```php
<?php
/*
 * mailscript.php versenden über SMTP
 * Version vom 27.10.2025
 * Erfolgreich getestet mit: PHPMailer Version: 7.0.0
 *
 * Erweiterungen:
 * - Dynamisches Einsammeln ALLER Formularfelder (egal wie sie heißen)
 * - Automatischer Mailtext-Aufbau per Schleife
 * - Optionale Reply-To-Setzung anhand gefundener E-Mail im Formular
 */

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;

require "PHPMailer-master/src/Exception.php";
require "PHPMailer-master/src/PHPMailer.php";
require "PHPMailer-master/src/SMTP.php";

// Danke- und Fehlerseiten
$dankeSeite  = "danke.html";  // wird nach erfolgreichem Versand aufgerufen
$fehlerSeite = "fehler.html"; // wird bei Fehler aufgerufen

// Fester Betreff (da das Formular selbst keinen Betreff liefert)
$betreffEmail = "Neue Email Anfrage von der Website";

// Wurden POST-Daten gesendet?
if ($_SERVER["REQUEST_METHOD"] == "POST") {

    // 1) Zeitzone + Meta-Infos
    date_default_timezone_set("Europe/Berlin");
    $datum = date("d.m.Y H:i");         // z. B. "27.10.2025 15:42"
    $ipAdr = $_SERVER["REMOTE_ADDR"];  // IP des Absenders

    // 2) Formularwerte vorbereiten / säubern
    //    Wir erzeugen ein neues Array $bereinigt, in dem alle Felder aus $_POST
    //    sauber escaped sind (HTML-Sonderzeichen maskiert).
    $bereinigt = [];

    foreach ($_POST as $feldName => $wert) {

        // 2a) Falls das Feld selbst ein Array ist (z. B. Checkbox-Gruppen mit []),
        //     machen wir daraus eine komma-separierte Liste.
        if (is_array($wert)) {
            $wert = implode(", ", $wert);
        }

        // 2b) Whitespace an den Rändern entfernen
        $wert = trim($wert);

        // 2c) In Mailtext nur entschärfte Inhalte einfügen (XSS-/HTML-Schutz)
        //     ENT_QUOTES: wandelt auch ' und " um
        //     UTF-8 als Charset
        $wert = htmlspecialchars($wert, ENT_QUOTES, 'UTF-8');

        // 2d) Feldname wie gesendet übernehmen (darf
        Groß/Klein/Umlaute/Binde-/Unterstrich haben)
        $bereinigt[$feldName] = $wert;
    }

    // 3) Schönen Mail-Body zusammenbauen
    //    Wir schreiben zuerst Meta-Infos, dann alle Felder dynamisch.
    $inhaltEmail  = "Gesendet am: $datum Uhr\n";
    $inhaltEmail .= "IP-Adresse: $ipAdr\n\n";
    $inhaltEmail .= "Formularinhalt:\n";

```

```php
    foreach ($bereinigt as $feldName => $wert) {
        // Jeder Eintrag eine eigene Zeile:
        // Beispiel: "E-Mail: max@test.de"
        $inhaltEmail .= $feldName . ": " . $wert . "\n";
    }

    // Optional am Ende noch eine Leerzeile
    $inhaltEmail .= "\n";

    // 4) Versuch, eine Absender-Email aus den Formulardaten zu erkennen,
    //    damit wir später Reply-To setzen können.
    //
    //    Hintergrund:
    //    Unterschiedliche Formulare benutzen unterschiedliche Feldnamen:
    //    "email", "Email", "E-Mail", "mail", "kontakt_mail", ...
    //
    //    Wir probieren mehrere gängige Varianten durch.
    $absenderEmail = '';
    $absenderName  = '';

    // Diese Feldnamen gelten als "könnte die E-Mail-Adresse des Besuchers sein"
    $moeglicheEmailFelder = [
        'email', 'Email', 'EMAIL', 'e-mail', 'E-Mail', 'E-mail', 'mail', 'Mail',
        'kontakt', 'Kontakt', 'kontakt_mail', 'kontaktMail'
    ];

    foreach ($moeglicheEmailFelder as $feld) {
        // Wir vergleichen case-insensitive:
        // Das bedeutet: Falls das Formular z. B. "E-Mail" sendet,
        // und hier "e-mail" steht, wird es trotzdem erkannt.
        foreach ($bereinigt as $name => $wert) {
            if (strcasecmp($name, $feld) === 0) { // strcasecmp = vergleicht ohne
            Groß/Kleinschreibung
                // checken ob das Feld wie eine echte E-Mail aussieht
                $valid = filter_var($wert, FILTER_VALIDATE_EMAIL);
                if ($valid !== false) {
                    $absenderEmail = $valid;
                    break 2; // wir haben eine brauchbare Adresse gefunden -> beide
                    Schleifen verlassen
                }
            }
        }
    }

    // Versuch für einen Absender-Namen (ist optional, aber schön)
    // Wir suchen z. B. nach Feldern wie "Name", "Vorname", "Nachricht" ignorieren
    wir natürlich.
    $moeglicheNamenFelder = [
        'name', 'Name', 'fullname', 'Fullname', 'Vorname', 'Kontaktperson'
    ];

    foreach ($moeglicheNamenFelder as $feld) {
        foreach ($bereinigt as $name => $wert) {
            if (strcasecmp($name, $feld) === 0 && $wert !== '') {
                $absenderName = $wert;
                break 2;
            }
        }
    }

    // 5) PHPMailer vorbereiten
    $mail = new PHPMailer();
    $mail->CharSet = "UTF-8";

    // SMTP Einstellungen
    $mail->isSMTP();
    $mail->Host       = "w020e202.kasserver.com";                // SMTP-Server -
```

```
        hier beispielhaft die Adresse vom IONOS Server
130     $mail->SMTPAuth   = true;                          // SMTP-Authentifizierung
        aktivieren
131     $mail->Username   = "        ";      // SMTP-Benutzername - meist Deine E-Mail
        Adresse
132     $mail->Password   = "               ";                    // SMTP-Passwort
133
134     // --- Variante 1: Port 465 mit SMTPS (implizite TLS-Verschlüsselung) ---
135     $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;    // Verschlüsselung: SMTPS
        (TLS ab Verbindungsbeginn)
136     $mail->Port       = 465;                           // Port 465 - klassische
        SSL/TLS-Verbindung
137
138     // --- Alternative Variante 2: Port 587 mit STARTTLS (explizite TLS-Aushandlung)
        - wenn diese verwendet wird muss Variante 1 // deaktiviert werden ---
139     // $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS; // Verschlüsselung:
        STARTTLS (TLS wird nach Verbindungsaufbau aktiviert)
140     // $mail->Port       = 587;                           // Port 587 - alternative,
        oft in offenen Netzwerken verwendete Verbindung
141
142     // Absender (Diese E-Mail muss beim Provider erlaubt und hinterlegt sein)
143     $mail->setFrom("                          ", "             ");
144
145     // Empfänger (Du selbst)
146     $mail->addAddress("                       ", "        ");
147
148     // OPTIONAL: Antwort-an (Reply-To) setzen,
149     // damit du direkt dem Besucher antworten kannst,
150     // aber nur wenn eine valide Adresse gefunden wurde.
151     if (!empty($absenderEmail)) {
152         $mail->addReplyTo($absenderEmail, $absenderName);
153     }
154
155     // Betreff & Body setzen
156     $mail->Subject = $betreffEmail;
157     $mail->Body    = $inhaltEmail;
158
159     // 6) Senden & Weiterleitung
160     if ($mail->send()) {
161         header("Location: " . $dankeSeite);
162         exit;
163     } else {
164         header("Location: " . $fehlerSeite);
165         exit;
166     }
167 }
168 ?>
169
```