



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение*

*Высшего образования*

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

---

---

Институт информационных технологий (ИИТ)

Кафедра Математического обеспечения и стандартизации

информационных технологий (МОСИТ)

Техническое задание программного продукта

Выполнили студенты: Мишанин А.Д.

Ройтман А.А.

Сиротина Д.А.

Шкитырь К.В.

Группа: ИКБО-13-23

Москва 2025

# СОДЕРЖАНИЕ

1 Введение .....	3
2 Основания для разработки .....	3
3 Назначение разработки .....	4
4 Требования к программе .....	4
4.1 Функциональные требования .....	4
4.2 Требования к надёжности .....	4
4.3 Требования эксплуатации .....	6
4.4 Требования к совместимости.....	7
5 Требования к интерфейсу .....	7
6 Критерии приёмки .....	8
7 Требования к документации .....	8
8 Порядок контроля и приёмки .....	9
9 Этапы и сроки разработки .....	10

## **1 Введение**

«Bakery Shop» — веб-приложения для интернет-магазина выпечки - полнофункциональное решение, предназначенное для автоматизации и упрощения процессов выбора товаров, оформления заказов. Ключевая функциональность включает управление учётной записью (регистрация, аутентификация, редактирование профиля), просмотр каталога, детализацию товаров, управление корзиной, а также оформление заказа. Тестирование приложения будет проводиться по методу «чёрного ящика», при котором проверка работы системы осуществляется исключительно на основе входных данных и получаемых результатов без доступа к внутреннему коду, с фокусом на пользовательские сценарии и соответствие заявленным требованиям.

## **2 Основания для разработки**

Разработка веб-приложения «Bakery Shop» осуществляется в рамках учебного фуллстек-проекта и преследует три ключевые цели:

1)Практическая: Создание работающего прототипа приложения с полным циклом.

2)Обучающая: Моделирование и реализация стандартной функциональности современного интернет-магазина.

3)Контроль качества: Проверка стабильности, надёжности и безопасности работы приложения через комплексное тестирование пользовательских сценариев.

### **3 Назначение разработки**

Разрабатываемая система призвана решить следующие задачи:

- 1) Автоматизация продаж: Предоставить инструмент для полностью автоматического оформления онлайн-заказов.
- 2) Повышение эффективности: Сократить время, затрачиваемое пользователем на процесс покупки. Ключевой метрикой является выполнение полного цикла от момента добавления последнего товара в корзину до получения подтверждения успешной оплаты не более чем за 3 минуты.
- 3) Удобство пользователей: Обеспечить клиентам простой и интуитивно понятный личный кабинет для управления своей контактной информацией, данными для доставки и историей заказов.

### **4 Требования к программе**

#### **4.1 Функциональные требования**

##### **1) Пользователь и профиль**

- Регистрация: Возможность создания учётной записи с использованием email-адреса и пароля. Система должна проводить валидацию вводимых данных (корректность email, сложность пароля) и отображать понятные сообщения об ошибках. Также система должна блокировать попытки создать разные профили с одним и тем же email.

- Авторизация: Процедура входа в систему по учётным данным. Механизм сессий или JWT-токенов должен обеспечивать автоматический выход после периода бездействия пользователя.

- Управление профилем: Авторизованный пользователь должен иметь возможность просматривать и редактировать все перечисленные данные своего профиля: имя, номер телефона, список адресов доставки, пароль.

- Удаление профиля: У авторизованных пользователей должна быть возможность безвозвратного удаления своего профиля в разделе «Параметры».

## 2) Каталог и товары

- Страница каталога: Отображение списка товаров в виде сетки или списка. Каждый товар в списке должен содержать: название, цену, изображение (превью).

- Карточка товара: Отдельная страница с полным описанием товара, включая: галерея изображений (одно или несколько фото), подробное описание и состав, вес или объем продукта, актуальная цена.

- Поле для выбора количества товара: при нажатии на кнопку «+» должно изменяться количество добавленного товара в корзине и в её иконке в правом верхнем углу.

- Кнопка «Добавить в корзину»: должна быть неактивна, если товара нет в наличии.

- Кнопка «Загрузить ещё»: при нажатии на кнопку должен загружаться следующий товар.

## 3) Корзина и оформление заказа

- Страница корзины: Отображение списка всех добавленных товаров с возможностью: изменения количества для каждой позиции, удаления любой позиции, автоматического пересчёта промежуточных сумм и общей стоимости заказа.

- Актуальность данных: На момент оформления заказа система обязана провести проверку актуальности цен и наличия товаров. Пользователь должен быть уведомлен об изменениях, если они произошли с момента добавления товара в корзину.

- Создание заказа: После проверки и оформления заказа должно появиться сообщение об успешном выполнении данного действия. После этого корзина

должна очиститься, а вся информация о заказе обязана появиться в истории заказов.

#### 4) Служебные функции

- Обработка ошибок: Все возможные ошибки (сетевые, серверные, ошибки валидации) должны быть обработаны. Пользователю должно показываться понятное сообщение на естественном языке, а не технический код ошибки.

- Логирование: В системе должны вестись логи (журналы аудита) для регистрации ключевых действий пользователей и системных событий, таких как: создание заказа, попытка оплаты, смена пароля, вход/выход из системы.

### 4.2 Требования к надёжности

1) Целостность данных: Пользователь всегда должен видеть актуальное состояние своей корзины и историю заказов, независимо от устройства, с которого он выполнил вход.

2) Доступность: В условиях учебной среды целевой показатель доступности составляет 99%.

3) Устойчивость к сбоям: Критические операции должны быть защищены от потери данных в случае сетевых сбоев.

### 4.3 Требования эксплуатации

1) Браузеры: Приложение должно корректно работать в последних версиях популярных веб-браузеров: Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge.

2) Сетевое окружение: Приложение должно быть оптимизировано для работы в условиях нестабильных мобильных сетей (4G/LTE) и Wi-Fi. Критичные

для бизнеса операции (оформление заказа, оплата) должны сохранять работоспособность при: задержке сети до 300 мс, потере пакетов до 2%.

#### **4.4 Требования к совместимости**

1) Адаптивный интерфейс: Вёрстка должна быть адаптивной и корректно отображаться на всех типах устройств: от мобильных телефонов с минимальным разрешением до desktop-мониторов с разрешением Full HD и выше.

2) Touch-интерфейс: Все интерактивные элементы (кнопки, поля ввода, ссылки) на мобильных устройствах должны быть удобны для управления с помощью касаний.

### **5 Требования к интерфейсу**

1) Навигация: Единая шапка сайта на всех страницах с ссылками: «Каталог», «Корзина» (с иконкой и цифровым счётчиком количества товаров в ней), «Личный кабинет» (для авторизованных) или «Войти» (для анонимных).

2) Карточка товара: Чистый и современный дизайн, делающий акцент на изображении товара. Кнопка «В корзину» должна быть визуально выделена.

3) Корзина: Интуитивно понятное представление в виде таблицы или списка. Поля для изменения количества должны быть удобными. Итоговая сумма должна быть чётко видна. Обязательно наличие кнопки «Проверка».

4) Сообщения: Сообщения об ошибках и успехе должны быть хорошо заметными, привязанными к соответствующему полю или расположенными вверху формы.

5) Адаптивность: На мобильных устройствах все интерактивные элементы (кнопки, поля) должны иметь минимальный размер 44×44 пикселя для удобного касания. Элементы не должны наезжать друг на друга или обрезаться.

## **6 Критерии приёмки**

Приёмка проекта будет считаться успешной при выполнении следующих условий:

1) Прохождение тестов: Успешное выполнение не менее 95% всех подготовленных тест-кейсов, включая позитивные и негативные сценарии (проверка ошибочных действий пользователя).

2) Корректность статусов заказов: Система должна корректно обрабатывать сценарии как успешной, так и неуспешной оплаты, присваивая заказу соответствующие финальные статусы («Оплачен», «Отменён»).

3) Безопасность сессий: Механизм сессий/токенов должен гарантированно завершать работу по истечении заданного времени бездействия. Повторная отправка успешного платежа за один и тот же заказ не должна создавать дубликаты.

4) Кросс-платформенность: Интерфейс должен быть полностью доступен, функционален и корректен на тестовых устройствах: мобильном (разрешение 360×640) и настольном (разрешение не менее 1366×768).

## **7 Требования к документации**

Для обеспечения возможности запуска и использования проекта должна быть предоставлена следующая документация:

1) README для пользователя: Краткое описание основных функций для конечного пользователя: как зарегистрироваться, выбрать товар, оформить заказ, отредактировать профиль.

2) DOCS для разработчика: Подробное техническое руководство по развёртыванию проекта локально или на тестовом стенде.



Приложение запускается из main.tsx где App обернут в Zustand и BrowserRouter для управления состоянием и маршрутизации.

Главная страница содержит Header, Hero-блок и список популярных товаров из useProductsStore.

Страница авторизации поддерживает вход и регистрацию, хранит JWT в куки и сторе и выводит уведомления через Toast.

Корзина реализована через useCartStore с компонентом CartItem, изменением количества, удалением и подсчётом итоговой суммы.

Страница товара подгружает данные по ID и отображает описание, характеристики и отзывы.

## **8 Порядок контроля и приёмки**

Приёмка будет включать в себя следующие этапы тестирования:

1) Функциональное тестирование: Сквозная проверка всех основных пользовательских сценариев:

- Регистрация, вход, выход
- Просмотр каталога
- Просмотр карточки товара.
- Полный цикл работы с корзиной: добавление, изменение количества, удаление
- Оформление заказа
- Редактирование профиля.

2) Негативное тестирование: Целенаправленная проверка обработки ошибок:

- Ввод неверных данных в формы (некорректный email, слабый пароль).
- Попытка отправить форму с пустыми обязательными полями.
- Попытка ввести email, который уже используется
- Действия с истёкшей сессией

3) UI/UX-тестирование: Верификация адаптивности вёрстки, доступности элементов с клавиатуры, понятности и видимости системных сообщений.

## **9 Этапы и сроки разработки**

Предлагаемый план разработки и тестирования выглядит следующим образом:

Этап 1: Аналитика и уточнение требований — 1 неделя.

- Детализация всех пунктов ТЗ, создание прототипов ключевых экранов.

Этап 2: Проектирование — 1 неделя.

- Разработка детальных UI/UX-поток.

- Проектирование и описание API-контрактов.

Этап 3: Реализация функционала — 2–3 недели.

- Последовательная разработка модулей: каталог → корзина → заказ → оплата → профиль.

Этап 4: Тестирование и исправление ошибок — 1–2 недели.

- Проведение функционального, негативного и нагрузочного тестирования.

- Исправление выявленных багов и недочётов.

Этап 5: Итоговая приёмка и релиз — 1 неделя.

- Финальный прогон тестов, подготовка документации, развёртывание на production-сервере.