

```

<?php
// Load Composer's autoloader (if using .env)
if (file_exists('vendor/autoload.php')) {
    require 'vendor/autoload.php';
}

// Load environment variables (if using .env)
if (class_exists('Dotenv\Dotenv')) {
    $dotenv = Dotenv\Dotenv::createImmutable(__DIR__);
    $dotenv->load();
}

// Get API Keys from Environment Variables
$aisensy_api_key =
getenv("eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY3YjcwNzY0YWWRiZWZhMGJmZjIxYWQwMmIsIm5hbnWUiOiJWYW5zaCBidXNpbmVzcyAiLCJhcHB0YW1lIjojQWITZW5zeSIsImNsYWVudElkIjojNjdiNzA3NjRhZGJlZmEwYmZmOTFhY2Y3IiwiaWYWN0aXZIUGxhbil6IIBST19NT05USeXZliwiaWF0IjoxNzQwODI2MTMxfQ.e3fHweLzP0YMW9IGR7yPC0I4vPDN_ceGH5WwNwKFBxY") ?: "";
$mailerlite_api_key =
getenv("eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiI0IiwianRpljojZDU0ZjRmYzU4ZWVjZjIhNGEzNjA5MTU2ODVkdNDYzNmMyNzA3Mzg2YWU2Njc4N2Y1YWNIzMzMzOTRiYjQ1NTBhMzc2ODViODMyYmI1ZmY5ZGUlLCJpYXQiOiJlE3NDE5MDc3ODguMDEyNjUyLCJmYmYiOiJlE3NDE5MDc3ODguMDEyNjU0LCJleHAiOiJlE3NDE5MDc3ODguMDA4NDc3LCJzdWliOiI0IiwiaWYWN0aXZIUGxhbil6IIBST19NT05USeXZliwiaWF0IjoxNzQwODI2MTMxfQ.e3fHweLzP0YMW9IGR7yPC0I4vPDN_ceGH5WwNwKFBxY") ?: "";
$mailerlite_group_id = getenv("146740816165995646") ?: "";
$instamojo_salt = getenv("f3b6692e4d8a44249aa8083ab11391d0") ?: ""; // Secret Key for
Signature Validation

// Validate API Keys
if (!$aisensy_api_key || !$mailerlite_api_key || !$instamojo_salt) {
    jsonResponse(500, "API Keys missing.");
}

// Get JSON payload from Instamojo

```

```

$payload = file_get_contents("php://input");
$data = json_decode($payload, true);

// Log payload for debugging (Remove in production)
file_put_contents("instamojo_webhook_log.txt", date("Y-m-d H:i:s") . " | " . json_encode($data) .
PHP_EOL, FILE_APPEND);

// Check if data exists
if (!$data) {
    jsonResponse(400, "No data received.");
}

// Extract details from Instamojo webhook
$name = htmlspecialchars($data['buyer_name'] ?? "Unknown User");
$email = filter_var($data['buyer_email'], FILTER_SANITIZE_EMAIL) ?: null;
$phone = preg_replace("/[^0-9]/", "", $data['buyer_phone'] ?? null);
$order_id = htmlspecialchars($data['payment_request_id'] ?? "No Order ID");
$payment_status = strtolower($data['status'] ?? "failed");
$instamojo_signature = $_SERVER['HTTP_X_INSTAMOJO_SIGNATURE'] ?? null;

// Verify Instamojo Signature (Security Check)
if (!verifyInstamojoSignature($payload, $instamojo_signature, $instamojo_salt)) {
    jsonResponse(403, "Invalid Instamojo signature.");
}

// Select Aisensy campaign based on payment status
$campaign = in_array($payment_status, ["credit", "successful"]) ? "Webinar invitation" :
"Payment pending";

// Function to send WhatsApp message via Aisensy
function sendToAisensy($phone, $name, $order_id, $campaign, $aisensy_api_key) {
    if (!$phone) return ["error" => "Phone number missing"];

    return makeApiRequest("https://backend.aisensy.com/campaign/t1/api/send", [
        "Authorization: Bearer $aisensy_api_key",
        "Content-Type: application/json"
    ], [
        "campaignName" => $campaign,
        "phoneNumbers" => [$phone],
        "parameters" => [$name, $order_id]
    ]);
}

// Function to add a subscriber to MailerLite

```

```

function sendToMailerLite($email, $name, $mailerlite_api_key, $mailerlite_group_id) {
    if (!$email) return ["error" => "Email missing"];

    return makeApiRequest("https://api.mailerlite.com/api/v2/subscribers", [
        "Authorization: Bearer $mailerlite_api_key",
        "Content-Type: application/json"
    ], [
        "email" => $email,
        "name" => $name,
        "resubscribe" => true,
        "groups" => [$mailerlite_group_id]
    ]);
}

// Function to make API requests with error handling
function makeApiRequest($url, $headers, $payload) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($payload));
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

    $response = curl_exec($ch);
    $http_code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
    $curl_error = curl_error($ch);
    curl_close($ch);

    if (!$response || $http_code !== 200) {
        error_log("API Error: $response | HTTP Code: $http_code | cURL Error: $curl_error");
        return ["error" => "API request failed", "http_code" => $http_code, "curl_error" =>
$curl_error];
    }

    return json_decode($response, true);
}

// Function to verify Instamojo webhook signature
function verifyInstamojoSignature($payload, $signature, $salt) {
    if (!$signature) return false;
    $calculated_hash = hash_hmac("sha256", $payload, $salt);
    return hash_equals($calculated_hash, $signature);
}

```

```

// Function to send JSON response
function jsonResponse($status, $message, $data = []) {
    http_response_code($status);
    echo json_encode(["message" => $message, "data" => $data]);
    exit;
}

// Initialize response array
$response = ["message" => "Webhook processed successfully"];

// Send WhatsApp message via Aisensy
if ($phone) {
    $response["aisensy_response"] = sendToAisensy($phone, $name, $order_id, $campaign,
    $aisensy_api_key);
} else {
    error_log("Skipping Aisensy - Phone number missing");
}

// Add subscriber to MailerLite
if ($email) {
    $response["mailerlite_response"] = sendToMailerLite($email, $name, $mailerlite_api_key,
    $mailerlite_group_id);
} else {
    error_log("Skipping MailerLite - Email missing");
}

// Log response summary
error_log("Webhook Processed | Order ID: $order_id | Phone: $phone | Email: $email");

// Return JSON response
jsonResponse(200, "Webhook processed successfully", $response);
?>

```