МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационные системы и технологии

Специальность 1–40 05 01 Информационные системы и технологии

Специализация 1 –40 01 01 Программное обеспечение информационных

технологий (программирование интернет – изданий)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

«Сеть магазинов по продаже автозапчастей»

Выполнил студент Шуст Юрий Олегович

(Ф.И.О.)

Руководитель проекта асс. Нистюк О.А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Заведующий кафедрой к.т.н., доц. Смелов В.В .

(учен. степень, звание, должность, Ф.И.О., подпись)

Консультант: асс. Нистюк О.А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Нормоконтролер: асс. Нистюк О.А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой

Минск 2021

**Реферат**

Пояснительная записка курсового проекта содержит 29 страниц пояснительной записки, 30 иллюстраций, 6 источников литературы, 2 приложения.

NODE.JS, JAVASCRIPT, MICROSOFT SQL SERVER, EXPRESS.JS, NODE-MSSQL, MSNODESQLV8, BACKUP DATABASE, RESTORE DATABASE

Основная цель курсового проекта: проектирование базы данных для сети магазинов и технология резервного копирования и восстановления базы данных.

В первой главе проводится аналитический обзор литературы по тематике курсового проекта и содержит анализ методов решения поставленных задач.

Вторая глава посвящена процессу проектирования системы.

В третьей главе приведено обоснование технических приёмов.

В четвертой главе описано тестирование, позволяющее подробно понять интерфейс программного средства.

В заключении приведены результаты проделанной работы.

**Abstract**

The explanatory note of the course project contains 29 pages of the explanatory note, 30 illustrations, 6 sources of literature, 2 appendices.

NODE.JS, JAVASCRIPT, MICROSOFT SQL SERVER, EXPRESS.JS, NODE-MSSQL, MSNODESQLV8, BACKUP DATABASE, RESTORE DATABASE

The main goal of the course project: designing a database for a chain of stores and technology for backing up and restoring a database.

The first chapter provides an analytical review of the literature on the subject of the course project and contains an analysis of methods for solving the tasks.

The second chapter is devoted to the system design process.

The third chapter provides the rationale for technical techniques.

The fourth chapter describes testing, which allows you to understand in detail the interface of a software tool.

In conclusion, the results of the work done are presented.

**Содержание**

[Введение 5](#_Toc90374581)

[1 Аналитический обзор литературы 6](#_Toc90374582)

[2 Разработка программной системы 8](#_Toc90374583)

[2.1 Разработка модели базы данных 8](#_Toc90374584)

[2.2 Хранимые процедуры 10](#_Toc90374585)

[2.2.1 Выборка данных для таблиц 10](#_Toc90374586)

[2.2.2 Выборка данных по поисковому запросу 10](#_Toc90374587)

[2.2.3 Заполнение таблиц 100 000 строк 10](#_Toc90374588)

[2.2.4 Вход администратора в аккаунт 11](#_Toc90374589)

[2.2.5 Удаление данных из таблиц 11](#_Toc90374590)

[2.2.6 Добавление данных в таблицы 11](#_Toc90374591)

[2.2.7 Изменение данных в таблицах 11](#_Toc90374592)

[2.2.8 Экспорт и импорт таблицы PRODUCTS в формате XML 12](#_Toc90374593)

[2.2.9 Резервное копирование и восстановление 13](#_Toc90374594)

[2.3 Индексы 14](#_Toc90374595)

[3 Обоснование технических приемов программирования 15](#_Toc90374596)

[4 Тестирование, проверка работоспособности и анализ данных 17](#_Toc90374597)

[Заключение 26](#_Toc90374598)

[Список литературы 27](#_Toc90374599)

[Приложение А 28](#_Toc90374600)

[Приложение Б 29](#_Toc90374601)

# **Введение**

Целью данной работы была разработка реляционной базы данных на тему «Сеть магазинов по продаже автозапчастей». База данных должна быть составлена для работы торговых сетей. Так же было необходимо разработать соответствующее приложение, для демонстрации её работы.

База данных — это организованная структура, предназначенная для хранения информации, систематизированная таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины.

Реляционная база данных — база данных, основанная на реляционной модели данных.

В качестве СУБД для базы данных была выбрана Microsoft SQL Server, в связи с ее простотой, производительностью и надежностью.

Так же было необходимо создать программу для демонстрации работы базы данных. Написанное приложение строится на клиент-серверной архитектуре. В качестве клиента выступает приложение написанное на HTML и JavaScript, которое отображается в браузере. Для сервера был выбрана платформа Node.js, в связи с кроссплатформенностью, простотой развертки сервера и гибкости работы за счёт npm.

Node.js – программная платформа, основанная на движке V8. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API, подключать другие внешние библиотеки. Node.js применяется преимущественно на сервере. В основе Node.js лежит событийно-ориентированное и асинхронное программирование.

Npm – менеджер пакетов, входящий в состав Node.js, позволяющий расширить возможности приложения.

Для работы с Node.js были выбраны следующий пакеты:

* Express.js – предоставляет собой слой фундаментальных функций для упрощения разработки веб-приложений;
* Msnodesqlv8 – TDS драйвер для работы с Microsoft SQL Server, основан на драйвере node-mssql;
* Node-mssql – Node.js драйвер клиента для Microsoft SQL Server;
* Fs – стандартный пакет Node.js для работы с вводом-выводом файловой системы.

# **1 Аналитический обзор литературы**

Раньше, для администрирования магазинов, вели бухгалтерский учет, что осложняло весь процесс. Сейчас же используются базы данных с разработанным для них программным обеспечением, позволяя уменьшить объем бумажных работ и оптимизировать работу всей системы магазинов, попутно увеличивая прибыль компании, использующей современные технологии.

Постепенно с развитием программного обеспечения ЭВМ появились идеи создания управляющих систем, которые позволяли бы накапливать, хранить и обновлять взаимосвязанные данные по целому комплексу решаемых задач, например, при автоматизации бухгалтерского учета на предприятии. Эти идеи нашли свое воплощение в системах управления базами данных (СУБД). СУБД взаимодействуют не с локальными, а взаимосвязанными по информации массивами, называемыми базами данных. С появлением персональных компьютеров СУБД становятся наиболее популярным средством обработки табличной информации. Они являются инструментальным средством проектирования банков данных при обработке больших объемов информации.

Базы данных, написанные для различных сетей супермаркетов, используются для работы за кассовым оборудованием и управления непосредственно базой от лица администраторов магазинов. В открытом доступе рассмотреть напрямую системы управления магазинами невозможно, так они пишутся на заказ. Но можно рассмотреть структуру организации товаров среди онлайн-магазинов по продаже автозапчастей. Одним из таких магазинов является «Auto1.by».

Рассмотрим в качестве примера каталог товаров данного сайта на рисунке 1.1.

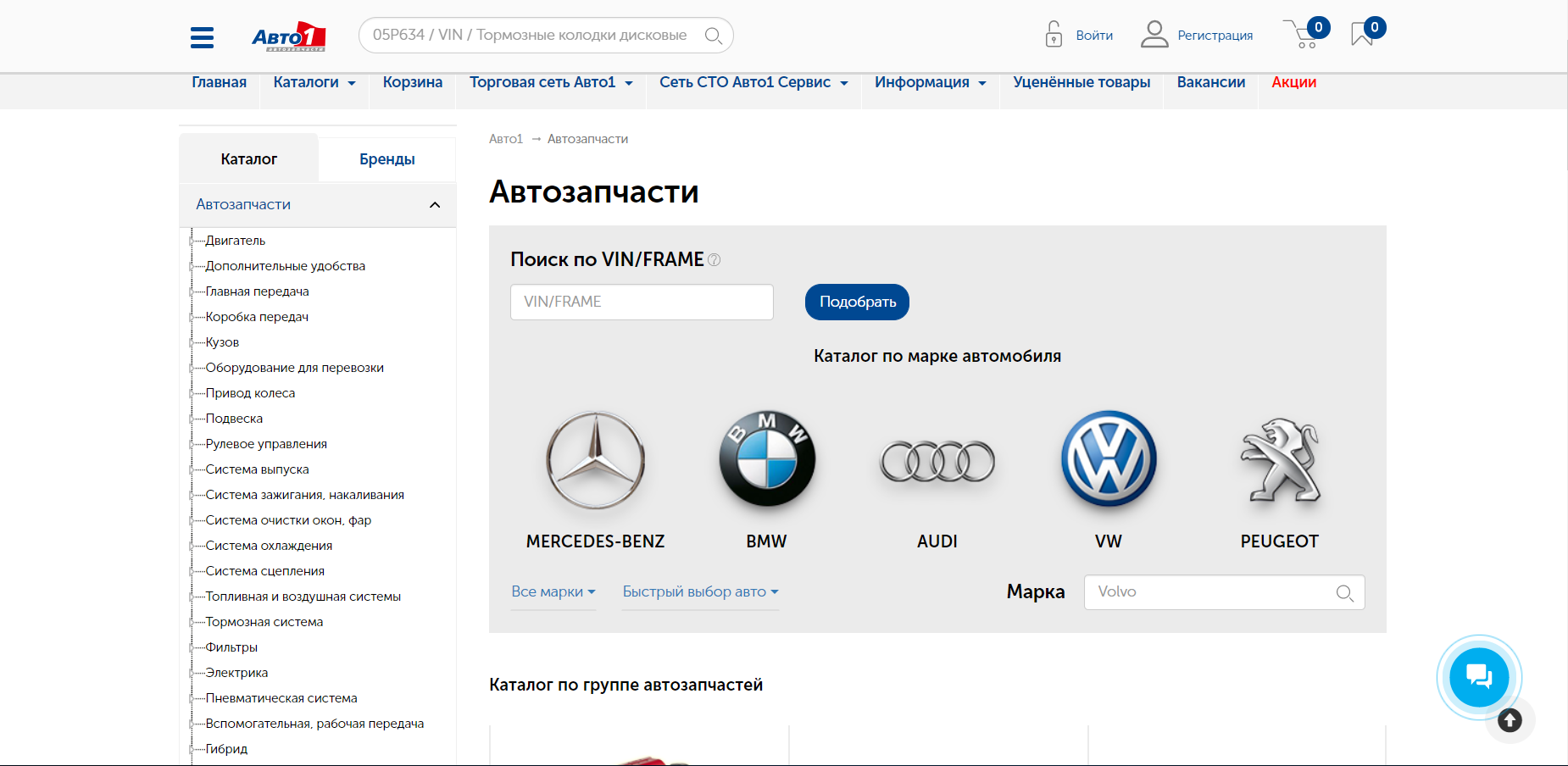


Рисунок 1.1 – Пример каталога «Auto1.by»

Тут структура каталога товаров делится на категории и подкатегории, и типы, а если опираться на структуру реальных магазинов, то это можно описать как деление категории и типы продукции. У товара есть следующие особенности:

* Цена;
* Название;
* Артикул;
* Ссылки на категории и типы к которому относится товар;
* Возможность добавления в корзину.

Так же рассмотрим интернет-магазин автозапчастей «exist.by», одна из страниц из которого представлена на рисунке 1.2.

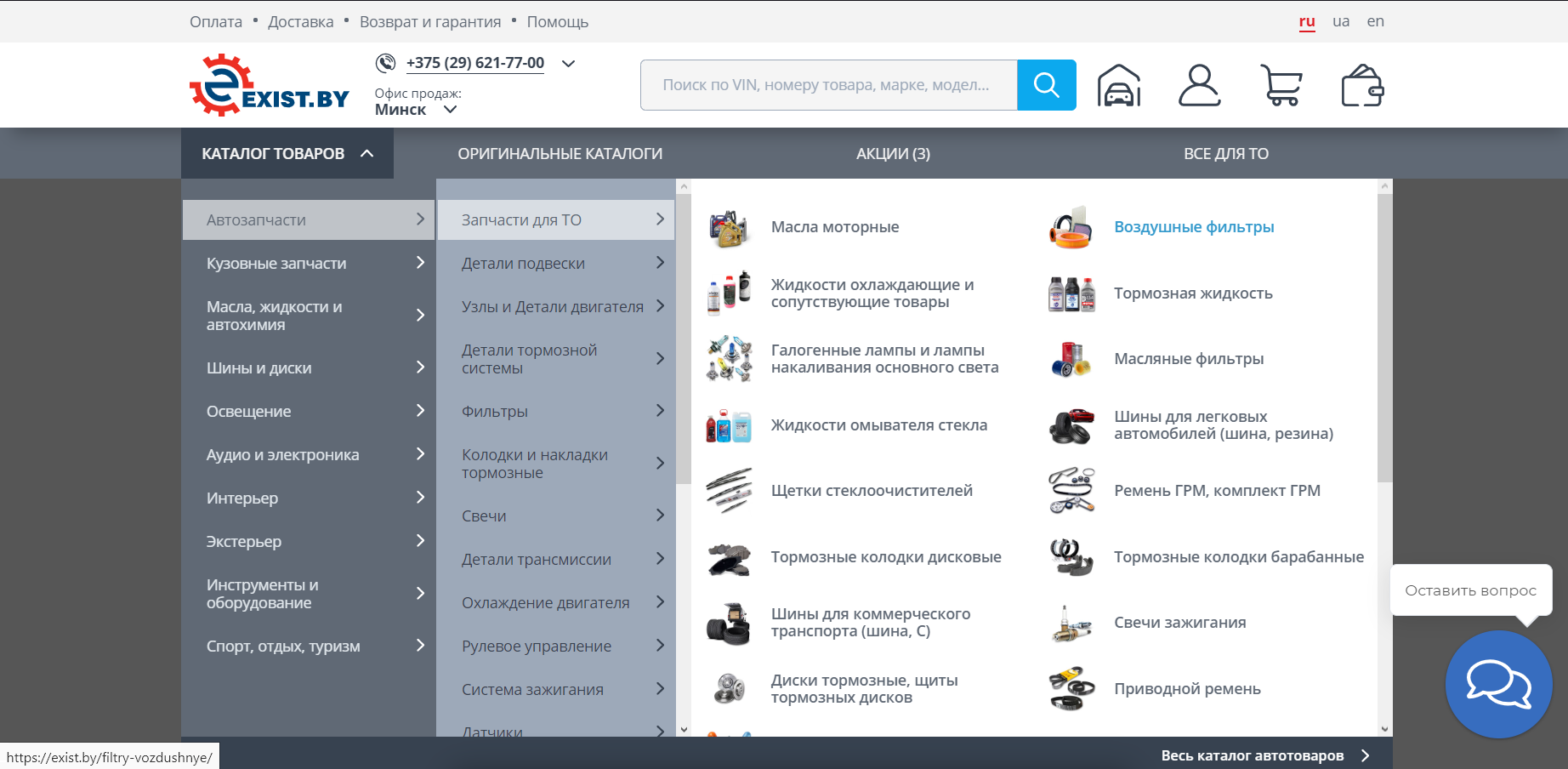


Рисунок 1.2 – Пример каталога «exist.by»

Его основные особенности такие же, как и у «Auto1.by», кроме того, что в характеристику товара так же входит информация о поставщиках данного товара.

Исходя из приведенных данных была разработана база данных, описанная в пункте 2.

# **2 Разработка программной системы**

При разработке курсового проекта понадобились следующие объекты:

1. Таблицы;
2. Хранимые процедуры;
3. Индексы.

## **2.1 Разработка модели базы данных**

Для реализации базы данных для сети магазинов было разработано 12 таблиц. Диаграмма базы со структурой связей представлена в приложении А.

Логически можно вывести 6 основных таблиц: STORES, EMPLOYEES, PRODUCTS, STORE\_ADMIN, USERS и SUPPLIERS.

ТаблицаSTORESпредставляет магазин, в котором осуществляется продажа продукции, состоит из столбцов:

* Id – идентификатор магазина, тип int, первичный ключ;
* storeName – название магазина, тип nvarchar;
* address – адрес магазина, тип nvarchar;

Таблица EMPLOYEES представляет собой людей, работающих в магазинах, состоит из столбцов:

* id – идентификатор работника, тип int, первичный ключ;
* firstName – имя работника, тип nvarchar;
* lastName – фамилия работника, тип nvarcahr;
* sex – пол, тип nchar;
* age – возраст работника, тип int;
* experience – опыт работы работника, тип int;
* idPost – идентификатор должности, тип int, внешний ключ;
* idDep – идентификатор отдела, тип int, внешний ключ;
* idStore – идентификатор магазина, тип int, внешний ключ;

Таблица PRODUCTS представляет собой товары магазина, состоит из столбцов:

* id – идентификатор продукта, тип int, первичный ключ;
* prodCode – код продукта, тип int;
* prodName – наименование продукта, тип int;
* typeProdId – идентификатор категории продукта, тип int, внешний ключ;
* prodPrice – цена за единицу продукта, тип int;
* prodStock – количество товара в наличии, тип int;
* supId – идентификатор поставщика, тип int;

Таблица STORE\_ADMIN представляет собой данные об администраторе, состоит из столбцов:

* id – идентификатор администратора, тип int, первичный ключ;
* adminLogin – логин администратора, тип nvarchar;
* adminPassword – пароль администратора, тип nvarchar;
* adminType – идентификатор типа аккаунта, тип int;
* employeeId – идентификатор работника, который является администратором, тип nvarchar;

Таблица USERS представляет пользователей платформы, состоит из столбцов:

* id – идентификатор пользователя, тип int, первичный ключ;
* login – логин пользователя, тип nvarchar;
* password – пароль пользователя, тип nvarchar;
* email – адрес электронной почты пользователя, тип nvarchar;
* firstName – имя пользователя, тип nvarchar;
* secondName – фамилия пользователя, тип nvarcahr;
* sex – пол, тип nchar;
* age – возраст пользователя, тип int;
* userType – идентификатор типа аккаунта, тип int, внешний ключ;

Таблица SUPPLIERS представляет поставщиков, состоит из столбцов:

* id – идентификатор магазина, тип int, первичный ключ;
* supName – имя поставщика, тип nvarchar;
* supAddress – адрес поставщика, тип nvarchar;

Так же есть еще 7 таблиц для поддержания всей остальной логической структуры базы данных сети магазинов.

Таблица USER\_TYPE представляет тип покупателей магазина, состоит из столбцов:

* id – идентификатор типа пользователя, тип int, первичный ключ;
* userType – наименование типа пользователя, тип nvarchar;

Таблица PRODUCT\_TYPEпредставляет тип товаров в магазине, состоит из столбцов:

* id – идентификатор типа товара, тип int, первичный ключ;
* prodType – наименование типа товара, тип nvarchar;
* idDep – идентификатор отдела к которому относится тип товара, тип int, внешний ключ.

Таблица USER\_BUCKET представляет товары, купленные покупателем, состоит из столбцов:

* prodId – идентификатор товара, тип int, первичный ключ, внешний ключ;
* userId – идентификатор покупателя, тип int, первичный ключ, внешний ключ;
* amount – количество единиц товара в корзине, тип int;

Таблица POSTS представляет должности сотрудников магазина, состоит из столбцов:

* id – идентификатор типа должности, тип int, первичный ключ;
* post – наименование типа должности, тип nvarchar.

Таблица DEPARTMENTSпредставляет отделы магазина, состоит из столбцов:

* id – идентификатор отдела магазина, тип int, первичный ключ;
* depart – наименование отдела магазина, тип nvarchar.

Таблица ADMIN\_TYPEпредставляет тип администратора, состоит из столбцов:

* id – идентификатор типа администратора, тип int, первичный ключ;
* adminType– наименование типа администратора, тип nvarchar.

**2.2 Хранимые процедуры**

Хранимая процедура – объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере.

При разработке курсового проекта было создано 32 процедур для следующих целей:

1. Выборка данных для таблиц;
2. Выборка данных по поисковому запросу;
3. Заполнение таблиц 100 000 строк;
4. Вход администратора в аккаунт;
5. Вход пользователя в аккаунт;
6. Создание новой базы данных и резервное копирование в нее основной базы данных
7. Удаление данных из таблиц;
8. Добавление данных в таблицы;
9. Изменение данных в таблицах;
10. Экспорт и импорт таблицы PRODUCTS в формате xml;
11. Резервное копирование и восстановление базы данных;

### **2.2.1 Выборка данных для таблиц**

Для вывода данных из таблиц были написаны следующие процедуры: OProducts, OprodTypes, OStores, OSuppliers и OUsers. Основной их особенностью является нумерация через row\_number(), необходимое отсечение некоторого диапазона значений. Вывод таблицы размером в 100 000 строк на сайте занимал около 20 секунд, ограничение вывода до 20 строк и создание индексов позволило уменьшить время вывода таблицы до минимума.

### **2.2.2 Выборка данных по поисковому запросу**

Для вывода данных из таблиц по поисковому запросу была написана процедура SearchProd. SearchProd выводит информацию о товаре из таблицы PRODUCTS при вводе штрихкода.

### **2.2.3 Заполнение таблиц 100 000 строк**

Для заполнения таблиц было разработаны процедуры для генерации случайных строк и процедуры заполнения таблиц этими строками.

Процедуры заполнения таблиц: AGSuppliers, AGProducts. Они были разработаны соответственно для заполнения таблиц:SUPPLIERS и PRODUCTS.

Для генерации случайных имен была создана процедура CreateName, для генерации случайного штрихкода – CreateBarcode. Код процедур CreateName и CreateBarcode представлен в приложении Б.

### **2.2.4 Вход администратора в аккаунт**

Процедура logAdmin ищет данные о администраторе по его логину. Процедура возвращает данные приложению, так как со стороны логики приложение проводится сверка пароля и хэша, так как пароли пользователей и администраторов хранятся в зашифрованном виде с помощью алгоритма bcrypt. Данная процедура представлена ниже на рисунке 2.1.

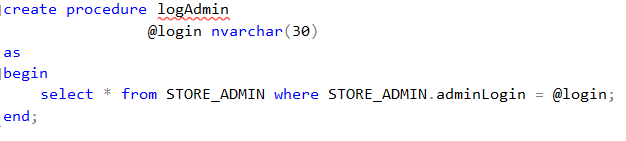


Рисунок 2.1 – Пример создания процедуры logAdmin

### **2.2.5 Удаление данных из таблиц**

Для управления базой данных были созданы процедуры удаления: OStoresDelete, OProductsDelete, OPoststDelete и DeleteFromBucket. Они были разработаны соответственно для удаления строк из таблиц: STORES, PRODUCTS, POSTS, USER\_BUCKET.

### **2.2.6 Добавление данных в таблицы**

Для управления базой данных были созданы процедуры добавления новых строк: OStoresAdd, OProductsAdd, OPostsAdd и AddToBucket. Они были разработаны соответственно для заполнения таблиц: STORES, PRODUCTS, POSTS и USER\_BUCKET.

### **2.2.7 Изменение данных в таблицах**

Для управления базой данных были созданы процедуры обновления: OStoresUpdate, OProductsUpdate и OPostUpdate. Они были разработаны соответственно для обновления данных в таблицах: STORES, PRODUCTS и POSTS.

### **2.2.8 Экспорт и импорт таблицы PRODUCTS в формате XML**

Для экспорта таблицы PRODUCTS в формате xml была разработана процедура ExProdtoXml. Для формирования xml в select запросе используется конструкция FOR XML. Для вывода в файл была использована расширенная хранимая процедура xp\_cmdshell. Процедура ExProdtoXml представлена ниже на рисунке 2.2. Отрывок из xml представлен в приложении В.

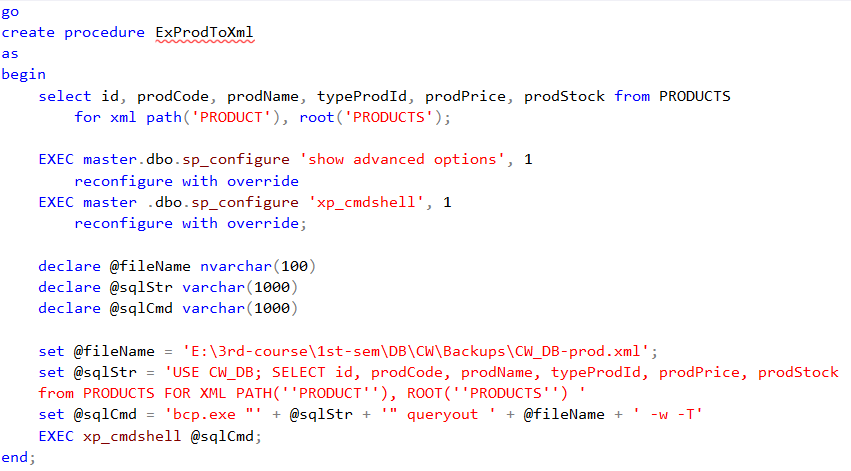


Рисунок 2.2 – Пример создания процедуры ExProdToXml

Для импорта данных в таблицу PRODUCTS, из файла формата xml, была разработана процедура ImProdfromXml. Для получения xml файла и последующего разбора со вставкой используется конструкция FROM OPENROWSET совместно с параметром BULK. Процедура ImProdfromXml представлена ниже на рисунке 2.3.

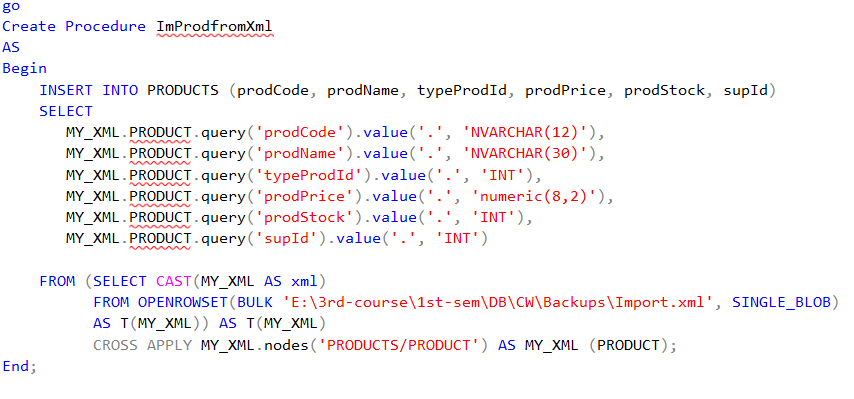


Рисунок 2.3 – Пример создания процедуры ImProdfromXml

### **2.2.9 Резервное копирование и восстановление**

Для копирования базы данных была разработана процедура backupDB. Для этого используется конструкция backup database, которая экспортирует базу данных в файл с расширением bak. Эта процедура представлена ниже на рисунке 2.4.

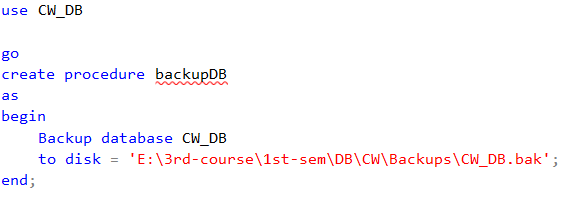


Рисунок 2.4 – Пример создания процедуры backupDB

Для восстановления базы данных была разработана процедура restoreDB. Для этого используется конструкция restore database, которая экспортирует базу данных в файл с расширением bak. Эта процедура представлена ниже на рисунке 2.5.

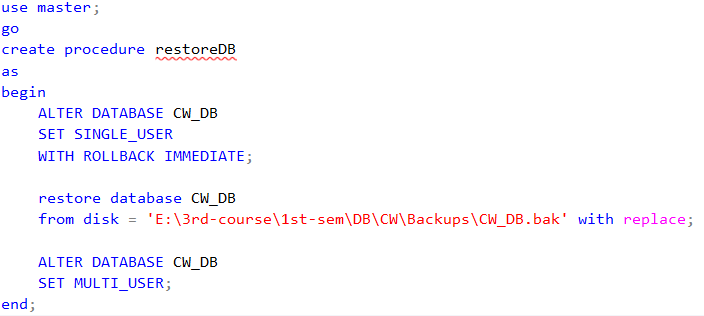


Рисунок 2.5 – Пример создания процедуры restoreDB

Для запуска и исполнения данной процедуры необходимо ее создавать системной базе данных master. Это необходимо для избегания коллизий при восстановлении. Так же восстановление невозможно, при наличии подключенных к базе пользователей, поэтому переводим базу данных в монопольный режим с помощью set single\_user и отменяем все изменения через параметр rollback immediate. После восстановления возвращаем базу данных в прежнее состояние через set multi\_user.

## **2.3 Индексы**

Индекс – объект базы данных, создаваемый с целью повышения производительности поиска данных. Таблицы в базе данных могут иметь большое количество строк, которые хранятся в произвольном порядке, и их поиск по заданному критерию путём последовательного просмотра таблицы строка за строкой может занимать много времени. В связи с необходимостью выборки таблиц, заполненных большим количеством строк, было принято решение разработать 6 индексов для таблиц: PRODUCTS, STORES, SUPPLIERS, DEPARTMENTS, POSTS, PRODUCT\_TYPE.

# **3 Обоснование технических приемов программирования**

При проектировании программного средства в качестве программной платформы для сервера был выбран Node.js с пакетами Express.js, fs, msnodesqlv8 и node-mssql. Непосредственно для взаимодействия с Microsoft SQL Server нам необходимы только пакеты msnodesqlv8 и node-mssql. Схема взаимодействия с Microsoft SQL Server представлена на рисунке 3.1.

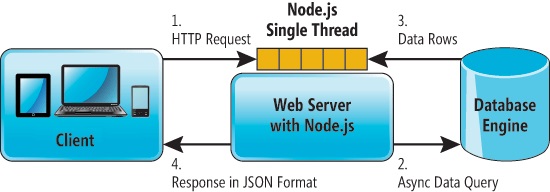


Рисунок 3.1 – Схема взаимодействия с Microsoft SQL Server

Для работы с базой данных используется пакет node-mssql, как клиент базы данных. TDS драйвером выступает msnodesqlv8, позволяя упростить работу с СУБД. Пример конфигурации, необходимой для подключения к Microsoft SQL Server, представлена ниже на рисунке 3.2.

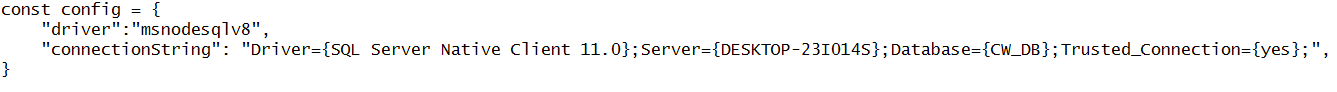


Рисунок 3.2 – Конфигурация подключения к Microsoft SQL Server

Далее необходимо получить доступ к пулу подключений, задействовав ранее представленную конфигурацию. Пример подключения представлен ниже на рисунке 3.3.

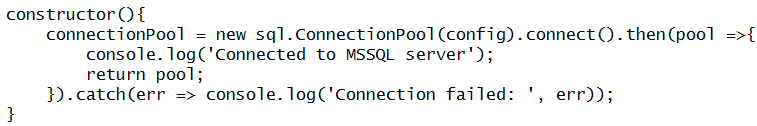


Рисунок 3.3 - Подключение к Microsoft SQL Server

После запроса к базе данных, представленном ниже на рисунке 3.4, данные помещаются в пул и передаются серверу.

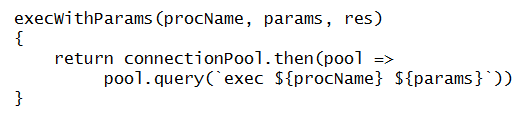


Рисунок 3.4 – Запрос к Microsoft SQL Server

Получив данные, сервер их обрабатывает и, обработав результат в формат JSON, отправляет их клиенту.

# **4 Тестирование, проверка работоспособности и анализ данных**

Протестируем для начала страницу авторизации администратора.

Введём логин и пароль. Страница авторизации продемонстрировано ниже на рисунке 4.1.

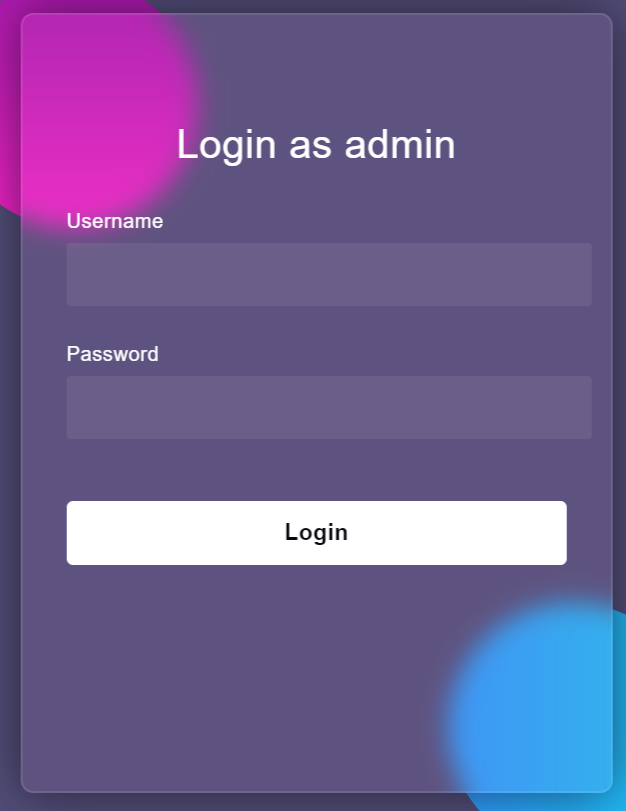


Рисунок 4.1 – Страница авторизации

В случае не верно введенных данных, система вернет пользователя снова на эту же страницу. Если же данные верны, то пользователь перейдет на страницу администратора «AutoLight». На которой расположены элементы управления. Эта страница представлена на рисунке 4.2.



Рисунок 4.2 – Страница «AutoLight» администратора

На верхней панели управления страницы «AutoLight» слева направо расположены кнопки для:

1. Просмотр пользователей;
2. Просмотр категорий товаров;
3. Просмотр товаров, удаления, добавления и изменения товаров;
4. Просмотр должностей, удаления, добавления и изменения должностей;
5. Просмотр магазинов, удаления, добавления и изменения магазинов;
6. Просмотр поставщиков;
7. Выход с аккаунта администратора;
8. Поиск по коду товара.

На нижней панели управления страницы «AutoLight» сверху вниз расположены кнопки для:

1. Резервного копирования базы данных;
2. Резервного восстановления базы данных;
3. Импорта данных из xml;
4. Экспорта данных в xml;
5. Создание новой базы данных и восстановление копии в нее;
6. Переключение базы данных.

Пример нижней панели управления представлен на рисунке 4.3.

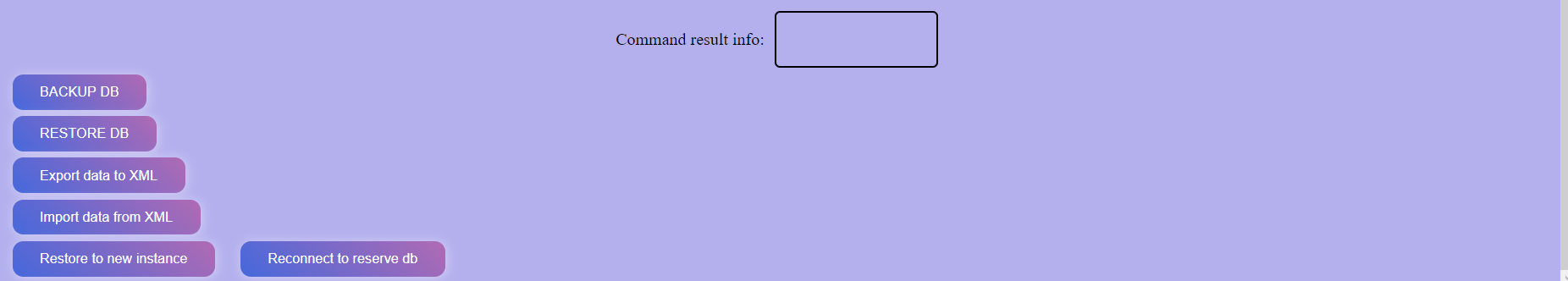


Рисунок 4.3 – Пример нижней панели управления

Для проверки функциональности главной страницы выведем список доступных товаров. Пример списка товаров приведен на рисунке 4.4.



Рисунок 4.4 – Вывода списка товаров

Ниже списка есть кнопки добавления, обновления и удаления товара. Попробуем добавить новую товарную позицию. Добавление товара представлено на рисунке 4.5.

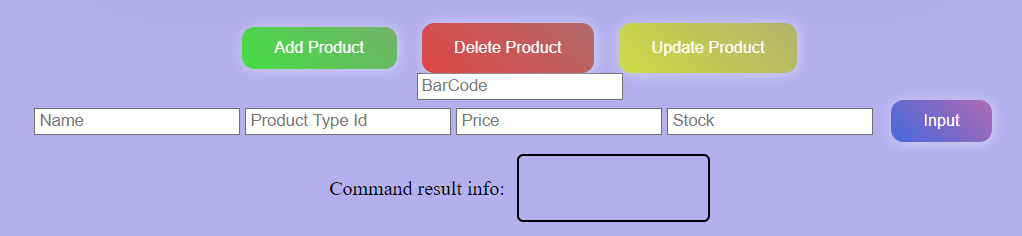


Рисунок 4.5 – Добавление товарной позиции

В случае ошибки, например при вводе не валидного значения, такого, как текстовое значение в поле цены или количества на складе, будет выведено сообщение со статусом результата выполнения операции. Пример сообщения об ошибке представлен на рисунке 4.6.

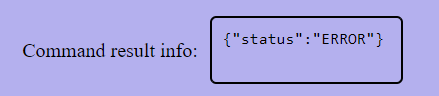


Рисунок 4.6 – Сообщение о ошибке

Попробуем удалить товарную позицию. Удаление товара представлено на рисунке 4.7.

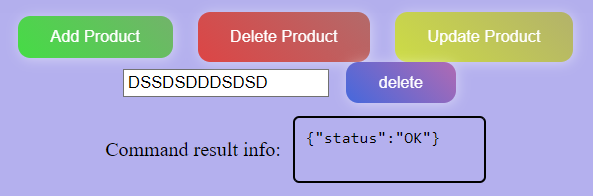


Рисунок 4.7 – Удавление товара

Далее выведем список работников магазина. Пример списка пользователей приведен на рисунке 4.8.

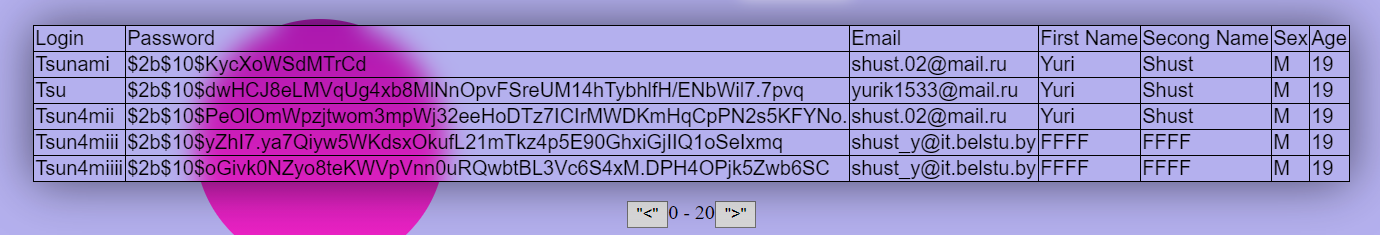


Рисунок 4.8 – Вывод списка пользователей

Выведем список типов продуктов. Пример списка категорий товаров приведен на рисунке 4.9.



Рисунок 4.9 – Вывод списка категорий товаров

Выведем список поставщиков. Пример списка поставщиков товаров приведен на рисунке 4.10.

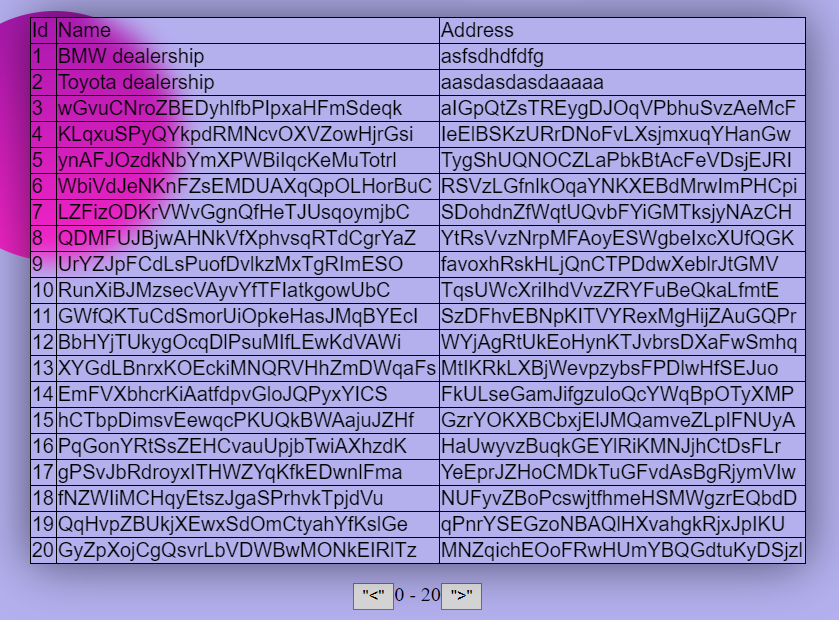


Рисунок 4.10 – Вывод списка поставщиков товаров

Найдем товар по штрих коду, для этого сначала необходимо ввести код необходимого товара. Поиска товара приведен на рисунке 4.11.

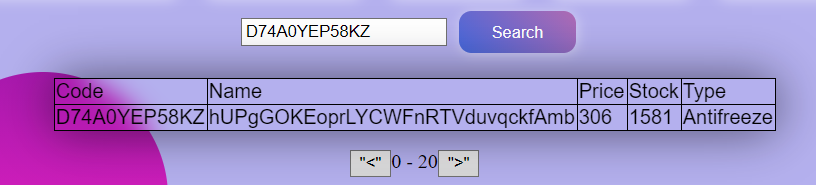


Рисунок 4.11 – Результат поиска товара по коду

Выведем список магазинов. Пример списка магазинов приведен на рисунке 4.12.

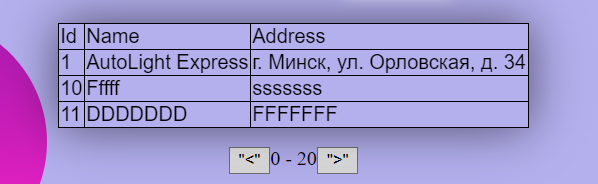


Рисунок 4.12 – Вывод списка магазинов

А теперь подробнее рассмотрим кнопки в нижнем углу.

Самой первой расположена кнопка резервного копирования базы данных, которая будет сохранена на стороне сервера в формате bak. Еще ниже расположена кнопка резервного восстановления базы данных. На ней остановимся по подробнее.

Кнопка резервного восстановления базы данных запускает данный процесс на стороне СУБД. Так как базе данных для восстановления необходим режим монополии, в связи с чем сервер подключается к системной базе данных master, а СУБД переходит в режим монополии. Пользователь не должен пытаться производить запросы в течении некоторого времени, иначе может потребоваться перезапуск сервера, так как возникнет критическая ошибка, связанная с отказом подключения к Microsoft SQL Server.

Еще ниже расположена кнопка импорта данных из xml файла, расположенного на стороне сервера, в таблицу PRODUCTS.

Еще ниже расположена кнопка экспорта данных таблицы PRODUCTS в xml файл на стороне сервера. Так же данные в xml формате будут отправлены пользователю в браузер и будут отображены на новой странице. Пример вывода в браузере приведен на рисунке 4.13.



Рисунок 4.13 – Вывод результата экспорта данных в браузер

Ниже находится создания новой базы данных и восстановления в нее копии основной базы данных. При нажатии появляется поле, в которое надо ввести имя для создания новой базы данных. При подтверждении на стороне Microsoft SQL Server запускается процедура «RestoreNewInstance». Пример создания процедуры приведен на рисунке 4.14.

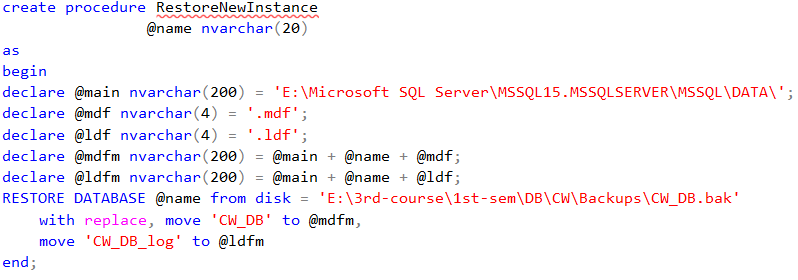


Рисунок 4.14 – Пример создания процедуры «RestoreNewInstance»

Ниже находится кнопка переключения между базами данных. Это позволяет переключаться между существующими базами данных приложения. Для этого мы вводим в появившееся поле имя базы данных к которой мы хотим подключиться. После отправки запроса на сервер закрываются все существующие подключения и происходит подключение к новой базе данных. В консоли сервера можно увидеть новую строку подключения, сгенерированную приложением и сообщение при успешном подключении к новой базе данных. Пример функции переключения базы данных приведен на рисунке 4.15.

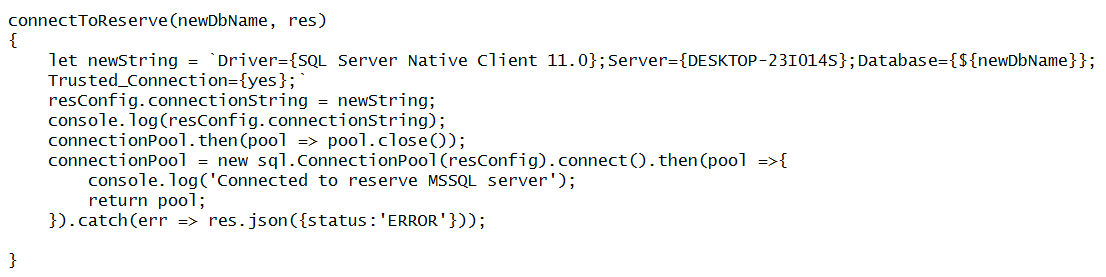


Рисунок 4.15 – Пример функции переключения базы данных

Теперь рассмотрим клиентскую часть приложения. При авторизации за покупателя мы увидим главную страницу. Пример главной страницы приведен на рисунке 4.16.

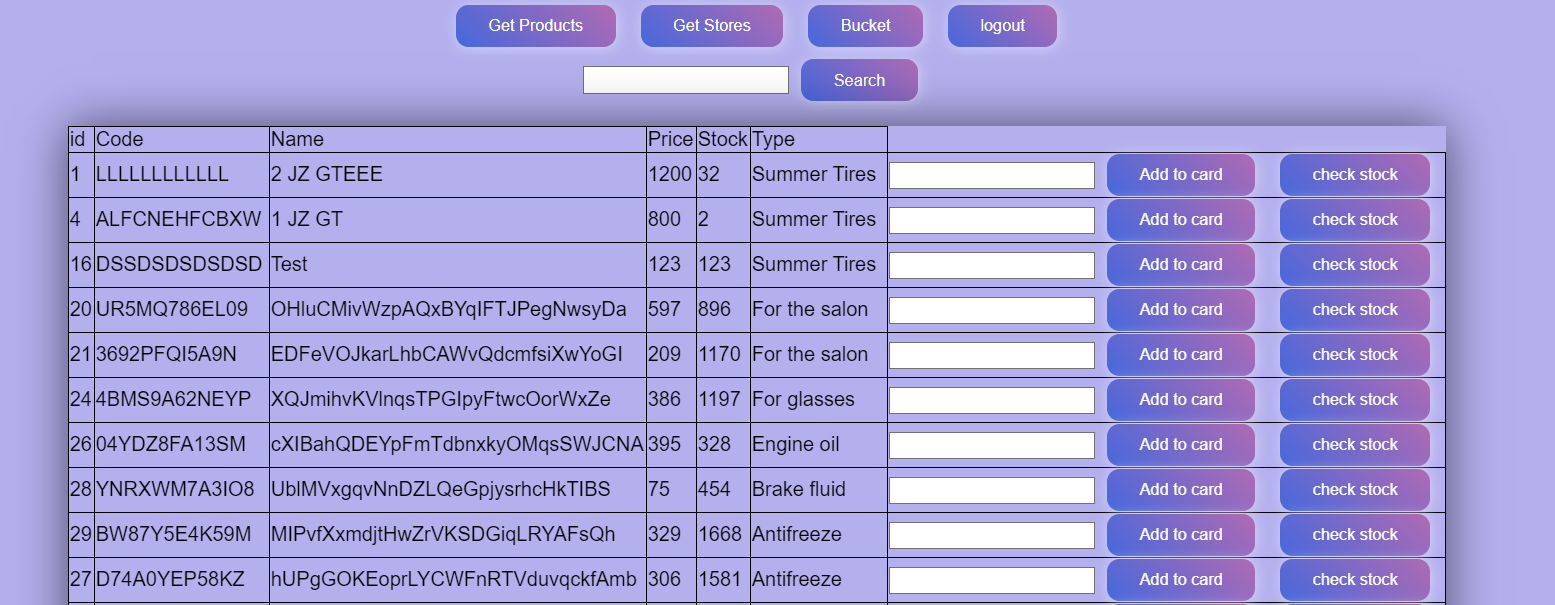


Рисунок 4.16 – Пример главной страницы покупателя

Рассмотри кнопки, которые находятся внутри таблицы слева на право.

Первая это «Add to card». При ее нажатии товар добавляется в корзину пользователя. Для корректной работы должно быть введено количество заказываемого товара в поле левее от кнопки.

Правее находится кнопка проверки наличия товара в магазинах. При ее нажатии пользователю будет выведен список магазинов в которых товар находится в наличии и его количество в этом магазине. Пример результирующего вывода проверки наличия приведен на рисунке 4.17.

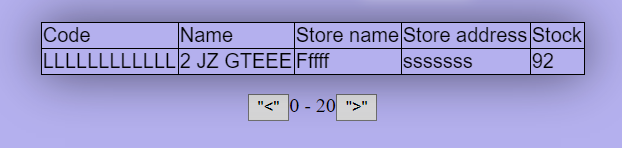


Рисунок 4.17 – Пример отображения наличия товара в магазинах

Так же на главной странице покупателя присутствует кнопка вывода корзины пользователя. Пример вывода пользовательской корзины приведен на рисунке 4.18.

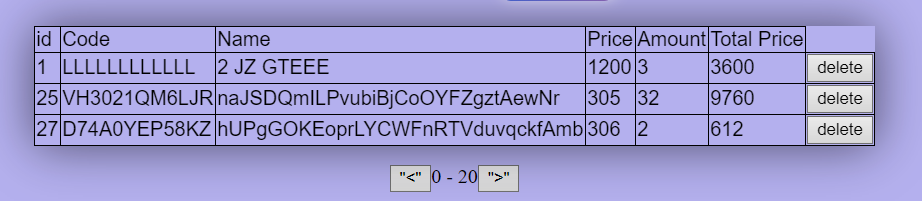


Рисунок 4.18 – Пример вывода пользовательской корзины

В данной таблице присутствует кнопка «delete», которая позволяет удалить товар из пользовательской корзины. Пример таблицы после удаления товара из пользовательской корзины приведен на рисунке 4.19.

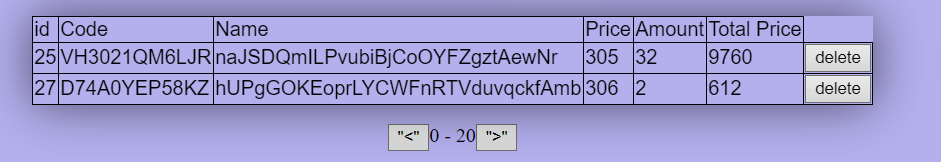


Рисунок 4.19 – Пример таблицы корзины пользователя после удаления товара

# **Заключение**

В процессе решения поставленной задачи была достигнута поставленная цель по созданию базы данных «Сеть магазинов по продаже автозапчастей», которая в купе с приложением формирует сайт для управления сетью магазинов. Основой целью курсового проекта стало проектирование базы данных для дальнейшей интеграции с приложением, которое помогло облегчить взаимодействие с базой данных посредством программного интерфейса. При разработке выполнены следующие пункты:

* Регистрация и авторизация администраторов магазина и покупателей;
* Разделение приложение на сервер и клиент;
* Технология резервного копирования и восстановления базы данных;
* Просмотр информации о сотрудниках;
* Просмотр информации о товарах;
* Просмотр информации о магазинах;
* Просмотр информации о поставщиках;
* Просмотр информации о пользователях;
* Просмотр информации о должностях
* Поиск товара по кодовому полю;
* Импорт и экспорт XML.
* Резервное копирование с созданием новой базы данных;
* Переключение базы данных;

В курсовом проекте были реализованы следующие задачи:

* Создание базы данных;
* Создание сервера;
* Создание клиента с пользовательским интерфейсом;
* Реализация функций работы приложения;
* Тестирование программного продукта.

Данный проект сложно сравнить с аналогами, ведь для каждой сети магазинов системы разрабатываются индивидуально. По сути, данный проект является показательным примером базы данных для различных сетей магазинов, позволяя охватить необходимый минимум требований для их администрирования.

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объёме.

**Список литературы**

1. Node.js v11.15.0 Documentation [Электронный ресурс] / OpenJS Foundation. – Режим доступа: https://nodejs.org/docs/latest-v11.x/api. – Дата доступа: 28.10.2021.

2. Express 4.x API [Электронный ресурс] / StrongLoop, Inc. – Режим доступа: https://expressjs.com/en/4x/api.html. – Дата доступа: 28.10.2021.

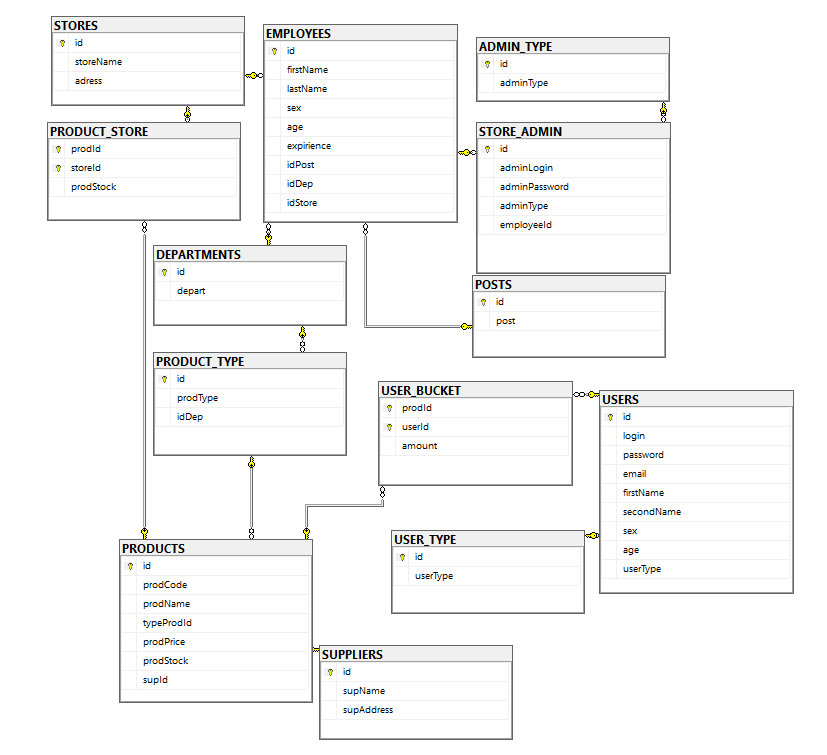
3. Mssql [Электронный ресурс] / OpenJS Foundation. – Режим доступа: https://www.npmjs.com/package/mssql. – Дата доступа: 28.10.2021.

4. RESTORE Statements (Transact-SQL) [Электронный ресурс] / Microsoft. – Режим доступа: https://docs.microsoft.com/en-us/sql/t-sql/statements/restore-statements-transact-sql?view=sql-server-ver15. – Дата доступа: 04.11.2021.

4. BACKUP (Transact-SQL) [Электронный ресурс] / Microsoft. – Режим доступа: https://docs.microsoft.com/en-us/sql/t-sql/statements/backup-transact-sql?view=sql-server-ver15. – Дата доступа: 04.11.2021.

6. Restore a Database Backup Using SSMS [Электронный ресурс] / Microsoft. – Режим доступа: https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/restore-a-database-backup-using-ssms?view=sql-server-ver15. – Дата доступа: 15.11.2021.

**Приложение А**



# **Приложение Б**

