```php
<?php session_start();
header('Content-Type: application/json; charset=utf-8');
require_once('inc/dbConnect.php');
require once('sendgrid/sender.php');
$lang = $ SESSION['lang'] ?? 'ar';
$trans = ($lang == 'ar') ? "ar.php" : "en.php";
require_once('inc/' . $trans);


ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error reporting(E ALL);



date_default_timezone_set('Asia/Amman');
$currDateTime = date("Y-m-d H:i:s");
define('CURRENT_DATE_TIME', $currDateTime);
// START LOGS FILE
$today = date("Y-m");
$directory name = "logs/$today";
if (!file exists($directory name)) {
    mkdir($directory_name, 0777, true);
}
$log_file = $directory_name . '/log-' . date('d') . '.log';
// $req_print_r = print_r($_REQUEST,true);
$req_json_encode = json_encode($_REQUEST);
$txt = "\n ----------------------" . date('H:i') . " ---------------------- \n \n";
$fp = fopen($log file, 'a');
fwrite($fp, $txt . $req json encode);
fclose($fp);
if (!isset($_REQUEST['action']) || empty($_REQUEST['action'])) {
    echo json_encode(array('response' => array(
        'error_code' => "-1",
        'error_msg' => "لم يتم تعيين action",
    )));
    die;
}
$action = $ REQUEST['action'];


$lang = $_SESSION['lang'] ?? 'ar';
$langDB = $lang . "_";
$trans = ($lang == 'ar') ? "ar.php" : "en.php";
$ar assets = ($lang == 'ar') ? "/ar assets" : "";
$dir = ($lang == 'ar') ? "rtl" : "ltr";


switch ($action) {
    case "add_to_fave":
        $product_id = $_REQUEST['product_id'] ?? null;
        $user_id = $_SESSION['user']['id'];
        $stmt = $con->prepare("select * from fav_products where product_id=? and user_id=? ");
        $stmt->bind param("ii", $product id, $user id);
        $stmt->execute();
        $result = $stmt->get_result();
        if (mysqli_num_rows($result) > 0) {
            $stmt = $con->prepare("DELETE FROM fav_products  where product_id=? and user_id=? ");
            $stmt->bind_param("ii", $product_id, $user_id);
            if ($stmt->execute()) {
                echo json_encode(array('response' => array(
                    'error code' => 0,
                    'code' => 200,
                    'error msg' => "Removed ",
                )));
                die;
            } else {
                echo json_encode(array('response' => array(
                    'error_code' => -1,
                    'error_msg' => "Something went wrong ",
                )));
                die;
            }
        } else {
            $stmt = $con->prepare("insert into fav_products ( product_id, user_id) values (?,?)");
            $stmt->bind_param("ii", $product_id, $user_id);
            if ($stmt->execute()) {
                echo json_encode(array('response' => array(
                    'error code' => 0,
                    'code' => 300,
                    'error_msg' => "Added ",
                )));
                die;
            } else {
                echo json_encode(array('response' => array(
                    'error_code' => -1,
                    'error msg' => "Something went wrong ",
                )));
                die;
```

```
            }
        }
        break;


    case "get_ajax_products":
        require_once('inc/functions.php');
        require_once('inc/product_cell.php');
        require_once('inc/solar_system_cell.php');
        $page = $_REQUEST['page'] ?? 1;

        $category id = $ REQUEST['category id'] ?? null;
        $passFilter = [
            'category id' => $category id,
            'with_count' => false,
            'pagination' => true,
            'pagelimit' => 8,
            'pageno' => $page,
        ];


        if (isset($ REQUEST['price from']) && is numeric($ REQUEST['price from'])) {
            $passFilter['price_from'] = $_REQUEST['price_from'];
        }
        if (isset($_REQUEST['price_to']) && is_numeric($_REQUEST['price_to'])) {
            $passFilter['price_to'] = $_REQUEST['price_to'];
        }
        if (isset($_REQUEST['order_by'])) {
            $passFilter['order by'] = $ REQUEST['order by'];
        }
        $products = getProducts($passFilter);
        foreach ($products as $Product) {
            if ($category_id == 12) {
                solar_system_cell($Product);
            } else {
                product_cell($Product);
            }
        }
        break;


    case "change_email":
        $user_id = $_SESSION['user']['id'];
        $email = $_REQUEST['new_email'] ?? null;
        $email = strtolower($email);
        $stmt = $con->prepare("UPDATE users SET   email=?  , status email=0  WHERE id=? ");
        $stmt->bind param("si", $email, $user id);
        if ($stmt->execute()) {
            $_SESSION['user'] = getUserSession($user_id);
            $_SESSION['logged_in'] = true;
            echo json_encode(array('response' => array(
                'error_code' => "0",
                'error_msg' => "Updated ",
            )));
            die;
        }
        break;


    case "update_profile":
        $user_id = $_SESSION['user']['id'];
        $old logo = $ SESSION['user']['logo'];
        $logo = Upload('logo', 'uploads/logos/');
        if ($logo == false) {
            $logo = $old_logo;
        }
        $name = $_REQUEST['name'] ?? null;
        $mobile = $_REQUEST['mobile'] ?? null;
        $city_id = $_REQUEST['city_id'] ?? null;
        $email = $ REQUEST['email'] ?? null;
        $password = $ REQUEST['new password'] ?? null;


        $country_code = $_REQUEST['country_code'] ?? null;


        $email = strtolower($email);
        $mobile = ltrim($mobile, '0');
        if (!empty($password)) {
            $password = hash('sha256', $ REQUEST['new password']);
            $stmt = $con->prepare("UPDATE users SET name=?, mobile=?,country_code=?, email=? ,password=? ,
            logo=? WHERE id=? ");
            $stmt->bind_param("ssssssi", $name, $mobile, $country_code, $email, $password, $logo, $user_id);
        } else {
            $stmt = $con->prepare("UPDATE users SET name=?, mobile=?,country_code=?,  email=? , logo=? WHERE
            id=? ");
            $stmt->bind param("sssssi", $name, $mobile, $country code, $email, $logo, $user id);
        }
        if ($stmt->execute()) {
```

```php
            $stmt = $con->prepare("DELETE FROM user_address  WHERE user_id = ?");
            $stmt->bind_param("i", $user_id);
            $stmt->execute();
            $country id = 1;
            $stmt = $con->prepare("insert into user address ( user id,country id,city id) values (?,?,?)");
            $stmt->bind_param("iii", $user_id, $country_id, $city_id);
            $stmt->execute();
            $_SESSION['user'] = getUserSession($user_id);
            $_SESSION['logged_in'] = true;
            echo json_encode(array('response' => array(
                'error_code' => "0",
                'error msg' => "Updated ",
            )));
            die;
        }
        break;


    case "contact_us_form":
        $country code = $ REQUEST['country code'] ?? null;
        $name = $ REQUEST['name'] ?? null;
        $email = $ REQUEST['email'] ?? null;
        $mobile = $country_code . $_REQUEST['mobile'] ?? null;
        $subject = $_REQUEST['subject'] ?? null;
        $message = $_REQUEST['message'] ?? null;
        $table = "
        <table  border='1'>
            <thead>
            <tr>
                <th>name</th>
                <th>email</th>
                <th>mobile</th>
                <th>subject</th>
                <th>message</th>
            </tr>
            </thead>
            <tbody>
            <tr>
                <td >{$name}</td>
                <td>{$email}</td>
                <td>{$mobile}</td>
                <td>{$subject}</td>
                <td>{$message}</td>
            </tr>
            </tbody>
            </table>

    ";

        echo json_encode(array('response' => array(
            'error_code' => "0",
            'error_msg' => MESSAGE_SENT,
        )));
        SendGrid Sender(array(
            'email msg' => $table,
            'subject' => $subject,
            'to_email' => 'hbahadili@gmail.com',
        ));

        break;


    case "resend verification email":
        $user id = $ SESSION['user']['id'];
        $digits = 4;
        $rand = rand(pow(10, $digits - 1), pow(10, $digits) - 1);
        $stmt = $con->prepare("UPDATE  users SET verification_code=? where id=? ");
        $stmt->bind_param("ii", $rand, $user_id);
        $stmt->execute();
        $subject = "Solar Market :Verification Code";
        $message = "Your email verification code is: " . $rand;
        // sendEmail($ SESSION['user']['email'], $message, $subject);
        SendGrid_Sender(array(
            'email_msg' => $message,
            'subject' => $subject,
            'to_email' => $_SESSION['user']['email'],
        ));
        echo json_encode(array('response' => array(
            'error code' => "0",
            'error msg' => " An email has been sent with a verification code",
        )));
        die;
        break;
    case "verification_email":
        $user_id = $_SESSION['user']['id'];
        $code = $_REQUEST['code'];
        $stmt = $con->prepare("select verification code from users where id=? ");
        $stmt->bind param("i", $user id);
        $stmt->execute();
```

```php
        $result = $stmt->get_result();
        $row = $result->fetch_assoc();
        if ($row['verification_code'] === $code) {
            $stmt = $con->prepare("UPDATE  users SET status email=1 where id=? ");
            $stmt->bind param("i", $user id);
            $stmt->execute();
            $_SESSION['user'] = getUserSession($user_id);
            echo json_encode(array('response' => array(
                'error_code' => "0",
                'error_msg' => " The user account has been successfully activated ",
            )));
            die;
        } else {
            echo json encode(array('response' => array(
                'error_code' => "-1",
                'error_msg' => INVALID_VERIFICATION_CODE,
            )));
            die;
        }
        break;
    case "forget password":
        $email = $ REQUEST['email'];
        $email = strtolower($email);
        $stmt = $con->prepare("select * from users where email=? ");
        $stmt->bind_param("s", $email);
        $stmt->execute();
        $result = $stmt->get_result();
        if (mysqli_num_rows($result) > 0) {
            $row = $result->fetch assoc();


            //            $old_reset_password_token = $row['token'];
            //            if (!empty($old_reset_password_token)) {
            //                $old_token_time = explode('-', $old_reset_password_token)[1];
            //            }
            //

            $token64 = bin2hex(random bytes(46));
            $token32 = bin2hex(random bytes(46));
            $reset_password_token = $token64 . base64_encode(random_string(15)) . $token32;
            $reset_password_token . '-' . time();
            $stmt = $con->prepare("update users set token=? where id =?");
            $stmt->bind_param("si", $reset_password_token, $row['id']);
            $stmt->execute();
            $link = $ SERVER['HTTP HOST'] . '/reset-password.php?reset key=' . $reset password token;
            $subject = "Reset Password";
            $message = "
                Please visit the following link to recover your password
                <br> <a href='$link'>$link </a></br>
                or ignore the mail
                <br>
                This link is valid for 30 minutes only
                ";
            // sendEmail($email, $message, $subject);
            SendGrid Sender(array(
                'email_msg' => $message,
                'subject' => $subject,
                'to_email' => $email,
            ));
            echo json_encode(array('response' => array(
                'error_code' => "0",
                'error msg' => "An email has been sent to your email containing a password recovery link",
            )));
            die;
        } else {
            echo json_encode(array('response' => array(
                'error_code' => "-1",
                'error_msg' => "Email not found",
            )));
            die;
        }
        break;
    case "login_account":
        $email = $_REQUEST['email'];
        $email = strtolower($email);
        $password = hash('sha256', $_REQUEST['password']);

        $stmt = $con->prepare("select * from users where email=? and password = ?");
        $stmt->bind param("ss", $email, $password);
        $stmt->execute();
        $result = $stmt->get_result();
        if (mysqli_num_rows($result) > 0) {
            $row = $result->fetch_assoc();
            $_SESSION['user'] = getUserSession($row['id']);
            $_SESSION['logged_in'] = true;


            /*    Disable Email  Verification
            $is_verified = ($_SESSION['user']['status_email'] == 1);
```

```
355
356             if (!$is_verified) {
357
358                 $digits = 4;
359                 $rand = rand(pow(10, $digits - 1), pow(10, $digits) - 1);
360
361                 $stmt = $con->prepare("UPDATE  users SET verification_code=? where id=? ");
362                 $stmt->bind_param("ii", $rand, $row['id']);
363                 $stmt->execute();
364
365                 $subject = "Solar Market :Verification Code";
366                 $message = "Your email verification code is: " . $rand;
367                 SendGrid Sender(array(
368                     'email msg' => $message,
369                     'subject' => $subject,
370                     'to_email' => $email,
371                 ));
372             }
373
374
375             */
376
377             $is_verified =  true ;
378             echo json_encode(array('response' => array(
379                 'error_code' => "0",
380                 'type' => $row['type'] == 'company',
381                 'is_verified' => $is_verified,
382                 'is_subscribed' => (($_SESSION['user']['subscription'] == 1) && (strtotime($_SESSION['user'][
                    'exp date']) >= time())),
383                 'error msg' => "User account created successfully ",
384             )));
385             die;
386         } else {
387             echo json_encode(array('response' => array(
388                 'error_code' => "-1",
389                 'error_msg' => USERNAME_OR_PASSWORD_IS_INCORRECT,
390
391             )));
392             die;
393         }
394         $stmt->close();
395         break;
396     case "check_email_forget_pass":
397         $email = $_REQUEST['email'];
398         $email = strtolower($email);
399         $stmt = $con->prepare("select * from users where email=? ");
400         $stmt->bind param("s", $email);
401         $stmt->execute();
402         $result = $stmt->get_result();
403         echo json_encode((mysqli_num_rows($result) > 0));
404         break;
405
406
407     case "check change email":
408         $user id = $ SESSION['user']['id'];
409         $email = $_REQUEST['email'];
410         $email = strtolower($email);
411         $stmt = $con->prepare("select * from users where email=?  and id  !=?");
412         $stmt->bind_param("si", $email, $user_id);
413         $stmt->execute();
414         $result = $stmt->get_result();
415         echo json encode((mysqli num rows($result) <= 0));
416         break;
417
418
419     case "check_tel":
420         $tel = $_REQUEST['tel'];
421         $tel = ltrim($tel, '0');
422         $stmt = $con->prepare("select * from users where tel=? ");
423         $stmt->bind param("s", $tel);
424         $stmt->execute();
425         $result = $stmt->get result();
426         echo json_encode((mysqli_num_rows($result) <= 0));
427         break;
428     case "check_old_password":
429         $user_id = $_SESSION['user']['id'];
430         $password = hash('sha256', $_REQUEST['old_password']);
431         $stmt = $con->prepare("select * from users where password=?  and id  =?");
432         $stmt->bind param("si", $password, $user id);
433         $stmt->execute();
434         $result = $stmt->get_result();
435         echo json_encode((mysqli_num_rows($result) > 0));
436         break;
437     case "check_mobile":
438         $country_code = $_REQUEST['country_code'] ?? '+964';
439         if (empty($country_code))
440             $country code = '+964';
441         $mobile = $ REQUEST['mobile'] ?? null;
442         $mobile = ltrim($mobile, '0');
```

```
443         if (isset($_SESSION['user'])) {
444             $user_id = $_SESSION['user']['id'];
445             $stmt = $con->prepare("select * from users where country_code=? AND mobile=?  AND  id!=?");
446             $stmt->bind param("ssi", $country code, $mobile, $user id);
447         } else {
448             $stmt = $con->prepare("select * from users where country_code=? AND mobile=?");
449             $stmt->bind_param("ss", $country_code, $mobile);
450         }
451         $stmt->execute();
452         $result = $stmt->get_result();
453         echo json_encode((mysqli_num_rows($result) <= 0));
454         break;
455     case "check email":
456         $email = $ REQUEST['email'] ?? null;
457         $email = strtolower($email);
458         if (isset($_SESSION['user'])) {
459             $user_id = $_SESSION['user']['id'];
460             $stmt = $con->prepare("select * from users where email=?   AND  id!=?");
461             $stmt->bind_param("si", $email, $user_id);
462         } else {
463             $stmt = $con->prepare("select * from users where email=? ");
464             $stmt->bind param("s", $email);
465         }
466         $stmt->execute();
467         $result = $stmt->get_result();
468         echo json_encode((mysqli_num_rows($result) <= 0));
469         break;
470     case "create_account":
471         $name = $ REQUEST['name'] ?? null;
472         $country code = $ REQUEST['country code'] ?? null;
473         $mobile = $_REQUEST['mobile'] ?? null;
474         $country_id = $_REQUEST['country_id'] ?? 1;
475         $city_id = $_REQUEST['city_id'] ?? null;
476         $email = $_REQUEST['email'] ?? null;
477         $password = $_REQUEST['password'] ?? null;
478         $password_confirmation = $_REQUEST['password_confirmation'] ?? null;
479         $email = strtolower($email);
480         $mobile = ltrim($mobile, '0');
481         if ($password != $password confirmation) {
482             echo json_encode(array('response' => array(
483                 'error_code' => "-1",
484                 'error_msg' => "Password does not match",
485             )));
486             die;
487         }
488         $password = hash('sha256', $ REQUEST['password']);
489         $type = 'user';
490         $stmt = $con->prepare("insert into users (name, type, email, country_code, mobile, password) values
            (?,?,?,?,?,?)");
491         $stmt->bind_param("ssssss", $name, $type, $email, $country_code, $mobile, $password);
492         if ($stmt->execute()) {
493             $user_id = $stmt->insert_id;
494             $stmt = $con->prepare("insert into user_address (user_id, country_id, city_id) values (?,?,?)");
495             $stmt->bind param("iii", $user id, $country id, $city id);
496             $stmt->execute();
497             $_SESSION['user'] = getUserSession($user_id);
498             $_SESSION['logged_in'] = true;
499             echo json_encode(array('response' => array(
500                 'error_code' => "0",
501                 'error_msg' => " User account created successfully",
502             )));
503             die;
504         }
505         break;
506     case "create_merchant_account":
507         $digits = 4;
508         $rand = rand(pow(10, $digits - 1), pow(10, $digits) - 1);
509
510         //Verification Code
511         //Status Email
512         //Status Mobile
513
514
515     /*
516         if (!empty($website))
517         {
518             if (strpos($website, 'http') === false)
519             {
520                 $website = "https://" . $website;
521             }
522         }
523
524         if (!empty($youtube))
525         {
526             if (strpos($youtube, 'http') === false)
527             {
528                 $youtube = "https://" . $youtube;
529             }
530         }
```

```php

        if (!empty($facebook))
        {
            if (strpos($facebook, 'http') === false)
            {
                $facebook = "https://" . $facebook;
            }
        }

        if (!empty($instagram))
        {
            if (strpos($instagram, 'http') === false)
            {
                $instagram = "https://" . $instagram;
            }
        }
    */

        $logo = $_REQUEST['logo'] ?? null;
        $name = $_REQUEST['name'] ?? null;
        $alias = $ REQUEST['alias'] ?? null;
        $text = $ REQUEST['text'] ?? null;
        $country_id = $_REQUEST['country_id'] ?? 1;
        $city_id = $_REQUEST['city_id'] ?? null;

        $email = strtolower($_REQUEST['email']) ?? null;
        $status_email = 1;
        $username = ""; // strtolower($_REQUEST['username']) ?? null;
  //     $username confirmation = strtolower($ REQUEST['confirm username']) ?? null;
/*
        if ($username != $username_confirmation)
        {
            echo json_encode(array('response' => array(
                'error_code' => "-1",
                'error_msg' => "Username does not match",
            )));
            die;
        }
*/
        $password = $_REQUEST['password'] ?? null;
        $password_confirmation = $_REQUEST['password_confirmation'] ?? null;

        if ($password != $password_confirmation)
        {
            echo json_encode(array('response' => array(
                'error code' => "-1",
                'error msg' => "Password does not match",
            )));
            die;
        }


        $mobile = $_REQUEST['mobile'] ?? null;
        $tel = $ REQUEST['tel'] ?? null;
        $tel code = $ REQUEST['tel code'] ?? 1;
        $country_code = $_REQUEST['country_code'] ?? null; // '+964';

        $website = $_REQUEST['website'] ?? null;
        $youtube = $_REQUEST['youtube'] ?? null;
        $facebook = $_REQUEST['facebook'] ?? null;
        $twitter = $_REQUEST['twitter'] ?? null;
        $instagram = $ REQUEST['instagram'] ?? null;
        $tiktok = $ REQUEST['tiktok'] ?? null;

        $info = $_REQUEST['info'] ?? null;
        $verification_code = $rand;

        $mobile = ltrim($mobile, '0');
        $tel = ltrim($tel, '0');

        $stmt = $con->prepare("select * from users where email=? ");
        $stmt->bind param("s", $email);
        $stmt->execute();
        $result = $stmt->get_result();

        if (mysqli_num_rows($result) > 0)
        {
            $row = $result->fetch_assoc();
            echo json encode(array('response' => array(
                'error code' => "1",
                'error_msg' => EMAIL_ALREADY_USED,
            )));
            die;

        /*    Disable Email  Verification

            if ($row['status email'] == 1)
            {
                echo json_encode(array('response' => array(
```

```
620                     'error_code' => "1",
621                     'error_msg' => EMAIL_ALREADY_USED,
622                 )));
623                 die;
624             }
625             else
626             {
627                 $stmt = $con->prepare("delete  from users where id=? ");
628                 $stmt->bind_param("i", $row['id']);
629                 $stmt->execute();
630 
631                 $stmt = $con->prepare("delete  from user address where user id=? ");
632                 $stmt->bind param("i", $row['id']);
633                 $stmt->execute();
634             }
635 
636             */
637         }
638 
639         $password = hash('sha256', $_REQUEST['password']);
640         $type = 'company';
641         $logo = Upload('logo', 'uploads/logos/');
642         if ($logo == false)
643         {
644             $logo = null;
645         }
646 
647 
648         /*   Disable Email  Verification
649 
650         $subject = "Solar Market :Verification Code";
651         $message = "Your email verification code is: " . $rand;
652         // sendEmail($email, $message, $subject);
653         SendGrid_Sender(array(
654             'email_msg' => $message,
655             'subject' => $subject,
656             'to email' => $email,
657         ));
658 
659 
660         */
661 
662         $stmt = $con->prepare
663         ("insert into users (type, mobile, tel, website, youtube, facebook, twitter, instagram, tiktok,
          username, password, email, name, country code, tel code, logo, status email, country id, city id, alias,
          comments)
664         values (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
665         $stmt->bind_param("sssssssssssssssiiiss", $type, $mobile, $tel, $website, $youtube, $facebook, $twitter,
           $instagram, $tiktok, $username, $password, $email, $name, $country_code, $tel_code, $logo, $status_email
          , $country_id, $city_id, $alias, $info);
666         if ($stmt->execute())
667         {
668             $user_id = $stmt->insert_id;
669             $stmt = $con->prepare("insert into user address (user id, country id, city id, text) values
              (?,?,?,?)");
670             $stmt->bind_param("iiis", $user_id, $country_id, $city_id, $text);
671             $stmt->execute();
672 
673             // $_SESSION['user'] = getUserSession($user_id);
674             // $_SESSION['logged_in'] = true;
675 
676             echo json encode(array('response' => array(
677                 'error code' => "0",
678                 'error msg' => " User account created successfully ",
679             )));
680             die;
681         }
682         break;
683 }
684 
685 function getUserSession($uid)
686 {
687     global $con;
688     $userArray = [];
689     $stmt = $con->prepare("select * from users where id=?");
690     $stmt->bind_param("i", $uid);
691     $stmt->execute();
692     $result = $stmt->get_result();
693     $row = $result->fetch assoc();
694     $userArray = $row;
695     $userArray['address'] = getUserAddress($uid);
696     // $userArray['subscription'] = getUserSubscription($uid); //  ملغي
697 
698     $stmt->close();
699     return $userArray;
700 }
701 
702 function getUserSubscription($uid) //  ملغي
703 {
```

```php
    global $con;
    $stmt2 = $con->prepare("select * from user_subscription where user_id=? and `expiry_date` > now() ");
    $stmt2->bind_param("i", $uid);
    $stmt2->execute();
    $result2 = $stmt2->get result();
    if (mysqli_num_rows($result2) <= 0) {
        return [
            'is_subscribed' => false
        ];
    } else {
        $row2 = $result2->fetch_assoc();
        return [
            'is subscribed' => true,
            'expiry date' => $row2['expiry date'],
            'start_date' => $row2['start_date'],
            'validity' => $row2['validity'],
        ];
    }
}

function getUserAddress($uid)
{
    global $con;
    $UserAddress = [];
    $stmt2 = $con->prepare("select * from user_address where user_id=?");
    $stmt2->bind_param("i", $uid);
    $stmt2->execute();
    $result2 = $stmt2->get_result();
    while ($row2 = $result2->fetch assoc()) {
        array push($UserAddress, $row2);
    }
    return $UserAddress;
}

function random_string($length)
{
    $key = '';
    $keys = array merge(range(0, 9), range('a', 'z'), range('A', 'Z'));
    for ($i = 0; $i < $length; $i++) {
        $key .= $keys[array_rand($keys)];
    }
    return $key;
}

function sendEmail($to, $message, $subject)
{
    $headers = 'From:muradabujamous@gmail.com' . "\r\n" .
        'Reply-To:muradabujamous@gmail.com' . "\r\n" .
        'X-Mailer: PHP/' . phpversion();
    $headers .= "MIME-Version: 1.0\r\n";
    $headers .= "Content-Type: text/html; charset=ISO-8859-1\r\n";
    (mail($to, $subject, $message, $headers));
}

function Upload($fname, $path, $old = null)
{
    $Allowed_ext = array("JPG", "jpg", "jpeg", "png", "JPEG", "PNG", "JFIF", "jfif", "pdf", "PDF");
    if (isset($_FILES[$fname])) {
        if ($_FILES[$fname]["name"] != '') {
            $temp = explode('.', $_FILES[$fname]["name"]);
            $ext = end($temp);
            if (in array($ext, $Allowed ext)) {
                $name = md5(rand()) . '.' . $ext;
                $path = $path . $name;
                move_uploaded_file($_FILES[$fname]["tmp_name"], $path);
                return $path;
            } else {
                echo json_encode(array(
                    'error_code' => "1",
                    'error_msg' => "Image extension is not allowed",
                ));
                exit;
                die();
            }
        } else {
            return false;
        }
    } else {
        return false;
    }
}
```