**Создание проекта на Laravel**

# Содержание

[Содержание 2](#_Toc163765872)

[1 Подготовка 3](#_Toc163765873)

[1.1 PHP 3](#_Toc163765874)

[1.2 Composer 4](#_Toc163765875)

[2. Создание проекта 5](#_Toc163765876)

[2.1 Инициализация Laravel 5](#_Toc163765877)

[2.2 Модели 5](#_Toc163765878)

[2.3 Маршруты 7](#_Toc163765879)

[2.4 Шаблоны 8](#_Toc163765880)

[2.5 Формы 9](#_Toc163765881)

[2.6 Авторизация 10](#_Toc163765882)

# 1 Подготовка

Для создания приложения на Laravel для начала нужно настроить ПК и скачать необходимое ПО.

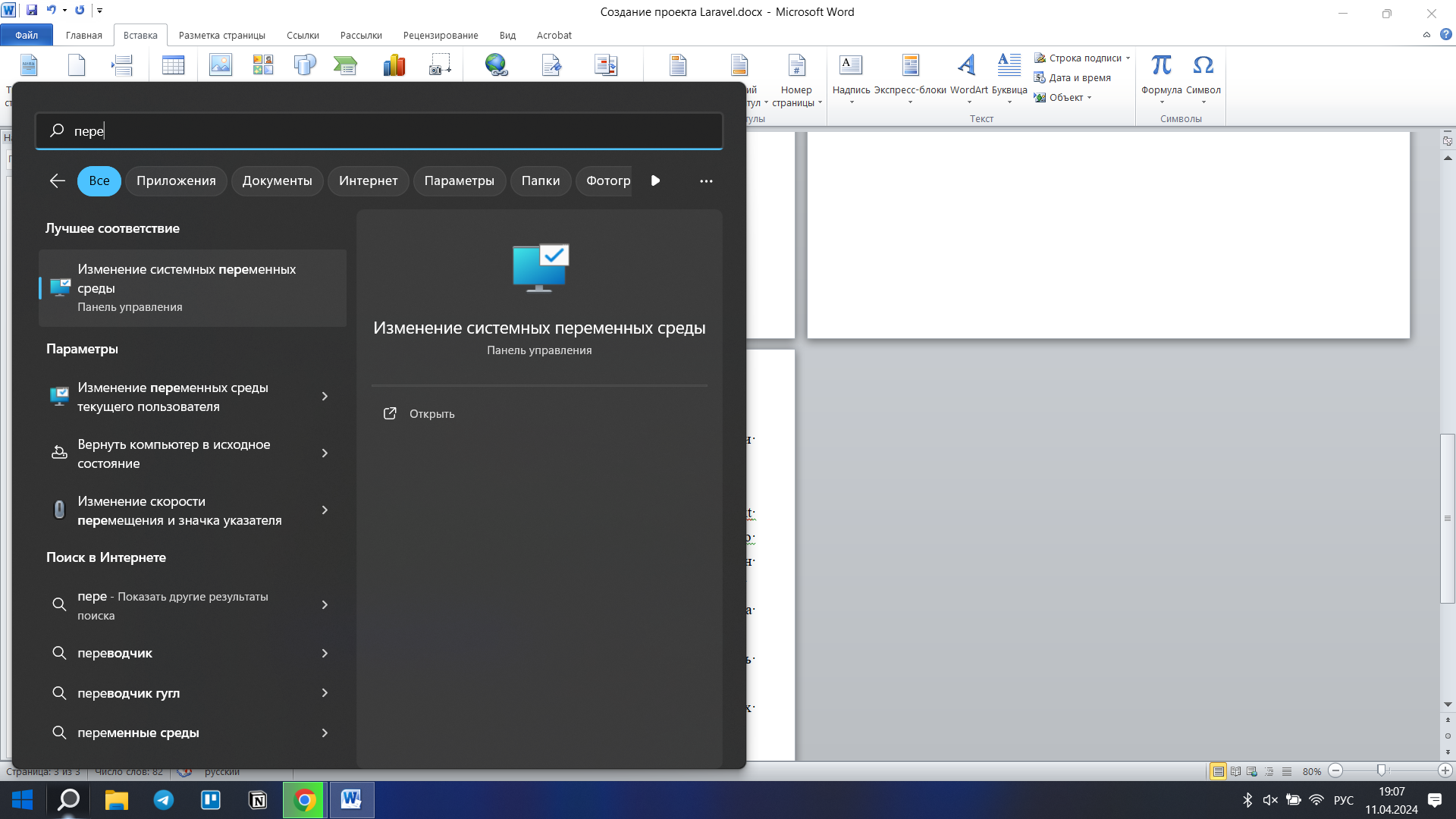
## 1.1 PHP

PHP (рекурсивный акроним словосочетания PHP: Hypertext Preprocessor) - это распространённый язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML.

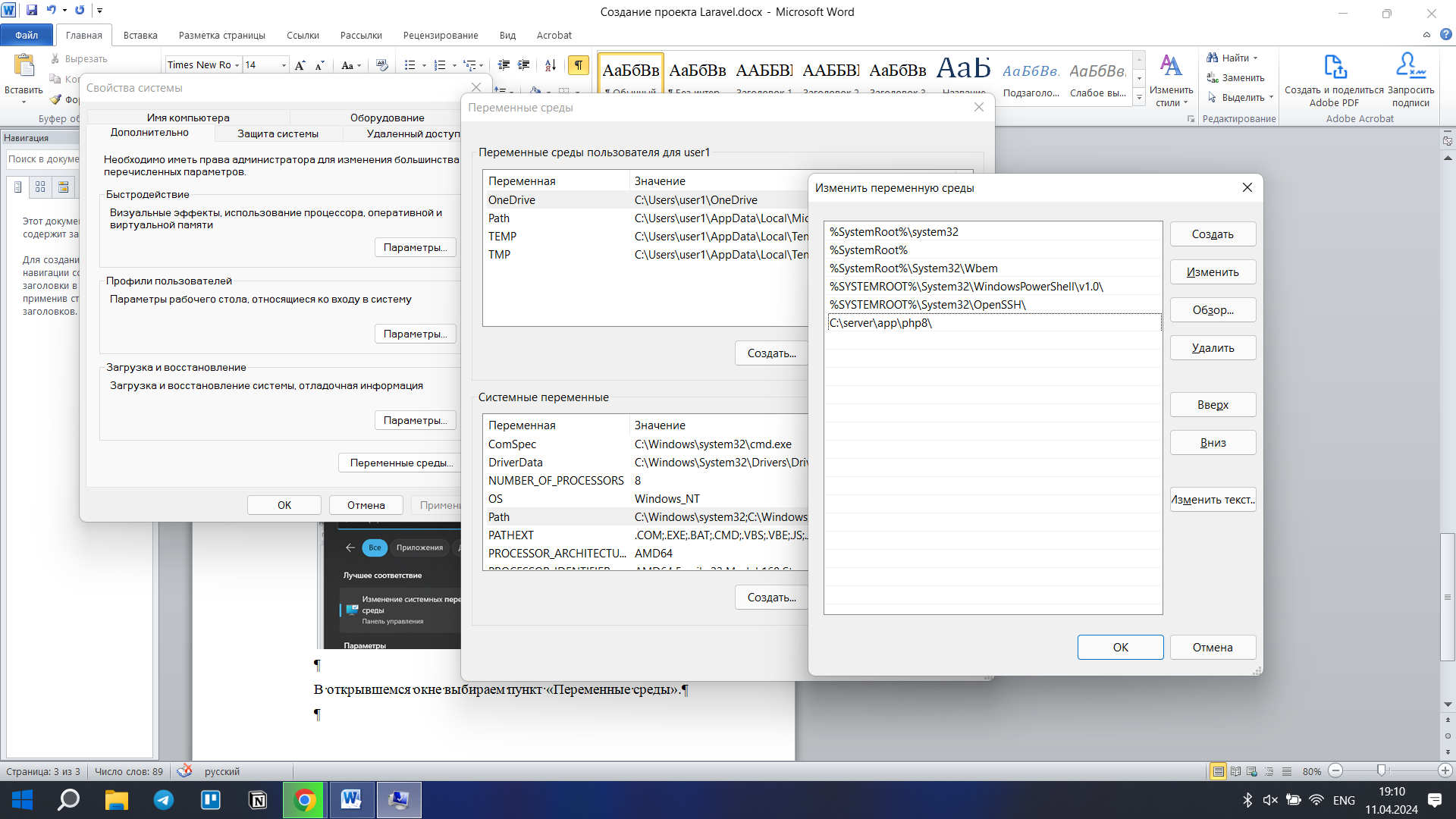
Для начала скачиваем архив с сайта <https://windows.php.net/download#php-8.3>.

Затем необходимо распаковать архив. Рекомендуется распаковать архив в C:/server/app/php8.

Затем необходимо прописать путь к PHP в переменных средах системы. Для этого в поиске вводим «Переменные среды».



В открывшемся окне выбираем пункт «Переменные среды».



Добавляем путь к C:/server/app/php8.

Чтобы проверить, всё ли корректно настроено, необходимо запустить PowerShell и выполнить команду «php –v».

Если всё настроено верно, то в консоль вернётся текущая версия PHP.

## 1.2 Composer

Composer — пакетный менеджер уровня приложений для языка программирования PHP, который предоставляет средства по управлению зависимостями в PHP-приложении.

Для установки Composer необходимо перейти на официальный сайт <https://getcomposer.org/download/> и скачать установщик.

Установка происходит по принципу Далее – Далее – Далее – Готово.

Для проверки правильности установки в PowerShell необходимо выполнить команду «composer -v».

Должна появиться надпись Composer.

# 2. Создание проекта

## 2.1 Инициализация Laravel

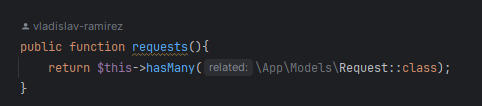
Для инициализации проекта на Laravel необходимо перейти в рабочую папку и выполнить в терминале команду «composer create-project laravel/laravel .».

Composer автоматически создаст структуру папок и файлов.

После успешной установки можно выполнить команду «php artisan serve», которая запустит тестовый сервер, и в браузере откроется приветственная страница Laravel.

## 2.2 Модели

Модель User создаётся Laravel автоматически. Для пользователя необходимо получать список его заявок, поэтому добавим метод в модель.



Модель Request описывает заявку. Для создания модели необходимо в консоли выполнить команду «php artisan make:model Request -cm». Флаги c, m указываются для создания контроллера и миграции. Настраиваем связи в app\models\request



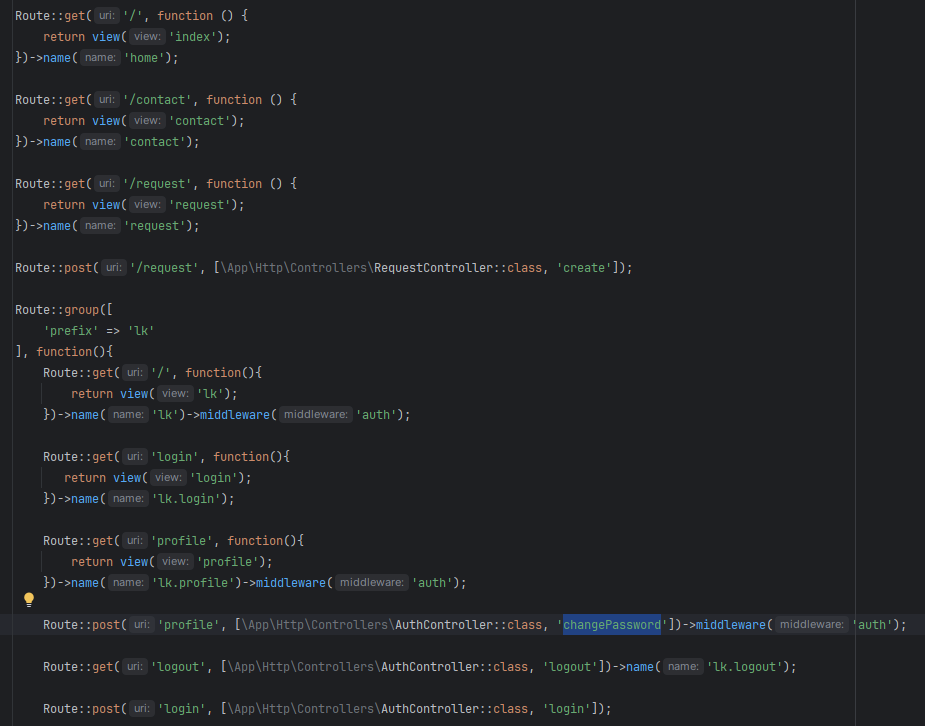
В файле миграции описываем структуру таблицы заявок.



После описания структуры таблицы выполним миграцию. Для этого в терминале необходимо выполнить команду «php artisan migrate». Таблица будет создана. Таблицы создаются в БД sqlite. Для MySQL действия аналогичны.

## 2.3 Маршруты

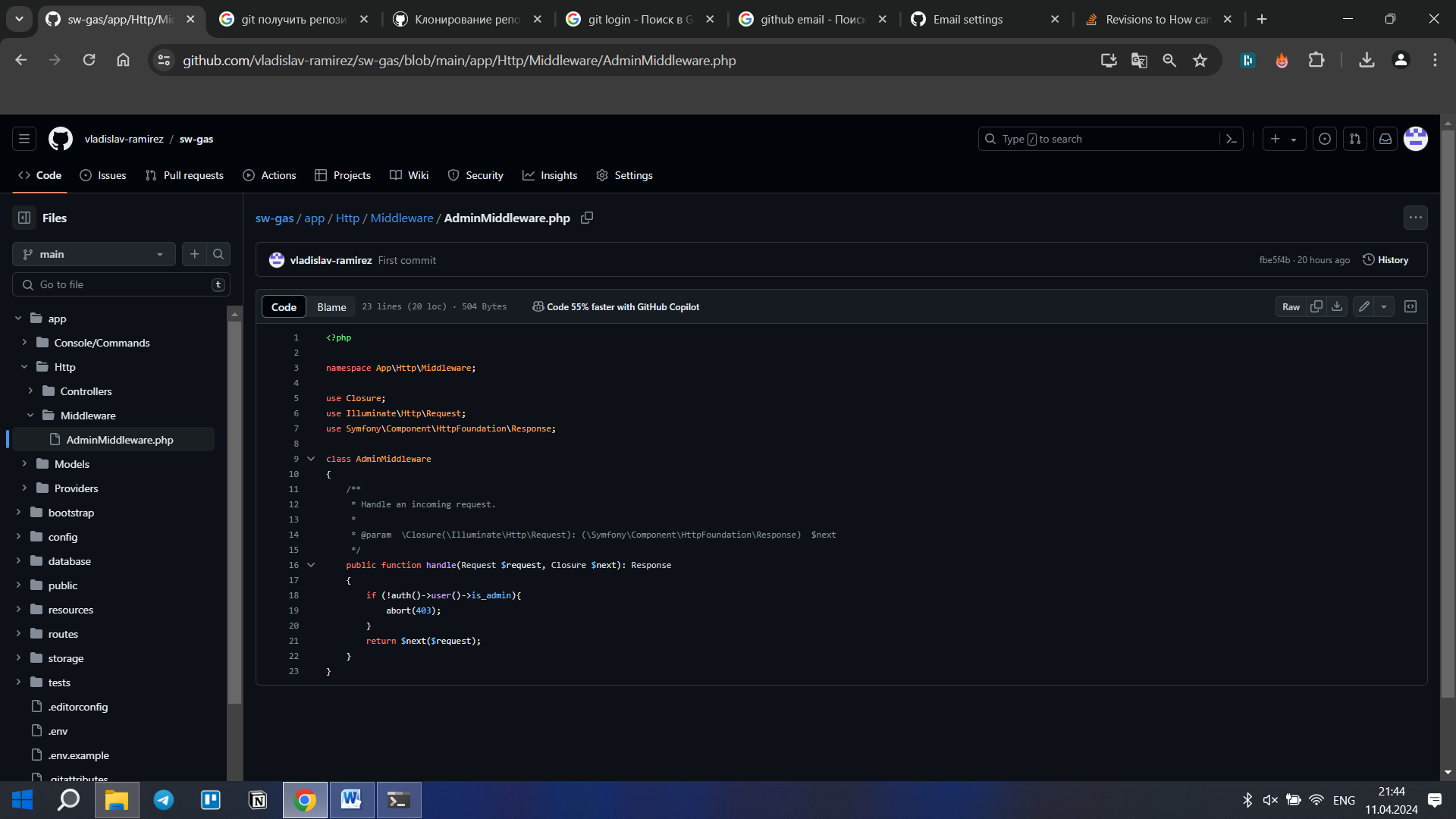
Маршруты настраиваются в routes/web.php



Для маршрутов, к которым требуется доступ только авторизованных пользователей, необходимо применить Middleware «auth». Это штатный посредник.

Для контроля аккаунта администратора создадим своего посредника: «php artisan make:middleware AdminMiddleware ». Посредник появится в App\Http\Middleware.

Добавим код, который будет проверять флаг is\_admin для пользователя.

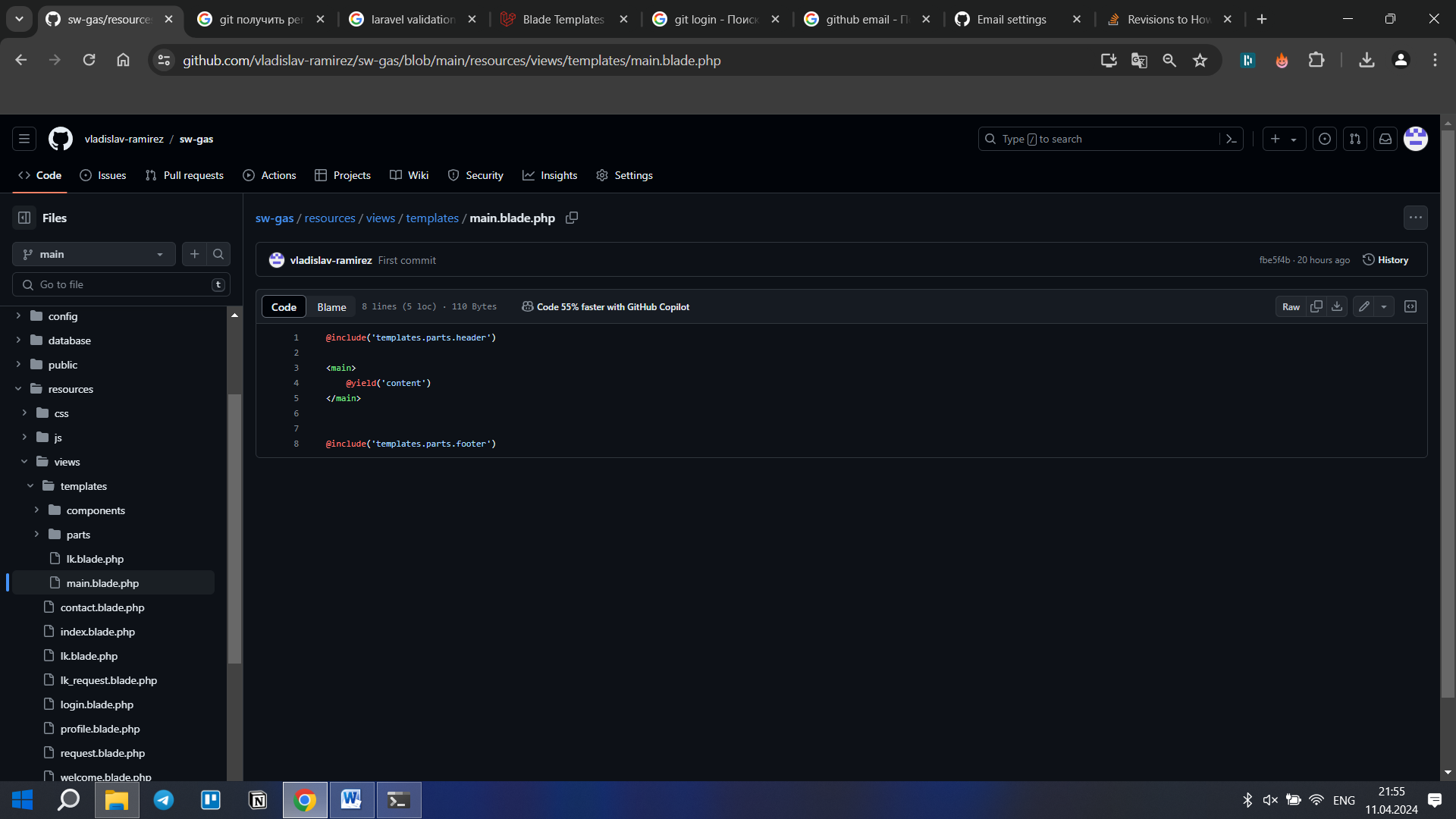


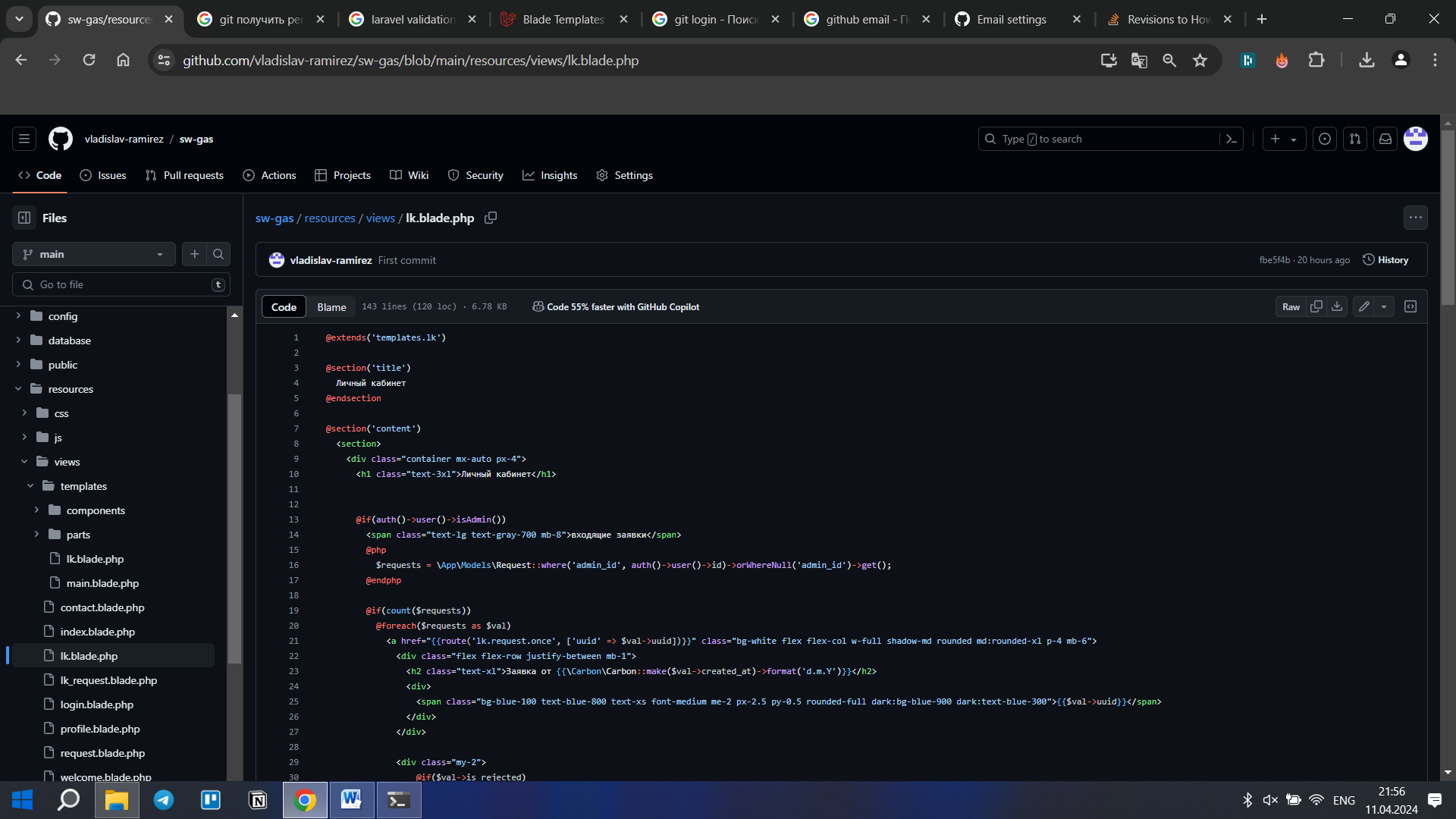
## 2.4 Шаблоны

Шаблоны расположены в папке /resourses/views/, и имеют расширение .blade.php. Для шаблонов Laravel использует шаблонизатор Blade. Полная документация расположена на сайте <https://laravel.com/docs/11.x/blade>.

В blade используются директивы @if, @section, @php, @foreach.

Они позволяют отрисовывать нужные элементы HTML основываясь на данных PHP.

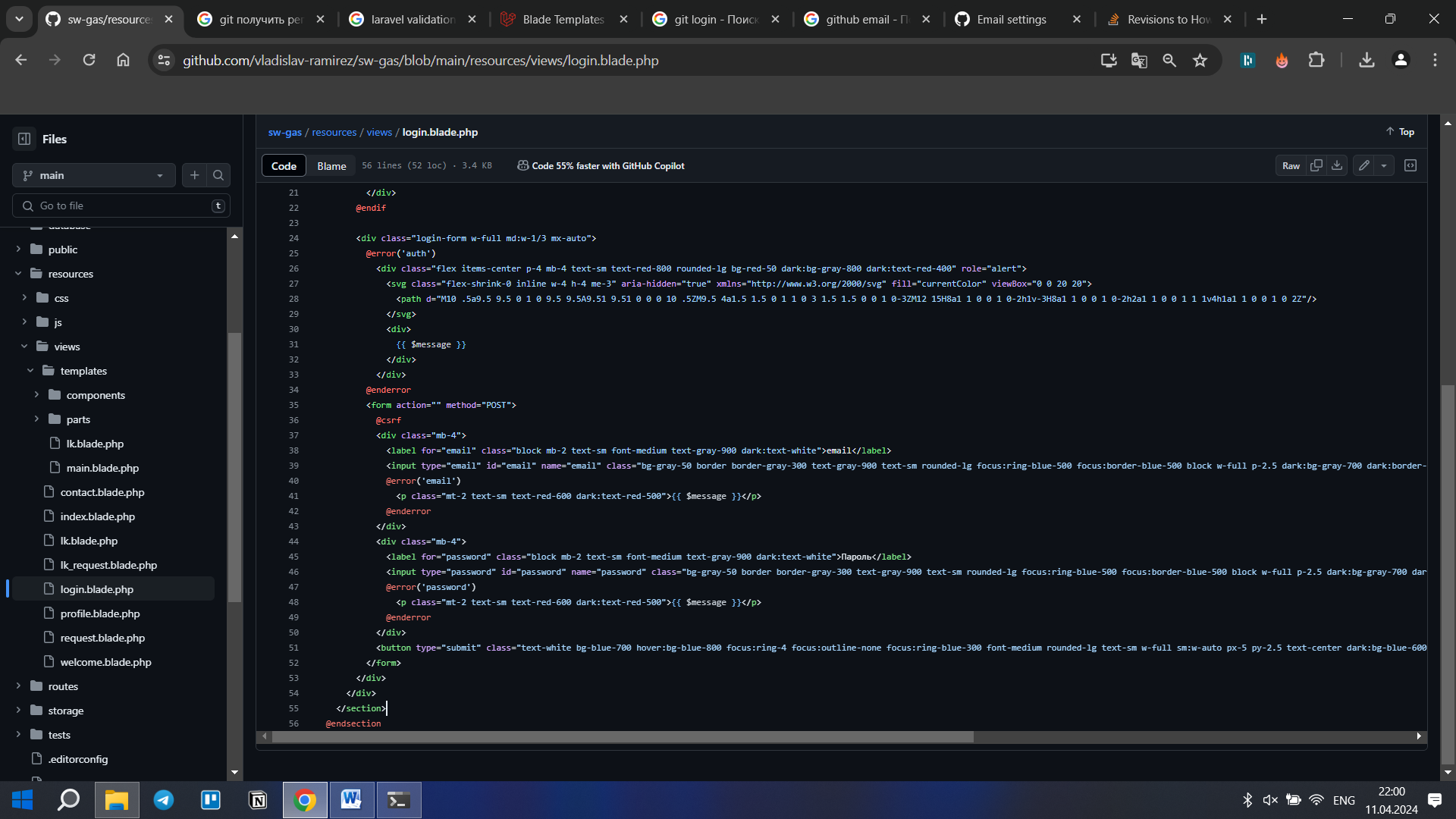




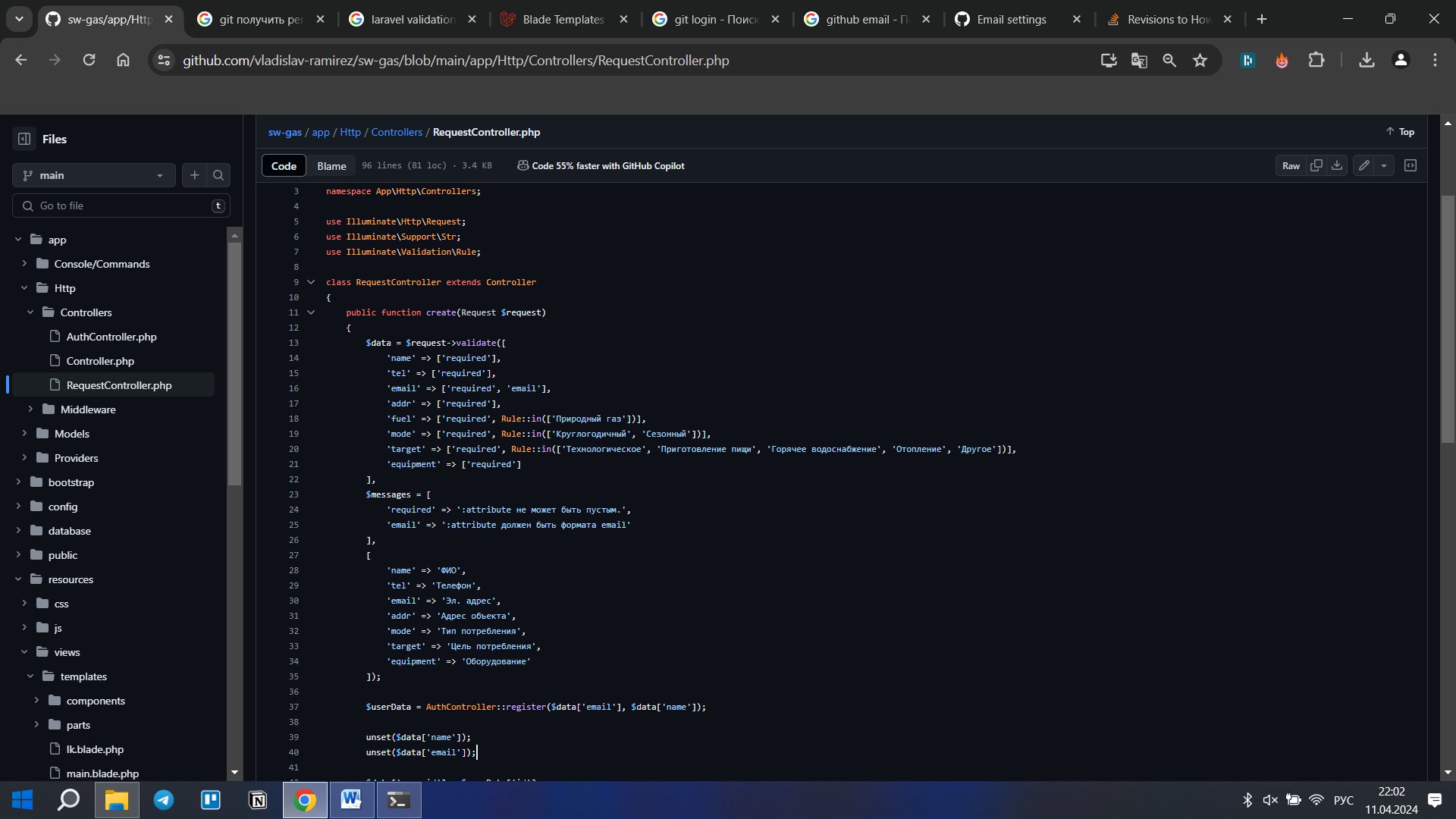
Функция view() возвращает шаблон по имени.

## 2.5 Формы

Для форм в шаблоне необходимо обязательно указывать директиву @csrf. Laravel имеет встроенную защиту от CSRF-атак, и если не будет CSRF-токена, приложение не будет обрабатывать форму, а вернет HTTP 419.



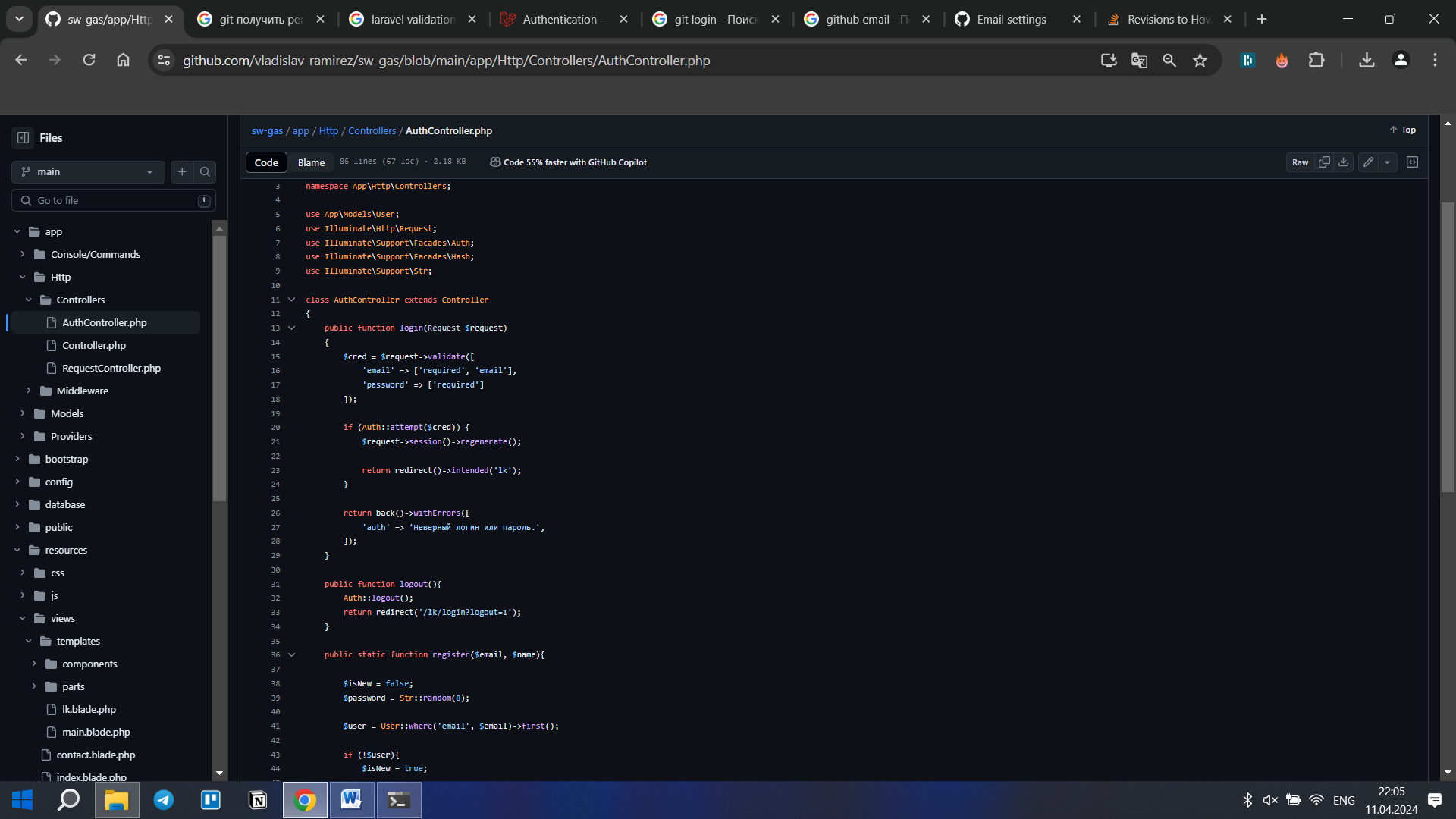
Формы обрабатываются контроллерами (app\http\controllers) и имеют механизм встроенной валидации (https://laravel.com/docs/11.x/validation#available-validation-rules).



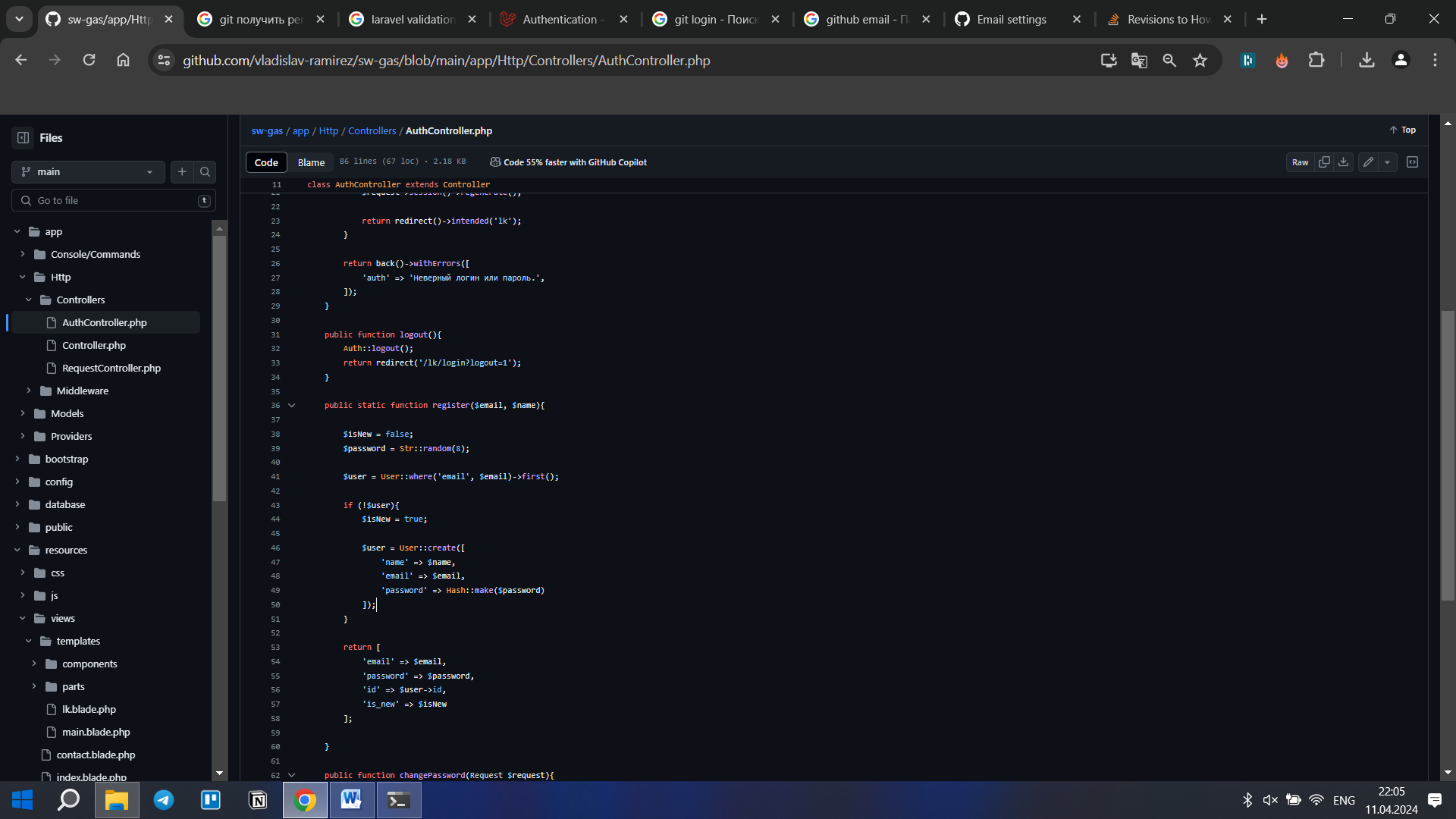
## 2.6 Авторизация

Авторизация реализована стандартным методом laravel (<https://laravel.com/docs/11.x/authentication>).

Контроллер авторизации (AuthController) имеет вид:



Регистрация пользователя:



Смена пароля:

