## **Руководство по эксплуатации: Flower-shop**

### **Описание проекта**

**Flower-shop** — это настольное приложение, разработанное с использованием WPF на языке C#. Оно предназначено для управления цветочным магазином, включая авторизацию персонала и управление каталогом товаров. Проект использует архитектуру MVVM и подключается к базе данных SQL Server Express.

### **Технологии и инструменты**

* **Язык программирования**: C#
* **Платформа**: .NET Framework
* **Технология интерфейса**: WPF (Windows Presentation Foundation)
* **Архитектура**: MVVM (Model-View-ViewModel)
* **База данных**: SQL Server Express
* **Среда разработки**: Visual Studio

### **Инструкция по запуску**

1. **Клонирование репозитория**

git clone https://github.com/gigafem/Flower-shop.git

1. **Открытие проекта**
   * Откройте файл WpfApp1.sln в Visual Studio.
2. **Настройка строки подключения**
   * В файле App.config убедитесь, что строка подключения к базе данных соответствует вашей локальной конфигурации SQL Server Express.
3. **Создание базы данных**
   * В SQL Server Management Studio создайте базу данных с необходимыми таблицами: Catalog, Bouquets, Compositions, Gifts.
   * Убедитесь, что структура таблиц соответствует ожиданиям приложения.
4. **Запуск приложения**
   * Постройте и запустите проект через Visual Studio.

### **Интерфейс пользователя**

* **Окно авторизации (**Window3.xaml**)**
  + Предназначено для входа персонала в систему.
  + Содержит поля для ввода логина и пароля, а также кнопки для входа и возврата.
* **Каталог товаров**
  + После успешной авторизации пользователь получает доступ к каталогу товаров, объединяющему данные из таблиц Bouquets, Compositions и Gifts.

### **Работа с базой данных**

* **Таблица** Catalog
  + Объединяет данные из таблиц Bouquets, Compositions и Gifts.
  + Содержит поля: ProductID, ProductName, Description, Price, Stock, Category.
* **Связи между таблицами**
  + Для объединения данных рекомендуется использовать представление (VIEW) или объединяющую таблицу с внешними ключами.

**Возможные ошибки при работе с базой данных**

* **Ошибка**: "Необходимо объявить скалярную переменную"
  + **Решение**: Убедитесь, что все параметры в SQL-запросах объявлены и передаются корректно.
* **Ошибка**: "Недопустимое имя объекта 'EmployeeLogIn'"
  + **Решение**: Проверьте, существует ли таблица EmployeeLogIn в базе данных и правильно ли указано ее имя в запросе.
* **Ошибка**: "Недопустимое имя столбца 'password'
  + **Решение**: Убедитесь, что в таблице EmployeeLogIn есть столбец с именем password и что имя столбца указано правильно в запросе.
* **Ошибка**: "Невозможно вставить явное значение для столбца идентификаторов в таблице 'Catalog', когда параметр IDENTITY\_INSERT имеет значение OFF."
  + **Решение**: Если необходимо вставить явное значение в столбец с автоинкрементом, временно включите IDENTITY\_INSERT:
  + SET IDENTITY\_INSERT Catalog ON;
  + -- Ваш INSERT-запрос
  + SET IDENTITY\_INSERT Catalog OFF;
* **Ошибка**: "Table 'Catalog' is marked for deletion"
  + **Решение**: Это означает, что таблица была помечена для удаления в режиме конструктора, но изменения еще не сохранены. Либо сохраните изменения, чтобы удалить таблицу, либо закройте окно конструктора без сохранения, чтобы отменить удаление.