Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

ИНСТИТУТ ЭЛЕКТРОННЫХ И ИНФОРМАЦИОННЫХ СИСТЕМ

**ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ**

РАЗРАБОТКА ПРОГРАММНОГО

ПРОДУКТА «ИНТЕРНЕТ-МАГАЗИН»

Курсовой проект по учебной дисциплине

«Проектирование и разработка веб-приложений»

по специальности 09.02.07 Информационные системы и программирование

ПТК.КП 0902 07.000ПЗ

|  |
| --- |
| Руководители  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 года |
| Студент группы 2994  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 года |

Содержание

[Введение 3](#_heading=h.3rdcrjn)

[1 Общая часть 5](#_heading=h.26in1rg)

[1.1 Постановка задачи 5](#_heading=h.lnxbz9)

[1.2 Обоснование проектных решений 6](#_heading=h.35nkun2)

[1.3 Обзор и анализ существующих программных систем 1](#_heading=h.2jxsxqh)2

[2 Специальная часть 1](#_heading=h.z337ya)3

[2.1 Анализ задачи 1](#_heading=h.3j2qqm3)3

[2.2 Описание логической структуры 1](#_heading=h.1y810tw)6

[2.3 Описание работы программы 1](#_heading=h.4i7ojhp)8

[2.4 Руководство оператора](#_heading=h.2xcytpi) 21

[Заключение 2](#_heading=h.1ci93xb)3

[Список литературы 2](#_heading=h.3whwml4)4

[Приложение А 2](#_heading=h.2bn6wsx)5

[Приложение Б 2](#_heading=h.3as4poj)6

[Приложение В 3](#_heading=h.49x2ik5)0

# Введение

В соответствии с заданием на курсовую работу требовалось разработать сервис по продаже и покупке электронных устройств, то есть интернет-магазин, специализация которого заключается в торговле приспособлениями электроники, требующихся в повседневной жизни обычного человека. Интернет-магазин — сайт, торгующий товарами посредством сети Интернет. Позволяет пользователям онлайн, в своём браузере или через мобильное приложение, сформировать заказ на покупку, выбрать способ оплаты и доставки заказа, оплатить заказ. При этом продажа товаров осуществляется дистанционным способом и она накладывает ограничения на продаваемые товары. Типичный интернет-магазин позволяет клиенту просматривать ассортимент продуктов и услуг фирмы, просматривать фотографии или изображения продуктов, а также информацию о технических характеристиках продукта и ценах. Интернет-магазины обычно позволяют покупателям использовать функции поиска, чтобы найти конкретные модели, бренды или предметы.

Актуальность данной курсовой работы заключается в том, что в настоящее время интернет-магазины пользуются популярностью, как и у обычных пользователей, так и у фирм, желающих купить или же продать свой товар. Интернет-магазины сегодня обладают куда большими преимуществами перед обычными магазинами. Новые технологии не стоят на месте, и с помощью таких специализированных сайтов клиенты могут совершать любые покупки, даже не выходя из дома. Покупки в интернет-магазинах экономят время потребителя. Интернет-магазины работают без перерывов, выходных и праздников. На сайте, в отличие от обычных магазинов, нет очередей. Покупку любого выбранного товара можно отложить, при этом вся информация о нём сохраниться в аккаунте. Многим пользователям интернет-магазинов важен комфорт покупателей - отсутствие очередей и назойливых консультантов, возможность в любое время отказаться от выбранного товара или поменять какие-то его характеристики одним щелчком мыши. Интернет-покупки также можно легко и быстро оплатить наличными, банковской картой, переводом или электронными деньгами. Для этого нет необходимости выходить из дома.

# 1 Общая часть

# 1.1 Постановка задачи

1.1.1 Обоснование необходимости разработки

В соответствии с заданием на курсовую работу требовалось разработать сервис по продаже электронных устройств.

1.1.2 Технико-математическое описание задачи

Для реализации были использованы следующие средства: HTML, CSS, HTTP и ООП.

HTML — стандартизированный язык разметки для веб-документов. Большинство веб-страниц содержат разметку, написанную на HTML или XHTML; HTML интерпретируется браузером, и полученный форматированный текст отображается на мониторе компьютера или мобильного устройства. HTML до версии 5 определяется как применение SGML, стандартного языка разметки общего назначения, основанного на ISO 8879. Спецификация HTML5 сформулирована в терминах объектной модели документа (DOM). В Интернете страницы HTML обычно отправляются с сервера в браузер по протоколу HTTP или HTTPS открытым текстом или в зашифрованном виде.

CSS — формальный язык для описания внешнего вида документов, написанных с помощью языков разметки. В основном он используется как средство описания и оформления внешнего вида веб-страниц, написанных на языках разметки, таких как HTML и XHTML, но также может применяться к любым XML-документам, таким как SVG и XUL.

HyperText Transfer Protocol (HTTP) — это протокол передачи данных прикладного уровня. Он основан на технологии клиент-сервер и в первую очередь предназначен для передачи веб-страниц, а именно текстовых файлов с HTML разметкой, но также успешно передает изображения или приложения, как связанные, так и несвязанные с веб-страницами. HTTP предполагает, что клиентское программное обеспечение, веб-браузер, может отображать гипертекстовые веб-страницы и другие типы файлов в удобной для пользователя форме.

Объектно-ориентированное программирование (ООП) — метод программирования, основанный на представлении программы в виде набора объектов, где каждый объект является экземпляром определенного класса, а классы образуют иерархию наследования. Объектная ориентация была разработана для того, чтобы связать поведение объекта с его классом. Подобное представление облегчает людям восприятие окружающего мира как объекты, пригодные для какой-либо классификации, например, деление на живые и неживые существа.

1.1.3 Характеристика бизнес-процессов

Чтобы пользователи могли использовать сервис продажи и покупки электронных устройств им необходимо зарегистрироваться. После регистрации автоматически создается пустая корзина, в которую они могут добавлять и удалять товары по своему желанию.

1.1.4 Требование к программе

Для корректной работы программного продукта требуется веб-сервер с поддержкой Node.js версии 20.12.2, PHP версии 8.2, а также поддержка СУБД PostgreSQL версии 16.2. Пользователю необходим компьютер с выходом в интернет и браузер Google Chrome, Firefox 110 версий и старше или их аналоги.

# 1.2 Обоснование проектных решений

1.2.1 Обоснование выбора языков программирования

Для разработки сервиса были выбраны: язык программирования JavaScript и языки программирования SQL и PHP.

JavaScript — язык программирования. Был выбран в качестве языка программирования для клиентской стороны, т.е. функциональности браузера. Он поддерживает объектно-ориентированный, императивный и функциональный стили; является реализацией стандарта ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Это наиболее широко используемый язык сценариев в браузерах для добавления интерактивности веб-страницам. Основные архитектурные особенности: динамическая типизация, слабая типизация, автоматическое управление памятью, программирование прототипов, функции как объекты первого класса. JavaScript находится под влиянием многих языков и стремится быть похожим на Java, но в то же время быть простым в использовании даже для непрограммистов. JavaScript не принадлежит какой-либо компании или организации, что отличает его от некоторых языков программирования, используемых в веб-разработке.

SQL — декларативный язык программирования, используемый для манипулирования и управления базами данных. SQL служит для создания, модификации и управления данными в реляционных базах данных, которые управляются соответствующей системой управления базами данных. SQL является, прежде всего, реляционным языком информационной логики, предназначенным для описания, модификации и извлечения данных, хранящихся в базах данных. SQL считается декларативным языком программирования. Изначально он был основным способом работы с базами данных и мог выполнять следующие операции: создание новых таблиц в базе данных, добавление новых записей в таблицу, изменение, удаление и выборка записей из одной или нескольких таблиц (по заданным критериям) и изменение структуры таблицы. Со временем SQL усложнился, обогатился новым синтаксисом, получил возможность писать и управлять новыми хранимыми объектами, такими как индексы, представления, триггеры и хранимые процедуры, а также приобрел функции, характерные для языков программирования. Даже после всех этих изменений SQL остается наиболее распространенным языковым средством для взаимодействия между прикладным программным обеспечением и базами данных. В то же время современные СУБД и использующие их информационные системы предоставляют пользователям сложные инструменты для визуальных запросов.

PHP — скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов. Вот некоторые из ключевых возможностей PHP:

* динамические веб-страницы: PHP позволяет создавать динамические веб-страницы, которые могут адаптироваться к пользовательскому вводу и предоставлять интерактивные функции;
* обработка форм: PHP легко интегрируется с HTML-формами, что делает его идеальным для обработки данных, отправляемых с веб-страниц;
* работа с базами данных: PHP имеет широкую поддержку для множества систем управления базами данных (например, MySQL, PostgreSQL, SQLite), что делает его отличным выбором для создания веб-приложений с базами данных;
* работа с файлами: PHP предоставляет функции для чтения, записи и обработки файлов на сервере, что полезно для управления содержимым сайта;
* создание сессий: PHP может создавать и управлять сессиями пользователей, что полезно для отслеживания состояния пользователя между запросами;
* работа с изображениями: PHP позволяет манипулировать изображениями, изменять их размеры, форматы и многое другое;
* создание API: PHP может использоваться для создания веб-сервисов и API, что делает его полезным для разработки клиент-серверных приложений;
* интеграция с другими технологиями: PHP может легко интегрироваться с другими технологиями, такими как HTML, CSS, JavaScript, что позволяет создавать полноценные веб-приложения;
* кросс-платформенность: PHP поддерживается на различных операционных системах (Windows, Linux, macOS) и работает с различными веб-серверами (Apache, Nginx);

1.2.2 Инструментальные средства

Для разработки был выбран Laravel —— это бесплатный PHP-фреймворк с открытым исходным кодом, специально разработанный для создания сложных сайтов и веб-приложений. Позволяет упростить аутентификацию, маршрутизацию, сессии, кэширование, архитектуру приложения, работу с базой данных.

Composer —— пакетный менеджер уровня приложений для языка программирования PHP, который предоставляет средства по управлению зависимостями в PHP-приложении. Composer работает через интерфейс командной строки и устанавливает зависимости для приложения.

Node.js —— программная платформа, построенная на базе JS-движка V8. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API, написанный на C++, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера, но есть возможность разрабатывать на Node.js и десктопные оконные приложения. Node.js основан на событийно-ориентированном и асинхронном (или реактивном) программировании с неблокирующим вводом-выводом.

Bootstrap — открытый и бесплатный HTML, CSS и JS фреймворк, который используется веб-разработчиками для быстрой вёрстки адаптивных дизайнов сайтов и веб-приложений. Фреймворк Bootstrap используется по всему миру не только независимыми разработчиками, но иногда и целыми компаниями. Основная область его применения – это фронтенд разработка сайтов и интерфейсов. Среди аналогичных систем фреймворк Bootstrap является самым популярным.

Фреймворк Bootstrap – это набор набор CSS и JavaScript файлов. Чтобы его использовать эти файлы необходимо просто подключить к странице. После этого станут доступны инструменты данного фреймворка: колоночная система (сетка Bootstrap), классы и компоненты*.*

PostgreSQL — свободная объектно-реляционная система управления базами данных (СУБД), основанная на языке SQL.

СУБД отличается высокой надёжностью и хорошей производительностью, поддерживает транзакции (ACID), репликация реализована встроенными механизмами. При этом система расширяемая — можно создавать свои типы данных и индексов, а также расширять поведение при помощи языков программирования.

Из дополнительных возможностей PostgreSQL хочется отдельно отметить тот факт, что эта СУБД позволяет работать не только со структурированными и нормализованными, но и со слабоструктурированными данными (в форматах json / jsonb), при этом эти данные индексируются и работа с ними действительно удобна. Особенную ценность эти возможности обретают при разработке систем, где модели данных должны быть кастомизуемыми в режиме реального времени на уровне GUI — в интернет-магазинах, где у товаров может быть множество разнообразных характеристик, в системах электронного документооборота и в других бизнес-приложениях.

Git —— система управления версиями с распределенной архитектурой. В Git каждая рабочая копия кода сама по себе является репозиторием. Это позволяет всем разработчикам хранить историю изменений в полном объеме. Разработка в Git ориентирована на обеспечение высокой производительности, безопасности и гибкости распределенной системы.

Postman — инструментальная платформа для создания и использования API. С помощью приложения можно создавать и редактировать простые или сложные HTTP-запросы. Составленные запросы автоматически сохраняются для повторного использования в будущем. Ответы сервера также могут быть сохранены в виде файлов на жестком диске. Таким образом, Postman экономит много времени веб-разработчикам и ручным тестировщикам.

1.2.3 Обоснование выбора среды программирования

Для разработки программного продукта был выбран Visual Studio Code (VS Code) — текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией. Visual Studio Code основан на Electron и реализуется через веб-редактор Monaco, разработанный для Visual Studio Online. VS Code позволяет легко писать, форматировать и редактировать код на разных языках. С его помощью можно быстро создать проект и структуру файлов в нем, он подсвечивает синтаксис кода и помогает автоматически править ошибки. В нем есть возможности для отладки и запуска кода на некоторых языках.

1.2.4 Информационное обеспечение

Для решения прикладных задач по оформлению документации использовался пакет программ Microsoft Office 365 - программный продукт от компании Microsoft, объединяющий набор веб-сервисов, который распространяется на основе подписки по схеме «программное обеспечение как услуга». Набор предоставляет доступ к различным программам и услугам на основе платформы Microsoft Office, электронной почте бизнес-класса, функционалу для общения и управления документами. В частности, данный текстовый процессор использовался для создания, просмотра, редактирования и форматирования текста.

При проектирование реляционной схемы базы данных и схемы функциональной модели использован draw.io — бесплатное онлайн-программное обеспечение для создания блок-схем, диаграмм процессов, организационных диаграмм, UML, ER и сетевых диаграмм.

# 1.3 Обзор и анализ существующих программных систем

Перед разработкой программного продукта требуется произвести поиск аналогов.

Tilda — конструктор для разработки веб-сайта. Платформа предлагает использование более 400 блоков. Каждый блок имеет свои настройки, позволяющие создать сайт с индивидуальным дизайном. Он является популярным решением при разработке сайтов, в частности сервисов по покупке и продаже, но имеет более серьёзные недостатки:

* 1 Гб, который Tilda дает в аренду для фотографий и файлов, может оказаться мало, если количество товаров превышает 200;
* простой фильтр, предлагаемый в Tilda, может не подойти для некоторых категорий товаров;
* отсутствует личный кабинет с историей покупок, бонусами и пр.;
* нельзя произвести сравнение товаров;
* не изменяется внешний вид корзины;
* большинство блоков неактивны для разработки;
* имеет высокую цену;
* закрытая кодовая база;
* несоблюдение современных правил вёрстки страниц;
* отсутствие возможности дальнейшей самостоятельной разработки.

# 2 Специальная часть

# 2.1 Анализ задачи

2.1.1 Информационное моделирование предметной области

В соответствии с заданием на курсовую работу требовалось разработать сервис по продаже и покупке электронных устройств. Сервис предназначен для продажи и покупки электронных устройств.

2.1.1.1 Построение информационной модели

Работа над сервисом началась с проектирования и разработки базы данных. Чтобы работать с сервисом, пользователю необходим аккаунт, поэтому необходима таблица, которая хранит данные о пользователях сервиса. Для этого необходимо создать таблицу users со следующими полями:

* идентификатор пользователя, который является первичным ключом id;
* адрес электронной почты email;
* роль пользователя name;
* имя пользователя role;
* телефон пользователя phone;
* адрес пользователя address;
* пароль password.

При авторизации создаётся токен, который предназначен для выполнения авторизированных запросов, а также для автоматического входа при переходе на сайт.

Для редактирования собственной корзины, пользователю необходимо авторизоваться. Для этого также создаются таблица cart со следующими полями:

* идентификатор корзины id;
* идентификатор добавленного в корзину продукта product\_id;
* количество добавленного продукта quantity;
* идентификатор пользователя, которому принадлежит корзина, user\_id.

Для хранения продуктов, то есть электронных девайсов, в базе данных создается таблица products.

Таблица products хранит непосредственно все продукты в магазине. Содержит поля:

* идентификатор продукта id;
* название продукта title;
* цена продукта price;
* изображение продукта img;
* описание продукта description;
* количество продукта в наличии quantity.

Для хранения заказов пользователей в базе данных создается таблица orders.

Таблица orders хранит непосредственно все заказы пользователей в магазине. Для отправки заказа необходимо авторизироваться. Содержит поля:

* идентификатор заказа id;
* идентификатор корзины пользователя cart\_id;
* статус заказа status.

Диаграмма классов UML представлена в приложении А на рисунке А.1.

2.1.2 Проектирование пользовательского интерфейса

2.1.2.1 Экран регистрации

На экране регистрации по центру располагается логотип сайта, через который можно вернуться на главную страницу, ниже находится надпись «Already registered?» с ссылкой на экран авторизации, если у пользователя уже есть аккаунт. Также по центру находиться форма для регистрации нового аккаунта, в которой нужно ввести имя нового аккаунта, почту, телефон, адрес, придумать и подтвердить пароль.

2.1.2.2 Экран авторизации

На экране авторизации по центру располагается логотип сайта, через который можно вернуться на главную страницу, ниже которой находится надпись «Want to register?», перенаправляющая на экран регистрации. Также по центру находиться форма для авторизации, в которой нужно ввести свою почту и пароль.

2.1.2.3 Главный экран магазина

Главный экран магазина состоит из нескольких компонентов:

* панель навигации;
* баннер с возможностью пролистывания слайдов;
* информационные блоки;
* панель товаров.

При открытии страницы первым компонентом встречается панель навигации, которая расположена в верхней части экрана, внутри панели перечислены страницы сайта. Ниже находится баннер с возможностью пролистывания слайдов. После список товаров, каждый товар представлен в виде названия, изображения, краткого описания и рейтинга. Предусмотрен поиск товара по названию. Товары можно добавить в корзину через кнопку и указания количества приобретаемых товаров. При большом количестве товаров, товары будут расположены по страницам. Ниже списка товаров располагаются информационные блоки.

2.1.2.4 Корзина пользователя

При открытии страницы пользователь увидит список товаров, добавленных в корзину. Каждый товар представлен в виде названия, количества товара в корзине, цены и кнопки удаления из корзины. Под списком товаров находится кнопка для покупки товаров.

2.1.2.5 Страница профиля

У каждого профиля существует своя страница редактирования. При открытии страницы пользователь предложения для редактирования информации профиля. Можно изменить имя профиля, почту, пароль. Есть кнопка для удаления аккаунта.

2.1.2.6 Панель администратора

Панель администратора состоит из нескольких страниц. Есть страницы для добавления, изменения и удаления товаров и отправленных пользователями заказов.

Макеты экранов можно увидеть в приложении Б на рисунках Б.1 – Б.6.

# 2.2 Описание логической структуры

2.2.1 Алгоритм программы

Алгоритм программы – это точное предписание, совокупность последовательных шагов, схема действий, которое определяет процесс перехода от первичных данных к желаемому результату.

Рассмотрим для примера алгоритм работы процедуры addcart, которая отвечает за добавление товара в корзине пользователя в базу данных. При нажатии кнопки «Add Cart» происходит отправка запроса на сервер, содержащего указанное количество товара, id пользователя и id товара. Сначала проявляется авторизирован ли пользователь. На сервере происходит проверка, авторизирован ли на сайте пользователь. Если да, товара в корзине пользователя добавляется в базу данных, если нет, пользователя перенаправляют на страницу входа.

2.2.2 Используемые методы

2.2.2.1 Стандартные методы

redirect – этот метод перенаправляет пользователя по определенному адресу.

view – этот метод возвращает пользователю определенное представление.

get – метод для получения определенных значений из базы данных.

count – метод для получения количества определенных значений из базы данных.

delete – метод для удаления определенных значений из базы данных.

find – метод для поиска определенных значений из базы данных.

save – метод для сохранения определенных значений в базу данных.

all – метод для получения всех значений из базы данных.

move – метод для перемещения файла под определенным именем.

time – метод для получения текущего времени.

2.2.2.2 Пользовательские методы

login – метод, который отправляет запрос к серверу для авторизации и получения токена.

registration – метод регистрации, который отправляет запрос к серверу для регистрации.

redirect – метод для проверки типа пользователя.

index – метод для показа главной страницы.

search – метод для поиска товара по названию.

addcart – метод для добавления товара в корзину.

showcart – метод для показа товаров в корзине.

deletecart – метод для удаления товаров из корзины.

confirmorder – метод для добавления заказа пользователя в базу данных.

createDevice – метод для добавления в базу данных товаров.

product – метод для показа всех товаров.

showcart – метод для показа товаров в корзине.

deleteproduct – метод для удаления товаров из базы данных.

uploadproduct – метод для добавления товаров в базу данных.

updateview – метод для показа страницы для изменения товаров в базе данных.

updateproduct – метод для изменения товаров в базе данных.

showorder – метод для показа всех заказов.

adminpanel – метод для показа страницы панели администратора.

Были разработаны Blade шаблоны, предназначенные для отрисовки элементов на странице, которые являются многоразовыми автономными блоками кода имея свой HTML, CSS, JavaScript и PHP код.

2.2.3 Составные части программы и связи между ними

Данный программный продукт основан на монолитной архитектуре.

Приложение построено как цельный продукт с единой кодовой базой. Взаимодействие с сервисом при этом происходит через API или веб-интерфейс. Каждая компонент имеет взаимодействие с другими.

# 2.3 Описание работы программы

2.3.1 Общие сведения

Сервис по продаже и покупке электронных устройств предназначен для покупки товаров пользователями. Также предоставляется возможность оставлять свои оценки каждому товару.

2.3.2 Функциональное назначение

Сервис предоставляет возможность пользователю:

* зарегистрироваться, если у него нет аккаунта;
* войти под своей электронной почтой и паролем;
* добавлять товары в корзину в определенном количестве;
* удалять товары из корзины;
* покупать товары из корзины, отправляя свой заказ;
* редактировать свой профиль.

А также возможность администратору просматривать, добавлять, изменять и удалять товары и заказы пользователей непосредственно на сайте.

2.3.4 Входные данные

Входными данными являются адрес электронной почты и пароль пользователя при авторизации или же регистрации. Также ввод текстовой информации или же иных свойств на странице.

2.3.5 Выходные данные

В качестве выходных данных пользователь получает доступ к странице со всеми товарами, страницам с каждым товаром отдельно и корзине. Администратор также получает доступ к панели администратора.

2.3.6 Тестирование

В данном разделе представлены результаты тестирования программы.

Таблица 1 - Работа программы

|  |  |  |
| --- | --- | --- |
| Этап работы программы | Действия пользователя | Ожидаемое поведение программы |
| Открытие приложения | Пользователь переходит на главную страницу приложения | Корректно выводится содержимое страницы |
| Регистрация | Пользователь переходит на страницу регистрации | Открывается страница с формой для регистрации |
| Ввод | Пользователь учетные данные для создания учетной записи | Происходит автоматическая авторизация, пользователь попадает на главную страницу |
| Ввод | Пользователь вводит данные профиля | Пользователь попадает на главную страницу |
| Поиск | Пользователь вводит поисковый запрос | Отображаются корректные результаты поискового запроса |
| Переход на страницу корзины | Пользователь переходит на страницу корзины | Открывает страница корзины с корректными данными |
| Ввод | Пользователей добавляет товар в корзину | Страницу перезагружается, в корзину корректно добавлен товар |

Таблица 2 - Тест-кейс 1

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TU01 |
| Заголовок | | | Проверка работы авторизации учетной записи при вводе корректных данных. |
| Предусловие | | | Открыта страница авторизации. |
| Тестовые данные | | | Адрес электронной почты, пароль |
| Ожидаемый результат | | | Успешный вход и перенаправление на главную |
| Шаг | Действие | Предполагаемый результат | Pass / Fail |
|  | Переход на страницу авторизации | Открылась страница с формой для ввода учетных данных | Pass |
|  | Ввод адреса электронной почты | Поле для ввода адреса электронной почты заполнено | Pass |
|  | Ввод пароля | Поле для ввода пароля заполнено символами “\*” | Pass |
|  | Отправка формы | Произошла успешная авторизация учетной записи, произошла переадресация на главную страницу. | Pass |
| Постусловие | | | Pass |
| Результат | | | Pass |

Таблица 3 - Тест-кейс 2

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TU01 |
| Заголовок | | | Проверка работы авторизации учетной записи при вводе некорректных данных. |
| Предусловие | | | Открыта страница авторизации. |
| Тестовые данные | | | Адрес электронной почты, пароль |
| Ожидаемый результат | | | Успешный вход и перенаправление на главную |
| Шаг | Действие | Предполагаемый результат | Pass / Fail |
|  | Переход на страницу авторизации | Открылась страница с формой для ввода учетных данных | Pass |
|  | Ввод адреса электронной почты | Поле для ввода адреса электронной почты заполнено | Pass |
|  | Ввод пароля | Поле для ввода пароля заполнено символами “\*” | Pass |
|  | Отправка формы | Произошла переадресация на страницу авторизации. Для поля с некорректными данными выведена ошибка. | Pass |
| Постусловие | | | Pass |
| Результат | | | Pass |

2.3.7 Вызов и загрузка

Программный продукт не требует установки. Он размещён на веб-сервере, для получения доступа необходимо ввести адрес сайта в браузере.

# 2.4 Руководство оператора

2.4.1 Назначение программы

Сервис по продаже и покупке электронных устройств предназначен для покупки товаров пользователями. Также предоставляется возможность редактировать свой профиль.

2.4.1.1 Работа пользователя

Для того, чтобы начать работать с программным продуктом, необходимо зайти с персонального компьютера в браузер, после этого ввести в адресную строку ссылку с доменным именем приложения. Неавторизированному пользователю необходимо будет авторизироваться, а незарегистрированный пользователь должен перейти на страницу с формой регистрации и зарегистрироваться.

После входа в свой аккаунт происходит перенаправление на главную страницу магазина, где можно просмотреть все доступные товары, осуществить поиск по названию или добавить товары в определенном количестве в корзину. Если в корзине есть товары, то на странице с корзиной можно отправить заказ.

# Заключение

Результатом курсовой работы стала разработка сервиса по продаже и покупке электронных устройств, который предоставляет возможность пользователю просмотра списка предлагаемых товаров, добавлять отдельные товары в корзину, удалять их из корзины, а также отправлять заказы на странице корзины. Также предоставляется возможность редактировать профиль, а администратор может удалять или изменять существующие товары и просматривать отправленные заказы.

Работы, выполненные в ходе проектирования, и разработанный программный продукт описаны в настоящей пояснительной записке с соблюдением требований ГОСТ ЕСПД.

Таким образом задание на курсовую работу выполнено в полном объеме.

# Список литературы

1. Глушаков С.В. Базы данных / С.В. Глушаков, Д.В. Ломотько. – М.: Харьков: Фолио, 2018. - 504 c;
2. ГОСТ 19.701-90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. Москва, Издательство стандартов, 2010. – 24с;
3. Методические указания по оформлению курсовых и выпускной квалификационной работ: учеб. пособие для студентов ПТК НовГУ/ Авт.сост. Чернега А. М., ПТК НовГу им. Ярослава Мудрого. — Великий Новгород, 2017–27 с;
4. официальная документация Bootstrap на английском языке [Электронный ресурс] URL: https://getbootstrap.com/docs/, дата обращения к ресурсу 20.04.2023;
5. официальная документация Laravel на русском языке [Электронный ресурс] URL: https://laravel.su/docs/11.x/, дата обращения к ресурсу 10.04.2023.

# Приложение А

# (обязательное)

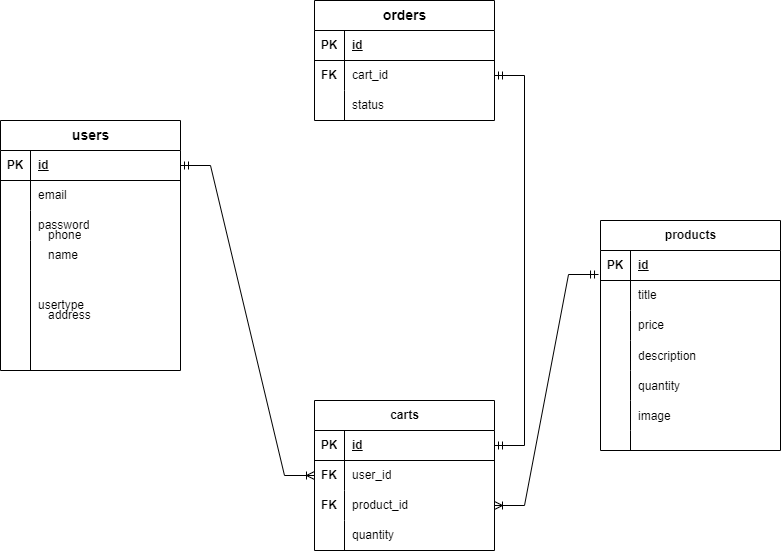


Рисунок А.1 – UML-диаграмма классов

# Приложение Б

# (обязательное)

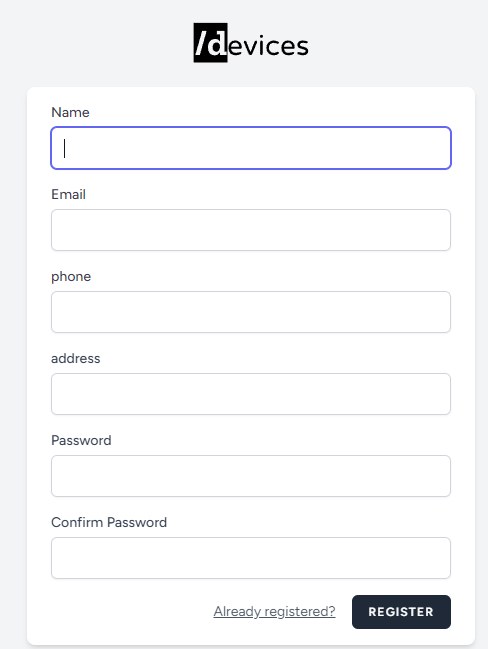


Рисунок Б.1 – Экран регистрации

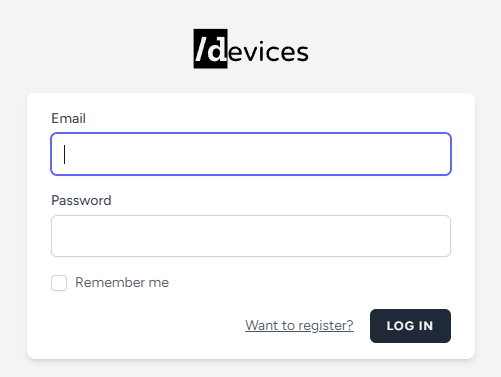


Рисунок Б.2 – Экран авторизации

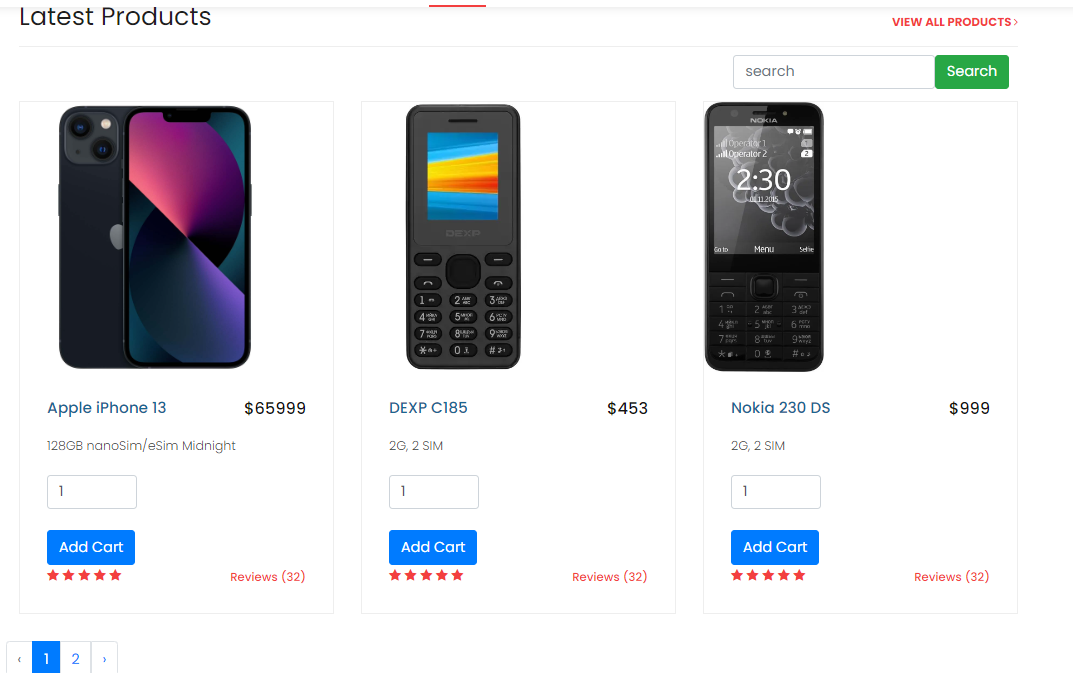


Рисунок Б.3 – Главный экран магазина

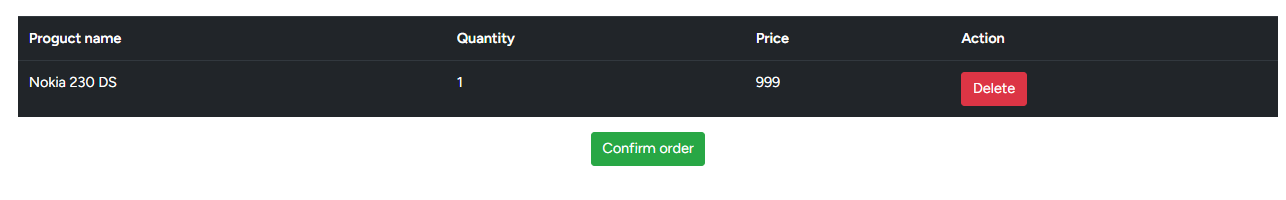


Рисунок Б.4 – Экран корзины пользователя

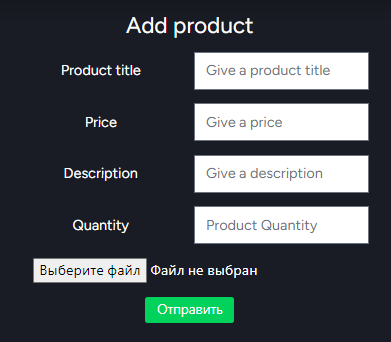


Рисунок Б.5 – Экран добавления товара на панели администратора

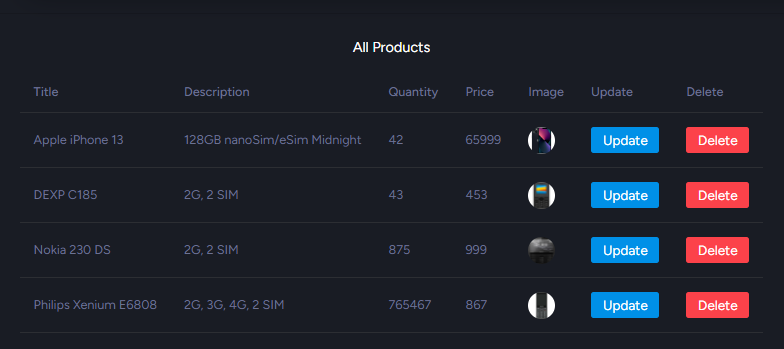


Рисунок Б.6 – Экран показа всех товаров на панели администратора

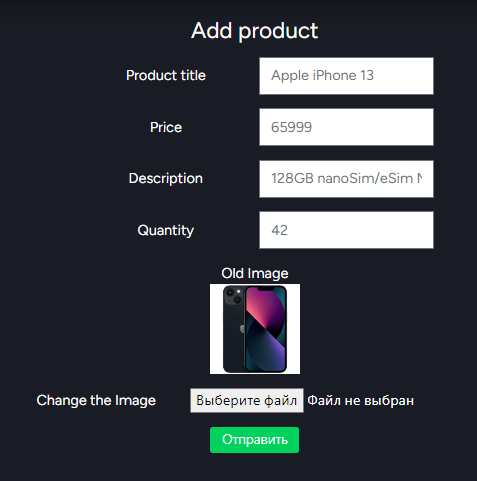


Рисунок Б.6 – Экран изменения товара на панели администратора

# Приложение В

# (обязательное)

public function addcart(Request $request, $id)

{

if(Auth::id())

{

$user = auth()->user();

$product = product::find($id);

$cart = new Cart;

$cart->user\_id = $user->id;

$cart->product\_id = $product->id;

$cart->quantity = $request->quantity;

$cart->save();

return redirect()->back()->with('message', 'Product Added Successfully');

}

else return redirect('login');

}

В.1 – метод addCart для выставления пользовательской оценки