🚀 Video AI Backend - Complete Setup Guide

📁 Project Structure

Create this exact folder structure on your machine:

```
video-ai-backend/
├── app/
│   ├── __init__.py                 # Empty file
│   ├── main.py                     # FastAPI app entry point
│   ├── core/
│   │   ├── __init__.py             # Empty file
│   │   ├── config.py               # Settings and configuration
│   │   └── database.py             # Database connection
│   ├── models/
│   │   ├── __init__.py             # Empty file
│   │   └── video.py                # Database models
│   ├── schemas/
│   │   ├── __init__.py             # Empty file
│   │   └── video.py                # Pydantic schemas
│   ├── api/
│   │   ├── __init__.py             # Empty file
│   │   ├── dependencies.py         # FastAPI dependencies (create empty)
│   │   └── v1/
│   │       ├── __init__.py         # Empty file
│   │       ├── api.py              # Main API router
│   │       └── endpoints/
│   │           ├── __init__.py     # Empty file
│   │           ├── videos.py       # Video endpoints
│   │           ├── revisions.py    # Revision endpoints
│   │           └── webhooks.py     # Webhook endpoints
│   ├── services/
│   │   ├── __init__.py             # Empty file
│   │   ├── base44_integration.py   # Base44 callbacks
│   │   ├── ai_clients/
│   │   │   ├── __init__.py         # Empty file
│   │   │   ├── fal_client.py       # FAL AI client
│   │   │   ├── elevenlabs_client.py # ElevenLabs client
│   │   │   ├── openai_client.py    # OpenAI client
│   │   │   └── lyria_client.py     # Lyria client
│   │   └── video_generation/
│   │       ├── __init__.py         # Empty file
│   │       ├── main_pipeline.py    # Main video pipeline
│   │       └── revision_pipeline.py # Revision pipeline
│   ├── tasks/
│   │   ├── __init__.py             # Empty file
│   │   ├── celery_app.py           # Celery configuration
│   │   ├── video_tasks.py          # Video generation tasks
│   │   └── revision_tasks.py       # Revision tasks
│   └── utils/
│       ├── __init__.py             # Empty file
│       ├── errors.py               # Custom exceptions
```

```
│   └── logging.py              # Logging setup
├── requirements.txt            # Python dependencies
├── docker-compose.yml          # Docker services
├── Dockerfile                  # Docker image
├── .env.example                # Environment template
├── .gitignore                  # Git ignore file
├── run.sh                      # Startup script
├── migrate_from_n8n.py         # Migration script
└── README.md                   # Documentation
```

🛠️ Step-by-Step Setup

Step 1: Create Project Directory

bash

```bash
mkdir video-ai-backend
cd video-ai-backend
```

Step 2: Create All Directories

bash

```bash
# Create directory structure
mkdir -p app/{core,models,schemas,api/v1/endpoints,services/{ai_clients,video_generation},tasks,utils}

# Create empty __init__.py files
find app -type d -exec touch {}/__init__.py \;

# Create empty dependencies file
touch app/api/dependencies.py
```

Step 3: Copy All Files

Copy each file from the artifacts I created above into the corresponding location:

1. **Root Files:**
   - `requirements.txt` → `/requirements.txt`
   - `docker-compose.yml` → `/docker-compose.yml`
   - `Dockerfile` → `/Dockerfile`
   - `.env.example` → `/.env.example`
   - `run.sh` → `/run.sh`
   - `migrate_from_n8n.py` → `/migrate_from_n8n.py`

- README.md → /README.md

2. **App Core Files:**
   - app/core/config.py → /app/core/config.py
   - app/core/database.py → /app/core/database.py
   - app/main.py → /app/main.py
3. **Models & Schemas:**
   - app/models/video.py → /app/models/video.py
   - app/schemas/video.py → /app/schemas/video.py
4. **API Endpoints:**
   - app/api/v1/api.py → /app/api/v1/api.py
   - app/api/v1/endpoints/videos.py → /app/api/v1/endpoints/videos.py
   - app/api/v1/endpoints/revisions.py → /app/api/v1/endpoints/revisions.py
   - app/api/v1/endpoints/webhooks.py → /app/api/v1/endpoints/webhooks.py
5. **AI Clients:**
   - app/services/ai_clients/fal_client.py → /app/services/ai_clients/fal_client.py
   - app/services/ai_clients/elevenlabs_client.py → /app/services/ai_clients/elevenlabs_client.py
   - app/services/ai_clients/openai_client.py → /app/services/ai_clients/openai_client.py
   - app/services/ai_clients/lyria_client.py → /app/services/ai_clients/lyria_client.py
6. **Video Generation:**
   - app/services/video_generation/main_pipeline.py → /app/services/video_generation/main_pipeline.py
   - app/services/video_generation/revision_pipeline.py → /app/services/video_generation/revision_pipeline.py
   - app/services/base44_integration.py → /app/services/base44_integration.py
7. **Tasks:**
   - app/tasks/celery_app.py → /app/tasks/celery_app.py
   - app/tasks/video_tasks.py → /app/tasks/video_tasks.py
   - app/tasks/revision_tasks.py → /app/tasks/revision_tasks.py
8. **Utils:**
   - app/utils/errors.py → /app/utils/errors.py
   - app/utils/logging.py → /app/utils/logging.py

Step 4: Set Permissions

```bash
chmod +x run.sh
chmod +x migrate_from_n8n.py
```

Step 5: Configure Environment

```bash
# Copy environment template
cp .env.example .env

# Edit the .env file with your actual values
nano .env  # or vim .env or code .env
```

Required environment variables to update in `.env`:

```env
# Your actual API keys
FAL_API_KEY=Key 225b91e5-24b6-48eb-8687-3cb9a239805b:e1047a86c96ac36163a602b47289c707
OPENAI_API_KEY=sk-your-actual-openai-key-here
ELEVENLABS_API_KEY=your-actual-elevenlabs-key-here

# Your Base44 callback URL
BASE44_CALLBACK_URL=https://base44.app/api/apps/68b4aa46f5d6326ab93c3ed0/functions/n8nVideoCallb

# Change this secret key
SECRET_KEY=your-super-secret-production-key-here-make-it-random-and-long
```

Step 6: Start the System

```bash
./run.sh
```

This will:

- Check Docker is running
- Validate environment variables
- Build containers
- Start all services
- Show you the URLs to access

🔗 Access Points

After running `./run.sh`, you can access:

- **API Documentation**: http://localhost:8000/docs
- **Health Check**: http://localhost:8000/health
- **Celery Monitor**: http://localhost:5555
- **Database**: localhost:5432 (postgres/password)
- **Redis**: localhost:6379

🧪 Test Your Setup

1. Health Check

```bash
curl http://localhost:8000/health
```

2. Test Video Generation

```bash
curl -X POST http://localhost:8000/api/v1/webhooks/video-generation \
  -H "Content-Type: application/json" \
  -d '{
    "prompt": "Create a product advertisement video for a luxury skincare product",
    "image_url": "https://example.com/product-image.jpg",
    "user_id": "test-user-123",
    "chat_id": "test-chat-456"
  }'
```

3. Check Video Status

```bash
# Replace VIDEO_ID with the ID from the previous response
curl http://localhost:8000/api/v1/videos/{VIDEO_ID}/status
```

🔄 Migrate from n8n (Optional)

If you have existing data in Supabase:

1. **Update migration script**:

```bash
```

```
# Edit migrate_from_n8n.py
# Update SUPABASE_URL with your actual Supabase database URL
nano migrate_from_n8n.py
```

2. **Run migration:**

```
bash

python migrate_from_n8n.py
```

🔗 Update Base44 Webhooks

In your Base44 functions, update webhook URLs from:

```
OLD: https://n8n-instance.com/webhook/5d109062-fa9e-406d-b43a-b5df3b385c0c
NEW: https://your-domain.com/api/v1/webhooks/video-generation

OLD: https://n8n-instance.com/webhook/1949f7a3-6527-473c-9234-d325606233ee
NEW: https://your-domain.com/api/v1/webhooks/revision
```

🪲 Troubleshooting

Common Issues:

1. **"Docker not running"**
   - Start Docker Desktop
   - Run `docker --version` to verify
2. **"Permission denied: ./run.sh"**
   - Run `chmod +x run.sh`
3. **"Missing environment variables"**
   - Check your `.env` file has all required keys
   - No quotes around values in .env
4. **API key errors**
   - Verify your FAL_API_KEY format
   - Test OpenAI key with a simple API call
5. **Port conflicts**
   - Change ports in docker-compose.yml if 8000, 5432, or 6379 are in use

Get Logs:

```
bash
```

```
# API logs
docker-compose logs -f api

# Celery worker logs
docker-compose logs -f celery-worker

# All logs
docker-compose logs -f
```

Stop Services:

```
bash

docker-compose down
```

Restart Services:

```
bash

docker-compose down && docker-compose up -d
```

🎯 Next Steps

1. **Test thoroughly** with your actual video prompts
2. **Update Base44 webhooks** to point to new endpoints
3. **Monitor performance** in production
4. **Scale workers** if needed: `docker-compose up --scale celery-worker=4`
5. **Set up monitoring** with Sentry (optional)

📞 Support

If you run into issues:

1. Check the logs first: `docker-compose logs -f`
2. Verify environment variables in `.env`
3. Test individual components
4. Check network connectivity to AI services

🎉 **You now have a complete, production-ready video AI backend that replaces your n8n workflows!**