

File - C:\Users\Tevin\Desktop\Development\Homework\CS-490\api\common.php

```
1 <?php
2
3 if( str_compare( $_SERVER['SERVER_SOFTWARE'], 'Apache') ) {
4     error_reporting( error_reporting() & ~E_NOTICE );
5 }
6
7 function postRequest($url, $headers, $fields) {
8     $ch = curl_init();
9     curl_setopt($ch, CURLOPT_URL, $url);
10    curl_setopt($ch, CURLOPT_USERAGENT, 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET
11    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
12    curl_setopt($ch, CURLOPT_POST, 1);
13    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, TRUE);
14    curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 5);
15    curl_setopt($ch, CURLOPT_TIMEOUT, 5);
16    curl_setopt($ch, CURLOPT_SSL_VERIFYPeer, false);
17    curl_setopt($ch, CURLOPT_POSTFIELDS, $fields);
18    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
19    $data = curl_exec($ch);
20    $err = curl_error($ch);
21    curl_close($ch);
22    if ($err) {
23        return "cURL Error #:" . $err;
24    } else {
25        return $data;
26    }
27 }
28
29 function createReview($ucid, $class_id, $professor_id, $rating, $text) {
30     date_default_timezone_set('UTC');
31     $timestamp = date('Y-m-d H:i:s',time()) ;
32     $profileId = getProfileId($ucid);
33     $fields = "opcode=26&professorID=$professor_id&studentID=$profileId&classID=$class_id&timegiven=$timestamp";
34     $result = postToDatabase($fields);
35
36     $message = $result['message'];
37     if (!str_compare($message, "created review")) {
38         die(json_encode(['message' => "There was an error creating review." . $message, 'error' => false]));
39     } else {
40         return selectProfessorReviews($professor_id);
41     }
42 }
43 //SELECT reviewText FROM reviews WHERE reviewID = 11
44
45 function selectProfessorReviews($professor_id) {
46     $fields = "opcode=24&professorID=$professor_id";
47     $result = postToDatabase($fields);
48
49     $name = selectProfessorName($professor_id);
50     if (count($result['message']['data']) == 0) {
51         die(json_encode(['message' => "This professor does not exist.", 'error' => true]));
52     } else {
53         $average = $result['globalaverage']['grade'];
54         $reviews = selectReviews($professor_id);
55         $review = ['id' => $professor_id, 'name' => $name, 'average' => $average, 'reviews' => $reviews];
56         return $review;
57     }
58 }
```

File - C:\Users\Tevin\Desktop\Development\Homework\CS-490\api\common.php

```
59
60 function selectReviews($professor_id) {
61     $fields = "opcode=0&sql=SELECT reviews.*, classes.className, professorName FROM reviews JOIN
62     classes.classID = reviews.search_classID JOIN professors ON professorID = reviews.search_pr
63     WHERE search_professorID=$professor_id ORDER BY timegiven DESC";
64     $result = postToDatabase($fields);
65     $arr = [];
66     foreach ($result['data'] as $value) {
67         array_push($arr, ['id' => $value['reviewID'], 'class' => $value['className'], 'time' =>
68     }
69     return $arr;
70 }
71
72 function selectStudentReviews($profile_id) {
73     $fields = "opcode=0&sql=SELECT reviews.*, classes.className, professorName FROM reviews JOIN
74     classes.classID = reviews.search_classID JOIN professors ON professorID = reviews.search_pr
75     WHERE search_studentprofileID=$profile_id ORDER BY timegiven DESC";
76     $result = postToDatabase($fields);
77     $arr = [];
78     foreach ($result['data'] as $value) {
79         array_push($arr, ['id' => $value['reviewID'], 'name' => $value['professorName'], 'class'
80     }
81     return $arr;
82 }
83
84
85 function selectProfessorName($professor_id) {
86     $fields = "opcode=0&sql=SELECT professorName FROM professors WHERE professorID=$professor_id";
87     $result = postToDatabase($fields);
88     return $result['data'][0]['professorName'];
89 }
90
91
92 function createGroup($ucid, $groupName, $interests) {
93     $profileId = getProfileId($ucid);
94     $fields = "opcode=18&groupName=$groupName&ownerID=$profileId";
95     $result = postToDatabase($fields);
96
97     $message = $result['message'];
98     if (str_compare($message, 'Exists')) {
99         die(encode_json(['message' => "There was an error creation group. $groupName already e
100     } else if (strcmp($message, 'inserted')) {
101         $groupId = $result['groupID'];
102         if(updateGroup($groupId, $groupName, $profileId, $interests)) {
103             return selectUserOptions($ucid, ["profile" => true, "groups_own" => true]);
104         }
105     }
106     return null;
107 }
108
109 function searchGroupsByName($keyword) {
110     /*SELECT * FROM groups WHERE groupName LIKE '%%'*/
111     $fields = "opcode=0&sql=SELECT * FROM groups WHERE groupName LIKE '%$keyword%'";
112     $result = postToDatabase($fields);
113     $arr = [];
114     foreach ($result['data'] as $value) {
115         array_push($arr, ['id' => $value['groupID'], 'name' => $value['groupName']]);
116     }
```

```

File - C:\Users\Tevin\Desktop\Development\Homework\CS-490\api\common.php

117     return $arr;
118 }
119
120 function searchGroupsByInterest($interest_id) {
121     $fields = "opcode=0&sql=SELECT groups.* FROM groups JOIN groupsIntrests ON groupID = search_
122     intrests ON intrestID = search_intrestID WHERE intrestID = $interest_id";
123     $result = postToDatabase($fields);
124     $arr = [];
125     foreach ($result['data'] as $value) {
126         array_push($arr, ['id' => $value['groupID'], 'name' => $value['groupName']]);
127     }
128     return $arr;
129 }
130
131 function updateGroup($groupId, $groupName, $ownerID, $interests) {
132     $fields = "opcode=21&groupID=$groupId";
133
134     if (!empty($groupName)) {
135         $fields .= "&groupName=$groupName";
136     }
137     if (!empty($ownerID)) {
138         $fields .= "&ownerID=$ownerID";
139     }
140     $fields .= "&interestsIDs=". json_encode($interests);
141     $result = postToDatabase($fields);
142
143     if (str_compare($result['message'], 'updated')) {
144         return true;
145     } else {
146         die(encode_json(['message' => "There was an error creation group. Could not insert ids",
147     ])
148 }
149
150 function selectProfileGroups($profileId) {
151     $fields = "opcode=20&ownerID=$profileId";
152     $result = postToDatabase($fields);
153     $groups = [];
154     foreach ($result['data'] as $value) {
155         array_push($groups, ["id" => $value['groupID'], "name" => $value['groupName']]);
156     }
157     return $groups;
158 }
159
160 function selectGroup($groupId, $options = []) {
161     $fields = "opcode=19&groupID=$groupId";
162     $result = postToDatabase($fields);
163
164     if (!isset($result['data']['search_ownerprofileID'])) {
165         die(encode_json(['message' => "The group does not exist", 'error' => true]));
166     }
167     $ownerUcid = getUcid($result['data']['search_ownerprofileID']);
168
169     $group = ['id' => $groupId, 'name' => $result['data']['groupName'],
170             'ownerId' => $result['data']['search_ownerprofileID'],
171             'ownerUcid' => $ownerUcid];
172
173     if (isset($options['posts'])) {$options['posts']:false} {

```

File - C:\Users\Tevin\Desktop\Development\Homework\CS-490\api\common.php

```

175     $group['posts'] = selectGroupPosts($groupId);
176 }
177 if (isset($options['interests'])) {$options['interests']=false} {
178     $interests = [];
179
180     foreach($result['data']['interests'] as $value) {
181         array_push($interests, ['id' => $value['intrestID'], 'name' => $value['intrestName']]);
182     }
183     $group['interests'] = $interests;
184 }
185 return $group;
186 }
187
188 function createGroupPost($group_id, $from_ucid, $postText) {
189     $from_profileId = getProfileId($from_ucid);
190     date_default_timezone_set('UTC');
191     $timestamp = date('Y-m-d H:i:s', time());
192     $fields = "opcode=9&posterID=$from_profileId&groupID=$group_id&postText=$postText&timeStamp=$timestamp";
193     $result = postToDatabase($fields);
194     $message = $result['message'];
195     $postId = $result['postID'];
196     var_dump($result);
197     if (str_compare($message, 'Inserted')) {
198         return true;
199     } else {
200         return null;
201     }
202 }
203
204 function selectGroupPosts($groupId) {
205     $fields = "opcode=11&groupID={$groupId}";
206     $result = postToDatabase($fields);
207     $posts = [];
208     foreach ($result['data'] as $item) {
209         array_push($posts, ['id' => $item['postID'], 'postText' => $item['postText'], 'timeStamp' => $item['posted_by']]);
210     }
211     return $posts;
212 }
213
214
215 function isAdmin($ucid) {
216     $profileId = getProfileId($ucid);
217     $fields = "opcode=16&profileID=$profileId";
218     $result = postToDatabase($fields);
219     return str_compare($result['message'], "is admin");
220 }
221
222 function createProfilePost($to_ucid, $from_ucid, $postText) {
223     $to_profileId = getProfileId($to_ucid);
224     $from_profileId = getProfileId($from_ucid);
225     date_default_timezone_set('UTC');
226     $timestamp = date('Y-m-d H:i:s', time());
227     $fields = "opcode=9&posterID={$from_profileId}&profileID={$to_profileId}&postText={$postText}";
228     $result = postToDatabase($fields);
229     $message = $result['message'];
230     if (str_compare($message, 'Inserted')) {
231         return true;
232     } else {

```

File - C:\Users\Tevin\Desktop\Development\Homework\CS-490\api\common.php

```
233         return null;
234     }
235 }
236
237 function deletePost($post_id) {
238     $fields = "opcode=13&postID=$post_id";
239     $result = postToDatabase($fields);
240     return str_compare($result['message'], "Deleted");
241 }
242
243
244 function selectProfilePosts($profileId) {
245     $fields = "opcode=11&profileID={$profileId}";
246     $result = postToDatabase($fields);
247     $posts = [];
248     foreach ($result['data'] as $item) {
249         array_push($posts, ['id' => $item['postID'], 'postText' => $item['postText'], 'timeStamp'
250                           'posted_by' => getCacheProfile($item['search_senderprofileID'])]);
251     }
252     return $posts;
253 }
254
255 function insertInterest($profileId, $interestId) {
256     $fields = "opcode=0&sql=INSERT INTO profileIntrests (search_profileID, search_intrestID) VAI
257     ON DUPLICATE KEY UPDATE search_intrestID=search_intrestID";
258     $result = postToDatabase($fields);
259     $message = $result['message'];
260     if (str_compare($message, 'worked')) {
261         return true;
262     } else {
263         return false;
264     }
265 }
266
267 function selectInterests($profileId) {
268     $fields = "opcode=0&sql=SELECT intrestID, intrestName FROM profileIntrests JOIN intrests ON
269     profileIntrests.search_intrestID WHERE search_profileID = $profileId ORDER BY intrestName";
270     $result = postToDatabase($fields);
271     $arr = [];
272     foreach ($result['data'] as $val) {
273         array_push($arr, ['id' => $val['intrestID'], 'name' => $val['intrestName']]);
274     }
275     return $arr;
276 }
277
278 function getCacheProfile($profileId) {
279     if (!isset($GLOBALS['profileCache'][$profileId])) {
280         $GLOBALS['profileCache'][$profileId] = selectSmallProfile($profileId);
281     }
282     return $GLOBALS['profileCache'][$profileId];
283 }
284
285 function selectUser($ucid) {
286     return selectUserOptions($ucid, ['profile' => true]);
287 }
288
289 function selectUserOptions($ucid, $options = []) {
290     $fields = "opcode=2&ucid=$ucid";
```

File - C:\Users\Tevin\Desktop\Development\Homework\CS-490\api\common.php

```
291     $result = postToDatabase($fields);
292
293     if (isset($result['username'])) {
294         $username = $result['username'];
295         $profileId = $result['search_profileID'];
296         $email = $result['email'];
297
298         $json = ['ucid' => $ucid, 'username' => $username, 'email' => $email];
299         if (isset($options['profile']) ? $options['profile'] : false) {
300             $json['profile'] = selectProfile($profileId);
301         }
302         if (isset($options['posts']) ? $options['posts'] : false) {
303             $json['profile']['posts'] = selectProfilePosts($profileId);
304         }
305         if (isset($options['interests']) ? $options['interests'] : false) {
306             $json['profile']['interests'] = selectInterests($profileId);
307         }
308         if (isset($options['groups_own']) ? $options['groups_own'] : false) {
309             $json['profile']['groups_own'] = selectProfileGroups($profileId);
310         }
311         if (isset($options['reviews']) ? $options['reviews'] : false) {
312             $json['reviews'] = selectStudentReviews($profileId);
313         }
314         return $json;
315     } else {
316         die(encode_json(['message' => "There was an error retrieving user profile." . json_encode($result)]));
317     }
318 }
319
320
321 function getPasswordId($ucid) {
322     $fields = "opcode=2&ucid={$ucid}";
323     $result = postToDatabase($fields);
324     if (isset($result['passwordID'])) {
325         return $result['passwordID'];
326     } else {
327         return null;
328     }
329 }
330
331 function getProfileId($ucid) {
332     if (!isset($GLOBALS['ucidCache'][$ucid])) {
333         $fields = "opcode=2&ucid={$ucid}";
334         $result = postToDatabase($fields);
335         $profileId = isset($result['search_profileID']) ? $result['search_profileID'] : '';
336         if (empty($profileId)) {
337             return null;
338         }
339         $GLOBALS['ucidCache'][$ucid] = $profileId;
340     }
341     return $GLOBALS['ucidCache'][$ucid];
342 }
343
344 function getUcid($profileId) {
345     $fields = "opcode=15&profileID={$profileId}";
346     $result = postToDatabase($fields);
347     if (isset($result['ucid'])) {
348         return $result['ucid'];
```

File - C:\Users\Tevin\Desktop\Development\Homework\CS-490\api\common.php

```
349     }
350     return null;
351 }
352
353 function updateUser($ucid, $username, $last, $first, $password, $profileId) {
354     //ucid mandatory
355     $passId = getPasswordID($ucid);
356     $fields = "opcode=3&ucid={$ucid}&passwordID={$passId}";
357     if (!empty($username)) {
358         $fields .= "&username={$username}";
359     }
360     if (!empty($last)) {
361         $fields .= "&lastname={$last}";
362     }
363     if (!empty($first)) {
364         $fields .= "&firstname={$first}";
365     }
366     if (!empty($password)) {
367         $password = password_hash($password, PASSWORD_DEFAULT);
368         $fields .= "&password={$password}";
369     }
370     if (!empty($profileId)) {
371         $fields .= "&search_profileID={$profileId}";
372     }
373
374     $result = postToDatabase($fields);
375     $message = $result['message'];
376     if (str_compare($message, 'updated')) {
377         return selectUser($ucid);
378     } else {
379         return null;
380     }
381
382 }
383
384 function createProfile($ucid, $firstname, $lastname, $relationshipId, $classId, $genderId, $stat
385     // Check if profile already exists
386 /*    $result = selectUser($ucid);
387    if (!is_null($result['profile'])) {
388        return null;
389    }*/
390    $profileId = initProfile($firstname, $lastname);
391    $result = updateUser($ucid, "", "", "", "", $profileId);
392    if (is_null($result)) {
393        return null;
394    }
395    $fields = "opcode=7&profileID={$profileId}";
396    if (!empty($lastname)) {
397        $lastname = addslashes($lastname);
398        $fields .= "&lastname={$lastname}";
399    }
400    if (!empty($firstname)) {
401        $firstname = addslashes($firstname);
402        $fields .= "&firstname={$firstname}";
403    }
404    if (!empty($relationshipId)) {
405        $fields .= "&search_relationshipID={$relationshipId}";
406    }
```

File - C:\Users\Tevin\Desktop\Development\Homework\CS-490\api\common.php

```
407     if (!empty($genderId)) {
408         $fields .= "&search_genderID={$genderId}";
409     }
410     if (!empty($classId)) {
411         $fields .= "&search_gradeID={$classId}";
412     }
413     if (!empty($status)) {
414         $status = addslashes($status);
415         $fields .= "&status={$status}";
416     }
417     if (!empty($image)) {
418         $fields .= "&profilePicPath={$image}";
419     }
420     $result = postToDatabase($fields);
421     $message = $result['message'];
422     if (str_compare($message, 'updated')) {
423         foreach ($interests as $key => $value) {
424             insertInterest($profileId, $value);
425         }
426         return selectUserOptions($ucid, ['profile' => true, 'interests' => true]);
427     } else {
428         return null;
429     }
430 }
431
432 function deleteUser($ucid) {
433     $fields = "opcode=4&ucid=$ucid";
434     $result = postToDatabase($fields);
435     return str_compare($result['message'], "Deleted");
436 }
437
438 function updateProfile($ucid, $firstname, $lastname, $relationshipId, $classId, $genderId, $stat
439     $profileId = getProfileId($ucid);
440     if (is_null($profileId)) {
441         return null;
442     }
443
444     $fields = "opcode=7&profileID={$profileId}";
445     if (!empty($lastname)) {
446         $lastname = addslashes($lastname);
447         $fields .= "&lastname={$lastname}";
448     }
449     if (!empty($firstname)) {
450         $firstname = addslashes($firstname);
451         $fields .= "&firstname={$firstname}";
452     }
453     if (!empty($relationshipId)) {
454         $fields .= "&search_relationshipID={$relationshipId}";
455     }
456     if (!empty($genderId)) {
457         $fields .= "&search_genderID={$genderId}";
458     }
459     if (!empty($classId)) {
460         $fields .= "&search_gradeID={$classId}";
461     }
462     if (!empty($status)) {
463         $status = addslashes($status);
464         $fields .= "&status={$status}";
```

File - C:\Users\Tevin\Desktop\Development\Homework\CS-490\api\common.php

```
465     }
466     if (!empty($image)) {
467         $image = addslashes($image);
468         $fields .= "&profilePicPath={$image}";
469     }
470     foreach ($interests as $key => $value) {
471         insertInterest($profileId, $value);
472     }
473     if (!empty($first_name) || !empty($last_name) || !empty($class_level) || !empty($gender) ||
474         $result = postToDatabase($fields);
475     $message = $result['message'];
476     if (str_compare($message, 'updated')) {
477         return selectUserOptions($ucid, ['profile' => true, 'interests' => true]);
478     } else {
479         return null;
480     }
481 }
482 return selectUserOptions($ucid, ['profile' => true, 'interests' => true]);
483 }
484
485 function initProfile($firstname, $lastname) {
486     $fields = "opcode=5&firstname={$firstname}&lastname={$lastname}";
487     $result = postToDatabase($fields);
488     $profileId = $result['profileID'];
489     return $profileId;
490 }
491
492 function selectProfile($profileId) {
493     $fields = "opcode=6&profileID={$profileId}";
494     $result = postToDatabase($fields);
495     if (!isset($result['profileID'])) return null;
496
497     $grade = getGrade($result['search_gradeID']);
498     $relationship = getRelationship($result['search_relationshipID']);
499     $gender = getGender($result['search_genderID']);
500
501     $json = ['profile_id' => $profileId, 'first_name' => $result['firstName'], 'last_name' => $re
502         'class_level' => $grade, 'relationship' => $relationship, 'gender' => $gender, 'about' =
503         'image' => $result['profilePicPath']];
504     return $json;
505 }
506
507 function selectSmallProfile($profileId) {
508     $fields = "opcode=6&profileID={$profileId}";
509     $result = postToDatabase($fields);
510     if (!isset($result['profileID'])) return null;
511     $json = ['profile_id' => $profileId, 'ucid' => getUcid($profileId), 'first_name' => $result|
512     return $json;
513 }
514
515 function createUser($user, $pass, $email, $ucid) {
516     $pass = password_hash($pass, PASSWORD_DEFAULT);
517     $fields = "opcode=1&username={$user}&password={$pass}&ucid={$ucid}&email={$email}";
518     $result = postToDatabase($fields);
519     if (str_compare(strtolower($result['message']), 'exists')) {
520         return null;
521     }
522     return selectUser($ucid);
```

```
523 }
524
525 function checkPassword($user, $pass) {
526     $fields = "opcode=2&ucid={$user}";
527     $result = postToDatabase($fields);
528     if (isset($result['password']) && password_verify($pass, $result['password'])) {
529         return true;
530     } else {
531         return false;
532     }
533 }
534
535 function postToDatabase($fields) {
536     //var_dump($fields);
537     $result = postRequest('https://web.njit.edu/~maz9/DB/P3/', [], $fields);
538     //var_dump($result);
539     return json_decode($result, true);
540 }
541
542 function loginNjit($user, $pass) {
543     $headers = array(
544         'Origin' => 'https://www.njit.edu',
545         'Accept-Encoding' => 'gzip, deflate',
546         'Accept-Language' => 'en-US,en;q=0.8',
547         'Upgrade-Insecure-Requests' => '1',
548         'User-Agent' => 'Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36',
549         'Content-Type' => 'application/x-www-form-urlencoded',
550         'Accept' => 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8',
551         'Cache-Control' => 'max-age=0',
552         'Referer' => 'https://www.njit.edu/cp/login.php',
553         'Connection' => 'keep-alive',
554         'DNT' => '1',
555     );
556     $fields = "user={$user}&pass={$pass}&uuid=0xACA021";
557     return postRequest('https://cp4.njit.edu/cp/home/login', $headers, $fields);
558 }
559
560
561 /*
562 * 1 Freshman
563 * 2 Sophomore
564 * 3 Junior
565 * 4 Senior
566 */
567
568 function getGrade($gradeID) {
569     switch ($gradeID) {
570         case 1:
571             return "Freshman";
572         case 2:
573             return "Sophomore";
574         case 3:
575             return "Junior";
576         case 4:
577             return "Senior";
578         default:
579             return "Error getting grade, should not happen";
580     }
}
```

File - C:\Users\Tevin\Desktop\Development\Homework\CS-490\api\common.php

```
581 }
582
583
584 /*
585 * 1 Single
586 * 2 Dating
587 * 3 Married
588 * 4 Complicated
589 */
590
591 function getRelationship($relationshipID) {
592     switch ($relationshipID) {
593         case 1:
594             return "Single";
595         case 2:
596             return "Dating";
597         case 3:
598             return "Married";
599         case 4:
600             return "Complicated";
601         default:
602             return "Error getting relationship, should not happen";
603     }
604 }
605
606 function getGender($genderID) {
607     switch ($genderID) {
608         case 1:
609             return "Male";
610         case 2:
611             return "Female";
612         case 3:
613             return "Other";
614         default:
615             return "Error getting gender, should not happen";
616     }
617 }
618
619 function encode_json($value) {
620     return json_encode($value, JSON_PRETTY_PRINT | JSON_UNESCAPED_SLASHES);
621 }
622
623 function str_compare($str1, $str2) {
624     return (strcmp(strtolower($str1), strtolower($str2)) == 0);
625 }
626
627 function getUploaded() {
628     if (!empty($_FILES["file"]["name"])) {
629         $target_dir = "uploads/";
630         if (strcmp($_SERVER['SERVER_SOFTWARE'], 'Apache')) {
631             $target_dir = $_SERVER["CONTEXT_DOCUMENT_ROOT"] . '/api/profile/uploads/';
632         }
633         $file_name = $_SERVER['REQUEST_TIME'] * 1000 . '_' . basename($_FILES["file"]["name"]);
634         $target_file = $target_dir . $file_name;
635
636         $uploadOk = 1;
637         $imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));
638         // Check if image file is a actual image or fake image
```

File - C:\Users\Tevin\Desktop\Development\Homework\CS-490\api\common.php

```
639     if (isset($_POST["submit"])) {
640         $check = getimagesize($_FILES["file"]["tmp_name"]);
641         if ($check !== false) {
642             $uploadOk = 1;
643         } else {
644             http_response_code(400);
645             die(json(['message' => "Bad request - File provided was not a png, jpeg or gif"]));
646         }
647     }
648
649     // Check file size
650     if ($_FILES["file"]["size"] > 5000000) {
651         http_response_code(413);
652         die(json(['message' => "Payload too large - File provided was too large. Must be less than 5MB"]));
653     }
654
655     // Allow certain file formats
656     if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
657         && $imageFileType != "gif")
658     ) {
659         http_response_code(400);
660         die(json(['message' => "Bad request - File provided was not a png, jpeg or gif"]));
661     }
662
663     if (move_uploaded_file($_FILES["file"]["tmp_name"], $target_file)) {
664         return 'http://web.njit.edu/~tj76/api/profile/uploads/' . $file_name;
665     } else {
666         return "";
667     }
668 }
669 }
670
671 function checkAuth() {
672     if (!isset($_SERVER['PHP_AUTH_USER'])) {
673         header('WWW-Authenticate: Basic realm="My Realm"');
674         header('HTTP/1.0 401 Unauthorized');
675         die(json(['message' => "Must supply authorization header. http://stackoverflow.com"]));
676     } else {
677         if (!checkPassword($_SERVER['PHP_AUTH_USER'], $_SERVER['PHP_AUTH_PW'])) {
678             die(json(['message' => "UCID or Password is incorrect", 'error' => true]));
679         }
680     }
681 }
682
683
```