

Руководство пользователя

Программное обеспечение

«Интерактивная 3D «Физика. 8 класс»»

Инструкция по установке



Москва, 2021

Оглавление

1.Общее описание	3
2. Системные требования	4
3. Инструкция по установке	4
3.1. Инструкция по размещению ПО на сервере	5
4. Техническая поддержка.....	14
4.1. Устранение ошибок	14
4.2. Перенос сроков исполнения обращения	15
4.3. Контакты линии поддержки.....	15

1.Общее описание

Программное обеспечение «Интерактивная 3D «Физика. 8 класс» является виртуальной трехмерной интерактивной средой, с включенными в неё Виртуальными лабораториями, предназначенной для помощи школьникам 8-го класса, изучающим физику.

2. Системные требования

Минимальные системные требования для работы серверной части программного обеспечения «Интерактивная 3D «Физика. 8 класс»:

- Процессор: 4 ядра (8 логических потоков), частота – 2 ГГц и больше;
- Оперативная память: 4 Гб и больше;
- Свободное дисковое пространство: 50 Гб (зависит от размера хранимых в системе документов);
- Пропускная способность сетевого интерфейса: от 1 Гбит/с;
- Тип накопителя: SSD;
- СУБД: MySQL 8.0;
- ОС: Ubuntu 20.04.1 LTS или новее;
- Веб-сервер: Nginx, PHP 7.4.

Минимальные системные требования для работы клиентской части программного обеспечения «Интерактивная 3D «Физика. 8 класс»:

- Оперативная память: 4 Гб и больше;
- ОС: Windows, Linux, MacOS;
- Веб-браузер: Google Chrome 57 и новее (а также аналоги, созданные на основе Chromium).

3. Инструкция по установке

Данная инструкция содержит описание процесса установки следующего ПО:

- «Интерактивная 3D «Физика. 8 класс».

Перед началом установки необходимо полностью прочитать настоящую инструкцию, в случае возникновения вопросов, связаться с системным администратором. До разъяснения возникших вопросов не следует приступать к установке.

В случае возникновения ошибок в процессе установки или если установка не может быть произведена по каким-либо причинам, следует обратиться к системному администратору.

3.1. Инструкция по размещению ПО на сервере

```
## Учебник
```

```
### Настройка сервера
```

```
> OS Ubuntu 20.04.1 LTS
```

```
#### Первичная настройка Firewall
```

```
...
```

```
// если нужно активируем IPV6=yes  
sudo nano /etc/default/ufw
```

```
// дефолтные настройки  
sudo ufw default deny incoming  
sudo ufw default allow outgoing  
...
```

```
Добавить разрешение на SSH
```

```
...
```

```
sudo ufw allow OpenSSH  
...
```

```
Активировать firewall
```

```
...
```

```
sudo ufw enable
```

```
// статус  
sudo ufw status  
...
```

```
## LEMP server
```

```
#### Установка Nginx
```

```
...
```

```
sudo apt install nginx
```

```
// добавить правило в firewall ( Nginx Full - http и https )  
sudo ufw allow 'Nginx Full'
```

```
// проверить  
sudo ufw status  
...
```

```
#### Установка MySQL
```

```
...
```

```

// mysql 8.0
sudo apt install mysql-server

// настройка
sudo mysql_secure_installation
// password plugin - no
// change root password - yes
// остальные настройки - yes
'''

#### Установка PHP 7.4
'''

sudo apt install php7.4 php7.4-fpm php7.4-mysql php7.4-mbstring php7.4-xml php7.4-gmp php7.4-
curl php7.4-zip php7.4-gd php7.4-redis
'''

'''

// правки в файле конфигурации
sudo nano /etc/php/7.4/fpm/php.ini
'''

php.ini

'''

// Раскомментируйте параметр и измените его значение на 0
cgi.fix_pathinfo = 0

post_max_size = 50M
upload_max_filesize = 50M

max_file_uploads = 50

// для composer
memory_limit = -1
'''

После правки ини файла сохранить его и перезапустить php

'''

sudo systemctl restart php7.4-fpm
'''

#### Установка Git

'''

sudo apt install git
'''

#### Установка Composer

'''

sudo apt install unzip

```

```
cd ~
curl -sS https://getcomposer.org/installer -o composer-setup.php

// верификация установки - ключ тут - https://composer.github.io/pubkeys.html
HASH=тут хеш с сайта выше

php -r "if (hash_file('SHA384', 'composer-setup.php') === '$HASH') { echo 'Installer verified'; } else {
echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"

// В результате должно быть - Installer verified

sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer
'''

#### Установка Redis

'''
sudo apt install redis-server

// открыть конфиг
sudo nano /etc/redis/redis.conf
'''

В конфиг должны быть следующие настройки

'''
// только локальные соединения
bind 127.0.0.1 ::1

// автозапуск - пункт отсутствует в Ubuntu 16
supervised systemd
'''

Проверить

'''
// рестарт
sudo systemctl restart redis.service

// статус сервера - должен быть active
sudo systemctl status redis

'''

## Настройка

#### Создать нового пользователя (без root прав)

'''
sudo adduser www-root
'''
```

Добавить папку для проекта

'''

```
sudo mkdir -p /var/www/staging
sudo chown www-root:www-root /var/www/staging
```

'''

Создать нового пользователя для БД и саму БД

'''

```
sudo mysql -u root -p
```

'''

Создаем новую базу "staging":

'''

```
create database staging;
```

'''

Просмотреть список текущих баз данных:

'''

```
SHOW DATABASES;
```

'''

Создаем пользователя MySQL с именем staging_user и паролем 'pass':

'''

```
create user 'staging_user'@'localhost' identified by 'pass';
```

'''

Назначаем привилегии пользователю:

'''

```
GRANT ALL PRIVILEGES ON staging.* TO 'app_user'@'localhost';
```

'''

Перезагрузить права

'''

```
FLUSH PRIVILEGES;
```

'''

Выход из консоли mysql

'''

```
quit;
```

'''

Настройка приложения


```
#### Добавить публичный ключ для пользователя www-root
```

```
конфиг ssh
```

```
...
```

```
sudo nano /etc/ssh/sshd_config
```

```
...
```

```
// вход только по ключам
```

```
...
```

```
PasswordAuthentication no
```

```
...
```

```
// перезагрузить ssh
```

```
...
```

```
sudo service ssh restart
```

```
...
```

```
...
```

```
sudo mkdir -p /home/www-root/.ssh && sudo touch /home/www-root/.ssh/authorized_keys
```

```
// вставить публичный ключ
```

```
sudo nano /home/www-root/.ssh/authorized_keys
```

```
sudo chmod 700 /home/www-root/.ssh && sudo chmod 600 /home/www-root/.ssh/authorized_keys
```

```
chown -R www-root:www-root /home/www-root/.ssh
```

```
...
```

```
#### Добавить пул для php-fpm
```

```
Создать новый конфиг
```

```
...
```

```
sudo nano /etc/php/7.4/fpm/pool.d/staging.conf
```

```
...
```

```
Содержимое конфига
```

```
...
```

```
[staging]
```

```
; Unix user/group of processes
```

```
user = www-root
```

```
group = www-root
```

```
listen = /run/php/php7.4-fpm.staging.sock
```

```
listen.owner = www-data
```

```
listen.group = www-data
```

```
;listen.mode = 0660
```

```
pm = dynamic
```

```
pm.max_children = 15
```

```
pm.start_servers = 6
pm.min_spare_servers = 4
pm.max_spare_servers = 8
'''
```

Перезапустить PHP-FPM

```
'''
sudo systemctl restart php7.4-fpm
'''
```

Настройка Nginx

Добавить новую конфигурацию для домена

```
'''
sudo nano /etc/nginx/sites-available/satging
'''

'''
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    server_name book.vr-labs.ru www.book.vr-labs.ru;

    return 301 https://.ru:443$request_uri;
}

server {
    listen [::]:443 ssl ipv6only=on;
    listen 443 ssl;
    server_name www.book.vr-labs.ru;

    return 301 https://.ru$request_uri;
}

server {
    listen 443 ssl;
    server_name book.vr-labs.ru www.book.vr-labs.ru;

    set $root_path /var/www/staging/current/public;
    root $root_path;

    access_log /var/log/nginx/staging-access.log;
    error_log /var/log/nginx/staging-error.log;

    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-XSS-Protection "1; mode=block";
    add_header X-Content-Type-Options "nosniff";

    index index.html index.htm index.php;
}
'''
```

```

charset utf-8;

disable_symlinks if_not_owner from=$root_path;

# Unity
location ~ /\.unityweb$ {
    add_header Content-Encoding gzip;
}

# Static Content
location ~* ^\.+\.(jpg|jpeg|gif|png|ico|tiff|css|js)$ {
    if ($request_method = 'OPTIONS') {
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
        #
        # Custom headers and headers various browsers *should* be OK with but aren't
        #
        add_header 'Access-Control-Allow-Headers' 'DNT,User-Agent,X-Requested-With,If-Modified-
Since,Cache-Control,Content-Type,Range';
        #
        # Tell client that this pre-flight info is valid for 20 days
        #
        add_header 'Access-Control-Max-Age' 1728000;
        add_header 'Content-Type' 'text/plain; charset=utf-8';
        add_header 'Content-Length' 0;
        return 204;
    }
    if ($request_method = 'POST') {
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
        add_header 'Access-Control-Allow-Headers' 'DNT,User-Agent,X-Requested-With,If-Modified-
Since,Cache-Control,Content-Type,Range';
        add_header 'Access-Control-Expose-Headers' 'Content-Length,Content-Range';
    }
    if ($request_method = 'GET') {
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
        add_header 'Access-Control-Allow-Headers' 'DNT,User-Agent,X-Requested-With,If-Modified-
Since,Cache-Control,Content-Type,Range';
        add_header 'Access-Control-Expose-Headers' 'Content-Length,Content-Range';
    }
}

expires 6M;
add_header Cache-Control public;
}

location / {
    try_files $uri $uri/ /index.php?$query_string;
}

location = /favicon.ico { access_log off; log_not_found off; }

```

```

location = /robots.txt { access_log off; log_not_found off; }

error_page 404 /index.php;

location ~ /\.php$ {
    fastcgi_split_path_info ^(.+\.(php|php5|php7|php8|php9|phpcgi|phpt))(/.+)$;
    fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
}

location ~ /\.(!well-known).* {
    deny all;
}

}
...

```

Создать симлинк

```

...
sudo ln -s /etc/nginx/sites-available/staging /etc/nginx/sites-enabled/
...

```

Тестирование конфигурации и перезагрузка настроек

```

...
sudo nginx -t
sudo systemctl reload nginx
...

```

Установка и настройка сайта

!!! Вход от пользователя www-root

...

```

// выбрать папку приложения
cd /var/www/staging/current

```

```

// клонировать репозиторий
git clone https://gitlab.vizex.ru/demo-apps/schoolbookscan.git .

```

```

// копируем файлы настроек приложения
cp .env.example .env

```

```

// Редактируем файл настроек .env (Необходимо прописать пароли от БД, почту)
nano .env
...

```

Установка пакетов

```

...
composer install --no-dev
...

Сгенерировать ключи в файл .env

...
php artisan key:generate
...

Миграции БД

...
php artisan migrate --seed
...

Задать права на файлы и папки

...
find /var/www/staging/current -type f -exec chmod 644 {} \;
find /var/www/staging/current -type d -exec chmod 755 {} \;
find /var/www/staging/current/storage/* -type d -exec chmod 775 {} \;
find /var/www/staging/current/bootstrap/cache -type d -exec chmod 775 {} \;
...

Добавить задание для Cron

...
sudo crontab -e -u www-root

**** cd /var/www/staging/current && php artisan schedule:run >> /dev/null 2>&1
...

Установка окончена!

### Environments - для Laravel Envoy

...
local    - для локальной разработки - localhost
testing  - для запуска тестов      - localhost
staging  - обкатка                  - book.vr-labs.ru
...

#### Инициализация среды

...
// staging
php vendor/bin/envoy run staging:init --env=staging

```

```
// production
php vendor/bin/envoy run production:init --env=production
...
```

```
#### Запуск деплоя на staging
```

```
...
php vendor/bin/envoy run staging:deploy --env=staging
```

```
...
```

```
#### Запуск деплоя на production
```

```
...
php vendor/bin/envoy run production:deploy --env=production
```

```
...
```

В случае, если процедура установки будет изменена в связи с доработками приложения, об этом будет уведомлено отдельно.

4. Техническая поддержка

4.1. Устранение ошибок

Для более быстрого решения возникшей проблемы, частые вопросы, возникающие у Пользователей, будут занесены в руководство пользователя.

При возникновении ошибок, связанных с работой программного обеспечения или необходимости получения информации по работе продукта, следует отправить запрос на электронную почту технической поддержки или по номеру телефона «горячей линии». Консультации будут оказаны непосредственно Пользователям устно по телефону «горячей линии» или письменно по электронной почте, согласно регламенту «О технической поддержке ПО «Виртуальная лаборатория».

В случае сложных обращений, требующих глубокой проработки и больших трудозатрат специалистов технической поддержки, Пользователю, отправившему обращение, будет отправлено письмо с указанием этапа проработки обращения и сроки выполнения обращения.

Если после получения ответа на обращение Пользователь считает его не выполненным или качество исполнения неудовлетворительное, то следует отправить повторный запрос с указанием причин неудовлетворенности решением, и запрос возобновляется.

Обращения выполняются по мере их поступления. В случае во время поступления обращения специалист работает над решением другой задачи, то обращение ставиться «в очередь». Порядок обработки обращений определяется их приоритетом.

Вся переписка, обмен файлами и т. д. фиксируются и сохраняются в электронном архиве.