

# ROADMAP: All-in-One Video Content Factory

**Дата создания:** 20 октября 2025

**Версия:** 1.0

**Статус:** Планирование

---

## СОДЕРЖАНИЕ

1. Обзор проекта
  2. Архитектура системы
  3. Технологический стек
  4. База данных
  5. Модули и компоненты
  6. API Endpoints
  7. План реализации (этапы)
  8. Интеграции и зависимости
  9. Стоимость и ресурсы
  10. Метрики успеха
- 

## ОБЗОР ПРОЕКТА

### Миссия

Создать первую в мире **ALL-IN-ONE платформу** для производства видео-контента (Shorts/Reels), объединяющую все этапы от идеи до публикации в одном интерфейсе.

### Целевая аудитория

- Контент-мейкеры
- SMM-специалисты
- Предприниматели
- Инфобизнесмены
- Блогеры
- Маркетинговые агентства

### Ключевое отличие от конкурентов

**ВСЁ В ОДНОМ:** Голос + Персонажи + Сценарии + Монтаж + Публикация

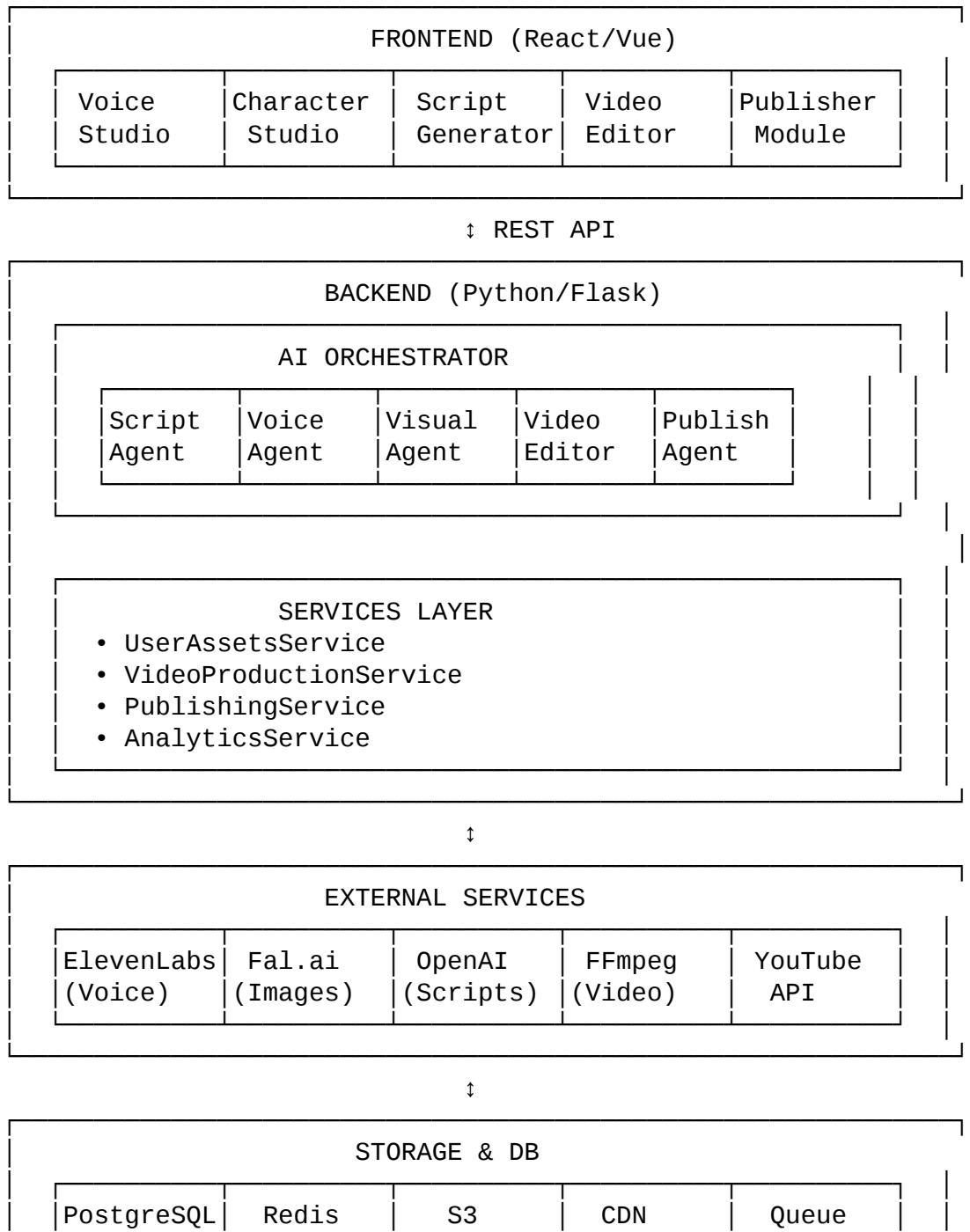
### Конкурентные преимущества

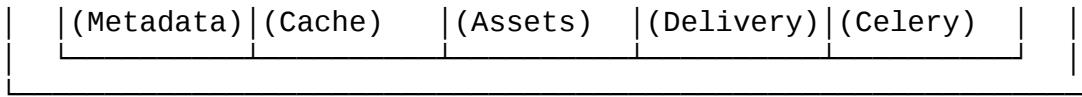
1. ☐ Клонирование голоса пользователя (личный контент)
2. ☐ Создание персональных AI-персонажей (консистентность)
3. ☐ Массовое производство (50+ роликов за раз)
4. ☐ Мультиплатформенная публикация (1 кнопка)

5. ☐ Многопользовательский режим (каждый со своими активами)
  6. ☐ Полная автоматизация workflow
- 

## ☐ АРХИТЕКТУРА СИСТЕМЫ

### High-Level Architecture





## □ ТЕХНОЛОГИЧЕСКИЙ СТЕК

### Backend

**Framework:** Flask (Python 3.11+)  
**API:** Flask-RESTX (Swagger docs)  
**Auth:** JWT (Flask-JWT-Extended)  
**Database ORM:** SQLAlchemy  
**Task Queue:** Celery + Redis  
**Video Processing:** FFmpeg, MoviePy  
**WebSocket:** Flask-SocketIO (для progress updates)

### External AI Services

#### Voice Cloning:

**Primary:** ElevenLabs API  
**Fallback:** PlayHT 3.0

#### Character Generation:

**Primary:** Fal.ai FLUX Pro 1.1  
**Alternative:** Leonardo.ai Phoenix  
**Fallback:** Midjourney API

#### Script Generation:

**Primary:** OpenAI GPT-4  
**Alternative:** Claude 3.5 Sonnet

#### Speech-to-Text (для субтитров):

**Primary:** OpenAI Whisper  
**Alternative:** AssemblyAI

### Storage

**Database:** PostgreSQL 15+  
**Cache:** Redis 7+  
**File Storage:**

- Development: Local filesystem
- Production: AWS S3 / CloudFlare R2

**CDN:** CloudFlare

## Video Processing Stack

### Libraries:

- FFmpeg (монтаж, конвертация)
- MoviePy (Python wrapper для FFmpeg)
- Pillow (обработка изображений)
- OpenCV (advanced video effects)

### Formats:

- Input: MP4, MOV, AVI, PNG, JPG
- Output: MP4 (H.264 + AAC)
- Resolution: 1080x1920 (9:16 для Shorts)

## Frontend (будущая реализация)

Framework: React 18+ или Vue 3

UI Library: Tailwind CSS + Shadcn/ui

Video Player: Video.js

Timeline Editor: fabric.js или Remotion

State Management: Redux Toolkit / Pinia

API Client: Axios + React Query

---

## БАЗА ДАННЫХ

### Новые таблицы (дополнение к существующим)

```
-- =====
-- VOICE MANAGEMENT
-- =====

CREATE TABLE user_voices (
    id SERIAL PRIMARY KEY,
    user_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,

    -- Voice Provider Details
    voice_id VARCHAR(255) NOT NULL UNIQUE,    -- ID в ElevenLabs/PlayHT
    provider VARCHAR(50) DEFAULT 'elevenlabs',
    voice_name VARCHAR(255) NOT NULL,

    -- Voice Samples
    sample_files JSONB NOT NULL,    -- ["s3://bucket/voice1.mp3", ...]
    sample_duration INTEGER,        -- Общая длительность в секундах
    sample_quality_score FLOAT,     -- 0-1, качество сэмплов

    -- Voice Characteristics
    language VARCHAR(10) DEFAULT 'ru',
    gender VARCHAR(20),
```

```

age_range VARCHAR(50),
accent VARCHAR(50),
tone VARCHAR(50), -- professional, friendly, energetic

-- Clone Status
clone_status VARCHAR(50) DEFAULT 'pending', -- pending, processing, ready, failed
clone_started_at TIMESTAMP,
clone_completed_at TIMESTAMP,
clone_error TEXT,

-- Usage Settings
is_active BOOLEAN DEFAULT TRUE,
is_default BOOLEAN DEFAULT FALSE,
is_public BOOLEAN DEFAULT FALSE, -- Для marketplace

-- Voice Settings (defaults)
default_stability FLOAT DEFAULT 0.5,
default_similarity_boost FLOAT DEFAULT 0.75,
default_style FLOAT DEFAULT 0.5,
default_speed FLOAT DEFAULT 1.0,

-- Stats
characters_generated BIGINT DEFAULT 0,
videos_created INTEGER DEFAULT 0,
last_used_at TIMESTAMP,

-- Billing
cost_per_1000_chars DECIMAL(10,4),
total_cost DECIMAL(10,2) DEFAULT 0,

created_at TIMESTAMP DEFAULT NOW(),
updated_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_user_voices_user_id ON user_voices(user_id);
CREATE INDEX idx_user_voices_status ON user_voices(clone_status);
CREATE INDEX idx_user_voices_active ON user_voices(user_id, is_active);

```

```

-- =====
-- CHARACTER/VISUAL ASSETS MANAGEMENT
-- =====

```

```

CREATE TABLE user_characters (
    id SERIAL PRIMARY KEY,
    user_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    -- Character Identity

```

```

character_name VARCHAR(255) NOT NULL,
character_description TEXT,
character_type VARCHAR(50), -- avatar, person, cartoon, logo

-- Reference Image
reference_image_url TEXT,
reference_image_path TEXT,

-- Generation Settings
generation_provider VARCHAR(50) DEFAULT 'fal_flux', -- fal_flux, leonardo, mic
character_seed INTEGER, -- Для консистентности
ip_adapter_weight FLOAT DEFAULT 0.8,
style_preset VARCHAR(100), -- cinematic, professional, cartoon, etc.

-- Visual Characteristics
gender VARCHAR(20),
age_range VARCHAR(50),
ethnicity VARCHAR(50),
clothing_style VARCHAR(100),
typical_background VARCHAR(100),

-- Variations Library
generated_variations JSONB, -- [{"pose": "standing", "url": "...", "prompt": ""}]
total_variations INTEGER DEFAULT 0,

-- Brand Assets
brand_colors JSONB, -- ["#FF5733", "#3498DB"]
logo_overlay_url TEXT,

-- Usage
is_active BOOLEAN DEFAULT TRUE,
is_default BOOLEAN DEFAULT FALSE,
times_used INTEGER DEFAULT 0,
last_used_at TIMESTAMP,

-- Stats
scenes_created INTEGER DEFAULT 0,
videos_featured INTEGER DEFAULT 0,

created_at TIMESTAMP DEFAULT NOW(),
updated_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_user_characters_user_id ON user_characters(user_id);
CREATE INDEX idx_user_characters_active ON user_characters(user_id, is_active);

-- =====

```

```

-- VIDEO SCRIPTS
-- =====

CREATE TABLE video_scripts (
    id SERIAL PRIMARY KEY,
    user_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,

    -- Script Metadata
    script_title VARCHAR(500) NOT NULL,
    topic TEXT NOT NULL,
    target_duration INTEGER DEFAULT 30, -- seconds

    -- Script Content
    hook_text TEXT,
    body_text TEXT,
    cta_text TEXT,
    full_script TEXT NOT NULL,

    -- Scenes Breakdown
    scenes JSONB NOT NULL, -- [{"id":1, "timing":"0-3", "text":"..."}, {"visual_promo": true}], total_scenes INTEGER,

    -- Style & Tone
    script_style VARCHAR(50), -- educational, motivational, entertaining, selling
    tone VARCHAR(50), -- professional, casual, friendly, authoritative

    -- SEO & Publishing
    suggested_title VARCHAR(500),
    description TEXT,
    hashtags JSONB, -- ["#business", "#entrepreneur"]
    keywords JSONB,

    -- Generation Info
    ai_model VARCHAR(100), -- gpt-4, claude-3.5-sonnet
    generation_prompt TEXT,

    -- Usage
    is_template BOOLEAN DEFAULT FALSE,
    times_used INTEGER DEFAULT 0,

    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_video_scripts_user_id ON video_scripts(user_id);
-- =====

```

```

-- VIDEO PROJECTS (SHORTS)
-- =====

CREATE TABLE video_projects (
    id SERIAL PRIMARY KEY,
    user_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,

    -- Project Metadata
    project_name VARCHAR(255) NOT NULL,
    project_type VARCHAR(50) DEFAULT 'short', -- short, reel, video
    status VARCHAR(50) DEFAULT 'draft', -- draft, processing, ready, published, fa

    -- Associated Assets
    script_id INTEGER REFERENCES video_scripts(id),
    voice_id INTEGER REFERENCES user_VOICES(id),
    character_id INTEGER REFERENCES user_characters(id),

    -- Content
    title VARCHAR(500),
    description TEXT,
    duration INTEGER, -- seconds

    -- Files
    video_path TEXT,
    thumbnail_path TEXT,
    subtitle_path TEXT, -- SRT file

    -- Video Specs
    resolution VARCHAR(20) DEFAULT '1080x1920', -- 9:16
    format VARCHAR(10) DEFAULT 'mp4',
    fps INTEGER DEFAULT 30,
    file_size BIGINT, -- bytes

    -- Production Pipeline
    pipeline_stage VARCHAR(50), -- script, voiceover, visuals, editing, rendering
    pipeline_progress INTEGER DEFAULT 0, -- 0-100
    pipeline_started_at TIMESTAMP,
    pipeline_completed_at TIMESTAMP,
    pipeline_error TEXT,

    -- Production Details
    scenes_data JSONB, -- Detailed scene information
    voiceover_url TEXT,
    background_music_url TEXT,

    -- Branding
    show_logo BOOLEAN DEFAULT TRUE,
    show_intro BOOLEAN DEFAULT FALSE,

```

```

show_outro BOOLEAN DEFAULT FALSE,
-- Publishing
platforms JSONB, -- ["youtube", "instagram", "tiktok"]
publish_status JSONB, -- {"youtube": "published", "instagram": "pending"}
published_urls JSONB, -- {"youtube": "https://...", ...}

-- Scheduling
scheduled_for TIMESTAMP,
published_at TIMESTAMP,

-- Analytics
total_views INTEGER DEFAULT 0,
total_likes INTEGER DEFAULT 0,
total_shares INTEGER DEFAULT 0,
engagement_rate FLOAT,

-- Cost Tracking
production_cost DECIMAL(10, 4),

created_at TIMESTAMP DEFAULT NOW(),
updated_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_video_projects_user_id ON video_projects(user_id);
CREATE INDEX idx_video_projects_status ON video_projects(status);
CREATE INDEX idx_video_projects_published ON video_projects(published_at);

-- =====
-- BATCH PRODUCTION JOBS
-- =====

CREATE TABLE batch_production_jobs (
    id SERIAL PRIMARY KEY,
    user_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    -- Job Info
    job_name VARCHAR(255),
    total_videos INTEGER NOT NULL,
    -- Input
    topics JSONB NOT NULL, -- ["Topic 1", "Topic 2", ...]
    batch_settings JSONB, -- Common settings for all videos
    -- Progress
    status VARCHAR(50) DEFAULT 'pending', -- pending, processing, completed, failed
    completed_videos INTEGER DEFAULT 0,

```

```

failed_videos INTEGER DEFAULT 0,

-- Timing
started_at TIMESTAMP,
completed_at TIMESTAMP,
estimated_completion TIMESTAMP,

-- Results
created_project_ids JSONB, -- [123, 124, 125, ...]

-- Cost
total_cost DECIMAL(10,2),

created_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_batch_jobs_user_id ON batch_production_jobs(user_id);
CREATE INDEX idx_batch_jobs_status ON batch_production_jobs(status);

-- =====
-- YOUTUBE ACCOUNTS (для публикации)
-- =====

CREATE TABLE youtube_accounts (
    id SERIAL PRIMARY KEY,
    user_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,

    -- OAuth Tokens (encrypted)
    encrypted_access_token TEXT NOT NULL,
    encrypted_refresh_token TEXT NOT NULL,
    token_expires_at TIMESTAMP,

    -- YouTube Channel Info
    channel_id VARCHAR(255) NOT NULL,
    channel_title VARCHAR(255),
    channel_username VARCHAR(255),
    channel_thumbnail_url TEXT,
    subscribers_count INTEGER,

    -- Account Name (for UI)
    account_name VARCHAR(255) NOT NULL,

    -- Status
    is_active BOOLEAN DEFAULT TRUE,
    is_default BOOLEAN DEFAULT FALSE,
    is_verified BOOLEAN DEFAULT FALSE,

```

```

-- Stats
shorts_uploaded INTEGER DEFAULT 0,
total_views INTEGER DEFAULT 0,
last_upload_at TIMESTAMP,
last_error TEXT,

created_at TIMESTAMP DEFAULT NOW(),
updated_at TIMESTAMP DEFAULT NOW(),

CONSTRAINT unique_user_youtube_channel UNIQUE (user_id, channel_id)
);

CREATE INDEX idx_youtube_accounts_user_id ON youtube_accounts(user_id);

-- =====
-- TIKTOK ACCOUNTS (для публикации)
-- =====

CREATE TABLE tiktok_accounts (
    id SERIAL PRIMARY KEY,
    user_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,

    -- OAuth Tokens (encrypted)
    encrypted_access_token TEXT NOT NULL,
    encrypted_refresh_token TEXT NOT NULL,
    token_expires_at TIMESTAMP,

    -- TikTok Account Info
    tiktok_user_id VARCHAR(255) NOT NULL,
    tiktok_username VARCHAR(255),
    tiktok_display_name VARCHAR(255),
    avatar_url TEXT,
    followers_count INTEGER,

    -- Account Name (for UI)
    account_name VARCHAR(255) NOT NULL,

    -- Status
    is_active BOOLEAN DEFAULT TRUE,
    is_default BOOLEAN DEFAULT FALSE,
    is_verified BOOLEAN DEFAULT FALSE,

    -- Stats
    videos_uploaded INTEGER DEFAULT 0,
    total_views INTEGER DEFAULT 0,
    last_upload_at TIMESTAMP,
    last_error TEXT,

```

```
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW(),

    CONSTRAINT unique_user_tiktok UNIQUE (user_id, tiktok_user_id)
);

CREATE INDEX idx_tiktok_accounts_user_id ON tiktok_accounts(user_id);
```

---

## □ МОДУЛИ И КОМПОНЕНТЫ

### MODULE 1: Voice Studio Agent

Файл: app/agents/voice\_studio\_agent.py

```
class VoiceStudioAgent(BaseAgent):
    """
    Управление голосовыми активами пользователя
    """

    # ФУНКЦИИ:
    - clone_voice(user_id, audio_files, voice_name)
    - get_user_VOICES(user_id)
    - test_voice(voice_id, test_text)
    - update_voice_settings(voice_id, settings)
    - delete_voice(voice_id)
    - generate_speech(voice_id, text, settings)
```

### MODULE 2: Character Studio Agent

Файл: app/agents/character\_studio\_agent.py

```
class CharacterStudioAgent(BaseAgent):
    """
    Создание и управление AI персонажами
    """

    # ФУНКЦИИ:
    - create_character(user_id, reference_image, description)
    - generate_character_variations(character_id, poses, emotions)
    - get_user_characters(user_id)
    - update_character(character_id, settings)
    - delete_character(character_id)
```

### MODULE 3: Script Generator Agent

Файл: app/agents/script\_generator\_agent.py

```
class ScriptGeneratorAgent(BaseAgent):
    """
    Генерация сценариев для видео
    """

    # Функции:
    - generate_script(topic, duration, style, tone)
    - generate_scenes_breakdown(script)
    - optimize_for_platform(script, platform)
    - generate_metadata(script) # title, description, hashtags
```

#### MODULE 4: Visual Producer Agent

Файл: app/agents/visual\_producer\_agent.py

```
class VisualProducerAgent(BaseAgent):
    """
    Генерация визуального ряда
    """

    # Функции:
    - generate_scene_image(scene, character_id, style)
    - generate_thumbnail(title, character_id, style)
    - batch_generate_scenes(scenes_list, character_id)
    - apply_branding(image, logo, colors)
```

#### MODULE 5: Video Editor Agent

Файл: app/agents/video\_editor\_agent.py

```
class VideoEditorAgent(BaseAgent):
    """
    Монтаж видео из компонентов
    """

    # Функции:
    - create_video_from_scenes(scenes, voiceover, music)
    - add_subtitles(video_path, subtitle_text)
    - add_branding(video_path, logo, intro, outro)
    - apply_transitions(scenes, transition_type)
    - render_final_video(project_id)
```

#### MODULE 6: Multi-Platform Publisher Agent

Файл: app/agents/multi\_platform\_publisher\_agent.py

```
class MultiPlatformPublisherAgent(BaseAgent):
    """
    Публикация на все платформы
    """
```

```
"""
```

```
# ФУНКЦИИ:  
- publish_to_youtube(video_path, metadata, account_id)  
- publish_to_instagram(video_path, metadata, account_id)  
- publish_to_tiktok(video_path, metadata, account_id)  
- publish_to_twitter(video_path, metadata, account_id)  
- schedule_publication(video_id, platforms, datetime)
```

---

## API ENDPOINTS

### Voice Studio API

```
POST /api/voice/clone
```

- Клонирование голоса из аудио файлов
- Body: multipart/form-data (audio files)
- Returns: voice\_id, status

```
GET /api/voice/my-voices
```

- Список голосов пользователя
- Returns: [{voice\_id, name, status, stats}, ...]

```
POST /api/voice/{voice_id}/test
```

- Тестирование голоса с текстом
- Body: {text: string}
- Returns: audio\_url

```
PUT /api/voice/{voice_id}/settings
```

- Обновление настроек голоса
- Body: {stability, similarity\_boost, speed, ...}

```
DELETE /api/voice/{voice_id}
```

- Удаление голоса

```
POST /api/voice/{voice_id}/generate
```

- Генерация речи голосом
- Body: {text: string, settings: {...}}
- Returns: audio\_url

### Character Studio API

```
POST /api/character/create
```

- Создание персонажа
- Body: multipart (reference\_image) + JSON (description)
- Returns: character\_id, status

```
GET      /api/character/my-characters
- Список персонажей пользователя

POST     /api/character/{id}/generate-variations
- Генерация вариаций (позы, эмоции, фоны)
- Body: {poses: [...], emotions: [...], backgrounds: [...]}
- Returns: variation_urls

PUT      /api/character/{id}/update
- Обновление персонажа

DELETE   /api/character/{id}
- Удаление персонажа
```

## Video Production API

```
POST    /api/video/create-short
- Создание Short в один клик
- Body: {
  topic: string,
  duration: number,
  voice_id: number,
  character_id: number,
  style: string,
  music: string
}
- Returns: project_id, status

GET    /api/video/projects
- Список проектов пользователя

GET    /api/video/project/{id}
- Детали проекта

GET    /api/video/project/{id}/status
- Статус создания (для polling)
- Returns: {
  stage: string,
  progress: number,
  estimated_time: number
}

POST   /api/video/batch-create
- Массовое создание роликов
- Body: {
  topics: [string, ...],
  settings: {...}
}
```

- **Returns:** batch\_job\_id

GET /api/video/batch/{job\_id}/status  
- Статус батч-задания

## Publishing API

POST /api/publish/video/{project\_id}  
- Публикация видео на платформы  
- **Body:** {  
  platforms: ["youtube", "instagram", "tiktok"],  
  schedule\_time: datetime (optional),  
  metadata\_overrides: {...}  
}  
- **Returns:** publication\_id

GET /api/publish/status/{publication\_id}  
- Статус публикации

GET /api/publish/youtube/accounts  
- Список подключенных YouTube каналов

POST /api/publish/youtube/connect  
- OAuth для подключения YouTube

# Аналогично для Instagram, TikTok, Twitter

## Analytics API

GET /api/analytics/overview  
- Общая аналитика пользователя  
- **Returns:** {  
  total\_videos: number,  
  total\_views: number,  
  engagement\_rate: number,  
  best\_performing: [...]  
}

GET /api/analytics/video/{project\_id}  
- Аналитика конкретного видео

GET /api/analytics/platform/{platform}  
- Аналитика по платформе

---

## □ ПЛАН РЕАЛИЗАЦИИ (ЭТАПЫ)

### PHASE 1: Foundation (2-3 недели)

**Цель:** Базовая инфраструктура

**Задачи:** - [ ] Создать БД таблицы (user\_voices, user\_characters, video\_projects) - [ ] Настроить S3/R2 для хранения файлов - [ ] Настроить Celery + Redis для очередей - [ ] Базовые models и migrations - [ ] Базовая структура API endpoints

**Deliverables:** - □ Работающая БД - □ File upload/storage - □ Task queue система

---

### PHASE 2: Voice Studio (2 недели)

**Цель:** Клонирование и управление голосами

**Задачи:** - [ ] Интеграция ElevenLabs API - [ ] VoiceStudioAgent implementation - [ ] API endpoints для voice management - [ ] Загрузка аудио файлов - [ ] Клонирование голоса - [ ] Тестирование голоса - [ ] Генерация speech

**Deliverables:** - □ Пользователь может клонировать голос - □ Пользователь может тестировать голос - □ Генерация речи работает

**Тестирование:**

```
# Test case
user uploads 3 audio files (30 sec each)
→ System clones voice in 2-3 minutes
→ User can generate speech with cloned voice
→ Quality is 9/10 or higher
```

---

### PHASE 3: Character Studio (2 недели)

**Цель:** Создание консистентных персонажей

**Задачи:** - [ ] Интеграция Fal.ai FLUX API - [ ] CharacterStudioAgent implementation - [ ] API endpoints для character management - [ ] Загрузка reference images - [ ] Генерация базового персонажа - [ ] Генерация вариаций (позы, эмоции) - [ ] IP-Adapter для консистентности

**Deliverables:** - □ Создание персонажа из референса - □ Консистентность на всех генерациях - □ Библиотека вариаций

**Тестирование:**

```
# Test case
user uploads reference photo
→ System creates character (30 sec)
→ System generates 10 variations (different poses)
→ All variations look like same person (95%+ similarity)
```

---

## PHASE 4: Script Generator (1 неделя)

**Цель:** AI генерация сценариев

**Задачи:** - [ ] ScriptGeneratorAgent implementation - [ ] Интеграция OpenAI GPT-4 - [ ] Prompt engineering для сценариев - [ ] Генерация hook/body/CTA - [ ] Разбивка на сцены - [ ] Генерация метаданных (title, description, hashtags)

**Deliverables:** - [ ] Генерация сценариев по теме - [ ] Автоматическая разбивка на сцены - [ ] SEO оптимизация

**Тестирование:**

```
# Test case
topic = "5 ошибок начинающих предпринимателей"
→ Script generated in 10 seconds
→ Has hook, 3-5 points, CTA
→ Duration = 30 seconds ±3 sec
→ Includes visual prompts for each scene
```

---

## PHASE 5: Visual Producer (2 недели)

**Цель:** Генерация визуального ряда

**Задачи:** - [ ] VisualProducerAgent implementation - [ ] Генерация изображений для сцен - [ ] Использование user character для консистентности - [ ] Генерация thumbnails - [ ] Batch generation (параллельная обработка) - [ ] Брендирование (логотип, цвета)

**Deliverables:** - [ ] Генерация сцен с user character - [ ] Генерация обложек - [ ] Быстрая параллельная обработка

**Тестирование:**

```
# Test case
script with 4 scenes + thumbnail
→ All 5 images generated in 30-60 seconds (parallel)
→ Character consistent across all scenes
→ High quality (1080x1920)
```

---

## PHASE 6: Video Editor (3 недели)

**Цель:** Монтаж финального видео

**Задачи:** - [ ] VideoEditorAgent implementation - [ ] FFmpeg pipeline setup - [ ] MoviePy integration - [ ] Scene assembly (images → video) - [ ] Voice overlay - [ ] Background music mixing - [ ] Transitions & effects - [ ] Subtitle generation (Whisper API) - [ ] Subtitle overlay - [ ] Branding (logo, intro, outro) - [ ] Final rendering

**Deliverables:** - [ ] Полный монтажный конвейер - [ ] Субтитры - [ ] Брендирование - [ ] Высокое качество экспорта

#### Тестирование:

```
# Test case E2E
topic = "Как увеличить продажи"
→ Full video created in 5 minutes
→ Quality: 1080x1920, 30fps
→ Voiceover matches scenes
→ Subtitles accurate (95%+)
→ Professional look
```

---

### PHASE 7: Multi-Platform Publisher (2 недели)

**Цель:** Публикация на все платформы

**Задачи:** - [ ] YouTube API integration - [ ] Instagram API integration (уже частично есть!) - [ ] TikTok API integration - [ ] Twitter API integration (уже есть!) - [ ] Telegram integration (уже есть!) - [ ] OAuth flows для всех платформ - [ ] Metadata optimization per platform - [ ] Scheduling system - [ ] Retry logic

**Deliverables:** - Публикация на YouTube Shorts - Публикация на Instagram Reels - Публикация на TikTok - Публикация на Twitter/X - Публикация в Telegram - Планирование публикаций

#### Тестирование:

```
# Test case
video ready
→ One click publishes to all 5 platforms
→ Each platform gets optimized metadata
→ Success rate: 95%+
```

---

### PHASE 8: Batch Production (1 неделя)

**Цель:** Массовое создание роликов

**Задачи:** - [ ] Batch job queue system - [ ] Parallel processing - [ ] Progress tracking - [ ] Error handling & retry - [ ] Cost estimation - [ ] Batch publishing

**Deliverables:** - Создание 50+ роликов за раз - Параллельная обработка - Progress dashboard

#### Тестирование:

```
# Test case
50 topics
→ All 50 videos created in 2-3 hours (parallel)
→ Success rate: 90%+
→ Total cost: $12.50 ($0.25 each)
```

---

## PHASE 9: Analytics & Dashboard (1 неделя)

**Цель:** Аналитика и insights

**Задачи:** - [ ] Сбор метрик с платформ - [ ] Analytics API endpoints - [ ] Dashboard визуализация  
- [ ] Performance insights - [ ] Recommendations engine

**Deliverables:** - ☐ Аналитика по всем платформам - ☐ Best performing content - ☐ AI рекомендации

---

## PHASE 10: Frontend (4-6 недель)

**Параллельно с backend разработкой**

**Задачи:** - [ ] Voice Studio UI - [ ] Character Studio UI - [ ] One-Click Creator UI - [ ] Timeline Editor UI - [ ] Publishing Dashboard - [ ] Analytics Dashboard - [ ] Settings & Billing

---

## PHASE 11: Polish & Optimization (2 недели)

**Задачи:** - [ ] Performance optimization - [ ] Cost optimization - [ ] Error handling improvements - [ ] UI/UX polish - [ ] Documentation - [ ] Video tutorials

---

## ☐ ИНТЕГРАЦИИ И ЗАВИСИМОСТИ

### External APIs Required

#### TIER 1 (Critical):

- ElevenLabs API: Voice cloning & TTS
- Fal.ai API: Image generation (FLUX)
- OpenAI API: Scripts (GPT-4), Subtitles (Whisper)
- YouTube Data API v3: Publishing

#### TIER 2 (Important):

- Instagram Graph API: Publishing Reels
- TikTok Content Posting API: Publishing
- Twitter API v2: Publishing videos

#### TIER 3 (Nice to have):

- Epidemic Sound API: Music library
- Pexels/Unsplash API: Stock footage
- AssemblyAI: Subtitle alternative

## **API Costs (Monthly estimates for 1000 videos)**

### **Voice (ElevenLabs):**

- Cloning: FREE (3 voices)
- Generation: \$150 (500k characters)

### **Images (Fal.ai FLUX):**

- 5000 images (5 per video): \$25

### **Scripts (OpenAI GPT-4):**

- 1000 scripts: \$10

### **Subtitles (Whisper):**

- 1000 videos (30 sec each): \$15

**YouTube API:** FREE

**Instagram API:** FREE

**TikTok API:** FREE

**Twitter API:** FREE

**TOTAL:** ~\$200/month for 1000 videos

= \$0.20 per video

## **Infrastructure Costs**

### **Database (PostgreSQL):**

- Development: FREE (Railway/Supabase)
- Production: \$50/month (Managed)

### **Storage (S3/R2):**

- 500GB storage: \$10/month
- Bandwidth: \$20/month

### **Server:**

- Development: Local
- Production: \$100/month (4 CPU, 16GB RAM)

### **CDN:**

- CloudFlare: FREE or \$20/month

### **Redis:**

- Development: FREE
- Production: \$30/month

**TOTAL:** ~\$230/month infrastructure

---

## □ СТОИМОСТЬ И РЕСУРСЫ

### Development Resources

#### Team Needed:

- Backend Developer (Python/Flask): 1 FTE
- Frontend Developer (React/Vue): 1 FTE
- DevOps Engineer: 0.5 FTE
- UI/UX Designer: 0.5 FTE

Timeline: 12-16 weeks

#### Budget Estimate:

- Development: \$40,000 - \$60,000
- Infrastructure (3 months): \$700
- APIs testing: \$500
- Total: \$41,200 - \$61,200

### Pricing Model (для пользователей)

#### FREE TIER:

- 5 videos/month
- 1 voice clone
- 1 character
- Watermark on videos
- \$0/month

#### CREATOR (\$29/month):

- 50 videos/month
- 3 voice clones
- 3 characters
- No watermark
- All platforms
- Basic analytics

#### PRO (\$99/month):

- 200 videos/month
- 10 voice clones
- 10 characters
- Priority processing
- Advanced analytics
- Batch production
- API access

#### AGENCY (\$299/month):

- Unlimited videos
- Unlimited voices
- Unlimited characters

- White-label
- Team collaboration
- Dedicated support

## Unit Economics

At \$99/month (PRO plan):

- Revenue: \$99
- Costs (200 videos × \$0.20): \$40
- Infrastructure: \$5
- Gross Margin: \$54 (54.5%)

Break-even: ~80 paying users

Healthy: 200+ paying users

---

## МЕТРИКИ УСПЕХА

### Technical Metrics

Performance:

- Video creation time: < 5 minutes per Short
- Batch processing: 50 videos in < 3 hours
- API response time: < 200ms (p95)
- Uptime: 99.9%

Quality:

- Voice clone quality: > 8/10 (user rating)
- Character consistency: > 95% similarity
- Subtitle accuracy: > 95%
- Publishing success rate: > 95%

Cost:

- Production cost per video: < \$0.25
- Gross margin: > 50%

### Business Metrics

Month 1-3 (Beta):

- 100 beta users
- 50 paying users
- 1000 videos created
- \$2,500 MRR

Month 4-6 (Growth):

- 500 total users
- 200 paying users

- 10,000 videos created
- \$12,000 MRR

#### Month 7-12 (Scale):

- 2000 total users
  - 800 paying users
  - 50,000 videos created
  - \$50,000 MRR
- 

## NEXT STEPS

### Immediate Actions

1. **Review this roadmap** и уточнить приоритеты
2. **Получить API ключи** для тестирования:
  - ElevenLabs API key
  - Fal.ai API key
  - OpenAI API key
3. **Начать с PHASE 1 + PHASE 2** (Foundation + Voice Studio)
4. **Настроить окружение** и базовую инфраструктуру

### Priority Order

1. Foundation (БД, storage, queue) [WEEK 1-2]
  2. Voice Studio (клонирование голоса) [WEEK 3-4]
  3. Script Generator (AI сценарии) [WEEK 5]
  4. Character Studio (AI персонажи) [WEEK 6-7]
  5. Visual Producer (генерация сцен) [WEEK 8-9]
  6. Video Editor (монтаж) [WEEK 10-12]
  7. Publisher (multi-platform) [WEEK 13-14]
  8. Batch Production [WEEK 15]
  9. Frontend [PARALLEL]
  10. Polish & Launch [WEEK 16]
- 

## NOTES

- Этот документ - живой roadmap, будет обновляться
  - Каждая фаза будет иметь свой detailed spec
  - Тестирование критично на каждом этапе
  - MVP можно запустить после Phase 6 (без batch и некоторых платформ)
  - Некоторые интеграции уже готовы (Instagram, Twitter, Telegram)
-

## **□ СВЯЗАННЫЕ ДОКУМЕНТЫ**

- SOCIAL\_MEDIA\_SETUP.md - Настройка соцсетей (Instagram, Twitter, Telegram)
  - TELEGRAM\_CHANNELS\_SETUP.md - Telegram интеграция
  - INSTAGRAM\_TWITTER\_READY.md - Instagram & Twitter готовность
- 

**Статус:** □ Roadmap готов к реализации

**Дата обновления:** 20 октября 2025

**Следующий шаг:** Получить одобрение и начать Phase 1

**□ Ready to build the future of content creation!**