<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Anime News Creator Dashboard</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

min-height: 100vh;

color: #333;

}

.container {

max-width: 1400px;

margin: 0 auto;

padding: 20px;

}

.header {

text-align: center;

color: white;

margin-bottom: 30px;

}

.header h1 {

font-size: 2.5em;

margin-bottom: 10px;

text-shadow: 2px 2px 4px rgba(0,0,0,0.3);

}

.dashboard {

display: grid;

grid-template-columns: 1fr 1fr;

gap: 20px;

margin-bottom: 30px;

}

.panel {

background: rgba(255, 255, 255, 0.95);

border-radius: 15px;

padding: 25px;

box-shadow: 0 8px 32px rgba(0,0,0,0.1);

backdrop-filter: blur(10px);

}

.panel h2 {

color: #4a5568;

margin-bottom: 20px;

font-size: 1.4em;

border-bottom: 2px solid #e2e8f0;

padding-bottom: 10px;

}

.news-item {

background: #f8f9fa;

border-left: 4px solid #667eea;

padding: 15px;

margin-bottom: 15px;

border-radius: 8px;

cursor: pointer;

transition: all 0.3s ease;

}

.news-item:hover {

transform: translateY(-2px);

box-shadow: 0 4px 15px rgba(0,0,0,0.1);

}

.news-item.selected {

background: #e6f3ff;

border-left-color: #3182ce;

}

.news-title {

font-weight: bold;

color: #2d3748;

margin-bottom: 8px;

}

.news-meta {

font-size: 0.85em;

color: #718096;

margin-bottom: 8px;

}

.news-summary {

color: #4a5568;

line-height: 1.4;

}

.content-creator {

grid-column: 1 / -1;

}

.template-selector {

display: flex;

gap: 10px;

margin-bottom: 20px;

flex-wrap: wrap;

}

.template-btn {

padding: 10px 20px;

border: 2px solid #e2e8f0;

background: white;

border-radius: 25px;

cursor: pointer;

transition: all 0.3s ease;

font-weight: 500;

}

.template-btn:hover, .template-btn.active {

background: #667eea;

color: white;

border-color: #667eea;

}

.content-preview {

background: #1a202c;

color: #e2e8f0;

padding: 20px;

border-radius: 10px;

font-family: 'Courier New', monospace;

margin-bottom: 20px;

min-height: 200px;

white-space: pre-wrap;

}

.controls {

display: flex;

gap: 15px;

flex-wrap: wrap;

align-items: center;

}

.btn {

padding: 12px 25px;

border: none;

border-radius: 25px;

cursor: pointer;

font-weight: 600;

transition: all 0.3s ease;

font-size: 14px;

}

.btn-primary {

background: linear-gradient(45deg, #667eea, #764ba2);

color: white;

}

.btn-primary:hover {

transform: translateY(-2px);

box-shadow: 0 5px 15px rgba(102, 126, 234, 0.4);

}

.btn-secondary {

background: #e2e8f0;

color: #4a5568;

}

.btn-secondary:hover {

background: #cbd5e0;

}

.input-group {

display: flex;

gap: 10px;

align-items: center;

}

.input-group input, .input-group select {

padding: 8px 12px;

border: 1px solid #e2e8f0;

border-radius: 5px;

font-size: 14px;

}

.stats {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(120px, 1fr));

gap: 15px;

margin-bottom: 20px;

}

.stat-card {

background: linear-gradient(45deg, #4fd1c7, #38b2ac);

color: white;

padding: 15px;

border-radius: 10px;

text-align: center;

}

.stat-number {

font-size: 1.8em;

font-weight: bold;

}

.stat-label {

font-size: 0.9em;

opacity: 0.9;

}

.loading {

display: none;

text-align: center;

color: #667eea;

font-weight: 500;

}

.hashtag-suggestions {

display: flex;

flex-wrap: wrap;

gap: 8px;

margin-top: 10px;

}

.hashtag {

background: #e6f3ff;

color: #3182ce;

padding: 5px 10px;

border-radius: 15px;

font-size: 0.85em;

cursor: pointer;

transition: all 0.2s ease;

}

.hashtag:hover {

background: #3182ce;

color: white;

}

@media (max-width: 768px) {

.dashboard {

grid-template-columns: 1fr;

}

.controls {

flex-direction: column;

align-items: stretch;

}

}

</style>

</head>

<body>

<div class="container">

<div class="header">

<h1>🎌 Anime News Creator</h1>

<p>Semi-automated content creation for TikTok</p>

</div>

<div class="stats">

<div class="stat-card">

<div class="stat-number" id="newsCount">0</div>

<div class="stat-label">News Items</div>

</div>

<div class="stat-card">

<div class="stat-number" id="contentGenerated">0</div>

<div class="stat-label">Content Generated</div>

</div>

<div class="stat-card">

<div class="stat-number" id="trending">5</div>

<div class="stat-label">Trending Topics</div>

</div>

</div>

<div class="dashboard">

<div class="panel">

<h2>📰 Latest Anime News</h2>

<div class="loading" id="newsLoading">Loading latest news...</div>

<div id="newsFeed">

<!-- News items will be populated here -->

</div>

<button class="btn btn-secondary" onclick="refreshNews()" style="margin-top: 15px;">

🔄 Refresh News

</button>

<button class="btn btn-secondary" onclick="addCustomSource()" style="margin-top: 15px; margin-left: 10px;">

➕ Add Source

</button>

</div>

<div class="panel">

<h2>⚙️ Content Settings</h2>

<div class="input-group" style="margin-bottom: 15px;">

<label>Content Style:</label>

<select id="contentStyle">

<option value="breaking">🚨 Breaking News</option>

<option value="casual">💬 Casual Update</option>

<option value="analysis">🔍 Deep Dive</option>

<option value="hype">🔥 Hype Beast</option>

<option value="comparison">⚡ Quick Facts</option>

</select>

</div>

<div class="input-group" style="margin-bottom: 15px;">

<label>Duration:</label>

<select id="duration">

<option value="15">15 seconds</option>

<option value="30" selected>30 seconds</option>

<option value="60">60 seconds</option>

</select>

</div>

<div class="input-group">

<label>Target Audience:</label>

<select id="audience">

<option value="general">General Anime Fans</option>

<option value="hardcore">Hardcore Otaku</option>

<option value="casual">Casual Viewers</option>

<option value="newcomers">Anime Newcomers</option>

</select>

</div>

</div>

</div>

<div class="panel content-creator">

<h2>🎬 Content Creator</h2>

<div class="template-selector">

<div class="template-btn active" onclick="selectTemplate('script')">📝 Script</div>

<div class="template-btn" onclick="selectTemplate('hooks')">🎣 Hooks</div>

<div class="template-btn" onclick="selectTemplate('hashtags')">🏷️ Hashtags</div>

<div class="template-btn" onclick="selectTemplate('thumbnail')">🖼️ Thumbnail Ideas</div>

<div class="template-btn" onclick="selectTemplate('videoSetup')">🎬 Video Setup</div>

<div class="template-btn" onclick="selectTemplate('automation')">🤖 Auto Script</div>

</div>

<div class="content-preview" id="contentPreview">

Select a news item and click "Generate Content" to create TikTok-ready material!

Your generated content will appear here with:

• Engaging script with timestamps

• Hook variations

• Trending hashtags

• Thumbnail concepts

• Call-to-action suggestions</div>

<div class="controls">

<button class="btn btn-primary" onclick="generateContent()">

✨ Generate Content

</button>

<button class="btn btn-primary" onclick="generateVideoPackage()" style="background: linear-gradient(45deg, #ff6b6b, #ee5a24);">

🎥 Generate Video Package

</button>

<button class="btn btn-secondary" onclick="copyContent()">

📋 Copy to Clipboard

</button>

<button class="btn btn-secondary" onclick="downloadVideoAssets()">

📦 Download Assets

</button>

<button class="btn btn-secondary" onclick="saveContent()">

💾 Save Template

</button>

<div class="input-group">

<input type="text" id="customKeywords" placeholder="Add custom keywords...">

<button class="btn btn-secondary" onclick="addKeywords()">Add</button>

</div>

</div>

<div class="hashtag-suggestions" id="hashtagSuggestions">

<!-- Hashtag suggestions will appear here -->

</div>

</div>

</div>

<script>

let selectedNews = null;

let currentTemplate = 'script';

let contentCount = 0;

let savedTemplates = [];

// News sources configuration

const newsSources = [

{

name: "Anime News Network",

rss: "https://www.animenewsnetwork.com/all/rss.xml",

category: "General",

priority: 1

},

{

name: "Crunchyroll News",

rss: "https://www.crunchyroll.com/news/rss",

category: "Streaming",

priority: 2

},

{

name: "MyAnimeList News",

rss: "https://myanimelist.net/rss/news.xml",

category: "General",

priority: 3

}

];

let newsCache = [];

let lastFetchTime = 0;

const CACHE\_DURATION = 30 \* 60 \* 1000; // 30 minutes

function initializeApp() {

loadNews();

updateStats();

generateHashtagSuggestions();

}

async function loadNews() {

const newsFeed = document.getElementById('newsFeed');

const newsLoading = document.getElementById('newsLoading');

newsLoading.style.display = 'block';

newsLoading.textContent = 'Fetching latest anime news...';

try {

// Check if we have cached news that's still fresh

const now = Date.now();

if (newsCache.length > 0 && (now - lastFetchTime) < CACHE\_DURATION) {

displayNews(newsCache);

return;

}

// Fetch from multiple sources

const newsPromises = newsSources.map(source => fetchFromSource(source));

const results = await Promise.allSettled(newsPromises);

let allNews = [];

results.forEach((result, index) => {

if (result.status === 'fulfilled' && result.value) {

allNews = allNews.concat(result.value);

} else {

console.warn(`Failed to fetch from ${newsSources[index].name}:`, result.reason);

}

});

// Sort by recency and trending indicators

allNews.sort((a, b) => {

if (a.trending && !b.trending) return -1;

if (!a.trending && b.trending) return 1;

return new Date(b.pubDate) - new Date(a.pubDate);

});

// Cache the results

newsCache = allNews.slice(0, 20); // Keep top 20 news items

lastFetchTime = now;

displayNews(newsCache);

} catch (error) {

console.error('Error loading news:', error);

newsLoading.textContent = 'Error loading news. Using cached data...';

// Fall back to sample data if available

if (newsCache.length === 0) {

newsCache = getSampleNews();

}

displayNews(newsCache);

}

}

async function fetchFromSource(source) {

try {

// Use a CORS proxy service for RSS feeds

const proxyUrl = `https://api.allorigins.win/get?url=${encodeURIComponent(source.rss)}`;

const response = await fetch(proxyUrl);

const data = await response.json();

// Parse XML content

const parser = new DOMParser();

const xml = parser.parseFromString(data.contents, 'text/xml');

const items = xml.querySelectorAll('item');

const news = [];

items.forEach((item, index) => {

if (index < 5) { // Limit to 5 items per source

const title = item.querySelector('title')?.textContent || '';

const description = item.querySelector('description')?.textContent || '';

const pubDate = item.querySelector('pubDate')?.textContent || '';

const link = item.querySelector('link')?.textContent || '';

// Clean up description (remove HTML tags)

const cleanDescription = description.replace(/<[^>]\*>/g, '').substring(0, 200);

// Determine if trending based on keywords

const trendingKeywords = ['breaking', 'announced', 'new', 'season', 'movie', 'release'];

const isTrending = trendingKeywords.some(keyword =>

title.toLowerCase().includes(keyword) ||

cleanDescription.toLowerCase().includes(keyword)

);

news.push({

id: `${source.name}-${index}`,

title: title.trim(),

summary: cleanDescription.trim(),

category: source.category,

trending: isTrending,

timestamp: formatTimestamp(pubDate),

source: source.name,

link: link,

pubDate: pubDate

});

}

});

return news;

} catch (error) {

console.error(`Error fetching from ${source.name}:`, error);

return [];

}

}

function displayNews(newsItems) {

const newsFeed = document.getElementById('newsFeed');

const newsLoading = document.getElementById('newsLoading');

newsLoading.style.display = 'none';

newsFeed.innerHTML = '';

if (newsItems.length === 0) {

newsFeed.innerHTML = '<div class="news-item">No news available. Please try refreshing.</div>';

return;

}

newsItems.forEach(item => {

const newsElement = document.createElement('div');

newsElement.className = 'news-item';

newsElement.onclick = () => selectNews(item);

newsElement.innerHTML = `

<div class="news-title">${item.trending ? '🔥 ' : ''}${item.title}</div>

<div class="news-meta">${item.category} • ${item.timestamp} • ${item.source}</div>

<div class="news-summary">${item.summary}</div>

`;

newsFeed.appendChild(newsElement);

});

}

function formatTimestamp(pubDate) {

if (!pubDate) return 'Unknown';

const date = new Date(pubDate);

const now = new Date();

const diffInHours = Math.floor((now - date) / (1000 \* 60 \* 60));

if (diffInHours < 1) return 'Just now';

if (diffInHours < 24) return `${diffInHours} hours ago`;

if (diffInHours < 48) return 'Yesterday';

return `${Math.floor(diffInHours / 24)} days ago`;

}

function getSampleNews() {

// Fallback sample data

return [

{

id: 1,

title: "New Attack on Titan Movie Announced for 2025",

summary: "Studio WIT confirms a new Attack on Titan movie focusing on Levi's backstory, set to release in spring 2025.",

category: "Movies",

trending: true,

timestamp: "2 hours ago",

source: "Sample Data"

},

{

id: 2,

title: "Demon Slayer Season 4 Gets Release Date",

summary: "The highly anticipated fourth season of Demon Slayer will premiere in April 2025, featuring the Infinity Castle arc.",

category: "TV Series",

trending: true,

timestamp: "4 hours ago",

source: "Sample Data"

}

];

}

function selectNews(newsItem) {

selectedNews = newsItem;

// Update visual selection

document.querySelectorAll('.news-item').forEach(item => {

item.classList.remove('selected');

});

event.currentTarget.classList.add('selected');

// Update content preview placeholder

const preview = document.getElementById('contentPreview');

preview.textContent = `Selected: ${newsItem.title}\n\nClick "Generate Content" to create TikTok material for this news item.`;

}

function selectTemplate(template) {

currentTemplate = template;

// Update button states

document.querySelectorAll('.template-btn').forEach(btn => {

btn.classList.remove('active');

});

event.currentTarget.classList.add('active');

}

function generateContent() {

if (!selectedNews) {

alert('Please select a news item first!');

return;

}

const style = document.getElementById('contentStyle').value;

const duration = document.getElementById('duration').value;

const audience = document.getElementById('audience').value;

const preview = document.getElementById('contentPreview');

let content = '';

if (currentTemplate === 'script') {

content = generateScript(selectedNews, style, duration, audience);

} else if (currentTemplate === 'hooks') {

content = generateHooks(selectedNews, style);

} else if (currentTemplate === 'hashtags') {

content = generateHashtags(selectedNews);

} else if (currentTemplate === 'thumbnail') {

content = generateThumbnailIdeas(selectedNews);

} else if (currentTemplate === 'videoSetup') {

content = generateVideoSetup(selectedNews, style, duration);

} else if (currentTemplate === 'automation') {

content = generateAutomationScript(selectedNews, style, duration);

}

preview.textContent = content;

contentCount++;

updateStats();

}

function generateVideoPackage() {

if (!selectedNews) {

alert('Please select a news item first!');

return;

}

const style = document.getElementById('contentStyle').value;

const duration = document.getElementById('duration').value;

const audience = document.getElementById('audience').value;

// Generate complete video package

const videoPackage = {

script: generateScript(selectedNews, style, duration, audience),

videoSetup: generateVideoSetup(selectedNews, style, duration),

automation: generateAutomationScript(selectedNews, style, duration),

thumbnails: generateThumbnailIdeas(selectedNews),

hashtags: generateHashtags(selectedNews),

hooks: generateHooks(selectedNews, style)

};

const preview = document.getElementById('contentPreview');

preview.textContent = `🎬 COMPLETE VIDEO PACKAGE GENERATED!

${videoPackage.script}

---

${videoPackage.videoSetup}

---

${videoPackage.automation}

---

${videoPackage.thumbnails}

---

${videoPackage.hashtags}

---

${videoPackage.hooks}`;

contentCount++;

updateStats();

}

function generateScript(news, style, duration, audience) {

const hooks = {

breaking: "🚨 BREAKING: This just happened in anime!",

casual: "Yo anime fam, you need to hear this!",

analysis: "Let's break down this major anime news...",

hype: "🔥 THE HYPE IS REAL! This is INSANE!",

comparison: "Quick anime fact that'll blow your mind!"

};

const endings = {

breaking: "What do you think? Drop your reactions below! 👇",

casual: "Are you excited? Let me know in the comments!",

analysis: "What's your take on this? I'd love to hear your thoughts!",

hype: "WHO ELSE IS HYPED?! Smash that follow for more! 🔥",

comparison: "Mind blown? Follow for more anime facts! 🤯"

};

return `🎬 TIKTOK SCRIPT (${duration}s)

HOOK (0-3s):

"${hooks[style]}"

MAIN CONTENT (3-${duration-5}s):

"${news.title}

${news.summary}

This is ${audience === 'newcomers' ? 'perfect for new anime fans' : audience === 'hardcore' ? 'what we hardcore fans have been waiting for' : 'exactly what the anime community needed'}!"

CALL TO ACTION (${duration-5}-${duration}s):

"${endings[style]}"

---

VISUAL CUES:

• Start with eye-catching text overlay

• Show relevant anime clips/images

• Use trending effects/transitions

• End with subscribe animation

TRENDING SOUNDS:

• Use popular anime opening remix

• Or trending TikTok audio with anime theme

• Keep music engaging but not overpowering`;

}

function generateHooks(news, style) {

const hooks = [

`🚨 ${news.title.toUpperCase()}`,

`POV: You're an anime fan and this just dropped...`,

`Tell me you're an otaku without telling me... I'll go first:`,

`This anime news has me SHOOK 😱`,

`Anime fans, we WON today! Here's why:`,

`I can't believe this is actually happening...`,

`${news.category} fans, this one's for you! 🔥`,

`Plot twist: This anime news is actually INSANE`,

`When someone asks why I love anime, I show them this:`,

`Breaking: The anime community is losing its MIND over this`

];

return `🎣 HOOK VARIATIONS FOR: ${news.title}

${hooks.map((hook, i) => `${i + 1}. ${hook}`).join('\n')}

💡 HOOK TIPS:

• Use first 3 seconds to grab attention

• Include trending keywords

• Create curiosity gap

• Use emojis strategically

• Test multiple hooks for same content`;

}

function generateHashtags(news) {

const baseHashtags = ['#anime', '#manga', '#otaku', '#animenews', '#animecommunity'];

const trendingHashtags = ['#fyp', '#viral', '#trending', '#animeftiktok', '#weeb'];

const specificHashtags = {

'Attack on Titan': ['#attackontitan', '#aot', '#shingekinokyojin', '#levi'],

'Demon Slayer': ['#demonslayer', '#kimetsunoyaiba', '#tanjiro', '#nezuko'],

'One Piece': ['#onepiece', '#luffy', '#strawhats', '#eiichirooda'],

'Studio Ghibli': ['#studioghibli', '#miyazaki', '#ghibli', '#totoro']

};

let specific = [];

Object.keys(specificHashtags).forEach(key => {

if (news.title.includes(key)) {

specific = specificHashtags[key];

}

});

const categoryHashtags = {

'Movies': ['#animemovie', '#film', '#cinema'],

'TV Series': ['#animeseries', '#episode', '#season'],

'Manga': ['#mangareader', '#chapter', '#mangaka'],

'Streaming': ['#crunchyroll', '#netflix', '#streaming']

};

const allHashtags = [

...baseHashtags,

...trendingHashtags,

...specific,

...(categoryHashtags[news.category] || [])

];

return `🏷️ HASHTAG STRATEGY FOR: ${news.title}

PRIMARY HASHTAGS (Always use):

${baseHashtags.join(' ')}

TRENDING HASHTAGS (For reach):

${trendingHashtags.join(' ')}

SPECIFIC HASHTAGS (For targeting):

${specific.length ? specific.join(' ') : '#animeupdates #newanime #animeannouncement'}

CATEGORY HASHTAGS:

${(categoryHashtags[news.category] || ['#general']).join(' ')}

📊 HASHTAG TIPS:

• Use 15-20 hashtags maximum

• Mix popular and niche hashtags

• Check trending hashtags daily

• Create branded hashtag for your account

• Avoid banned or shadowbanned hashtags

COPY-READY:

${allHashtags.slice(0, 18).join(' ')}`;

}

function generateVideoSetup(news, style, duration) {

return `🎬 VIDEO PRODUCTION SETUP FOR: ${news.title}

📋 SHOT LIST (${duration}s video):

1. OPENING HOOK (0-3s)

- Quick zoom transition

- Bold text overlay: "${getHookText(style)}"

- Background: Dark gradient with anime aesthetic

2. NEWS REVEAL (3-8s)

- Slide transition revealing news title

- Text animation: Typewriter effect

- Background: Relevant anime screenshots/artwork

3. MAIN CONTENT (8-${duration-7}s)

- Split screen: Text + visuals

- Smooth scroll through key points

- Background: Animated anime scenes (low opacity)

4. CALL TO ACTION (${duration-7}-${duration}s)

- Zoom in effect

- Subscribe button animation

- End screen with channel branding

🎨 VISUAL ELEMENTS:

• Color Scheme: ${getColorScheme(style)}

• Font: Bold, modern sans-serif (Impact/Arial Black)

• Animations: Smooth transitions, no jarring cuts

• Overlay Graphics: News ticker style at bottom

🎵 AUDIO RECOMMENDATIONS:

• Background Music: Trending TikTok audio or lo-fi anime remix

• Volume: 60% music, 40% voiceover

• Effects: Subtle whoosh sounds for transitions

📱 MOBILE OPTIMIZATION:

• Text size: Minimum 48px for readability

• Safe areas: Keep important elements in center 80%

• Vertical format: 9:16 aspect ratio

🛠️ TOOLS TO USE:

1. CapCut (Mobile/Desktop) - Free, TikTok optimized

2. InShot (Mobile) - Quick editing

3. Canva (Web) - Text overlays and graphics

4. Loom (Desktop) - Screen recording if needed

⚙️ EXPORT SETTINGS:

• Resolution: 1080x1920 (vertical)

• Frame Rate: 30fps

• Bitrate: 8-10 Mbps

• Format: MP4 (H.264)

• Audio: AAC, 128kbps`;

}

function generateAutomationScript(news, style, duration) {

return `🤖 AUTOMATION SCRIPT FOR: ${news.title}

// CapCut API Automation Script (Pseudo-code)

const videoConfig = {

"project\_name": "${news.title.replace(/[^a-zA-Z0-9]/g, '\_')}",

"duration": ${duration},

"aspect\_ratio": "9:16",

"template": "${style}\_news\_template"

};

// 1. TEXT OVERLAYS

const textElements = [

{

"text": "${getHookText(style)}",

"start\_time": 0,

"duration": 3,

"position": "center",

"font\_size": 48,

"color": "#FFFFFF",

"background": "gradient\_red"

},

{

"text": "${news.title}",

"start\_time": 3,

"duration": 5,

"position": "center",

"font\_size": 36,

"animation": "typewriter"

},

{

"text": "${news.summary.substring(0, 100)}...",

"start\_time": 8,

"duration": ${duration-15},

"position": "lower\_third",

"font\_size": 24

}

];

// 2. BACKGROUND MEDIA

const mediaElements = [

{

"type": "image",

"source": "anime\_background\_${style}.jpg",

"start\_time": 0,

"duration": ${duration},

"opacity": 0.3

}

];

// 3. AUDIO TRACK

const audioConfig = {

"background\_music": "trending\_anime\_remix.mp3",

"voice\_over": "generated\_voice\_${news.id}.mp3",

"music\_volume": 0.3,

"voice\_volume": 0.7

};

📝 PYTHON AUTOMATION SCRIPT:

\`\`\`python

import requests

import json

from moviepy.editor import \*

def create\_tiktok\_video(news\_data):

# 1. Generate voiceover using TTS

voice\_script = "${generateVoiceScript(news)}"

# 2. Create text clips

title\_clip = TextClip("${news.title}",

fontsize=48,

color='white',

duration=3)

# 3. Add background

bg = ColorClip(size=(1080,1920),

color=(0,0,0),

duration=${duration})

# 4. Compose final video

final\_video = CompositeVideoClip([

bg,

title\_clip.set\_position('center')

])

# 5. Export

final\_video.write\_videofile(

"${news.title.replace(/[^a-zA-Z0-9]/g, '\_')}.mp4",

fps=30,

codec='libx264'

)

\`\`\`

🔧 BATCH PROCESSING:

• Set up folder structure: /scripts, /audio, /visuals, /output

• Use templates for consistent branding

• Automate upload with TikTok API (when available)

• Schedule posts using Buffer/Hootsuite

⚡ QUICK SETUP COMMANDS:

1. pip install moviepy requests

2. Download anime background pack

3. Set up TTS service (ElevenLabs/Azure)

4. Configure export presets

5. Run automation script`;

}

function getHookText(style) {

const hooks = {

breaking: "🚨 BREAKING ANIME NEWS!",

casual: "Yo anime fam! 👋",

analysis: "Let's analyze this... 🤔",

hype: "THE HYPE IS REAL! 🔥",

comparison: "Mind = Blown 🤯"

};

return hooks[style] || "Check this out! ✨";

}

function getColorScheme(style) {

const schemes = {

breaking: "Red/Black/White (Urgent)",

casual: "Blue/Purple (Friendly)",

analysis: "Dark Blue/Gold (Professional)",

hype: "Orange/Red (Energetic)",

comparison: "Green/Blue (Informative)"

};

return schemes[style] || "Purple/Blue (Default)";

}

function generateVoiceScript(news) {

return `${news.title}. ${news.summary}. What do you think about this? Let me know in the comments!`;

}

function generateThumbnailIdeas(news) {

return `🖼️ THUMBNAIL IDEAS FOR: ${news.title}

DESIGN CONCEPTS:

1. 🔥 Split screen: Before/After or Character comparison

2. 📰 Newspaper style with bold headlines

3. 🎭 Character reaction faces (shocked/excited)

4. ⚡ Action scene with dramatic lighting

5. 📺 TV screen mockup showing announcement

COLOR SCHEMES:

• High contrast (Black/White/Red)

• Anime-inspired (Bright blues, oranges, yellows)

• Dark mode with neon accents

• Classic manga black and white

TEXT ELEMENTS:

• "BREAKING" or "NEW" badges

• Large, bold title text

• Question marks for curiosity

• Countdown timers or dates

• Your channel branding

VISUAL EFFECTS:

• Glowing borders or highlights

• Motion blur for action

• Explosion or impact effects

• Trending symbols (🔥⚡💥)

• Character cutouts with shadows

TOOLS TO USE:

• Canva (templates available)

• Photoshop/GIMP

• Figma (free design tool)

• Procreate (mobile)

💡 PRO TIPS:

• Test multiple thumbnails

• Use faces when possible (higher CTR)

• Keep text readable on mobile

• Match your brand colors

• A/B test different styles`;

}

function refreshNews() {

// Clear cache to force fresh fetch

newsCache = [];

lastFetchTime = 0;

loadNews();

updateStats();

}

function updateStats() {

document.getElementById('newsCount').textContent = newsCache.length;

document.getElementById('contentGenerated').textContent = contentCount;

// Update trending count based on actual trending news

const trendingCount = newsCache.filter(item => item.trending).length;

document.getElementById('trending').textContent = trendingCount;

}

function generateHashtagSuggestions() {

const suggestions = ['#animenews', '#otaku', '#weeb', '#manga', '#animecommunity', '#fyp'];

const container = document.getElementById('hashtagSuggestions');

container.innerHTML = suggestions.map(tag =>

`<span class="hashtag" onclick="addHashtagToContent('${tag}')">${tag}</span>`

).join('');

}

function addHashtagToContent(hashtag) {

const preview = document.getElementById('contentPreview');

preview.textContent += ' ' + hashtag;

}

function copyContent() {

const content = document.getElementById('contentPreview').textContent;

navigator.clipboard.writeText(content).then(() => {

alert('Content copied to clipboard!');

});

}

function saveContent() {

const content = document.getElementById('contentPreview').textContent;

if (content.trim()) {

savedTemplates.push({

content,

template: currentTemplate,

timestamp: new Date().toLocaleString()

});

alert('Template saved successfully!');

}

}

function addKeywords() {

const keywords = document.getElementById('customKeywords').value;

if (keywords.trim()) {

const preview = document.getElementById('contentPreview');

preview.textContent += '\n\nCustom Keywords: ' + keywords;

document.getElementById('customKeywords').value = '';

}

}

// Initialize the app

initializeApp();

// Auto-refresh news every 30 minutes

setInterval(() => {

console.log('Auto-refreshing news...');

refreshNews();

}, 30 \* 60 \* 1000);

</script>

</body>

</html>