**3.3 Тестирование программного модуля**

Тестирование программного обеспечения является важным процессом, требующим глубокого понимания особенностей программного продукта и требований к нему. Тестирование сайта состоит из 2-х этапов: функциональное тестирование которое проверяет работы бэк-энда и «ручное» когда пользователь самостоятельно выполняет тестовые действия на сайте

**3.3.1 Ручное тестирование**

Основные пункты работы сайта которые необходимо проверить это:

1. Работоспосбность сайта в целом
2. Проверка работоспособности формы авторизации
3. Проверка создания контента
4. Проверка выхода из системы
5. Проверка рассылки

Результаты тестирования представлены в таблице 1.

|  |  |  |
| --- | --- | --- |
| Действие актера | Действие бэк-енда | Отметка о правильной работе или описание ошибки |
| Открыть главную страницу | Запуск главного модуля приложения | Действие выполнено успешно |
| Перейти на страницу авторизации | Проверить валидность данных и проверить соответствие в базе данных введенных данных | Действие выполнено успешно |
| Создать новую страницу | Получить POST данные и внести изменения в базу данных | Действие выполнено успешно |
| Добавить изображение в галерею | Получить POST данные и внести изменения в базу данных | Действие выполнено успешно |
| Добавить пользовательский имейл в список рассылки | Получить POST данные и внести изменения в базу данных | Действие выполнено успешно |
| Создать пользователя | Получить POST данные и внести изменения в базу данных | Действие выполнено успешно |
| Выбрать пункт в админ-меню «выход» | Провести очистку данных в таблице sessions | Действие выполнено успешно |
| В админ-панели запустить рассылку | В цикле перебрать все имейл-адреса из БД и отправить на них письмо | Действие выполнено успешно |

Таблица 1 – Журнал тестирования

**3.3.2 Функциональное тестирование**

Функциональное тестирование — процесс обеспечения качества в рамках [цикла разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%B4%D0%B8%D0%B8_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), необходимый для проверки реализуемости [функциональных](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) [требований,](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%BC%D1%83_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8E) согласно [спецификации](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) тестируемого [программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5). Функциональное тестирование проводится для [оценки соответствия](https://ru.wikipedia.org/wiki/%D0%9E%D1%86%D0%B5%D0%BD%D0%BA%D0%B0_%D1%81%D0%BE%D0%BE%D1%82%D0%B2%D0%B5%D1%82%D1%81%D1%82%D0%B2%D0%B8%D1%8F) [системы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) или [компонента](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) заданным функциональным требованиям.

Функциональное тестирование проводится по принципу [черного ящика](https://ru.wikipedia.org/wiki/%D0%A7%D1%91%D1%80%D0%BD%D1%8B%D0%B9_%D1%8F%D1%89%D0%B8%D0%BA), в связи с чем функциональность ПО можно протестировать, не зная принципа его внутренней работы. Это снижает требования к [тестировщикам](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D1%89%D0%B8%D0%BA) в части знания [языков программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) или конкретных аспектов реализации [программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5).

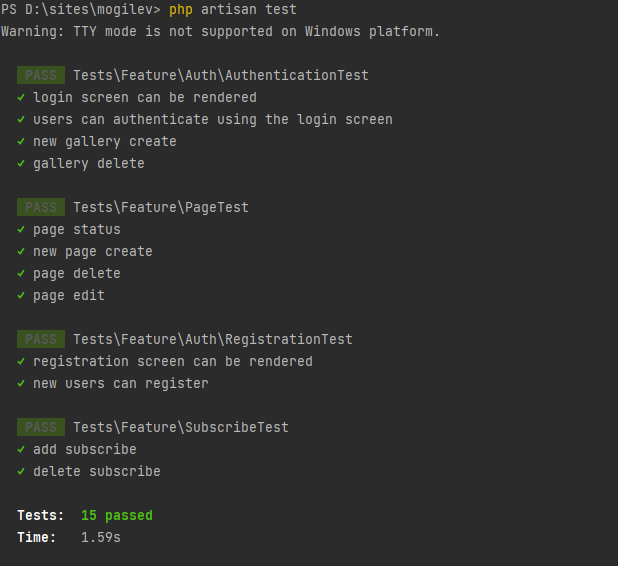
В Laravel предусмотрены 3 типа тестов: unit, feature и browser тесты. Для нашей задачи были выбраны Feature тесты - это тесты, которые фокусируются на очень небольшой изолированной части нашего кода. Тесты должны покрывать 4 сущности нашего сайта:

1. **Страницы**
   1. Просмотр страниц сайта
   2. Создание страницы сайта
   3. Редактирование существующей страницы
   4. Удаление страницы
2. **Изображения в галереи**
   1. Просмотр галереи
   2. Удаление изображения
   3. Добавление нового изображения
3. **Пользователи**
   1. Создание нового пользователя
   2. Редактирование существующего пользователя
   3. Удаление пользователя
   4. Авторизация пользователя
4. **Подписки на рассылку**
   1. Добавление нового имейла в список
   2. Создание рассылки
   3. Удаление имейла из списка

Все тесты разбиты на 6 файлов кода, в которых находятся 15 методов для проверки работоспособности нашего проекта: отправка запроса с данными и получение ответа, если получен ответ от сервера «200» значит тест пройден успешно, либо правильным ответам может быть статус «301» - редирект на нужную страницу после выполнения действий

Так же для корректной работы тестирование мы создаем отдельный файл переменных окружения «.env\_testing» и отдельную базу данных, чтобы тестирование во время проверок не поломало основную базу данных и все данные остались в целости.

Результаты тестирования отображены на рисунке:



Все 15 тестов пройдены успешно, за время 1.59с. Время тестирования может варьироваться в зависимости от конфигурации серверного оборудования. Для данных результатов тестов использовалась следующая конфигурация оборудование:

CPU: Intel Xeon E5-2699 v3

Memory: 64Gb DDR4 ECC, quad channel mod

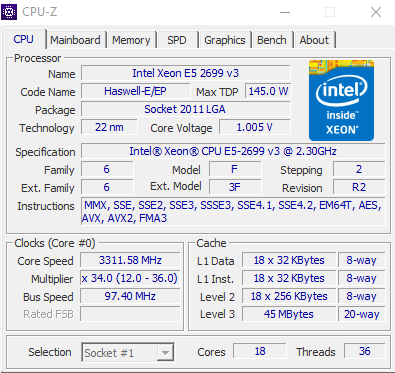
SSD m2 512Gb

Motherboard: Attermiter x99

GPU: NVidia RTX 2060 Super 8Gb  
  
Server: Apache 2.4

PHP: 8.1

ОС: Windows 10



Весь код тестовых модуль представлен в Приложении В