**Full PHP Code for Advanced AI Image Title Generator**

<?php

/\*\*

\* Pluginless snippet: AI Image Title Generator (rich JSON I/O)

\* Shortcode: [ai\_image\_title\_generator]

\*

\* SECURITY:

\* - OpenAI API key is inserted below (Option A). It is used server-side only.

\* - Nonce-protected AJAX.

\*

\* FEATURES:

\* - Structured inputs (image context + preferences + advanced controls).

\* - Validated JSON payload sent to OpenAI.

\* - Strict JSON output with titles, scores, platform fit, variations, and recommendations.

\* - Pretty, responsive UI with client-side rendering.

\*/

/\*\* =======================

\* 1) CONFIGURATION

\* ======================= \*/

if (!defined('OPENAI\_API\_KEY')) {

// Option A) Your API key inserted here (single place)

define('OPENAI\_API\_KEY', 'sk-proj-fWdpZZ51YMORKv4cb\_eCpZD23HE5k1FpkCiROnlhl7XPsvLmHCiPZYJ8ASBYjbbFd2BUN1zPJRT3BlbkFJ2NMDXbqRo3HB4ovPGDuJn-WaPUWe7P8hVAilPt5u2NyalABmOJI9TBuvInKNQZys16lgmlQMcA');

}

if (!defined('OPENAI\_MODEL\_ID')) {

// Use a JSON-capable model. Adjust if you have access to other models.

define('OPENAI\_MODEL\_ID', 'gpt-4o-mini'); // e.g., 'gpt-4.1-mini', 'gpt-4o', etc.

}

/\*\* =======================

\* 2) SHORTCODE UI

\* ======================= \*/

add\_shortcode('ai\_image\_title\_generator', function () {

if (defined('REST\_REQUEST') && REST\_REQUEST) return ''; // avoid rendering in REST previews

$ajax\_url = admin\_url('admin-ajax.php');

$nonce = wp\_create\_nonce('ai\_image\_title\_gen\_nonce');

ob\_start(); ?>

<style>

.aiitg-wrap { max-width: 980px; margin: 24px auto; font-family: system-ui, -apple-system, Segoe UI, Roboto, Helvetica, Arial, sans-serif; }

.aiitg-card { background: #fff; border: 1px solid #e5e7eb; border-radius: 16px; padding: 20px; box-shadow: 0 2px 10px rgba(0,0,0,.04); }

.aiitg-grid { display: grid; grid-template-columns: 1fr; gap: 16px; }

@media (min-width: 900px) { .aiitg-grid { grid-template-columns: 1fr 1fr; } }

.aiitg-h { margin: 6px 0 10px; font-weight: 700; }

.aiitg-sub { color: #6b7280; font-size: 14px; margin-bottom: 10px; }

.aiitg-label { display:block; font-weight:600; margin:6px 0; }

.aiitg-input, .aiitg-select, .aiitg-textarea {

width: 100%; border: 1px solid #d1d5db; border-radius: 12px; padding: 10px 12px; font-size: 14px; outline: none; background: #fafafa;

}

.aiitg-row { display: grid; grid-template-columns: 1fr 1fr; gap: 10px; }

.aiitg-actions { display:flex; gap:12px; align-items:center; }

.aiitg-btn {

appearance:none; cursor:pointer; border-radius: 12px; border:1px solid #111827;

padding: 10px 14px; font-weight:700; background:#111827; color:#fff;

}

.aiitg-btn[disabled] { opacity:.6; cursor:not-allowed; }

.aiitg-muted-btn { background:#fff; color:#111827; border-color:#d1d5db; }

.aiitg-results { margin-top: 18px; }

.aiitg-title-card { border:1px solid #e5e7eb; border-radius:12px; padding:14px; margin:12px 0; background:#fcfcfd; }

.aiitg-badge { display:inline-block; font-size:12px; padding:2px 8px; border:1px solid #e5e7eb; border-radius:999px; margin-right:6px; background:#fff; }

.aiitg-small { font-size:12px; color:#6b7280; }

.aiitg-code { background:#111827; color:#e5e7eb; font-family: ui-monospace, SFMono-Regular, Menlo, Monaco, Consolas, "Liberation Mono", "Courier New", monospace; padding:12px; border-radius:12px; overflow:auto; max-height:260px; }

.aiitg-loader { display:none; gap:8px; align-items:center; }

.aiitg-dot { width:8px; height:8px; border-radius:50%; background:#111827; animation: aiitg-blink 1.2s infinite ease-in-out; }

.aiitg-dot:nth-child(2) { animation-delay: .15s; } .aiitg-dot:nth-child(3) { animation-delay: .3s; }

@keyframes aiitg-blink { 0%, 80%, 100% { opacity:.2 } 40% { opacity:1 } }

.aiitg-hint { background:#f8fafc; border:1px dashed #d1d5db; padding:10px 12px; border-radius:10px; margin-top:8px; }

</style>

<div class="aiitg-wrap">

<div class="aiitg-card">

<h2 class="aiitg-h">AI Image Title Generator — Pro Controls</h2>

<div class="aiitg-sub">Feed the model rich context and get back titles with SEO, emotion, clickbait rating, platform fit, and ready-to-use recommendations.</div>

<div class="aiitg-grid">

<!-- Image Context -->

<div>

<h3 class="aiitg-h">Image Context</h3>

<label class="aiitg-label">Image Category</label>

<select class="aiitg-select" id="image\_category">

<option value="">(select)</option>

<option>fantasy</option><option>scifi</option><option>product</option>

<option>landscape</option><option>portrait</option><option>abstract</option>

<option>concept art</option><option>surrealism</option><option>anime</option>

</select>

<label class="aiitg-label">Subject Focus (what’s in the image?)</label>

<input class="aiitg-input" id="subject\_focus" placeholder="e.g., dragon in a lightning storm over mountains" />

<div class="aiitg-row">

<div>

<label class="aiitg-label">Style</label>

<input class="aiitg-input" id="style" placeholder="realistic, watercolor, cyberpunk..." />

</div>

<div>

<label class="aiitg-label">Lighting</label>

<input class="aiitg-input" id="lighting" placeholder="golden hour, studio, neon glow..." />

</div>

</div>

<div class="aiitg-row">

<div>

<label class="aiitg-label">Color Palette</label>

<input class="aiitg-input" id="color\_palette" placeholder="warm, pastel, monochrome..." />

</div>

<div>

<label class="aiitg-label">Composition</label>

<input class="aiitg-input" id="composition" placeholder="close-up, wide shot, isometric..." />

</div>

</div>

<div class="aiitg-row">

<div>

<label class="aiitg-label">Mood / Tone</label>

<input class="aiitg-input" id="mood\_tone" placeholder="dramatic, whimsical, mysterious..." />

</div>

<div>

<label class="aiitg-label">Audience Type</label>

<input class="aiitg-input" id="audience\_type" placeholder="gamers, kids, nature lovers..." />

</div>

</div>

<label class="aiitg-label">Purpose / Use</label>

<input class="aiitg-input" id="purpose" placeholder="social media, poster, ad campaign, concept art..." />

</div>

<!-- Title Preferences -->

<div>

<h3 class="aiitg-h">Title Preferences</h3>

<label class="aiitg-label">Keywords (comma separated)</label>

<input class="aiitg-input" id="keywords" placeholder="fantasy, epic, storm" />

<div class="aiitg-row">

<div>

<label class="aiitg-label">Title Length</label>

<select class="aiitg-select" id="title\_length">

<option value="">auto</option><option>short</option><option>medium</option><option>long</option>

</select>

</div>

<div>

<label class="aiitg-label">Language</label>

<input class="aiitg-input" id="language" placeholder="English" value="English" />

</div>

</div>

<div class="aiitg-row">

<div>

<label class="aiitg-label">Tone</label>

<input class="aiitg-input" id="tone" placeholder="serious, humorous, poetic, clickbait..." />

</div>

<div>

<label class="aiitg-label">Number of Titles</label>

<input type="number" min="1" max="12" value="5" class="aiitg-input" id="number\_of\_titles" />

</div>

</div>

<div class="aiitg-row">

<div>

<label class="aiitg-label">SEO Focus</label>

<select class="aiitg-select" id="seo\_focus"><option>true</option><option>false</option></select>

</div>

<div>

<label class="aiitg-label">Include Emotion Trigger</label>

<select class="aiitg-select" id="include\_emotion\_trigger"><option>true</option><option>false</option></select>

</div>

</div>

<div class="aiitg-row">

<div>

<label class="aiitg-label">Wordplay Allowed</label>

<select class="aiitg-select" id="wordplay\_allowed"><option>true</option><option>false</option></select>

</div>

<div>

<label class="aiitg-label">Target Platform</label>

<input class="aiitg-input" id="target\_platform" placeholder="Instagram, Pinterest, YouTube Thumbnail..." />

</div>

</div>

<label class="aiitg-label">Avoid Words (comma separated)</label>

<input class="aiitg-input" id="avoid\_words" placeholder="boring, generic, stock" />

<label class="aiitg-label">Reference Titles (one per line)</label>

<textarea class="aiitg-textarea" id="reference\_titles" rows="4" placeholder="Whispers of the Forgotten Sky&#10;Neon Samurai"></textarea>

<label class="aiitg-label">Inspiration Sources (URLs or artist names, comma separated)</label>

<input class="aiitg-input" id="inspiration\_sources" placeholder="https://..., H.R. Giger, Beeple" />

</div>

</div>

<div class="aiitg-actions" style="margin-top:14px;">

<button class="aiitg-btn" id="aiitg-generate">Generate Titles</button>

<button class="aiitg-btn aiitg-muted-btn" id="aiitg-reset" type="button">Reset</button>

<div class="aiitg-loader" id="aiitg-loader"><div class="aiitg-dot"></div><div class="aiitg-dot"></div><div class="aiitg-dot"></div><span class="aiitg-small">&nbsp;Thinking…</span></div>

</div>

<div class="aiitg-hint">

Tip: Fill at least <strong>Image Category</strong>, <strong>Subject Focus</strong>, and a few <strong>Keywords</strong> for best results.

</div>

<div class="aiitg-results" id="aiitg-results" style="display:none;">

<h3 class="aiitg-h">Results</h3>

<div id="aiitg-summary"></div>

<div id="aiitg-title-list"></div>

<details style="margin-top:10px;">

<summary><strong>Raw JSON (debug)</strong></summary>

<pre class="aiitg-code" id="aiitg-raw"></pre>

</details>

</div>

</div>

</div>

<script>

(function(){

const $ = (sel) => document.querySelector(sel);

const ajaxUrl = <?php echo json\_encode($ajax\_url); ?>;

const nonce = <?php echo json\_encode($nonce); ?>;

const toList = (str) => (str || '').split(',').map(s => s.trim()).filter(Boolean);

const toLines= (str) => (str || '').split('\n').map(s => s.trim()).filter(Boolean);

function buildPayload(){

return {

project\_info: {

project\_id: "",

session\_id: "",

user\_id: ""

},

image\_context: {

image\_category: $("#image\_category").value || "",

subject\_focus: $("#subject\_focus").value || "",

style: $("#style").value || "",

lighting: $("#lighting").value || "",

color\_palette: $("#color\_palette").value || "",

composition: $("#composition").value || "",

mood\_tone: $("#mood\_tone").value || "",

audience\_type: $("#audience\_type").value || "",

purpose: $("#purpose").value || ""

},

title\_generation\_preferences: {

keywords: toList($("#keywords").value),

title\_length: $("#title\_length").value || "auto",

language: $("#language").value || "English",

tone: $("#tone").value || "",

seo\_focus: ($("#seo\_focus").value === "true"),

include\_emotion\_trigger: ($("#include\_emotion\_trigger").value === "true"),

wordplay\_allowed: ($("#wordplay\_allowed").value === "true"),

number\_of\_titles: Math.max(1, Math.min(12, parseInt($("#number\_of\_titles").value || "5", 10)))

},

advanced\_controls: {

avoid\_words: toList($("#avoid\_words").value),

reference\_titles: toLines($("#reference\_titles").value),

inspiration\_sources: toList($("#inspiration\_sources").value),

target\_platform: $("#target\_platform").value || ""

}

};

}

function renderResults(json){

const resultsWrap = $("#aiitg-results");

const list = $("#aiitg-title-list");

const summary = $("#aiitg-summary");

const raw = $("#aiitg-raw");

resultsWrap.style.display = 'block';

list.innerHTML = "";

raw.textContent = JSON.stringify(json, null, 2);

if (json?.overall\_analysis) {

summary.innerHTML = `

<div class="aiitg-title-card">

<div><span class="aiitg-badge">Best for Engagement</span> ${json.overall\_analysis.best\_for\_engagement || '-'}</div>

<div><span class="aiitg-badge">Highest SEO</span> ${json.overall\_analysis.highest\_seo || '-'}</div>

<div><span class="aiitg-badge">Most Emotional</span> ${json.overall\_analysis.most\_emotional || '-'}</div>

<div class="aiitg-small">Diversity Score: ${json.overall\_analysis.diversity\_of\_titles\_score ?? '-'}</div>

</div>

`;

}

(json?.titles || []).forEach((t, idx) => {

const platformFit = t.platform\_fit ? Object.entries(t.platform\_fit).map(([k,v]) => `<span class="aiitg-badge">${k}: ${v}</span>`).join(' ') : '';

const variations = (t.variations || []).map(v => `<li>${v}</li>`).join('');

const keywords = (t.keywords\_used || []).map(k => `<span class="aiitg-badge">${k}</span>`).join(' ');

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

el.className = 'aiitg-title-card';

el.innerHTML = `

<div style="display:flex;justify-content:space-between;gap:8px;align-items:baseline;">

<h4 class="aiitg-h">#${idx+1} — ${t.title\_text || '-'}</h4>

<div class="aiitg-small">

SEO: ${t.seo\_score ?? '-'} | Clickbait: ${t.clickbait\_rating ?? '-'} | Readability: ${t.readability\_score ?? '-'}

</div>

</div>

<div class="aiitg-small" style="margin-bottom:6px;">Emotion: ${t.emotion\_trigger || '-'}</div>

<div>${t.explanation || ''}</div>

<div style="margin-top:8px;">${keywords}</div>

<div style="margin-top:8px;">${platformFit}</div>

${variations ? `<div style="margin-top:8px;"><strong>Variations:</strong><ul>${variations}</ul></div>` : ''}

`;

list.appendChild(el);

});

if (json?.recommendations) {

const tags = (json.recommendations.suggested\_hashtags || []).map(h => `<span class="aiitg-badge">${h}</span>`).join(' ');

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

rec.className = 'aiitg-title-card';

rec.innerHTML = `

<div><strong>Suggested Description:</strong><br>${json.recommendations.suggested\_description || '-'}</div>

<div style="margin-top:8px;">${tags}</div>

`;

list.appendChild(rec);

}

}

function setLoading(isLoading){

$("#aiitg-generate").disabled = isLoading;

$("#aiitg-loader").style.display = isLoading ? 'flex' : 'none';

}

$("#aiitg-reset").addEventListener('click', () => {

document.querySelectorAll('#aiitg-results, #aiitg-title-list, #aiitg-summary, #aiitg-raw').forEach(el => {

if (el.id === 'aiitg-results') el.style.display = 'none'; else el.innerHTML = '';

});

document.querySelectorAll('.aiitg-input, .aiitg-select, .aiitg-textarea').forEach(el => {

if (el.id === 'language') { el.value = 'English'; return; }

if (el.id === 'seo\_focus' || el.id === 'include\_emotion\_trigger' || el.id === 'wordplay\_allowed') { el.value = 'true'; return; }

if (el.id === 'number\_of\_titles') { el.value = 5; return; }

el.value = '';

});

});

$("#aiitg-generate").addEventListener('click', function(e){

e.preventDefault();

const payload = buildPayload();

// basic client-side validation

if (!payload.image\_context.image\_category || !payload.image\_context.subject\_focus) {

alert("Please fill at least Image Category and Subject Focus.");

return;

}

setLoading(true);

const form = new FormData();

form.append('action', 'ai\_image\_title\_generator');

form.append('nonce', nonce);

form.append('payload', JSON.stringify(payload));

fetch(ajaxUrl, { method: 'POST', body: form })

.then(r => r.json())

.then(data => {

if (data?.success) {

renderResults(data.data);

} else {

console.error(data);

alert(data?.message || 'Failed to generate. Check console for details.');

}

})

.catch(err => {

console.error(err);

alert('Network or server error.');

})

.finally(() => setLoading(false));

});

})();

</script>

<?php

return ob\_get\_clean();

});

/\*\* =======================

\* 3) AJAX HANDLER

\* ======================= \*/

add\_action('wp\_ajax\_ai\_image\_title\_generator', 'ai\_image\_title\_generator\_cb');

add\_action('wp\_ajax\_nopriv\_ai\_image\_title\_generator', 'ai\_image\_title\_generator\_cb');

function ai\_image\_title\_generator\_cb() {

if (!isset($\_POST['nonce']) || !wp\_verify\_nonce($\_POST['nonce'], 'ai\_image\_title\_gen\_nonce')) {

wp\_send\_json(['success' => false, 'message' => 'Invalid nonce'], 403);

}

$payload\_raw = isset($\_POST['payload']) ? wp\_unslash($\_POST['payload']) : '';

if (!$payload\_raw) {

wp\_send\_json(['success' => false, 'message' => 'Missing payload']);

}

$payload = json\_decode($payload\_raw, true);

if (!is\_array($payload)) {

wp\_send\_json(['success' => false, 'message' => 'Invalid JSON payload']);

}

// Minimal sanitization (you can harden further as needed)

$image\_category = sanitize\_text\_field($payload['image\_context']['image\_category'] ?? '');

$subject\_focus = sanitize\_text\_field($payload['image\_context']['subject\_focus'] ?? '');

if (!$image\_category || !$subject\_focus) {

wp\_send\_json(['success' => false, 'message' => 'Please provide image\_category and subject\_focus']);

}

// Build a strict system + user prompt to force JSON output

$now = gmdate('c');

$output\_schema = [

"request\_metadata" => [

"generated\_at" => $now,

"model\_used" => OPENAI\_MODEL\_ID,

"processing\_time\_seconds" => "\_\_MODEL\_FILLS\_\_"

],

"titles" => [

[

"title\_text" => "string",

"explanation" => "string",

"keywords\_used" => ["string"],

"seo\_score" => 0,

"clickbait\_rating" => 0.0,

"emotion\_trigger" => "string",

"readability\_score" => 0,

"platform\_fit" => [ "Instagram" => "High", "Pinterest" => "High", "YouTube" => "Medium" ],

"variations" => ["string", "string"]

]

],

"overall\_analysis" => [

"best\_for\_engagement" => "string",

"highest\_seo" => "string",

"most\_emotional" => "string",

"diversity\_of\_titles\_score" => 0.0

],

"recommendations" => [

"suggested\_hashtags" => ["string"],

"suggested\_description" => "string"

]

];

$system = "You are a professional art director and AI prompt engineer.

Return strictly valid JSON for image TITLE ideation with analytics.

No prose, no markdown, no extra text. JSON only.";

$user = [

"instruction" => "Generate rich, SEO-aware, emotionally resonant TITLES for an image based on the input.

Include analysis and platform fit. Avoid disallowed words. Use the requested language.

Respect title\_length and tone. If keywords are provided, use them naturally.",

"input" => $payload,

"output\_schema" => $output\_schema,

"rules" => [

"json\_only" => true,

"titles\_min" => max(1, intval($payload['title\_generation\_preferences']['number\_of\_titles'] ?? 5)),

"language" => $payload['title\_generation\_preferences']['language'] ?? 'English',

"avoid\_words" => $payload['advanced\_controls']['avoid\_words'] ?? [],

"reference\_titles" => $payload['advanced\_controls']['reference\_titles'] ?? [],

"platform" => $payload['advanced\_controls']['target\_platform'] ?? ''

]

];

$api\_key = OPENAI\_API\_KEY;

if (empty($api\_key) || $api\_key === 'REPLACE\_WITH\_YOUR\_OPENAI\_API\_KEY') {

wp\_send\_json(['success' => false, 'message' => 'OpenAI API key missing. Set OPENAI\_API\_KEY in the snippet or as an environment variable.'], 500);

}

// Chat Completions request (OpenAI-compatible)

$body = [

"model" => OPENAI\_MODEL\_ID,

"response\_format" => [ "type" => "json\_object" ],

"temperature" => 0.9,

"messages" => [

[ "role" => "system", "content" => $system ],

[ "role" => "user", "content" => json\_encode($user, JSON\_UNESCAPED\_SLASHES | JSON\_UNESCAPED\_UNICODE) ]

]

];

$args = [

'headers' => [

'Content-Type' => 'application/json',

'Authorization' => 'Bearer ' . $api\_key,

],

'body' => wp\_json\_encode($body),

'timeout' => 45,

];

$response = wp\_remote\_post('https://api.openai.com/v1/chat/completions', $args);

if (is\_wp\_error($response)) {

wp\_send\_json(['success' => false, 'message' => $response->get\_error\_message()], 500);

}

$code = wp\_remote\_retrieve\_response\_code($response);

$res\_body = wp\_remote\_retrieve\_body($response);

if ($code < 200 || $code >= 300) {

wp\_send\_json(['success' => false, 'message' => 'OpenAI API error', 'details' => $res\_body], $code);

}

$data = json\_decode($res\_body, true);

$json\_text = $data['choices'][0]['message']['content'] ?? '';

// Ensure returned content is valid JSON

$json = json\_decode($json\_text, true);

if (!is\_array($json)) {

// Attempt to coerce or return raw for debugging

wp\_send\_json(['success' => false, 'message' => 'Model returned non-JSON output', 'raw' => $json\_text], 500);

}

// Inject/normalize minimal metadata if missing

if (!isset($json['request\_metadata'])) $json['request\_metadata'] = [];

$json['request\_metadata']['generated\_at'] = $json['request\_metadata']['generated\_at'] ?? $now;

$json['request\_metadata']['model\_used'] = $json['request\_metadata']['model\_used'] ?? OPENAI\_MODEL\_ID;

wp\_send\_json(['success' => true, 'data' => $json]);

}