

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математического обеспечения
вычислительных комплексов и информационных систем

**Разработка клиент-серверного приложения на основе Node.js и OZON
API**

АФТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы
направления 02.03.03 – Математическое обеспечение и администрирование
информационных систем (МОиАИС)
факультета компьютерных наук и информационных технологий (КНиИТ)
Вартаняна Готти Вартгесовича

Научный руководитель

д. ф.-м. н., профессор

_____ Д.К. Андрейченко

подпись, дата

Зав. кафедрой

д. ф.-м. н., профессор

_____ Д.К. Андрейченко

подпись, дата

Саратов 2020

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ПОСТАНОВКА ЗАДАЧИ	5
2 РЕАЛИЗАЦИЯ WEB-ПРИЛОЖЕНИЯ НА ОСНОВЕ OZON API С ИСПОЛЬЗОВАНИЕМ NODE.JS.....	7
2.1 ОБЩИЕ СВЕДЕНИЯ ОБ OZON API.....	7
2.2 ОПРЕДЕЛЕНИЕ NODE	7
2.3 СТРУКТУРА И АЛГОРИТМ WEB-ПРИЛОЖЕНИЯ	8
2.4 РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ НА NODE.JS.....	9
ЗАКЛЮЧЕНИЕ.....	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	14

ВВЕДЕНИЕ

В настоящее время мы сталкиваемся с необходимостью обработки все большего и большего потока информации, в том числе и в области электронной торговли. Растущие потоки, объемы и скорость поступления, генерирования, хранения и обработки информации обязывают использовать вычислительные системы различной мощности (от сотовых телефонов до кластеров высокопроизводительных вычислений).

Связанные между собой компьютеры в Американской локальной сети Arpanet стали прототипом глобальную сеть Internet. Сейчас Internet объединяет и позволяет управлять значительным числом современных устройств, от бытовой техники и сотовых телефонов и до высокопроизводительных вычислительных систем. На этой платформе по всему миру постоянно функционируют миллионы коммерческих организаций, начиная от международных торговых площадок, заканчивая лотками выездной торговли. Ежедневно миллионы сотрудников и покупателей этих организаций нуждаются в оперативной информационной поддержке. Современная потребность бизнеса заключается в максимальной автоматизации Internet-продаж. В этом случае наиболее эффективно применение распределенного клиент-серверного Web-приложения, у которого разработанная на основе Node.js высоконагруженная серверная часть функционирует на достаточно мощной параллельной вычислительной системе, а разработанные на JavaScript клиентские части – на относительно недорогих устройствах в сети Internet. Дальнейшая эффективность разработанного ПО определяется использованием API той или иной имеющейся системы Internet-торговли (англ. Marketplace). В Российской Федерации традиционно используются сервисы и API Marketplace ozon.ru. Вместе с тем, к системе Internet-торговли одновременно обращается большое количество покупателей и продавцов, и для успешной конкуренции требуется интеллектуальное управление ценой товара и другими аспектами процесса торговли.

Таким образом, сейчас актуальна разработка клиент-серверного приложения для электронной торговли на Marketplace ozon.ru с интеллектуальным управлением ценой товара и другими аспектами процесса торговли.

1 ПОСТАНОВКА ЗАДАЧИ

Объект исследования – автоматизация процесса продаж с помощью использования OZON API.

Предмет исследования – создание клиент-серверного приложения на Node.js с использованием OZON API.

Цель дипломной работы – разработать сервис, способный автоматизированно конкурировать на маркетплейсе OZON, тем самым повышая позицию в выдаче товаров и сами продажи.

К данному сервису предъявляются следующие требования:

1. Решение должно быть достаточно общим, чтобы каждый мог его использовать.
2. Настройка должна быть простой и интуитивно понятной для пользователя.
2. Настройку должна иметь возможность учета поступления товаров из-за рубежа
3. При наличии конкурентов, но в пределах целесообразного, сервис должен держать самую низкую цену.
4. При отсутствии конкурентов сервис должен автоматически повышать цену в пределах целесообразного.
5. Сервис должен позволять задать ценовой диапазон, в частности, минимальную и максимальную цену.
6. Сервис должен показывать рентабельность на текущий момент с учетом комиссии OZON

Для достижения поставленной цели требуется решить следующие задачи:

1. Изучить основные технологии разработки приложений, реализующих Web-сервисы.
2. Произвести анализ языков программирования и выбрать язык программирования для разработки.

3. Произвести анализ баз данных и выбрать базу данных для разработки.
4. Разработать алгоритм и структуру веб-приложения.
5. Реализовать web-сервис и соответствующее серверное приложение.

Таким образом, указаны актуальность данной дипломной работы, а также предмет и объект исследования, сформулирована основная цель дипломного проекта.

2 РЕАЛИЗАЦИЯ WEB-ПРИЛОЖЕНИЯ НА ОСНОВЕ OZON API С ИСПОЛЬЗОВАНИЕМ NODE.JS

OZON.ru – российская торговая площадка, предоставляющая своим покупателям более 5 млн позиций в 24 категориях, среди которых: электроника, бытовая техника, товары для дома и сада, товары для мам и детей, ремонт, спорт и отдых, красота и здоровье, одежда и обувь, автотовары, зоотовары, продукты питания, книги, мультимедиа, DVD, программное обеспечение, игры, музыка, антиквариат и другие. На площадке разработан API-интерфейс, позволяющий управлять интернет-магазином.

2.1 Общие сведения об OZON API

Основные категории методов Seller API:

- товары,
- отправления,
- чат с покупателями,
- отчеты.

Запросы выполняются по протоколу HTTP методами POST или GET. Входные и выходные структуры данных передаются в теле запроса и ответа. API поддерживает формат взаимодействия JSON. В ответе в HTTP-заголовке передается идентификатор связки запрос-ответ — Trace-ID.

2.2 Определение NODE

NODE – это новая программная платформа для разработки быстрых веб-приложений, веб-серверов, любых серверов и клиентов в принципе.

Достоинством Node.JS является и использование одного языка программирования и на сервере, и на клиенте. У такого подхода есть очевидные преимущества

1) одни и те же программисты могут работать над обеими сторонами приложения (и серверной, и клиентской) – хотя ранее было разделение разработчиков, например, на PHP и JS.

2) листинги легко портируются с клиента на сервер и в обратном направлении;

3) один универсальный формат данных (JSON);

4) общие средства написания кода, отладки и тестирования.

2.3 Структура и алгоритм Web-приложения

Клиент-серверное приложение TOPSE11ER AUTOREPRICER на Node.js с использованием OZON Merchants API написано для продавцов, которые продают свои товары на маркетплейсе OZON. Приложение позволяет автоматизировать конкуренцию с другими продавцами на маркетплейсе.

Площадка OZON позволяет размещать на своей программной платформе свой собственный магазин любой организации/любому юрлицу. Для этого необходимо лишь пройти регистрацию. Однако продажи при этом не гарантируются на высоком уровне из-за не самой выгодной цены. Покупатели всегда стремятся найти аналогичный товар дешевле и покупают зачастую именно в той торговой точке, где это дешевле. Отследить цену конкурентов продавцам не затруднительно, когда ассортимент составляет, например, до десяти позиций. Но если ассортимент исчисляется сотнями и тысячами товаров, проследить изменение цен и наличие более выгодных для покупателя вариантов не представляется возможным, используя лишь ручной человеческий труд. С этой целью удобно использовать API, которое предоставляет площадка Ozon. С помощью API можно автоматизировать следующие процессы:

1) поиск аналогичных товаров у конкурентов;

2) мониторинг снижения цен у конкурентов;

3) снижение цены в нашем магазине до такой цены, чтобы она была дешевле, например, на 1 рубль.

Такой подход позволит находиться всегда на BuyBox и продаваться на рынке. BuyBox – это лучшее предложение по цене для конкретного товара.

Продавец, чья цена ниже, будет отображаться на кнопке «добавить в корзину», и, следовательно, количество продаж у него будет больше.

Продавцы, которые конкурируют между собой, часто снижают цену на 1 рубль. Но через определенное время и вторая компания снижает цену на 1 рубль. И эта борьба может продолжаться очень долго. В случае, если количество товаров у продавца составляет несколько сотен единиц, отслеживать конкуренцию на конкурентных карточках товара становится очень затратным по времени и если ли вообще в принципе возможно вручную при таком количестве позиций товара.

Маркетплейсы учитывают множество факторов при распределении процента нахождения продавцов на BUY BOX (кнопке «купить»), этот алгоритм не раскрывается. Однако всегда основным фактором является цена. С ростом количества поставщиков отслеживать цены в ручном режиме становится сложнее и на это уходит много времени. Сервис TOPSE11ER авторепрайсер поможет автоматизировать процесс торга с конкурентами и получить до 40% больше продаж за счет постоянного нахождения товаров на кнопке “купить”.

2.4 Разработка Web-приложения на Node.JS

Приложение находится по адресу: <https://ar.topse11er.ru/>

Доступна регистрация, вход, кроме этого имеются страницы: главная, о сервисе, тарифы, отзывы.

Страница регистрации имеет код:

```
let user = require( '../user' );

module.exports.registerRequest = async function ( req , res ) {

  if (req.body.name && req.body.lastname && req.body.email && req.body.password ) {

    if(req.body.password != req.body.passwordcopy){

      return res.render('register', { error: 'Введенные пароли не совпадают' });
```

```

    }

    if (!req.body.check) {
        return res.render('register', { error: 'Не приняты условия' });
    }

    let user = await req.db.collection( 'user' ).findOne( { email: req.body.email } );

    if ( null !== user ) {
        return res.render( 'register' , { error: 'Пользователь с такой эл. почтой существует' } );
    }

    req.db.collection('user').insertOne({ name: req.body.name, lastname: req.body.lastname, email:req.body.email , password:req.body.password } , function ( error , user ) {

        if ( error ) {

            return res.sendStatus( 500 );

        } else {

            req.session.username = req.body.email;

            req.session.tsUser = req.body.email;

            req.session.userName = user.name + " " + (user.lastname || "");

            req.session.userId = user.ops[ 0 ]._id;

            req.session.validated = true;

            return res.redirect( '/items/' );

        }

    });

} else {

    return res.render( 'register' , { error: 'Проверьте вводимые данные' } );

}

}

```

Здесь идет проверка на корректность заполнений полей регистрационной формы: должны быть непустыми поля: имя, фамилия, email и пароль. Причем пароль должен быть введен дважды и, если пароли не совпадают, пользователь получит сообщение об ошибке. Также должны быть приняты условия

(соответствующая галочка на форме), иначе пользователь получит ошибку «Не приняты условия». В скрипте есть и защита от повторной регистрации. Для этого проверяется наличие в базе пользователей с таким email-адресом. Если нет – регистрация пройдет успешно, иначе – пользователь получит ошибку: «Пользователь с такой эл. почтой существует».

В случае отсутствия ошибок пользователь будет зарегистрирован, произойдет запись в базу данных регистрационных данных.

Код страницы авторизации приведен ниже:

```
let user = require( '../user' );

module.exports.loginRequest = async function ( req , res ) {

  if ( req.body.email && req.body.password ) {

    req.db.collection( 'user' ).findOne( { email: req.body.email , password: req.body.
password } , function ( error , user ) {

      //console.log(req.session)

      if ( error ) {

        return res.sendStatus( 500 );

      } else {

        if ( user == null ) {

          return res.render( 'login' , { error: 'Неверный логин или пароль' } );

        } else {

          req.session.username = req.body.email;

          req.session.userName = user.name;

          req.session.tsUser = req.body.email;

          req.session.userId = user._id;

          req.session.validated = true;

          return res.redirect( '/items/' );

        }

      }

    }

  }

}
```

```
    });  
  } else {  
    return res.render( 'login' , { error: 'Неверный логин или пароль' } );  
  }  
}
```

В этом коде принимается на вход email и пароль из формы авторизации. Если эти параметры не пустые, идет поиск в базе данных соответствующего пользователя, возвращаются его данные: email, логин, id. Если пользователь не найден, он получит сообщение об ошибке: «Неверный логин или пароль».

ЗАКЛЮЧЕНИЕ

В результате данной работы были изучены технологии по созданию web-приложений, в частности, с использованием программной платформы Node.JS и API, предоставляемым торговой площадкой Ozon.ru.

Во время работы над данной дипломной работой были решены следующие задачи:

1. Была изучена предметная область.
2. Спроектирована структура приложения.
3. Создан веб-приложение на Node.js с использованием Ozon API.
4. Разработана база данных MongoDB с использованием модуля Mongoose.

Сервис достаточно прост в установке, настройке и обслуживании.

Разработанное приложение не имеет конкурентов и используется селлерами – продавцами Marketplace OZON.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дунаев Вадим. HTML, скрипты и стили. Спб.: «БХВ-Петербург» – 2008. - 1024с
2. А. Каменнова, А. Громов, М. Ферапонтов. Моделирование бизнеса. Методология ARIS. – М.: Весть-МетаТехнология, 2002. – 333с.
3. Кондратьев В.В., Кузнецов М.Н. Показываем бизнес-процессы от модели процессов компании до регламентов процедур. - М.: Эксмо, 2008. — 256 с.
4. Ллойд Йен. Создай свой веб-сайт с помощью HTML и CSS. - СПб.: Питер, 2013. - 416 с.
5. Локхарт Джош. Современный PHP. Новые возможности и передовой опыт. - М.: ДМК, 2016. - 304 с.
6. Макфарланд Дэвид JavaScript. Подробное руководство. - М.: Эксмо, 2009. - 608 с.
7. Мальчук Е.В. HTML и CSS. Самоучитель. М.: Вильямс –2008. - 416 с.
8. Мержевич Влад. HTML и CSS на примерах. Спб.: Издательство “БХВ-Петербург” – 2005. - 448с.
9. Никсон Роберт. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. СПб.: ООО "Питер", 2019. - 816 с.
10. Пауэрс Дэвид. PHP. Создание динамических страниц. - М.: Аст, 2012. - 640 с.
11. В. В. Репин, В. Г. Елиферов Процессный подход к управлению. Моделирование бизнес-процессов: Издательство Стандарты и качество, 2008. - 408 с.
12. Робсон Элизабет, Фримен Эрик. Изучаем HTML, XHTML и CSS 2-е изд. - М.: Питер, 2013. - 720 с.
13. Скляр Дэвид. Изучаем PHP 7. Руководство по созданию интерактивных веб-сайтов. - М.: Вильямс, 2017. - 464 с.

14. Информационные технологии управления: Учебное пособие для ВУЗов / под ред. Г.А. Титоренко – М.: ЮНИТИ–ДАНА, 2004. – 439с.
15. Холмогоров В. Основы Web-мастерства. Учебный курс. - СПб.: Питер, 2001. - 352 с.
16. Хэррон Д. «Node.js. Разработка серверных веб-приложений в JavaScript»: Пер. с англ. Слинкина А.А. - М.: ДМК Пресс, 2012. - 144с.: ил.
17. Чаффер Д. Изучаем jQuery 1.3. Эффективная веб-разработка на JavaScript. - М.: Символ-плюс, 2010. - 448 с.
18. Шеер А.В. ARIS - моделирование бизнес-процессов. – М.: Вильямс, 2009. - 224с.
19. Шпильман Сью JSTL. Практическое руководство для JSP-программистов. - М.: КУДИЦ-Образ, 2012. - 272 с.
20. Эспозито Дино. Разработка современных веб-приложений. Анализ предметных областей и технологий. - М.: Вильямс, 2017. - 464 с.