

# Технический аудит кода и архитектуры

## «MarketFlow»

Дата: 23.05.2026

Аудитор: Олег Камчатный, СТО

Заказчик: ООО «МаркетФлоу» (образец)

Стек: Vue/Nuxt · NestJS · PostgreSQL · Docker

### РЕЗЮМЕ ДЛЯ РУКОВОДСТВА

**Общая оценка:** ☐ проект работоспособен, но есть критичные риски безопасности и накопленный техдолг, который уже замедляет релизы и удорожает поддержку.

**Рекомендация:** стабилизировать — не переписывать с нуля, но в ближайшие 30 дней закрыть критичные уязвимости и настроить CI/CD.

- Секреты и API-ключи в репозитории — риск компрометации платёжной интеграции.
- Отсутствие автоматических бэкапов БД — риск потери заказов и клиентских данных.
- Монолитный backend-модуль заказов (~2 400 строк) — каждый релиз ломает смежные сценарии.

### ОБЗОР ПРОЕКТА

SaaS-платформа для автоматизации маркетинговых воронок: личный кабинет, каталог услуг, оплата через ЮKassa, интеграция с AmoCRM и Telegram-уведомлениями. Кодовая база ~18 000 строк, 2 репозитория (frontend/backend), один production-сервер на VPS.

| Параметр         | Значение                                   |
|------------------|--------------------------------------------|
| Backend          | NestJS 10, TypeORM, PostgreSQL 15          |
| Frontend         | Nuxt 3, Pinia, SSR                         |
| Деплой           | Docker Compose, ручной SSH-деплой          |
| Интеграции       | ЮKassa, AmoCRM, SendGrid, Telegram Bot API |
| Покрытие тестами | ~12% (критичные модули без тестов)         |

**BACKEND**

| Проверка                         | Статус |
|----------------------------------|--------|
| Структура модулей NestJS         | OK     |
| Валидация входящих данных (DTO)  | OK     |
| Авторизация JWT + refresh-токены | MED    |
| Секреты в .env / CI variables    | CRIT   |
| Логирование и трассировка ошибок | HIGH   |
| Unit/integration тесты           | HIGH   |

**Ключевые замечания:** файл `config/payments.ts` содержит live-ключ ЮKassa; refresh-токены хранятся без rotation; модуль `orders.service.ts` перегружен бизнес-логикой и прямыми SQL-запросами.

**FRONTEND**

| Проверка                         | Статус |
|----------------------------------|--------|
| Архитектура Nuxt / composables   | OK     |
| Обработка ошибок API             | MED    |
| Хранение токенов (localStorage)  | HIGH   |
| Core Web Vitals / SSR-гидратация | MED    |
| ESLint + Prettier в CI           | LOW    |

**БАЗА ДАННЫХ И ИНФРАСТРУКТУРА**

PostgreSQL: отсутствуют индексы на поля `orders.user_id` и `orders.created_at` — при росте базы запросы дашборда деградируют. Бэкапы выполняются вручную, восстановление не

тестировалось. Деплой — SSH + docker compose pull, без staging-окружения.

## БЕЗОПАСНОСТЬ

- CRIT — API-ключи в Git-истории (коммит от 14.01.2025, ветка main).
- HIGH — CORS разрешает все origin на /api/admin/.\*
- HIGH — Rate limiting отсутствует на /auth/login и /payments/webhook.
- MED — SSL настроен, но HSTS и автообновление сертификата не проверены.

КАРТА ТЕХНИЧЕСКОГО ДОЛГА

| Зона                     | Влияние на бизнес                                 | Сложность |
|--------------------------|---------------------------------------------------|-----------|
| Модуль заказов (backend) | Каждый релиз — риск регрессий в оплате            | Высокая   |
| Auth / tokens            | Утечка сессий, сложный logout на всех устройствах | Средняя   |
| CI/CD отсутствует        | Долгие релизы, человеческий фактор                | Средняя   |
| Бэкапы                   | Потеря данных = простой и репутационный удар      | Низкая    |

ПЛАН ДЕЙСТВИЙ НА 30 ДНЕЙ

| Неделя | Задачи                                                               | Приоритет |
|--------|----------------------------------------------------------------------|-----------|
| 1      | Ротация всех секретов; вынос в CI/Vault; git filter-repo для истории | CRIT      |
| 1–2    | Автобэкапы PostgreSQL + тест восстановления на staging               | CRIT      |
| 2      | Rate limit + CORS whitelist; refresh token rotation                  | HIGH      |
| 3      | CI pipeline: lint → test → build → deploy на staging                 | HIGH      |
| 4      | Рефакторинг orders.service: выделить payment/webhook слои            | MED       |
| 4+     | Индексы БД; мониторинг (uptime + error tracking)                     | MED       |

ИТОГОВОЕ ЗАКЛЮЧЕНИЕ

Проект «MarketFlow» можно развивать без полной переписки, но текущие риски безопасности и отсутствие автоматизации уже стоят денег: каждый релиз занимает 4–6 часов, а инцидент с утечкой ключей может остановить приём платежей. Рекомендуем начать с недели 1 плана — это минимальный порог для спокойного масштабирования.