Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

Лабораторная работа №5

по дисциплине "Основы разработки веб-приложений"

ТЕМА РАБОТЫ:

Laravel PHP

Выполнил:

студент гр. ПРИ-120

Парахин К.В.

Принял:

Спирин И.В.

Владимир 2023 г.

Цель работы:

Изучить основы языка PHP. Познакомиться с принципами работы с фреймворком Laravel для разработки MVC веб-проектов на языке PHP.

Ход работы:

1. Воспроизвести шаги из методички и показать работоспособность
2. Реализовать 2 CRUD операции с продутами на выбор
3. Реализовать логику работы корзины покупателя с использованием объекта session (при перезагрузке страницы данные не теряются)

Выполнение работы:

1. Создадим проект Laravel в VS Code:

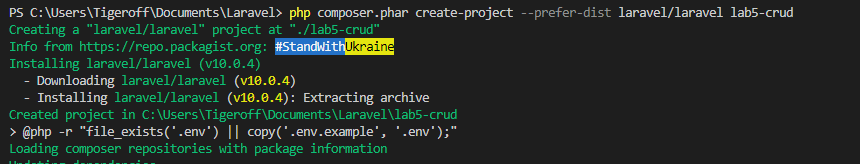


Рисунок 1. Создание проекта laravel

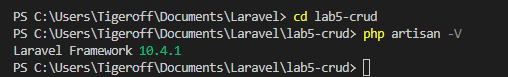


Рисунок 2. Правильность конфигурации

Попробуем запустить встроенный сервер при помощи ввода команды **php artisan serve** в консоль:

1. Далее настроим переменные в разделе .env для доступа миграций к нашей базе данных провайдера MySQL

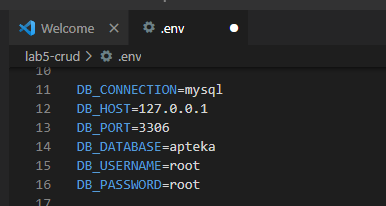


Рисунок 3. Настройка переменных доступа к БД

1. Создадим новую модель под названием Product с помощью команды «php artisan make:model Product –m».

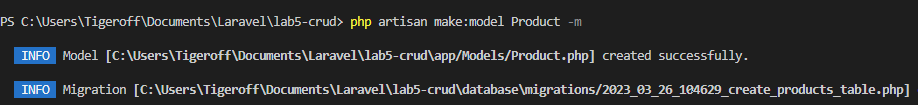


Рисунок 4. Создание новой модели и начальной миграции

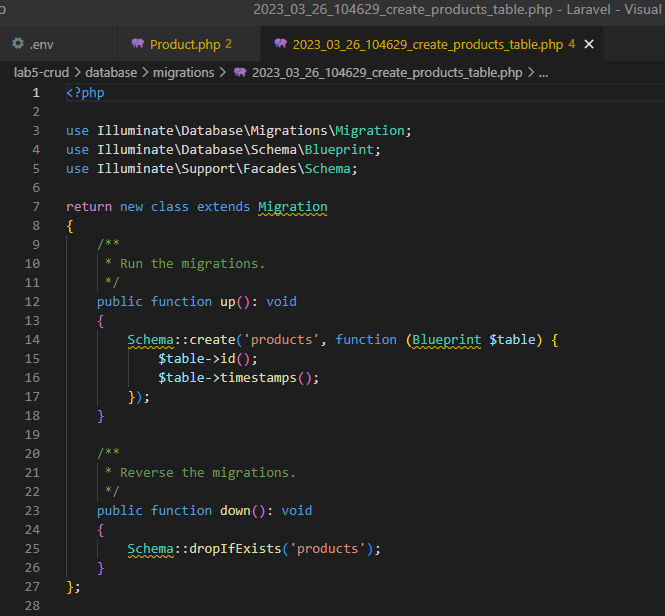


Рисунок 5. Полученная миграция

1. Далее изменяем созданную миграцию, добавляем в нее новые поля для модели Product и запускаем скрипт обновления базы данных

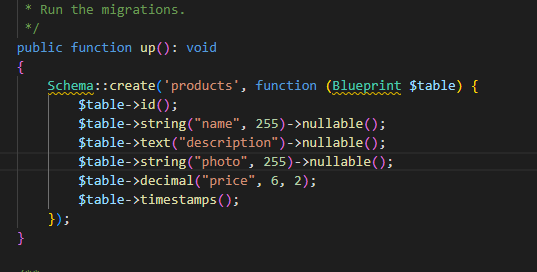


Рисунок 6. Добавление новых полей

Далее с помощью команды «php artisan migrate» запускаем новую миграцию:

Убираем преждевременно ; в php.ini у следующей записи:

;extension=pdo\_mysql.dll

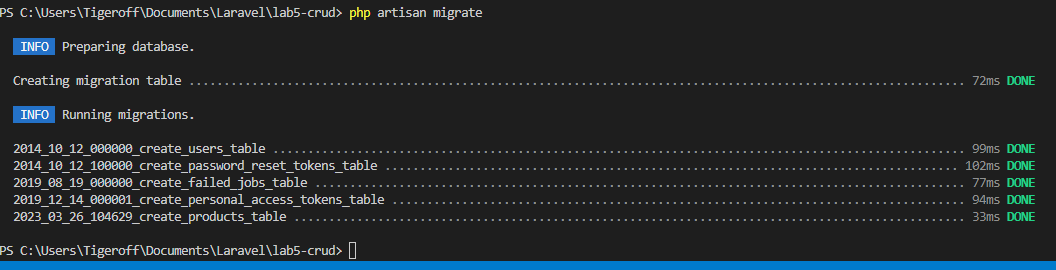


Рисунок 7. Запуск миграций

Кроме таблицы Product также создаются еще и таблицы с пользователями и их личными данными (по умолчанию сгенерированные миграции)

1. Далее необходимо заполнить таблицу Products некоторыми изначальными данными. Для этого с помощбю команды «php artisan make:seed ProductsSeeder» создаем новый Seeder, который затем вручную заполняем некоторыми данными

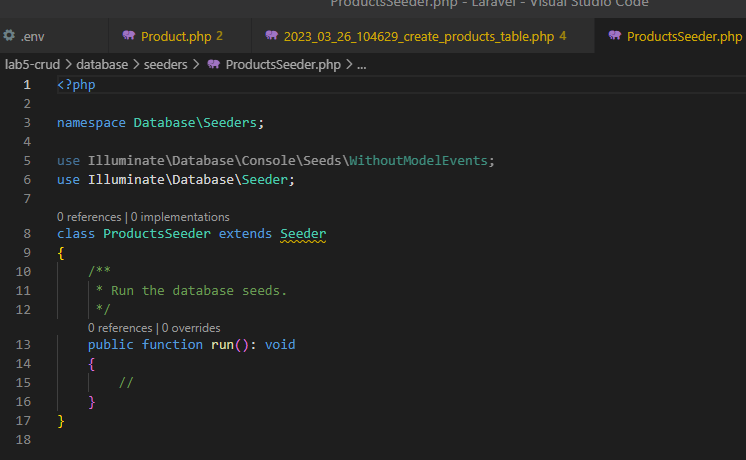


Рисунок 8. Создание пустого ProductsSeeder

Добавление одной записи по умолчанию:

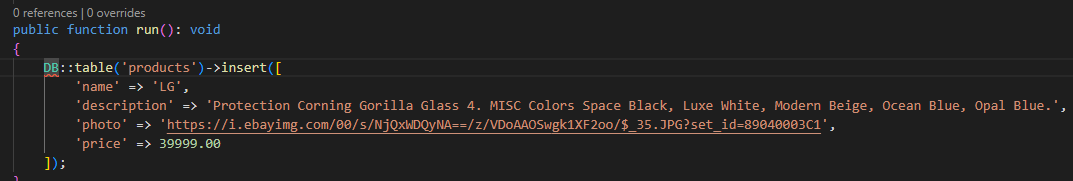


Рисунок 9. Добавление записи

Затем самостоятельно добавляю еще 5 записей в таблицу Products:

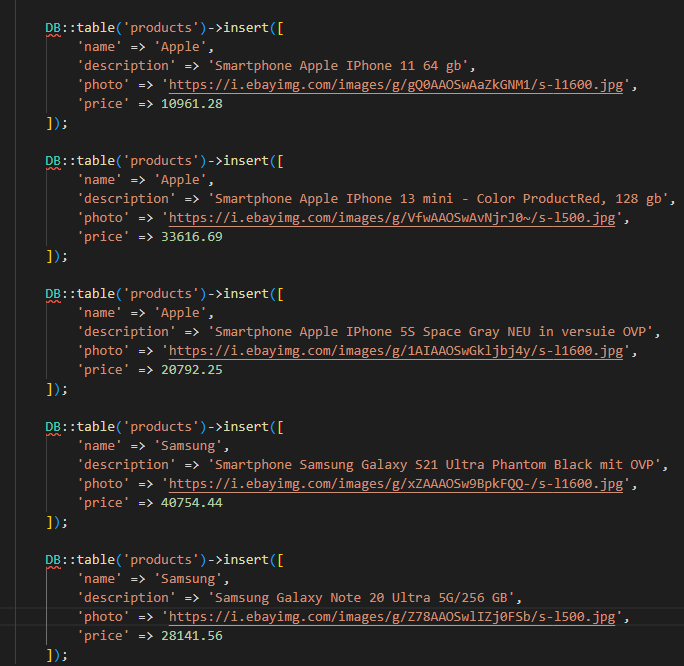


Рисунок 10. Добавление 5 новых изначальных записей

1. С помощью двух команд выполним создание записей в таблице Products нашей базы данных

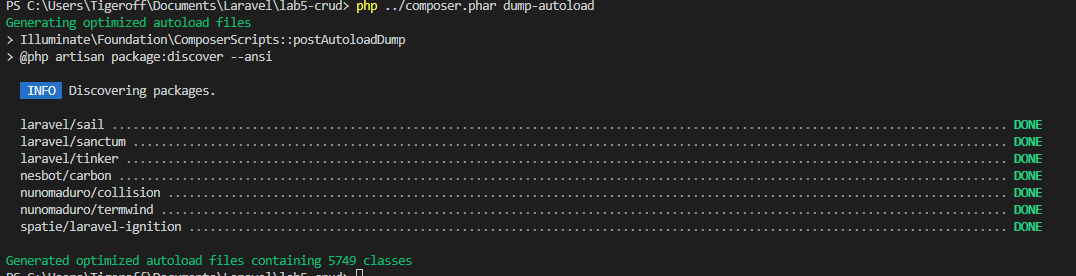


Рисунок 11. Выполнение команды «php ../composer.phar dump-autoload»



Рисунок 12. Добавление ссылки на класс DB в config/app.php

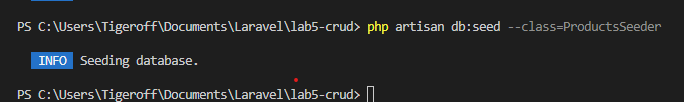


Рисунок 13. Выполнение наполнения таблицы данными

1. Далее выполняем команду по созданию нового контроллера ProductsController

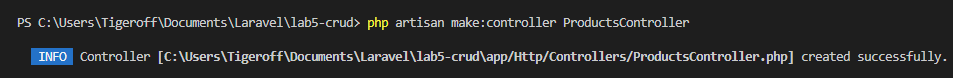


Рисунок 14. Выполнение команды php artisan make:controller ProductsController

Затем добавляем в этот контроллер метод index(), который будет возвращать представление «products» со всеми продуктами:



Рисунок 15. Создание пустого ProductsController

1. В папке resources/views создаем новый файл layout.blade.php со следующим содержимым:

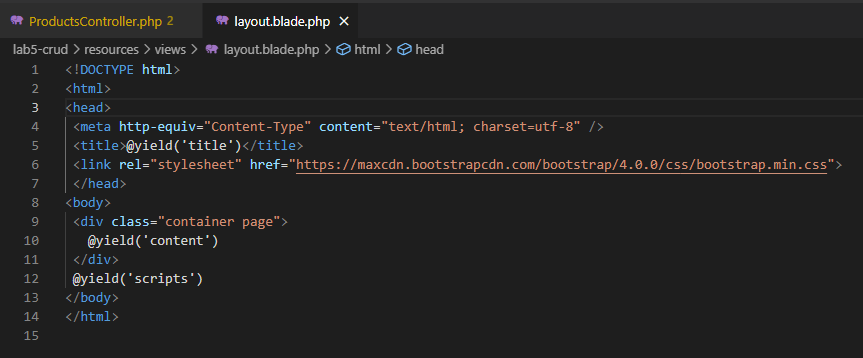


Рисунок 16. Создание файла представления для layout

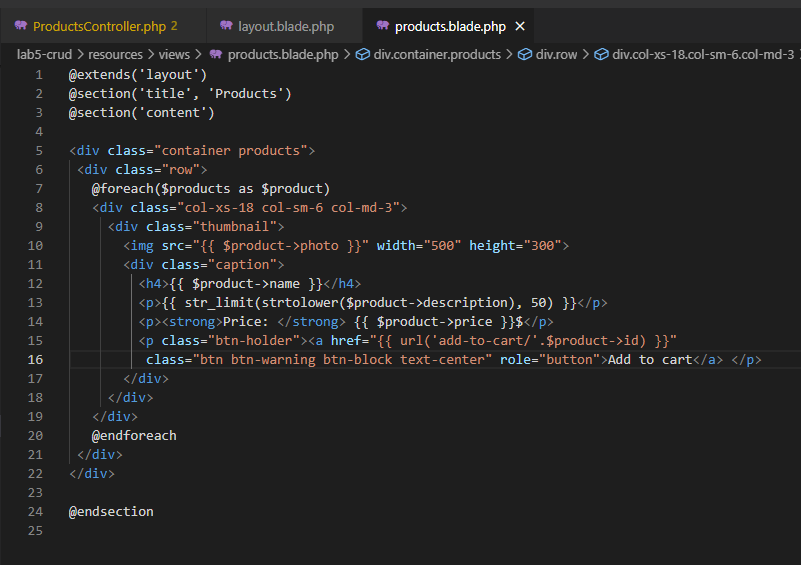


Рисунок 17. Создания файла представления для products

1. Меняем в файле web.php путь route, задавая начальной страницей страницу продуктов, то есть то, что возвращает метод Index контроллера ProductsController:

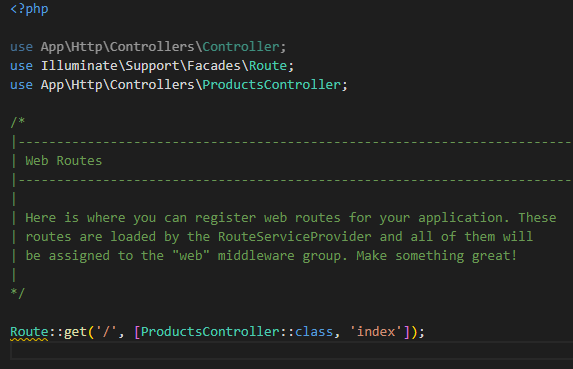


Рисунок 18. Изменение начальной страницы

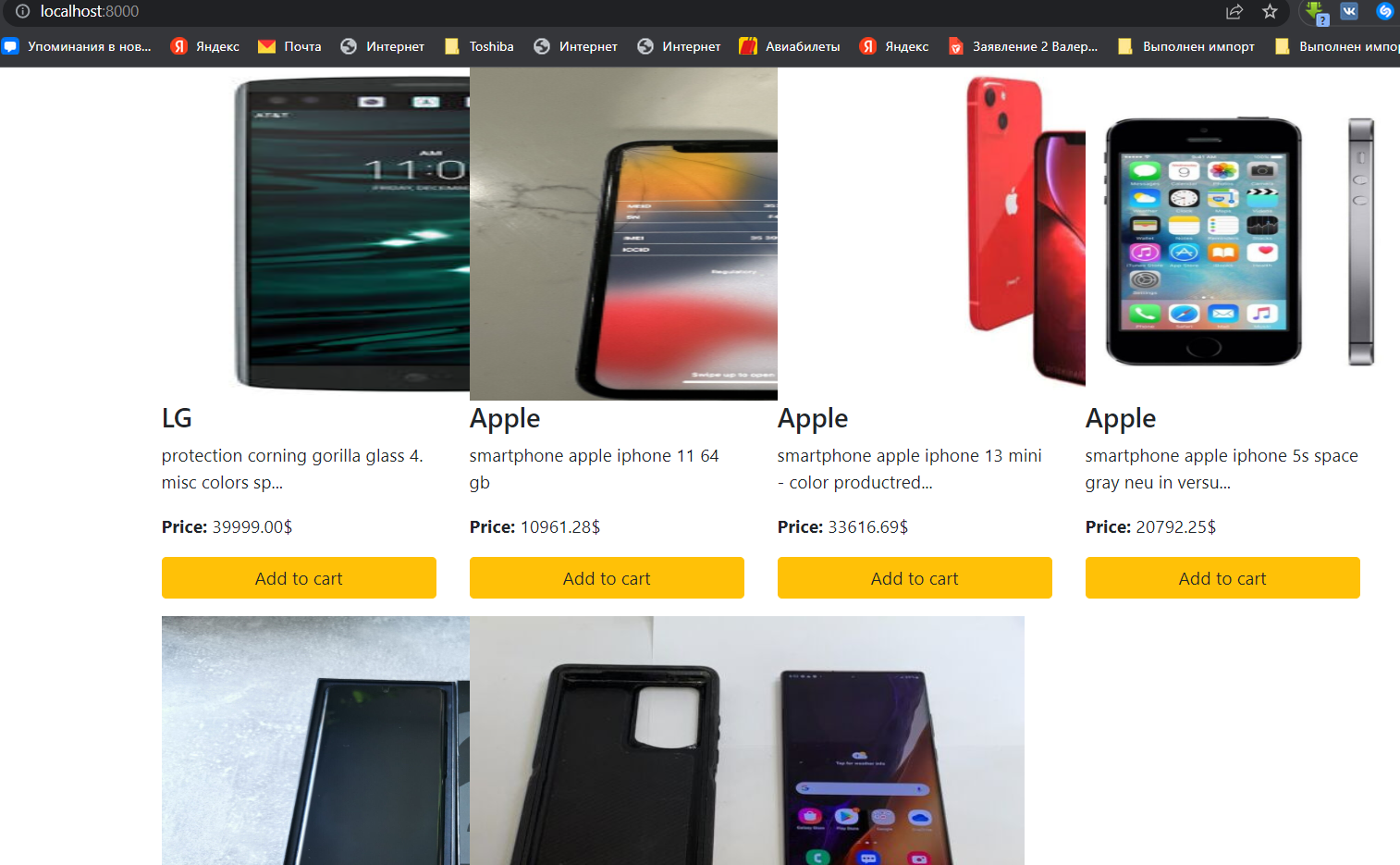


Рисунок 19. Начальная страница index с продуктами

1. Далее добавляем 2 CRUD операции в ProductsController



Рисунок 20. Методы контроллера для операции «Create»

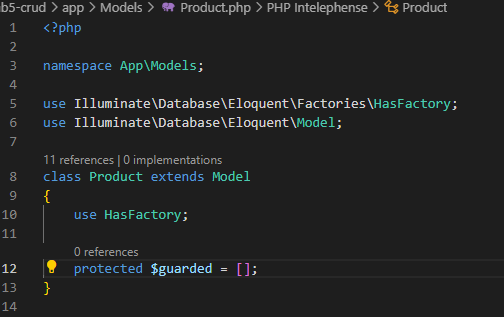


Рисунок 21. Добавления поля в модель Product (для корректной вставки нового объекта в БД – требуется поле $quarded или $fillable с некоторым ключом)

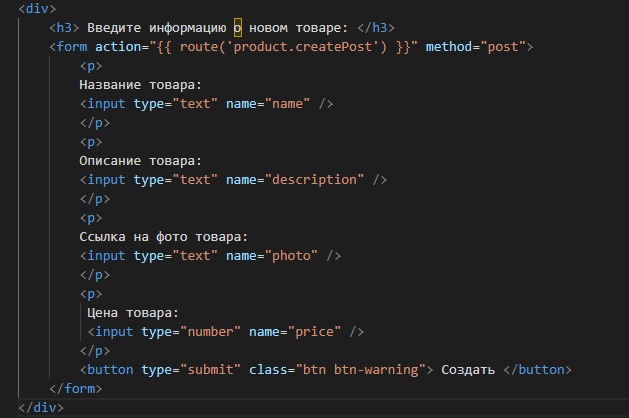


Рисунок 22. Создание файла представления для формы «Create»

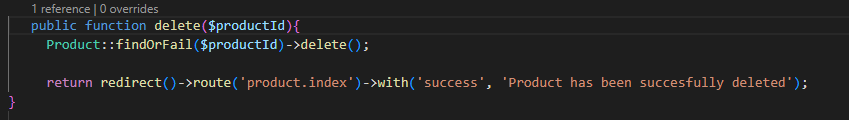


Рисунок 23. Метод контроллера для операции «Delete»

1. Далее добавим на основную страницу функции создания нового товара, открытие корзины, а также для каждого товара из списка сделаем функции: добавить в корзину и удалить.

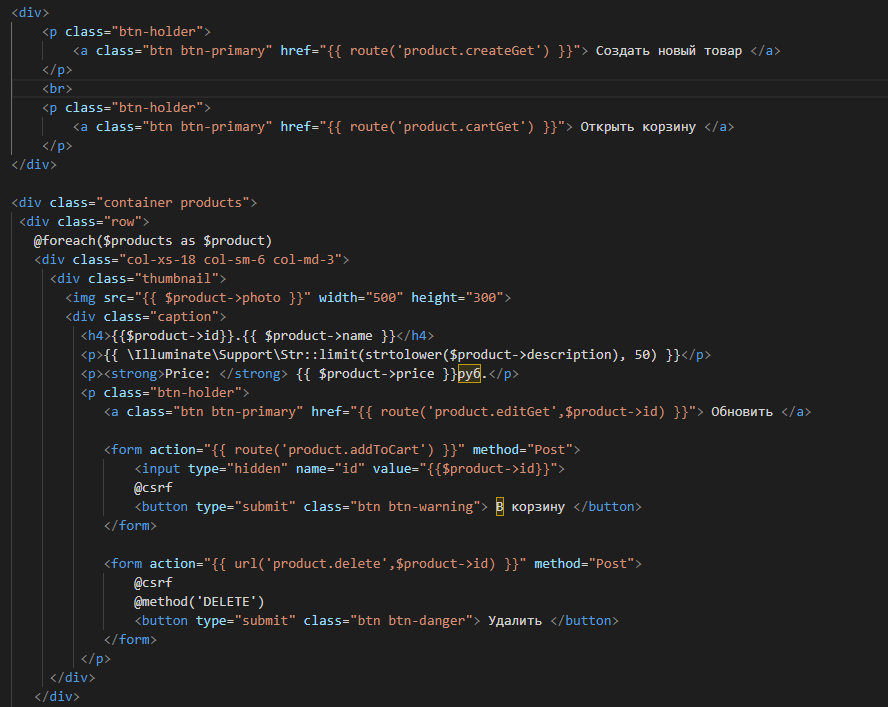


Рисунок 24. Дополнение основного представления products новыми функциями и блоками

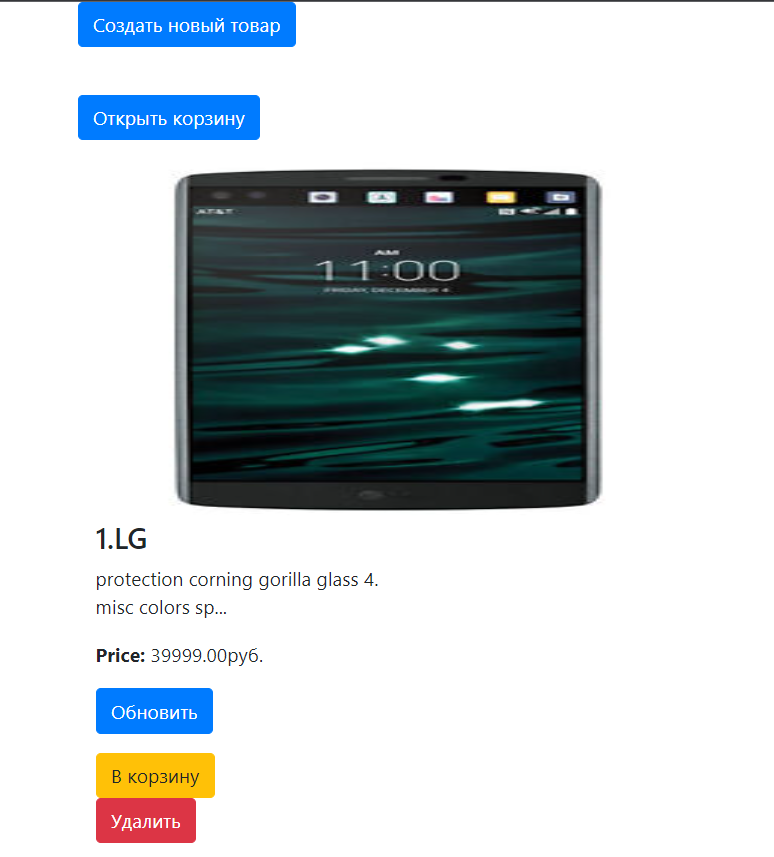


Рисунок 25. Скриншот обновленной index - страницы

С помощью миграции было добавлено ровно 6 товаров, попробуем добавить новый седьмой товар через форму, а затем удалить его:

Протестируем добавление нового товара на сайте:

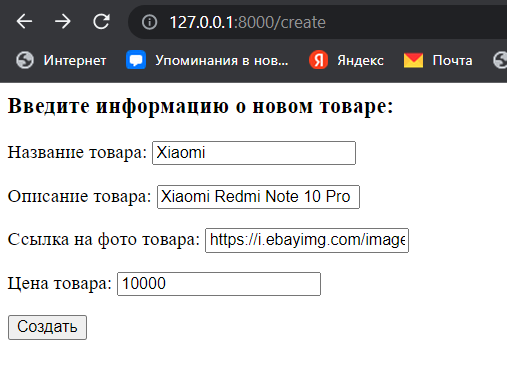


Рисунок 26. Форма для создания нового товара

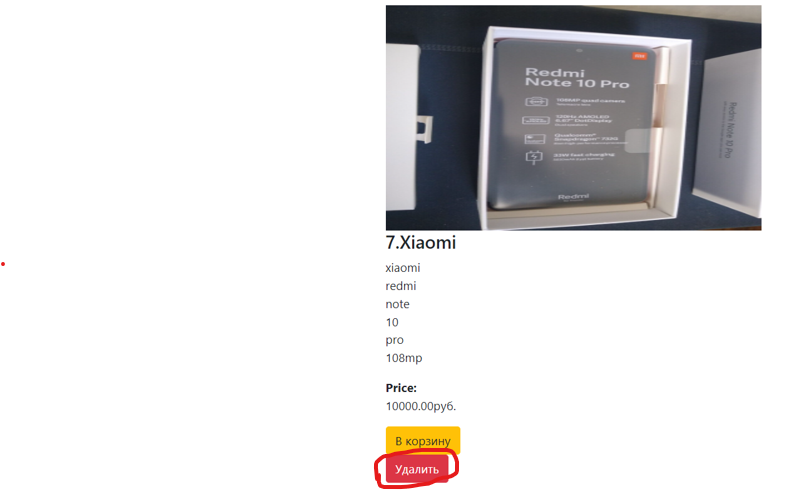


Рисунок 27. Добавленный товар в каталоге

Протестируем удаление товара на сайте:

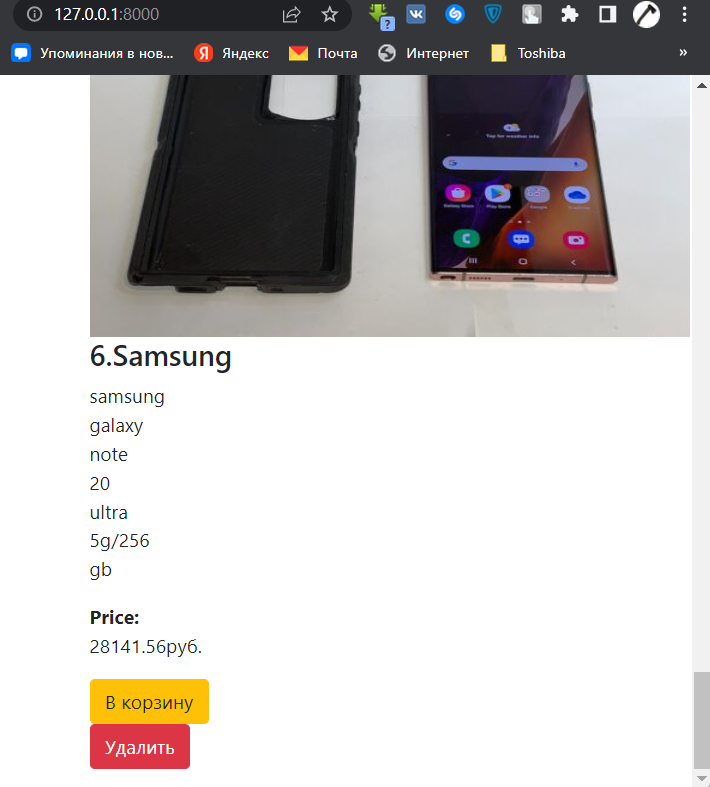
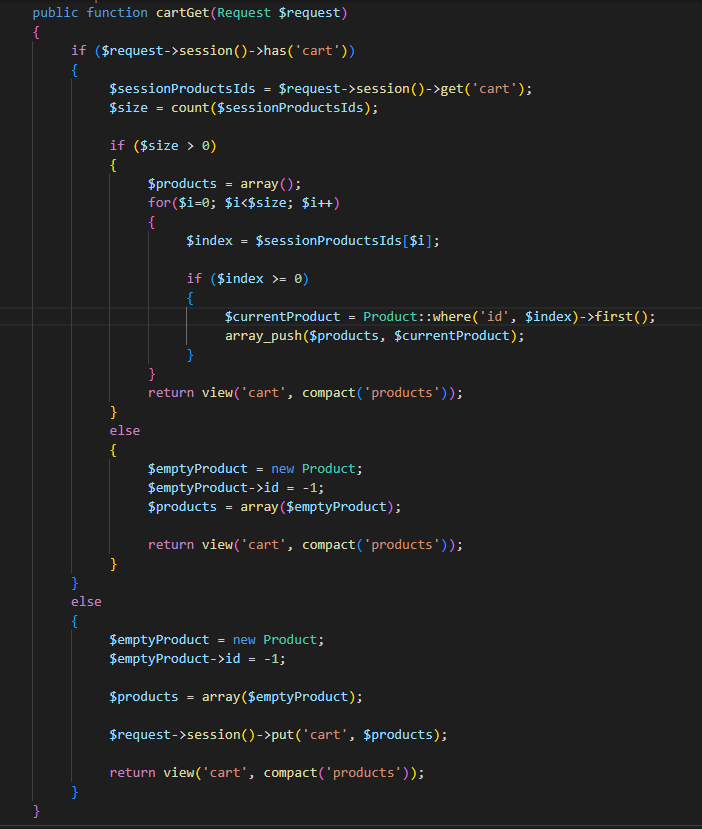
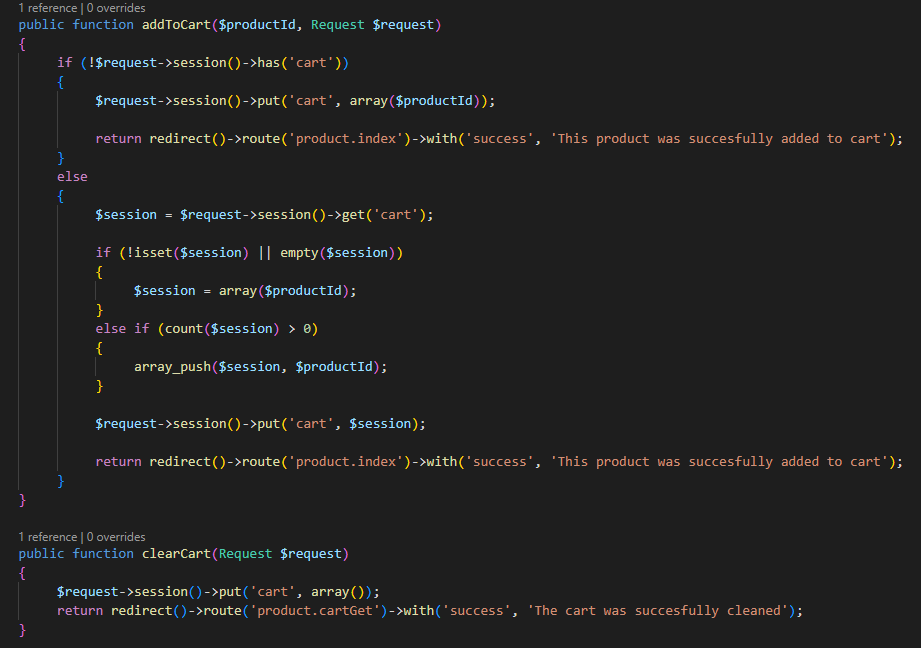


Рисунок 28. Результат удаления товара из каталога

1. Далее добавим функционал приложения «Корзина пользователя» на сайт. При добавлении пользователем товара в корзину, он будет добавлен в текущую пользовательскую сессию в браузере. И при открытии корзины будут отображаться все те товары, которые были добавлены пользователем в сессию.





Рисунки 29.1 – 29.2. Добавление методов работы с корзиной

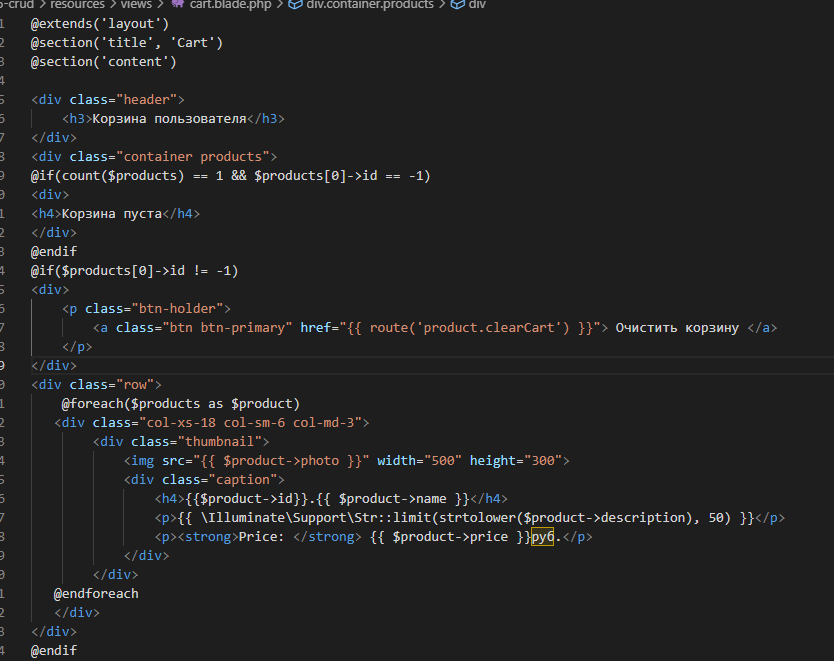


Рисунок 30. Создания файла представления для корзины (cart)

Теперь получается мы можем добавлять товары в корзину (добавляя их в раздел «cart» сессии в нашем браузере. Таким образом, сессия сохраняется и доступна при перезагрузке страницы и даже при перезапуске веб-приложения.

При нажатии на кнопку «Очистить корзину» происходит полное очищение только раздела «cart» у сессии в браузере.

Затем протестируем добавление товаров в корзину пользователя на сайте:

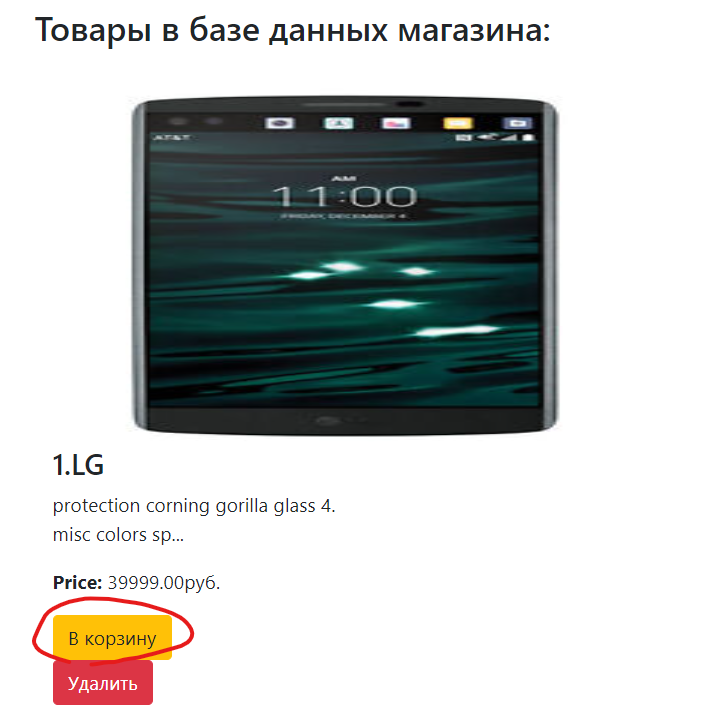


Рисунок 31. Пример добавления товара в корзину

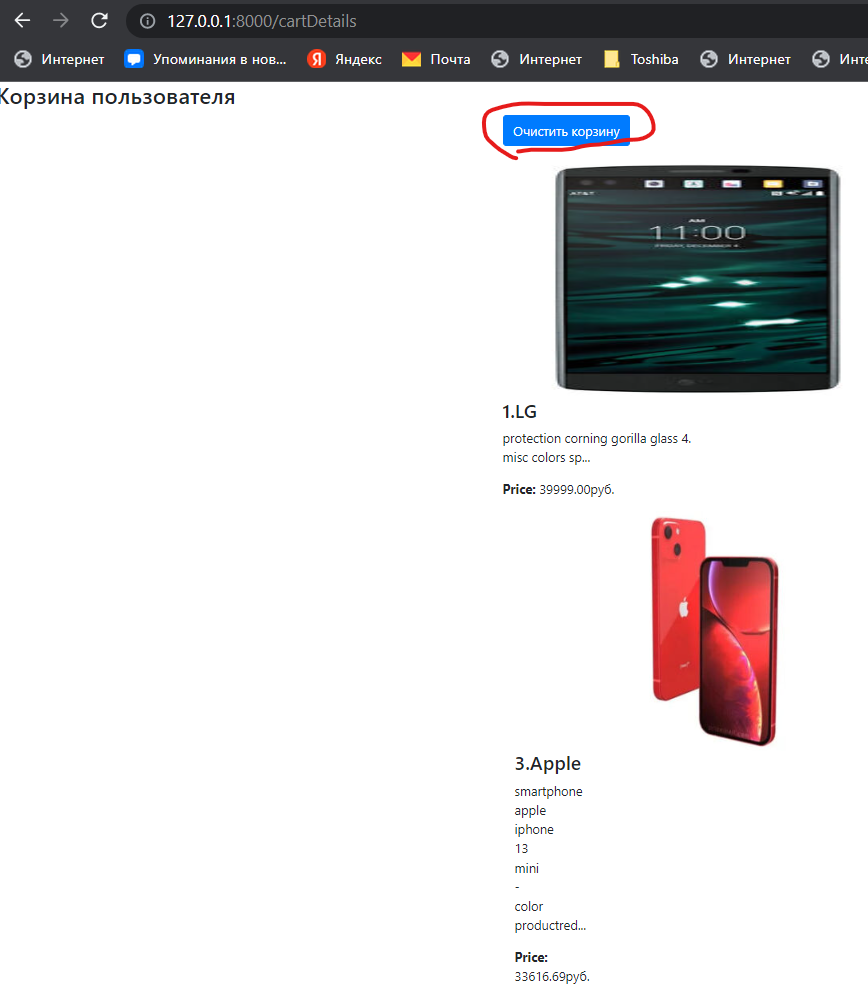


Рисунок 33. Отображение добавленных в сессию товаров в корзине

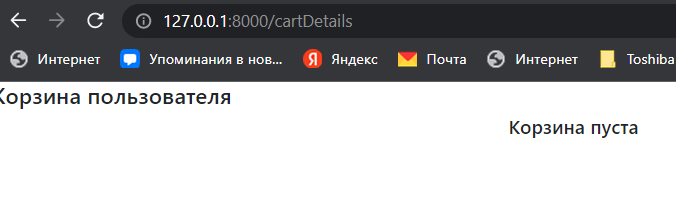


Рисунок 34. Результат очищения сессии пользователя

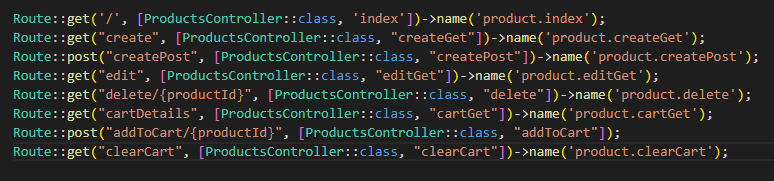


Рисунок 35. Конфигурирование путей к методам контроллеров в файле web.php

Вывод

В результате выполнения работы, я изучил основы языка PHP. Кроме того, познакомился с принципами работы с фреймворком Laravel для разработки MVC веб-проектов на языке PHP.