

Подключение аутентификация к сайту с помощью фреймворка Laravel

Ульянов Егор Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается и описывается интеграция аутентификации к сайту при помощи фреймворка «Laravel», и библиотеки «ui». Практическим результатом является рабочая логика аутентификации на сайте.

Ключевые слова: сайт, фреймворк, аутентификация, библиотека

Connection authentication to the site using the Laravel framework

Ulianov Egor Andreevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses and describes the integration of authentication to the site using the "Laravel" framework and the "ui" library. The practical result is a working authentication logic on the site.

Keywords: website, framework, authentication, library

Аутентификация - процедура проверки подлинности, пользователя путём сравнения введённого им пароля (для указанного логина) с паролем, сохранённым в базе данных пользовательских логинов. На сегодняшний день каждый сайт имеет процедуру аутентификации.

Цель данной статьи создать аутентификацию на сайт с помощью фреймворка Laravel.

В статье С.М. Гардейчик, А.И. Шербаф приводится краткое описание PHP-фреймворка Laravel, реализующего шаблон MVC [1]. Д.А. Баранов в своей статье рассмотрел возможности ORM-фреймворка Ларавель, проанализировал стандартные и продвинутое средства и спецификации их применения, а также описал, что представляет собой библиотека, продемонстрировал использование некоторых функций библиотеки [2]. Свою работу Якимов А. С., Баженов Р. И. посвятили пошаговому описанию инструкции по разработке модуля страниц для панели администратора, основанной на фреймворке Laravel [3].

После создания нового или открытия старого проекта необходимо скачать и установить пакетный менеджер «Composer» необходимый для установки и подключения различных библиотек, далее необходимо

установить пакетный менеджер в проект прописав команды, указанные на сайте в терминал среды разработки (рис.1-2).

[Home](#) | [Getting Started](#) | [Download](#) | [Documentation](#) | [Browse Packages](#)

Download Composer Latest: v2.2.5

Windows Installer

The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

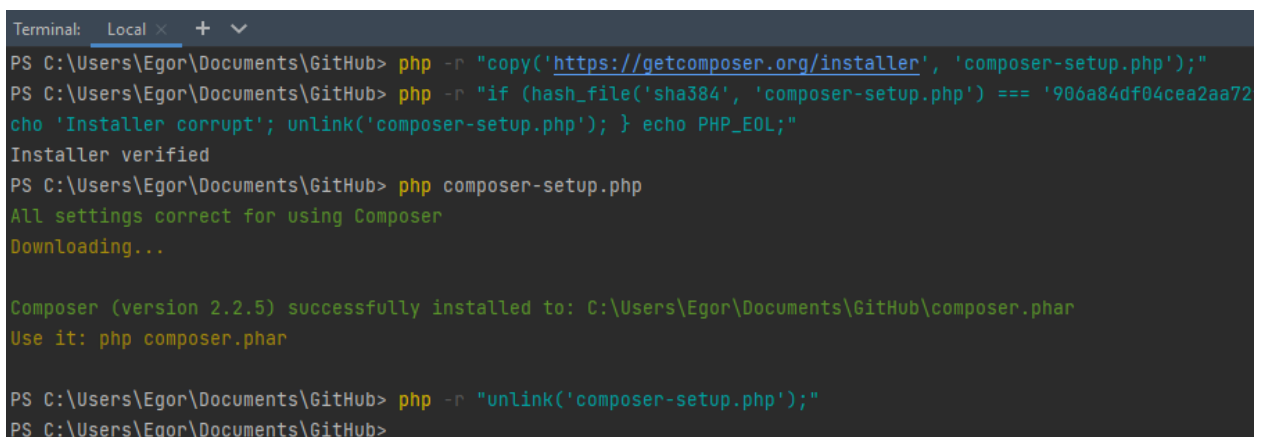
Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === '906a84df04cea2aa72f40b5f787e49f22d4c2f19492ac310e8c1
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

Рисунок 1 – Скачивание пакетного менеджера



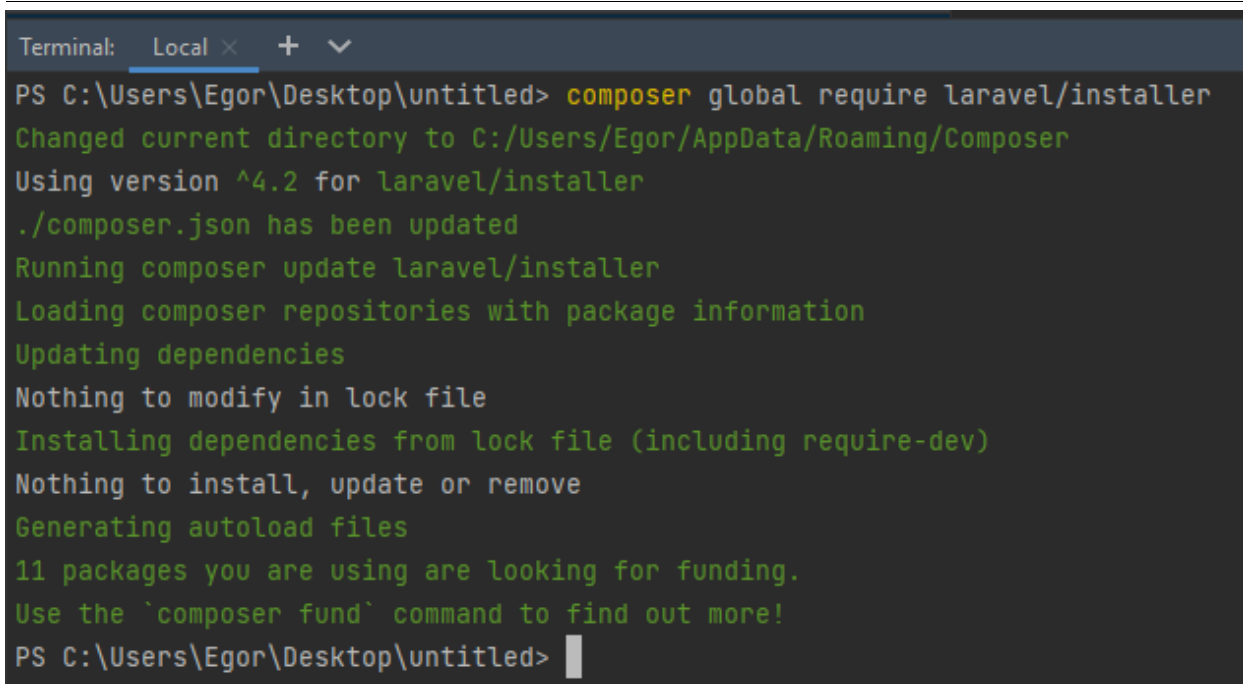
```
Terminal: Local x + v
PS C:\Users\Egor\Documents\GitHub> php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
PS C:\Users\Egor\Documents\GitHub> php -r "if (hash_file('sha384', 'composer-setup.php') === '906a84df04cea2aa72f40b5f787e49f22d4c2f19492ac310e8c1
cho 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
Installer verified
PS C:\Users\Egor\Documents\GitHub> php composer-setup.php
All settings correct for using Composer
Downloading...

Composer (version 2.2.5) successfully installed to: C:\Users\Egor\Documents\GitHub\composer.phar
Use it: php composer.phar

PS C:\Users\Egor\Documents\GitHub> php -r "unlink('composer-setup.php');"
PS C:\Users\Egor\Documents\GitHub>
```

Рисунок 2 – Установка менеджера в проект

Теперь устанавливаем сам фреймворк «Laravel». Для этого опять обратимся к пакетному менеджеру и пропишем команду «`composer global require laravel/installer`» (рис.3).

A screenshot of a Windows terminal window. The title bar shows 'Terminal: Local x + v'. The command prompt shows the execution of 'composer global require laravel/installer'. The output text is as follows:

```
PS C:\Users\Egor\Desktop\untitled> composer global require laravel/installer
Changed current directory to C:/Users/Egor/AppData/Roaming/Composer
Using version ^4.2 for laravel/installer
./composer.json has been updated
Running composer update laravel/installer
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating autoload files
11 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
PS C:\Users\Egor\Desktop\untitled> |
```

Рисунок 3 – Установка фреймворка

Далее необходимо скачать файлы фреймворка, для этого создаем новый «Laravel» проект используя команду «`laravel new ****`», где **** имя проекта (рис.4-5).

```
PS C:\Users\Egor\Desktop\untitled> laravel new app

- - - - -
| | | | | | | |
| | / _ | ' _ / _ \ \ / _ \ |
| | ___| C | | | C | \ \ / _ \ |
| | ___| \ _ / | | \ _ / _ \ |
- - - - -

Creating a "laravel/laravel" project at "./app"
Installing laravel/laravel (v8.6.10)
- Installing laravel/laravel (v8.6.10): Extracting archive
Created project in C:\Users\Egor\Desktop\untitled\app
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 106 installs, 0 updates, 0 removals
- Locking asm89/stack-cors (v2.1.1)
- Locking brick/math (0.9.3)
- Locking doctrine/inflector (2.0.4)
- Locking doctrine/instantiator (1.4.0)
- Locking doctrine/lexer (1.2.2)
- Locking dragonmantank/cron-expression (v3.3.1)
- Locking egulias/email-validator (2.1.25)
- Locking facade/flare-client-php (1.9.1)
- Locking facade/ignition (2.17.4)
- Locking facade/ignition-contracts (1.0.2)
- Locking fakerphp/faker (v1.17.0)
- Locking filp/whoops (2.14.5)
- Locking fruitcake/laravel-cors (v2.0.5)
- Locking graham-campbell/result-type (v1.0.4)
- Locking guzzlehttp/guzzle (7.4.1)
- Locking guzzlehttp/promises (1.5.1)
- Locking guzzlehttp/psr7 (2.1.0)
- Locking hamcrest/hamcrest-php (v2.0.1)
```

Рисунок 4 – Создание «Laravel» проекта

```
77 package suggestions were added by new dependencies, use `composer suggest` to see details.
Package swiftmailer/swiftmailer is abandoned, you should avoid using it. Use symfony/mailer instead.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/sail
Discovered Package: laravel/sanctum
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
76 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force
No publishable resources for tag [laravel-assets].
Publishing complete.
> @php artisan key:generate --ansi
Application key set successfully.
Application ready! Build something amazing.
```

Рисунок 5 – Создание «Laravel» проекта

После создания проекта необходимо создать базу данных, где будут храниться пользовательские данные, в данном случае будет использоваться локальный сервер «OpenServer» и веб-приложение «phpMyAdmin», представляющее собой веб-интерфейс для администрирования СУБД «MySQL». Запускаем «OpenServer» и в трее по щелчку правой кнопки мыши переходим в пункт дополнительно, «phpMyAdmin». После авторизации создаем пустую базу данных (рис.6-7).

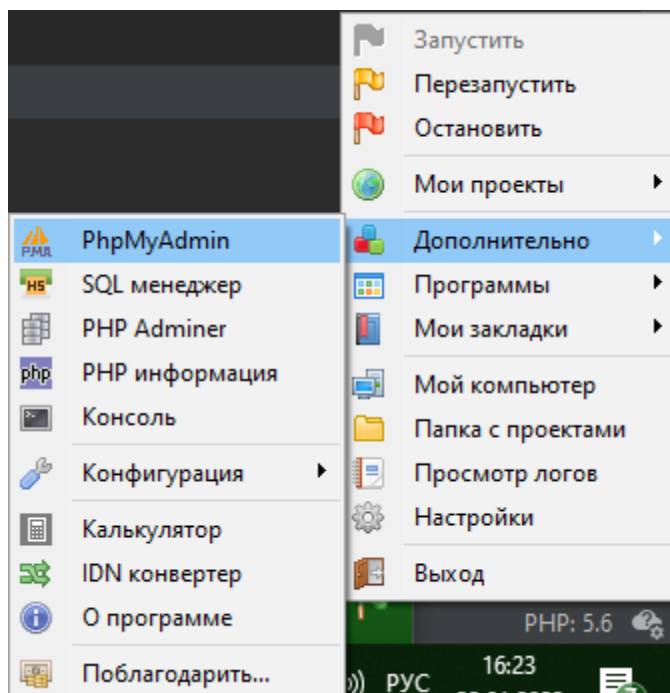


Рисунок 6 – Открытие веб-интерфейса «phpMyAdmin»

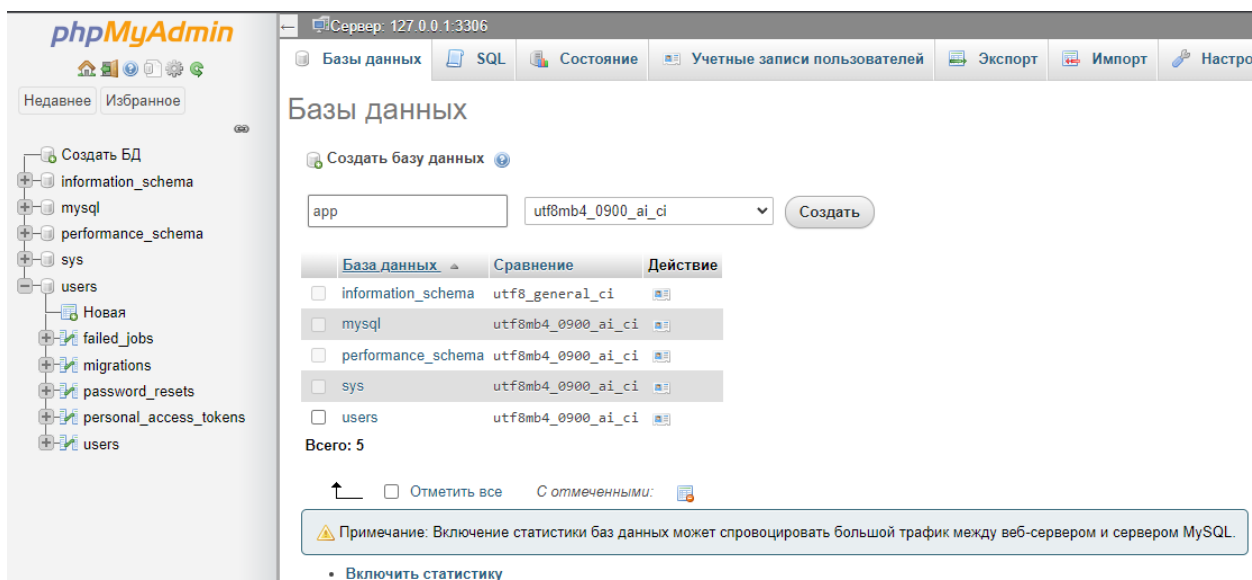
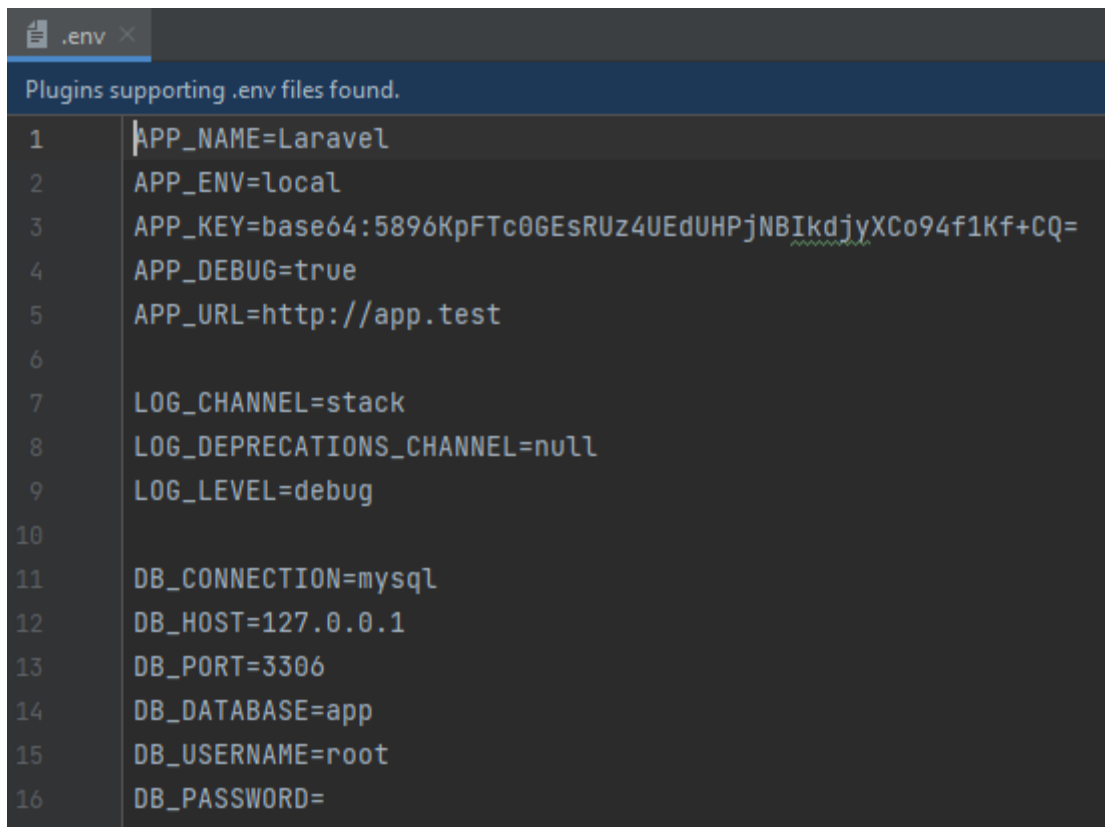


Рисунок 7 – Создание базы данных

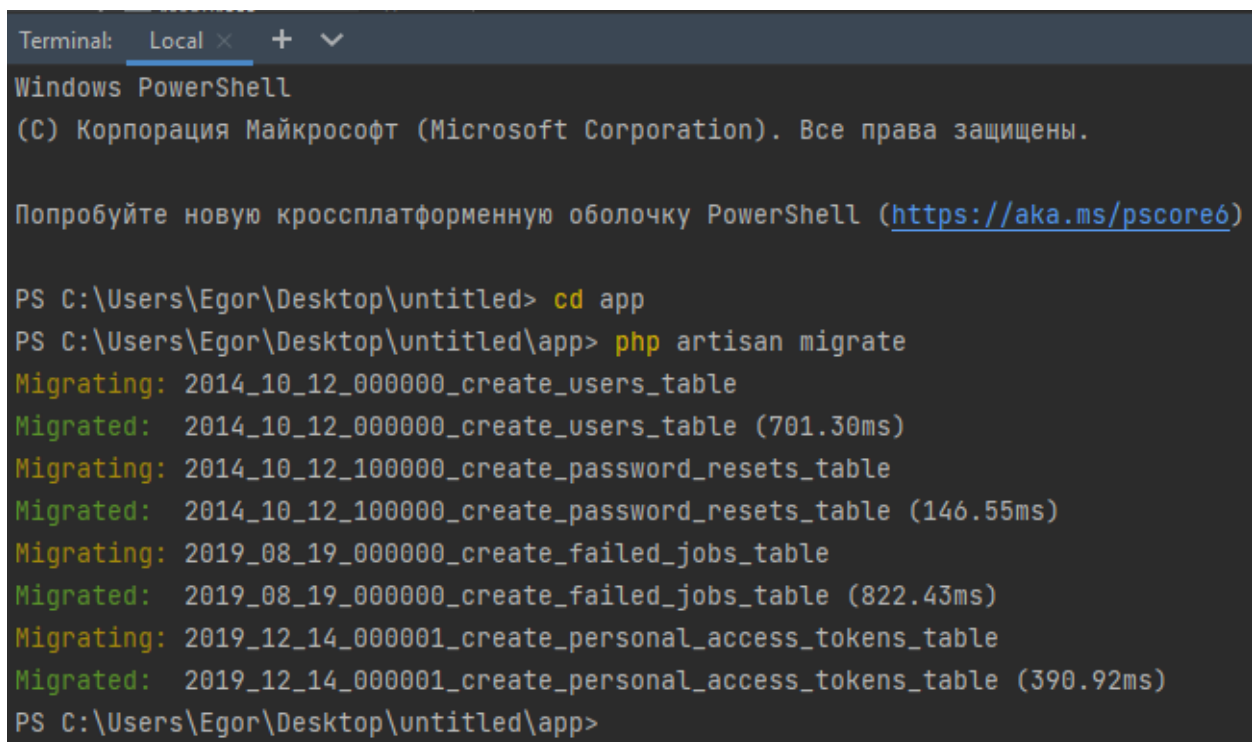
Для подключения базы данных к сайту необходимо в файле «.env» прописать название СУБД, адрес, порт, название БД, а также логин и пароль к «phpMyAdmin» (рис.8).



```
.env x
Plugins supporting .env files found.
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:5896KpFTc0GEsRUz4UEdUHPjNBikdjyXCo94f1Kf+CQ=
4 APP_DEBUG=true
5 APP_URL=http://app.test
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATED_CHANNELS=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=app
15 DB_USERNAME=root
16 DB_PASSWORD=
```

Рисунок 8 – Подключение базы данных

Далее в терминале переходим в папку с проектом и прописываем команду переноса БД на сервер (рис.9).



```
Terminal: Local x + v
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS C:\Users\Egor\Desktop\untitled> cd app
PS C:\Users\Egor\Desktop\untitled\app> php artisan migrate
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (701.30ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (146.55ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (822.43ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (390.92ms)
PS C:\Users\Egor\Desktop\untitled\app>
```

Рисунок 9 – Перенос базы данных

Для подключения аутентификации необходимо подключить дополнительную библиотеку «Laravel» под названием «ui», чтобы это сделать необходимо в терминале прописать команду «composer require laravel/ui» (рис.10).

```
PS C:\Users\Egor\Desktop\untitled\app> composer require laravel/ui
Using version ^3.4 for laravel/ui
./composer.json has been updated
Running composer update laravel/ui
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
  - Locking laravel/ui (v3.4.1)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Installing laravel/ui (v3.4.1): Extracting archive
Package swiftmailer/swiftmailer is abandoned, you should avoid using it. Use symfony/mailer instead.
Generating optimized autoload files
```

Рисунок 10 – Подключение дополнительной библиотеки

Запускаем специальный скрипт из библиотеки командой «php artisan ui:auth» (рис.11).

```
PS C:\Users\Egor\Desktop\untitled\app> php artisan ui:auth
Authentication scaffolding generated successfully.
```

Рисунок 11 – Подключение дополнительной библиотеки

Для проверки работы аутентификации сайта запустим локальный сервер (рис.12).

```
PS C:\Users\Egor\Desktop\untitled\app> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[Sat Jan 22 16:44:33 2022] 127.0.0.1:59922 [200]: /favicon.ico
```

Рисунок 12 – Запуск локального сервера

Переходим на сайт и в верхнем левом углу автоматически создались кнопки регистрации и авторизации, проверяем регистрацию (рис.13-14).

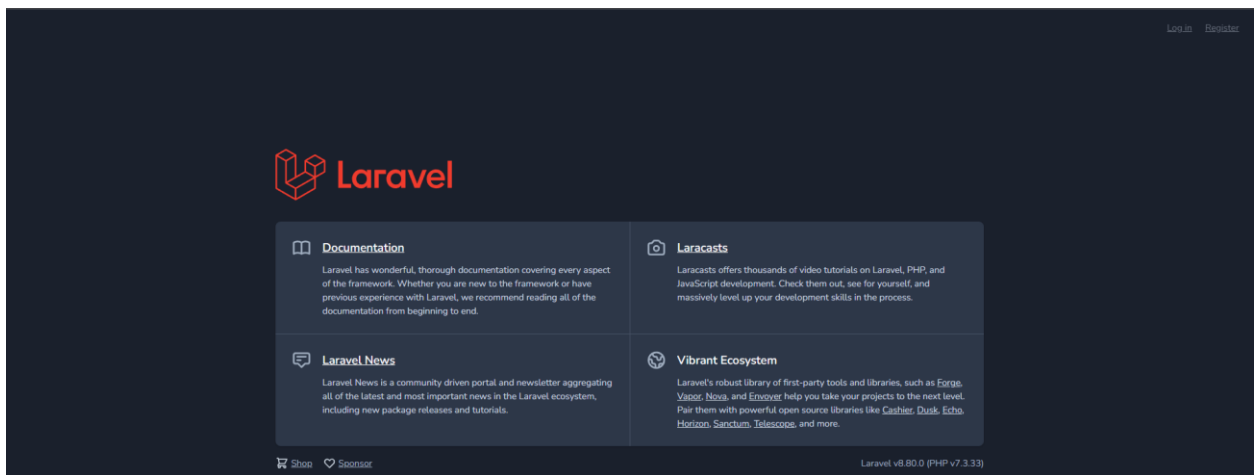


Рисунок 13 – Начальная страница с кнопками регистрации и авторизации

[Laravel](#)

- [Login](#)
- [Register](#)

Register

Name

E-Mail Address

Password

Confirm Password

Рисунок 14 – Форма регистрации пользователя

После регистрации попадаем на страницу с сообщением об успешном входе. Также запись о регистрации можно увидеть в таблице «users» базы данных сайта (рис.15-16).

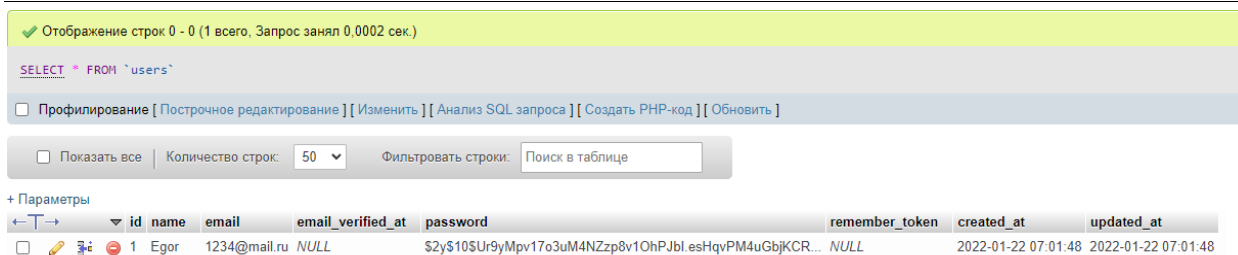
[Laravel](#)

- [Egor](#)
- [Logout](#)

Dashboard

You are logged in!

Рисунок 15 – Сообщение об успешном входе



Отображение строк 0 - 0 (1 всего, Запрос занял 0,0002 сек.)

```
SELECT * FROM `users`
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице

+ Параметры

	id	name	email	email_verified_at	password	remember_token	created_at	updated_at
1	Egor	1234@mail.ru	NULL		\$2y\$10\$Ur9yMpv17o3uM4NZp8v1OhPJbl esHqvPM4uGbjKCR...	NULL	2022-01-22 07:01:48	2022-01-22 07:01:48

Рисунок 16 – Запись в базе данных

Аутентификация работает отлично и требует всего лишь визуальной доработки.

В данной статье была рассмотрена возможность легкой интеграции аутентификации к сайту с использованием фреймворка «Laravel». Практическим результатом является рабочая аутентификация сайта.

Библиографический список

1. Гардейчик С.М., Шербаф А.И. PHP-Фреймворк Laravel с использованием архитектурной модели MVC // Перспективные направления развития отечественных информационных технологий. 2017. С. 133-135.
2. Баранов Д.А. Laravel eloquent orm // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2018. №3. С. 73-79.
3. Якимов А. С., Баженов Р. И. Разработка модуля страниц для панели администратора на laravel // Постулат. 2017. №1(15). С. 46.