Министерство науки и высшего образования 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**

Рейтинг-контроль №6

по дисциплине "Основы разработки веб-приложений"

по теме: **«Использование Docker для запуска приложений на Laravel»**

Выполнил:

студент гр. ПРИ-117

Журавлев Н.М.

Приняла:

Козаев А.Т.

Владимир 2020 г.

ЦЕЛЬ РАБОТЫ

Познакомиться со средством контейнеризации Docker и использовать его для запуска приложения на Laravel.

ЗАДАНИЯ

1. Воспроизвести шаги из методички и показать работоспособность.
2. Добавить сервис БД и показать работоспособность взаимодействия приложения с БД запущенной в Docker.
3. Подключить сервис для обработки кэширования. Включить

ВЫПОЛНЕНИЕ ЗАДАНИЙ

Воспроизвести шаги из методички и показать работоспособность.

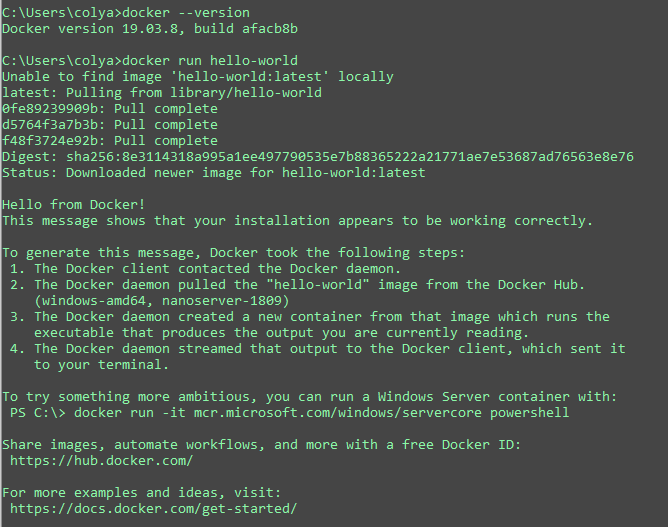


Рисунок 1. Успешная установка docker вместе с compose

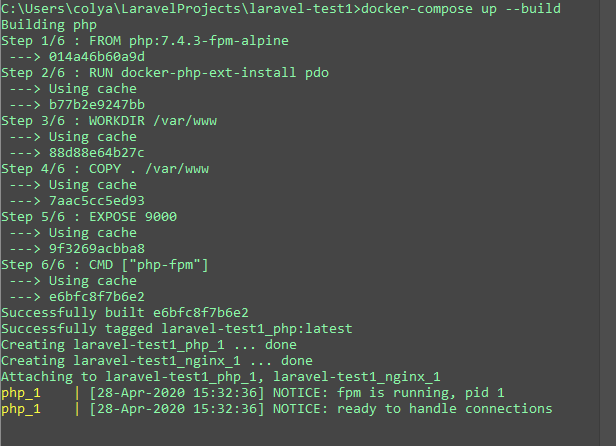


Рисунок 2. Сборка контейнеров после изменения кода согласно методическим указаниям

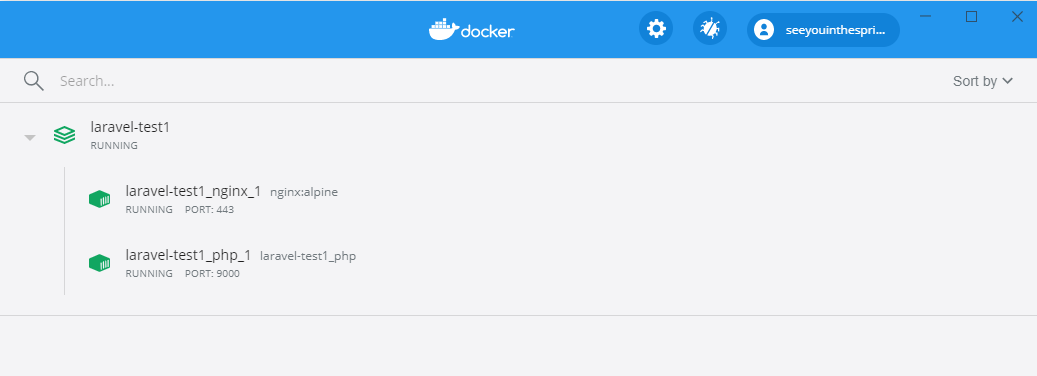


Рисунок 3. Запущенные контейнеры

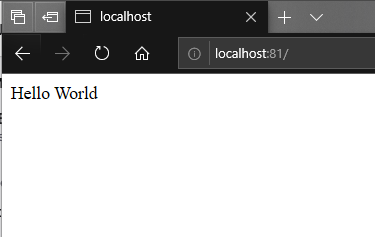


Рисунок 4. Вывод тестовой страницы var/www/index.php

Добавить сервис БД и показать работоспособность взаимодействия приложения с БД запущенной в Docker.

Код файла docker-compose.yml был изменен на следующий:

version: "3.2"

services:

nginx:

image: nginx:alpine

volumes:

- .:/var/www

- ./nginx/conf.d/:/etc/nginx/conf.d/

ports:

- 81:81

- 443:443

depends\_on:

- php

php:

build:

context: .

volumes:

- .:/var/www

ports:

- 9000:9000

depends\_on:

- db

db:

image: mysql:latest

environment:

MYSQL\_DATABASE: new\_db

MYSQL\_ROOT\_PASSWORD: root

ports:

- 3307:3306

Параметры подключения к БД в файле .env были изменены на следующие:

DB\_CONNECTION=mysql

DB\_HOST=db

DB\_PORT=3306

DB\_DATABASE=products

DB\_USERNAME=root

DB\_PASSWORD=

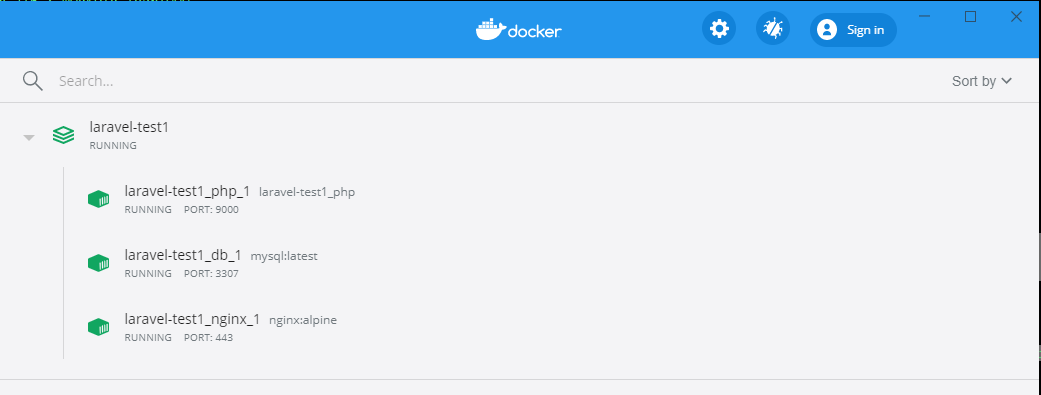


Рисунок 5. Результат сборки и запуска контейнеров

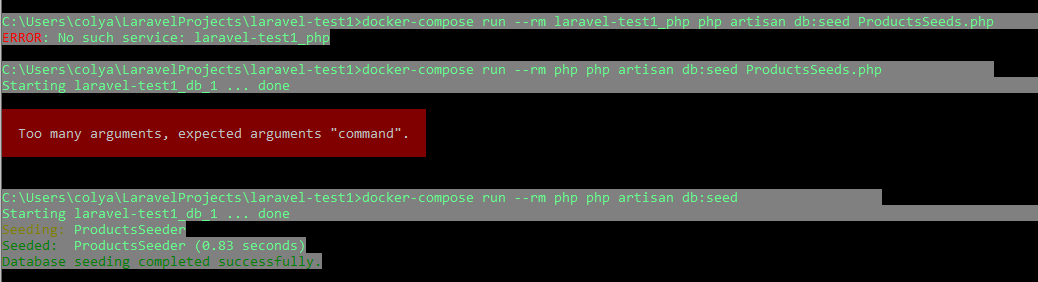


Рисунок 6. Заполнение базы данных в контейнере

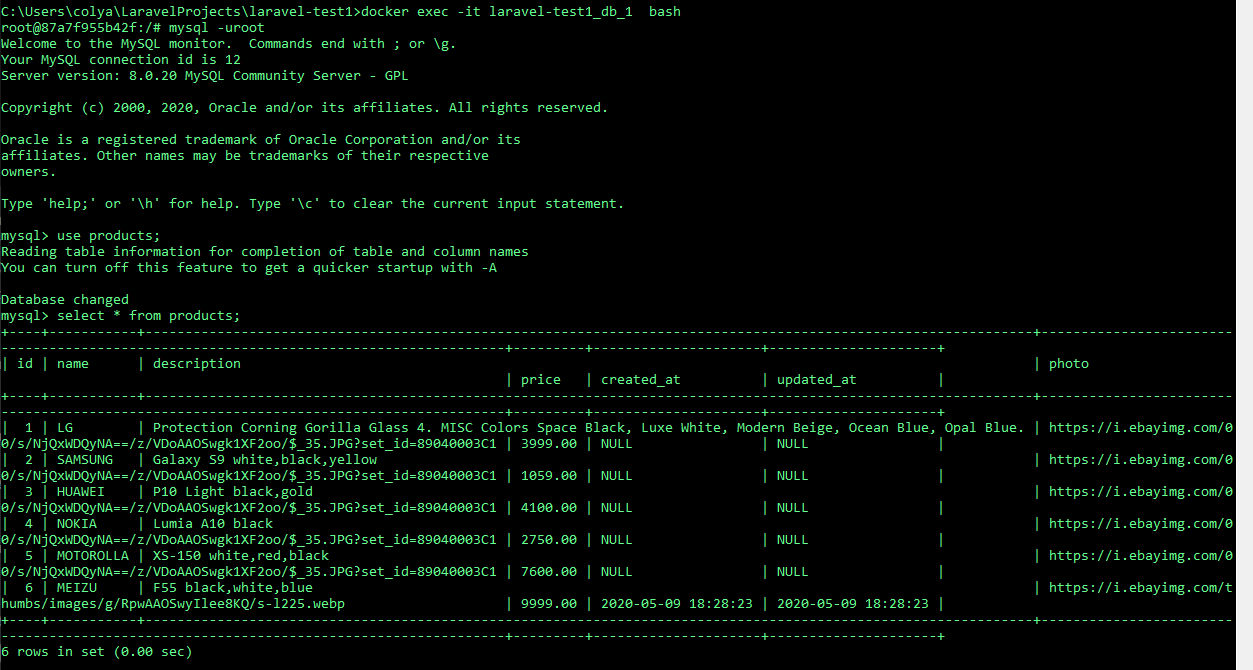


Рисунок 7. Отображение БД в контейнере

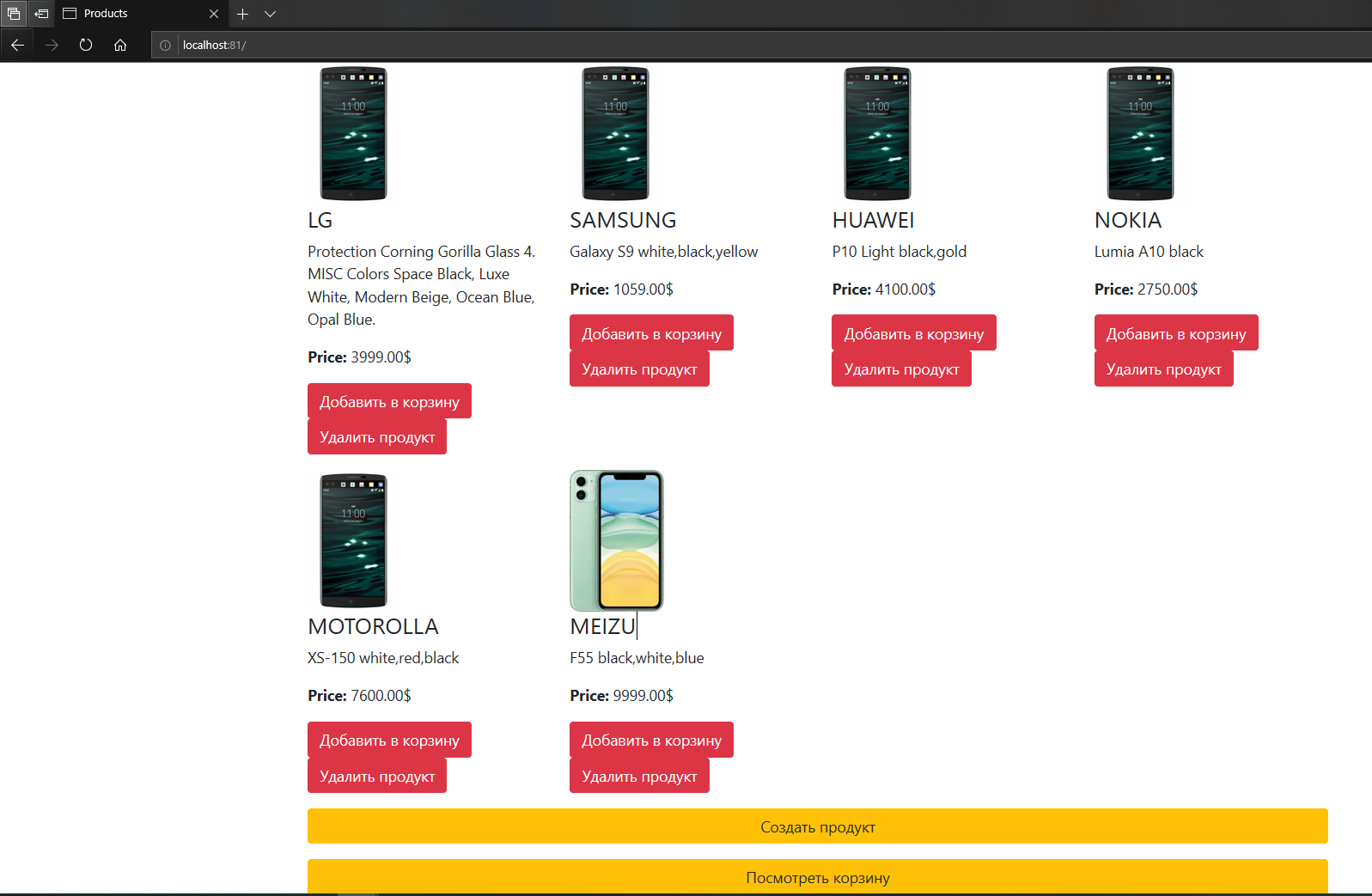


Рисунок 8. Результат работы приложения

Подключить сервис для обработки кэширования. Включить кэширование в Laravel. Показать работоспособность.

Код файла docker-compose.yml был изменен на следующий:

version: "3.2"

services:

nginx:

image: nginx:alpine

volumes:

- .:/var/www

- ./nginx/conf.d/:/etc/nginx/conf.d/

ports:

- 81:81

- 443:443

depends\_on:

- php

php:

build:

context: .

volumes:

- .:/var/www

ports:

- 9000:9000

depends\_on:

- db

- redis

redis:

image: redis:4.0

ports:

- 6379:6379

db:

image: mysql:latest

environment:

MYSQL\_DATABASE: new\_db

MYSQL\_ROOT\_PASSWORD: root

ports:

- 3307:3306

Параметры подключения к сервису кеширования в файле .env были изменены на следующие:

BROADCAST\_DRIVER=log

CACHE\_DRIVER=redis

QUEUE\_CONNECTION=sync

SESSION\_DRIVER=file

SESSION\_LIFETIME=120

REDIS\_HOST=redis

REDIS\_PASSWORD=null

REDIS\_PORT=6379

Параметры redis в файле database.php были изменены на следующие:

'redis' => [

'client' => env('REDIS\_CLIENT', 'predis'),

'options' => [

'cluster' => env('REDIS\_CLUSTER', 'redis'),

'prefix' => env('REDIS\_PREFIX', Str::slug(env('APP\_NAME', 'laravel'), '\_').'\_database\_'),

],

'default' => [

'url' => env('REDIS\_URL'),

'host' => env('REDIS\_HOST', 'redis'),

'password' => env('REDIS\_PASSWORD', null),

'port' => env('REDIS\_PORT', '6379'),

'database' => env('REDIS\_DB', '0'),

],

'cache' => [

'url' => env('REDIS\_URL'),

'host' => env('REDIS\_HOST', '127.0.0.1'),

'password' => env('REDIS\_PASSWORD', null),

'port' => env('REDIS\_PORT', '6379'),

'database' => env('REDIS\_CACHE\_DB', '1'),

],

],

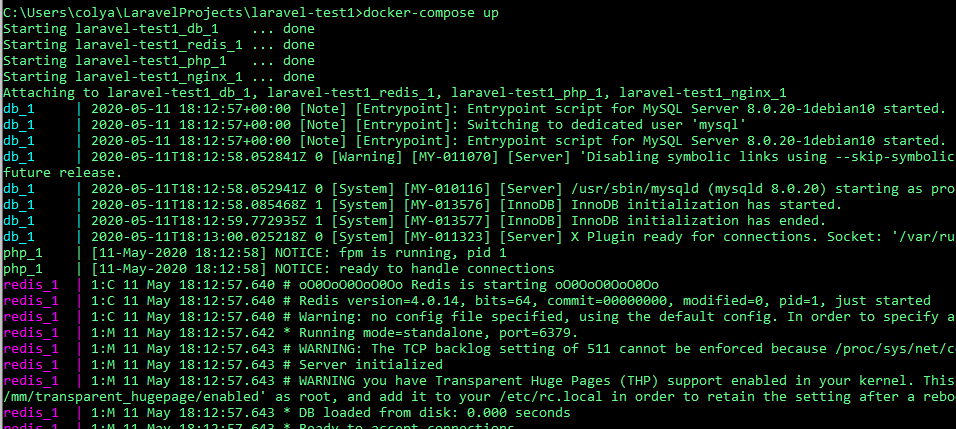


Рисунок 9. Создание и запуск контейнеров

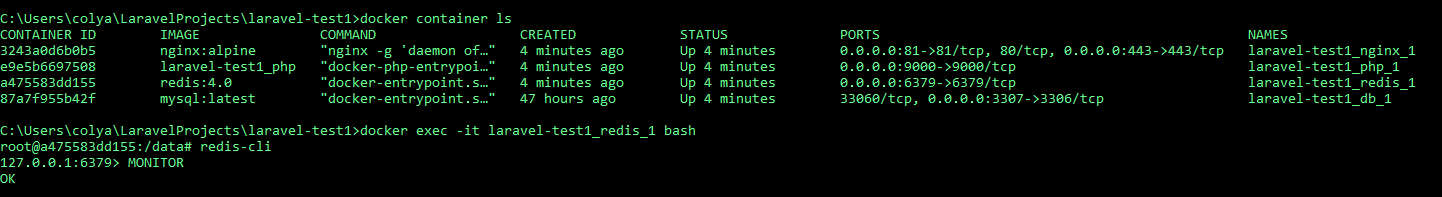


Рисунок 10. Список запущенных контейнеров и обращение к контейнеру redis

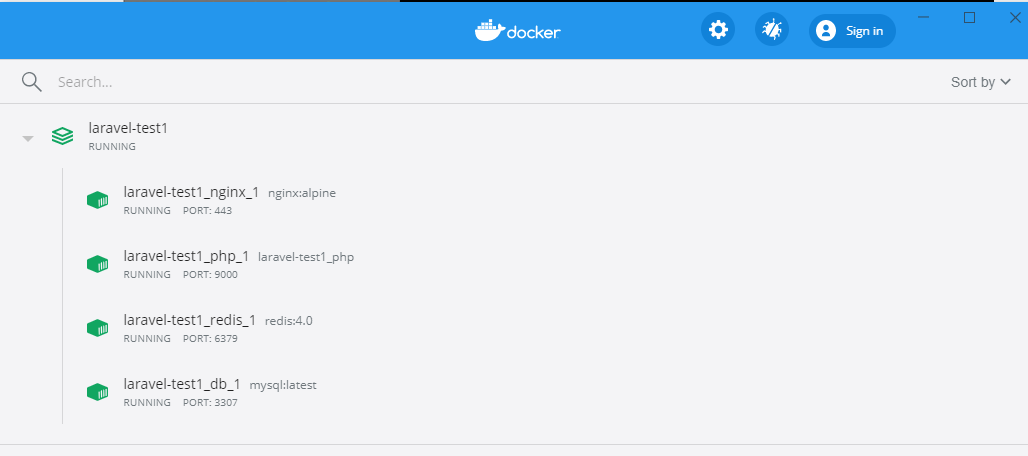


Рисунок 11. Список запущенных контейнеров в docker

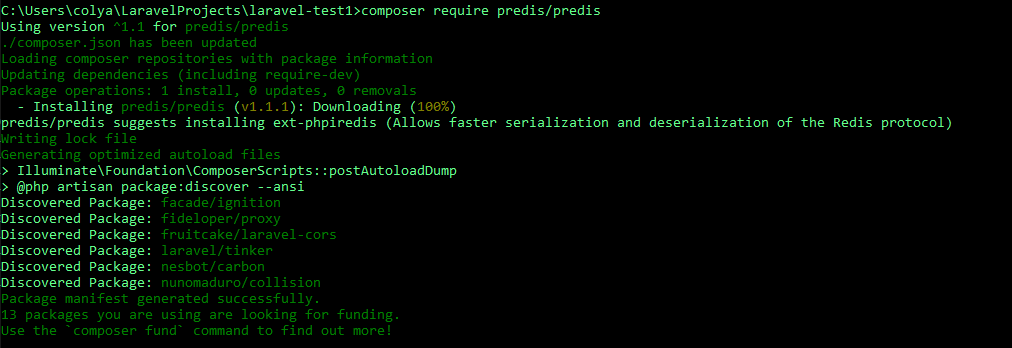


Рисунок 12. Подключение библиотеки predis

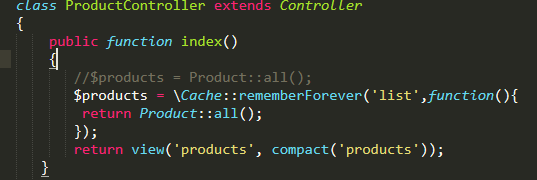


Рисунок 13. Изменение контроллера для вывода всех записей с помощью кеширования

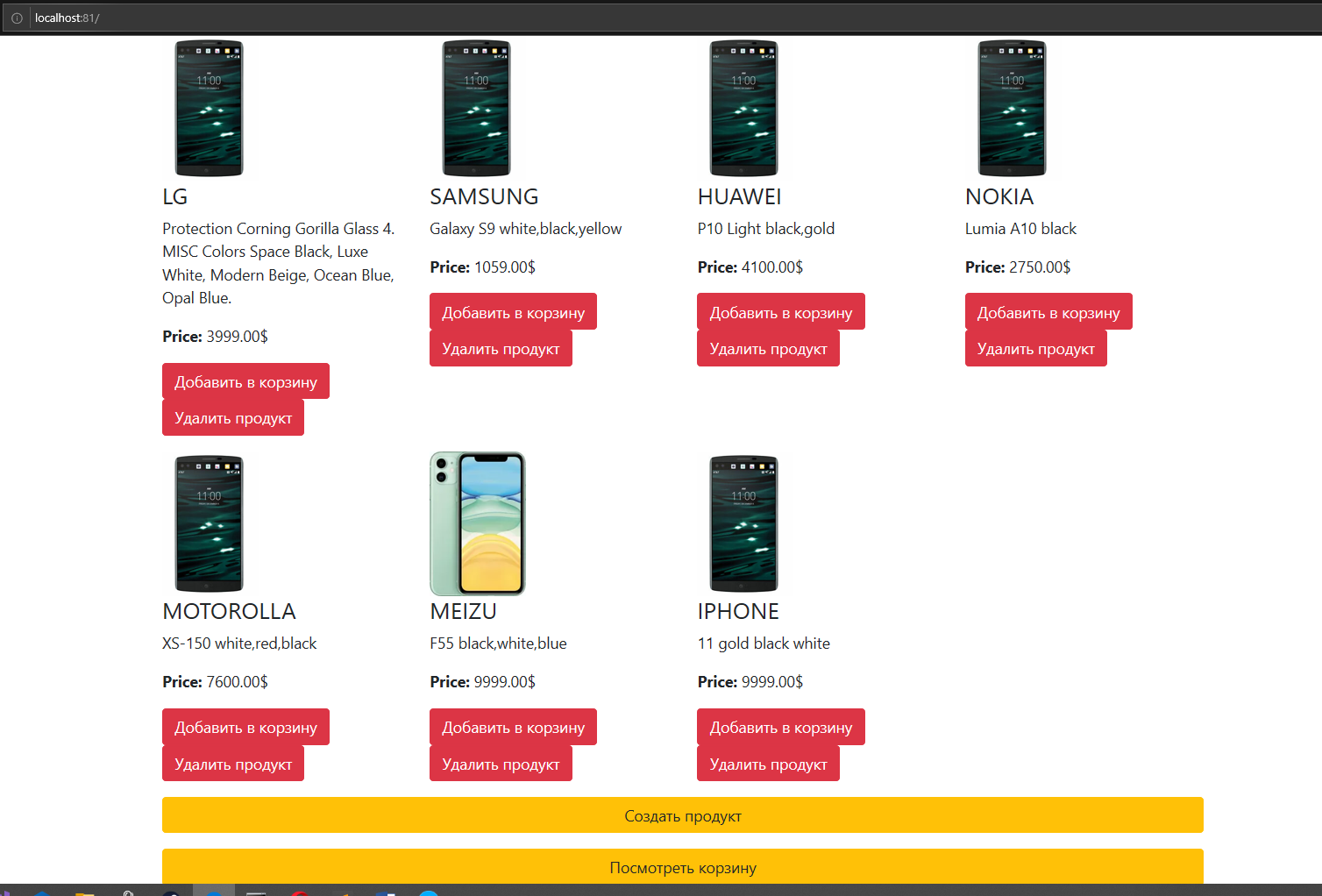


Рисунок 14. Результат работы приложения

**Ссылка на репозиторий GitHub:** https://github.com/seeyuointhespring/PHPLabs

ВЫВОД

В ходе работы было проведено ознакомление со средством контейнеризации Docker, оно было использовано для запуска приложения на Laravel.