# TopInBeauty → 3GIS: План адаптации архитектуры

## 📊 Анализ существующей архитектуры TopInBeauty

### Технологический стек (готов к использованию)

{

"framework": "Next.js 14 + App Router",

"telegram": "@telegram-apps/sdk v3.x + @telegram-apps/sdk-react",

"database": "Prisma + PostgreSQL",

"ui": "shadcn/ui + Radix UI + Tailwind CSS",

"state": "Zustand + React Query",

"animations": "Framer Motion",

"files": "AWS S3 + Sharp для изображений",

"auth": "JWT + Telegram WebApp initData"

}

### Структура проекта

topinbeauty/

├── src/app/

│ ├── (auth)/ # Группа авторизации

│ ├── (website)/ # Публичный сайт

│ ├── (standalone)/ # Standalone режим

│ ├── tg/ # Telegram Mini App

│ ├── admin/ # Админ панель

│ ├── api/ # API endpoints

│ └── m/ # Короткие ссылки

├── src/components/

│ ├── telegram/ # Telegram-специфичные компоненты

│ ├── ui/ # shadcn/ui компоненты

│ └── [domains]/ # Доменные компоненты

└── prisma/

└── schema.prisma # Database schema

### Ключевые особенности архитектуры

#### 1. Multi-Context Architecture

TopInBeauty использует **3 различных контекста**:

* **Website** (/) - обычный веб-сайт
* **Telegram** (/tg) - Telegram Mini App
* **Admin** (/admin) - админ панель

#### 2. Intelligent Client Provider

// ClientProvider определяет контекст по URL

function determineContext(pathname: string): 'website' | 'telegram' | 'admin' {

if (pathname.startsWith('/admin')) return 'admin';

if (pathname.startsWith('/tg')) return 'telegram';

return 'website';

}

#### 3. Telegram SDK v3.x Integration

* Автоматическая инициализация через initTelegramSDK()
* Graceful fallback для development
* Haptic feedback и native кнопки
* Viewport и SafeArea обработка

## 🔄 План адаптации для 3GIS

### Этап 1: Структурная адаптация (День 1)

#### 1.1 Создание нового проекта

# Клонируем структуру (без .git)

cp -r D:/dev/tib3 D:/dev/3gis

cd D:/dev/3gis

# Очищаем от TopInBeauty специфики

rm -rf .git node\_modules

rm .env

# Обновляем package.json

#### 1.2 Обновление package.json

{

"name": "3gis-usa",

"version": "0.1.0",

"description": "Russian-speaking business directory for USA",

// ... остальные зависимости такие же

}

#### 1.3 Обновление конфигурации

// next.config.mjs - обновить домены и URL

const nextConfig = {

env: {

NEXT\_PUBLIC\_APP\_URL: "https://3gis.us",

NEXT\_PUBLIC\_TELEGRAM\_BOT\_USERNAME: "ThreeGIS\_bot",

// ...

}

}

### Этап 2: Адаптация базы данных (День 2)

#### 2.1 Новая Prisma схема для 3GIS

// prisma/schema.prisma - адаптированная версия

enum UserRole {

GUEST

USER

BUSINESS\_OWNER

ADMIN

}

enum BusinessStatus {

PENDING // На модерации

ACTIVE // Активен

SUSPENDED // Приостановлен

REJECTED // Отклонен

}

enum PremiumTier {

FREE

BASIC // $49/месяц

STANDARD // $149/месяц

PREMIUM // $299/месяц

}

model User {

id Int @id @default(autoincrement())

telegramId String @unique

username String?

firstName String @default("")

lastName String @default("")

role UserRole @default(USER)

avatar String?

createdAt DateTime @default(now())

// Геолокация

city City? @relation(fields: [cityId], references: [id])

cityId Int?

// Связи

businesses Business[] @relation("BusinessOwner")

favorites BusinessFavorite[]

reviews BusinessReview[]

@@map("users")

}

model Business {

id Int @id @default(autoincrement())

name String

nameEn String?

description String?

descriptionEn String?

// Категория (заменяет Service)

category BusinessCategory @relation(fields: [categoryId], references: [id])

categoryId Int

// Адрес и контакты

address String

city City @relation(fields: [cityId], references: [id])

cityId Int

state String @default("NY")

zipCode String?

phone String?

website String?

email String?

// Геолокация

latitude Float?

longitude Float?

// Время работы (JSON)

businessHours Json? // {mon: "9:00-18:00", tue: "9:00-18:00", ...}

// Особенности для русскоговорящих

languages String[] @default(["ru", "en"])

acceptsCrypto Boolean @default(false)

hasParking Boolean @default(false)

hasWiFi Boolean @default(false)

// Статус и верификация

status BusinessStatus @default(PENDING)

isVerified Boolean @default(false)

// Премиум функции

premiumTier PremiumTier @default(FREE)

premiumUntil DateTime?

// Статистика

viewCount Int @default(0)

rating Float @default(0)

reviewCount Int @default(0)

// Владелец

owner User @relation("BusinessOwner", fields: [ownerId], references: [id])

ownerId Int

// Связи

photos BusinessPhoto[]

reviews BusinessReview[]

favorites BusinessFavorite[]

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

@@map("businesses")

}

model BusinessCategory {

id Int @id @default(autoincrement())

name String @unique

nameEn String @unique

slug String @unique

icon String // emoji или название иконки

description String?

order Int @default(0)

businesses Business[]

@@map("business\_categories")

}

model City {

id Int @id @default(autoincrement())

name String @unique

state String

country String @default("USA")

latitude Float?

longitude Float?

population Int?

users User[]

businesses Business[]

@@map("cities")

}

model BusinessPhoto {

id Int @id @default(autoincrement())

url String

caption String?

order Int @default(0)

business Business @relation(fields: [businessId], references: [id])

businessId Int

createdAt DateTime @default(now())

@@map("business\_photos")

}

model BusinessReview {

id Int @id @default(autoincrement())

rating Int // 1-5 звезд

comment String?

isVerified Boolean @default(false)

business Business @relation(fields: [businessId], references: [id])

businessId Int

user User @relation(fields: [userId], references: [id])

userId Int

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

@@unique([businessId, userId]) // Один отзыв от пользователя

@@map("business\_reviews")

}

model BusinessFavorite {

id Int @id @default(autoincrement())

business Business @relation(fields: [businessId], references: [id])

businessId Int

user User @relation(fields: [userId], references: [id])

userId Int

createdAt DateTime @default(now())

@@unique([businessId, userId])

@@map("business\_favorites")

}

#### 2.2 Seed данные для тестирования

// prisma/seed.ts

const categories = [

{ name: "Рестораны и кафе", nameEn: "Restaurants", slug: "restaurants", icon: "🍽️" },

{ name: "Медицина", nameEn: "Healthcare", slug: "healthcare", icon: "⚕️" },

{ name: "Юристы", nameEn: "Legal Services", slug: "legal", icon: "⚖️" },

{ name: "Красота", nameEn: "Beauty", slug: "beauty", icon: "💄" },

{ name: "Автосервисы", nameEn: "Auto Services", slug: "auto", icon: "🔧" },

{ name: "Финансы", nameEn: "Finance", slug: "finance", icon: "🏦" },

{ name: "Образование", nameEn: "Education", slug: "education", icon: "🎓" },

{ name: "Недвижимость", nameEn: "Real Estate", slug: "realestate", icon: "🏠" }

];

const cities = [

{ name: "New York", state: "NY", latitude: 40.7128, longitude: -74.0060 },

{ name: "Los Angeles", state: "CA", latitude: 34.0522, longitude: -118.2437 },

{ name: "Chicago", state: "IL", latitude: 41.8781, longitude: -87.6298 },

// ...

];

### Этап 3: Адаптация API (День 3-4)

#### 3.1 Новые API endpoints

// src/app/api/businesses/route.ts

export async function GET(request: NextRequest) {

const { searchParams } = new URL(request.url);

const category = searchParams.get('category');

const city = searchParams.get('city');

const search = searchParams.get('search');

const page = parseInt(searchParams.get('page') || '1');

const businesses = await prisma.business.findMany({

where: {

status: 'ACTIVE',

...(category && { category: { slug: category } }),

...(city && { city: { name: city } }),

...(search && {

OR: [

{ name: { contains: search, mode: 'insensitive' } },

{ description: { contains: search, mode: 'insensitive' } }

]

})

},

include: {

category: true,

city: true,

photos: { orderBy: { order: 'asc' }, take: 1 },

\_count: { select: { reviews: true, favorites: true } }

},

orderBy: [

{ premiumTier: 'desc' }, // Премиум сначала

{ rating: 'desc' },

{ createdAt: 'desc' }

],

take: 20,

skip: (page - 1) \* 20

});

return NextResponse.json(businesses);

}

// src/app/api/businesses/[id]/route.ts - детали заведения

// src/app/api/categories/route.ts - список категорий

// src/app/api/cities/route.ts - список городов

### Этап 4: Адаптация UI компонентов (День 4-5)

#### 4.1 Новые компоненты для 3GIS

// src/components/business/BusinessCard.tsx

interface BusinessCardProps {

business: {

id: number;

name: string;

category: { name: string; icon: string };

city: { name: string };

rating: number;

reviewCount: number;

photos: { url: string }[];

premiumTier: string;

languages: string[];

phone?: string;

};

}

export function BusinessCard({ business }: BusinessCardProps) {

return (

<div className="bg-white rounded-xl border border-gray-200 p-4 mb-3">

{/\* Фото + Premium badge \*/}

<div className="relative mb-3">

{business.photos[0] && (

<img src={business.photos[0].url} className="w-full h-32 object-cover rounded-lg" />

)}

{business.premiumTier !== 'FREE' && (

<div className="absolute top-2 right-2 bg-gold text-black px-2 py-1 rounded-md text-xs font-bold">

PREMIUM

</div>

)}

</div>

{/\* Название и категория \*/}

<div className="mb-2">

<h3 className="font-semibold text-gray-900">{business.name}</h3>

<p className="text-sm text-gray-600 flex items-center">

<span className="mr-1">{business.category.icon}</span>

{business.category.name}

</p>

</div>

{/\* Рейтинг и отзывы \*/}

<div className="flex items-center mb-2">

<div className="flex text-yellow-400">

★ {business.rating.toFixed(1)}

</div>

<span className="text-sm text-gray-500 ml-2">

({business.reviewCount} отзывов)

</span>

</div>

{/\* Локация \*/}

<p className="text-sm text-gray-600 mb-3">📍 {business.city.name}</p>

{/\* Языки и особенности \*/}

<div className="flex flex-wrap gap-1 mb-3">

{business.languages.includes('ru') && (

<span className="bg-blue-100 text-blue-800 px-2 py-1 rounded-md text-xs">

🇷🇺 Русский

</span>

)}

</div>

{/\* Кнопки действий \*/}

<div className="flex gap-2">

<button className="flex-1 bg-blue-500 text-white py-2 px-4 rounded-lg text-sm">

📞 Позвонить

</button>

<button className="flex-1 bg-gray-100 text-gray-700 py-2 px-4 rounded-lg text-sm">

🗺️ Маршрут

</button>

</div>

</div>

);

}

#### 4.2 Адаптация существующих компонентов

// src/components/categories/CategoryGrid.tsx - адаптировать под 8 категорий

// src/components/search/SearchInput.tsx - поиск по бизнесам

// src/components/filters/FilterChips.tsx - фильтры для 3GIS

### Этап 5: Адаптация страниц (День 5-6)

#### 5.1 Главная страница Telegram App

// src/app/tg/page.tsx

export default async function ThreeGISHomePage() {

const { businesses, categories, cities } = await getInitialData();

return (

<div className="p-4">

{/\* Header с логотипом и городом \*/}

<div className="mb-6">

<h1 className="text-2xl font-bold text-center mb-2">

3<span className="text-yellow-500">GIS</span>

</h1>

<p className="text-center text-gray-600 text-sm">

Твой проводник в Америке

</p>

</div>

{/\* Поиск \*/}

<SearchInput placeholder="Поиск ресторанов, врачей, услуг..." />

{/\* Фильтры \*/}

<FilterChips />

{/\* Сетка категорий 2x4 \*/}

<CategoryGrid categories={categories} />

{/\* Список заведений \*/}

<BusinessList businesses={businesses} />

</div>

);

}

#### 5.2 Страница заведения

// src/app/tg/business/[id]/page.tsx

export default async function BusinessPage({ params }: { params: { id: string } }) {

const business = await getBusinessById(params.id);

return (

<div>

{/\* Фото галерея \*/}

<PhotoGallery photos={business.photos} />

{/\* Основная информация \*/}

<BusinessInfo business={business} />

{/\* Отзывы \*/}

<ReviewsList businessId={business.id} />

{/\* Кнопки действий (fixed bottom) \*/}

<ActionButtons business={business} />

</div>

);

}

### Этап 6: Платежи Telegram Stars (День 6-7)

#### 6.1 Интеграция Telegram Stars

// src/components/premium/TelegramStarsPurchase.tsx

import { useTelegramWebApp } from '@telegram-apps/sdk-react';

const PREMIUM\_PLANS = {

basic: { stars: 500, price: 49, name: 'Базовый' },

standard: { stars: 1500, price: 149, name: 'Стандарт' },

premium: { stars: 3000, price: 299, name: 'Премиум' }

};

export function TelegramStarsPurchase({ businessId, plan }: Props) {

const webApp = useTelegramWebApp();

const purchasePremium = async () => {

const planConfig = PREMIUM\_PLANS[plan];

const invoice = {

title: `3GIS ${planConfig.name}`,

description: `Премиум размещение на 1 месяц`,

currency: 'XTR', // Telegram Stars

prices: [{

label: `${planConfig.name} план`,

amount: planConfig.stars

}]

};

webApp?.openInvoice(invoice, (status) => {

if (status === 'paid') {

// Активировать премиум через API

activatePremium(businessId, plan);

}

});

};

return (

<button onClick={purchasePremium} className="bg-yellow-500 text-black py-3 px-6 rounded-lg">

⭐ Купить за {PREMIUM\_PLANS[plan].stars} Stars

</button>

);

}

#### 6.2 API для обработки платежей

// src/app/api/payments/telegram-stars/route.ts

export async function POST(request: NextRequest) {

const { businessId, plan, telegramPaymentId } = await request.json();

// Верифицируем платеж через Telegram API

const paymentVerified = await verifyTelegramPayment(telegramPaymentId);

if (paymentVerified) {

// Активируем премиум

await prisma.business.update({

where: { id: businessId },

data: {

premiumTier: plan.toUpperCase(),

premiumUntil: new Date(Date.now() + 30 \* 24 \* 60 \* 60 \* 1000) // +30 дней

}

});

return NextResponse.json({ success: true });

}

return NextResponse.json({ error: 'Payment verification failed' }, { status: 400 });

}

## 🚀 План выполнения по дням

### День 1: Настройка проекта

* [x] Копирование и очистка TopInBeauty структуры
* [x] Обновление package.json и конфигов
* [x] Создание нового Git репозитория
* [x] Настройка окружения разработки

### День 2: База данных

* [ ] Адаптация Prisma схемы под 3GIS
* [ ] Создание seed файлов с категориями и городами
* [ ] Настройка Supabase/PostgreSQL
* [ ] Запуск миграций

### День 3: API Endpoints

* [ ] Создание /api/businesses endpoints
* [ ] Адаптация /api/categories и /api/cities
* [ ] Настройка поиска и фильтрации
* [ ] Тестирование API через Postman

### День 4: Telegram компоненты

* [ ] Адаптация Telegram SDK инициализации
* [ ] Создание BusinessCard компонента
* [ ] Адаптация CategoryGrid под 8 категорий
* [ ] Настройка Search и Filters

### День 5: Страницы приложения

* [ ] Главная страница /tg
* [ ] Список заведений /tg/businesses
* [ ] Детальная страница /tg/business/[id]
* [ ] Форма добавления /tg/add-business

### День 6: Telegram Stars

* [ ] Интеграция Telegram Stars API
* [ ] Создание PremiumPurchase компонентов
* [ ] API для обработки платежей
* [ ] Тестирование платежей

### День 7: Финальная доработка

* [ ] Тестирование всего функционала
* [ ] Оптимизация производительности
* [ ] Deployment на Vercel
* [ ] Настройка домена 3gis.us

## 🔧 Необходимые изменения в конфигурации

### Environment Variables (.env)

# Database

DATABASE\_URL="postgresql://..."

# Telegram

NEXT\_PUBLIC\_TELEGRAM\_BOT\_USERNAME="ThreeGIS\_bot"

TELEGRAM\_BOT\_TOKEN="your\_bot\_token"

# App URLs

NEXT\_PUBLIC\_APP\_URL="https://3gis.us"

NEXT\_PUBLIC\_WEBSITE\_URL="https://3gis.us"

# File Storage (можно оставить тот же S3)

S3\_BUCKET\_NAME="3gis-uploads"

AWS\_ACCESS\_KEY\_ID="..."

AWS\_SECRET\_ACCESS\_KEY="..."

# Google Maps (для геолокации)

GOOGLE\_MAPS\_API\_KEY="..."

### Telegram Bot Setup

// Создать нового бота через @BotFather

// Команды:

// /newbot

// Название: 3GIS - Russian Business Directory

// Username: ThreeGIS\_bot

// Настроить Mini App:

// /newapp

// Выбрать бота ThreeGIS\_bot

// URL: https://3gis.us/tg

## 📊 Сравнение с TopInBeauty

| Аспект | TopInBeauty | 3GIS |
| --- | --- | --- |
| **Доменная модель** | Мастера → Услуги → Записи | Бизнесы → Категории → Отзывы |
| **Целевая аудитория** | Клиенты красоты | Русскоговорящие в США |
| **Монетизация** | Премиум для мастеров | Премиум для бизнесов |
| **Географический охват** | Россия, города | США, штаты и города |
| **Основной функционал** | Бронирование услуг | Поиск и информация |
| **Платежи** | Возможно рубли | Telegram Stars → USD |

## ✅ Готовые компоненты для переиспользования

### Можно использовать как есть:

* ✅ Telegram SDK инициализация
* ✅ ClientProvider архитектура
* ✅ UI компоненты (shadcn/ui)
* ✅ Image upload с AWS S3
* ✅ Authentication через Telegram
* ✅ Layout и navigation
* ✅ Error handling
* ✅ Loading states

### Требуют адаптации:

* 🔄 Prisma schema (Service → Business)
* 🔄 API endpoints (новые ресурсы)
* 🔄 Основные страницы (новый контент)
* 🔄 Search и Filters (новые критерии)
* 🔄 Категории (8 вместо beauty)

### Нужно создать новые:

* 🆕 BusinessCard компонент
* 🆕 Telegram Stars интеграция
* 🆕 Premium tiers логика
* 🆕 US Cities и States данные
* 🆕 Reviews система

## 🎯 Заключение

**TopInBeauty предоставляет идеальную основу для 3GIS:**

1. **Архитектура готова** - Multi-context setup идеален
2. **Telegram SDK настроен** - работает с v3.x
3. **Database ORM готов** - нужна только адаптация схемы
4. **UI компоненты готовы** - shadcn/ui + Tailwind CSS
5. **API структура готова** - нужно изменить только endpoints

**Экономия времени:** ~80% кода можно переиспользовать с минимальными изменениями. Основная работа - адаптация доменной модели и создание нового контента.

**Готовы начинать адаптацию уже сегодня!** 🚀