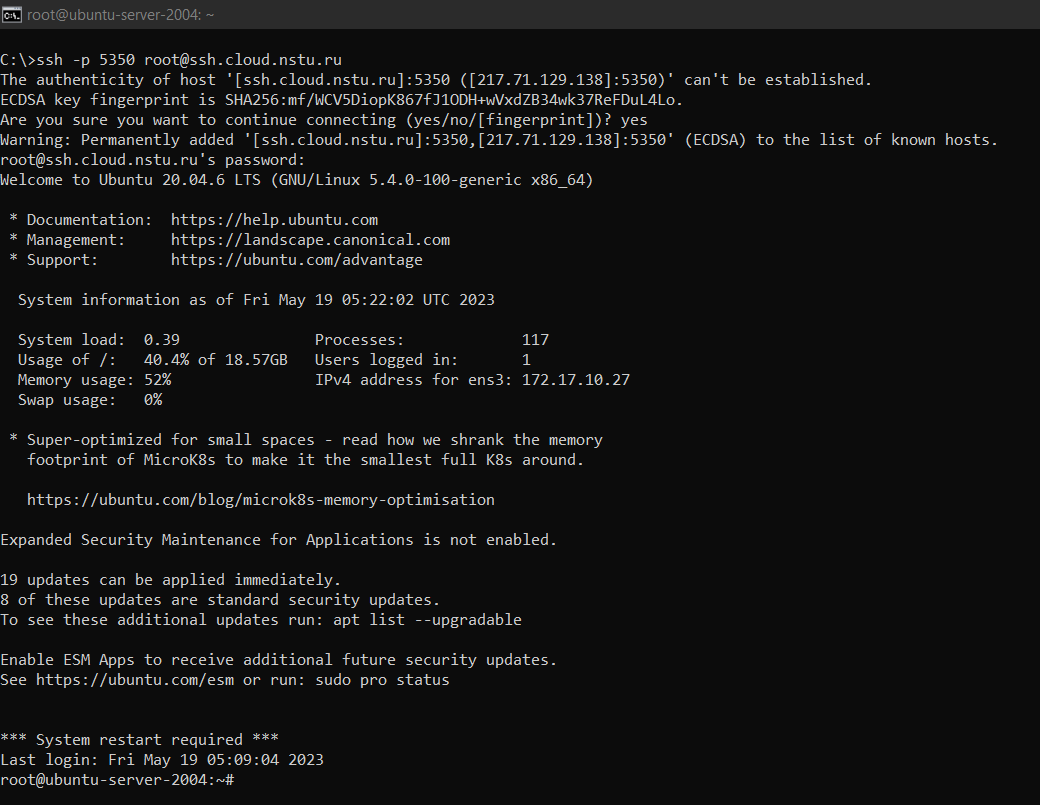


Рисунок 4 – Подключение SSH

На рисунке 4 подключаемся к виртуальному серверу с локальной машины при помощи консольной утилиты SSH, которая позволяет производить удалённое управление операционной системой.

1. Установка необходимых компонентов рабочего окружения и развертывание веб – сервиса:
   1. Собственный VPN сервер;
   2. Веб – сайт;
   3. Приложение.

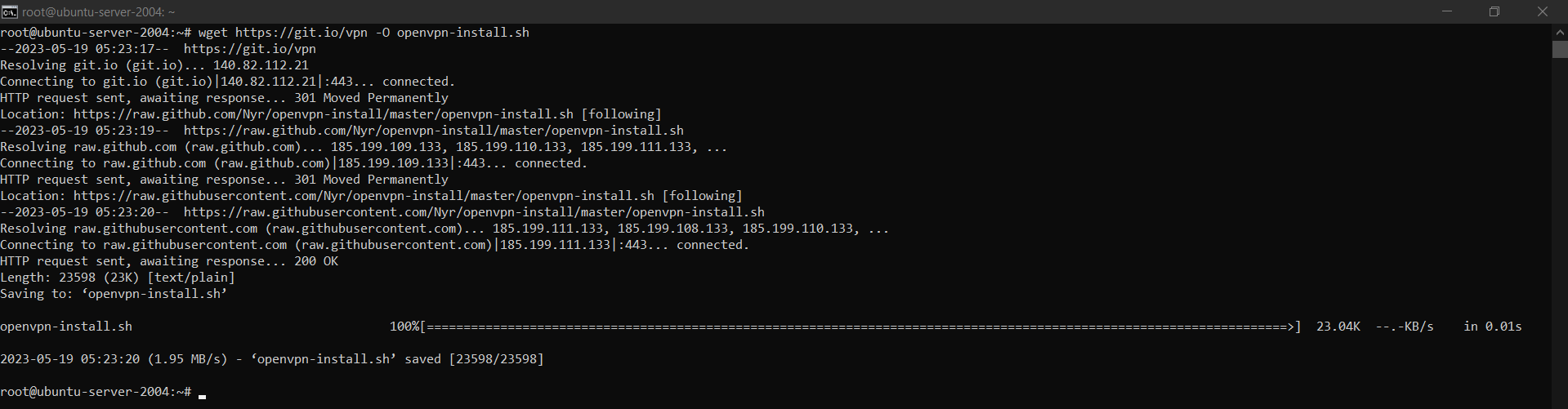


Рисунок 5 – Установка скрипта, который упрощает процедуру настройки и создания клиентов OpenVPN

На рисунке 5 показана установка OpenVPN, производиться с помощью команды wget https://git/io/vpn -O openvpn-install.sh. Оно упрощает процедуру настройки создания клиентов OpenVPN, продемонстрированную ниже.

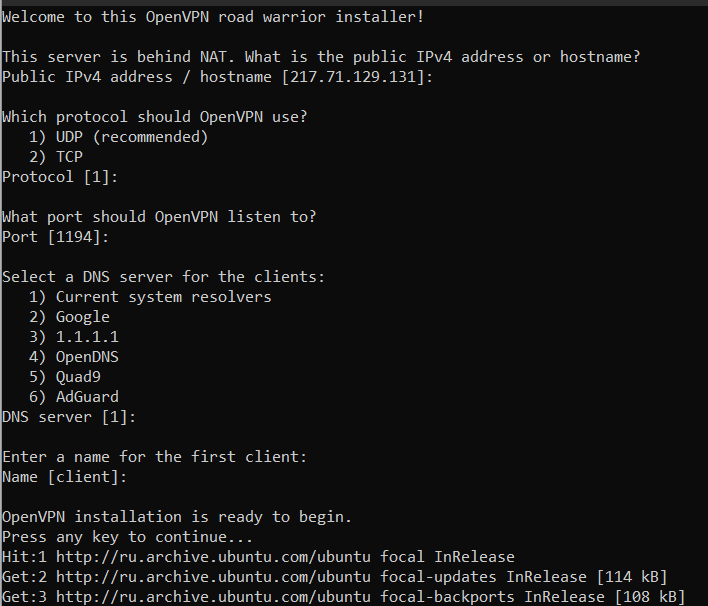


Рисунок 6 – Установка зависимостей

На рисунке 6 показана настройка OpenVPN, выбираем значения по умолчанию, так как они нам подходят. Многие вещи делаются под капотом с помощью установленного скрипта: sudo bash ./openvpn-install.sh.

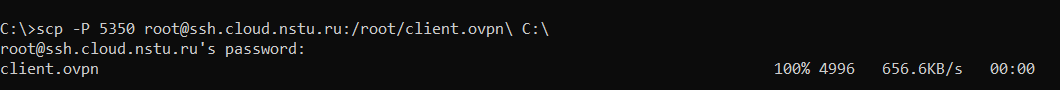


Рисунок 7 – Создание конфигурационного файла

Сохраняем клиентский конфигурационный файл (client.ovpn), который в дальнейшем будем использовать для авторизации клиента. На этом работа на сервере закончена. Результат представлен на рисунке 7.

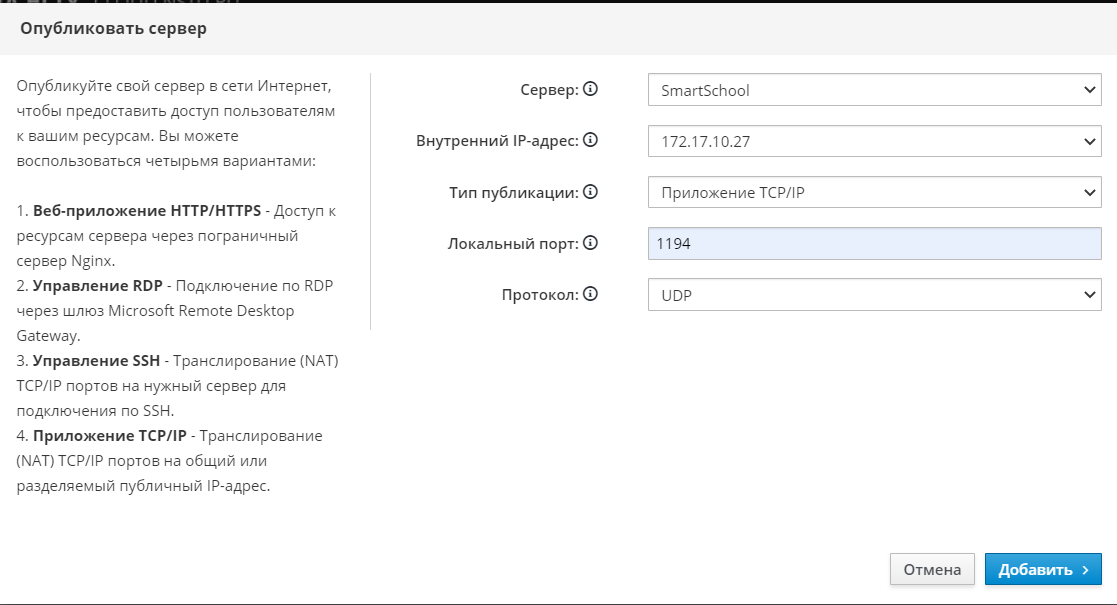


Рисунок 8 – Публикация сервера по протоколу UDP

На рисунке 8 представлена публикация сервера с портом 1194 по протоколу UDP. UDP – быстрый протокол с минимальным механизмом для передачи данных. Он не гарантирует доставку сообщений и не требует открывать соединение. То есть данные могут быть отправлены, как только они будут подготовлены

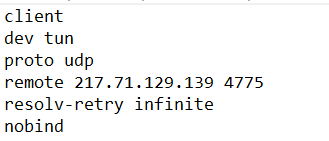


Рисунок 9 – Замена IP PORT

На рисунке 9 представлена замена IP PORT в файле client.ovpn, на тот, который указан на сайте хостинга в разделе «Сети» -> «Внешний адрес», после публикации сервера.

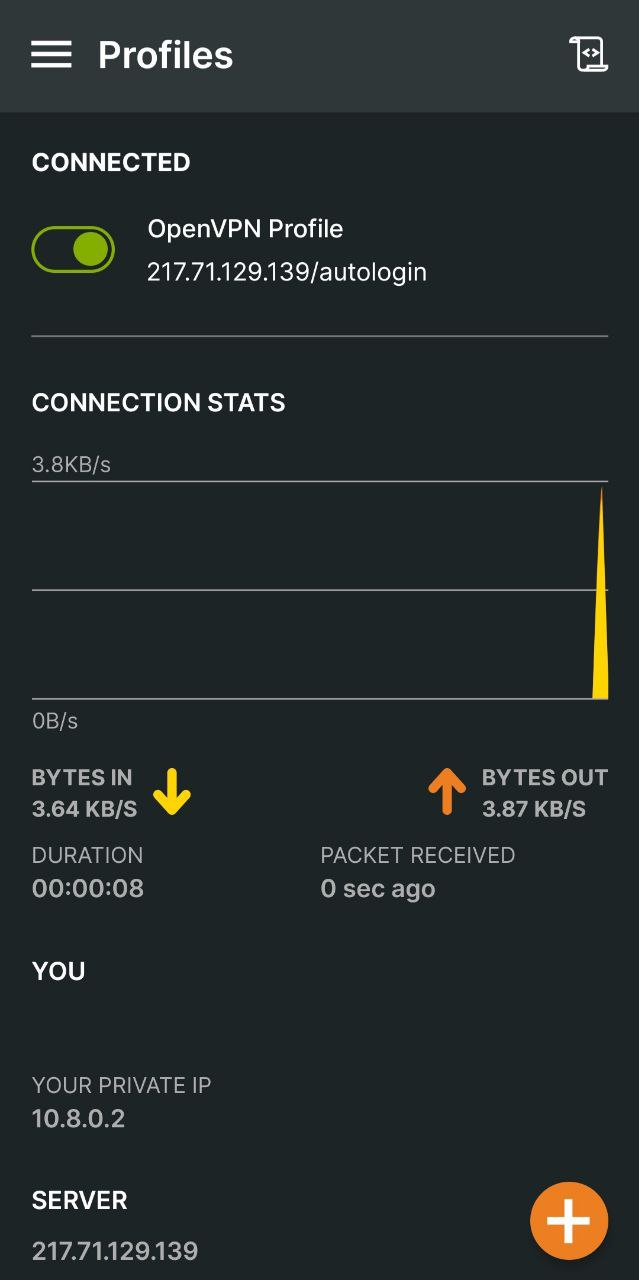


Рисунок 10 – Подключение к локальной сети

Наш виртуальный сетевой интерфейс имеет IP адрес в локальной сети, таким образом мы оказались внутри локальной сети внешней инфраструктуры. Различные компании часто предоставляют возможность доступа во внутреннюю сеть из разных точек мира свои сотрудникам, посредством VPN. На рисунке 10 представлено успешное подключение к локальной сети.

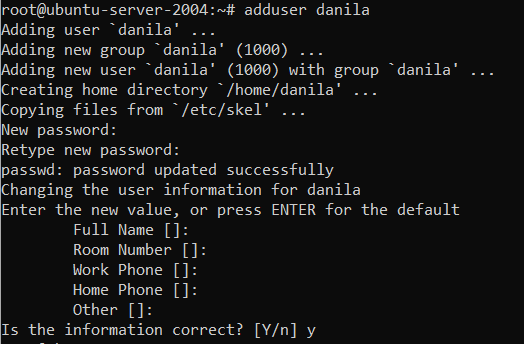


Рисунок 11 – Создание нового пользователя

На рисунке 11 представлено создание нового пользователя со стандартными параметрами на виртуальном сервере под именем «danila».

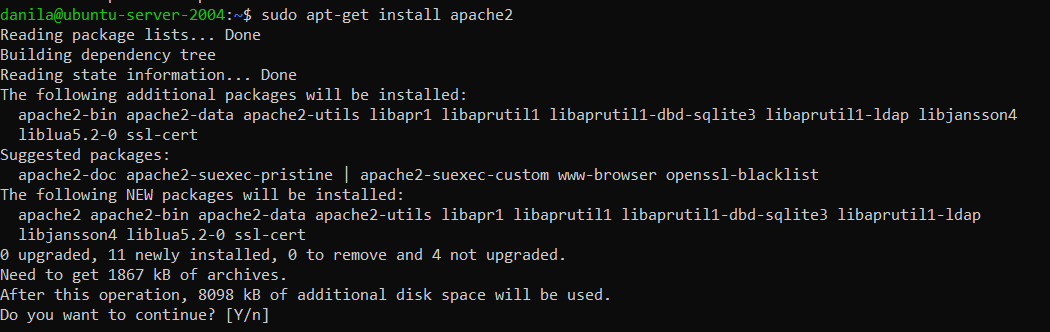


Рисунок 12 – Установка пакетов Apache2

На рисунке 12 представлена установка пакета apache2, производимая с помощью команды sudo apt-get install apache2. Apache – самый широко используемый веб-сервер в мире. Это открытое кроссплатформенное ПО для размещения и поддержки веб-сервера.

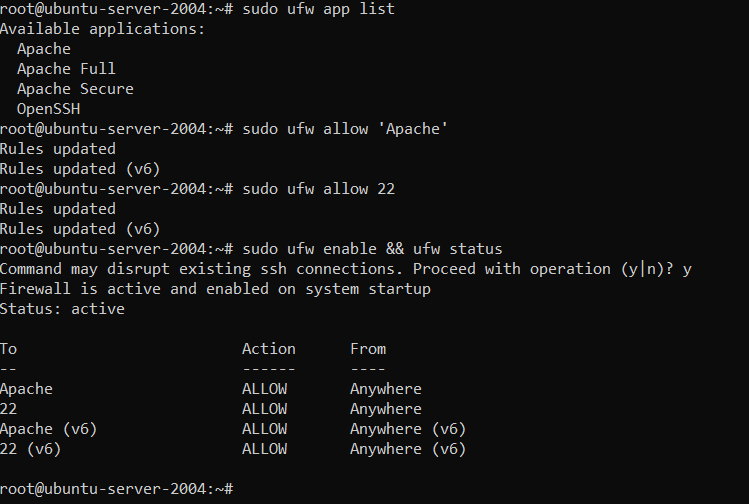


Рисунок 13 – Настройка профилей

На рисунок 13 выведен список профилей приложений ufw, применен самый ограничивающий профиль, который будет разрешать заданный трафик. Поскольку в этом модуле мы еще не настроили SSL для нашего сервера, нам нужно будет только разрешить трафик на порту 80 – протокол HTTP, а также открыт 22 порт, по которому осуществляет SSH соединение, если мы не сделаем, то не сможем подключаться вновь в будущем.

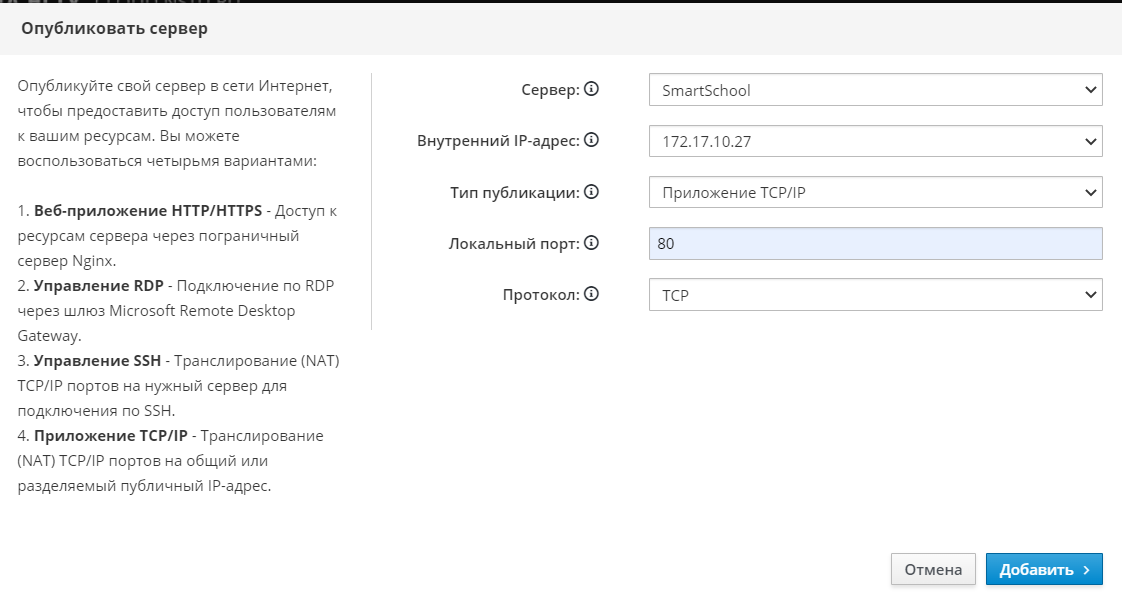


Рисунок 14 – Публикация по TCP

На рисунке 14 представлена публикация по TCP с портом 80. 80 порт предоставляет нормальный веб-трафик без шифрования.

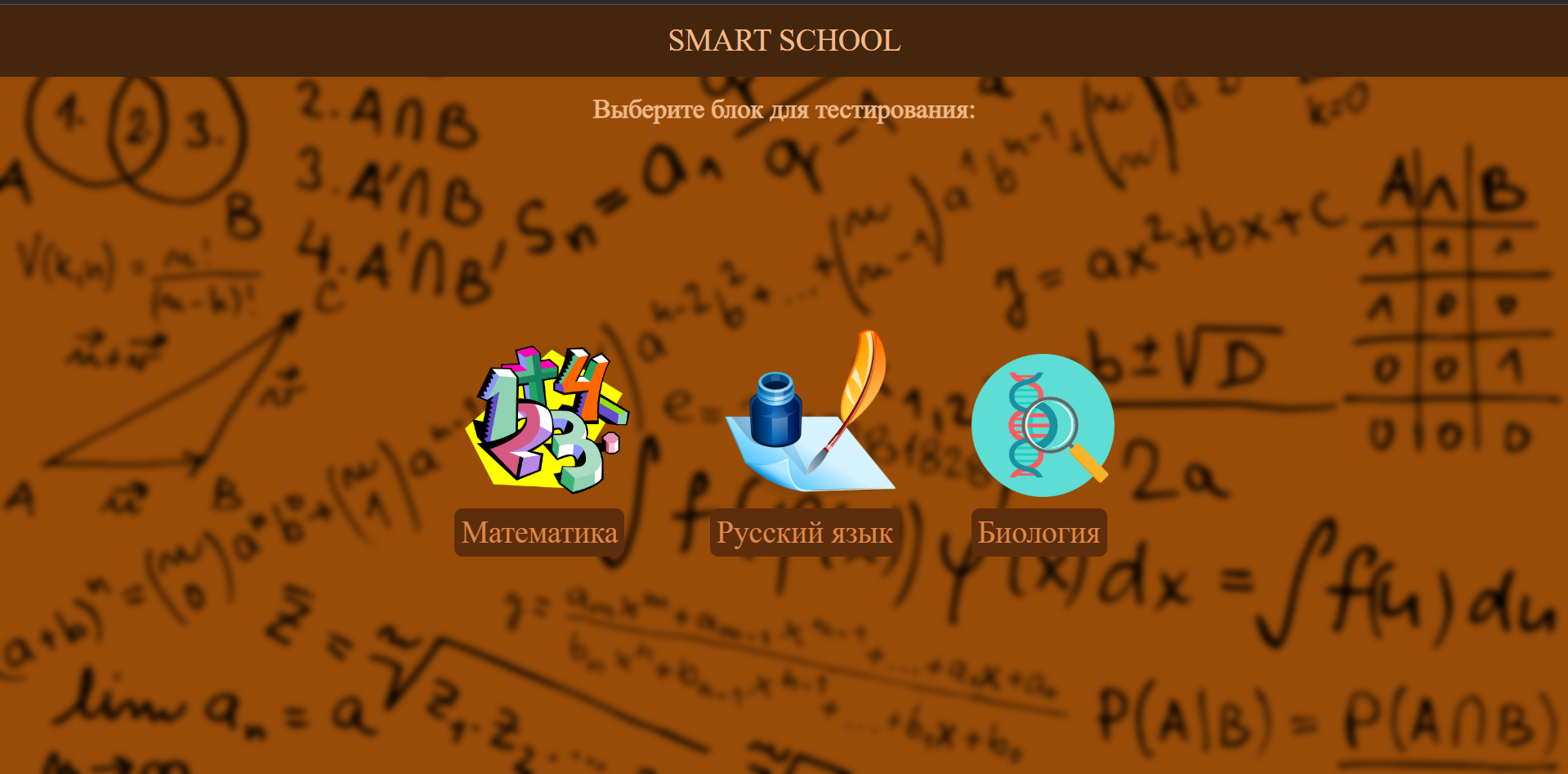


Рисунок 15.1 – Работа веб-приложения

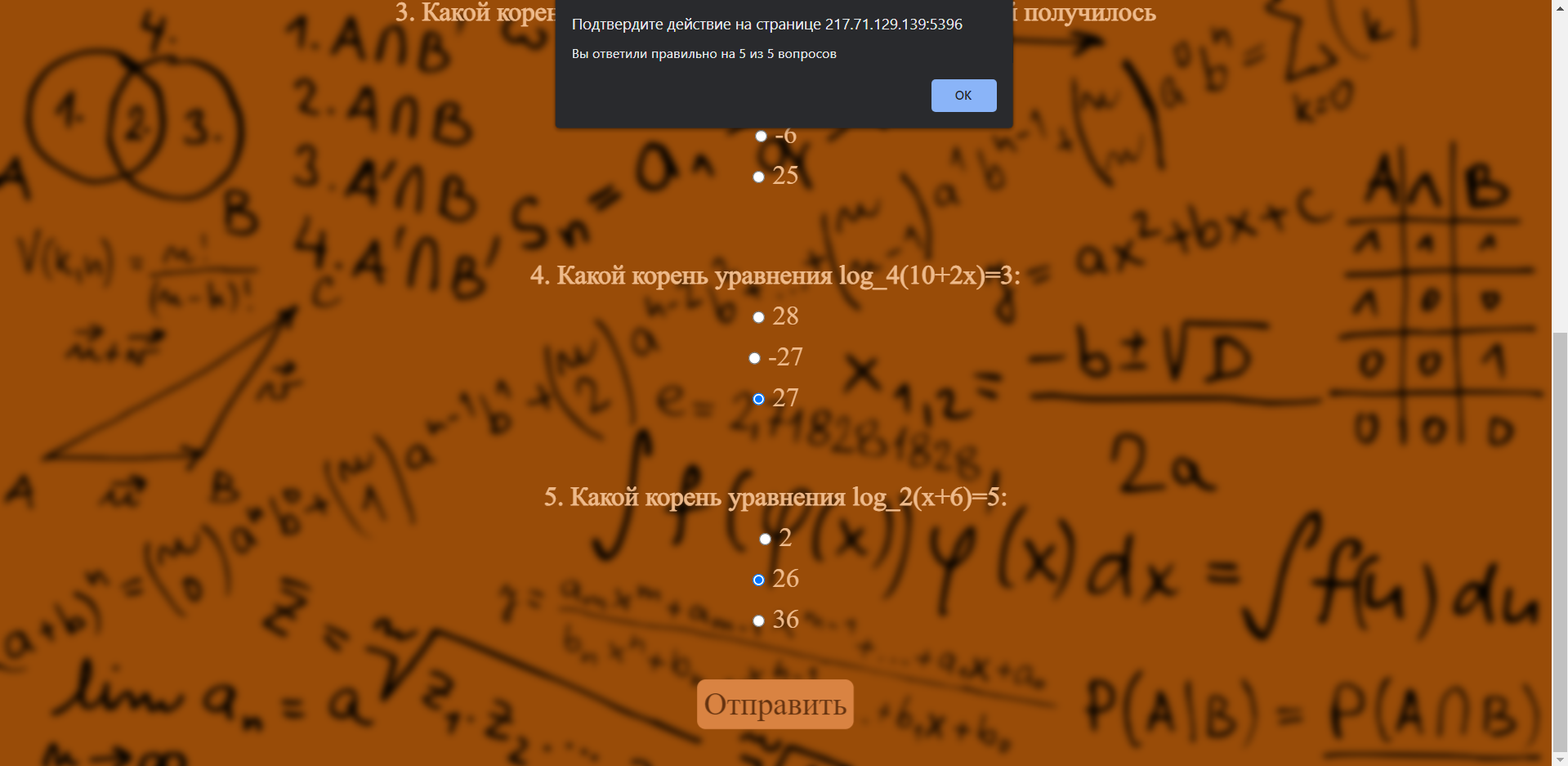


Рисунок 15.2 – Работа веб-приложения

На рисунках 15.1 и 15.2 представлено веб-приложение. Данный сайт содержит три блока, в каждом из которых представлено пять тестовых вопросов. Сайт очень легко дополнять, поэтому добавить новые вопросы для тестов или новые тестовые блоки не составит труда.

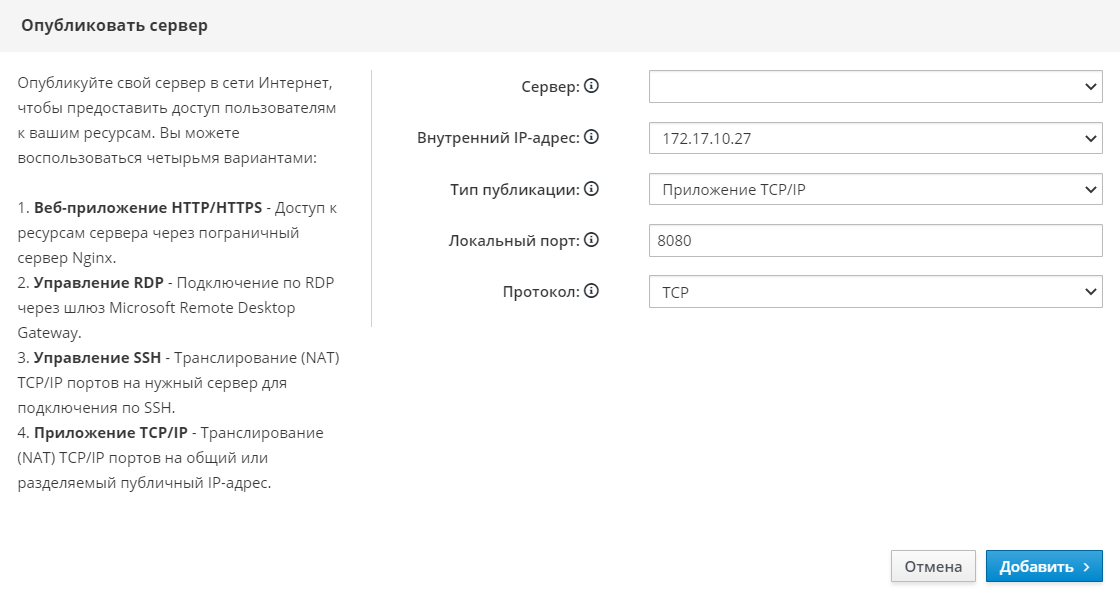


Рисунок 16 – Публикация по TCP по порту 8080

Публикация сервера по протоколу TCP c портом 8080 представлена на рисунке 16.

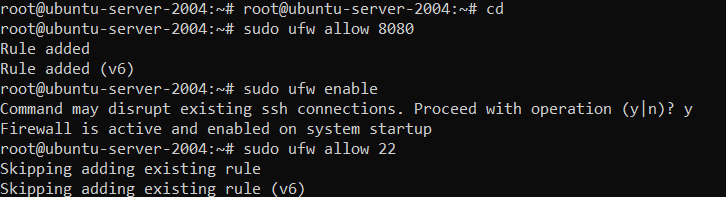


Рисунок 17 – Открытие порта 22, разрешение пота в брандмауэру, включение брандмауэра

Открываем порт, который будет слушать сокет. Сокет – это программный интерфейс, который обеспечивает обмен данными между двумя процессами. По порту 22 осуществляется подключение по SSH, поэтому его необходим так же открыть. Результат представлен на рисунке 17.

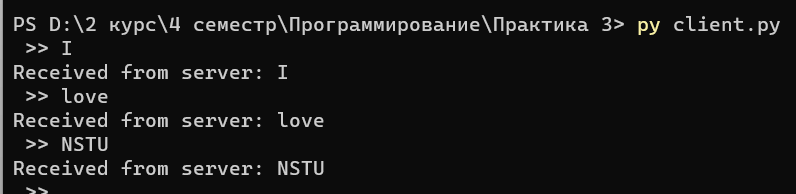


Рисунок 18 – Запуск клиентской части

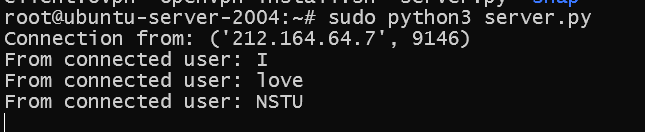


Рисунок 19 – Запуск серверной части

После включения скрипта на сервере мы можем запустить скрипт клиента и отправлять запросы. Результат работы скриптов представлен на рисунке 18, 19.

**Вывод:** При выполнении практической работы были изучены этапы создания виртуального сервера на облачной платформе НГТУ cloud.nstu.ru. Был развёрнут собственный веб-сервер, на который был установлен веб-сайт, разработанный заранее по варианту предметной области.

Были созданы приложения с протоколами UDP - быстрый протокол с минимальным механизмом для передачи данных, не требующий открывать соединение, - и TCP - протокол для обмена данными, который может обеспечивать надёжную связь, используя «трехстороннее рукопожатие». Эти протоколы нужны для корректной работы веб-сервера. Так, например, открытый 80 порт (HTTP/HTTPS) по протоколу TCP позволил нам заходить на веб-сервер с внешних носителей.

Так же было изучено управление веб-сервером с помощью консольной утилиты SSH, которая позволяет производить удаленное управление ОС и туннелирование TCP-соединений.

С помощью python, используя сокеты, мы смогли создать эхо скрипт и осуществлять взаимодействие с ним посредством обращения к серверу.

ПРИЛОЖЕНИЕ

**1.html**

<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="1.css">

<title>SMART SCHOOL</title>

</head>

<body bgcolor="#994c08" style="height: 100vh;">

<div class="head">

<p>SMART SCHOOL</p>

</div>

<div class="title">

<p class="forTest">Выберите блок для тестирования:</p>

</div>

<div class="title1">

<div style="margin-top: 15px;">

<div class="img1"><img src="https://i.ibb.co/kyxh2T3/pngwing-com-7.png" style="width: 170px;"/></div>

<div><button onclick="newWindow2()">Математика</button></div>

</div>

<div>

<div class="img2"><img src="https://i.ibb.co/PYG9Fgv/pngwing-com-4.png" style="width: 170px;"/></div>

<div class="divBtn2"><button onclick="newWindow3()">Русский язык</button></div>

</div>

<div>

<div class="img3"><img src="https://i.ibb.co/1J9ykm8/pngwing-com-5.png" style="width: 140px;"/></div>

<div style="margin-top: 5px;"><button onclick="newWindow4()">Биология</button></div>

</div>

</div>

<script src="1.js"></script>

<script src="2.js"></script>

<script src="3.js"></script>

</body>

</html>

**2.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="1.css">

<title>SMART SCHOOL</title>

</head>

<body bgcolor="#994c08">

<div class="head">

<p>МАТЕМАТИКА</p>

</div>

<form>

<p class="html2Questions">1. Какой корень уравнения log\_5⁡(4+x)=2: </p>

<input type="radio" name="q1" value="a"> 15

<br>

<input type="radio" name="q1" value="b"> -21

<br>

<input type="radio" name="q1" value="c"> 21<!--ВЕРНО-->

<br><br>

<p class="html2Questions">2. Какой корень уравнения x^2-17x+72=0, если корней получилось несколько, укажите наименьший:</p>

<input type="radio" name="q2" value="a"> -9

<br>

<input type="radio" name="q2" value="b"> 8<!--ВЕРНО-->

<br>

<input type="radio" name="q2" value="c"> -1

<br><br>

<p class="html2Questions">3. Какой корень уравнения 4x^2+7=7+24x, если корней получилось несколько, укажите наименьший:</p>

<input type="radio" name="q3" value="a"> 0<!--ВЕРНО-->

<br>

<input type="radio" name="q3" value="b"> -6

<br>

<input type="radio" name="q3" value="c"> 25

<br><br>

<p class="html2Questions">4. Какой корень уравнения log\_4⁡(10+2x)=3:</p>

<input type="radio" name="q4" value="a"> 28

<br>

<input type="radio" name="q4" value="b"> -27

<br>

<input type="radio" name="q4" value="c"> 27<!--ВЕРНО-->

<br><br>

<p class="html2Questions">5. Какой корень уравнения log\_2⁡(x+6)=5:</p>

<input type="radio" name="q5" value="a"> 2

<br>

<input type="radio" name="q5" value="b"> 26<!--ВЕРНО-->

<br>

<input type="radio" name="q5" value="c"> 36

<br><br>

<input type="submit" value="Отправить" class="input">

<br><br>

</form>

<script src="1.js"></script>

</body>

</html>

Script1.html

let endSum = 0, endCount = 0, arr = [];

const display = document.getElementById("price\_");

if (localStorage.getItem('endSum') != null) {

display.innerHTML += Number(localStorage.getItem('endSum')) + " капибарных рублей";

}

else {

display.innerHTML += 0 + " капибарных рублей";

}

function Price(numPrice, count, name) {

let basket, res = [0, 0], sum, sumCount;

if (localStorage.getItem(name) != null) {

basket = localStorage.getItem(name);

res = JSON.parse(basket);

}

sum = Number(res[1]);

sum += numPrice;

sumCount = Number(res[0]);

sumCount += count;

endSum = Number(localStorage.getItem('endSum'));

endSum += numPrice;

endCount = Number(localStorage.getItem('endCount'));

endCount += count;

display.innerHTML = ''

display.innerHTML += endSum + " капибарных рублей";

arr = [];

arr[0] = sumCount;

arr[1] = sum;

localStorage.setItem(name, JSON.stringify(arr));

localStorage.setItem('endSum', Number(endSum));

localStorage.setItem('endCount', Number(endCount));

}

**3.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="1.css">

<title>SMART SCHOOL</title>

</head>

<body bgcolor="#994c08">

<div class="head">

<p>РУССКИЙ ЯЗЫК</p>

</div>

<form>

<p class="html2Questions">1. В каких словах пропущена одна и та же буква:</p>

<input type="radio" name="q1" value="a"> удоста..вать, масл..це<!--ВЕРНО-->

<br>

<input type="radio" name="q1" value="b"> угр..ватый, досто..н

<br>

<input type="radio" name="q1" value="c"> локт..вой, ключ..к

<br><br>

<p class="html2Questions">2. В каких словах пропущена одна и та же буква:</p>

<input type="radio" name="q2" value="a"> улыбч..вый, ноч..вать

<br>

<input type="radio" name="q2" value="b"> развед..вательный, размаз..вать<!--ВЕРНО-->

<br>

<input type="radio" name="q2" value="c"> слев.., вин..ватый

<br><br>

<p class="html2Questions">3. Укажите варианты ответов, в которых верно выделены буквы, обозначающие ударный гласный звук:</p>

<input type="radio" name="q3" value="a"> бантЫ, позвонИшь

<br>

<input type="radio" name="q3" value="b"> сИроты, каталОг

<br>

<input type="radio" name="q3" value="c"> позвонИшь, вОвремя<!--ВЕРНО-->

<br><br>

<p class="html2Questions">4. В одном из выделенных ниже слов допущена ошибка в образовании формы слова. Какое этос слово:</p>

<input type="radio" name="q4" value="a"> ЛОПНУТЫЕ пружины<!--ВЕРНО-->

<br>

<input type="radio" name="q4" value="b"> у ОБЕИХ сестёр

<br>

<input type="radio" name="q4" value="c"> ЛЯГТЕ на коврик

<br><br>

<p class="html2Questions">5. В каких словах пропущена одна и та же буква:</p>

<input type="radio" name="q5" value="a"> вопл..тить, обр..мление

<br>

<input type="radio" name="q5" value="b"> заж..гать, сомн..ваться

<br>

<input type="radio" name="q5" value="c"> покл..ниться, ог..рчение<!--ВЕРНО-->

<br><br>

<input type="submit" value="Отправить" class="input">

<br><br>

</form>

<script src="2.js"></script>

</body>

</html>

**4.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="1.css">

<title>SMART SCHOOL</title>

</head>

<body bgcolor="#994c08">

<div class="head">

<p>БИОЛОГИЯ</p>

</div>

<form>

<p class="html2Questions">1. У грибов, как и у бактерий нет:</p>

<input type="radio" name="q1" value="a"> клеточной оболочки

<br>

<input type="radio" name="q1" value="b"> цитоплазмы

<br>

<input type="radio" name="q1" value="c"> пластид(хлоропластов)<!--ВЕРНО-->

<br><br>

<p class="html2Questions">2. Растения необходимо охранять, так как они:</p>

<input type="radio" name="q2" value="a"> служат пищей растительноядным животным;<!--ВЕРНО-->

<br>

<input type="radio" name="q2" value="b"> поглощают из почвы воду

<br>

<input type="radio" name="q2" value="c"> поглощают из почвы минеральные соли

<br><br>

<p class="html2Questions">3. К голосеменным относятся растения:</p>

<input type="radio" name="q3" value="a"> имеющие цветки, но не образующие плодов

<br>

<input type="radio" name="q3" value="b"> не имеющие цветков, но образующие семена<!--ВЕРНО-->

<br>

<input type="radio" name="q3" value="c"> не имеющие цветков, но образующие плоды

<br><br>

<p class="html2Questions">4. Органы растений увеличиваются в размере благодаря</p>

<input type="radio" name="q4" value="a"> разрушению клеточных стенок

<br>

<input type="radio" name="q4" value="b"> накоплению солей

<br>

<input type="radio" name="q4" value="c"> делению и росту клеток<!--ВЕРНО-->

<br><br>

<p class="html2Questions">5. Характерный признак бактерий:</p>

<input type="radio" name="q5" value="a"> наличие ядра

<br>

<input type="radio" name="q5" value="b"> отсутствие цитоплазмы

<br>

<input type="radio" name="q5" value="c"> отсутствие ядра<!--ВЕРНО-->

<br><br>

<input type="submit" value="Отправить" class="input">

<br><br>

</form>

<script src="3.js"></script>

</body>

</html>

**1.js**

function newWindow2() {

window.location.href = '2.html';

}

// Получаем форму и кнопку отправки

const form1 = document.querySelector('form');

const submitBtn1 = form1.querySelector('input[type="submit"]');

// Обработчик события на кнопку отправки

submitBtn1.addEventListener('click', function(e) {

e.preventDefault(); // Отменяем стандартное поведение кнопки

// Получаем все выбранные радиокнопки

const answers = form1.querySelectorAll('input[type="radio"]:checked');

let correctAnswers = 0;

// Сравниваем ответы пользователя с правильными ответами

if (answers[0].value === 'c') {

correctAnswers++;

}

if (answers[1].value === 'b') {

correctAnswers++;

}

if (answers[2].value === 'a') {

correctAnswers++;

}

if (answers[3].value === 'c') {

correctAnswers++;

}

if (answers[4].value === 'b') {

correctAnswers++;

}

// Выводим результат пользователю

alert(`Вы ответили правильно на ${correctAnswers} из 5 вопросов`);

});

**2.js**

function newWindow3() {

window.location.href = '3.html';

}

// Получаем форму и кнопку отправки

const form2 = document.querySelector('form');

const submitBtn2 = form2.querySelector('input[type="submit"]');

// Обработчик события на кнопку отправки

submitBtn2.addEventListener('click', function(e) {

e.preventDefault(); // Отменяем стандартное поведение кнопки

// Получаем все выбранные радиокнопки

const answers = form2.querySelectorAll('input[type="radio"]:checked');

let correctAnswers = 0;

// Сравниваем ответы пользователя с правильными ответами

if (answers[0].value === 'a') {

correctAnswers++;

}

if (answers[1].value === 'b') {

correctAnswers++;

}

if (answers[2].value === 'c') {

correctAnswers++;

}

if (answers[3].value === 'a') {

correctAnswers++;

}

if (answers[4].value === 'c') {

correctAnswers++;

}

// Выводим результат пользователю

alert(`Вы ответили правильно на ${correctAnswers} из 5 вопросов`);

});

**3.js**

function newWindow3() {

window.location.href = '3.html';

}

// Получаем форму и кнопку отправки

const form2 = document.querySelector('form');

const submitBtn2 = form2.querySelector('input[type="submit"]');

// Обработчик события на кнопку отправки

submitBtn2.addEventListener('click', function(e) {

e.preventDefault(); // Отменяем стандартное поведение кнопки

// Получаем все выбранные радиокнопки

const answers = form2.querySelectorAll('input[type="radio"]:checked');

let correctAnswers = 0;

// Сравниваем ответы пользователя с правильными ответами

if (answers[0].value === 'a') {

correctAnswers++;

}

if (answers[1].value === 'b') {

correctAnswers++;

}

if (answers[2].value === 'c') {

correctAnswers++;

}

if (answers[3].value === 'a') {

correctAnswers++;

}

if (answers[4].value === 'c') {

correctAnswers++;

}

// Выводим результат пользователю

alert(`Вы ответили правильно на ${correctAnswers} из 5 вопросов`);

});

**1.css**

\* {

margin: 0;

padding: 0;

}

body{

background-image: url("https://i.ibb.co/8jybWCX/pngwing-com-1.png");

background-repeat: no-repeat;

background-position: center center;

background-attachment: fixed;

background-size: cover;

backdrop-filter: blur(3.5px);

-webkit-backdrop-filter: blur(3.5px);

height: 100%;

}

.head {

text-align: center;

color: #efbb8c;

background-color: #44260e;

font-family: 'Times New Roman', Times, serif;

font-size: 30px;

color:#efbb8c;

height: 70px;

display: flex;

justify-content: center;

align-items: center;

position: relative;

}

.title {

color: #efbb8c;

font-family: 'Times New Roman', Times, serif;

font: bold;

font-size: 26px;

margin-top: 17px;

justify-content: center;

align-items: center;

display: flex;

flex-direction: row;

position: relative;

}

.title1 {

color: #efbb8c;

font-family: 'Times New Roman', Times, serif;

font: bold;

font-size: 26px;

margin-top: 200px;

justify-content: center;

align-items: center;

display: flex;

flex-direction: row;

position: relative;

}

.forTest{

-webkit-text-stroke: 0.4px #efbb8c;

}

button {

color: #df8845;

font-family: 'Times New Roman', Times, serif;

background-color: #5e2d0c;

border: 7px solid #5e2d0c;

border-radius: 8px;

font-size: 30px;

text-align: center;

transition: 0.3s ease all;

}

button:hover {

color: #5e2d0c;

background-color: #df8845;

border: 7px solid #df8845;

transform: scale(1.03);

}

.divBtn2 {

margin-left: 70px;

margin-right: 70px;

}

.img3 {

margin-top: 25px;

}

.img2 {

margin-left: 83px;

}

.img1 {

margin-left: 10px;

}

.html2Questions{

color: #efbb8c;

font-family: 'Times New Roman', Times, serif;

font: bold;

font-size: 26px;

margin-top: 17px;

justify-content: center;

align-items: center;

display: flex;

position: relative;

-webkit-text-stroke: 0.4px #efbb8c;

}

form{

color: #efbb8c;

font-family: 'Times New Roman', Times, serif;

font: bold;

text-align: center;

font-size: 26px;

margin-left: auto;

margin-right: auto;

width: 800px;

line-height: 40px;

}

.input{

color: #df8845;

font-family: 'Times New Roman', Times, serif;

background-color: #5e2d0c;

border: 7px solid #5e2d0c;

border-radius: 8px;

font-size: 30px;

text-align: center;

transition: 0.3s ease all;

}

.input:hover{

color: #5e2d0c;

background-color: #df8845;

border: 7px solid #df8845;

transform: scale(1.03);

}

**Client.py**

import socket

def client() -> None:

"""

Эхо

"""

HOST = '217.71.129.139' # Внешний IP PORT сервера, в нашем случае гипервизора

# который перенаправляет запрос на нужный порт нашего сервера

PORT = 5399

client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) # Создаем instance класса

client\_socket.connect((HOST, PORT)) # Подключаемся к серверу

message = input(" >> ") # Вводим данные

while message.lower().strip() != 'exit':

client\_socket.send(message.encode()) # Отправялем данные

data = client\_socket.recv(1024).decode() # Получаем ответ

print('Received from server: ' + data)

message = input(" >> ") # Повтор ввода

client\_socket.close() # закрываем соединение

if \_\_name\_\_ == '\_\_main\_\_':

client()

**Server.py**

def server() -> None:

"""

Эхо

"""

HOST = '172.17.10.27' # Слушаем не localhost, а внутренний адрес VDS

# Узнать его можно либо на сайте хостинга, либо через команду ip a

PORT = 8080 # Инициализируем порт, не ниже 1024, т.к. ниже - зарезервированно

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) # instance класса socket

server\_socket.bind((HOST, PORT)) # биндит hostname и необходимый порт. Принимает кортеж

server\_socket.listen(2) # Кол-во клиентов, которых сервер одновременно слушает

conn, address = server\_socket.accept() # Принимаем новое соединение

print("Connection from: " + str(address))

while True:

# Получаем поток данных. Не принимаем данные больше 1 Кб

data = conn.recv(1024).decode()

if not data:

# Если данные не получены - обрыв

break

print("From connected user: " + str(data))

conn.send(data.encode()) # Отправляем клиенту его же данные

conn.close() # Закрываем соединение

if \_\_name\_\_ == '\_\_main\_\_':

server()