

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

Кафедра информационных систем

КУРСОВАЯ РАБОТА

Тема: Разработка веб-приложения «Цветочный магазин “Лаванда”»

Работу выполнил студент: Гонос Анна Сергеевна **группы** М3307
(фамилия, имя, отчество) (номер группы)

Руководитель Григорьев Г. С., Университет ИТМО, преподаватель (квалификационная категория "преподаватель практики")
(фамилия, имя, отчество)

Работа защищена " _____ " _____ 2025 г. с оценкой _____

Подписи членов комиссии: _____

**САНКТ-ПЕТЕРБУРГ
2025**

Оглавление

Введение	3
Определения, обобщения и сокращения.....	4
Описание предметной области	6
Описание прикладного процесса.....	6
Формирование требований	6
Проектирование.....	12
Используемый стек технологий	12
Системная архитектура	12
Архитектура данных	15
Программная архитектура	19
Разработка.....	29
Реализация серверного API.....	29
Реализация пользовательского интерфейса	31
Заключение	45
Список использованной литературы.....	46

Введение

Объектом разработки является веб-приложение «Цветочный магазин “Лаванда”» на языке программирования TypeScript с использованием фреймворка Nest – для серверной части приложения и языка программирования JavaScript, без использования фреймворков – для клиентской части приложения. Выбранный фреймворк один из самых поддерживаемых и распространённых.

Целью работы является разработка веб-приложения и пользовательского интерфейса, анализ требования и моделирование процессов, средств автоматизации и архитектуры информационной системы.

В ходе работы были получены следующие результаты:

- Серверная часть системы, принимающая запросы.
- Клиентская часть системы, предоставляющая интерфейс пользователя.
- База данных для хранения информации о пользователях, файлах, а также служебной информации внутри системы.
- Реализованы механизмы безопасности, такие как JWT-аутентификация, защита маршрутов и управление правами пользователей (покупатель, флорист, администратор).

Определения, обобщения и сокращения

Браузер – программное обеспечение, предназначенное для отображения веб-страниц и взаимодействия с веб-ресурсами.

Фреймворк – программная основа, предоставляющая стандартные структуры кода и набор готовых решений, позволяющих ускорить разработку. В данном проекте для серверной части используется NestJS, один из наиболее популярных фреймворков на платформе Node.js.

Серверная часть (backend) – логика и функционал приложения, выполняемые на сервере. Обеспечивают обработку запросов, взаимодействие с базой данных и отправку данных клиенту.

Клиентская часть (frontend) – сторона приложения, с которой взаимодействует пользователь. В данном проекте реализована с использованием чистого JavaScript, HTML и CSS.

HTTP (HyperText Transfer Protocol) – протокол передачи данных между клиентом и сервером.

JWT (JSON Web Token) – открытый стандарт для безопасной передачи данных между клиентом и сервером в формате JSON. Используется в проекте для реализации аутентификации и авторизации.

REST API – стиль архитектуры, определяющий принципы организации взаимодействия между клиентом и сервером через HTTP-запросы.

База данных – организованная совокупность данных, хранимых в виде таблиц. В проекте используется PostgreSQL — система управления реляционными базами данных.

Пользовательское представление – графический интерфейс, через который конечный пользователь взаимодействует с приложением.

ORM (Object Relational Mapping) – технология, позволяющая работать с базой данных как с объектами. В проекте использовала TypeORM.

Кэширование – временное хранение часто запрашиваемых данных с целью повышения производительности. Реализовано в проекте с помощью CacheModule NestJS.

Сессия – временный интерфейс взаимодействия между клиентом и сервером. Управление сессией осуществляется с помощью JWT-токенов.

Yandex Object Storage – облачное хранилище файлов, в котором хранятся изображения товаров. В проекте используются переменные окружения для доступа к внешнему хранилищу.

Роль «админ», роль «флорист» (admin, florist) – пользователи, которые могут добавлять, редактировать и удалять свои «карточки товаров», просматривать оформленные заказы, менять их статусы, просматривать отзывы, которые оставили покупатели.

(Пока права и возможности у них одинаковые, но в дальнейшем при добавлении модуля платежей у админа будет больше возможностей, поэтому сделала сразу разделение)

Роль «покупатель» (user) – пользователь, имеющий возможность просматривать товары, добавлять их в корзину или избранное, делать заказы, оставлять отзывы на сайте магазина.

Описание предметной области

Описание прикладного процесса

Разрабатываемое веб-приложение предназначено для цифровой трансформации деятельности реального цветочного магазина «Лаванда», расположенного в г. Партизанск. Цель проекта — создание онлайн-платформы, позволяющей клиентам выбирать и заказывать товары (цветы, букеты, композиции), а сотрудникам магазина (администратору и флористам) — самостоятельно добавлять и управлять карточками товаров без участия разработчиков.

В рамках приложения можно выделить следующие ключевые процессы:

- Добавление новых товаров - флористы и администратор загружают данные о цветах, ценах, фото и описании.
- Редактирование информации о товарах - изменение цены, наличия, состава и фотографий.
- Удаление устаревших товаров - удаление композиций, которые больше не актуальны.
- Просмотр заказов - отслеживание текущих и завершённых заказов.
- Обработка отзывов - просмотр и управление обратной связью от покупателей.

Формирование требований

В ходе анализа прикладного процесса был получен следующий список функциональных требований:

1. Модуль работы с товарами

- Возможность добавления новой карточки товара с указанием следующих полей:

- Название;
- Описание (опционально);
- Цена;
- Цена с учетом скидки (опционально, по умолчанию 0);
- Количество;
- Изображение;
- Описание к изображению (опционально);
- Редактирование карточек товара с возможностью изменить:
 - Название;
 - Описание (опционально);
 - Цена;
 - Цена со скидкой (опционально);
 - Количество;
 - Изображение;
 - Описание к изображению (опционально);
- Удаление карточек товаров;
- Загрузка и управление изображениями товаров;
- Отображение всех товаров по категориям;
- Фильтрация и поиск товаров по названию, цене, категории.

2. Модуль работы с категориями товаров

- Создание новых категорий
 - Название категории
 - Описание (опционально)
- Редактирование существующих категорий

- Возможность изменения названия и описания
- Удаление категорий
- Просмотр списка категорий
- Вывод информации о категории по ID
- Фильтрация товаров по категории
 - Используется при отображении товаров клиентам (покупателям)

3. Модуль заказов

Для покупателя:

- Оформление заказа на основе содержимого корзины с указанием следующих полей:
 - Имя получателя;
 - Телефон получателя;
 - Способ доставки (курьер / самовывоз)
 - Адрес доставки (если выбран способ «Курьер»);
 - Время доставки;
 - Способ оплаты (наличными при самовывозе или онлайн, если выбран способ «Курьер»);
 - Выбор использования бонусов: списать или копить;
- Получение подтверждения о создании заказа;
- Просмотр истории своих заказов;

Для админа/флориста:

- Просмотр всех заказов;
- Изменение статуса заказа (создан → собран → доставлен / отменён);
- Детальный просмотр состава заказа;
- Отслеживание метода оплаты и статуса платежей;

4. Модуль пользователей

- Регистрация пользователя с указанием:
 - Имя;
 - Фамилия;
 - Телефон;
 - Электронная почта;
- Разделение ролей: *admin, florist, user*;
- Личный кабинет:
 - Просмотр персональных данных: имя, фамилия, телефон, email
 - Редактирование имени, фамилии, телефона, email
 - Просмотр истории заказов
 - Накопленные бонусные баллы
 - Уровень бонусной карты

5. Модуль отзывов

- Оставить отзыв о магазине с указанием следующих полей:
 - Оценка (от 1 до 5 включительно)
 - Текстовый комментарий
 - Изображение (опционально)
- Редактировать свои отзывы с корректировкой всех полей
- Просмотреть свои и чужие отзывы

6. Модуль корзины

- Добавление товара в корзину;
- Удаление одного товара из корзины;

- Увеличение количества какого-то товара на 1;
- Уменьшение количества какого-то товара на 1;
- Очищение корзины полностью;
- Подсчёт количества товаров в корзине.

7. Модуль избранного

- Добавление товара в «избранное»;
- Удаление товара из «избранного»;
- Сохранение товара в избранном с пометкой и невозможностью добавить данный товар в корзину, после того как флорист или админ добавили данный товар в архив.

Нефункциональные требования

1. Безопасность

- Обязательная аутентификация и авторизация для всех пользователей.
- Использование JWT-токенов для безопасной передачи данных между клиентом и сервером.
- Хранение токена в защищённых куках.
- Разграничение прав доступа по ролям: user, florist, admin.

2. Документация API

- Все программные интерфейсы должны быть задокументированы по спецификации OpenAPI v3.0 и выше с использованием библиотеки Swagger .
- Каждый эндпоинт должен содержать:
 - Описание метода
 - Примеры запросов и ответов
 - Перечень возможных ошибок

- Система должна обрабатывать запросы пользователя с минимальным временем ожидания.

3. Производительность

- Реализация кэширования часто запрашиваемых данных (список товаров для каталога).

4. Интеграция с внешними системами

- Интеграция с облачным хранилищем изображений через Yandex Object Storage.
- Отправка уведомлений флористам и администраторам через Telegram Bot API (@BotFather).

5. Удобство использования

- Интерфейс должен быть удобным и понятным как для покупателей, так и для флористов.
- Поддержка мобильных устройств (адаптивный дизайн).

Проектирование

Используемый стек технологий

Решение создания именно веб-приложения обусловлено тем, что необходимо было обеспечить доступ к системе с любого устройства, в любое время. Веб-приложение решает этот вопрос, а также снимает вопрос обновлений на стороне клиента.

Проект использует стек стандартных технологий, характерный для большинства веб-приложений: HTML, CSS, Javascript.

В качестве веб-сервера используется Express.

В качестве базы данных используется PostgreSQL, ввиду того, что данная СУБД является самой стабильной в данной связке. Библиотекой для работы с данными была выбрана TypeORM.

В проекте используется свободная распределённая система управления версиями Git, хранилищем исходных кодов является крупнейший веб-сервис GitHub, а в качестве хостинга используется облачный сервис Render.com.

Системная архитектура

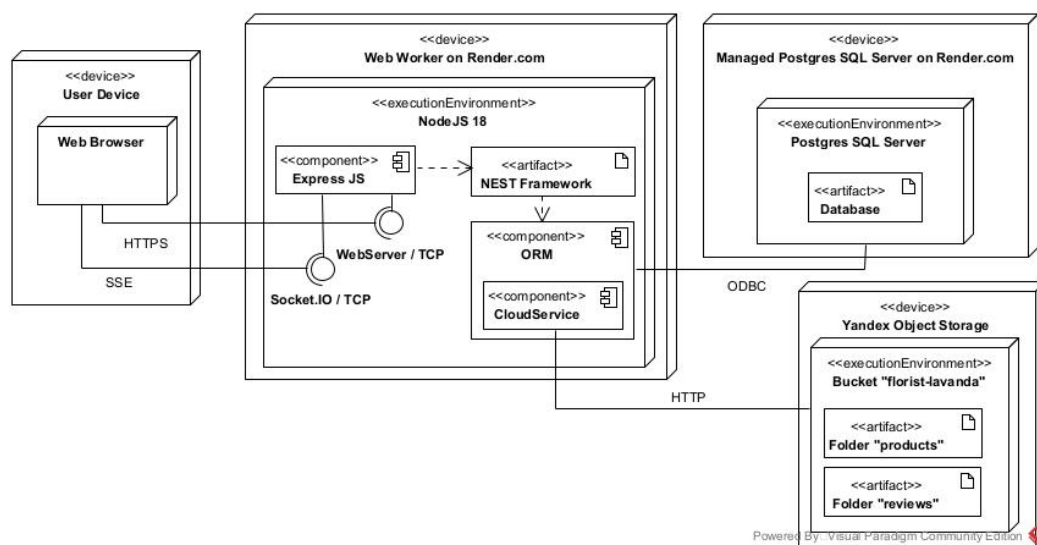


Рисунок 1. Системная архитектура приложения.

1. User Device

Пользовательское устройство, с которого осуществляется доступ к системе.

Внутри:

- **Web Browser** – браузер, поддерживающий современные стандарты HTML5, CSS3, JavaScript
- Отправляет запросы через протокол HTTPS на сервер
- Получает SSR-рендеринг страниц (рендеринг происходит на стороне сервера)

2. Web Worker on Render.com

Серверное окружение, где развернуто веб-приложение.

Внутри:

- **Node.js 18** – среда выполнения, в которой запускается сервер
- **Express JS** – встроенный сервер, обрабатывающий HTTP-запросы
- **Nest Framework** – фреймворк, предоставляющий структуру MVC, маршруты, контроллеры, провайдеры
- **ORM (TypeORM)** – слой абстракции над базой данных
- **CloudService** – сервис, интегрированный с Yandex Object Storage, используется для загрузки и получения изображений

Серверная часть обрабатывает:

- Регистрацию и авторизацию пользователей
- Добавление, редактирование и удаление товаров
- Работу с корзиной, избранным, заказами и отзывами
- Загрузку изображений в облачное хранилище

3. Managed PostgreSQL Server on Render.com

Облачный SQL-сервер, используемый для хранения данных.

Внутри:

- **PostgreSQL 14** – реляционная СУБД
- **Database** – набор таблиц:
 - **User, Product, Order, Cart, Favorite, Review, Category, Payment, OrderItem**

Сервер взаимодействует с базой данных через:

- **ODBC** – стандартный протокол для связи с реляционными БД
- **TypeORM** – ORM-библиотека, которая абстрагирует SQL-запросы

4. Yandex Object Storage

Облачное хранилище, используемое для загрузки и хранения изображений.

Внутри:

- **Bucket: "florist-lavanda"** – аналог логической директории
- **Folder: "products/"** – папка с изображениями товаров
- **Folder: "reviews/"** – папка с изображениями отзывов

Сервер взаимодействует с этим хранилищем через:

- **HTTP** – методы **PUT, GET, DELETE** для работы с объектами

Архитектура данных

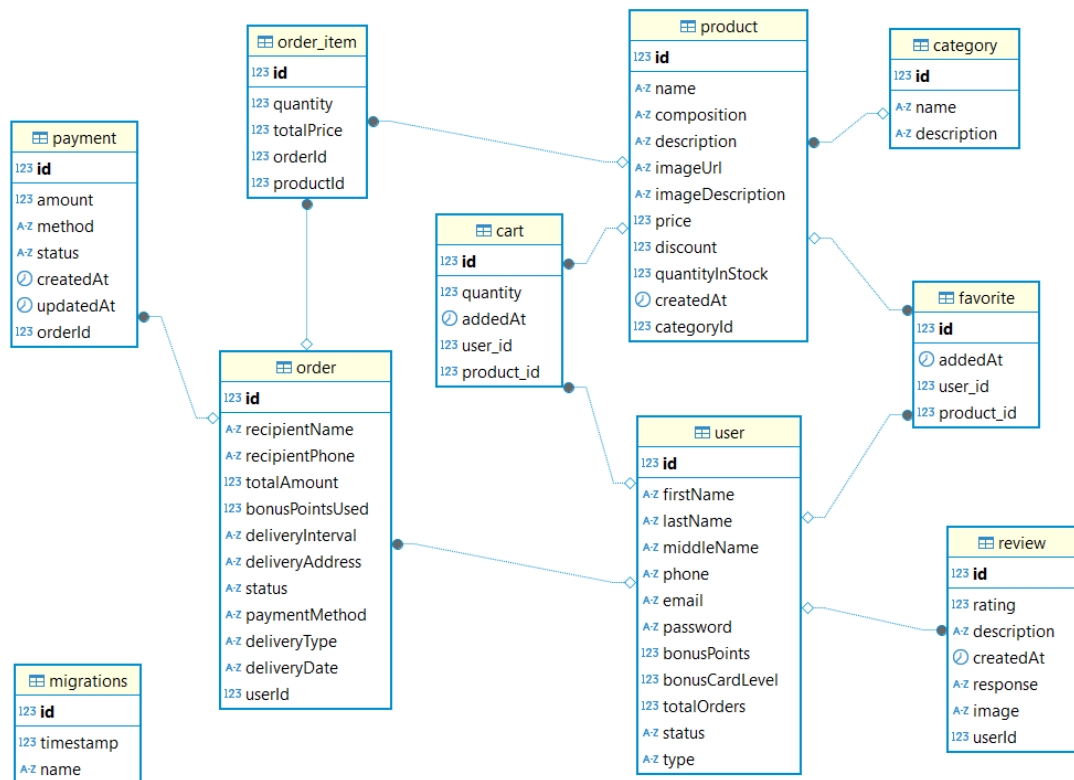


Рисунок 2. Схема таблиц базы данных

1) User (пользователь): основная сущность системы, представляющая клиента, администратора или флориста. Хранит персональные данные, информацию о бонусной программе, статусе и роли пользователя.

1. id - уникальный идентификатор пользователя.
2. firstName - имя пользователя, необязательно уникальное.
3. lastName - фамилия пользователя, необязательно уникальная.
4. middleName - отчество пользователя (опционально).
5. phone - номер телефона пользователя.
6. email - электронная почта пользователя.
7. password - хэшированный пароль пользователя.
8. bonusPoints - количество бонусных баллов пользователя.
9. bonusCardLevel - уровень бонусной карты пользователя.

10. totalOrders - общее количество заказов пользователя (используется для бонусной программы).

11. status - статус пользователя: ACTIVE, INACTIVE, BLOCKED.

12. type - роль пользователя: USER, ADMIN, FLORIST (Пользователь (покупатель), флорист, админ).

13. orders - список заказов пользователя.

14. reviews - список отзывов пользователя.

15. favorites - список избранных товаров пользователя.

16. cart - список товаров в корзине пользователя.

2) Product (Товар): сущность, представляющая товар (букет, коробка для композиции и т.п.)

1. id - уникальный идентификатор товара.

2. name - название товара, необязательно уникальное.

3. category - категория, к которой принадлежит товар (ссылка на сущность Category).

4. composition - состав товара (опционально).

5. description - описание товара (опционально).

6. imageUrl - URL изображения товара (опционально).

7. imageDescription - описание изображения товара (опционально).

8. price - цена товара.

9. discount - новая скидочная цена товара (опционально).

10. quantityInStock - количество товара в магазине.

11. cartItems - список товаров, добавленных в корзину пользователя.

12. favorites - список товаров, добавленных в избранное пользователя.

13. createdAt - дата создания товара.

3) Category (Категория): сущность, представляет собой категорию товара (например, "Монобукеты", "Свадебные букеты").

1. id - уникальный идентификатор категории.
2. name - название категории, необязательно уникальное.
3. description - описание категории (опционально).
4. products - список товаров, принадлежащих данной категории.

4) Review (Отзыв): сущность, позволяющая пользователям оставлять отзывы о магазине

1. id - уникальный идентификатор отзыва.
2. user - пользователь, который оставил отзыв (ссылка на сущность User).
3. rating - оценка товара (число от 1 до 5).
4. description - текст отзыва (опционально).
5. createdAt - дата создания отзыва.
6. response - ответ на отзыв от администратора (опционально).
7. image - URL изображения, прикрепленного к отзыву (опционально).

5) Payment (Платежи): сущность, отвечающая за обработку платежей

1. id - уникальный идентификатор платежа.
2. order - заказ, связанный с платежом (ссылка на сущность Order).
3. amount - сумма платежа.
4. method - метод оплаты: CASH, ONLINE.
5. status - статус платежа: PENDING, COMPLETED, FAILED, REFUNDED.

6. createdAt - дата создания платежа.

7. updatedAt - дата обновления платежа (опционально).

6) Order (Заказ): сущность, представляющая собой заказ пользователя, включающий в себя информацию о товарах, доставке и платеже

1. id - уникальный идентификатор заказа.

2. user - пользователь, создавший заказ (ссылка на сущность User).

3. items - список товаров в заказе (ссылка на сущность OrderItem).

4. recipientName - имя получателя (опционально).

5. recipientPhone - телефон получателя (опционально).

6. totalAmount - общая сумма заказа.

7. bonusPointsUsed - количество использованных бонусных баллов.

8. deliveryInterval - интервал доставки (опционально).

9. deliveryAddress - адрес доставки (обязательно при выборе типа доставки COURIER).

10. status - статус заказа: CREATED, CONFIRMED, ASSEMBLED, DELIVERED, CANCELLED.

11. payments - список платежей по заказу (ссылка на сущность Payment).

12. paymentMethod - метод оплаты: CASH, CARD, ONLINE.

13. deliveryType - тип доставки: PICKUP, COURIER.

14. deliveryDate - дата доставки (опционально).

7) Order Item (Позиция заказа): сущность, представляющая отдельный товар в заказе. Связана с сущностью "Заказ"

1. id - уникальный идентификатор товара в заказе.

2. order - заказ, к которому принадлежит товар (ссылка на сущность Order).

3. product - товар, добавленный в заказ (ссылка на сущность Product).

4. quantity - количество товара в заказе.

5. totalPrice - общая стоимость товара (вычисляется как $\text{price} * \text{quantity}$).

8) Cart (Корзина): сущность, позволяющая добавлять товары в корзину:

1. id - уникальный идентификатор записи в корзине.

2. user - пользователь, который добавил товар в корзину (ссылка на сущность User).

3. product - товар, добавленный в корзину (ссылка на сущность Product).

4. quantity - количество товара в корзине (по умолчанию 1).

5. addedAt - дата добавления товара в корзину.

9) Favorite (Избранное): сущность, позволяющая пользователям добавлять товары в список избранного

1. id - уникальный идентификатор записи в избранном.

2. user - пользователь, который добавил товар в избранное (ссылка на сущность User).

3. product - товар, добавленный в избранное (ссылка на сущность Product).

4. addedAt - дата добавления товара в избранное.

Программная архитектура

Таблица 1. Отношения модулей и классов

Название модуля	Название класса	Назначение класса
ProductModule	ProductController	Контроллер
	ProductApiController	REST API для внешних запросов
	ProductService	Сервис
	ProductResolver	GraphQL-резольвер
	CreateProductDto	Модель передачи данных
	UpdateProductDto	Модель передачи данных
	EditProductPageData	Данные для страницы редактирования
	CatalogQuery	Параметры фильтрации и пагинации
	Pagination	Информация о страницах
	CatalogData	Сборка данных для каталога
	AddProductPageData	Данные для формы добавления товара
	Product	ORM-сущность товара
	ProductRepository	Работа с базой данных
CategoryModule	CategoryController	Контроллер
	CategoryApiController	REST API для внешних запросов
	CategoryService	Сервис
	CategoryResolver	GraphQL-резольвер
	CreateCategoryDto	Модель передачи данных
	UpdateCategoryDto	Модель передачи данных
	CreateCategoryInput	GraphQL InputType для создания
	UpdateCategoryInput	GraphQL InputType для обновления
	Category	ORM-сущность категории
	CategoryRepository	Работа с базой данных
OrderModule	OrderController	Контроллер
	OrderApiController	REST API

	OrderService	Сервис
	OrderResolver	GraphQL-резольвер
	Order	ORM-сущность заказа
	OrderItem	ORM-сущность элемента заказа
	Payment	ORM-сущность платежа
	PaymentMethod	Перечисление методов оплаты
	PaymentStatus	Перечисление статусов платежей
	DeliveryType	Перечисление типов доставки
	OrderStatus	Перечисление статусов заказа
	CreateOrderInput	GraphQL InputType для создания
	UpdateOrderInput	GraphQL InputType для обновления
	OrderPageData	Данные для страницы оформления заказа
	OrderSuccessData	Данные для страницы успеха после заказа
	CreateOrderDto	Модель передачи данных
	UpdateOrderStatusDto	Модель передачи данных
	OrderRepository	Работа с базой данных
	OrderItemRepository	Работа с элементами заказа
	PaymentRepository	Работа с платежами
ReviewModule	ReviewController	Контроллер
	ReviewApiController	REST API
	ReviewService	Сервис
	ReviewResolver	GraphQL-резольвер
	CreateReviewDto	Модель передачи данных
	UpdateReviewDto	Модель передачи данных
	CreateReviewInput	GraphQL InputType для создания
	UpdateReviewInput	GraphQL InputType для обновления
	Review	ORM-сущность отзыва

	ReviewRepository	Работа с базой данных
CartModule	CartController	Контроллер
	CartApiController	REST API для внешних запросов
	CartService	Сервис
	CartMiddleware	Промежуточный слой, отвечает за подсчёт товаров в корзине
	CartResolver	GraphQL-резольвер
	Cart	ORM-сущность товара в корзине
	CartRepository	Работа с базой данных
FavoriteModule	FavoriteController	Контроллер
	FavoriteApiController	REST API для внешних запросов
	FavoriteService	Сервис
	FavoritesMiddleware	Промежуточный слой, считает количество избранных товаров
	FavoriteResolver	GraphQL-резольвер
	Favorite	ORM-сущность товара в избранном
	FavoriteRepository	Работа с базой данных
	PaginatedFavoritesDto	Модель ответа с пагинацией для GraphQL
	PaginationInfo	Информация о страницах для GraphQL
NotificationModule	TelegramService	Сервис отправки уведомлений через Telegram
	TelegramBot	Объект Telegram-бота из библиотеки node-telegram-bot-api
CloudModule	CloudService	Сервис загрузки файлов в облако
UserModule	UserController	Контроллер
	UserApiController	REST API для внешних запросов
	UserService	Сервис

	UserResolver	GraphQL-резольвер
	User	ORM-сущность пользователя
	UserRepository	Работа с базой данных
	UserRole	Перечисление ролей
	UserStatus	Статусы пользователя
AuthModule	AuthController	Контроллер
	AuthService	Сервис
	JwtAuthGuard	Защита маршрутов по JWT
	AuthMiddleware	Прослойка проверки авторизации
	LoginDto	DTO для входа
	RegisterDto	DTO для регистрации
	JwtAuthRedirectFilter	Фильтр исключений
JwtModule	JwtService	Генерация и верификация токена

Таблица 2. Описание классов

Название класса	Описание класса
ProductController	Обрабатывает HTTP-запросы, связанные с товарами: просмотр, добавление, редактирование, удаление
ProductApiController	Предоставляет REST API для интеграции с другими системами или клиентскими приложениями
ProductService	Реализует бизнес-логику: создание, обновление, удаление, фильтрация товаров
CreateProductDto	DTO для создания нового товара.
UpdateProductDto	DTO для редактирования информации о товаре.
ProductResolver	Предоставляет данные о товарах через GraphQL API
EditProductPageData	Ответ контроллера для страницы редактирования товара в GraphQL
CatalogQuery	Содержит параметры фильтрации и пагинации для вывода каталога в GraphQL

Pagination	Информация о текущей странице, общем количестве товаров и навигации по страницам в GraphQL
CatalogData	Комплексный объект данных для отображения каталога товаров на странице в GraphQL
AddProductPageData	Возвращает данные для страницы добавления товара в GraphQL
Product	ORM-сущность, представляющая товар в базе данных
ProductRepository	TypeORM-репозиторий, обеспечивающий работу с базой данных
CategoryController	Обрабатывает HTTP-запросы, связанные с категориями товара: создание, редактирование, удаление
CategoryApiController	Предоставляет REST API для работы с категориями
CategoryService	Реализует бизнес-логику: создание, обновление, удаление, пагинация категорий
CategoryResolver	(GraphQL) предоставляет данные о категориях через GraphQL API
CreateCategoryDto	DTO для создания новой категории
UpdateCategoryDto	DTO для редактирования информации о категории (наследуется от CreateCategoryDto)
CreateCategoryInput	Входной тип для создания сущности в GraphQL
UpdateCategoryInput	Входной тип для обновления сущности в GraphQL
Category	ORM-сущность, представляющая категорию товаров в базе данных
CategoryRepository	TypeORM-репозиторий, обеспечивающий работу с данными о категориях
OrderController	Обрабатывает HTTP-запросы по работе с заказами: оформление, просмотр истории, изменение статуса
OrderApiController	Предоставляет REST API для интеграции с другими системами или клиентскими приложениями
OrderService	Реализует полную логику заказов: создание, обновление статуса, начисление бонусов, работа с корзиной и уменьшение остатков товаров

OrderResolver	(GraphQL) предоставляет данные о заказах через GraphQL API
Order	ORM-сущность, представляющая заказ в базе данных, содержит информацию о пользователе, товарах, сумме, статусе, способе доставки и оплаты
OrderItem	Сущность, описывающая отдельный товар внутри заказа, включает количество и вычисляемую стоимость
Payment	ORM-сущность, описывающая оплату по заказу
PaymentMethod	Метод оплаты. Перечисление (cash, online)
PaymentStatus	Статус оплаты. Перечисление (pending, completed, failed, refunded)
DeliveryType	Тип доставки. Перечисление (pickup, courier)
OrderStatus	Статус заказа. Перечисление (created, confirmed, assembled, delivered, cancelled)
CreateOrderInput	Входной тип GraphQL, используемый при создании заказа
UpdateOrderInput	Входной тип GraphQL, позволяет обновлять статус заказа
OrderPageData	Объект данных для отображения формы оформления заказа
OrderSuccessData	Объект данных для отображения успешного заказа
CreateOrderDto	DTO для передачи данных при создании заказа
UpdateOrderStatusDto	DTO для обновления статуса заказа
OrderRepository	TypeORM-репозиторий для работы с заказами
OrderItemRepository	Работа с товарами внутри заказа
PaymentRepository	Управление состоянием оплаты по заказу
ReviewController	Обрабатывает HTTP-запросы: создание, вывод, редактирование и удаление отзывов
ReviewApiController	Предоставляет REST API для внешних систем
ReviewService	Реализует бизнес-логику: создание, обновление, удаление и пагинация отзывов

ReviewResolver	(GraphQL) предоставляет данные о отзывах через GraphQL API
CreateReviewDto	DTO для передачи данных при создании отзыва
UpdateReviewDto	DTO для частичного обновления отзыва
CreateReviewInput	Входной тип GraphQL, используемый при создании отзыва
UpdateReviewInput	Входной тип GraphQL, используемый при обновлении отзыва
Review	ORM-сущность, представляющая отзыв в базе данных
ReviewRepository	TypeORM-репозиторий, обеспечивающий работу с отзывами
CartController	Обрабатывает HTTP-запросы: добавление, удаление, изменение количества товаров в корзине
CartApiController	Предоставляет REST API для работы с корзиной через клиентские приложения
CartService	Реализует бизнес-логику: добавление, редактирование, удаление товаров в корзине
CartMiddleware	Прослойка, которая автоматически считает количество товаров в корзине у пользователя
CartResolver	(GraphQL) предоставляет данные о корзине через GraphQL API
Cart	ORM-сущность, представляющая товар в корзине
CartRepository	TypeORM-репозиторий, обеспечивающий работу с данными о корзине
FavoriteController	Обрабатывает HTTP-запросы: просмотр избранного, добавление/удаление товаров
FavoriteApiController	Предоставляет REST API для работы с избранным через клиентские приложения
FavoriteService	Реализует бизнес-логику: добавление, удаление, подсчёт и проверка наличия товаров в избранном
FavoritesMiddleware	Прослойка, автоматически считает количество избранных товаров у пользователя

FavoriteResolver	(GraphQL) предоставляет данные о избранном через GraphQL API
Favorite	ORM-сущность, представляющая товар в избранном
FavoriteRepository	TypeORM-репозиторий, обеспечивающий работу с данными
PaginatedFavoritesDto	DTO с данными о списке товаров и пагинации
PaginationInfo	Данные о текущей странице, общем количестве товаров, наличии следующей/предыдущей страницы
TelegramService	Класс, реализующий логику отправки уведомлений во внешние системы — в Telegram
TelegramBot	Экземпляр бота, используемый для отправки сообщений в Telegram
CloudService	Реализует логику загрузки файлов в облачное хранилище (Yandex Object Storage)
UserController	Обрабатывает HTTP-запросы: просмотр профиля, редактирование данных, вывод истории заказов
UserApiController	Предоставляет REST API для работы с пользователями через клиентские приложения
UserService	Реализует бизнес-логику: регистрация, обновление данных, начисление бонусов, управление ролям
UserResolver	(GraphQL) предоставляет данные о пользователях через GraphQL API
User	ORM-сущность, представляющая пользователя в системе
UserRepository	TypeORM-репозиторий, обеспечивающий работу с данными пользователя
UserRole	Перечисление ролей: user, florist, admin
UserStatus	Статусы пользователя: active, inactive, blocked
AuthController	Обрабатывает HTTP-запросы: регистрация, вход, выход, отрисовка страниц
AuthService	Реализует логику регистрации и авторизации через JWT

JwtAuthGuard	Проверяет наличие и валидность токена перед доступом к защищённым маршрутам
AuthMiddleware	Промежуточный слой, устанавливает req.user, если пользователь авторизован
LoginDto	Модель данных для формы входа
RegisterDto	Модель данных для формы регистрации
JwtAuthRedirectFilter	Перехватывает ошибки аутентификации и перенаправляет на /auth/login
JwtService	Используется для генерации и проверки JWT-токена

Разработка

Реализация серверного API

В качестве описания программного интерфейса был выбран инструмент, поддерживающий стандарт OAS 3.0 – Swagger. Далее представлена полученная документация API полученная автоматически по директивам, указанным в декораторах различных методов и структурах данных внутри разрабатываемой информационной системы.

Florist API

1.0 OAS 3.0

API documentation for Florist application

Authorize

products Товары

GET /api/products/archived

Получение страницы архивированных товаров

GET /api/products

Получение каталога товаров

GET /api/products/new

Получение самых новых товаров для главной страницы

GET /api/products/{id}

Отдельная страница товара с основной информацией

DELETE /api/products/{id}

Удаление товара

GET /api/products/add

Получение страницы добавления нового товара

POST /api/products/add

Добавление нового товара

Parameters

Try it out

No parameters

Request body ^{required}

application/json

Example Value | Schema

```
{
  "name": "Букет #23",
  "composition": "хризантемы, ромашки, лилии",
  "description": "Необычный букет хризантем",
  "price": 999,
  "discount": 500,
  "quantityInStock": 3,
  "categoryName": "Новобукеты",
  "imageUrl": "/uploads/1743381876586-369252395.jpg",
  "imageDescription": "Необычный букет хризантем"
}
```

Responses

Code	Description	Links
201	Товар успешно добавлен	No links
401	Неавторизованный доступ	No links
403	Нет прав для добавления товаров	No links

GET /api/products/{id}/edit

Страница редактирования товара

PATCH /api/products/florist/products/{id}

Обновление товара

PATCH /api/products/{id}/archive

Архивирование товара

Рисунок 3. Программный интерфейс серверного API.

PATCH	/api/products/{id}/archive	Архивирование товара
profile		
Личный кабинет		
GET	/api/profile	
GET	/api/profile/orders	
GET	/api/profile/orders/{id}	
GET	/api/profile/edit	
PATCH	/api/profile/edit	
GET	/api/profile/bonus-card	
orders		
Заказы и заказы		
GET	/api/orders	Получение списка заказов клиента
GET	/api/orders/orders/success	Список успешно выполненных заказов
POST	/api/orders/orders/create	Создание нового заказа
GET	/api/orders/orderList/orders	Получение списка заказов пользователя
GET	/api/orders/orderList/order/{id}	Список заказов определенного заказа
POST	/api/orders/orderList/order/{id}/update-status	Обновление статуса заказа
cart		
Корзина		
GET	/api/cart	Получение корзины клиента
GET	/api/cart/count	Получение количества товаров в корзине
GET	/api/cart/status	Получение статуса корзины в корзине
POST	/api/cart/{productId}/add	Добавление товара в корзину
POST	/api/cart/{productId}/inc/wash	Увеличение количества товара в корзине
POST	/api/cart/{productId}/dec/wash	Уменьшение количества товара в корзине
DELETE	/api/cart/{productId}/remove	Удаление товара из корзины
DELETE	/api/cart/clear	Очищение корзины
favorites		
Открытые и избранные товары		
GET	/api/favorites	Получение списка избранных товаров
GET	/api/favorites/count	Получение количества товаров в избранном
GET	/api/favorites/{productId}	Получение списка товаров в избранном
POST	/api/favorites/api/{productId}	Добавление или удаление товара из избранного
categories		
Категории и категории		
POST	/api/categories	Создание новой категории
GET	/api/categories	Список всех категорий с пагинацией
GET	/api/categories/{id}	Список в категорию по ID
PATCH	/api/categories/{id}	Обновление категории
DELETE	/api/categories/{id}	Удаление категории
reviews		
Отзывы		
GET	/api/reviews	Список всех отзывов с пагинацией
POST	/api/reviews	Создание нового отзыва
GET	/api/reviews/{id}	Получение отзыва по ID
PATCH	/api/reviews/{id}	Обновление отзыва
DELETE	/api/reviews/{id}	Удаление отзыва
auth		
Авторизация		
Auth		
GET	/auth/login	
POST	/auth/login	
GET	/auth/register	
POST	/auth/register	
GET	/auth/logout	
Schemas		
Category		
Product		
OrderItem		

Рисунок 4. Программный интерфейс серверного API.

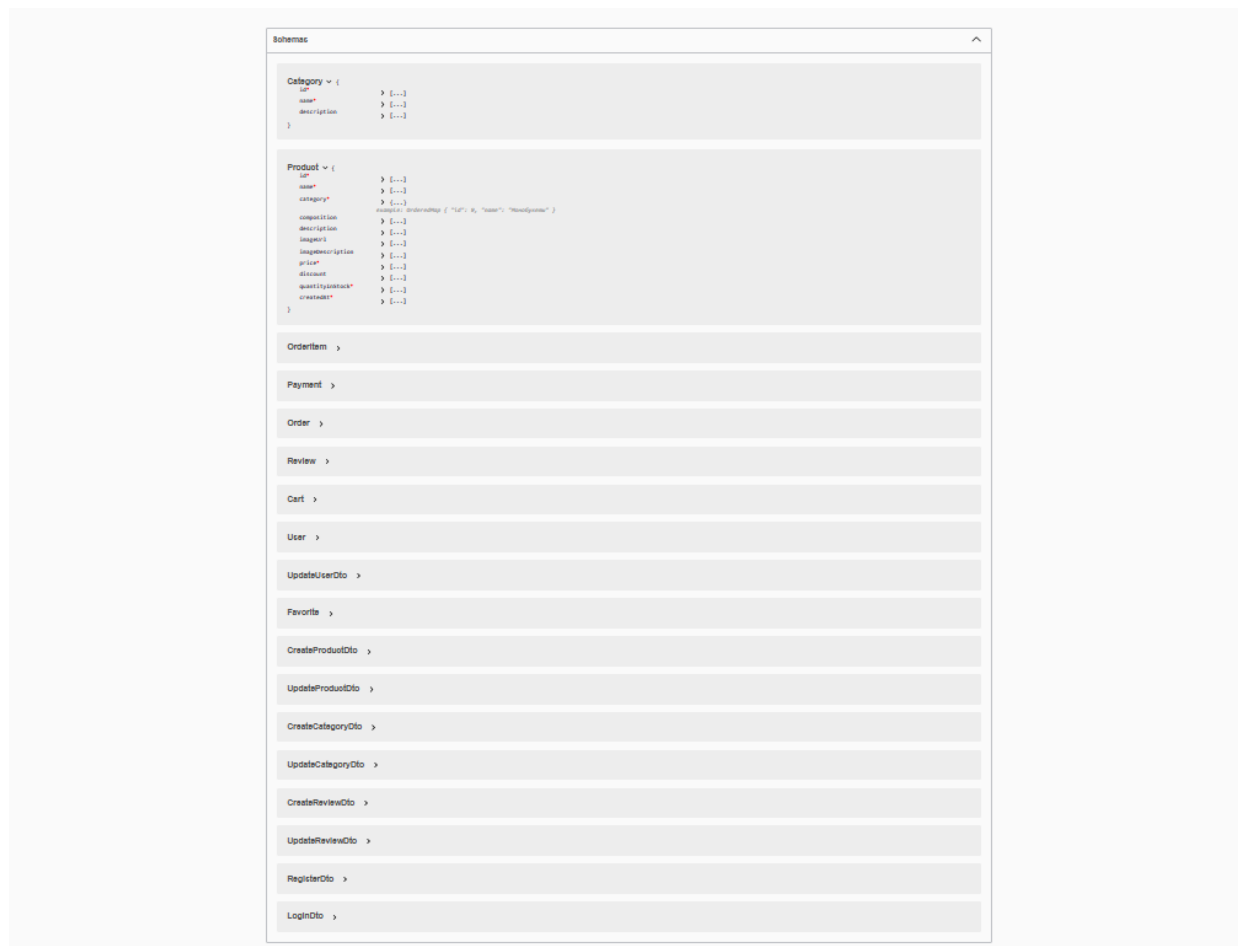


Рисунок 5. Программный интерфейс серверного API.

Реализация пользовательского интерфейса

1. Модуль работы с товарами

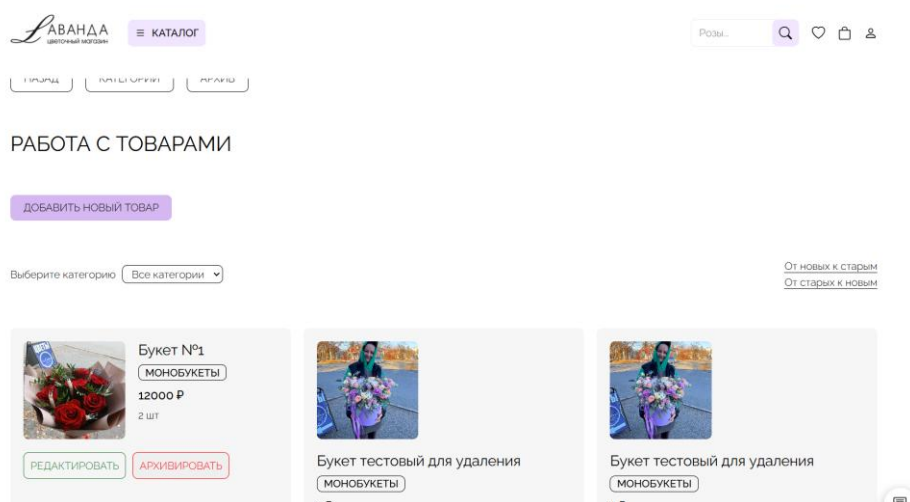


Рисунок 6. Отображение всех товаров в «Аккаунт флориста -> Товары».

ДОБАВИТЬ НОВЫЙ ТОВАР

Название
Букет №1

Описание
Добавить описание (не обязательно)

Состав
роза

Цена
10000

Цена со скидкой
0

Количество в магазине
2

Категория
Монобукеты

Выберите категорию
Монобукеты

Выберите файл | 144026_3e578640.jpg

Описание изображения
Добавить описание изображения

СОХРАНИТЬ ТОВАР

ЛАВАНДА
цветочный магазин

КАТАЛОГ

НАЗАД

КАТЕГОРИИ

АРХИВ

РАБОТА С ТОВАРАМИ

ДОБАВИТЬ НОВЫЙ ТОВАР

Выберите категорию

Все категории



Букет №1

МОНОБУКЕТЫ

10000 Р

2 шт

РЕДАКТИРОВАТЬ

АРХИВИРОВАТЬ

Рисунок 7. Форма добавления товара в «Аккаунт флориста -> Товары».

Справа видно, что после сохранения все корректно отображается.

РЕДАКТИРОВАНИЕ ТОВАРА

Название
Букет №1

Описание
Букет прекрасных ярких роз

Состав
роза

Цена
10000

Цена со скидкой
0

Количество в магазине
2

Категория Монобукеты

Текущее изображение

Заменить изображение

Выберите файл | Файл не выбран

СОХРАНИТЬ ИЗМЕНЕНИЯ

ЛАВАНДА
цветочный магазин

КАТАЛОГ

НАЗАД

КАТЕГОРИИ

АРХИВ

РАБОТА С ТОВАРАМИ

ДОБАВИТЬ НОВЫЙ ТОВАР

Выберите категорию

Все категории



Букет №1

МОНОБУКЕТЫ

12000 Р

2 шт

РЕДАКТИРОВАТЬ

АРХИВИРОВАТЬ

Рисунок 8. Форма редактирования товара в «Аккаунт флориста ->

Товары». Поменяла цену с 10000 руб. до 12000 руб. и описание.

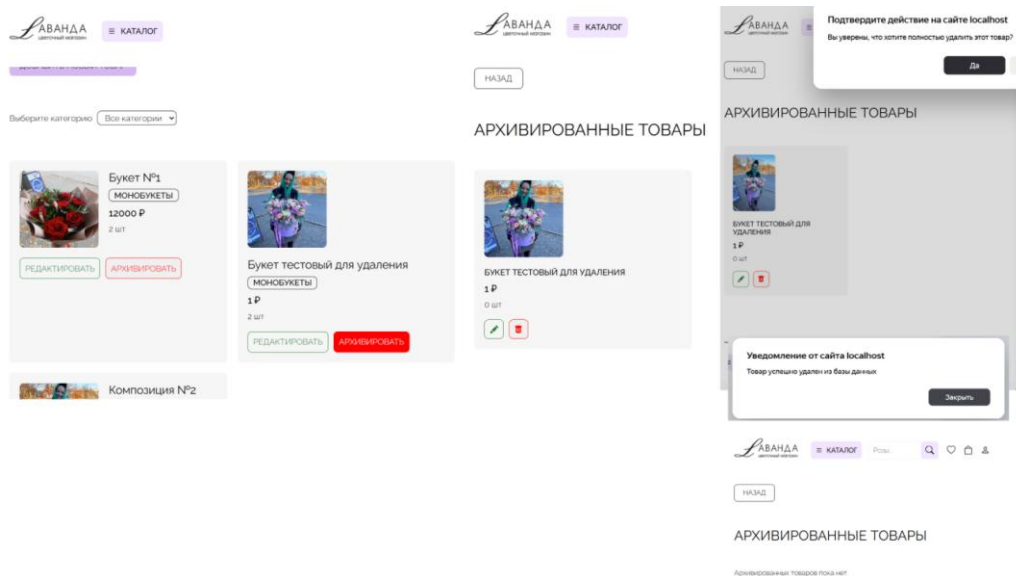


Рисунок 9. Товар можно архивировать, а уже только после этого в разделе «Архив» удалить.

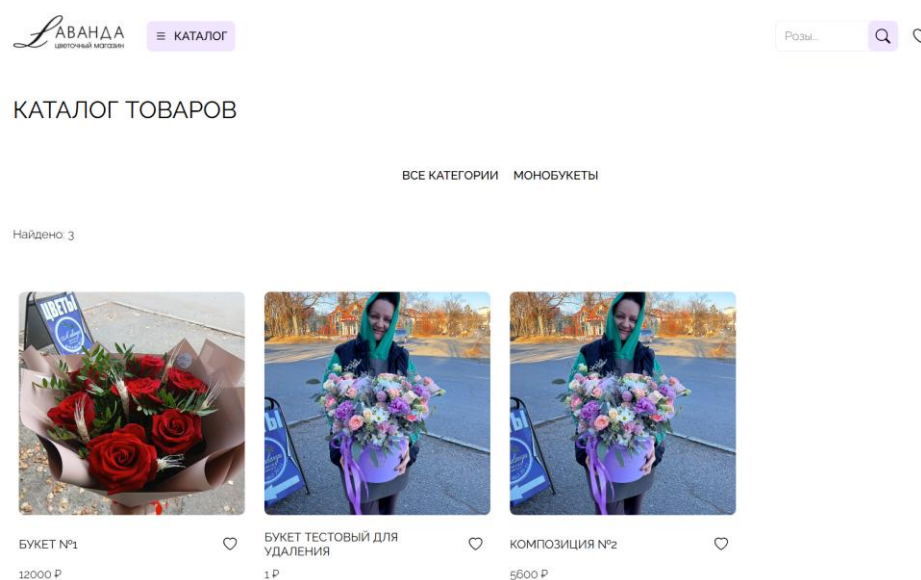


Рисунок 10. Страница «Каталог»

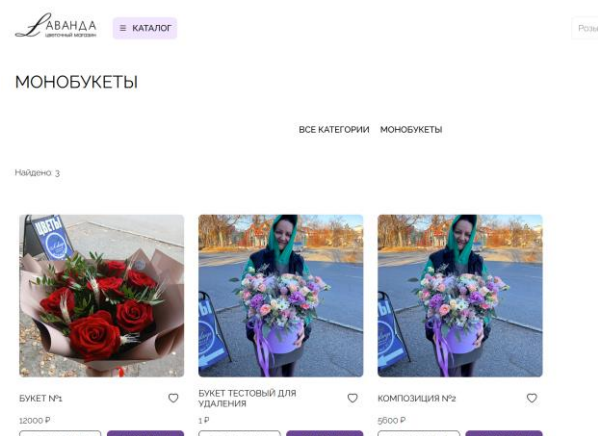


Рисунок 11. Фильтрация товаров по категории (заголовок у сайта так же меняется на соответствующий)

2. Модуль работы с категориями товаров

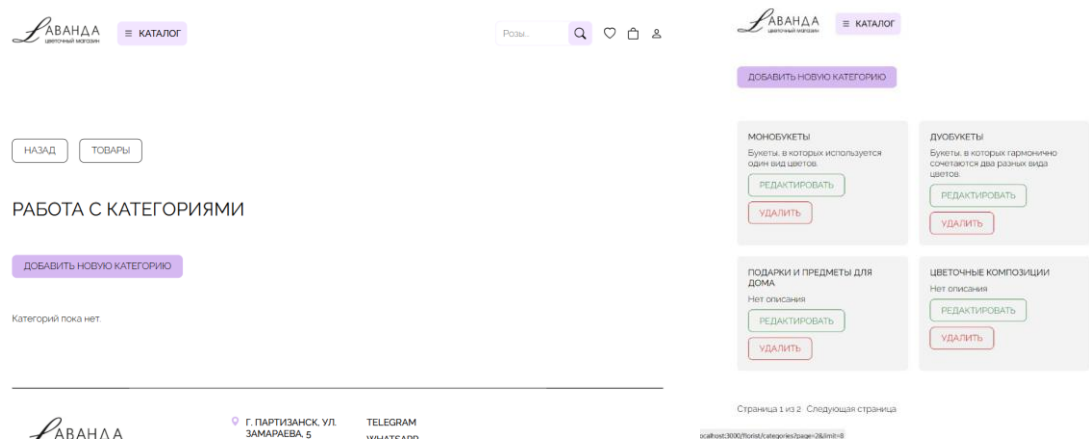


Рисунок 12. Страница Аккаунт флориста -> Категории. Страница со всеми категориями

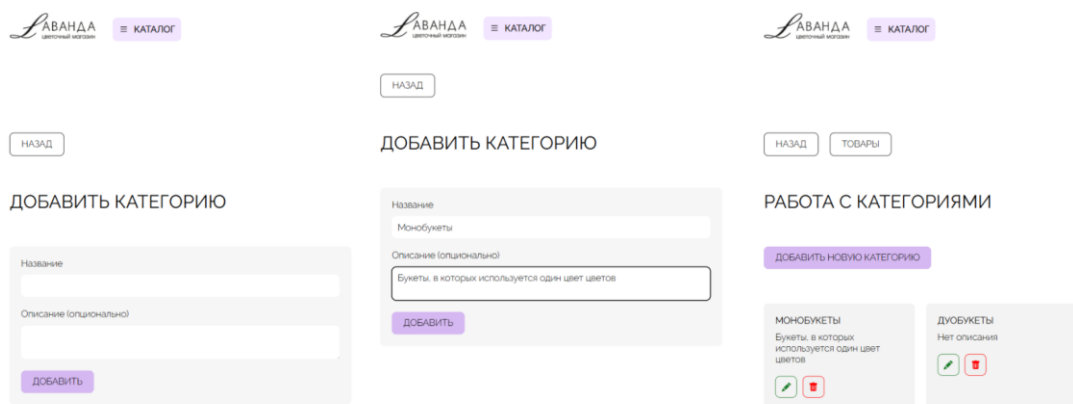


Рисунок 13. Создание новой категории с указанием названия и описания (опционально) с последующим рендингом на страницу со всеми категориями

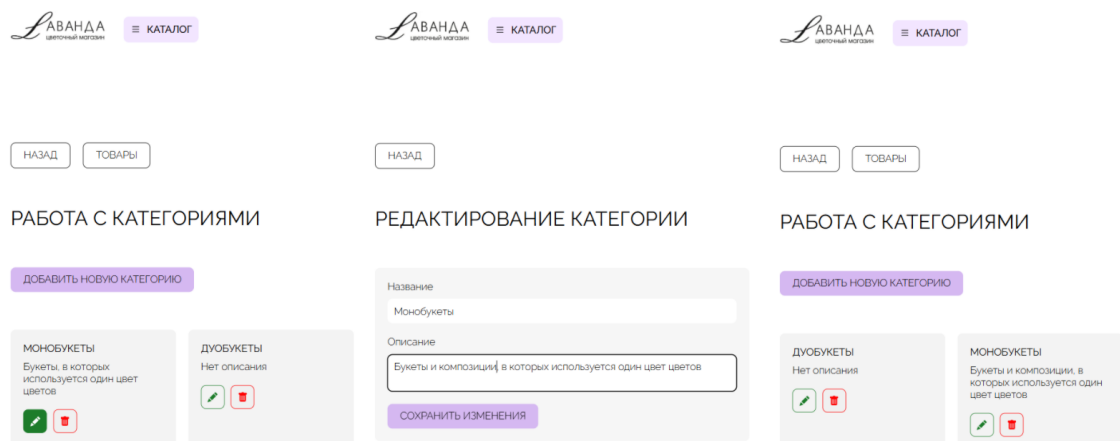


Рисунок 14. Редактирование категории. Редактирую описание категории

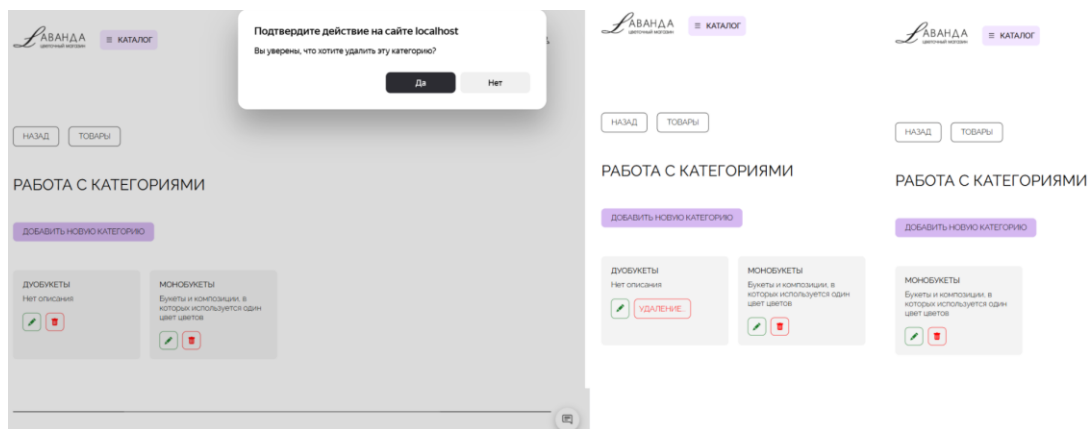


Рисунок 15. Удаление категории. Удалила категорию дуобукеты. При удалении текст на кнопке с иконки меняется на «Удаление...»

3. Модуль заказов

Для покупателя:

ЛАВАНДА

КАТАЛОГ

ОФОРМЛЕНИЕ ЗАКАЗА

КОНТАКТНЫЕ ДАННЫЕ

Ваши данные

ГоносСергеевнаgonosanna@mail.ru

Данные получателя

Если доставку получаете вы, то эти данные не заполняйте

Светлана89681393338

ДОСТАВКА

Способ доставки

Курьерская доставка

Выберите дату

30.05.2025

Выберите интервал получения

09:00 - 12:00

Адрес доставки

Введите адрес, укажите так же дом и квартиру

г. Санкт-Петербург, Кц

ОПЛАТА

Способ оплаты

Онлайн

БОНУСНЫЕ БАЛЛЫ

Монеты на вашем счете: 0

☒ Копить бонусы
 ☐ Списать все бонусы

ИТОГО

Стоимость товаров	23600 Р
Стоимость доставки	0.00 Р
Итоговая сумма	23600.00 Р

ОФОРМИТЬ ЗАКАЗ

Нажимая продолжить я даю свое согласие на обработку персональных данных в соответствии с Политикой конфиденциальности

Рисунок 16. Из корзины перейдем в «оформить заказ». Страница оформления заказа. Заполняем необходимые данные, бонусы копим. Видим итоговую стоимость с учетом всех скидок и бонусов. Нажимаем «Оформить заказ».



Рисунок 17. Получение подтверждения о создании заказа.

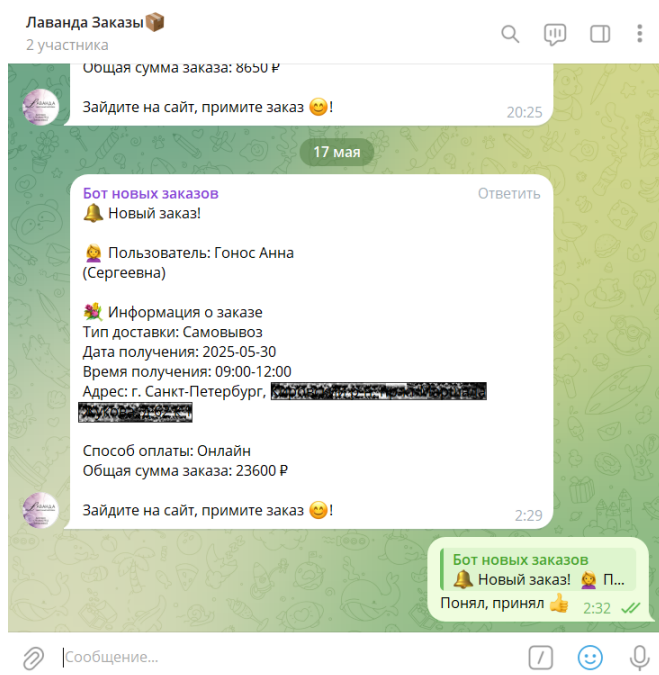


Рисунок 18. С помощью телеграм бота, при оформлении заказа на сайте, флористы будут видеть оповещение в телеграме и оперативно обрабатывать заказы.



Рисунок 19. Просмотр истории своих заказов

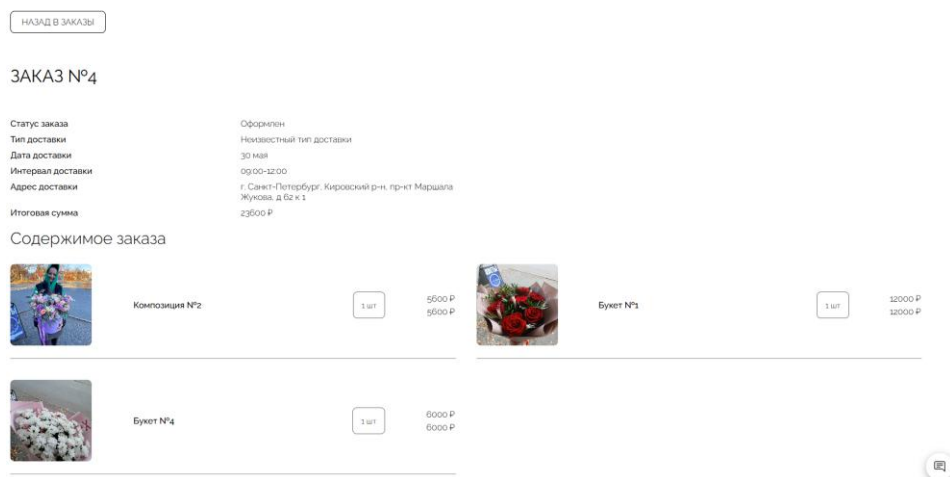


Рисунок 20. Просмотр своего отдельного заказа

Для админа/флориста:

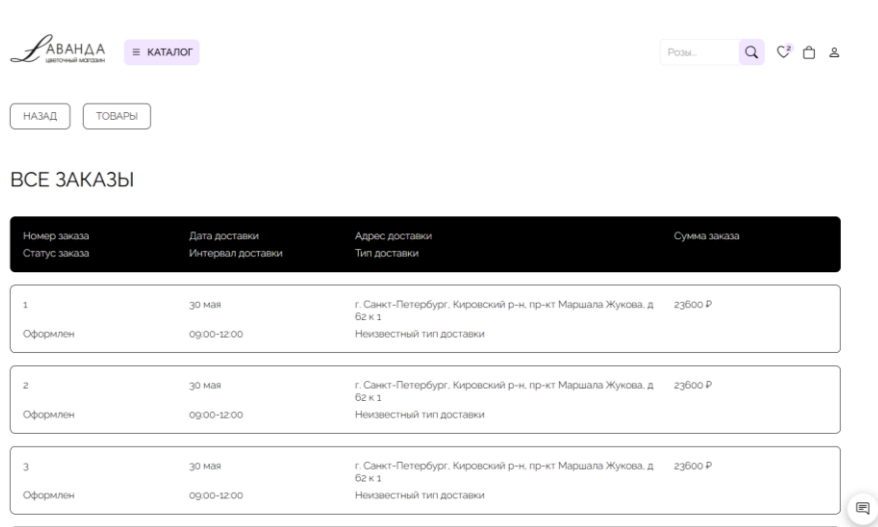


Рисунок 21. Просмотр всех заказов

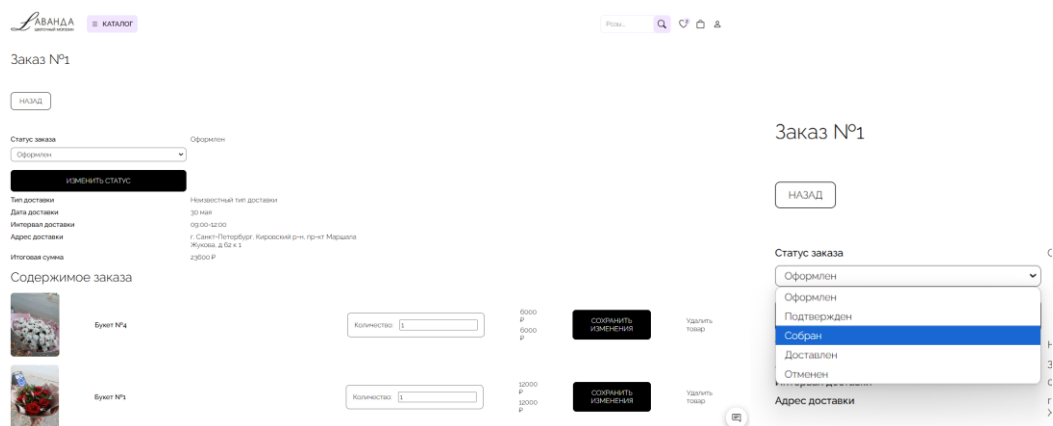


Рисунок 22. Просмотр отдельной страницы заказа. Изменение статуса заказа.

4. Модуль пользователей

The screenshot shows the 'РЕГИСТРАЦИЯ' (Registration) and 'ВХОД' (Login) forms. The registration form includes fields for Name (Анна), Surname (Гонос), Phone (89681393331), Email (gonosanna1@mail.ru), and Password. The login form includes fields for Email (gonosanna1@mail.ru) and Password. A 'ВОЙТИ В ЛИЧНЫЙ КАБИНЕТ' (Login to personal cabinet) button is present. Links for 'Не помню пароль' (Forgot password), 'Впервые здесь?' (First time here?), and 'Регистрация' (Registration) are provided. The header features the 'Лаванда' logo, a 'КАТАЛОГ' button, and a search bar. The footer includes social media links for Telegram, WhatsApp, and Instagram, along with the address: г. Партизанск, ул. Замаева.

Рисунок 23. Страницы с формами для регистрации и входом

The screenshot shows the 'ЛИЧНЫЙ КАБИНЕТ' (Personal Cabinet) page. On the left, a sidebar menu lists: Главная (Home), Редактировать данные (Edit data), Мои заказы (My orders), Программа лояльности (Loyalty program), and Выход (Logout). The main area displays 'УЧЕТНЫЕ ДАННЫЕ' (Account Data) for 'Анна Гонос' (Anna Gonos), including fields for ФИО (FIO), Телефон (Phone: 89681393331), and Почта (Email: gonosanna1@mail.ru). The header features the 'Лаванда' logo, a 'КАТАЛОГ' button, and a search bar.

Рисунок 24. После регистрации и входа перенаправление идет в личный кабинет. В кабинете видны личные данные пользователя. Это при регистрации всех users, если заходит админ или флорист, то см. рис.25

This screenshot is identical to the previous one, showing the 'ЛИЧНЫЙ КАБИНЕТ' (Personal Cabinet) page for 'Анна Гонос'. However, it includes an additional button labeled 'АККАУНТ ФЛОРИСТА' (Florist Account) located above the 'ЛИЧНЫЙ КАБИНЕТ' header. The sidebar menu and account data remain the same.

Рисунок 25. Зашла с аккаунта админа. Видна кнопка аккаунт флориста

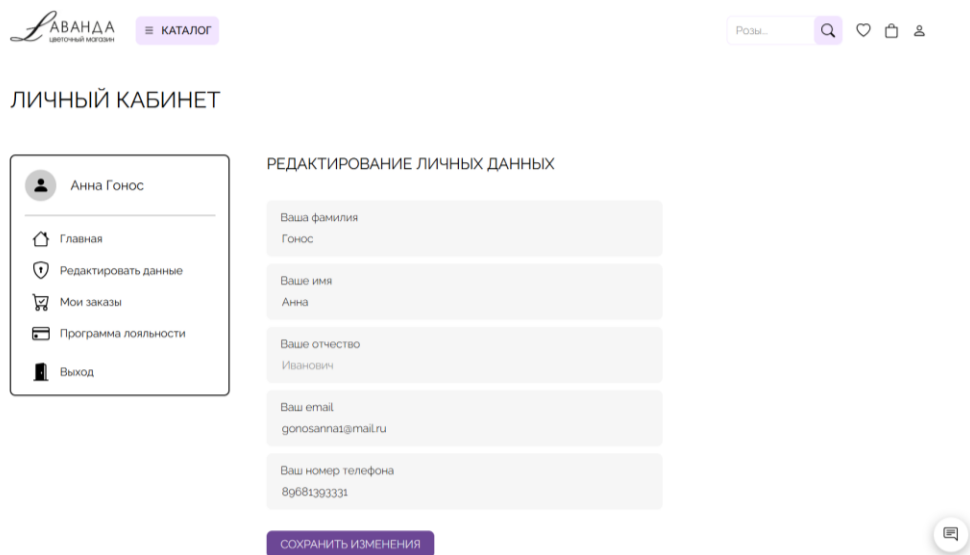


Рисунок 26. Редактирование информации пользователя. Поменяем отчество

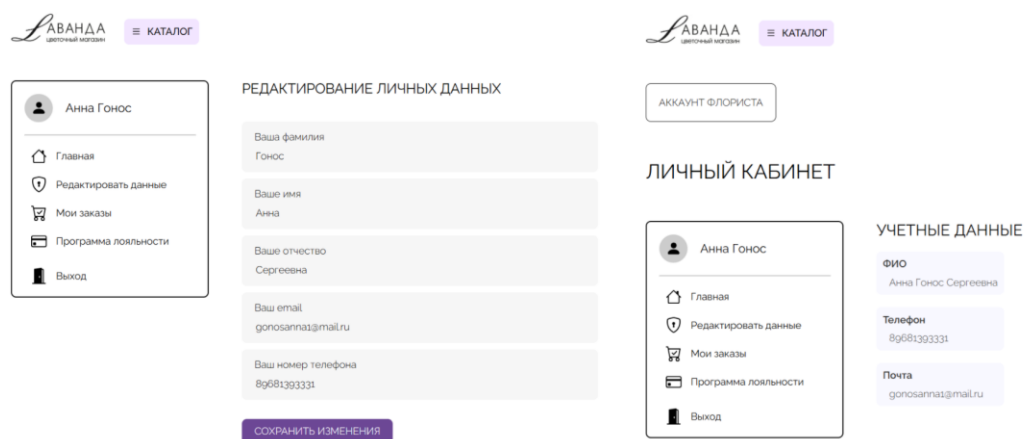


Рисунок 27. Ввела отчество – «Сергеевна», после нажатия «Сохранить изменения» перенаправление идет на главную страницу личного кабинета, где видно, что отчество сохранилось.

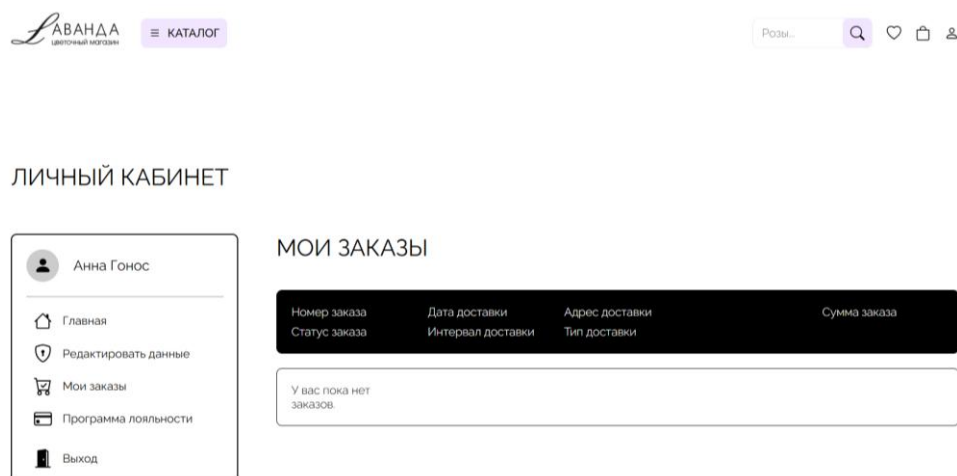


Рисунок 28. Страница заказов в личном кабинете.



Рисунок 29. Страница бонусов/программа лояльности в личном кабинете.

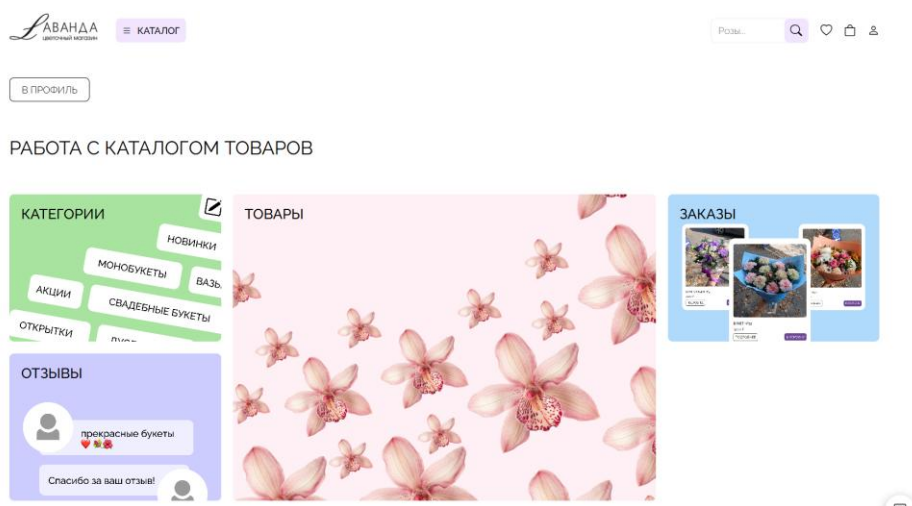


Рисунок 30. «Аккаунт флориста».

5. Модуль корзины

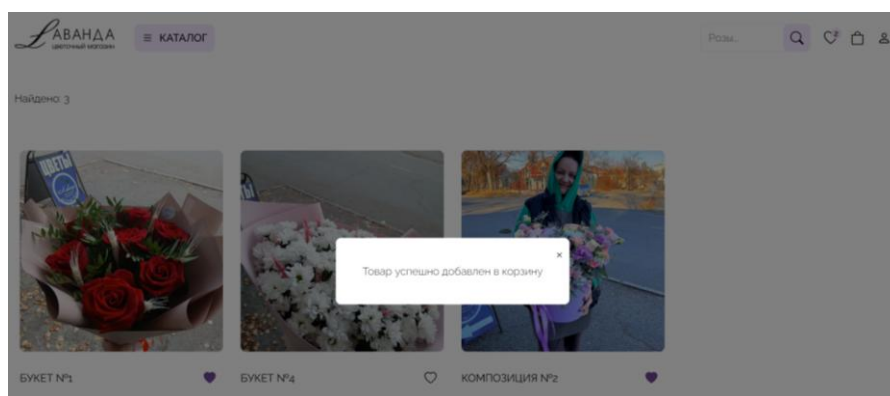


Рисунок 32. Страница «Каталог». Добавляем товар в корзину, после успешного добавления получаем соответствующее сообщение

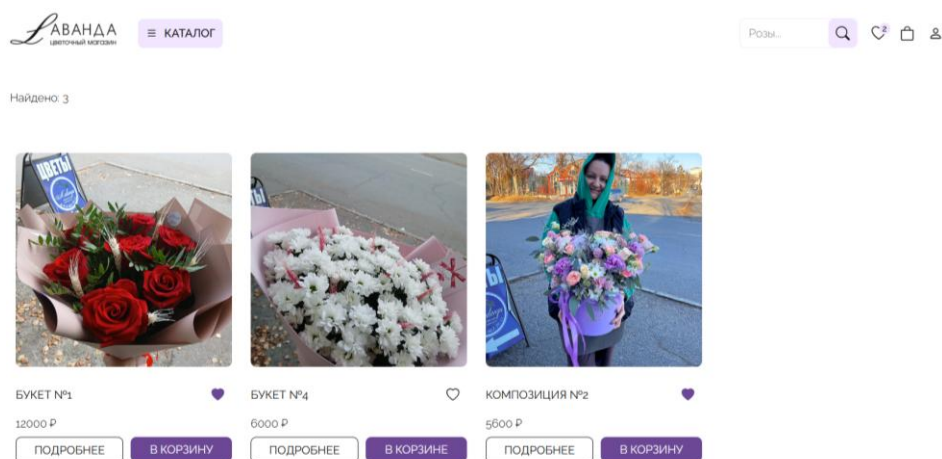


Рисунок 33. После добавления товара в корзину надпись на кнопке меняется с «В корзину» на «В корзине»

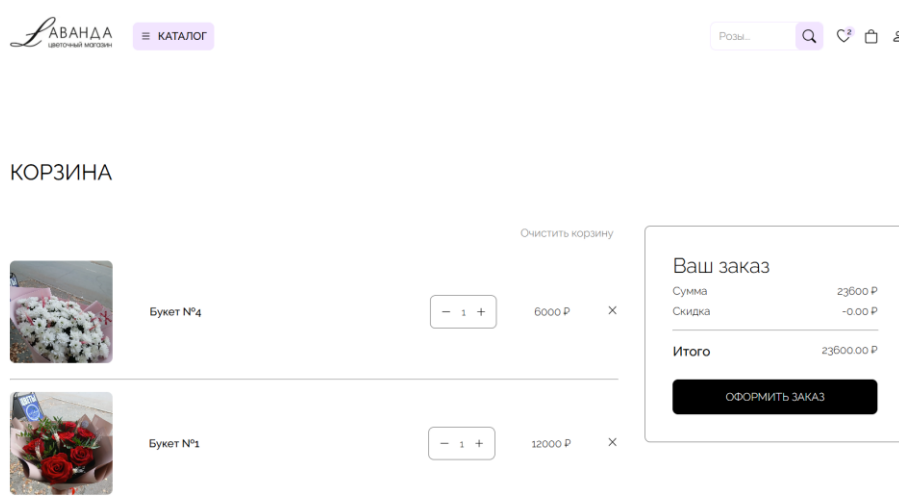


Рисунок 34. Отображение всех товаров, добавленных в корзину

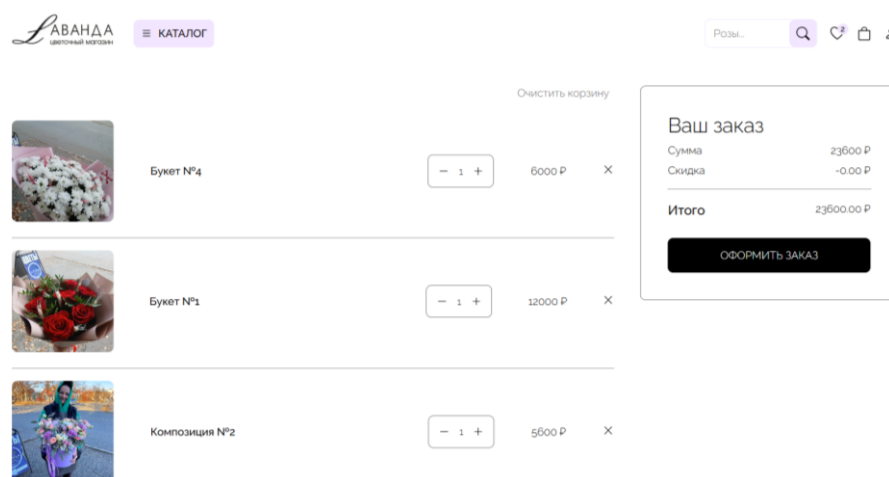
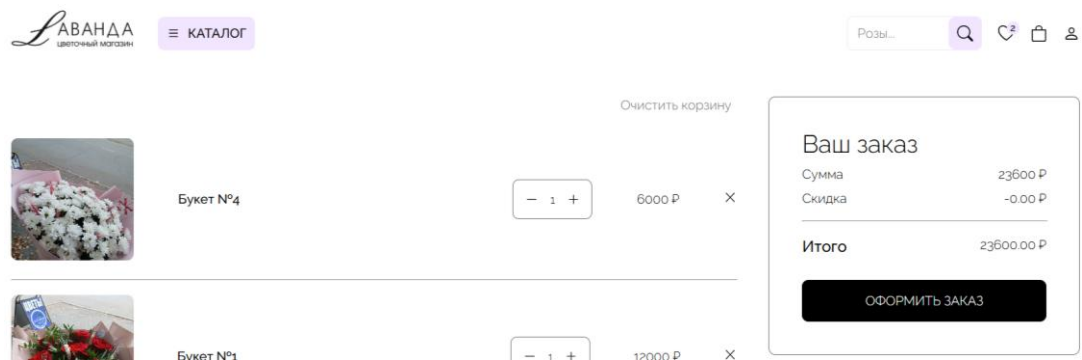


Рисунок 35. Удаление одного товара из корзины с помощью нажатия крестика у соответствующей позиции – как итог, удаление (см.скрин выше)



КОРЗИНА

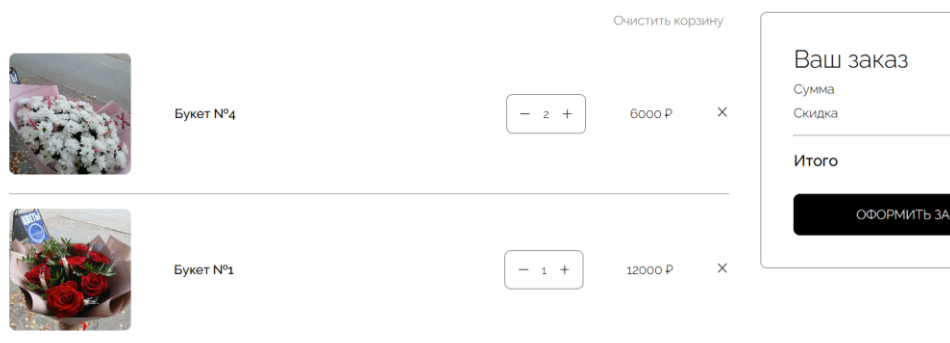


Рисунок 36. Увеличение «Букет №4» на 1 при помощи нажатия на «+»

КОРЗИНА

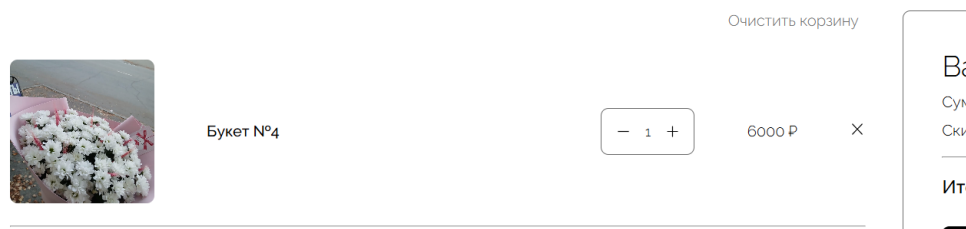


Рисунок 37. Итог после уменьшения позиции «Букет №4» на 1 при помощи нажатия на «-»

КОРЗИНА

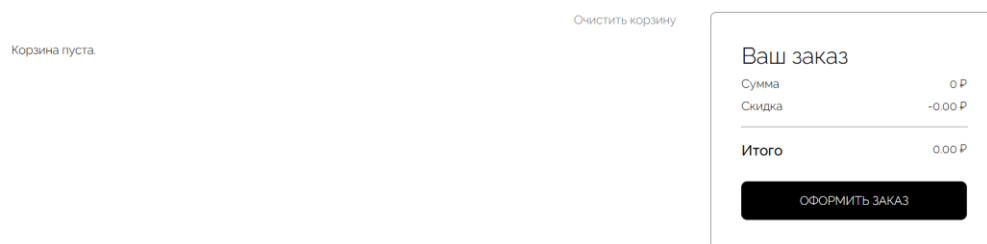


Рисунок 38. Итог после нажатия на «Очистить корзину»

7. Модуль избранного

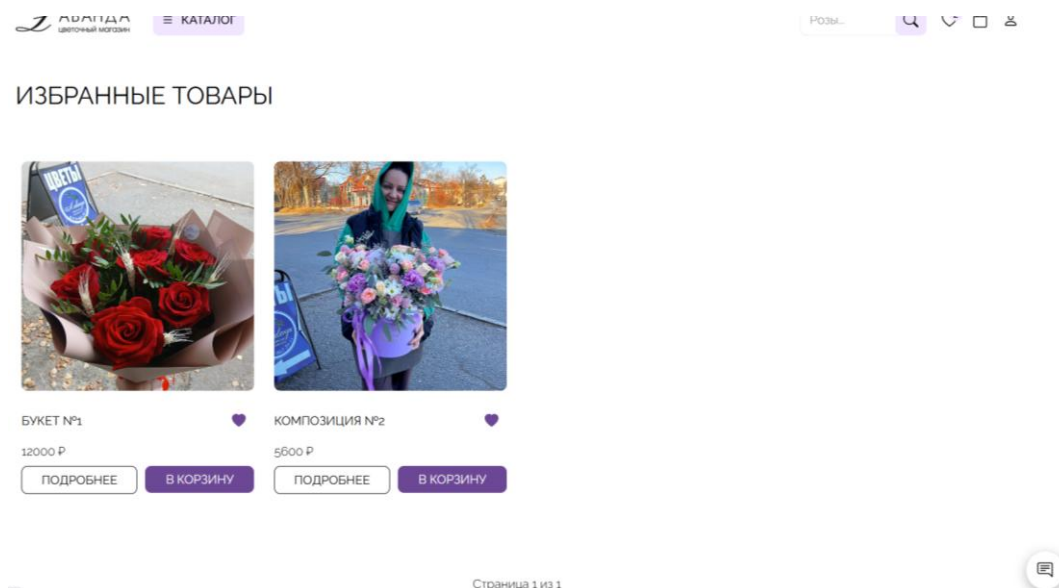


Рисунок 39. Страница «Избранное» с отображением всех товаров, которые добавляли в «избранное»

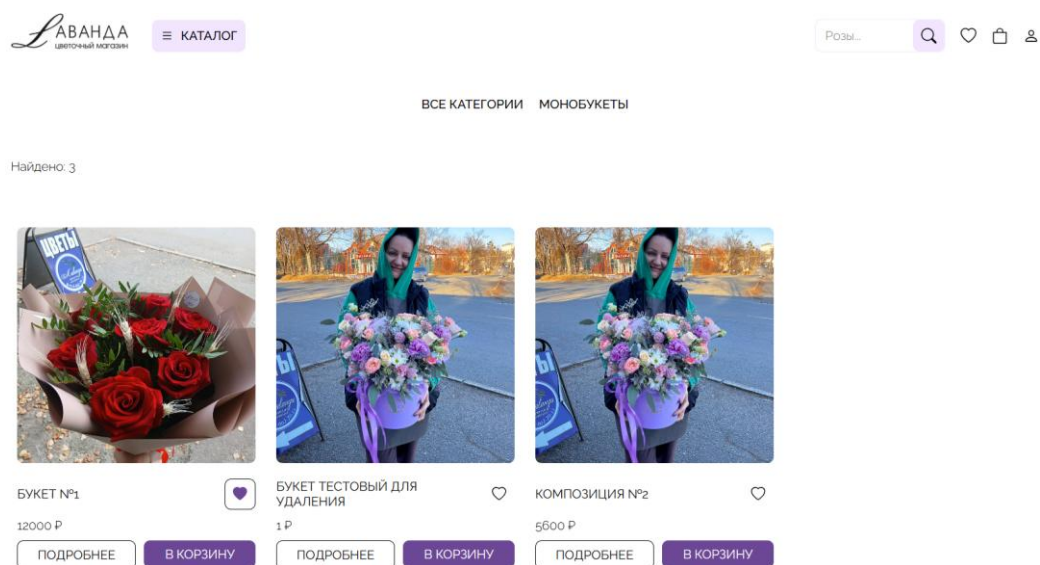


Рисунок 40. Добавление товара в «избранное»;

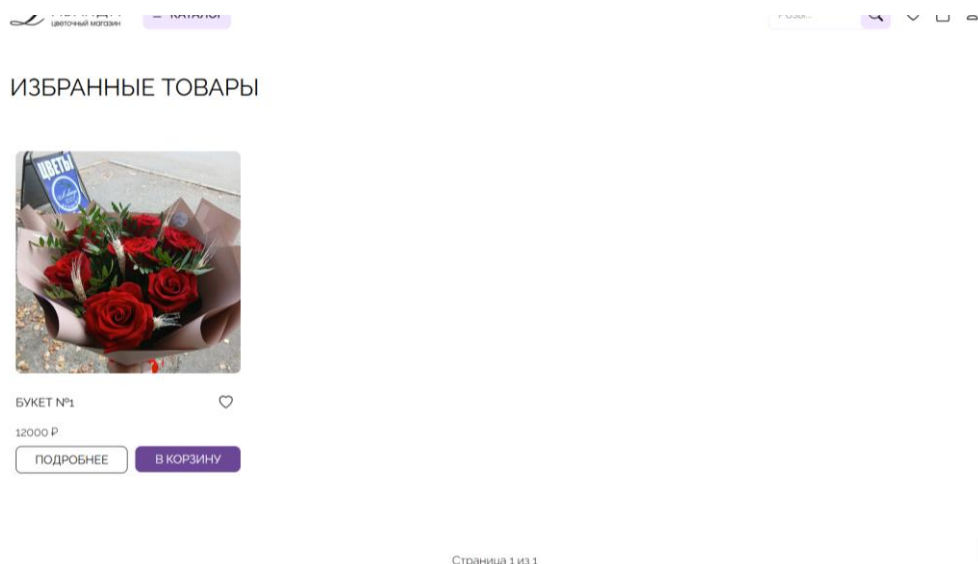


Рисунок 41. Удаление товара из «избранного»;

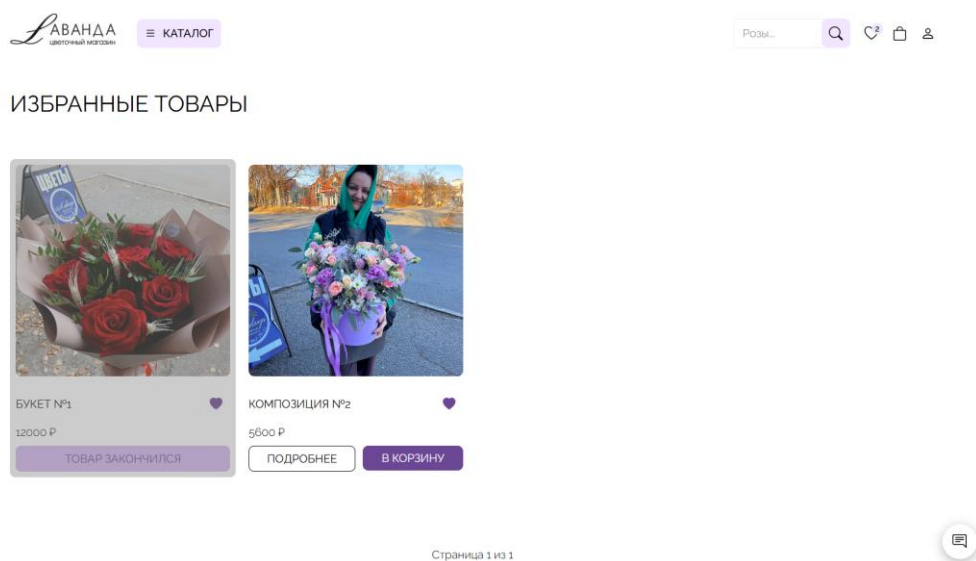


Рисунок 42. Сохранение товара в избранном с пометкой и невозможностью добавить данный товар в корзину, после того как флорист или админ добавили данный товар в архив.

Заключение

В ходе выполнения курсовой работы был проведён анализ работы классических сайтов цветочных магазинов, исходя из которого были выявлены и сформированы требования к разрабатываемому веб-приложению.

Исходя из выбранной архитектуры и наложенных ограничений были сформированы требования к используемым технологиям внутри модулей. Была спроектирована архитектура данных, программная и системная архитектура в виде набора диаграмм в нотации UML.

Опираясь на выше изложенные требования и стек технологий было разработано веб-приложение и пользовательский интерфейс в рамках дисциплины «Web-программирование».

Таким образом, все поставленные ранее цели были выполнены.

Разработанное приложение является результатом данной курсовой работы.

Список использованной литературы

1. Мартин Фаулер - Архитектура корпоративных программных приложений. Издательский дом "Вильямс". 2006 г.
2. Флэнаган, Дэвид. JavaScript. Полное руководство, 7-е изд. : Пер. с англ. — СПб. : ООО “Диалектика”, 2021. — 720 с .
3. Янг А., Мек Б., Кантелон М. Node.js в действии. 2-е изд. — СПб.: Питер, 2018. — 432 с.
4. Браун И. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript. 2-е издание. — СПб.: Питер, 2021. — 336 с.
5. Современный учебник JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/>. – Дата доступа: 04.05.2021.

УНИВЕРСИТЕТ ИТМО

АННОТАЦИЯ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)

Студент Гонос А.С.

(Фамилия, И.О.)

Факультет Информационных технологий и программирования

Кафедра Информационных систем Группа М3307

Направление (специальность) 09.03.02 «Разработка программного обеспечения / Software Engineering»

Руководитель Григорьев Г. С., Университет ИТМО, преподаватель (квалификационная категория "преподаватель практики")

(Фамилия, И.О., место работы, должность, ученое звание, степень)

Дисциплина Web-программирование

Наименование темы Разработка веб-приложения «Цветочный магазин “Лаванда”»

ХАРАКТЕРИСТИКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)

1. Цель и задачи работы

☐ Предложены студентом

☐ Сформулированы при участии студента

☐ Определены руководителем

Цель: Разработать веб-приложение.

Задачи:

1) Сформировать функциональные требования к веб-приложению.

2) Проанализировать функциональные возможности и выбрать набор средств автоматизации.

3) Сформировать функциональную и информационную архитектуру решения.

2. Характер работы

☐ Расчет

☐ Конструирование

☐ Моделирование

☐ Другое,

4. Содержание работы

1) Определение основных понятий

2) Анализ и моделирование процессов

3) Анализ средств автоматизации процессов

4) Проектирование архитектуры ИС

5) Реализация пользовательского интерфейса

5. Выводы

Студент _____

(подпись)

Руководитель _____

(подпись)

« _____ » _____ 2025 г.

УНИВЕРСИТЕТ ИТМО

О Т З Ы В РУКОВОДИТЕЛЯ

о выполнении курсового проекта (работы)

Студент Гонос А.С.

(Фамилия, И.О.)

Факультет Информационных технологий и программирования

Кафедра Информационных систем Группа М3307

Направление (специальность) 09.03.02 «Разработка программного обеспечения / Software Engineering»

Руководитель Григорьев Г. С., Университет ИТМО, преподаватель (квалификационная категория "преподаватель практики")

(Фамилия, И.О., место работы, должность, ученое звание, степень)

Дисциплина Web-программирование

Наименование темы Разработка веб-приложения «Цветочный магазин “Лаванда”»

ОЦЕНКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)

№ п/п	Показатели	Оценка			
		5	4	3	0*
1.	Способность к работе с литературными источниками, справочной литературой, Интернет-ресурсами и т. п.				
2.	Использование иностранных источников				
3.	Способность к анализу и обобщению информационного материала				
4.	Владение базовыми знаниями в профессиональной области				
5.	Владение базовыми знаниями в смежных областях				
6.	Владение навыками решения технических задач				
7.	Способность применять знания на практике				
8.	Уровень и корректность использования в работе методов численного моделирования, инженерных расчетов и статистической обработки данных				
9.	Владение навыками использования современных пакетов компьютерных программ и технологий				
10.	Владение навыками оформления отчетных материалов с применением современных пакетов программ				
11.	Качество оформления пояснительной записки (общий уровень грамотности, стиль изложения, качество иллюстраций, корректность цитирования и пр.**)				
12.	Качество оформления презентации				
13.	Владение навыками публичного выступления и межперсональной коммуникации				
14.	Владение навыками планирования и управления временем при выполнении работы				
ИТОГОВАЯ ОЦЕНКА					

* - не оценивается (трудно оценить)

**Отмеченные
достоинства:** _____

**Отмеченные
недостатки:** _____

Заключение _____

Руководитель _____
(подпись)

Дата « _____ » _____ 2025 г.