

GoGoMarket.uz - Полный анализ проекта

Дата анализа: 4 января 2026

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

Ветка: devin/1767373941-gogomarket-mvp

Работающий сайт: <http://64.226.94.133> и <https://64-226-94-133.sslip.io>

1. ОБЗОР ПРОЕКТА

1.1 Описание

GoGoMarket - это социальный видео-маркетплейс (аналог TikTok Shop) для рынка Узбекистана, где:

- **Продавцы** загружают видео/роллы с товарами
- **Покупатели** покупают товары прямо из видео
- **Курьеры** доставляют заказы с QR-подтверждением
- **Админы** управляют платформой

1.2 Ключевые особенности

- **ВидеоКоммерция:** Вертикальная лента видео как в TikTok
- **4 роли:** Админ, Продавец, Покупатель, Курьер
- **Безопасные платежи:** Escrow система, Payme/Click, наличные
- **QR-логистика:** Цепочка подтверждений Продавец → Курьер → Покупатель
- **Биометрия:** Face ID / Touch ID для оплаты
- **Мультиязычность:** RU, UZ, EN
- **Оффлайн режим:** Кэширование и синхронизация

2. ТЕХНОЛОГИЧЕСКИЙ СТЕК

2.1 Backend (Node.js/TypeScript)

```
{
  "runtime": "Node.js",
  "framework": "Express 5.2.1",
  "language": "TypeScript 5.9.3",
  "database": "PostgreSQL",
  "orm": "Sequelize 6.37.7",
  "auth": "JWT (jsonwebtoken 9.0.3)",
  "security": "bcryptjs 3.0.3, helmet 8.1.0",
  "qr": "qrcode 1.5.4",
  "storage": "AWS SDK (S3-compatible)",
  "upload": "multer 2.0.2"
}
```

Основные зависимости:

- express - веб-фреймворк

- `sequelize + pg` - ORM для PostgreSQL
- `jsonwebtoken` - JWT аутентификация
- `bryptjs` - хеширование паролей
- `qrcode` - генерация QR-кодов
- `@aws-sdk/client-s3` - загрузка медиа (DigitalOcean Spaces)
- `helmet` - безопасность HTTP заголовков
- `cors` - CORS middleware
- `morgan` - логирование запросов

2.2 Frontend Mobile (Flutter)

```

sdk: '>=3.0.0 <4.0.0'
dependencies:
  # State Management
  provider: ^6.1.1

  # HTTP & API
  http: ^1.1.0
  dio: ^5.4.0

  # Local Storage
  shared_preferences: ^2.2.2
  flutter_secure_storage: ^9.0.0

  # UI Components
  google_fonts: ^6.1.0
  flutter_svg: ^2.0.9
  cached_network_image: ^3.3.1
  shimmer: ^3.0.0

  # Video Player
  video_player: ^2.8.2
  chewie: ^1.7.4

  # Camera & Media
  camera: ^0.10.5+9
  image_picker: ^1.0.7

  # QR Code
  qr_flutter: ^4.1.0
  mobile_scanner: ^5.1.1

  # Biometrics
  local_auth: ^2.1.8

  # Navigation
  url_launcher: ^6.2.4

```

2.3 Web Frontend (React/TypeScript)

```
{  
  "framework": "React 18.3.1",  
  "language": "TypeScript 5.6.2",  
  "build": "Vite 6.0.1",  
  "styling": "Tailwind CSS 3.4.16",  
  "ui": "Radix UI + shadcn/ui",  
  "routing": "React Router 7.11.0",  
  "forms": "React Hook Form 7.69.0 + Zod 4.3.4",  
  "charts": "Recharts 2.12.4",  
  "notifications": "Sonner 2.0.7"  
}
```

UI компоненты (Radix UI):

- Accordion, Alert Dialog, Avatar, Badge, Button, Card, Carousel
 - Checkbox, Dialog, Dropdown Menu, Form, Input, Label
 - Navigation Menu, Popover, Progress, Radio Group, Select
 - Separator, Sheet, Sidebar, Slider, Switch, Tabs, Toast, Tooltip
-

3. СТРУКТУРА ПРОЕКТА

3.1 Общая структура (218 файлов)

```

gogomarke.uz/
├── README.md
├── REQUIREMENTS.md
├── FUNCTIONAL_LOGIC.md
├── GAP_ANALYSIS.md
└── WORK_LOG.md
# Общее описание
# Полное ТЗ (139 функций)
# Логика работы, API контракты
# Анализ готовности (27% DONE, 23% PARTIAL, 50% MISSING)

backend/
├── src/
│   ├── config/
│   └── controllers/
# Конфигурация (database, index)
# Контроллеры (auth, product, order, video, payment, chat, upload)
├── middleware/ # Middleware (auth, errorHandler, validation)
├── models/ # Sequelize модели (User, Product, Order, Video, Transaction, Review, Dispute, Address, Message)
├── routes/ # Маршруты API
├── services/ # Сервисы (notification, qr, sms, upload)
├── types/ # TypeScript типы и enums
└── app.ts # Точка входа
package.json
.env.example

frontend/ # Flutter мобильное приложение
├── lib/
│   ├── config/
│   ├── l10n/
│   ├── models/
│   ├── providers/
│   └── locale/
│       ├── screens/
│       │   ├── auth/
│       │   ├── home/
│       │   ├── video/
│       │   ├── product/
│       │   ├── cart/
│       │   ├── checkout/
│       │   ├── order/
│       │   ├── profile/
│       │   ├── seller/
│       │   ├── courier/
│       │   └── qr/
│       ├── services/
│       ├── utils/
│       ├── widgets/
│       └── main.dart
pubspec.yaml
assets/

web/gogomarket-web/ # React веб-приложение
├── src/
│   ├── components/
│   ├── contexts/
│   ├── hooks/
│   ├── pages/
│   │   ├── auth/
│   │   ├── buyer/
│   │   ├── seller/
│   │   └── admin/
│   └── transactions/
│       ├── services/
│       └── types/
# UI компоненты (Layout, BottomNav, ui/*)
# React контексты (Auth, Cart)
# Custom hooks (use-mobile, use-toast)
# Страницы
# Логин, регистрация
# Главная, товары, корзина, заказы
# Панель продавца, товары
# Админ панель (dashboard, users, orders, transactions)
# API сервис
# TypeScript типы

```

```

    lib/          # Утилиты
    App.tsx      # Главный компонент
    main.tsx     # Точка входа
    package.json
    public/       # Статические файлы

```

4. БАЗА ДАННЫХ (PostgreSQL + Sequelize)

4.1 Модели данных

User (Пользователь)

```

{
  id: UUID (PK)
  phone: string (unique)           // +998XXXXXXXXX
  email?: string (unique)
  passwordHash: string            // bcrypt
  firstName: string
  lastName: string
  role: enum('admin', 'seller', 'buyer', 'courier')
  avatar?: string
  isVerified: boolean             // OTP подтвержден
  isActive: boolean               // Не заблокирован
  language: enum('en', 'ru', 'uz')
  oneIdVerified: boolean          // Верификация через OneID
  acceptedTerms: boolean
  acceptedPrivacy: boolean
  lastLoginAt?: Date
  // Wallet для продавцов и курьеров
  availableBalance: decimal(12,2)
  pendingBalance: decimal(12,2)
  totalEarnings: decimal(12,2)
  fcmToken?: string                // Push уведомления
  createdAt: Date
  updatedAt: Date
}

```

Product (Товар)

```
{
  id: UUID (PK)
  sellerId: UUID (FK ↞ users)
  title: string
  titleRu?: string // Локализация
  titleUz?: string
  description: string
  descriptionRu?: string
  descriptionUz?: string
  price: decimal(12,2)
  originalPrice?: decimal(12,2) // Для скидок
  currency: string // 'UZS'
  category: string
  sizes?: string[] // Варианты размеров
  colors?: string[] // Варианты цветов
  stock: integer
  images: string[] // URLs
  isActive: boolean
  rating: decimal(2,1) // 0-5
  reviewCount: integer
  createdAt: Date
  updatedAt: Date
}
```

Video (Видео/Рилс)

```
{
  id: UUID (PK)
  sellerId: UUID (FK ↞ users)
  productId?: UUID (FK ↞ products)
  videoUrl: string
  thumbnailUrl?: string
  title: string
  titleRu?: string
  titleUz?: string
  description?: string
  descriptionRu?: string
  descriptionUz?: string
  duration: integer // Секунды
  viewCount: integer
  likeCount: integer
  isLive: boolean // Live-трансляция
  isActive: boolean
  // AI-извлечение данных из речи
  aiExtractedTitle?: string
  aiExtractedPrice?: decimal(12,2)
  aiExtractedDescription?: string
  createdAt: Date
  updatedAt: Date
}
```

Order (Заказ)

```
{
  id: UUID (PK)
  orderNumber: string (unique)      // GGM-XXXXXX (auto-generated)
  buyerId: UUID (FK ↲ users)
  sellerId: UUID (FK ↲ users)
  courierId?: UUID (FK ↲ users)
  productId: UUID (FK ↲ products)
  videoId?: UUID (FK ↲ videos)    // Откуда купили
  quantity: integer
  unitPrice: decimal(12,2)
  totalAmount: decimal(12,2)
  courierFee: decimal(12,2)
  platformCommission: decimal(12,2) // 5%
  sellerAmount: decimal(12,2)
  currency: string                // 'UZS'
  status: enum(OrderStatus)       // pending, confirmed, picked_up, in_transit,
  delivered, cancelled, disputed
  paymentMethod: enum(PaymentMethod) // card, cash, payme, click
  paymentStatus: enum(PaymentStatus) // pending, held, completed, refunded, failed
  shippingAddress: string
  shippingCity: string
  shippingPhone: string
  buyerNote?: string
  sellerQrCode?: string           // QR для передачи курьеру
  courierQrCode?: string           // QR для доставки
  deliveryCode?: string           // 4-6 значный код
  pickedUpAt?: Date
  deliveredAt?: Date
  cancelledAt?: Date
  cancelReason?: string
  createdAt: Date
  updatedAt: Date
}
```

Transaction (Транзакция)

```
{
  id: UUID (PK)
  orderId: UUID (FK ↲ orders)
  userId?: UUID (FK ↲ users)
  type: enum(TransactionType)    // payment, escrow_hold, escrow_release,
  seller_payout, courier_payout, platform_commission, commission_reversal, refund
  amount: decimal(12,2)
  currency: string                // 'UZS'
  status: enum(PaymentStatus)     // pending, held, completed, refunded, failed
  description: string
  referenceId?: string            // ID в Payme/Click
  metadata?: jsonb
  createdAt: Date
  updatedAt: Date
}
```

Review (Отзыв)

```
{
  id: UUID (PK)
  orderId: UUID (FK → orders)
  productId: UUID (FK → products)
  buyerId: UUID (FK → users)
  sellerId: UUID (FK → users)
  rating: integer // 1-5
  comment?: string
  images?: string[]
  isVerifiedPurchase: boolean
  createdAt: Date
  updatedAt: Date
}
```

Dispute (Спор)

```
{
  id: UUID (PK)
  orderId: UUID (FK → orders)
  buyerId: UUID (FK → users)
  sellerId: UUID (FK → users)
  reason: string
  description: string
  status: enum(DisputeStatus) // open, in_review, resolved, closed
  resolution?: string
  resolvedBy?: UUID (FK → users)
  resolvedAt?: Date
  createdAt: Date
  updatedAt: Date
}
```

Address (Адрес)

```
{
  id: UUID (PK)
  userId: UUID (FK → users)
  label: string // "Дом", "Работа"
  fullAddress: string
  city: string
  region?: string
  postalCode?: string
  latitude?: decimal(10,8)
  longitude?: decimal(11,8)
  isDefault: boolean
  createdAt: Date
  updatedAt: Date
}
```

Message (Сообщение в чате)

```
{
  id: UUID (PK)
  senderId: UUID (FK → users)
  receiverId: UUID (FK → users)
  orderId?: UUID (FK → orders)
  content: string
  attachments?: string[]
  isRead: boolean
  createdAt: Date
  updatedAt: Date
}
```

4.2 Связи между таблицами

```

users (1) └─< (N) products (sellerId)
users (1) └─< (N) videos (sellerId)
users (1) └─< (N) orders (buyerId, sellerId, courierId)
users (1) └─< (N) reviews (buyerId, sellerId)
users (1) └─< (N) addresses (userId)
users (1) └─< (N) messages (senderId, receiverId)
users (1) └─< (N) transactions (userId)

products (1) └─< (N) orders (productId)
products (1) └─< (N) reviews (productId)
products (1) └─< (N) videos (productId)

orders (1) └─< (N) transactions (orderId)
orders (1) └─< (N) reviews (orderId)
orders (1) └─< (N) disputes (orderId)
orders (1) └─< (N) messages (orderId)

videos (1) └─< (N) orders (videoId)

```

5. API ENDPOINTS

5.1 Аутентификация (/api/v1/auth)

POST /register	# Регистрация (phone, password, role)
POST /verify-otp	# Подтверждение OTP
POST /login	# Логин (phone, password)
POST /refresh	# Обновление токена
GET /me	# Текущий пользователь
PUT /me	# Обновление профиля

5.2 Товары (/api/v1/products)

GET /	# Список товаров (фильтры, пагинация)
GET /:id	# Детали товара
POST /	# Создать товар (seller only)
PUT /:id	# Обновить товар (seller only)
DELETE /:id	# Удалить товар (seller only)
GET /seller/:sellerId	# Товары продавца

5.3 Видео (/api/v1/videos)

```

GET   /feed           # Лента видео (пагинация)
GET   /:id            # Детали видео
POST  /              # Загрузить видео (seller only)
PUT   /:id            # Обновить видео (seller only)
DELETE/:id           # Удалить видео (seller only)
POST  /:id/view       # Увеличить счетчик просмотров
POST  /:id/like       # Лайк видео

```

5.4 Заказы (/api/v1/orders)

```

GET   /               # Список заказов (роль-зависимый)
GET   /:id            # Детали заказа
POST  /              # Создать заказ (buyer only)
PUT   /:id/confirm    # Подтвердить заказ (seller only)
PUT   /:id/assign-courier # Назначить курьера (seller/admin)
POST  /:id/scan-pickup # Сканировать QR забора (courier only)
POST  /:id/confirm-delivery # Подтвердить доставку (courier only)
PUT   /:id/cancel     # Отменить заказ
GET   /available-for-courier # Доступные заказы для курьера
POST  /:id/accept      # Принять заказ (courier only)

```

5.5 Платежи (/api/v1/payments)

```

POST  /payme/webhook   # Webhook от Payme
POST  /click/webhook   # Webhook от Click
GET   /transactions    # История транзакций
POST  /payout-request  # Запрос на вывод средств (seller/courier)

```

5.6 Чат (/api/v1/chat)

```

GET   /conversations    # Список чатов
GET   /conversations/:id/messages # Сообщения в чате
POST  /conversations/:id/messages # Отправить сообщение

```

5.7 Загрузка файлов (/api/v1/upload)

```

POST  /image           # Загрузить изображение
POST  /video            # Загрузить видео

```

5.8 Здоровье (/api/v1/health)

```

GET   /               # Проверка работоспособности API

```

6. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ (из REQUIREMENTS.md)

6.1 Роли и их функции

ГОСТЬ (без регистрации) - 5 функций

1. Смотреть ленту видео/рилсов
2. Искать и фильтровать товары
3. Открывать карточку товара
4. Читать отзывы
5. Шарить ссылки

ПОКУПАТЕЛЬ - 35 функций

Поиск и навигация:

1. Умный поиск с фильтрами (PARTIAL)
2. Лента рилсов (PARTIAL)
3. Страница продавца (MISSING)
4. Карточка товара (DONE)

Избранное и подписки:

5. Избранное (MISSING)
6. Подписка на продавцов (MISSING)

Уведомления и коммуникация:

7. Push-уведомления (MISSING)
8. Комментарии под видео (MISSING)
9. Чат с продавцом (MISSING)
10. Чат по заказу (MISSING)

Покупка:

11. Купить из видео (DONE)
12. Варианты товара (MISSING)
13. Корзина (PARTIAL)
14. Оформление заказа (DONE)

Скидки и промо:

15. Купоны и промокоды (MISSING)

Адреса и доставка:

16. Адресная книга (PARTIAL)
17. Геолокация (MISSING)
18. Способы доставки (PARTIAL)

Оплата:

19. Наличные курьеру (DONE)
20. Escrow (PARTIAL)
21. Биометрическая оплата (PARTIAL)
22. Трекинг заказа (MISSING)

Заказы:

23. История заказов (DONE)

- 24. Отмена заказа (DONE)
- 25. Повторить заказ (MISSING)

Отзывы и жалобы:

- 26. Отзывы и рейтинг (PARTIAL)
- 27. Возвраты и споры (PARTIAL)
- 28. Жалобы на контент (MISSING)

Персонализация:

- 29. История просмотров (MISSING)
- 30. Рекомендации (MISSING)

Шеринг:

- 31. Шеринг через соцсети (MISSING)

Профиль:

- 32. Профиль покупателя (DONE)
- 33. Центр уведомлений (MISSING)

Поддержка:

- 34. Обращение в поддержку (MISSING)

Технические:

- 35. Оффлайн режим (PARTIAL)

ПРОДАВЕЦ - 37 функций

Базовые:

- 1. Регистрация продавца (DONE)
- 2. Верификация продавца (MISSING)
- 3. Страница магазина (MISSING)

Управление товарами:

- 4. Добавление товара (DONE)
- 5. AI-заполнение из речи (MISSING)
- 6. Редактирование товара (DONE)
- 7. Удаление/скрытие товара (DONE)
- 8. Категории товаров (DONE)
- 9. Варианты товара (MISSING)
- 10. Несколько фото/видео (DONE)
- 11. Управление остатками (DONE)

Контент:

- 12. Создание видео/роликов (PARTIAL)
- 13. Создание сторис (MISSING)
- 14. Несколько товаров к видео (DONE)
- 15. Закрепленные товары (MISSING)
- 16. Модерация контента (MISSING)

Заказы:

- 17. Управление заказами (PARTIAL)
- 18. Чат с покупателем (MISSING)

Скидки и промо:

- 19. ✗ Создание купонов (MISSING)
- 20. ⚠ Скидки на товары (PARTIAL - есть originalPrice)
- 21. ✗ Платное продвижение (MISSING)

Аналитика:

- 22. ✗ Аналитика продаж (MISSING)
- 23. ✗ Live-трансляции (MISSING)

Уведомления:

- 24. ✗ Push-уведомления (MISSING)
- 25. ✗ Центр уведомлений (MISSING)

Финансы:

- 26. ✗ Вывод средств (MISSING)
- 27. ✓ Профиль продавца (DONE)

Отзывы и споры:

- 28. ✗ Ответ на отзывы (MISSING)
- 29. ⚠ Работа со спорами (PARTIAL)

Подписчики:

- 30. ✗ Видеть подписчиков (MISSING)

Поддержка:

- 31. ✗ Обращение в поддержку (MISSING)

Доставка:

- 32. ⚠ Адрес для курьера (PARTIAL)
- 33. ✗ Политика возврата (MISSING)
- 34. ✗ Самостоятельная доставка (MISSING)

Ограничения:

- 35-36. ✓ Изоляция данных (DONE)

КУРЬЕР - 32 функции**Базовые:**

1. ✓ Регистрация курьера (DONE)
2. ✗ Верификация курьера (MISSING)
3. ✗ Зона работы (MISSING)

Заказы:

4. ✓ Список назначенных заказов (DONE)
5. ✓ Принять/отклонить заказ (DONE)
6. ✓ Детали заказа (DONE)
7. ✗ Несколько заказов одновременно (MISSING)

Навигация:

8. ✗ Переход в навигатор (MISSING)
9. ✗ Геолокация realtime (MISSING)

Статусы:

10. ✓ Смена статуса заказа (DONE)

11. Расширенные статусы (MISSING)
12. Перенос доставки (MISSING)

Подтверждение:

13. QR-код подтверждения (DONE)
14. Скан QR товара (DONE)
15. Фото при доставке (MISSING)

Оплата:

16. Прием наличных (DONE)
17. Сумма к сдаче (PARTIAL)

Коммуникация:

18. Чат с покупателем (MISSING)
19. Чат с продавцом (MISSING)
20. Push-уведомления (MISSING)
21. Звонок покупателю (MISSING)
22. Звонок продавцу (MISSING)

Проблемы:

23. Проблема с заказом (MISSING)

Возвраты:

24. Возврат товара (MISSING)

Профиль:

25. Профиль курьера (DONE)
26. Онлайн/оффлайн статус (MISSING)

Финансы:

27. Баланс и выплаты (MISSING)

Поддержка:

28. Обращение в поддержку (MISSING)
29. Центр уведомлений (MISSING)

АДМИН - 30 функций

Структура:

- 1-4. RBAC система (MISSING)

Модерация:

5. Очереди модерации (MISSING)
6. Управление категориями (MISSING)
7. Модерация комментариев (MISSING)

Пользователи:

8. Управление пользователями (PARTIAL)
9. KYC/Верификация (MISSING)
10. Чат с пользователями (MISSING)

Заказы:

11. Управление заказами (PARTIAL)
12. Назначение курьеров (DONE)

Споры и поддержка:

13. ✗ Споры и проблемный отдел (MISSING)
14. ✗ Поддержка/тикеты (MISSING)
15. ✗ FAQ/Центр помощи (MISSING)

Финансы:

16. ⚠ Финансы платформы (PARTIAL)
17. ✗ Купоны/акции платформы (MISSING)
18. ✗ Экспорт данных (MISSING)

Склад:

19. ✗ Складской модуль (MISSING)

Курьеры:

20. ✗ Модуль курьеров (MISSING)

Аналитика:

21. ⚠ Дашборд (PARTIAL)
22. ✗ Аналитика/отчёты (MISSING)
23. ✗ Аудит-лог (MISSING)

Настройки:

24. ✗ Глобальные настройки (MISSING)
25. ✗ Интеграции (MISSING)
26. ✗ Anti-fraud (MISSING)
27. ✗ Служебные режимы (MISSING)

Маркетинг:

28. ✗ Push-рассылки (MISSING)
29. ✗ Баннеры и реклама (MISSING)
30. ✗ Контент от платформы (MISSING)

6.2 Сводка по готовности функций

Роль	Всего функций	DONE	PARTIAL	MISSING	% готовности
Гость	5	5	0	0	100%
Покупатель	35	8	8	19	23% + 23%
Продавец	37	10	5	22	27% + 14%
Курьер	32	9	2	21	28% + 6%
Админ	30	2	4	24	7% + 13%
ИТОГО	139	34	19	86	24% + 14%

Общая готовность проекта: ~38% (DONE + PARTIAL)

7. БИЗНЕС-ЛОГИКА

7.1 Жизненный цикл заказа (Order Lifecycle)

```

PENDING (создан покупателем)
↓
CONFIRMED (подтвержден продавцом)
↓
ASSIGNED (назначен курьер)
↓
PICKED_UP (курьер забрал товар, QR-скан)
↓
IN_TRANSIT (курьер в пути)
↓
DELIVERED (доставлено, QR/код подтверждения)

Отмена возможна на любом этапе → CANCELLED
Проблемы с доставкой → DISPUTED

```

7.2 Escrow система (Безопасная сделка)

Двухфазная модель комиссий:

1. При создании/оплате заказа:

- Создается PAYMENT транзакция (PENDING → HELD)
- Создается PLATFORM_COMMISSION транзакция (HELD)
- Деньги удерживаются на платформе

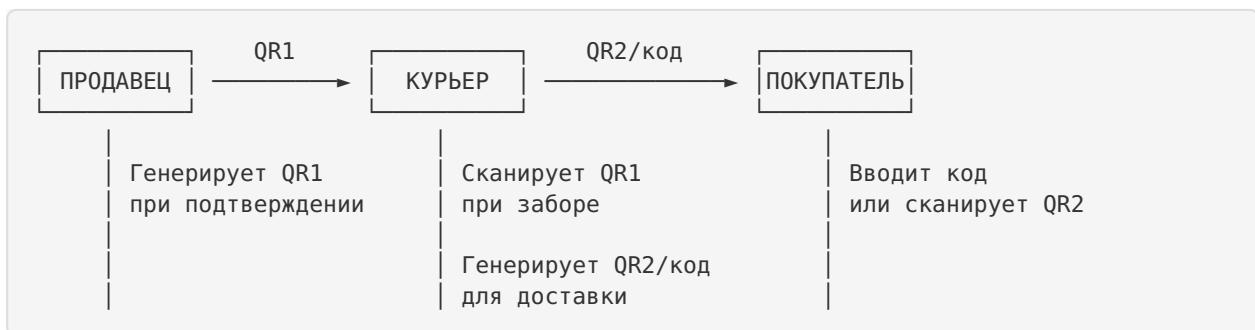
2. При подтверждении доставки:

- PLATFORM_COMMISSION → COMPLETED
- Создается SELLER_PAYOUT (COMPLETED)
- Создается COURIER_PAYOUT (COMPLETED)
- Распределение:
 - Продавец: totalAmount - platformCommission (5%) - courierFee
 - Курьер: courierFee (15,000 UZS)
 - Платформа: platformCommission (5%)

3. При отмене заказа:

- Создается REFUND для покупателя
- Создается COMMISSION_REVERSAL (реверс комиссии)

7.3 QR-цепочка подтверждений



QR-код содержит:

```
{
  "type": "seller_pickup" || "courier_delivery",
  "orderId": "uuid",
  "timestamp": "ISO8601",
  "signature": "HMAC-SHA256"
}
```

7.4 Бизнес-правила

Комиссии и тарифы:

- Комиссия платформы: **5%** от суммы товара
- Стоимость доставки: **15,000 UZS** (фиксированная)
- Минимальная сумма заказа: **50,000 UZS**

Лимиты:

- Максимум товаров в корзине: 50
- Максимум адресов: 10
- Максимум фото товара: 10
- Максимум видео товара: 5
- Длительность видео: 15-180 секунд
- Размер файла: 100 MB

Сроки:

- Подтверждение заказа продавцом: 24 часа
- Доставка: 1-3 дня
- Возврат: 14 дней
- Срок хранения сторис: 24 часа
- Срок открытия спора: 7 дней после доставки

Валидация:

- Телефон: +998XXXXXXXXX (9 цифр после кода)
- Пароль: минимум 8 символов, буквы + цифры
- Цена: > 0, целое число в UZS
- Количество: > 0, <= stock

8. ИНТЕГРАЦИИ (ПЛАНИРУЕМЫЕ)

8.1 SMS (Eskiz.uz)

```
POST https://notify.eskiz.uz/api/message/sms/send
{
  mobile_phone: "998901234567",
  message: "GoGoMarket: Kod podtverzhdeniya 123456",
  from: "4546"
}
```

Статус: ✘ НЕ РЕАЛИЗОВАНО (mock OTP)

8.2 Payme

```
POST https://checkout.paycom.uz/api
{
  method: "cards.create",
  params: {
    card: { number, expire },
    amount: 1500000, // В тийинах (15000 UZS * 100)
    save: true
  }
}
```

Статус: ✗ НЕ РЕАЛИЗОВАНО

8.3 Click

```
POST https://api.click.uz/v2/merchant/invoice/create
{
  service_id: "xxx",
  amount: 15000,
  phone_number: "998901234567",
  merchant_trans_id: "order-uuid"
}
```

Статус: ✗ НЕ РЕАЛИЗОВАНО

8.4 Firebase Cloud Messaging (Push)

```
POST https://fcm.googleapis.com/fcm/send
{
  to: "device_token",
  notification: {
    title: "Новый заказ",
    body: "Заказ #GGM-123456"
  },
  data: {
    type: "ORDER_CREATED",
    orderId: "uuid"
  }
}
```

Статус: ✗ НЕ РЕАЛИЗОВАНО (есть fcmToken в User модели)

8.5 DigitalOcean Spaces (S3-compatible)

```
PUT https://gogomarket.fra1.digitaloceanspaces.com/videos/uuid.mp4
Authorization: AWS4-HMAC-SHA256 ...
Content-Type: video/mp4
```

Статус: ⚠ ЧАСТИЧНО (есть AWS SDK, нет полной реализации)

9. БЕЗОПАСНОСТЬ

9.1 Аутентификация

- ✓ JWT токены (jsonwebtoken 9.0.3)
- ✓ Bcrypt хеширование паролей (salt rounds = 12)
- ⚠ Refresh токены (не реализовано)
- ✗ Rate limiting (не реализовано)

9.2 Авторизация

- ✓ Role-based access control (RBAC)
- ✓ Middleware проверки роли
- ✗ Детальные permissions (не реализовано)

9.3 Защита данных

- ✓ Helmet для HTTP заголовков
- ✓ CORS настройка
- ✓ Sequelize ORM (защита от SQL injection)
- ✓ QR-коды с HMAC-SHA256 подписью
- ✗ XSS sanitization (не реализовано)
- ✗ CSRF protection (не реализовано)

9.4 Биометрия (Flutter)

- ⚠ local_auth пакет подключен
 - ✗ Интеграция в checkout не завершена
-

10. ПРИОРИТЕТЫ ДЛЯ MVP

P0 - Критично для запуска (блокеры)

1. ✗ SMS верификация (Eskiz.uz) - сейчас mock OTP
2. ✗ Payme/Click интеграция - нет реальных платежей
3. ✗ Media upload (DO Spaces) - нет загрузки видео/фото
4. ✗ Push-уведомления (FCM) - нет уведомлений
5. ✗ Чат (buyer↔seller, buyer↔courier) - критично для коммуникации

P1 - Важно для UX

1. ✗ Варианты товара (размер, цвет) - базовая функция e-commerce
2. ✗ Избранное - стандарт для маркетплейсов
3. ✗ Подписки на продавцов - социальная составляющая
4. ✗ Трекинг на карте - важно для доставки
5. ⚠ Seller orders management UI - нужен полноценный интерфейс

P2 - После MVP

1. ✗ AI-заполнение карточек (Speech-to-Text)
2. ✗ Live-трансляции
3. ✗ Рекомендации

-
4. ✗ Складской модуль
 5. ✗ RBAC для админов
-

11. ЭКРАНЫ И СТРАНИЦЫ

11.1 Flutter Mobile (20+ экранов)

Auth:

- ✗ splash_screen.dart
- ✗ onboarding_screen.dart
- ✗ login_screen.dart
- ✗ register_screen.dart
- ✗ otp_screen.dart

Main:

- ✗ main_screen.dart (bottom navigation)
- ✗ home_screen.dart

Video:

- ✗ video_feed_screen.dart (TikTok-like)

Product:

- ✗ product_detail_screen.dart

Cart & Checkout:

- ✗ cart_screen.dart
- ✗ checkout_screen.dart

Orders:

- ✗ orders_screen.dart
- ✗ order_detail_screen.dart

Profile:

- ✗ profile_screen.dart
- ✗ edit_profile_screen.dart
- ✗ settings_screen.dart
- ✗ addresses_screen.dart

Seller:

- ✗ seller_dashboard_screen.dart

Courier:

- ✗ courier_dashboard_screen.dart

QR:

- ✗ qr_scanner_screen.dart

11.2 Web React (16 страниц)

Auth:

- ✗ LoginPage.tsx
- ✗ RegisterPage.tsx

Buyer:

- HomePage.tsx
- ProductsPage.tsx
- ProductDetailPage.tsx
- CartPage.tsx
- CheckoutPage.tsx
- OrdersPage.tsx

Seller:

- SellerDashboard.tsx
- SellerProducts.tsx
- ProductForm.tsx

Admin:

- AdminDashboard.tsx
 - AdminUsers.tsx
 - AdminOrders.tsx
 - AdminTransactions.tsx
 - FinancialOverview.tsx
-

12. КЛЮЧЕВЫЕ ВЫВОДЫ

12.1 Что работает (DONE - 27%)

Backend API:

- Аутентификация (JWT, bcrypt)
- CRUD товаров
- CRUD видео
- Создание и управление заказами
- QR-генерация и сканирование
- Базовая Escrow логика
- Транзакции

Frontend Mobile:

- Все основные экраны
- State management (Provider)
- API интеграция
- QR сканер
- Биометрия (частично)

Frontend Web:

- Все страницы для 3 ролей (buyer, seller, admin)
- Современный UI (Radix UI + Tailwind)
- Формы с валидацией (React Hook Form + Zod)

12.2 Что частично работает (PARTIAL - 23%)

⚠ Требует доработки:

- Корзина (нет синхронизации с backend)
- Escrow (логика есть, но не полная)
- Адресная книга (модель есть, CRUD API нет)

- Лента видео (UI есть, нет реальных видео)
- Seller/Courier dashboards (базовые, нужна аналитика)
- Оффлайн режим (сервис есть, не полностью интегрирован)

12.3 Что отсутствует (MISSING - 50%)

Критичные пробелы:

- SMS верификация (mock OTP)
- Payme/Click платежи
- Media upload (видео/фото)
- Push-уведомления
- Чат (buyer↔seller, courier↔buyer)
- Варианты товара (размер, цвет)
- Избранное
- Подписки на продавцов
- Трекинг на карте
- Купоны/промокоды
- Модерация контента
- KYC верификация
- Споры и возвраты (UI)
- Поддержка/тикеты
- Аналитика и отчеты
- RBAC для админов
- Live-трансляции
- AI-функции

12.4 Технический долг

-  Нет тестов (unit, integration, e2e)
-  Нет CI/CD
-  Нет мониторинга и логирования
-  Нет rate limiting
-  Нет CSRF protection
-  Нет XSS sanitization
-  Нет документации API (Swagger)
-  Нет миграций БД (Sequelize migrations)
-  Нет seed данных для разработки

13. РЕКОМЕНДАЦИИ ПО ПРОВЕРКЕ

13.1 Что проверить в первую очередь

Backend:

1.  Регистрация и логин работают
2.  OTP верификация (mock, не реальная)
3.  CRUD товаров
4.  Создание заказа
5.  QR-генерация

6. Escrow транзакции (проверить логику)
7. Платежи (Payme/Click не подключены)

Frontend Mobile:

1. Навигация между экранами
2. Регистрация/логин
3. Лента видео (UI есть, контент?)
4. Карточка товара
5. Корзина (локальная, не синхронизируется)
6. Оформление заказа
7. QR сканер

Frontend Web:

1. Buyer flow (просмотр, корзина, заказ)
2. Seller flow (добавление товара)
3. Admin flow (базовые функции, нет модерации)

13.2 Критические вопросы для уточнения

1. **Платежи:** Когда планируется интеграция Payme/Click?
2. **SMS:** Есть ли доступ к Eskiz.uz API?
3. **Media:** Где будут храниться видео/фото? (DO Spaces настроен?)
4. **Push:** Настроен ли Firebase проект?
5. **Чат:** Какая технология? (WebSocket, Firebase Realtime DB?)
6. **Deployment:** Где будет production? (текущий - 64.226.94.133)
7. **Database:** Есть ли production PostgreSQL?
8. **Домен:** Будет ли gogomarket.uz?

13.3 Тестовые сценарии

Сценарий 1: Покупка товара (Happy Path)

1. Регистрация покупателя
2. Просмотр ленты видео
3. Открытие карточки товара
4. Добавление в корзину
5. Оформление заказа (COD)
6. Продавец подтверждает заказ
7. Админ назначает курьера
8. Курьер сканирует QR забора
9. Курьер подтверждает доставку
10. Проверка транзакций (escrow)

Сценарий 2: Продавец добавляет товар

1. Регистрация продавца
2. Добавление товара (фото, описание, цена)
3. Загрузка видео с товаром
4. Проверка отображения в ленте

Сценарий 3: Курьер доставляет заказ

1. Регистрация курьера
2. Просмотр доступных заказов
3. Принятие заказа

4. Сканирование QR у продавца
 5. Доставка покупателю
 6. Подтверждение доставки (QR/код)
 7. Проверка баланса курьера
-

14. ФАЙЛЫ ДЛЯ ДЕТАЛЬНОГО ИЗУЧЕНИЯ

14.1 Документация

- README.md - общее описание
- REQUIREMENTS.md - полное ТЗ (139 функций)
- FUNCTIONAL_LOGIC.md - логика работы, API контракты
- GAP_ANALYSIS.md - анализ готовности
- WORK_LOG.md - история разработки

14.2 Backend ключевые файлы

- backend/src/app.ts - точка входа
- backend/src/config/database.ts - подключение к БД
- backend/src/models/ - все модели данных
- backend/src/controllers/orderController.ts - логика заказов (самый сложный)
- backend/src/services/qrService.ts - QR-генерация
- backend/src/middleware/auth.ts - аутентификация

14.3 Frontend Mobile ключевые файлы

- frontend/lib/main.dart - точка входа
- frontend/lib/config/api_config.dart - API endpoints
- frontend/lib/providers/ - state management
- frontend/lib/screens/video/video_feed_screen.dart - лента видео
- frontend/lib/screens/checkout/checkout_screen.dart - оформление заказа
- frontend/lib/services/qr_service.dart - QR сканирование

14.4 Frontend Web ключевые файлы

- web/gogomarket-web/src/App.tsx - главный компонент
 - web/gogomarket-web/src/context/AuthContext.tsx - аутентификация
 - web/gogomarket-web/src/pages/buyer/CheckoutPage.tsx - оформление заказа
 - web/gogomarket-web/src/pages/admin/AdminDashboard.tsx - админ панель
-

15. КОНТАКТЫ И ССЫЛКИ

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

Ветка: devin/1767373941-gogomarket-mvp

Работающий сайт:

- <http://64.226.94.133>
- <https://64-226-94-133.sslip.io>

API Endpoint: <https://64-226-94-133.sslip.io/api/v1>

Технологии:

- Backend: Node.js + Express + TypeScript + PostgreSQL + Sequelize
 - Mobile: Flutter 3.0+
 - Web: React 18 + TypeScript + Vite + Tailwind CSS
-

Дата анализа: 4 января 2026

Аналитик: AI Assistant

Версия документа: 1.0

ПРИЛОЖЕНИЕ: Структура файлов проекта (218 файлов)

Полный список файлов (нажмите для раскрытия)

```
.gitignore
FUNCTIONAL_LOGIC.md
GAP_ANALYSIS.md
README.md
REQUIREMENTS.md
WORK_LOG.md
backend/.env.example
backend/package-lock.json
backend/package.json
backend/src/app.ts
backend/src/config/database.ts
backend/src/config/index.ts
backend/src/controllers/authController.ts
backend/src/controllers/chatController.ts
backend/src/controllers/orderController.ts
backend/src/controllers/paymentController.ts
backend/src/controllers/productController.ts
backend/src/controllers/uploadController.ts
backend/src/controllers/videoController.ts
backend/src/middleware/auth.ts
backend/src/middleware/errorHandler.ts
backend/src/middleware/validation.ts
backend/src/models/Address.ts
backend/src/models/Dispute.ts
backend/src/models/Message.ts
backend/src/models/Order.ts
backend/src/models/Product.ts
backend/src/models/Review.ts
backend/src/models/Transaction.ts
backend/src/models/User.ts
backend/src/models/Video.ts
backend/src/models/index.ts
backend/src/routes/authRoutes.ts
backend/src/routes/chatRoutes.ts
backend/src/routes/index.ts
backend/src/routes/orderRoutes.ts
backend/src/routes/paymentRoutes.ts
backend/src/routes/productRoutes.ts
backend/src/routes/uploadRoutes.ts
backend/src/routes/videoRoutes.ts
backend/src/services/notificationService.ts
backend/src/services/qrService.ts
backend/src/services/smsService.ts
backend/src/services/uploadService.ts
backend/src/types/index.ts
backend/tsconfig.json
frontend/.gitignore
frontend/README.md
frontend/analysis_options.yaml
frontend/android/.gitignore
frontend/android/app/build.gradle.kts
frontend/android/app/src/debug/AndroidManifest.xml
frontend/android/app/src/main/AndroidManifest.xml
frontend/android/app/src/main/kotlin/com/example/gogomarket/MainActivity.kt
frontend/android/app/src/main/res/drawable-v21/launch_background.xml
frontend/android/app/src/main/res/drawable/launch_background.xml
frontend/android/app/src/main/res/mipmap-hdpi/ic_launcher.png
frontend/android/app/src/main/res/mipmap-mdpi/ic_launcher.png
frontend/android/app/src/main/res/mipmap-xhdpi/ic_launcher.png
frontend/android/app/src/main/res/mipmap-xxhdpi/ic_launcher.png
frontend/android/app/src/main/res/mipmap-xxxhdpi/ic_launcher.png
frontend/android/app/src/main/res/values-night/styles.xml
```

```
frontend/android/app/src/main/res/values/styles.xml
frontend/android/app/src/profile/AndroidManifest.xml
frontend/android/build.gradle.kts
frontend/android/gradle.properties
frontend/android/gradle/wrapper/gradle-wrapper.properties
frontend/android/settings.gradle.kts
frontend/assets/fonts/.gitkeep
frontend/assets/images/.gitkeep
frontend/ios/Flutter/Generated.xcconfig
frontend/ios/Flutter/ephemeral/flutter_lldb_helper.py
frontend/ios/Flutter/ephemeral/flutter_lldbinit
frontend/ios/Flutter/flutter_export_environment.sh
frontend/ios/Runner/GeneratedPluginRegistrant.h
frontend/ios/Runner/GeneratedPluginRegistrant.m
frontend/lib/config/api_config.dart
frontend/lib/config/routes.dart
frontend/lib/config/theme.dart
frontend/lib/l10n/app_localizations.dart
frontend/lib/main.dart
frontend/lib/models/order.dart
frontend/lib/models/product.dart
frontend/lib/models/user.dart
frontend/lib/models/video.dart
frontend/lib/providers/auth_provider.dart
frontend/lib/providers/locale_provider.dart
frontend/lib/providers/order_provider.dart
frontend/lib/providers/product_provider.dart
frontend/lib/providers/video_provider.dart
frontend/lib/screens/auth/login_screen.dart
frontend/lib/screens/auth/otp_screen.dart
frontend/lib/screens/auth/register_screen.dart
frontend/lib/screens/cart/cart_screen.dart
frontend/lib/screens/checkout/checkout_screen.dart
frontend/lib/screens/courier/courier_dashboard_screen.dart
frontend/lib/screens/home/home_screen.dart
frontend/lib/screens/main_screen.dart
frontend/lib/screens/onboarding_screen.dart
frontend/lib/screens/order/order_detail_screen.dart
frontend/lib/screens/order/orders_screen.dart
frontend/lib/screens/product/product_detail_screen.dart
frontend/lib/screens/profile/addresses_screen.dart
frontend/lib/screens/profile/edit_profile_screen.dart
frontend/lib/screens/profile/profile_screen.dart
frontend/lib/screens/profile/settings_screen.dart
frontend/lib/screens/qr/qr_scanner_screen.dart
frontend/lib/screens/seller/seller_dashboard_screen.dart
frontend/lib/screens/splash_screen.dart
frontend/lib/screens/video/video_feed_screen.dart
frontend/lib/services/api_service.dart
frontend/lib/services/biometric_service.dart
frontend/lib/services/offline_service.dart
frontend/lib/services/qr_service.dart
frontend/lib/utils/currency_formatter.dart
frontend/lib/widgets/live_seller_avatar.dart
frontend/lib/widgets/order_card.dart
frontend/lib/widgets/product_card.dart
frontend/lib/widgets/video_player_item.dart
frontend/pubspec.lock
frontend/pubspec.yaml
web/gogomarket-web/.gitignore
web/gogomarket-web/README.md
web/gogomarket-web/components.json
web/gogomarket-web/eslint.config.js
```

```
web/gogomarket-web/index.html
web/gogomarket-web/package-lock.json
web/gogomarket-web/package.json
web/gogomarket-web/postcss.config.js
web/gogomarket-web/public/vite.svg
web/gogomarket-web/src/App.css
web/gogomarket-web/src/App.tsx
web/gogomarket-web/src/assets/react.svg
web/gogomarket-web/src/components/BottomNav.tsx
web/gogomarket-web/src/components/Layout.tsx
web/gogomarket-web/src/components/ui/accordion.tsx
web/gogomarket-web/src/components/ui/alert-dialog.tsx
web/gogomarket-web/src/components/ui/alert.tsx
web/gogomarket-web/src/components/ui/aspect-ratio.tsx
web/gogomarket-web/src/components/ui/avatar.tsx
web/gogomarket-web/src/components/ui/badge.tsx
web/gogomarket-web/src/components/ui/breadcrumb.tsx
web/gogomarket-web/src/components/ui/button-group.tsx
web/gogomarket-web/src/components/ui/button.tsx
web/gogomarket-web/src/components/ui/card.tsx
web/gogomarket-web/src/components/ui/carousel.tsx
web/gogomarket-web/src/components/ui/checkbox.tsx
web/gogomarket-web/src/components/ui/collapsible.tsx
web/gogomarket-web/src/components/ui/command.tsx
web/gogomarket-web/src/components/ui/context-menu.tsx
web/gogomarket-web/src/components/ui/dialog.tsx
web/gogomarket-web/src/components/ui/drawer.tsx
web/gogomarket-web/src/components/ui/dropdown-menu.tsx
web/gogomarket-web/src/components/ui/empty.tsx
web/gogomarket-web/src/components/ui/field.tsx
web/gogomarket-web/src/components/ui/form.tsx
web/gogomarket-web/src/components/ui/hover-card.tsx
web/gogomarket-web/src/components/ui/input-group.tsx
web/gogomarket-web/src/components/ui/input-otp.tsx
web/gogomarket-web/src/components/ui/input.tsx
web/gogomarket-web/src/components/ui/item.tsx
web/gogomarket-web/src/components/ui/kbd.tsx
web/gogomarket-web/src/components/ui/label.tsx
web/gogomarket-web/src/components/ui/menu-bar.tsx
web/gogomarket-web/src/components/ui/navigation-menu.tsx
web/gogomarket-web/src/components/ui/pagination.tsx
web/gogomarket-web/src/components/ui/popover.tsx
web/gogomarket-web/src/components/ui/progress.tsx
web/gogomarket-web/src/components/ui/radio-group.tsx
web/gogomarket-web/src/components/ui/scroll-area.tsx
web/gogomarket-web/src/components/ui/select.tsx
web/gogomarket-web/src/components/ui/sePARATOR.tsx
web/gogomarket-web/src/components/ui/sheet.tsx
web/gogomarket-web/src/components/ui/sidebar.tsx
web/gogomarket-web/src/components/ui/skeleton.tsx
web/gogomarket-web/src/components/ui/slider.tsx
web/gogomarket-web/src/components/ui/sonner.tsx
web/gogomarket-web/src/components/ui/spinner.tsx
web/gogomarket-web/src/components/ui/switch.tsx
web/gogomarket-web/src/components/ui/table.tsx
web/gogomarket-web/src/components/ui/tabs.tsx
web/gogomarket-web/src/components/ui/textarea.tsx
web/gogomarket-web/src/components/ui/toast.tsx
web/gogomarket-web/src/components/ui/toaster.tsx
web/gogomarket-web/src/components/ui/toggle-group.tsx
web/gogomarket-web/src/components/ui/toggle.tsx
web/gogomarket-web/src/components/ui/tooltip.tsx
web/gogomarket-web/src/context/AuthContext.tsx
```

```
web/gogomarket-web/src-contexts/CartContext.tsx
web/gogomarket-web/src/hooks/use-mobile.tsx
web/gogomarket-web/src/hooks/use-toast.ts
web/gogomarket-web/src/index.css
web/gogomarket-web/src/lib/utils.ts
web/gogomarket-web/src/main.tsx
web/gogomarket-web/src/pages/admin/AdminDashboard.tsx
web/gogomarket-web/src/pages/admin/AdminOrders.tsx
web/gogomarket-web/src/pages/admin/AdminTransactions.tsx
web/gogomarket-web/src/pages/admin/AdminUsers.tsx
web/gogomarket-web/src/pages/admin/FinancialOverview.tsx
web/gogomarket-web/src/pages/auth/LoginPage.tsx
web/gogomarket-web/src/pages/auth/RegisterPage.tsx
web/gogomarket-web/src/pages/buyer/CartPage.tsx
web/gogomarket-web/src/pages/buyer/CheckoutPage.tsx
web/gogomarket-web/src/pages/buyer/HomePage.tsx
web/gogomarket-web/src/pages/buyer/OrdersPage.tsx
web/gogomarket-web/src/pages/buyer/ProductDetailPage.tsx
web/gogomarket-web/src/pages/buyer/ProductsPage.tsx
web/gogomarket-web/src/pages/seller/ProductForm.tsx
web/gogomarket-web/src/pages/seller/SellerDashboard.tsx
web/gogomarket-web/src/pages/seller/SellerProducts.tsx
web/gogomarket-web/src/services/api.ts
web/gogomarket-web/src/types/index.ts
web/gogomarket-web/src/vite-env.d.ts
web/gogomarket-web/tailwind.config.js
web/gogomarket-web/tsconfig.app.json
web/gogomarket-web/tsconfig.json
web/gogomarket-web/tsconfig.node.json
web/gogomarket-web/vite.config.ts
```

КОНЕЦ АНАЛИЗА