**МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОРЕНБУРГСКОЙ ОБЛАСТИ**

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«ОРЕНБУРГСКИЙ КОЛЛЕДЖ ЭКОНОМИКИ И ИНФОРМАТИКИ»**

**(ГАПОУ СПО ОКЭИ)**

**КУРСОВОЙ ПРОЕКТ**

*ОКЭИ 09.02.07. 4323. № ПЗ*

(код документа)

Количество листов

Дата готовности

Руководитель

Разработал

Защищен\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ с оценкой

(дата)

# Содержание

Введение  3  
1 Анализ предметной области 5

2 Проектирование приложения 8

3 Разработка программного обеспечения 10

3.1 Описание технологического стека разработки 10

3.2 Описание алгоритма работы 12

3.3 Описание интерфейса пользователя 13

4 Тестирование приложения 14

4.1 План тестирования 14

4.2 Оценка результатов проведения тестирования 16

Заключение 22

Список используемых источников 23

Приложение А 25

Приложение Б 26

Приложение В 27

Приложение Г 28

Приложение Д 30

# Введение

Веб-приложение разрабатывается для использования через веб-браузер. Оно хранится на удаленных серверах и обеспечивает пользователю доступ к функционалу и данным через интернет.

Они могут легко включать в себя инструменты для анализа данных и создания отчетов, что помогает в принятии информированных решений.

Пользователи могут использовать веб-приложения на разных устройствах, будь то компьютеры, смартфоны, планшеты или даже смарт-телевизоры, так как они не зависят от операционной системы.

Таким образом, веб-приложения широко используются в бизнесе, образовании, здравоохранении, государственном управлении, развлечениях и многих других сферах для улучшения производительности, доступности к информации и взаимодействия с пользователями.

В ходе курсовой должно быть решено несколько ключевых задач: разработка серверной части веб-приложения, включая создание интерфейса и определение функциональности, разработка системы .

Целью создания такого веб-приложения является улучшение доступности и удобства обслуживания клиентов.

Для успешной реализации намеченных задач требуется провести комплексную работу, которая включает в себя несколько ключевых этапов:

Прежде всего, необходимо определить и разработать логическую структуру будущего веб-приложения. Это включает в себя определение основных функциональных модулей, спецификацию их задач и взаимодействия между собой, также необходимо продумать, каким образом приложение будет собирать, хранить и обрабатывать.

Для успешной реализации намеченных задач требуется провести комплексную работу, которая включает в себя несколько ключевых этапов:

* прежде всего, необходимо определить и разработать логическую структуру будущего веб-приложения. Это включает в себя определение основных функциональных модулей, спецификацию их задач и взаимодействия между собой, также необходимо продумать, как пользователи будут взаимодействовать с системой;
* для удобства использования веб-приложения необходимо разработать дизайн пользовательского интерфейса. Это включает в себя выбор цветовой палитры, шрифтов, структуры страниц, расположения элементов управления и визуального оформления. Для реализации данного пункта должны быть разработаны эскизы и макеты будущего веб-приложения;
* после разработки дизайна необходимо перейти к верстке страниц приложения. Этот этап включает в себя создание HTML, CSS, JS, Bootstrap кода для каждой страницы, определенной в логике приложения. Важно обеспечить адаптивность и кросс-браузерность, чтобы приложение корректно отображалось на различных устройствах и в разных браузерах;
* далее необходимо разработать архитектуру базы данных для разрабатываемой серверной части, включая выбор подходящих таблиц;
* завершив этап разработки архитектуры базы данных, перейдем к созданию серверной части. Этот этап включает в себя подключение библиотеки ReadBeanPHP (ORM).

Для успешного решения этой проблемы необходимо тщательно спроектировать архитектуру приложения, уделяя особое внимание взаимосвязям между его компонентами, эффективности обработки данных и запросов пользователей, а также обеспечить масштабируемость при увеличении нагрузки.

Этот комплексный подход позволит создать полноценное приложение, которое удовлетворит потребности пользователей и поможет в достижении поставленных целей.

**1 Анализ предметной области**

NBB интернет магазин включает в себя услуги электронной коммерции. электронику, косметику, товары для дома и многое другое. Наш интернет-магазин позволяет клиентам удобно и безопасно совершать покупки, делать заказы онлайн и получать доставку в удобное для них время и место. Мы также предоставляем услуги по обработке платежей и обслуживанию клиентов, чтобы обеспечить нашим клиентам наилучший опыт покупок.

В предметной области интернет-магазина проводится анализ, который помогает определить основные характеристики и требования к магазину. Вот несколько основных направлений анализа предметной области:

- Рынок и конкуренция: Исследование рынка помогает определить долю рынка, понять потребности и предпочтения целевой аудитории, а также выявить конкурентов и изучить их стратегии.

- Товары и категории: Анализ товаров и категорий позволяет выявить самые востребованные товары и установить оптимальную структуру каталога.

- Цены и акции: Анализ цен и акций помогает определить оптимальные ценовые диапазоны, стратегии скидок и акций для привлечения клиентов.

- Корзина и оформление заказа: Анализ корзины и процесса оформления заказа помогает определить факторы, влияющие на конверсию клиентов и улучшить пользовательский опыт.

- Логистика и доставка: Анализ логистики и доставки позволяет оптимизировать процесс доставки товаров, выбрать наилучшие партнеры и улучшить качество обслуживания клиентов.

Все эти аспекты анализируются с целью оптимизации и улучшения работы интернет-магазина, удовлетворения потребностей клиентов и достижения успеха в данной предметной области.

1. **Проектирование приложения**

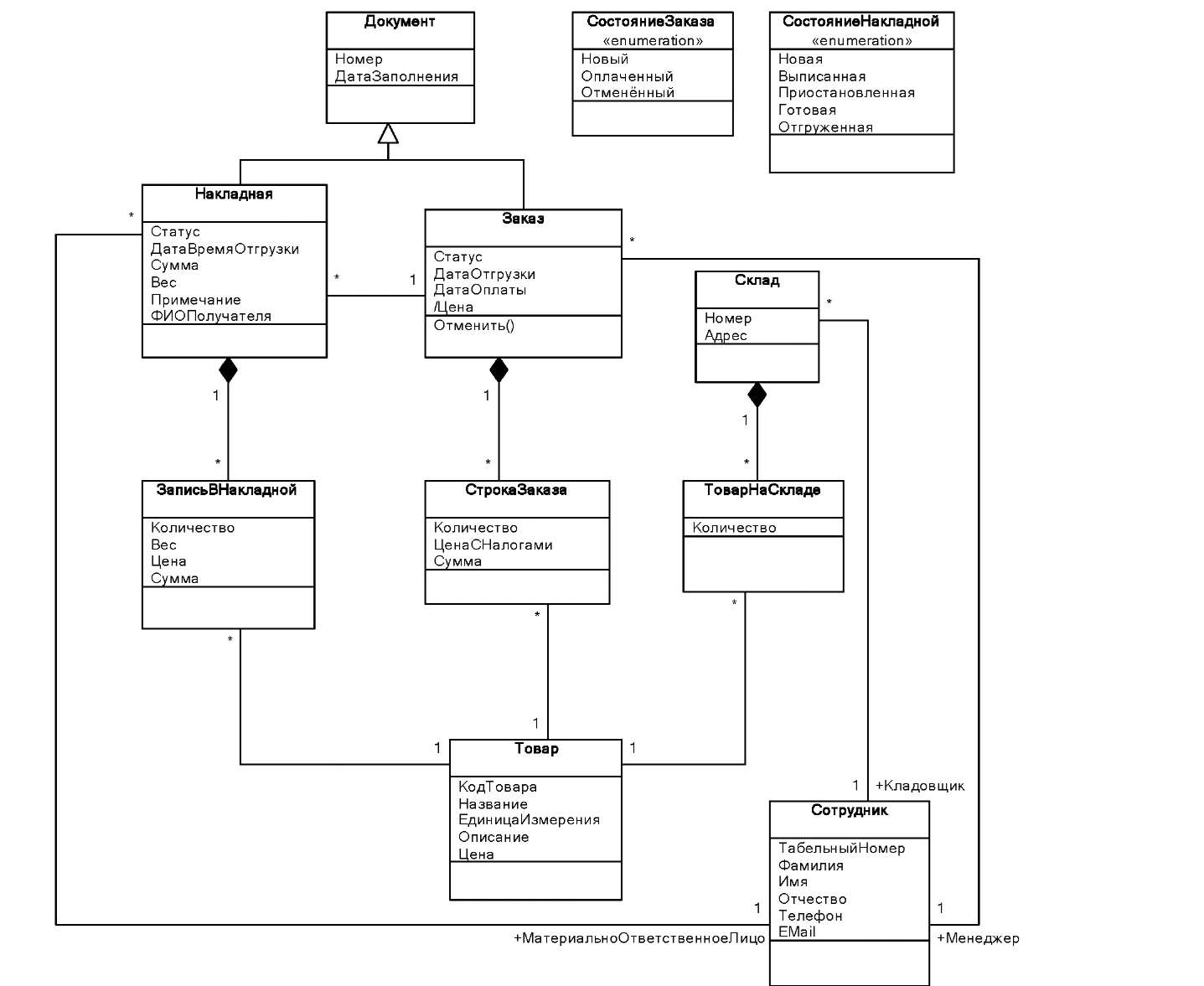
****

Рисунок 1 – Схема чат-бота

Функции пользователя:

* просмотр товара;
* добавления товара в корзину;
* удаление товара;
* фильтры товара;
* заказ товара.

Функции Администратора:

* просмотр статистики;
* просмотр заказов;
* добавления товара;
* удаления товара;
* управления учетными записями пользователей;

**3 Разработка программного обеспечения**

**3.1 Описание технологического стека разработки**

2.2.1 Среда разработки

В качестве интегрированной среды разработки рекомендуется использовать Visual Studio Code или PhpStorm. Для удобства работы с IDE могут использоваться такие плагины как Emmet.

2.2.2 Удаленный доступ

Для удаленного управления и развертывания приложения рекомендуется использовать SSH.

2.2.3 Технологии

Для реализации веб-приложения рекомендуется использовать веб-сервер Nginx или Apache.

В качестве языков программирования должны быть использованы:

* PHP 8.0 для разработки серверной логики;
* HTML для написания структуры сайта;
* JavaScript для клиентской логики.

CSS для создания и изменения стиля элементов веб-страниц и пользовательских интерфейсов.

Для реализации некоторых функций могут быть использованы библиотеки и фреймворки:

* PDO для упрощения работы с БД;
* Bootstrap для быстрого написания CSS кода.

**3.2 Описание алгоритма работы**

**3.3 Описание интерфейса пользователя**

**4 Тестирование приложения**

* 1. **План тестирования**

Для тестирования веб-приложения был выбрано несколько методов тестирования.

DDOS attack. Чат-бот может быть эффективным только в том случае,если он может быть устойчивым к DDOS атакам. Тестирование реализовано на рисунке 15.

Код программы тестирования DDOS атаки на веб-приложения:

import os

import time

class color:

    green = '\033[92m'

    red = '\033[91m'

    white = '\033[0m'

time.sleep(1)

os.system("clear")

time.sleep(1)

print(color.green + """

       \_\_\_\_  \_\_\_\_                \_\_\_   \_\_  \_\_             \_\_

      / \_\_ \/ \_\_ \\_\_\_\_  \_\_\_\_\_   /   | / /\_/ /\_\_\_\_\_ \_\_\_\_\_\_/ /\_\_

     / / / / / / / \_\_ \/ \_\_\_/  / /| |/ \_\_/ \_\_/ \_\_ `/ \_\_\_/ //\_/

    / /\_/ / /\_/ / /\_/ (\_\_  )  / \_\_\_ / /\_/ /\_/ /\_/ / /\_\_/ ,<

   /\_\_\_\_\_/\_\_\_\_\_/\\_\_\_\_/\_\_\_\_/  /\_/  |\_\\_\_/\\_\_/\\_\_,\_/\\_\_\_/\_/|\_| """ + color.red + """

                 (🅓🅓🅞🅢 🅐🅣🅣🅐🅒🅚)

""" + color.white)

print("\t1.start")

print("\t2.Exit")

choose = str(input("\nDDosattack/> "))

if(str(choose) == '1'):

  time.sleep(1)

  os.system("clear")

  time.sleep(1)

  os.system("figlet DDos Attack")

  try2 = str(input("Enter ip: "))

  time.sleep(1)

  try3 = str(input("Enter the number of packets: "))

  time.sleep(2)

  os.system("ping -s 1000 -w " + try3 + " " + try2)

  try4 = str(input("Do you want try again? [y/n] "))

  if(str(try4) == 'y'):

    os.system("python3 ddosattack.py")

  elif(str(try4) == 'n'):

      time.sleep(1)

      os.system("clear")

      time.sleep(1)

      print("good bye")

      exit(1)

  else:

      time.sleep(1)

      os.system("clear")

      time.sleep(1)

      print(color.red + "Error DDosAttack" + color.white)

      time.sleep(2)

      try5 = str(input("press Enter... "))

      if(str(try5) == ''):

        os.system("python3 ddosattack.py")

      else:

          os.system("python3 ddosattack.py")

elif(str(choose) == '2'):

    time.sleep(1)

    os.system("clear")

    time.sleep(1)

    print("good bye")

    exit(1)

else:

    time.sleep(1)

    os.system("clear")

    time.sleep(1)

    print(color.red + "Error DDosAttack!" + color.white)

    time.sleep(1)

    try1 = str(input("press Enter... "))

    if(str(choose) == ''):

      os.system("python3 ddosattack.py")

    else:

        os.system("python3 ddosattack.py")

# DDosAttack

* 1. **Оценка результатов проведения тестирования**

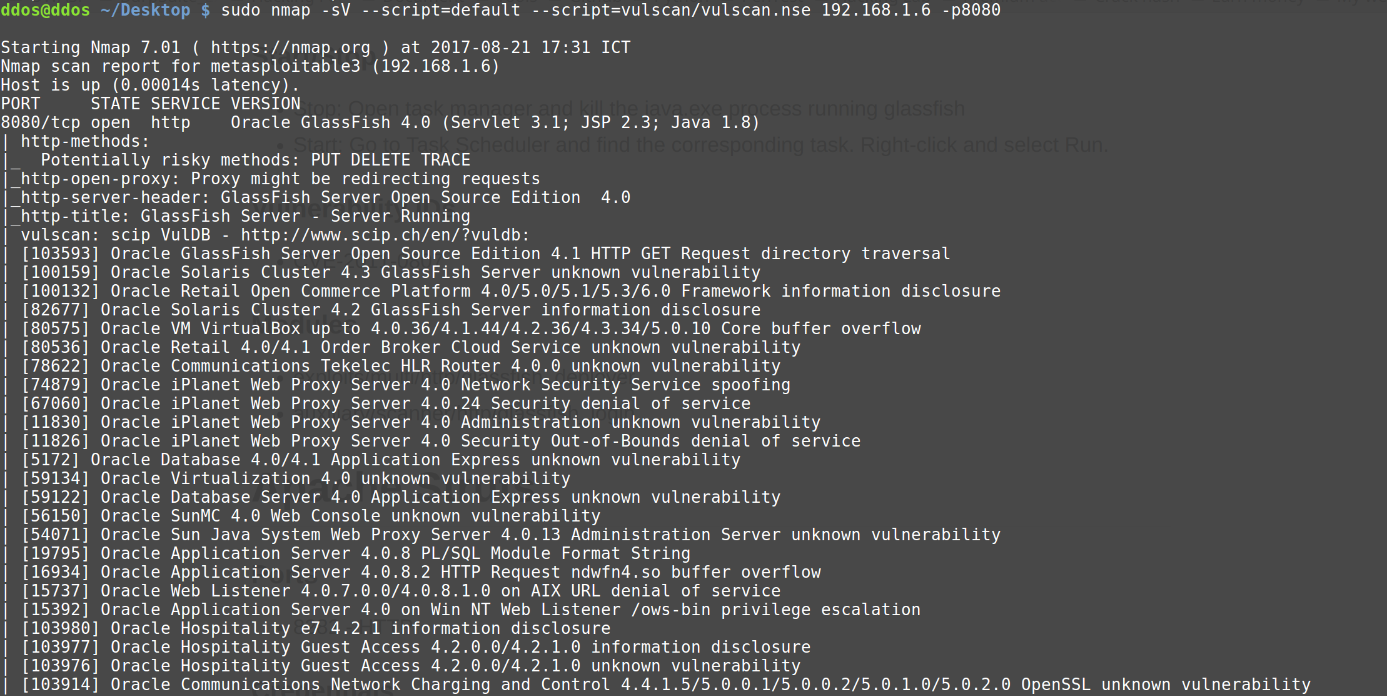


Рисунок 15 – Тестирование DDOS атака

# Заключение

В ходе курсовой работы были выполнены работы по проектированию и разработке веб-приложения.

Для реализации веб-приложения был проведен анализ предметной области, составлено техническое задание, в котором определялись аппаратные и технические требования к разработке, была выполнена разработка проектной документации, в том числе подготовлены макеты и эскизы веб-приложения, реализован подбор графических материалов, после чего была выполнена разработка веб-приложения, включающая в себя реализацию шаблонов и динамических элементов веб-приложения и выполнен контрольный пример и были **реализованы следующие фукнции**:

* просмотр товаров;
* добавления товара;
* удаление товара;
* редактирование товара администратором;
* добавления товара администратором;
* просмотр статистики;
* автоматизировать обработку запросов;
* осуществлять поиск и сбор данных;

В ходе курсовой работы были решены следующие задачи:

* определена и разработана логическая структура будущего веб-приложения, что включало в себя определение основных функциональных модулей, спецификацию их задач и взаимодействия между собой, также необходимо продумать, как пользователи будут взаимодействовать с системой;
* для удобства использования веб-приложения необходимо разработать дизайн пользовательского интерфейса. Это включает в себя выбор цветовой палитры, шрифтов, структуры страниц, расположения элементов управления и визуального оформления. Для реализации данного пункта были разработаны эскизы и макеты будущего веб-приложения;
* после разработки дизайна было решено перейти к верстке страниц приложения, что включало в себя создание HTML, CSS, JS, Bootstrap кода для каждой страницы, определенной в логике приложения. Важно было обеспечить адаптивность и кросс-браузерность, чтобы приложение корректно отображалось на различных устройствах и в разных браузерах, что и было достигнуто;
* была произведена разработка архитектуры базы данных для разрабатываемой серверной части, включая выбор подходящих таблиц;
* завершив этап разработки архитектуры базы данных, был совершён переход к созданию серверной части. Этот этап включает в себя подключение библиотеки ReadBeanPHP (ORM) и написания API для веб-приложения.

Одним из ключевых достоинств является его способность к масштабированию. Была разработана такая архитектура, которая позволяет легко адаптировать приложение под потребности различных регионов в России.

Подводя итоги выпускной квалификационной работы, можно сделать вывод, что поставленная цель исследования была достигнута, качество обработки клиентских обращений стало выше, путем разработки и внедрения на хостинг в декабре 2023 года.

# Список использованных источников

1 Алексеев Ю.М. Быстро и легко создаем, программируем, шлифуем и раскручиваем web-сайт: учеб. пособие / Ю.М. Алексеев. - М.: Лучшие книги, 2011. - 189 с.

2 Ахромов Я.В. Технологии Web-дизайна и Flash-технологии: учеб. пособие / Я.В. Ахромов. - Ростов н/Д: Феникс, 2009. - 265 с.

3 Басыров Р. 1С-Битрикс: постройте профессиональный сайт сами / Р. Басыров. - СПб.: Питер, 2010. - 237 с.

4 Белунцов В. Новейший самоучитель по созданию Web-страниц / В. Белунцов. - М.: NT Press, 2012. - 185 с.

5 Берд Дж. Веб-дизайн. Руководство разработчика: учебник. - СПб.: Питер, 2012. - 224 с.

6 Благодатских В.А. Стандартизация разработки программных средств: учеб. пособие / В.А. Благодатских. - М.: Финансы и статистика, 2011. - 544 с.

7 Бородаев Д.В. Web-сайт как объект графического дизайна: монография / Д.В. Бородаев. - Харьков: «Септима ЛТД», 2011. - 288 с.

8 Браун Брэдли Oracle Database. Создание Web-приложений: учебник / Брэдли Браун, Ричард Ниемик, Джозеф С. Треззо. - СПб.: Лори, 2011. - 722 с.

9 Смирнова И. Начала web-дизайна: учеб. пособие / И. Смирнова. - СПб.: БХВ-Петербург, 2009. - 231 с.

10 Советов Б.Я. Информационные технологии: учебник для ВУЗов / Б.Я. Советов, В.В. Цехановский. - М.: Высшая школа, 2010. - 263 с.

11 Стогний А.А. Основы компьютерной грамотности: учебник / А.А. Стогний. - Киев: Головное издательство, 2009. - 215 с.

12 Сырых Ю.А. Современный веб-дизайн. Рисуем сайт, который продает: учеб. пособие / Ю.А. Сырых. - М.: Вильямс, 2009. - 304 с.

13 Тихонов А.И. Публикация данных в Internet: учебник / А.И. Тихонов. - М.: Изд-во МЭИ, 2011. - 96 с.

14 Трофимов В.В. Информационные системы и технологии в экономике и управлении: учебник / В.В. Трофимов. - М.: Высшее образование, 2010. - 480 с.

15 Угринович Н.Д. Практикум по информатике и информационным технологиям / Н.Д. Угринович.- М.: Бином. Лаборатория занятий, 2012. - 400 с.

16 Программная среда для веб-разработки. OpenServer.: [Электронный ресурс], 2022 года – Режим доступа: https://ospanel.io/ – 05.06.2022.

17 Денвер - локальный сервер. DENVER.: [Электронный ресурс], 2022 года – Режим доступа: http://www.denwer.ru/ – 01.06.2022.

18 Когда display:none, а когда visibility: hiddenhtmlbook.ru.: [Электронный ресурс], 2022 года – Режим доступа: http://htmlbook.ru/ – 25.04.2022.

19 HTML и Web дизайн для начинающих: [Электронный ресурс], 2022 года – Режим доступа: http://sovet.h1.ru – 29.04.2022.

20 Создание шаблона сайта с помощью CSS: [Электронный ресурс], 2022 года – Режим доступа: http://ruseller.com – 21.04.2022.

21 Создание страниц сайта с помощью HTML: [Электронный ресурс], 2022 года – Режим доступа: http://roumik.ru – 21.04.2022.

22 Магия творчества. История вышивания: [Электронный ресурс], 2022 года – Режим доступа: http://myhobbi.net – 21.04.2022.

23 Архитектурная мастерская. Создание сайта: [Электронный ресурс], 2022 года – Режим доступа: http://wonderfulstyle.ucoz.com – 21.04.2022.

24 Создание сайтов: HTML: [Электронный ресурс], 2022 года – Режим доступа: http://www.codeharmony.ru – 21.04.2022.

25 Основы CSS и HTML: [Электронный ресурс], 2022 года – Режим доступа: http://www.web-lesson.ru – 21.04.2022.

**Приложение А**

**Информационная модель**

***(обязательное)***

Рисунок 16 – ER-диаграмма

**Приложение Б**

**Функциональная модель**

***(обязательное)***

Рисунок 17 – Диаграмма контекстная диаграмма IDEF0

Рисунок 18 – Диаграмма декомпозиции первого уровня

**Приложение В**

**Диаграмма прецендентов**

***(обязательное)***

Рисунок 19 – Диаграмма претендентов чат-бота

Рисунок 20 – Диаграмма претендентов сканер документов

**Приложение Г**

**Дизайн сайта**

***(обязательное)***

Рисунок 21 – Шаблон страницы главной страницы

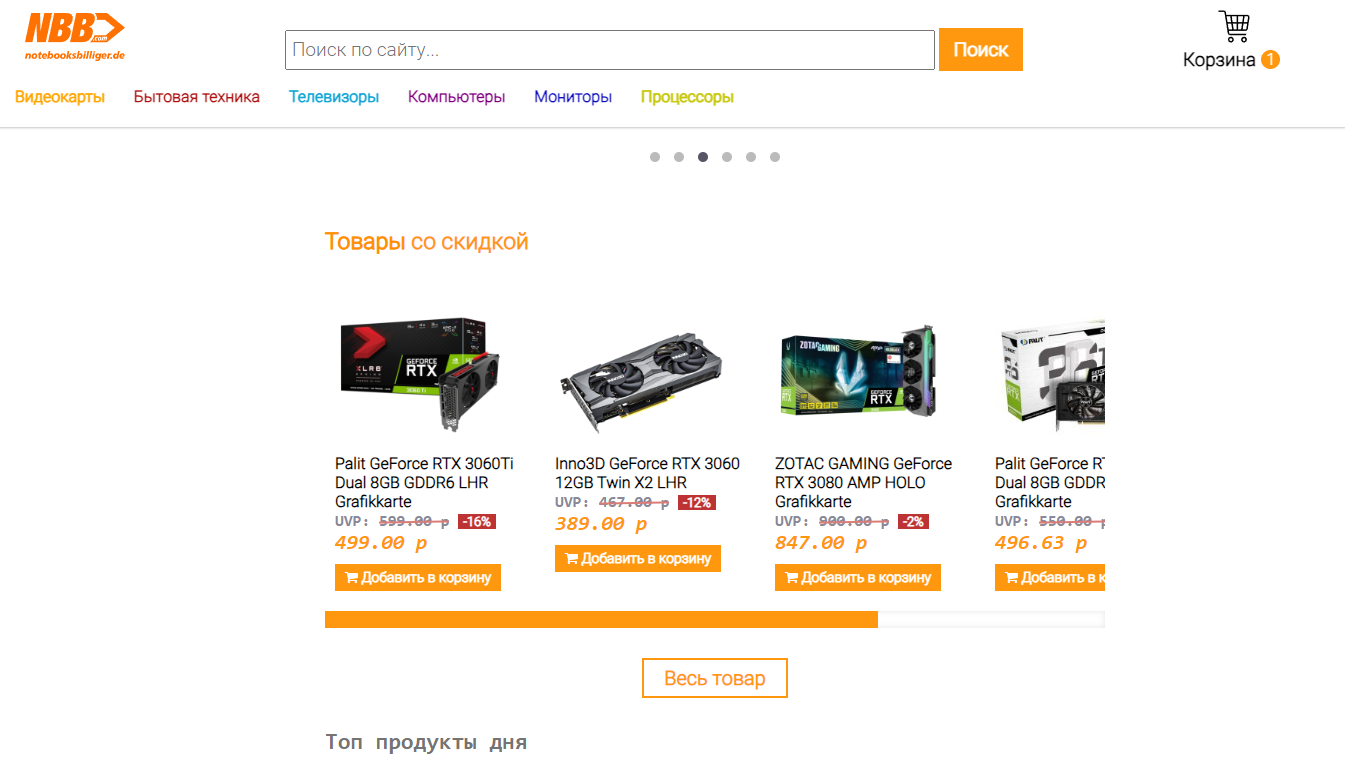


Рисунок 22 – Шаблон каталога

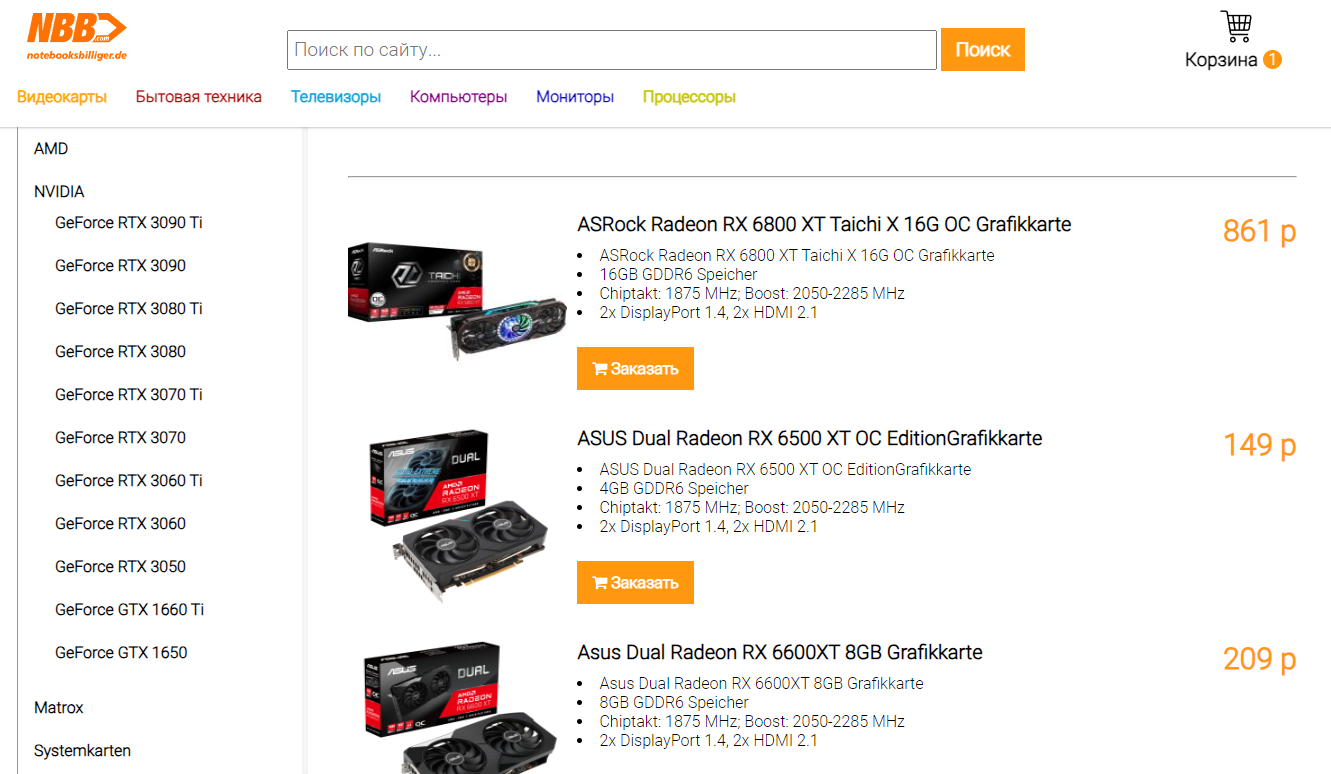


Рисунок 23 – Шаблон административной панели

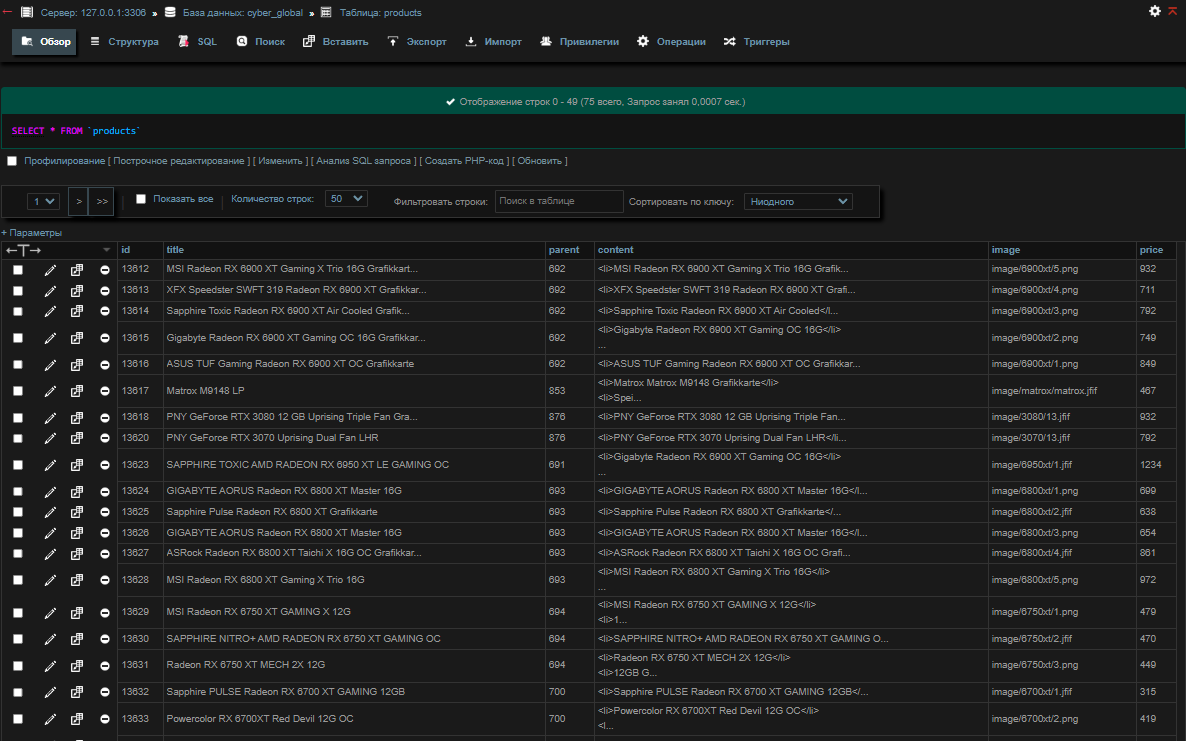


Рисунок 24 – Схема перехода между страницами

**Приложение Д**

**Листинг программы**

***(обязательное)***

<?php

function print\_arr($array){

echo "<pre>" . print\_r($array, true) . "</pre>";

}

function get\_cat(){

global $connection;

$query = "SELECT \* FROM categories";

$res = mysqli\_query($connection, $query);

$arr\_cat = array();

while($row = mysqli\_fetch\_assoc($res)){

$arr\_cat[$row['id']] = $row;

}

return $arr\_cat;

}

function map\_tree($dataset) {

$tree = array();

foreach ($dataset as $id=>&$node) {

if (!$node['parent']){

$tree[$id] = &$node;

}else{

$dataset[$node['parent']]['childs'][$id] = &$node;

}

}

return $tree;

}

function categories\_to\_string($data){

foreach($data as $item){

$string .= categories\_to\_template($item);

}

return $string;

}

function categories\_to\_template($category){

ob\_start();

include 'category\_template.php';

return ob\_get\_clean();

}

function breadcrumbs($array, $id){

if(!$id) return false;

$count = count($array);

$breadcrumbs\_array = array();

for($i = 0; $i < $count; $i++){

if($array[$id]){

$breadcrumbs\_array[$array[$id]['id']] = $array[$id]['title'];

$id = $array[$id]['parent'];

}else break;

}

return array\_reverse($breadcrumbs\_array, true);

}

function cats\_id($array, $id){

if(!$id) return false;

foreach($array as $item){

if($item['parent'] == $id){

$data .= $item['id'] . ",";

$data .= cats\_id($array, $item['id']);

}

}

return $data;

}

function get\_products($ids = false){

global $connection;

if($ids){

$query = "SELECT \* FROM products WHERE parent IN($ids) ORDER BY title";

}else{

$query = "SELECT \* FROM products ORDER BY title";

}

$res = mysqli\_query($connection, $query);

$products = array();

while($row = mysqli\_fetch\_assoc($res)){

$products[] = $row;

}

return $products;

} ?>

<?php

include 'config.php';

include 'functions.php';

$categories = get\_cat();

$categories\_tree = map\_tree($categories);

$categories\_menu = categories\_to\_string($categories\_tree);

if(isset($\_GET['category'])){

$id = (int)$\_GET['category'];

$breadcrumbs\_array = breadcrumbs($categories, $id);

if($breadcrumbs\_array){

$breadcrumbs = "<a href='/catalog/'>Heimat</a> / ";

foreach($breadcrumbs\_array as $id => $title){

$breadcrumbs .= "<a href='?category={$id}'>{$title}</a> / ";

}

$breadcrumbs = rtrim($breadcrumbs, " / ");

$breadcrumbs = preg\_replace("#(.+)?<a.+>(.+)</a>$#", "$1$2", $breadcrumbs);

}else{

$breadcrumbs = "<a href='/catalog/'>Heimat</a> / catalog";

}

$ids = cats\_id($categories, $id);

$ids = !$ids ? $id : rtrim($ids, ",");

if($ids) $products = get\_products($ids);

else $products = null;

}else{

$products = get\_products();

} ?>

function Sim(sldrId) {

let id = document.getElementById(sldrId);

if(id) {

this.sldrRoot = id

}

else {

this.sldrRoot = document.querySelector('.sim-slider')

};

// Slider objects

this.sldrList = this.sldrRoot.querySelector('.sim-slider-list');

this.sldrElements = this.sldrList.querySelectorAll('.sim-slider-element');

this.sldrElemFirst = this.sldrList.querySelector('.sim-slider-element');

this.leftArrow = this.sldrRoot.querySelector('div.sim-slider-arrow-left');

this.rightArrow = this.sldrRoot.querySelector('div.sim-slider-arrow-right');

this.indicatorDots = this.sldrRoot.querySelector('div.sim-slider-dots');

// Initialization

this.options = Sim.defaults;

Sim.initialize(this)

};

Sim.defaults = {

// Default options for the slider

loop: true, // Бесконечное зацикливание слайдера

auto: true, // Автоматическое пролистывание

interval: 5000, // Интервал между пролистыванием элементов (мс)

arrows: true, // Пролистывание стрелками

dots: true // Индикаторные точки

};

Sim.prototype.elemPrev = function(num) {

num = num || 1;

let prevElement = this.currentElement;

this.currentElement -= num;

if(this.currentElement < 0) this.currentElement = this.elemCount-1;

if(!this.options.loop) {

if(this.currentElement == 0) {

this.leftArrow.style.display = 'none'

};

this.rightArrow.style.display = 'block'

};

this.sldrElements[this.currentElement].style.opacity = '1';

this.sldrElements[prevElement].style.opacity = '0';

if(this.options.dots) {

this.dotOn(prevElement); this.dotOff(this.currentElement)

}

};

Sim.prototype.elemNext = function(num) {

num = num || 1;

let prevElement = this.currentElement;

this.currentElement += num;

if(this.currentElement >= this.elemCount) this.currentElement = 0;

if(!this.options.loop) {

if(this.currentElement == this.elemCount-1) {

this.rightArrow.style.display = 'none'

};

this.leftArrow.style.display = 'block'

};

this.sldrElements[this.currentElement].style.opacity = '1';

this.sldrElements[prevElement].style.opacity = '0';

if(this.options.dots) {

this.dotOn(prevElement); this.dotOff(this.currentElement)

}

};

Sim.prototype.dotOn = function(num) {

this.indicatorDotsAll[num].style.cssText =

'background-color:#BBB; cursor:pointer;'

};

Sim.prototype.dotOff = function(num) {

this.indicatorDotsAll[num].style.cssText =

'background-color:#556; cursor:default;'

};

Sim.initialize = function(that) {

// Constants

that.elemCount = that.sldrElements.length; // Количество элементов

// Variables

that.currentElement = 0;

let bgTime = getTime();

// Functions

function getTime() {

return new Date().getTime();

};

function setAutoScroll() {

that.autoScroll = setInterval(function() {

let fnTime = getTime();

if(fnTime - bgTime + 10 > that.options.interval) {

bgTime = fnTime; that.elemNext()

}

}, that.options.interval)

};

// Start initialization

if(that.elemCount <= 1) { // Отключить навигацию

that.options.auto = false;

that.options.arrows = false; that.options.dots = false;

that.leftArrow.style.display = 'none';

that.rightArrow.style.display = 'none'

};

if(that.elemCount >= 1) { // показать первый элемент

that.sldrElemFirst.style.opacity = '1';

};

if(!that.options.loop) {

that.leftArrow.style.display = 'none'; // отключить левую стрелку

that.options.auto = false; // отключить автопркрутку

}

else if(that.options.auto) { // инициализация автопрокруки

setAutoScroll();

// Остановка прокрутки при наведении мыши на элемент

that.sldrList.addEventListener('mouseenter', function() {

clearInterval(that.autoScroll)

}, false);

that.sldrList.addEventListener('mouseleave', setAutoScroll, false)

};

if(that.options.arrows) { // инициализация стрелок

that.leftArrow.addEventListener('click', function() {

let fnTime = getTime();

if(fnTime - bgTime > 1000) {

bgTime = fnTime; that.elemPrev()

}

}, false);

that.rightArrow.addEventListener('click', function() {

let fnTime = getTime();

if(fnTime - bgTime > 1000) {

bgTime = fnTime; that.elemNext()

}

}, false)

}

else {

that.leftArrow.style.display = 'none';

that.rightArrow.style.display = 'none'

};

if(that.options.dots) { // инициализация индикаторных точек

let sum = '', diffNum;

for(let i=0; i<that.elemCount; i++) {

sum += '<span class="sim-dot"></span>'

};

that.indicatorDots.innerHTML = sum;

that.indicatorDotsAll =

that.sldrRoot.querySelectorAll('span.sim-dot');

// Назначаем точкам обработчик события 'click'

for(let n=0; n<that.elemCount; n++) {

that.indicatorDotsAll[n].addEventListener('click', function(){

diffNum = Math.abs(n - that.currentElement);

if(n < that.currentElement) {

bgTime = getTime(); that.elemPrev(diffNum)

}

else if(n > that.currentElement) {

bgTime = getTime(); that.elemNext(diffNum)

}

// Если n == that.currentElement ничего не делаем

}, false)

};

that.dotOff(0); // точка[0] выключена, остальные включены

for(let i=1; i<that.elemCount; i++) {

that.dotOn(i)

}

}

};