

# 🍷 GoGoMarket - Итоговый сводный отчет

Дата формирования отчета: 4 января 2026, 16:50

Проект: GoGoMarket (Социальный видео-маркетплейс)

URL: <http://64.226.94.133>

GitHub: <https://github.com/buranovt2025-jpg/gogomarket>

Ветка: devin/1767373941-gogomarket-mvp

---

## 📋 EXECUTIVE SUMMARY

### Ключевые выводы

✓ **Что работает:** - Backend API частично функционирует (45% готовности) - Базовая структура проекта создана с использованием современных технологий - Flutter мобильное приложение скомпилировано (APK 66 MB) - Веб-приложение доступно и отвечает на запросы - Регистрация и базовая аутентификация работают - Просмотр каталога товаров функционирует

✗ **Критические проблемы:** 1. **Package name** содержит **"example"** - не production ready 2. **Target SDK 36 (Android 16)** - версия Android еще не выпущена 3. **Отсутствие HTTPS** - 4 небезопасных запроса обнаружено 4. **API Backend** недоступен на порту 3000 (404 ошибки) 5. **Корзина не работает** - товары не добавляются 6. **Функционал курьера полностью отсутствует** 7. **Видео-плеер не загружается** - Short видео не открываются 8. **Сессия не сохраняется** при перезагрузке страницы

📊 **Готовность проекта:** | Источник | Процент | Метод оценки | |  
-----|-----|-----| | GAP\_ANALYSIS.md | **38%** | Документация (53/139 функций) | | Реальное тестирование | **45%** |  
Функциональное тестирование | | **Разница** | **+7%** | Некоторые функции работают без документации |

🕒 **Оценка времени до MVP:** - **Минимум:** 4-6 недель (при полной занятости 2-3 разработчиков) - **Реалистично:** 8-12 недель (с учетом тестирования и исправлений) - **С текущей скоростью:** 16+ недель

🚫 **Готовность к Production: НЕТ ✗**

---

# АНАЛИЗ МОБИЛЬНОГО ПРИЛОЖЕНИЯ (APK)

## 1.1 Общая информация

Файл: gogomarket-business-cycle.apk  
Размер: 66 MB  
Дата компиляции: 4 января 2026

Параметр	Значение	Статус
Package Name	com.example.gogomarket	⚠ ПРОБЛЕМА
Version Name	1.0.0	✓ OK
Version Code	1	✓ OK
Min SDK	24 (Android 7.0)	✓ OK
Target SDK	36 (Android 16)	⚠ КРИТИЧНО
Compile SDK	36	⚠ КРИТИЧНО
Технология	Flutter	✓ OK
Размер APK	66 MB	⚠ Большой

## 1.2 Детальный анализ разрешений

#	Разрешение	Критичность	Назначение
1	CAMERA	⚠ HIGH	Сканирование QR-кодов для заказов Фото товаров продавцами
2	USE_BIOMETRIC	⚠ MEDIUM	Биометрическая аутентификация Безопасная оплата через Face ID
3	USE_FINGERPRINT	⚠ MEDIUM	Отпечаток пальца для входа Подтверждение транзакций
4	RECORD_AUDIO	⚠ HIGH	Неясно - не требуется для маркетплейса Проверка подключения к

5	ACCESS_NETWORK_STATE	LOW	интернету Управление режимом offline
6	INTERNET	HIGH	Обязательно для работы приложения API запросы, загрузка медиа

**Рекомендация:** Удалить разрешение RECORD\_AUDIO из AndroidManifest.xml, если не планируется функционал записи видео с аудио.

### 1.3 Технологические детали

**Фреймворк:** Flutter (современный кросс-платформенный фреймворк)

**Используемые Flutter пакеты:**

- cupertino\_icons      *# iOS-стиль иконок*
- wakelock\_plus      *# Предотвращение блокировки экрана при просмотре видео*
- image\_picker      *# Выбор/съемка фото для товаров*
- url\_launcher      *# Открытие внешних ссылок (поддержка, соцсети)*

**ML/AI компоненты (Google ML Kit):** - Barcode Scanning - Сканирование штрих-кодов товаров - TensorFlow Lite модели: - barcode\_ssd\_mobilenet\_v1\_dmp25\_quant.tflite (4.2 MB) - oned\_auto\_regressor\_mobile.tflite - oned\_feature\_extractor\_mobile.tflite

**Поддерживаемые архитектуры:** - arm64-v8a (64-bit ARM) - современные устройства - armeabi-v7a (32-bit ARM) - старые устройства - x86\_64 - эмуляторы Android

### 1.4 Структура и размеры компонентов

Компонент	Размер	Процент	Описание
Smali (код)	77 MB	52%	Декомпилированный Java/Kotlin код
Native libs	63 MB	43%	Flutter engine, ML модели
Resources	2.8 MB	2%	Изображения, иконки, макеты
Assets	1.4 MB	1%	Шрифты, конфигурации

<b>ИТОГО</b>	~144	100%	-
<b>(распакованный)</b>	MB		

---

#### Основные нативные библиотеки:

libflutter.so - 11 MB (Flutter engine)  
libapp.so - 6.5 MB (Dart код приложения)  
libbarhopper\_v3.so - 4.2 MB (ML Kit для штрих-кодов)

## 1.5 🎯 Критические проблемы APK

### Блокер #1: Target SDK 36 (Android 16 не существует)

Серьезность: 🎯 КРИТИЧЕСКАЯ

Приоритет: P0 (должно быть исправлено немедленно)

Проблема:

APK собран с targetSdkVersion="36" и compileSdkVersion="36"  
Android 16 (API 36) еще не выпущен (текущая версия - Android 15, API 35)

Последствия:

- ✗ Google Play Store отклонит приложение при публикации
- ✗ Приложение может работать нестабильно
- ✗ Невозможно протестировать на реальных устройствах

Решение:

- Изменить в android/app/build.gradle:  

```
compileSdkVersion 34 // или 35 для Android 15  
targetSdkVersion 34 // или 35
```
- Пересобрать APK:  

```
flutter clean  
flutter build apk --release
```
- Протестировать на реальном устройстве

Время на исправление: 1 час (+ 1 час на тестирование)

### Проблема #2: Package name содержит "example"

Серьезность: 🎯 ВЫСОКАЯ

Приоритет: P1 (критично для production)

Проблема:

Package name = "com.example.gogomarket"  
"example" - это дефолтное значение для тестовых приложений

**Последствия:**

- △ Google Play может отклонить как тестовое приложение
- △ Возможны конфликты с другими "example" приложениями
- △ Выглядит непрофессионально

**Решение:**

Переименовать package name на один из вариантов:

- uz.gogomarket.app (рекомендуется)
- com.gogomarket.uz
- app.gogomarket.uz

**Шаги:**

1. Использовать плагин change\_app\_package\_name:  
`flutter pub run change_app_package_name:main uz.gogomarket.app`
2. Вручную проверить файлы:
  - android/app/build.gradle
  - android/app/src/main/AndroidManifest.xml
  - android/app/src/main/kotlin/.../MainActivity.kt

3. Пересобрать проект

Время на исправление: 2-3 часа (+ тестирование)

### **Проблема #3: Неоправданное разрешение RECORD\_AUDIO**

Серьезность: 🟡 СРЕДНЯЯ

Приоритет: P2 (желательно исправить)

**Проблема:**

Запрашивается разрешение на запись аудио без явной необходимости  
Маркетплейс не требует записи голоса

**Последствия:**

- △ Пользователи могут отказать в установке из-за подозрительного разрешения
- △ Google Play может запросить объяснение использования
- △ Увеличивает поверхность атаки для malware

**Решение:**

Удалить из android/app/src/main/AndroidManifest.xml:

```
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
```

Исключение: оставить если планируется:

- Запись видео-отзывов с аудио
- Голосовой поиск товаров
- Видео-звонки с поддержкой

Время на исправление: 30 минут

#### Проблема #4: Большой размер APK (66 MB)

Серьезность: 🟡 СРЕДНЯЯ

Приоритет: P3 (оптимизация)

Проблема:

Размер APK = 66 MB (средний размер маркетплейс приложений: 20-40 MB)

Последствия:

- 👤 Пользователи могут отказаться от установки
- 📶 Долгая загрузка при медленном интернете (актуально для Узбекистана)
- 📱 Занимает много места на устройстве

Причины большого размера:

- ML Kit модели для штрих-кодов (4.2 MB)
- Поддержка 3 архитектур (arm64, armv7, x86\_64)
- Flutter engine (~11 MB на архитектуру)

Решение:

1. Использовать Android App Bundle (.aab):  
flutter build appbundle --release  
→ Пользователь скачает только нужную архитектуру (~25-30 MB)
2. Включить ProGuard/R8 минификацию:

```
buildTypes {
  release {
    minifyEnabled true
    shrinkResources true
  }
}
```
3. Оптимизировать ресурсы:
  - Сжать изображения (WebP)
  - Удалить неиспользуемые fonts
4. Lazy load ML Kit модели (скачивать при первом использовании)

Ожидаемый результат: 30-40 MB APK, 20-25 MB через App Bundle

Время на оптимизацию: 1-2 дня

## 1.6 Рекомендации и план действий по APK

🔥 Срочные (P0-P1) - до релиза

- ☐ Изменить target SDK на 34/35 (1 час)
- ☐ Переименовать package name (3 часа)
- ☐ Удалить RECORD\_AUDIO (30 мин)
- ☐ Настроить signing config для production (1 час)
- ☐ Протестировать на реальных устройствах (4 часа)

Общее время: 1 рабочий день

⚠ Важные (P2) - оптимизация

- ☐ Оптимизировать размер APK до 40 MB (2 дня)
- ☐ Перейти на Android App Bundle (.aab) (4 часа)
- ☐ Включить ProGuard/R8 (2 часа)
- ☐ Добавить crash reporting (Sentry/Firebase Crashlytics) (4 часа)

Общее время: 3 рабочих дня

💡 Желательные (P3) - улучшения

- ☐ Добавить Firebase Analytics
- ☐ Интегрировать push-уведомления
- ☐ Настроить dynamic links
- ☐ Добавить in-app updates (Play Core Library)

📊 АНАЛИЗ WEB-ПРИЛОЖЕНИЯ

2.1 Lighthouse Performance Audit

Методология: 4 независимых запуска Google Lighthouse  
Дата тестирования: 4 января 2026  
URL: http://64.226.94.133  
User Agent: Chrome 143.0.0.0 (Mobile simulation - Moto G Power)

Сводная таблица результатов

Тест	Performance	Accessibility	Best Practices	SEO
Run #1	64/100 🟡	81/100 🟢	78/100 🟡	83/100 🟢
Run #2	66/100 🟡	81/100 🟢	78/100 🟡	83/100 🟢
Run #3	60/100 🟡	81/100 🟢	78/100 🟡	83/100 🟢
Run #4	61/100 🟡	81/100 🟢	78/100 🟡	83/100 🟢

Среднее 63/100 81/100 78/100 83/100

Оценка: - 90-100: Отлично - 50-89: Требуется улучшение - 0-49: Плохо

## 2.2 Детальные метрики производительности

Метрика	Значение	Норма	Оценка	Описание
First Contentful Paint (FCP)	4.7s	<1.8s	13/100	Время до первого отображения контента
Largest Contentful Paint (LCP)	7.8s	<2.5s	3/100	Время до загрузки основного контента
Speed Index	4.7s	<3.4s	69/100	Скорость визуальной загрузки страницы
Total Blocking Time (TBT)	N/A	<200ms	-	Время блокировки главного потока
Cumulative Layout Shift (CLS)	N/A	<0.1	-	Стабильность визуального контента
Time to Interactive (TTI)	N/A	<3.8s	-	Время до интерактивности

Вердикт: Производительность неудовлетворительная

## 2.3 Критические проблемы веб-приложения

### Блокер #1: Отсутствие HTTPS

Серьезность: КРИТИЧЕСКАЯ

Приоритет: P0 (блокеры для production)

Затронуты: Все пользователи

Проблема:

Сайт работает только по незащищенному протоколу HTTP

Обнаружено 4 небезопасных запроса:



- `http://64.226.94.133/`
- `http://64.226.94.133/assets/index-CQg83n7K.js`
- `http://64.226.94.133/assets/index-CGG4Q_RY.css`
- `http://64.226.94.133/vite.svg`

Последствия:

- ✗ Браузеры показывают "Not Secure" в адресной строке
- ✗ Пароли и данные карт передаются незашифрованными
- ✗ Уязвимость для Man-in-the-Middle атак
- ✗ Google понижает в поисковой выдаче сайты без HTTPS
- ✗ Невозможно использовать PWA (Progressive Web App)
- ✗ Некоторые API (геолокация, камера) не работают без HTTPS
- ✗ Push-уведомления требуют HTTPS

Решение:

1. Получить SSL сертификат (бесплатно через Let's Encrypt)
2. Настроить Nginx/Apache для HTTPS
3. Перенаправить HTTP → HTTPS (301 redirect)
4. Обновить все внутренние ссылки на `https://`

Команды для установки (Nginx + Let's Encrypt):

```
sudo apt install certbot python3-certbot-nginx
sudo certbot --nginx -d gogomarket.uz -d www.gogomarket.uz
```

Или для DigitalOcean (по IP):

```
# Использовать самоподписанный сертификат или Cloudflare
```

Время на исправление: 2-4 часа

Ответственный: DevOps / Backend разработчик

## Проблема #2: Медленная загрузка страницы

Серьезность: 🟡 ВЫСОКАЯ

Приоритет: P1 (влияет на UX и конверсию)

Проблема:

First Contentful Paint (FCP) = 4.7s (норма < 1.8s)  
Largest Contentful Paint (LCP) = 7.8s (норма < 2.5s)

Последствия:

- 📉 53% пользователей покидают сайт, если загрузка > 3 секунд
- 📉 Каждая дополнительная секунда снижает конверсию на 7%
- 📉 Google понижает в поисковой выдаче медленные сайты

Причины медленной загрузки:

1. Отсутствие CDN для статических ресурсов
2. Изображения не оптимизированы (PNG вместо WebP)
3. Нет сжатия Gzip/Brotli

- 4. Render-blocking JavaScript и CSS
- 5. Отсутствует lazy loading для изображений

Решение:

A) Оптимизация изображений:

- Конвертировать в WebP (70% меньше размер)
- Использовать srcset для разных экранов
- Lazy loading: loading="lazy"

B) Включить сжатие (Nginx):

```
gzip on;  
gzip_comp_level 6;  
gzip_types text/plain text/css application/json  
        application/javascript text/xml application/xml;
```

C) Настроить кэширование:

```
location ~* \.(js|css|png|jpg|jpeg|gif|ico|svg|webp)$ {  
    expires 1y;  
    add_header Cache-Control "public, immutable";  
}
```

D) Использовать CDN:

- Cloudflare (бесплатный план)
- AWS CloudFront
- DigitalOcean Spaces CDN

Ожидаемый результат:

FCP: 4.7s → 1.5s (улучшение на 68%)  
LCP: 7.8s → 2.0s (улучшение на 74%)  
Performance Score: 63 → 90+

Время на оптимизацию: 1-2 дня

Ответственный: Frontend разработчик + DevOps

### Проблема #3: Отсутствует редирект HTTP → HTTPS

Серьезность: 🟡 ВЫСОКАЯ

Приоритет: P1 (после установки SSL)

Проблема:

После настройки HTTPS пользователи всё еще могут заходить по HTTP  
Lighthouse сообщает: "Does not redirect HTTP traffic to HTTPS"

Последствия:

- ⚠ Часть пользователей продолжит использовать небезопасное соединение
- ⚠ Смешанный контент (mixed content) в поисковой выдаче

Решение (Nginx):

```
server {  
    listen 80;  
    server_name gogomarket.uz www.gogomarket.uz;  
    return 301 https://$host$request_uri;  
}
```

Время на исправление: 30 минут

## 2.4 Рекомендации по веб-приложению

### 🚨 Критические (P0-P1)

- ☐ Установить SSL сертификат (4 часа)
- ☐ Настроить HTTPS (2 часа)
- ☐ Настроить HTTP → HTTPS редирект (30 мин)
- ☐ Включить Gzip/Brotli сжатие (1 час)
- ☐ Настроить кэширование статики (1 час)

Общее время: 1-2 дня

### ⚠ Важные (P2)

- ☐ Подключить CDN (Cloudflare) (4 часа)
- ☐ Оптимизировать изображения в WebP (1 день)
- ☐ Добавить lazy loading (4 часа)
- ☐ Минифицировать JS/CSS (2 часа)
- ☐ Настроить мониторинг (Sentry, Google Analytics) (4 часа)

Общее время: 2-3 дня

---

## 🔧 ФУНКЦИОНАЛЬНЫЕ ОШИБКИ

### 3.1 Критические баги (Блокеры Production)

#### BUG-001: API бэкенда недоступен (404)

Приоритет: 🟡 P0

Затронутые роли: Админ, все остальные роли косвенно

Статус: Открыто

Оценка времени: 2-3 дня

Описание:

Запросы к API бэкенда возвращают 404 Not Found

URL: [http://64.226.94.133:3000/api/\\*](http://64.226.94.133:3000/api/*)

Воспроизведение:

1. Войти под админом (admin@gogomarket.uz / admin123)
2. Перейти в раздел "Пользователи"
3. Открыть DevTools → Console
4. Увидеть: GET http://64.226.94.133:3000/api/admin/users 404 (Not Found)

Ожидаемое поведение:

API должен возвращать JSON с списком пользователей

Фактическое:

404 Not Found - эндпоинт не найден

Возможные причины:

1. Backend не запущен на порту 3000
2. Неправильный routing в Express
3. Cors проблемы
4. PM2 процесс упал

Проверка:

```
# SSH на сервер
ssh root@64.226.94.133

# Проверить процессы
pm2 status
pm2 logs backend

# Проверить порт
netstat -tulpn | grep 3000
curl http://localhost:3000/api/health
```

Решение:

Требуется доступ к серверу для диагностики

Зависимости:

- FEAT-010 (Админ не видит пользователей)
- Все функции, использующие API

**BUG-002: Корзина не функционирует**

Приоритет: 🟡 P0

Затронутые роли: Покупатель

Статус: Открыто

Оценка времени: 3-5 дней

Описание:

Невозможно добавить товары в корзину

Кнопка на странице товара ведет в избранное (wishlist) вместо корзины

Воспроизведение:

1. Войти под покупателем
2. Открыть любой товар из каталога
3. Нажать оранжевую кнопку внизу страницы
4. Ожидается: товар добавлен в корзину
5. Факт: происходит переход на /wishlist

Затронутые страницы:

- /products/:id (страница товара)
- /cart (корзина пустая даже после "добавления")

Технические детали:

- Неправильный роутинг в React Router
- Кнопка "Add to Cart" вызывает addToWishlist()
- State management (Redux/Context) не обновляется

Решение:

1. Исправить обработчик кнопки: onClick={handleAddToCart}
2. Проверить Redux actions (addToCart vs addToWishlist)
3. Добавить визуальное подтверждение (тост-уведомление)
4. Реализовать API endpoint POST /api/cart/add

Критичность:

Без корзины невозможны покупки - основная функция маркетплейса!

### **BUG-003: Функционал курьера полностью отсутствует**

Приоритет: 🟡 P0

Затронутые роли: Курьер

Статус: Открыто

Оценка времени: 2-3 недели

Описание:

Роль "Курьер" существует в ТЗ, но не реализована в приложении  
Курьер не может видеть заказы, сканировать QR-коды, обновлять статусы

Отсутствующие функции:

1. Список доступных заказов для доставки
2. Принятие заказа курьером
3. Навигация до адреса доставки
4. QR-сканер для подтверждения получения у продавца
5. QR-сканер для подтверждения доставки покупателю
6. История выполненных доставок
7. Статистика заработка

## 8. Чат с продавцом/покупателем

### Решение:

Требуется полная реализация модуля Курьера:

#### Backend (5-7 дней):

- API endpoints для заказов курьера
- QR-генерация и валидация
- Система статусов доставки
- Push-уведомления

#### Frontend Web (3-4 дня):

- Дашборд курьера
- Карта с заказами
- QR-сканер (WebRTC)

#### Mobile (5-7 дней):

- Экраны для курьера
- Нативный QR-сканер
- Геолокация и навигация

### Критичность:

Без курьеров невозможна доставка заказов!

## **BUG-004: Сессия не сохраняется**

Приоритет: 🟡 P0

Затронутые роли: Все

Статус: Открыто

Оценка времени: 1-2 дня

### Описание:

После перезагрузки страницы пользователь разлогинивается  
JWT token не сохраняется в localStorage/sessionStorage

### Воспроизведение:

1. Войти под любой ролью
2. Обновить страницу (F5)
3. Пользователь разлогинен, перенаправлен на /login

### Ожидаемое поведение:

Сессия должна сохраняться между перезагрузками страницы

### Причина:

- JWT token не сохраняется в localStorage
- Или: token сохраняется, но не проверяется при загрузке
- Или: AuthContext не инициализируется из сохраненного токена

Решение:

Frontend (React):

```
// При логине
localStorage.setItem('jwt_token', response.data.token);

// При загрузке приложения
useEffect(() => {
  const token = localStorage.getItem('jwt_token');
  if (token) {
    validateToken(token).then(user => {
      setAuthState({ user, isAuthenticated: true });
    });
  }
}, []);
```

Backend:

- Добавить endpoint GET /api/auth/me для проверки токена
- Увеличить срок действия JWT (7-30 дней)

Время на исправление: 1 день

### **BUG-005: Short видео не открываются**

Приоритет: 🟡 P1

Затронутые роли: Все

Статус: Открыто

Оценка времени: 2-3 дня

Описание:

При клике на Short видео в ленте плеер не загружается  
Экран остается пустым или показывает ошибку

Воспроизведение:

1. Войти в приложение
2. Перейти в раздел "Shorts" / "Видео"
3. Кликнуть на любое видео
4. Видео-плеер не появляется

Возможные причины:

- Видео файлы не загружены на сервер
- Неправильный путь к видео
- CORS проблемы с video source
- Видео-плеер (video.js/Plyr) не инициализирован

Решение:

1. Проверить наличие видео в storage:  
ls -la /var/www/gogomarket/uploads/videos/

2. Проверить URL видео в HTML:

```
<video src="/uploads/videos/test.mp4">
```

3. Проверить CORS заголовки:

```
Access-Control-Allow-Origin: *
```

4. Добавить обработку ошибок:

```
<video onError={(e) => console.error('Video error:', e)}>
```

Время на исправление: 2-3 дня

## 3.2 Высокий приоритет (Важные функции)

### FEAT-001: Отсутствует SMS верификация

Приоритет: 🚨 P1

Затронутые роли: Все (при регистрации)

Статус: Открыто

Оценка времени: 1 неделя

Описание:

При регистрации не требуется подтверждение номера телефона через SMS

Это критично для рынка Узбекистана, где телефон - основной ID

Риски:

- Фейковые аккаунты
- Спам от ботов
- Мошенничество

Решение:

Интеграция SMS-провайдера:

- Playmobile (uz) - <https://playmobile.uz/>
- SMSC.ru (международный)
- Eskiz.uz (uz)

Backend:

1. Генерация 6-значного кода
2. Отправка SMS
3. Проверка кода
4. Rate limiting (макс 3 попытки)

Стоимость:

~500-1000 сум за SMS (~\$0.04-0.08)

Время на реализацию: 3-5 дней

### FEAT-002: Нет интеграции Payme/Click



Приоритет: 🎯 P1

Затронутые роли: Покупатель, Продавец

Статус: Открыто

Оценка времени: 2-3 недели

Описание:

Отсутствуют популярные платежные системы Узбекистана

В ТЗ заявлены Рауме и Click как основные методы оплаты

Текущие методы:

- Только "Наличные при получении"

Требуется:

1. Рауме интеграция (документация: [raume.uz/docs](https://raume.uz/docs))
2. Click интеграция (документация: [click.uz/developers](https://click.uz/developers))

Процесс интеграции:

1. Регистрация в Рауме/Click как мерчант
2. Получение Merchant ID и Secret Key
3. Реализация webhook endpoints
4. Обработка callback'ов
5. Тестирование в sandbox
6. Переход на production

Сложность:

- Рауме: требует юридического лица в Узбекистане
- Click: проще для ИП/физ лиц

Время на реализацию: 2-3 недели (включая регистрацию)

Стоимость: комиссия 2-3% с каждой транзакции

### **FEAT-003: Загрузка изображений только через URL**

Приоритет: 🎯 P1

Затронутые роли: Продавец

Статус: Открыто

Оценка времени: 3-5 дней

Описание:

Продавец может добавить товар только введя URL изображения

Нет возможности загрузить фото с компьютера/телефона

Ожидается:

- Drag & drop загрузка изображений
- Выбор файлов через file picker
- Crop/resize изображений
- Предпросмотр перед загрузкой

Решение:

Frontend:

- React Dropzone для drag & drop
- react-image-crop для обрезки
- Компрессия изображений перед загрузкой

Backend:

- Multer middleware для загрузки файлов
- Sharp для обработки изображений
- Загрузка в DigitalOcean Spaces (S3)

Время на реализацию: 3-5 дней

## 📊 СВОДНАЯ ГОТОВНОСТЬ ПО МОДУЛЯМ

### 4.1 Backend API

Общая готовность: ~50%

Модуль	Статус	Готовность	Комментарий
Аутентификация	✔ Работает	80%	JWT auth работает, нет SMS
Управление пользователями	⚠ Частично	40%	API 404, админ не видит users
Управление товарами	✔ Работает	70%	CRUD работает, нет загрузки фото
Управление заказами	⚠ Частично	30%	Базовая структура, нет оформления
Корзина	✘ Не работает	10%	Функционал не реализован
Платежи	✘ Не работает	5%	Только "Наличные"
Видео/Shorts	✘ Не работает	20%	Модели есть, плеер не работает
QR-логистика	✘ Не работает	10%	Базовая генерация QR
Курьеры	✘ Не работает	0%	Модуль отсутствует

Чат	✘ Не работает	0%	Не реализован
Уведомления	✘ Не работает	10%	Базовая структура

## 4.2 Web Frontend

**Общая готовность: ~45%**

Модуль	Статус	Готовность	Комментарий
Регистрация/Вход	✓ Работает	75%	Работает, нет SMS, сессия не сохраняется
Дашборды по ролям	⚠ Частично	50%	Админ 404, остальные базовые
Каталог товаров	✓ Работает	70%	Список товаров работает
Страница товара	⚠ Частично	60%	Показывает, но кнопка багнутая
Корзина	✘ Не работает	10%	Страница есть, функционал нет
Оформление заказа	✘ Не работает	15%	Форма есть, не работает
Профиль пользователя	⚠ Частично	55%	Показывает данные, нет редактирования
Видео-лента (Shorts)	✘ Не работает	20%	Структура есть, плеер не работает
Курьерская панель	✘ Не работает	0%	Полностью отсутствует
Чат	✘ Не работает	0%	Не реализован
Админ-панель	⚠ Частично	30%	Интерфейс есть, API 404

## 4.3 Mobile App (Flutter)

**Общая готовность: ~35% (оценочно)**

Модуль	Статус	Готовность	Комментарий
Компиляция APK	✓ OK	100%	APK собирается
Структура приложения	✓ OK	80%	Flutter проект настроен
	✘		Требуется

Интеграция с API	Неизвестно	20%	тестирование
Аутентификация	✗ Неизвестно	30%	Требуется тестирование
Каталог товаров	✗ Неизвестно	40%	Требуется тестирование
Видео-плеер	✗ Неизвестно	30%	Плагины установлены
QR-сканер	△ Частично	50%	ML Kit интегрирован
Биометрия	△ Частично	60%	Плагины установлены
Push-уведомления	✗ Не работает	10%	Требуе т настройки Firebase

## ПЛАН ДЕЙСТВИЙ И РЕКОМЕНДАЦИИ

### 5.1 Фаза 1: Критические исправления (Неделя 1-2)

**Цель:** Устранить блокеры, сделать приложение работоспособным

#### Неделя 1: Срочные исправления

**День 1-2: Mobile App** - [ ] Исправить target SDK 36 → 34/35 (1ч) - [ ] Переименовать package name (3ч) - [ ] Убрать RECORD\_AUDIO (30м) - [ ] Настроить signing (1ч) - [ ] Тестирование на устройствах (4ч)

**День 3-4: Web Security** - [ ] Установить SSL сертификат (4ч) - [ ] Настроить HTTPS (2ч) - [ ] Настроить HTTP→HTTPS redirect (30м) - [ ] Обновить все ссылки на https:// (2ч) - [ ] Тестирование (2ч)

**День 5: Backend API** - [ ] Диагностика API 404 (2ч) - [ ] Исправление routing (4ч) - [ ] Тестирование endpoints (2ч)

#### Неделя 2: Базовый функционал

**День 1-2: Корзина** - [ ] Исправить routing корзины (2ч) - [ ] Реализовать добавление товаров (4ч) - [ ] Реализовать удаление товаров (2ч) - [ ] Тестирование (2ч)

**День 3-4: Сессии и авторизация** - [ ] Исправить сохранение JWT (2ч) - [ ] Добавить refresh tokens (4ч) - [ ] Тестирование (2ч)

**День 5: Видео-плеер** - [ ] Диагностика проблемы (2ч) - [ ]  
Исправление плеера (4ч) - [ ] Тестирование (2ч)

**Итого Фаза 1: 10 рабочих дней**

## 5.2 Фаза 2: Оптимизация и доработки (Неделя 3-6)

**Цель:** Довести до production-ready состояния

### Неделя 3-4: Производительность

**Задачи:** - [ ] Оптимизация веб-приложения (FCP, LCP) (3д) - [ ]  
Подключение CDN (1д) - [ ] Оптимизация изображений (2д) - [ ]  
Настройка кэширования (1д) - [ ] Оптимизация APK размера (2д) -  
[ ] App Bundle вместо APK (1д)

### Неделя 5-6: Недостающие функции

**Приоритет 1:** - [ ] SMS верификация (5д) - [ ] Загрузка фото  
товаров (3д) - [ ] Оформление заказов (3д) - [ ] Push-уведомления  
(2д)

**Приоритет 2:** - [ ] Интеграция Payme/Click (10д) - параллельно с  
другими задачами - [ ] QR-логистика (5д) - [ ] Базовый чат (5д)

**Итого Фаза 2: 20 рабочих дней**

## 5.3 Фаза 3: Курьеры и полировка (Неделя 7-10)

**Цель:** Реализовать модуль курьеров, полное тестирование

### Неделя 7-8: Курьеры

- ☐ Backend API для курьеров (5д)
- ☐ Web дашборд курьера (3д)
- ☐ Mobile экраны курьера (5д)
- ☐ Интеграция QR-сканера (2д)

### Неделя 9-10: Тестирование и исправления

- ☐ End-to-end тестирование (5д)
- ☐ Исправление найденных багов (5д)
- ☐ Нагрузочное тестирование (2д)
- ☐ Security audit (3д)

**Итого Фаза 3: 20 рабочих дней**

## 5.4 Фаза 4: Beta и Production (Неделя 11-12)

**Цель:** Запуск для ограниченной аудитории

**Неделя 11: Beta-тестирование**

- ☐ Настройка мониторинга (Sentry, Analytics) (2д)
- ☐ Запуск закрытой beta (10-50 пользователей) (1д)
- ☐ Сбор feedback (3д)
- ☐ Срочные исправления (4д)

**Неделя 12: Подготовка к Production**

- ☐ Финальные правки (3д)
- ☐ Подготовка маркетинговых материалов (2д)
- ☐ Настройка production окружения (2д)
- ☐ Soft launch (limited release) (1д)
- ☐ Мониторинг и оптимизация (2д)

**Итого Фаза 4: 10 рабочих дней**

## 6 ОЦЕНКА РЕСУРСОВ И СРОКОВ

### 6.1 Необходимая команда

**Минимальная команда (60 дней)**

Роль	FTE	Длительность	Человеко-дни
Backend Developer	1.0	60 дней	60
Frontend Developer (Web)	1.0	50 дней	50
Mobile Developer (Flutter)	1.0	55 дней	55
QA Engineer	0.5	40 дней	20
DevOps Engineer	0.3	20 дней	6
UI/UX Designer	0.2	15 дней	3
Project Manager	0.3	60 дней	18
ИТОГО			212 ч/д

**Оптимальная команда (40 дней)**

Роль	FTE	Длительность	Человеко-дни
Backend Developer	2.0	40 дней	80

<b>Frontend Developer (Web)</b>	1.5	40 дней	60
<b>Mobile Developer (Flutter)</b>	2.0	40 дней	80
<b>QA Engineer</b>	1.0	40 дней	40
<b>DevOps Engineer</b>	0.5	30 дней	15
<b>UI/UX Designer</b>	0.5	20 дней	10
<b>Project Manager</b>	0.5	40 дней	20
<b>ИТОГО</b>			<b>305 ч/д</b>

### 6.2 Временные оценки

Сценарий	Команда	Срок	Вероятность
<b>Оптимистичный</b>	Оптимальная (7 чел)	4-6 недель	20%
<b>Реалистичный</b>	Минимальная (4 чел)	8-12 недель	60%
<b>Пессимистичный</b>	Текущая (1-2 чел)	16-24 недели	20%

### 6.3 Бюджетная оценка

#### Затраты на персонал (Минимальная команда, 12 недель)

Примерные ставки для Узбекистана / удаленной работы:

Роль	Ставка/месяц	Месяцев	Стоимость
Backend Developer	\$1500-2000	3	\$4500-6000
Frontend Developer	\$1500-2000	2.5	\$3750-5000
Mobile Developer	\$1500-2000	2.8	\$4200-5600
QA Engineer (part-time)	\$800-1000	2	\$1600-2000
DevOps (part-time)	\$1000-1500	1	\$1000-1500
Designer (part-time)	\$800-1000	0.5	\$400-500
PM (part-time)	\$1000-1500	3	\$3000-4500
<b>ИТОГО персонал</b>			<b>\$18,450-25,100</b>

#### Инфраструктура (месяц)

Сервис	Стоимость/мес	Месяцев	Итого
Cloud Hosting (AWS/Azure)	\$100	3	\$300

DigitalOcean Droplet (4GB)	\$24	3	\$72
DigitalOcean Spaces (250GB)	\$5	3	\$15
PostgreSQL (Managed)	\$15	3	\$45
SSL сертификат	\$0 (Let's Encrypt)	-	\$0
CDN (Cloudflare)	\$0 (free plan)	-	\$0
Monitoring (Sentry)	\$0 (developer plan)	-	\$0
SMS-шлюз (тестирование)	~\$50	-	\$50
<b>ИТОГО инфраструктура</b>			<b>\$182</b>

#### Дополнительные расходы

Статья	Стоимость
Регистрация в Payme/Click	\$100-300
Google Play Developer (\$25 одноразово)	\$25
Apple Developer (если iOS)	\$99/год
Доменное имя gogomarket.uz	\$10-20/год
Резерв на непредвиденное (10%)	\$2000
<b>ИТОГО дополнительно</b>	<b>~\$2200-2500</b>

#### ОБЩИЙ БЮДЖЕТ

Персонал:	\$18,450-25,100
Инфраструктура:	\$182
Дополнительно:	\$2,200-2,500
<b>ИТОГО:</b>	<b>\$20,832-27,782</b>

**Диапазон:** \$21,000 - \$28,000 на 3 месяца разработки

## 📌 РИСКИ И МИТИГАЦИЯ

### 7.1 Технические риски

#### РИСК-1: Накопление технического долга

Вероятность: 🟡 ВЫСОКАЯ (70%)

Влияние: 🟡 КРИТИЧЕСКОЕ

Описание:



При спешке и фокусе на "быстрый результат" накапливаются:

- Неисправленные баги
- Костыльные решения
- Отсутствие тестов
- Плохая документация

Последствия:

- Замедление разработки в будущем
- Сложность добавления новых функций
- Высокая стоимость поддержки
- Риск критических багов в production

Митигация:

- ✓ Code review обязателен для всех PR
- ✓ Минимум 60% покрытие unit тестами
- ✓ Еженедельные рефакторинг-сессии
- ✓ Документировать архитектурные решения
- ✓ Использовать linter и code formatter

## **РИСК-2: Проблемы с интеграциями**

Вероятность: ● СРЕДНЯЯ (50%)

Влияние: ● ВЫСОКОЕ

Описание:

Интеграции с внешними сервисами могут затянуться:

- PayPal/Click (требует регистрацию компании)
- SMS-шлюз (может не работать в тестовом режиме)
- Push-уведомления (Firebase настройка)

Последствия:

- Задержка запуска на 2-4 недели
- Невозможность протестировать критические функции

Митигация:

- ✓ Начать регистрацию в PayPal/Click немедленно
- ✓ Использовать mock-сервисы для разработки
- ✓ Параллельная разработка (не блокировать другие задачи)
- ✓ Иметь plan B (альтернативные сервисы)

## **РИСК-3: Производительность при масштабировании**

Вероятность: ● НИЗКАЯ (30%)

Влияние: ● ВЫСОКОЕ

Описание:

При росте пользователей возможны проблемы:

- Медленные SQL запросы

- Нехватка памяти/CPU
- Проблемы с загрузкой видео

Последствия:

- Плохой UX для пользователей
- Негативные отзывы
- Отток пользователей

Митигация:

- ✓ Нагрузочное тестирование (JMeter/k6)
- ✓ Индексы в базе данных
- ✓ Кэширование (Redis)
- ✓ CDN для статики и видео
- ✓ Мониторинг производительности (New Relic)

## 7.2 Организационные риски

### РИСК-4: Нереалистичные сроки

Вероятность: ● ВЫСОКАЯ (60%)

Влияние: ● ВЫСОКОЕ

Описание:

Давление на команду "запустить быстрее" может привести к:

- Пропуску тестирования
- Низкому качеству кода
- Выгоранию разработчиков

Последствия:

- Критические баги в production
- Увольнение разработчиков
- Репутационные потери

Митигация:

- ✓ Реалистичное планирование (buffer 20%)
- ✓ Четкие приоритеты (что можно отложить)
- ✓ Регулярные ретроспективы
- ✓ Защита команды от overpromising

### РИСК-5: Недостаток ресурсов

Вероятность: ● СРЕДНЯЯ (40%)

Влияние: ● КРИТИЧЕСКОЕ

Описание:

Текущая команда может быть недостаточной:

- 1-2 разработчика на весь стек
- Отсутствие QA engineer

- Нет выделенного DevOps

Последствия:

- Затягивание сроков
- Низкое качество
- Burnout разработчиков

Митигация:

- ✓ Нанять дополнительных разработчиков
- ✓ Аутсорс для некритичных задач
- ✓ Использовать готовые решения где возможно
- ✓ Приоритизация функций (MVP vs nice-to-have)

## 7.3 Бизнес-риски

### РИСК-6: Конкуренция

Вероятность: ● СРЕДНЯЯ (40%)

Влияние: ● СРЕДНЕЕ

Описание:

На рынке Узбекистана есть конкуренты:

- Uzum Market (крупнейший)
- Olcha, Mediapark (электроника)
- Instagram Shopping (неформальный)

Последствия:

- Сложность привлечения пользователей
- Необходимость больших маркетинговых затрат
- Демпинг цен конкурентами

Митигация:

- ✓ Фокус на уникальность (видео-коммерция)
- ✓ Привлечение micro-influencers
- ✓ Программа лояльности для ранних пользователей
- ✓ Качественная поддержка клиентов

### РИСК-7: Юридические проблемы

Вероятность: ● НИЗКАЯ (20%)

Влияние: ● КРИТИЧЕСКОЕ

Описание:

Возможные юридические проблемы:

- Лицензия на электронную коммерцию
- Налоговая отчетность
- Защита персональных данных (GDPR-like)
- Договоры с Payme/Click

Последствия:

- Блокировка сервиса
- Штрафы
- Судебные иски

Митигация:

- ✓ Консультация с юристом до запуска
  - ✓ Регистрация ООО/ИП
  - ✓ Privacy Policy и Terms of Service
  - ✓ Compliance с местным законодательством
- 

## 📌 ФИНАЛЬНАЯ ОЦЕНКА И ВЕРДИКТ

✓ Можно ли запускать в production сейчас?

## ✗ НЕТ, КАТЕГОРИЧЕСКИ НЕ РЕКОМЕНДУЕТСЯ

Текущее состояние: 45% готовности

Причины НЕ запускать:

### 🔴 Критические блокиеры безопасности

1. **Отсутствие HTTPS** - все данные передаются открытым текстом
  - Пароли пользователей видны в сети
  - Данные банковских карт (в будущем) будут уязвимы
  - Нарушение PCI DSS, если будут платежи
2. **APK не готов к публикации**
  - Target SDK 36 (несуществующая версия Android)
  - Package name "com.example.\*" (тестовый)
  - Google Play Store отклонит приложение

### 🔴 Критические функциональные проблемы

3. **Корзина не работает** - невозможно совершить покупку
4. **Курьеры отсутствуют** - невозможна доставка
5. **Видео не открываются** - основная фича не работает
6. **Платежи** - только наличные (нет Payme/Click)

### 🔴 Серьезные UX проблемы

7. **Медленная загрузка** - 53% пользователей уйдут до загрузки (>3s)
8. **Сессия не сохраняется** - пользователи logout после F5
9. **API недоступен** - админ не видит пользователей

## Что будет, если запустить сейчас?

### Сценарий “запуск без исправлений”:

День 1:

- ✓ 100 пользователей регистрируются

День 2:

- ✗ 80 пользователей пытаются купить товары
- ✗ 70 не могут добавить в корзину
- ✗ 10 добавили, но не могут оформить заказ
- ✗ 0 успешных заказов

День 3:

- 📉 50 негативных отзывов в соцсетях
- 📉 Репутационные потери
- 📉 Сложность вернуть пользователей в будущем

День 7:

- ⚠ Возможные проблемы с безопасностью
- ⚠ Утечка данных пользователей
- ⚠ Юридические последствия

**Вывод:** Один шанс произвести первое впечатление - не стоит его тратить на сырой продукт.

---

## 🕒 Реалистичные сроки до Production Ready

### Сценарий 1: “Экспресс” (4-6 недель)

**Команда:** 6-7 человек full-time

**Что будет реализовано:** - ✓ Все критические блокеры исправлены - ✓ Базовый функционал работает (регистрация, каталог, заказы) - ✓ Безопасность (HTTPS, валидация, санитизация) - ⚠ Курьеры - базовая версия (без всех фич) - ⚠ Платежи - только наличные (Payme/Click - фаза 2) - ⚠ Видео - базовый плеер (без рекомендаций)

**Риски:** - Высокий технический долг - Возможны баги в production - Необходим активный мониторинг

**Рекомендуется:** Только если есть срочная бизнес-необходимость (инвесторы, конкуренты)

**Сценарий 2: “Стандартный” (8-12 недель) ✓**  
**РЕКОМЕНДУЕТСЯ**

**Команда:** 4-5 человек

**Что будет реализовано:** - ✓ Все функции MVP - ✓ Качественное тестирование - ✓ Интеграция Payme/Click - ✓ Полный функционал курьеров - ✓ SMS верификация - ✓ Оптимизация производительности - ✓ Beta-тестирование - ✓ Документация

**Риски:** - Минимальные, управляемые - Достаточно времени на исправления

**Рекомендуется:** Оптимальный баланс скорость/качество

**Сценарий 3: “Тщательный” (16-20 недель)**

**Команда:** 3-4 человека

**Что будет реализовано:** - ✓ Все функции из ТЗ (139 пунктов) - ✓ Полное покрытие тестами - ✓ Продвинутое фичи (AI рекомендации, аналитика) - ✓ Масштабируемая архитектура - ✓ Подробная документация

**Риски:** - Упущенное время на рынке - Возможная избыточность функций

**Рекомендуется:** Если нет спешки и хочется “сделать правильно”

---

**💰 Финальная стоимость проекта**

**До Production (Сценарий 2 - 12 недель)**

Разработка:	\$20,000 - \$28,000
Инфраструктура:	\$500 (3 месяца)
Маркетинг:	\$3,000 - \$5,000 (опционально)

---

**ИТОГО:** \$23,500 - \$33,500

**Первый год работы (после запуска)**

Поддержка:	\$3,000 - \$5,000/месяц
Инфраструктура:	\$200 - \$500/месяц (растет с трафиком)
Маркетинг:	\$5,000 - \$15,000/месяц
SMS:	\$500 - \$2,000/месяц (зависит от объема)
Комиссии Payme/Click:	2-3% от оборота

---

ИТОГО/месяц:        \$8,700 - \$22,500+

---

## 🎯 ФИНАЛЬНЫЕ РЕКОМЕНДАЦИИ

### ШАГ 1: Немедленные действия (1-2 дня)

#### Приоритет P0 - сделать СЕЙЧАС:

1. ⚠️ **Закрыть сайт от публичного доступа** до исправления HTTPS  
  
# Временно - требовать basic auth  
auth\_basic "Development";  
auth\_basic\_user\_file /etc/nginx/.htpasswd;
2. ⚠️ **НЕ публиковать APK** до исправления target SDK
3. ✔️ **Создать roadmap** с реалистичными сроками
4. ✔️ **Собрать/расширить команду** до 4-5 человек

### ШАГ 2: Спринт исправлений (2 недели)

**Исправить критические блокеры:** - HTTPS + SSL - Target SDK в APK - API 404 проблемы - Корзина - Сессии

**После этого можно:** - Открыть closed beta для тестировщиков - Собрать первый feedback

### ШАГ 3: Разработка MVP (6-8 недель)

**Фокус на core функциональности:** - Полный цикл заказа (каталог → корзина → оплата → доставка) - Курьеры (базовая версия) - Интеграция платежей (хотя бы 1 из Payme/Click) - Оптимизация производительности

### ШАГ 4: Beta-тестирование (2-3 недели)

**Ограниченный запуск:** - 50-200 тестовых пользователей - Активный сбор feedback - Оперативное исправление критичных багов

### ШАГ 5: Production Launch

**Soft launch:** - Таргетированная реклама (узкая аудитория) - Мониторинг метрик - A/B тестирование

**Full launch:** - Масштабирование рекламы - Расширение функционала - Постоянная оптимизация

---

**📊 МЕТРИКИ УСПЕХА**

**После запуска отслеживать:**

Метрика	Target	Критично
Регистраций/день	50-100	-
Успешных заказов/день	20-50	✓
Conversion rate	>2%	✓
Crash rate	<1%	✓
Retention (D7)	>20%	-
Retention (D30)	>10%	-
Avg. Order Value	>50,000 сум	-
Время загрузки (FCP)	<1.8s	✓
API uptime	>99.5%	✓

---

**📌 ЗАКЛЮЧЕНИЕ**

**GoGoMarket - амбициозный проект с потенциалом, но:**

✓ **Сильные стороны:** - Актуальная идея (видео-коммерция растет) - Современный tech stack - Подробное ТЗ

✗ **Слабые стороны:** - Недостаточная готовность (45%) - Критические проблемы безопасности - Недостаток ресурсов

**ВЕРДИКТ:**

НЕ ГОТОВ к production
Требуется 8-12 недель доработки
Бюджет: \$23,500-33,500
Команда: минимум 4-5 человек
⚠ РЕКОМЕНДАЦИЯ: Не спешить с запуском
✓ Лучше сделать качественно, чем быстро и плохо

**“Move fast and break things” НЕ работает, когда речь о реальных деньгах пользователей.**



Лучше потратить 3 месяца и запустить качественный продукт, чем 1 месяц и получить негативные отзывы, которые убьют проект.

**Отчет подготовлен:** 4 января 2026 в 16:50  
**Версия:** 1.0 (Final)  
**Следующий review:** Через 2 недели после начала Sprint 1

## ПРИЛОЖЕНИЯ

### A. Ссылки на детальные отчеты

- 1. GitHub Repository Analysis**  
/home/ubuntu/gogomarket\_analysis.md (52 KB)  
*Анализ кодовой базы, технологий, документации*
- 2. Functional Testing Report**  
/home/ubuntu/gogomarket\_testing\_report.md (32 KB)  
*Результаты тестирования по ролям, сценариям*
- 3. API Analysis**  
/home/ubuntu/gogomarket\_api\_analysis.md (12 KB)  
*Детальный анализ backend endpoints*
- 4. Issues Tracker**  
/home/ubuntu/gogomarket\_issues\_tracker.md (14 KB)  
*Полный список задач и багов с приоритетами*
- 5. APK Analysis (JSON)**  
/home/ubuntu/Uploads/apk\_analysis/apk\_analysis.json  
*Структурированные данные о мобильном приложении*
- 6. Lighthouse Reports (JSON)**  
/home/ubuntu/Uploads/html\_analysis/lighthouse\_data.json  
*Метрики производительности веб-приложения*

### B. Декомпилированный APK

**Location:** /home/ubuntu/Uploads/apk\_analysis/decompiled/

**Структура:**

```
decompiled/  
├─ AndroidManifest.xml      # Манифест приложения  
├─ assets/                  # Flutter assets  
│   └─ flutter_assets/  
├─ lib/                     # Нативные библиотеки
```

```
|   |─ arm64-v8a/
|   |─ armeabi-v7a/
|   └─ x86_64/
└─ res/                # Ресурсы (иконки, макеты)
   └─ smali/           # Декомпилированный код
```

## C. Lighthouse HTML отчеты

**Location:** /home/ubuntu/Uploads/\*.html

- 64.226.94.133-20260104T211038.html (Run #1)
- 64.226.94.133-20260104T211145.html (Run #2)
- 64.226.94.133-20260104T211257.html (Run #3)
- 64.226.94.133-20260104T211408.html (Run #4)

## D. Контакты и ресурсы

**GitHub:**

<https://github.com/buranovt2025-jpg/gogomarke.uz>

**Live Demo:**

<http://64.226.94.133>

**Branch:**

devin/1767373941-gogomarket-mvp

---

### Конец отчета

---

*Этот отчет является конфиденциальным и предназначен только для внутреннего использования командой GoGoMarket.*