

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра Вычислительной техники**

**ОТЧЕТ**  
**по лабораторной работе № 6**  
**по дисциплине «Web программирование»**  
**Тема: «Аутентификация и авторизация**  
**пользователей Web-приложения»**

Студент гр. 3311 Баймухамедов Р. Р. \_\_\_\_\_

Преподаватель Калмыков М.А. \_\_\_\_\_

Санкт-Петербург

2025

## Цель работы

Знакомство со способами реализации аутентификации и авторизации пользователей Web-приложения

## Выполнение лабораторной работы

Изменили следующие файлы

```
# HTTP – пока без редиректа (включим в конце, когда https взлетит)
server {
    listen 80;
    server_name _;

    root /usr/share/nginx/html;
    index index.html;
    charset utf-8;

    # i18n
    location ^~ /i18n/ {
        try_files $uri =404;
        default_type application/json;
        add_header Cache-Control "no-store, no-cache, must-revalidate" always;
        expires -1;
    }

    # API (без auth на http)
    location /api/ {
        proxy_pass http://app:3000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    # SPA
    location / {
        try_files $uri $uri/ /index.html;
    }
}

# HTTPS
server {
    listen 443 ssl;          # без 'http2' в listen – убиваем warning
    http2 on;                # так правильно включать http2 в новых версиях
    server_name _;

    ssl_certificate /etc/nginx/certs/server.crt;
    ssl_certificate_key /etc/nginx/certs/server.key;
    ssl_protocols TLSv1.2 TLSv1.3;

    root /usr/share/nginx/html;
```

```

index index.html;
charset utf-8;

# i18n
location ^~ /i18n/ {
    try_files $uri =404;
    default_type application/json;
    add_header Cache-Control "no-store, no-cache, must-revalidate" always;
    expires -1;
}

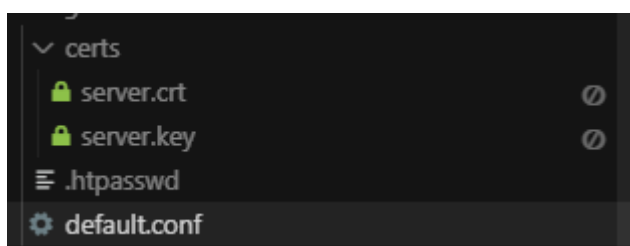
# API noð Basic-Auth
location /api/ {
    auth_basic "Movies API (lab6)";
    auth_basic_user_file /etc/nginx/conf.d/.htpasswd;

    proxy_pass http://app:3000;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}

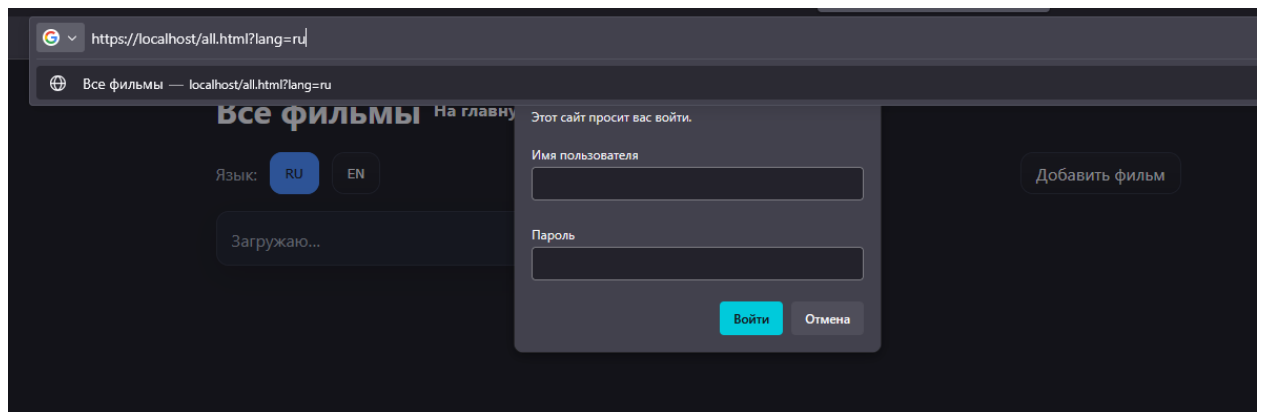
# SPA
location / {
    try_files $uri $uri/ /index.html;
}
}

```

Сгенерировали server.crt и server.key, а также .htpasswd



**Пример работоспособности программы**



## Заключение

В данной лабораторной работе мы познакомились со способами реализации аутентификации и авторизации пользователей Web-приложения