

```

1  <?php session_start();
2  header('Content-Type: application/json; charset=utf-8');
3  require_once('inc/dbConnect.php');
4  require_once('sendgrid/sender.php');
5  $lang = $_SESSION['lang'] ?? 'ar';
6  $strans = ($lang == 'ar') ? "ar.php" : "en.php";
7  require_once('inc/' . $strans);
8
9
10 ini_set('display_errors', 1);
11 ini_set('display_startup_errors', 1);
12 error_reporting(E_ALL);
13
14
15
16 date_default_timezone_set('Asia/Amman');
17 $currDateTime = date("Y-m-d H:i:s");
18 define('CURRENT_DATE_TIME', $currDateTime);
19 // START LOGS FILE
20 $today = date("Y-m");
21 $directory_name = "logs/$today";
22 if (!file_exists($directory_name)) {
23     mkdir($directory_name, 0777, true);
24 }
25 $log_file = $directory_name . '/log-' . date('d') . '.log';
26 // $req_print_r = print_r($_REQUEST, true);
27 $req_json_encode = json_encode($_REQUEST);
28 $txt = "\n -----" . date('H:i') . " ----- \n \n";
29 $fp = fopen($log_file, 'a');
30 fwrite($fp, $txt . $req_json_encode);
31 fclose($fp);
32 if (!isset($_REQUEST['action']) || empty($_REQUEST['action'])) {
33     echo json_encode(array('response' => array(
34         'error_code' => "-1",
35         'error_msg' => "لم يتم تعيين action",
36     )));
37     die;
38 }
39 $action = $_REQUEST['action'];
40
41
42 $lang = $_SESSION['lang'] ?? 'ar';
43 $langDB = $lang . " ";
44 $strans = ($lang == 'ar') ? "ar.php" : "en.php";
45 $ar_assets = ($lang == 'ar') ? "/ar assets" : "";
46 $dir = ($lang == 'ar') ? "rtl" : "ltr";
47
48
49 switch ($action) {
50     case "add_to_fave":
51         $product_id = $_REQUEST['product_id'] ?? null;
52         $user_id = $_SESSION['user']['id'];
53         $stmt = $con->prepare("select * from fav_products where product_id=? and user_id=? ");
54         $stmt->bind_param("ii", $product_id, $user_id);
55         $stmt->execute();
56         $result = $stmt->get_result();
57         if (mysqli_num_rows($result) > 0) {
58             $stmt = $con->prepare("DELETE FROM fav_products where product_id=? and user_id=? ");
59             $stmt->bind_param("ii", $product_id, $user_id);
60             if ($stmt->execute()) {
61                 echo json_encode(array('response' => array(
62                     'error_code' => 0,
63                     'code' => 200,
64                     'error_msg' => "Removed ",
65                 )));
66                 die;
67             } else {
68                 echo json_encode(array('response' => array(
69                     'error_code' => -1,
70                     'error_msg' => "Something went wrong ",
71                 )));
72                 die;
73             }
74         } else {
75             $stmt = $con->prepare("insert into fav_products ( product_id, user_id) values (?,?)");
76             $stmt->bind_param("ii", $product_id, $user_id);
77             if ($stmt->execute()) {
78                 echo json_encode(array('response' => array(
79                     'error code' => 0,
80                     'code' => 300,
81                     'error_msg' => "Added ",
82                 )));
83                 die;
84             } else {
85                 echo json_encode(array('response' => array(
86                     'error_code' => -1,
87                     'error_msg' => "Something went wrong ",
88                 )));
89                 die;

```

```

90         }
91     }
92     break;
93
94
95 case "get_ajax_products":
96     require_once('inc/functions.php');
97     require_once('inc/product_cell.php');
98     require_once('inc/solar_system_cell.php');
99     $page = $_REQUEST['page'] ?? 1;
100
101     $category_id = $_REQUEST['category_id'] ?? null;
102     $passFilter = [
103         'category_id' => $category_id,
104         'with_count' => false,
105         'pagination' => true,
106         'pagelimit' => 8,
107         'pageno' => $page,
108     ];
109
110
111     if (isset($_REQUEST['price_from']) && is_numeric($_REQUEST['price_from'])) {
112         $passFilter['price_from'] = $_REQUEST['price_from'];
113     }
114     if (isset($_REQUEST['price_to']) && is_numeric($_REQUEST['price_to'])) {
115         $passFilter['price_to'] = $_REQUEST['price_to'];
116     }
117     if (isset($_REQUEST['order_by'])) {
118         $passFilter['order_by'] = $_REQUEST['order_by'];
119     }
120     $products = getProducts($passFilter);
121     foreach ($products as $Product) {
122         if ($category_id == 12) {
123             solar_system_cell($Product);
124         } else {
125             product_cell($Product);
126         }
127     }
128     break;
129
130
131 case "change_email":
132     $user_id = $_SESSION['user']['id'];
133     $email = $_REQUEST['new_email'] ?? null;
134     $email = strtolower($email);
135     $stmt = $con->prepare("UPDATE users SET email=? , status_email=0 WHERE id=? ");
136     $stmt->bind_param("si", $email, $user_id);
137     if ($stmt->execute()) {
138         $_SESSION['user'] = getUserSession($user_id);
139         $_SESSION['logged_in'] = true;
140         echo json_encode(array('response' => array(
141             'error_code' => "0",
142             'error_msg' => "Updated ",
143         )));
144         die;
145     }
146     break;
147
148
149 case "update_profile":
150     $user_id = $_SESSION['user']['id'];
151     $old_logo = $_SESSION['user']['logo'];
152     $logo = Upload('logo', 'uploads/logos/');
153     if ($logo == false) {
154         $logo = $old_logo;
155     }
156     $name = $_REQUEST['name'] ?? null;
157     $mobile = $_REQUEST['mobile'] ?? null;
158     $city_id = $_REQUEST['city_id'] ?? null;
159     $email = $_REQUEST['email'] ?? null;
160     $password = $_REQUEST['new_password'] ?? null;
161
162
163     $country_code = $_REQUEST['country_code'] ?? null;
164
165
166     $email = strtolower($email);
167     $mobile = ltrim($mobile, '0');
168     if (!empty($password)) {
169         $password = hash('sha256', $_REQUEST['new_password']);
170         $stmt = $con->prepare("UPDATE users SET name=?, mobile=?,country_code=?, email=? ,password=? ,
171             logo=? WHERE id=? ");
172         $stmt->bind_param("sssssi", $name, $mobile, $country_code, $email, $password, $logo, $user_id);
173     } else {
174         $stmt = $con->prepare("UPDATE users SET name=?, mobile=?,country_code=?, email=? , logo=? WHERE
175             id=? ");
176         $stmt->bind_param("sssssi", $name, $mobile, $country_code, $email, $logo, $user_id);
177     }
178     if ($stmt->execute()) {

```

```

177     $stmt = $con->prepare("DELETE FROM user_address WHERE user_id = ?");
178     $stmt->bind_param("i", $user_id);
179     $stmt->execute();
180     $country_id = 1;
181     $stmt = $con->prepare("insert into user address ( user id,country id,city id) values (?,?,?)");
182     $stmt->bind_param("iii", $user_id, $country_id, $city_id);
183     $stmt->execute();
184     $_SESSION['user'] = getUserSession($user_id);
185     $_SESSION['logged_in'] = true;
186     echo json_encode(array('response' => array(
187         'error_code' => "0",
188         'error msg' => "Updated ",
189     )));
190     die;
191 }
192 break;
193
194
195 case "contact_us_form":
196     $country_code = $ REQUEST['country code'] ?? null;
197     $name = $ REQUEST['name'] ?? null;
198     $email = $ REQUEST['email'] ?? null;
199     $mobile = $country_code . $ REQUEST['mobile'] ?? null;
200     $subject = $ REQUEST['subject'] ?? null;
201     $message = $ REQUEST['message'] ?? null;
202     $table = "
203 <table border='1'>
204     <thead>
205     <tr>
206         <th>name</th>
207         <th>email</th>
208         <th>mobile</th>
209         <th>subject</th>
210         <th>message</th>
211     </tr>
212     </thead>
213     <tbody>
214     <tr>
215         <td>{$name}</td>
216         <td>{$email}</td>
217         <td>{$mobile}</td>
218         <td>{$subject}</td>
219         <td>{$message}</td>
220     </tr>
221     </tbody>
222     </table>
223 ";
224
225
226     echo json_encode(array('response' => array(
227         'error_code' => "0",
228         'error_msg' => MESSAGE_SENT,
229     )));
230     SendGrid Sender(array(
231         'email msg' => $table,
232         'subject' => $subject,
233         'to_email' => 'hbahadili@gmail.com',
234     ));
235
236     break;
237
238
239 case "resend verification email":
240     $user_id = $_SESSION['user']['id'];
241     $digits = 4;
242     $rand = rand(pow(10, $digits - 1), pow(10, $digits) - 1);
243     $stmt = $con->prepare("UPDATE users SET verification_code=? where id=? ");
244     $stmt->bind_param("ii", $rand, $user_id);
245     $stmt->execute();
246     $subject = "Solar Market :Verification Code";
247     $message = "Your email verification code is: " . $rand;
248     // sendEmail($_SESSION['user']['email'], $message, $subject);
249     SendGrid Sender(array(
250         'email msg' => $message,
251         'subject' => $subject,
252         'to_email' => $_SESSION['user']['email'],
253     ));
254     echo json_encode(array('response' => array(
255         'error code' => "0",
256         'error msg' => " An email has been sent with a verification code",
257     )));
258     die;
259     break;
260
261 case "verification_email":
262     $user_id = $_SESSION['user']['id'];
263     $code = $_REQUEST['code'];
264     $stmt = $con->prepare("select verification code from users where id=? ");
265     $stmt->bind_param("i", $user_id);
266     $stmt->execute();

```

```

266 $result = $stmt->get_result();
267 $row = $result->fetch_assoc();
268 if ($row['verification_code'] === $code) {
269     $stmt = $con->prepare("UPDATE users SET status email=1 where id=? ");
270     $stmt->bind_param("i", $user_id);
271     $stmt->execute();
272     $_SESSION['user'] = getUserSession($user_id);
273     echo json_encode(array('response' => array(
274         'error_code' => "0",
275         'error_msg' => " The user account has been successfully activated ",
276     )));
277     die;
278 } else {
279     echo json_encode(array('response' => array(
280         'error_code' => "-1",
281         'error_msg' => INVALID_VERIFICATION_CODE,
282     )));
283     die;
284 }
285 break;
286 case "forget password":
287     $email = $ REQUEST['email'];
288     $email = strtolower($email);
289     $stmt = $con->prepare("select * from users where email=? ");
290     $stmt->bind_param("s", $email);
291     $stmt->execute();
292     $result = $stmt->get_result();
293     if (mysqli_num_rows($result) > 0) {
294         $row = $result->fetch_assoc();
295
296         // $old_reset_password_token = $row['token'];
297         // if (!empty($old_reset_password_token)) {
298         //     $old_token_time = explode('-', $old_reset_password_token)[1];
299         // }
300         //
301         //
302
303         $token64 = bin2hex(random_bytes(46));
304         $token32 = bin2hex(random_bytes(46));
305         $reset_password_token = $token64 . base64_encode(random_string(15)) . $token32;
306         $reset_password_token . '-' . time();
307         $stmt = $con->prepare("update users set token=? where id =?");
308         $stmt->bind_param("si", $reset_password_token, $row['id']);
309         $stmt->execute();
310         $link = $ SERVER['HTTP_HOST'] . '/reset-password.php?reset key=' . $reset_password_token;
311         $subject = "Reset Password";
312         $message = "
313             Please visit the following link to recover your password
314             <br> <a href='$link'>$link </a></br>
315             or ignore the mail
316             <br>
317             This link is valid for 30 minutes only
318             ";
319         // sendEmail($email, $message, $subject);
320         SendGrid Sender(array(
321             'email_msg' => $message,
322             'subject' => $subject,
323             'to_email' => $email,
324         ));
325         echo json_encode(array('response' => array(
326             'error_code' => "0",
327             'error_msg' => "An email has been sent to your email containing a password recovery link",
328         )));
329         die;
330     } else {
331         echo json_encode(array('response' => array(
332             'error_code' => "-1",
333             'error_msg' => "Email not found",
334         )));
335         die;
336     }
337     break;
338 case "login_account":
339     $email = $ REQUEST['email'];
340     $email = strtolower($email);
341     $password = hash('sha256', $_REQUEST['password']);
342
343     $stmt = $con->prepare("select * from users where email=? and password = ?");
344     $stmt->bind_param("ss", $email, $password);
345     $stmt->execute();
346     $result = $stmt->get_result();
347     if (mysqli_num_rows($result) > 0) {
348         $row = $result->fetch_assoc();
349         $_SESSION['user'] = getUserSession($row['id']);
350         $_SESSION['logged_in'] = true;
351
352         /* Disable Email Verification
353         $is_verified = ($_SESSION['user']['status_email'] == 1);
354

```

```

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']['
383         'exp date']) >= time())),
384         'error msg' => "User account created successfully ",
385     ));
386     die;
387 } else {
388     echo json_encode(array('response' => array(
389         'error_code' => "-1",
390         'error_msg' => USERNAME_OR_PASSWORD_IS_INCORRECT,
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', $password);
489     $type = 'user';
490     $stmt = $con->prepare("insert into users (name, type, email, country_code, mobile, password) values
491     (?, ?, ?, ?, ?, ?)");
492     $stmt->bind_param("ssssss", $name, $type, $email, $country_code, $mobile, $password);
493     if ($stmt->execute()) {
494         $user_id = $stmt->insert_id;
495         $stmt = $con->prepare("insert into user_address (user_id, country_id, city_id) values (?, ?, ?)");
496         $stmt->bind_param("iii", $user_id, $country_id, $city_id);
497         $stmt->execute();
498         $_SESSION['user'] = getUserSession($user_id);
499         $_SESSION['logged_in'] = true;
500         echo json_encode(array('response' => array(
501             'error_code' => "0",
502             'error_msg' => " User account created successfully",
503         )));
504         die;
505     }
506     break;
507 case "create_merchant_account":
508     $digits = 4;
509     $rand = rand(pow(10, $digits - 1), pow(10, $digits) - 1);
510
511     //Verification Code
512     //Status Email
513     //Status Mobile
514
515 /*
516     if (!empty($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     }

```

```

531
532     if (!empty($facebook))
533     {
534         if (strpos($facebook, 'http') === false)
535         {
536             $facebook = "https://" . $facebook;
537         }
538     }
539
540     if (!empty($instagram))
541     {
542         if (strpos($instagram, 'http') === false)
543         {
544             $instagram = "https://" . $instagram;
545         }
546     }
547 */
548
549     $logo = $_REQUEST['logo'] ?? null;
550     $name = $_REQUEST['name'] ?? null;
551     $alias = $_REQUEST['alias'] ?? null;
552     $text = $_REQUEST['text'] ?? null;
553     $country_id = $_REQUEST['country_id'] ?? 1;
554     $city_id = $_REQUEST['city_id'] ?? null;
555
556     $email = strtolower($_REQUEST['email']) ?? null;
557     $status_email = 1;
558     $username = ""; // strtolower($_REQUEST['username']) ?? null;
559     // $username_confirmation = strtolower($_REQUEST['confirm username']) ?? null;
560 */
561     if ($username != $username_confirmation)
562     {
563         echo json_encode(array('response' => array(
564             'error_code' => "-1",
565             'error_msg' => "Username does not match",
566         )));
567         die;
568     }
569 */
570     $password = $_REQUEST['password'] ?? null;
571     $password_confirmation = $_REQUEST['password_confirmation'] ?? null;
572
573     if ($password != $password_confirmation)
574     {
575         echo json_encode(array('response' => array(
576             'error code' => "-1",
577             'error msg' => "Password does not match",
578         )));
579         die;
580     }
581
582
583     $mobile = $_REQUEST['mobile'] ?? null;
584     $tel = $_REQUEST['tel'] ?? null;
585     $tel code = $_REQUEST['tel code'] ?? 1;
586     $country_code = $_REQUEST['country_code'] ?? null; // '+964';
587
588     $website = $_REQUEST['website'] ?? null;
589     $youtube = $_REQUEST['youtube'] ?? null;
590     $facebook = $_REQUEST['facebook'] ?? null;
591     $twitter = $_REQUEST['twitter'] ?? null;
592     $instagram = $_REQUEST['instagram'] ?? null;
593     $tiktok = $_REQUEST['tiktok'] ?? null;
594
595     $info = $_REQUEST['info'] ?? null;
596     $verification_code = $rand;
597
598     $mobile = ltrim($mobile, '0');
599     $tel = ltrim($tel, '0');
600
601     $stmt = $con->prepare("select * from users where email=? ");
602     $stmt->bind_param("s", $email);
603     $stmt->execute();
604     $result = $stmt->get_result();
605
606     if (mysqli_num_rows($result) > 0)
607     {
608         $row = $result->fetch_assoc();
609         echo json_encode(array('response' => array(
610             'error code' => "1",
611             'error_msg' => EMAIL_ALREADY_USED,
612         )));
613         die;
614     }
615
616     /* Disable Email Verification
617
618     if ($row['status_email'] == 1)
619     {
620         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,
664 username, password, email, name, country code, tel code, logo, status email, country id, city id, alias,
665 comments)
666 values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
667 $stmt->bind_param("ssssssssssssssssssssss", $type, $mobile, $tel, $website, $youtube, $facebook, $twitter,
668 $instagram, $tiktok, $username, $password, $email, $name, $country_code, $tel_code, $logo, $status_email,
669 $country_id, $city_id, $alias, $info);
670 if ($stmt->execute())
671 {
672     $user_id = $stmt->insert_id;
673     $stmt = $con->prepare("insert into user address (user id, country id, city id, text) values
674     (?, ?, ?, ?)");
675     $stmt->bind_param("iiii", $user_id, $country_id, $city_id, $text);
676     $stmt->execute();
677
678     // $_SESSION['user'] = getUserSession($user_id);
679     // $_SESSION['logged_in'] = true;
680
681     echo json_encode(array('response' => array(
682         'error code' => "0",
683         'error msg' => " User account created successfully ",
684     )));
685     die;
686 }
687 break;
688 }
689
690 function getUserSession($uid)
691 {
692     global $con;
693     $userArray = [];
694     $stmt = $con->prepare("select * from users where id=?");
695     $stmt->bind_param("i", $uid);
696     $stmt->execute();
697     $result = $stmt->get_result();
698     $row = $result->fetch_assoc();
699     $userArray = $row;
700     $userArray['address'] = getUserAddress($uid);
701     // $userArray['subscription'] = getUserSubscription($uid); // ملفي
702
703     $stmt->close();
704     return $userArray;
705 }
706
707 function getUserSubscription($uid) // ملفي
708 {

```



```

704     global $con;
705     $stmt2 = $con->prepare("select * from user_subscription where user_id=? and `expiry_date` > now() ");
706     $stmt2->bind_param("i", $uid);
707     $stmt2->execute();
708     $result2 = $stmt2->get_result();
709     if (mysqli_num_rows($result2) <= 0) {
710         return [
711             'is_subscribed' => false
712         ];
713     } else {
714         $row2 = $result2->fetch_assoc();
715         return [
716             'is_subscribed' => true,
717             'expiry_date' => $row2['expiry_date'],
718             'start_date' => $row2['start_date'],
719             'validity' => $row2['validity'],
720         ];
721     }
722 }
723
724 function getUserAddress($uid)
725 {
726     global $con;
727     $UserAddress = [];
728     $stmt2 = $con->prepare("select * from user_address where user_id=?");
729     $stmt2->bind_param("i", $uid);
730     $stmt2->execute();
731     $result2 = $stmt2->get_result();
732     while ($row2 = $result2->fetch_assoc()) {
733         array_push($UserAddress, $row2);
734     }
735     return $UserAddress;
736 }
737
738 function random_string($length)
739 {
740     $key = '';
741     $keys = array_merge(range(0, 9), range('a', 'z'), range('A', 'Z'));
742     for ($i = 0; $i < $length; $i++) {
743         $key .= $keys[array_rand($keys)];
744     }
745     return $key;
746 }
747
748 function sendEmail($to, $message, $subject)
749 {
750     $headers = 'From:muradabujamous@gmail.com' . "\r\n" .
751         'Reply-To:muradabujamous@gmail.com' . "\r\n" .
752         'X-Mailer: PHP/' . phpversion();
753     $headers .= "MIME-Version: 1.0\r\n";
754     $headers .= "Content-Type: text/html; charset=ISO-8859-1\r\n";
755     (mail($to, $subject, $message, $headers));
756 }
757
758 function Upload($fname, $path, $old = null)
759 {
760     $Allowed_ext = array("JPG", "jpg", "jpeg", "png", "JPEG", "PNG", "JFIF", "jif", "pdf", "PDF");
761     if (isset($_FILES[$fname])) {
762         if ($_FILES[$fname]["name"] != '') {
763             $temp = explode('.', $_FILES[$fname]["name"]);
764             $ext = end($temp);
765             if (in_array($ext, $Allowed_ext)) {
766                 $name = md5(rand()) . '.' . $ext;
767                 $path = $path . $name;
768                 move_uploaded_file($_FILES[$fname]["tmp_name"], $path);
769                 return $path;
770             } else {
771                 echo json_encode(array(
772                     'error_code' => "1",
773                     'error_msg' => "Image extension is not allowed",
774                 ));
775                 exit;
776                 die();
777             }
778         } else {
779             return false;
780         }
781     } else {
782         return false;
783     }
784 }
785

```