

Техническое задание (ТЗ)

Проект: Система учёта продаж (mini-POS)

1. Цель проекта

Создать веб-приложение для магазина/кафе, позволяющее:

- администратору управлять товарами и кассирами;
- кассиру пробивать чеки, учитывать оплату, генерировать PDF и QR-коды чеков;
- администратору получать аналитику по продажам.

2. Пользовательские роли

1. Администратор (Admin)

- Управление кассирами (создание, блокировка, назначение прав).
- Управление товарами: добавление, редактирование, архивирование (is_active = False).
- Настройка скидок/акций.
- Просмотр всех чеков.
- Получение аналитики (графики продаж, отчёты).

2. Кассир (Cashier)

- Просмотр списка активных товаров.
- Создание чека (выбор товаров, количество, скидки).
- Фиксация оплаты (наличные, карта, смешанная оплата).
- Автоматический расчёт сдачи.
- Экспорт чека в **PDF**.
- Генерация **QR-кода** для чека.
- Просмотр истории своих чеков.

3. Основные функции

3.1 Управление товарами

- Поля товара: sku, название, описание, цена, фото, скидка %, is_active.
- Автоматический расчёт цены со скидкой.
- Возможность архивировать товары (перенос в неактивные).

3.2 Чеки

- Формирование чека: список товаров, количество, цены, скидки.
- Автоматический пересчёт суммы.
- Учёт оплаты: метод (нал/карта/смешанный), сумма оплаты, сдача.
- Генерация **PDF-чека**.
- Генерация **QR-кода** на чек (например, с ссылкой или данными чека).
- Хранение истории чеков.

3.3 Отчёты и аналитика

- API для получения статистики продаж (кол-во чеков и сумма по дням).
- Возможность строить графики (через Chart.js на фронте).
- Фильтры по датам, по кассиру.

4. Технические требования

- **Backend:** Django + DRF
- **База данных:** PostgreSQL (или SQLite для разработки)
- **Аутентификация:** JWT (djangorestframework-simplejwt)
- **Медиафайлы:** хранение фото товаров, PDF чеков, QR-кодов
- **Форматы экспорта:** PDF (reportlab), QR (qrcode)
- **Фильтрация и поиск:** django-filter, DRF search
- **Frontend:** React+Redux

5. API (REST)

1. /api/pos/products/ — CRUD для товаров (админ), просмотр (кассир).
2. /api/pos/receipts/ — создание и просмотр чеков.
3. /api/pos/receipts/sales_report?days=30 — отчёт по продажам (только админ).

6. Дополнительно

- Логирование действий (создание чека, изменение товара).
- Архивирование неактивных товаров.
- Возможность восстановления архива.
- Массовая загрузка товаров (опционально).

7. Нефункциональные требования

- Простота интерфейса.
- Защита API (доступ по JWT).
- Производительность: до 10k чеков в день.
- Возможность масштабирования (перенос генерации PDF/QR в Celery).

ТЗ (Backend)

1. Архитектура проекта

- **Django + DRF**
- **PostgreSQL** (или SQLite для разработки)
- **JWT аутентификация** (djangorestframework-simplejwt)
- **Модули проекта:**
 1. users — пользователи (админ, кассир)
 2. products — товары
 3. receipts — чеки

2. Пользователи (users)

Модель User

- email (уникальный, логин)
- role (admin | cashier)
- is_active
- password

Функционал

- Админ может создавать кассиров.
- Кассир сам регистрироваться не может.
- JWT логин/логаут.

3. Товары (products)

Модель Product

- sku (уникальный код товара)
- name (название)
- description
- price (цена без скидки)
- discount (в процентах, по умолчанию 0)
- image (фото)
- is_active (активный/архив)
- created_at, updated_at

API

- **Админ:** CRUD товаров.
- **Кассир:** только GET /products/ (список, поиск, фильтрация).

4. Чеки (receipts)

Модель Receipt

- cashier (FK → User)
- total_amount (сумма с учётом скидок)
- payment_method (cash | card | mixed)
- paid_amount (внесено клиентом)
- change (сдача, если наличка)
- created_at

Модель ReceiptItem

- receipt (FK → Receipt)
- product (FK → Product)

- quantity
- price (цена за единицу на момент покупки)
- discount (активная скидка на момент покупки)
- final_price (итог за позицию = qty * price - discount)

5. Бизнес-логика

- При создании чека:
 1. Кассир выбирает товары и количество.
 2. Система считает сумму со скидками.
 3. Указываем способ оплаты и сумму, внесённую клиентом.
 4. Система считает сдачу.
 5. Чек сохраняется и можно выгрузить в PDF.
 6. Генерируется QR-код (с ID чека или ссылкой).

6. API (Backend)

1. /api/auth/login/ — логин (JWT)
2. /api/users/cashiers/ — CRUD кассиров (только админ)
3. /api/products/ — список товаров (все), CRUD (только админ)
4. /api/receipts/ — список чеков (для кассира только свои, для админа все)
5. /api/receipts/{id}/pdf/ — скачать чек PDF
6. /api/receipts/{id}/qr/ — получить QR-код чека
7. /api/reports/sales/ — отчёт (сумма продаж за период)

7. Дополнительно

- **PDF** — reportlab
- **QR-коды** — qrcode
- **Фильтрация** (django-filter):
 - товары: по имени, активным, цене
 - чеки: по дате, кассиру