

Все результаты должны быть отправлены на проверку через открытый репозиторий системы контроля версий. В репозиторий должны быть загружены:

- исходный код приложения (структура с файлами, не архив),
- скрипт БД в формате \*.sql (схема и данные),
- данные всех полученных после импорта таблиц БД в формате \*.csv и \*.xls,
- ERD в формате \*.png (таблицы, связи между ними, атрибуты и ключи, обязательность и типы данных).

### **Задание 1. Разработка базы данных**

Заказчик системы (компания по продаже обуви) предоставил файлы с данными (с пометкой import) для переноса в новую систему. Необходимо подготовить данные файлов для импорта (провести нормализацию, например: вынести состав заказа в отдельную таблицу, создать таблицы поставщиков, производителей, категорий).

На основе подготовленных данных для импорта необходимо создать БД в выбранной СУБД (MSSQL/MySQL/Postgres) и загрузить данные импорта в разработанную БД.

В БД обязательны:

- 3 нормальная форма,
- первичные и внешние ключи,
- оптимально подходящие типы данных,
- наименования согласно стандарту выбранной СУБД,
- автоинкрементные значения первичных ключей (где применимы).

Сохранить ER-диаграмму БД в формате \*.png.

### **Задание 2. Создание решения и библиотеки классов**

Создать решение, в которое добавить:

- библиотеку классов,
- веб-API,
- веб-приложение,
- оконное приложение.

Добавить в библиотеку классов контекст данных и модели данных для работы с созданной БД, используя ORM. Созданную библиотеку классов использовать в клиентских приложениях.

При необходимости в библиотеку можно добавлять DTO и сервисы для выполнения следующих заданий по разработке клиентских приложений.

### **Задание 3. Создание веб-API**

Добавить в веб-API контроллеры:

- для авторизации по логину и паролю с использованием JWT,
- для работы с товарами (получение всех товаров, получение товара по артикулу, добавление товара, удаление товара, обновление товара),
- для работы с заказами (получение всех заказов определенного пользователя по логину пользователя, изменение статуса заказа и даты доставки для заказа с указанным номером).

Получение заказов по логину должно быть доступно для авторизованных пользователей.

Редактирование товаров и изменение статуса и даты доставки заказа должны быть доступны для авторизованных пользователей с ролью администратор или менеджер.

### **Задание 4. Просмотр списка товаров в веб-приложении**

Создать веб-приложение для просмотра списка товаров.

В приложении должны отображаться фотографии товаров (если фото нет, показывать заглушку picture.png). Вместо идентификаторов из связанных таблиц должны отображаться наименования. На рисунке показан формат отображения данных о товаре (после двоеточия выводить соответствующее значение):

<div style="border: 1px solid black; width: 50px; height: 50px; margin: 0 auto; text-align: center; line-height: 50px;">Фото</div>	<b>Категория товара   Наименование товара</b> Описание товара: Производитель: Поставщик: Цена: Единица измерения: Количество на складе:	Действующая скидка

В приложении должны быть возможности:

- просмотра цены после применения скидки (исходная цена должна быть зачеркнута, рядом с ней — новая цена после применения скидки),
  - поиска по части описания товара (указывается в поле ввода, без учета регистра),
  - фильтрации по производителю (выбирается в выпадающем списке с возможностью показа товаров всех производителей),
  - фильтрации по цене не более указанной в поле ввода,
  - отображения только тех товаров, у которых есть скидка (при отметке флажка),
  - отображения только тех товаров, которые есть на складе (при отметке флажка),
  - сортировки по названию, по поставщику, по цене, по убыванию цены).
- Фильтры, поиск и сортировка должны применяться совместно.

### **Задание 5. Работа с заказами в веб-приложении**

Добавить в веб-приложение возможность авторизации по логину и паролю. При переходе в любую учетную запись в интерфейсе (правый верхний угол) должны отображаться ФИО пользователя. После авторизации должна быть возможность выхода из системы.

У пользователей системы должны быть следующие возможности:

- гость (неавторизованный пользователь) может нажать на кнопку «Заказать» у любого товара. При нажатии должно выполняться перенаправление на страницу авторизации,
- клиент может нажать на кнопку «Заказать» у любого товара. При нажатии должно выполняться создание нового заказа для выбранного пользователя:
  - дата заказа — текущая дата,
  - дата доставки — через неделю после текущей,
  - код для получения — случайное число от 100 до 999,
  - статус — новый.

После авторизации пользователям становится доступен переход к странице «Заказы»:

- клиент может увидеть список своих заказов в виде таблицы,
- менеджер и администратор могут увидеть список всех заказов в виде таблицы.

В таблице заказов должны отображаться: номер заказа, состав, дата заказа, итоговая стоимость.

### **Задание 6. Работа с товарами в оконном приложении**

Создать оконное приложение для просмотра списка товаров аналогично возможностям приложения из задания 4. При запуске приложения окно входа — первое, что видит пользователь. На нем пользователю предлагается ввести свой логин и пароль. После удачной авторизации пользователь получает доступ к просмотру списка товаров.

Если у товара снижена цена, то основная цена должна быть написана красным цветом, а рядом с ней указана итоговая цена черным цветом.

Если товара нет на складе, строка выделяется голубым цветом.

Пользователи с ролью менеджер и администратор имеют возможность перехода к окну добавления нового товара или редактирования выбранного. Также у них должна быть возможность удаления выбранных товаров.

При редактировании можно добавить/заменить изображение у товара (должно сохраняться в папку с приложением, в БД необходимо хранить путь к изображению).

После модификации товаров данные в окне списка товаров должны быть обновлены.