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

```
1 *.png
2 *.jpg
3 *.jpeg
4 *.gif
5 *.PNG
6 *.JPG
7 *.JPEG
8 *.GIF
```

```php
<?php

if( str_compare( $_SERVER['SERVER_SOFTWARE'], 'Apache')) {
    error_reporting( error_reporting() & ~E_NOTICE );
}

function postRequest($url, $headers, $fields) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_USERAGENT, 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, TRUE);
    curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 5);
    curl_setopt($ch, CURLOPT_TIMEOUT, 5);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fields);
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    $data = curl_exec($ch);
    $err = curl_error($ch);
    curl_close($ch);
    if ($err) {
        return "cURL Error #:" . $err;
    } else {
        return $data;
    }
}


function getRecommendedPeople($profileId) {
    $ids = selectSimilarPeople($profileId);

    return $ids;
}

function selectSimilarPeople($profile_id){
    $fields = "opcode=0&sql=SELECT firstname, lastname, profilePicPath, profileID, intrestName, ucid, email, username FROM profiles JOIN profileIntrests
    ON profiles.profileID = profileIntrests.search_profileID JOIN intrests ON intrests.intrestID = profileIntrests.search_intrestID
    JOIN passwords ON passwords.search_profileID = profiles.profileID WHERE intrestID IN (
SELECT intrestID FROM profiles JOIN profileIntrests ON profiles.profileID = profileIntrests.search_profileID JOIN intrests
ON intrests.intrestID = profileIntrests.search_intrestID JOIN passwords ON passwords.search_profileID = profiles.profileID
WHERE profiles.profileID = $profile_id) AND profiles.profileID <> $profile_id";

    $result = postToDatabase($fields);
    $arr = [];
    foreach ($result['data'] as $value) {
        array_push($arr, ['id' => $value['profileID'],
            'ucid' =>  $value['ucid'],
            'first_name' =>  $value['firstname'],
            'last_name' =>  $value['lastname'],
            'username' =>  $value['username'],
            'email' =>  $value['email'],
            'image' =>  $value['profilePicPath'],
            'reason' =>  $value['intrestName']
        ]);
    }
    return $arr;
}

function getRecommendedGroups($profileId) {
    $ids = selectSimilarGroups($profileId);
    return $ids;
}


```

```
 66 function selectSimilarGroups($profile_id){
 67     $fields = "opcode=0&sql=SELECT groupID, groupName, intrestName, search_ownerprofileID,
    ucid FROM groups
 68     JOIN groupsIntrests ON groups.groupID = groupsIntrests.search_groupID
 69     JOIN intrests ON intrests.intrestID = groupsIntrests.search_intrestID
 70     JOIN passwords ON search_ownerProfileID = passwords.search_profileID
 71     WHERE intrestID IN (
 72     SELECT intrestID FROM profiles JOIN profileIntrests ON profiles.profileID =
    profileIntrests.search_profileID
 73     JOIN intrests ON intrests.intrestID = profileIntrests.search_intrestID
 74     JOIN passwords ON passwords.search_profileID = profiles.profileID WHERE profiles.
    profileID = $profile_id)
 75     AND search_ownerprofileID <> $profile_id";
 76
 77     $result = postToDatabase($fields);
 78     $arr = [];
 79     foreach ($result['data'] as $value) {
 80         array_push($arr, ['id' => $value['groupID'],
 81             'name' =>  $value['groupName'],
 82             'ownerId' =>  $value['search_ownerprofileID'],
 83             'ownerUcid' =>  $value['ucid'],
 84             'reason' =>  $value['intrestName']]);
 85     }
 86     return $arr;
 87 }
 88
 89 function createReview($ucid, $class_id, $professor_id, $rating, $text) {
 90     date_default_timezone_set('UTC');
 91     $timestamp = date('Y-m-d H:i:s',time()) ;
 92     $profileId = getProfileId($ucid);
 93     $text = addslashes($text);
 94     $fields = "opcode=26&professorID=$professor_id&studentID=$profileId&classID=$class_id&
    timegiven=$timestamp&reviewgrade=$rating&reviewtext=$text";
 95     $result = postToDatabase($fields);
 96
 97     $message = $result['message'];
 98     if (!str_compare($message, "created review")) {
 99         die(encode_json(['message' => "There was an error creation review. Does the user or
     professor exists?", 'error' => true]));
100     } else {
101         return selectProfessorReviews($professor_id);
102     }
103 }
104 //SELECT reviewText FROM reviews WHERE reviewID = 11
105
106 function selectProfessorReviews($professor_id) {
107     $fields = "opcode=24&professorID=$professor_id";
108     $result = postToDatabase($fields);
109
110     $name = selectProfessorName($professor_id);
111     if (count($result['data']) == 0) {
112         die(encode_json(['message' => "This professor does not exist.", 'error' => true]));
113     } else {
114         $average = $result['globalaverage']['grade'];
115         $reviews = selectReviewsByProfessor($professor_id);
116         $review = ['id' => $professor_id, 'name' => $name, 'average' => $average, 'reviews
    ' => $reviews];
117         return $review;
118     }
119 }
120
121 function removeReviews($review_id) {
122     $fields = "opcode=0&sql=DELETE FROM reviews WHERE reviewID = $review_id";
123     $result = postToDatabase($fields);
124     return ['result' => $result];
125 }
126
127 function selectReviewsByProfessor($professor_id) {
128     $fields = "opcode=0&sql=SELECT reviews.*, classes.className, professorName FROM reviews
     JOIN classes ON
```

```
129     classes.classID = reviews.search_classID JOIN professors ON professorID = reviews.
    search_professorID
130     WHERE search_professorID=$professor_id ORDER BY timegiven DESC";
131     $result = postToDatabase($fields);
132     date_default_timezone_set('UTC');
133     $arr = [];
134     foreach ($result['data'] as $value) {
135         array_push($arr, ['id' => $value['reviewID'],
136             'class_id' => $value['search_classID'],
137             'class' => $value['className'],
138             'time' => date("F j, Y, g:i a", strtotime($value['Timegiven'])),
139             'rating' => $value['Reviewgrade'], 'review' => $value['ReviewText']]);
140     }
141     return $arr;
142 }
143
144 function getClassAverageReview($class_id) {
145     $fields = "opcode=0&sql=SELECT avg(reviewgrade) avg FROM reviews
146     JOIN classes ON classes.classID = reviews.search_classID
147     JOIN professors ON professorID = reviews.search_professorID
148     WHERE classes.classID=$class_id ORDER BY timegiven DESC";
149
150     $result = postToDatabase($fields);
151     return $result['data'][0]['avg'];
152 }
153
154 function selectReviewsByClass($class_id) {
155
156     $average = getClassAverageReview($class_id);
157
158     $fields = "opcode=0&sql=SELECT reviews.*, classes.className, professorName FROM reviews
     JOIN classes ON
159     classes.classID = reviews.search_classID JOIN professors ON professorID = reviews.
    search_professorID
160     WHERE classes.classID=$class_id ORDER BY timegiven DESC";
161     $result = postToDatabase($fields);
162     date_default_timezone_set('UTC');
163
164     $class_name ="";
165     $arr = [];
166     foreach ($result['data'] as $value) {
167         $class_name = $value['className'];
168         array_push($arr, ['id' => $value['reviewID'],
169             'professor_id' => $value['search_professorID'],
170             'professor_name' => $value['professorName'],
171             'time' => date("F j, Y, g:i a", strtotime($value['Timegiven'])),
172             'rating' => $value['Reviewgrade'], 'review' => $value['ReviewText']]);
173     }
174
175     $review = ['id' => $class_id, 'name' => $class_name, 'average' => $average, 'reviews'
    => $arr];
176
177     return $review;
178 }
179
180 function selectStudentReviews($profile_id) {
181     $fields = "opcode=0&sql=SELECT reviews.*, classes.className, classes.classID,
    professorName FROM reviews JOIN classes ON
182     classes.classID = reviews.search_classID JOIN professors ON professorID = reviews.
    search_professorID
183     WHERE search_studentprofileID=$profile_id ORDER BY timegiven DESC";
184     $result = postToDatabase($fields);
185     $arr = [];
186     date_default_timezone_set('UTC');
187     foreach ($result['data'] as $value) {
188         array_push($arr, [
189             'id' => $value['reviewID'],
190             'professor_id' => $value['search_professorID'],
191             'professor_name' => $value['professorName'],
192             'class' => $value['className'],
```

```
193             'class_id' => $value['classID'],
194             'time' => date("F j, Y, g:i a", strtotime($value['Timegiven'])) ,
195             'rating' => $value['Reviewgrade'],
196             'review' => $value['ReviewText']]);
197     }
198     return $arr;
199 }
200
201
202 function selectProfessorName($professor_id) {
203     $fields = "opcode=0&sql=SELECT professorName FROM professors WHERE professorID=
    $professor_id";
204     $result = postToDatabase($fields);
205     return $result['data'][0]['professorName'];
206 }
207
208
209 function createGroup($ucid, $groupName, $interests) {
210     $profileId = getProfileId($ucid);
211     $groupName = addslashes($groupName);
212     $fields = "opcode=18&groupName=$groupName&ownerID=$profileId";
213     $result = postToDatabase($fields);
214
215     $message = $result['message'];
216     if (str_compare($message, 'Exists')) {
217         die(encode_json(['message' => "There was an error creation group.  $groupName
    already exists", 'error' => true]));
218     } else if (str_compare($message, 'inserted group')) {
219         $groupId = $result['groupID'];
220         if(updateGroup($groupId, "", "", $interests)) {
221             return selectUserOptions($ucid, ["groups_own" => true]);
222         }
223     }
224     return null;
225 }
226
227 function searchGroupsByName($keyword) {
228     /*SELECT * FROM groups WHERE groupName LIKE '%%'*/
229     $fields = "opcode=0&sql=SELECT * FROM groups
230      JOIN passwords on passwords.search_profileID = groups.search_ownerprofileID WHERE
    groupName LIKE '%$keyword%'";
231     $result = postToDatabase($fields);
232     $arr = [];
233     foreach ($result['data'] as $value) {
234         array_push($arr, ['id' => $value['groupID'],
235             'name' => $value['groupName'],
236             'ownerId' => $value['search_ownerprofileID'],
237             'ownerUcid' => $value['ucid'],
238
239         ]);
240     }
241     return $arr;
242 }
243
244 function searchGroupsByInterest($interest_id) {
245     $fields = "opcode=0&sql=SELECT * FROM groups JOIN groupsIntrests ON groupID =
    search_groupID JOIN
246     intrests ON intrestID = search_intrestID
247     JOIN passwords on passwords.search_profileID = groups.search_ownerprofileID
248     WHERE intrestID = $interest_id";
249     $result = postToDatabase($fields);
250     $arr = [];
251     foreach ($result['data'] as $value) {
252         array_push($arr, ['id' => $value['groupID'],
253             'name' => $value['groupName'],
254             'ownerId' => $value['search_ownerprofileID'],
255             'ownerUcid' => $value['ucid'],
256
257         ]);
258     }
```

```
259     return $arr;
260 }
261 
262 function deleteGroup($group_id) {
263     $fields = "opcode=22&groupID=$group_id";
264     $result = postToDatabase($fields);
265     return str_compare($result['message'], "Deleted");
266 }
267 
268 function updateGroup($groupId, $groupName, $ownerID, $interests) {
269     $fields = "opcode=21&groupID=$groupId";
270 
271     if (!empty($groupName)) {
272         $groupName = addslashes($groupName);
273         $fields .= "&groupName=$groupName";
274     }
275     if (!empty($ownerID)) {
276         $fields .= "&ownerID=$ownerID";
277     }
278     $fields .= "&intrestsIDs=". json_encode($interests);
279     $result = postToDatabase($fields);
280 
281     return str_compare($result['message'], 'updated');
282 }
283 
284 function selectProfileGroups($profileId) {
285     $fields = "opcode=20&ownerID=$profileId";
286     $result = postToDatabase($fields);
287     $groups = [];
288     foreach ($result['data'] as $value) {
289         array_push($groups, selectGroup($value['groupID'], ['interests' => true]));
290     }
291     return $groups;
292 }
293 
294 function selectGroup($groupId, $options = []) {
295     $fields = "opcode=19&groupID=$groupId";
296     $result = postToDatabase($fields);
297 
298     if (!isset($result['data']['search_ownerprofileID'])) {
299         return null;
300     }
301     $ownerUcid = getUcid($result['data']['search_ownerprofileID']);
302 
303 
304     $group = ['id' => $groupId, 'name' => $result['data']['groupName'],
305         'ownerId' => $result['data']['search_ownerprofileID'],
306         'ownerUcid' => $ownerUcid,];
307 
308     if (isset($options['posts'])?$options['posts']:false) {
309         $group['posts'] = selectGroupPosts($groupId);
310     }
311     if (isset($options['interests'])?$options['interests']:false) {
312         $interests = [];
313 
314         foreach($result['data']['intrests'] as $value) {
315             array_push($interests, ['id' => $value['intrestID'], 'name' => $value['
    intrestName']]);
316         }
317         $group['interests'] = $interests;
318     }
319     return $group;
320 }
321 
322 function createGroupPost($group_id, $from_ucid,$postText) {
323     $from_profileId = getProfileId($from_ucid);
324     date_default_timezone_set('UTC');
325     $timestamp = date('Y-m-d H:i:s',time()) ;
326     $postText = addslashes($postText);
327     $fields = "opcode=9&posterID=$from_profileId&groupID=$group_id&postText=$postText&
```

```
timeStamp=$timestamp";
    $result = postToDatabase($fields);
    $message = $result['message'];
    $postId = $result['postID'];
    if (str_compare($message, 'Inserted')) {
        return true;
    } else {
        return null;
    }
}

function selectGroupPosts($groupId) {
    $fields = "opcode=11&groupID={$groupId}";
    $result = postToDatabase($fields);
    $posts = [];
    date_default_timezone_set('UTC');
    foreach ($result['data'] as $item) {
        array_push($posts, ['id' => $item['postID'], 'postText' => $item['postText'], '
timeStamp' =>  date("F j, Y, g:i a", strtotime($item['timeStamp'])),
            'posted_by' => getCacheProfile($item['search_senderprofileID'])]);
    }
    return $posts;
}

function isAdmin($ucid) {
    $profileId = getProfileId($ucid);
    $fields = "opcode=16&profileID=$profileId";
    $result = postToDatabase($fields);
    return str_compare($result['message'], "is admin");
}

function createProfilePost($to_ucid, $from_ucid, $postText) {
    $to_profileId = getProfileId($to_ucid);
    $from_profileId = getProfileId($from_ucid);
    date_default_timezone_set('UTC');
    $timestamp = date('Y-m-d H:i:s',time()) ;
    $postText = addslashes($postText);
    $fields = "opcode=9&posterID={$from_profileId}&profileID={$to_profileId}&postText={
$postText}&timeStamp={$timestamp}";
    $result = postToDatabase($fields);
    $message = $result['message'];
    if (str_compare($message, 'Inserted')) {
        return true;
    } else {
        return null;
    }
}

function deletePost($post_id) {
    $fields = "opcode=13&postID=$post_id";
    $result = postToDatabase($fields);
    return str_compare($result['message'], "Deleted");
}


function selectProfilePosts($profileId) {
    $fields = "opcode=11&profileID={$profileId}";
    $result = postToDatabase($fields);
    $posts = [];
    foreach ($result['data'] as $item) {
        date_default_timezone_set('EST');
        array_push($posts, ['id' => $item['postID'], 'postText' => $item['postText'], '
timeStamp' => date("F j, Y, g:i a", strtotime($item['timeStamp'])),
            'posted_by' => getCacheProfile($item['search_senderprofileID'])]);
    }
    return $posts;
}

function insertInterest($profileId, $interestId) {
    $fields = "opcode=0&sql=INSERT INTO profileIntrests (search_profileID, search_intrestID
```

```
393 ) VALUES ($profileId, $interestId)
394      ON DUPLICATE KEY UPDATE search_intrestID=search_intrestID";
395     $result = postToDatabase($fields);
396     $message = $result['message'];
397     if (str_compare($message, 'worked')) {
398         return true;
399     } else {
400         return false;
401     }
402 }
403
404 function selectInterests($profileId) {
405     $fields = "opcode=0&sql=SELECT intrestID, intrestName FROM profileIntrests JOIN
    intrests ON intrests.intrestID =
406      profileIntrests.search_intrestID WHERE search_profileID = $profileId ORDER BY
    intrestName";
407     $result = postToDatabase($fields);
408     $arr = [];
409     foreach ($result['data'] as $val) {
410         array_push($arr, ['id' => $val['intrestID'], 'name' => $val['intrestName']]);
411     }
412     return $arr;
413 }
414
415 function getCacheProfile($profileId) {
416     if (!isset($GLOBALS['profileCache'][$profileId])) {
417         $GLOBALS['profileCache'][$profileId] = selectSmallProfile($profileId);
418     }
419     return $GLOBALS['profileCache'][$profileId];
420 }
421
422 function selectUser($ucid) {
423     return selectUserOptions($ucid, ['profile' => true]);
424 }
425
426 function selectUserOptions($ucid, $options = []) {
427     $fields = "opcode=2&ucid=$ucid";
428     $result = postToDatabase($fields);
429
430     if (isset($result['username'])) {
431         $username = $result['username'];
432         $profileId = $result['search_profileID'];
433         $email = $result['email'];
434
435         $json = ['ucid' => $ucid,
436             'username' => $username,
437             'email' => $email,
438             'admin' => isAdmin($ucid),
439
440         ];
441         if (isset($options['profile'])?$options['profile']:false) {
442             $json['profile'] = selectProfile($profileId);
443         }
444         if (isset($options['posts'])?$options['posts']:false) {
445             $json['profile']['posts'] = selectProfilePosts($profileId);
446         }
447         if (isset($options['interests'])?$options['interests']:false) {
448             $json['profile']['interests'] = selectInterests($profileId);
449         }
450         if (isset($options['groups_own'])?$options['groups_own']:false) {
451             $json['profile']['groups_own'] = selectProfileGroups($profileId);
452         }
453         if (isset($options['reviews'])?$options['reviews']:false) {
454             $json['reviews'] = selectStudentReviews($profileId);
455         }
456         if (isset($options['recommend_people'])?$options['recommend_people']:false) {
457             $json['profile']['recommend_people'] = getRecommendedPeople($profileId);
458         }
459         if (isset($options['recommend_groups'])?$options['recommend_groups']:false) {
460             $json['profile']['recommend_groups'] = getRecommendedGroups($profileId);
```

```
461         }
462         return $json;
463     } else {
464         die(encode_json(['message' => "There was an error retrieving user profile." .
    json_encode($result), 'error' => true]));
465     }
466 }
467
468
469 function getPasswordId($ucid) {
470     $fields = "opcode=2&ucid={$ucid}";
471     $result = postToDatabase($fields);
472     if (isset($result['passwordID'])) {
473         return $result['passwordID'];
474     } else {
475         return null;
476     }
477 }
478
479 function getProfileId($ucid) {
480     if (!isset($GLOBALS['ucidCache'][$ucid])) {
481         $fields = "opcode=2&ucid={$ucid}";
482         $result = postToDatabase($fields);
483         $profileId = isset($result['search_profileID'])? $result['search_profileID']:'';
484         if (empty($profileId)) {
485             return null;
486         }
487         $GLOBALS['ucidCache'][$ucid] =  $profileId;
488     }
489     return $GLOBALS['ucidCache'][$ucid];
490 }
491
492 function getUcid($profileId) {
493     $fields = "opcode=15&profileID=$profileId";
494     $result = postToDatabase($fields);
495     if (isset($result['ucid'])) {
496         return $result['ucid'];
497     }
498     return null;
499 }
500
501 function searchUsersByInterest($interest_id) {
502     $fields = "opcode=0&sql=SELECT * FROM profiles JOIN profileIntrests
503     ON profileID = search_profileID JOIN intrests ON intrestID = search_intrestID
504     JOIN passwords ON passwords.search_profileID = profiles.profileID
505     WHERE intrestID = $interest_id";
506     $result = postToDatabase($fields);
507     $arr = [];
508     foreach ($result['data'] as $value) {
509         array_push($arr, ['profile_id' => $value['profileID'],
510             'first_name' => $value['firstName'],
511             'last_name' => $value['lastName'],
512             'image' => $value['profilePicPath'],
513             'username' => $value['username'],
514             'ucid' => $value['ucid'],
515             'email' => $value['email'],
516         ]);
517     }
518     return $arr;
519 }
520
521 function searchUsersByName($name) {
522     $fields = "opcode=0&sql=SELECT * FROM profiles
523     LEFT JOIN passwords ON passwords.search_profileID = profiles.profileID
524     WHERE firstName LIKE '%$name%' OR lastName LIKE '%$name%'";
525     $result = postToDatabase($fields);
526     $arr = [];
527     foreach ($result['data'] as $value) {
528         array_push($arr, ['profile_id' => $value['profileID'],
529             'first_name' => $value['firstName'],
```

```
530             'last_name' => $value['lastName'],
531             'image' => $value['profilePicPath'],
532             'username' => $value['username'],
533             'ucid' => $value['ucid'],
534             'email' => $value['email'],
535         ]);
536     }
537     return $arr;
538 }
539 
540 function searchUsersByUcid($ucid) {
541     $fields = "opcode=0&sql=SELECT * FROM profiles JOIN profileIntrests
542     ON profileID = search_profileID JOIN intrests ON intrestID = search_intrestID
543     JOIN passwords ON passwords.search_profileID = profiles.profileID
544     WHERE ucid LIKE '%$ucid%'";
545     $result = postToDatabase($fields);
546     $arr = [];
547     foreach ($result['data'] as $value) {
548         array_push($arr, ['profile_id' => $value['profileID'],
549             'first_name' => $value['firstName'],
550             'last_name' => $value['lastName'],
551             'image' => $value['profilePicPath'],
552             'username' => $value['username'],
553             'ucid' => $value['ucid'],
554             'email' => $value['email'],
555         ]);
556     }
557     return $arr;
558 }
559 
560 function updateUser($ucid, $username, $last, $first, $password, $profileId) {
561     //ucid mandatory
562     $passId = getPasswordId($ucid);
563     $fields = "opcode=3&ucid={$ucid}&passwordID={$passId}";
564     if (!empty($username)) {
565         $fields .= "&username={$username}";
566     }
567     if (!empty($last)) {
568         $fields .= "&lastname={$last}";
569     }
570     if (!empty($first)) {
571         $fields .= "&firstname={$first}";
572     }
573     if (!empty($password)) {
574         $password = password_hash($password, PASSWORD_DEFAULT);
575         $fields .= "&password={$password}";
576     }
577     if (!empty($profileId)) {
578         $fields .= "&search_profileID={$profileId}";
579     }
580 
581     $result = postToDatabase($fields);
582     $message = $result['message'];
583     if (str_compare($message, 'updated')) {
584         return selectUser($ucid);
585     } else {
586         return null;
587     }
588 
589 }
590 
591 function createProfile($ucid, $firstname, $lastname, $relationshipId, $classId, $genderId,
    $status, $image, $interests) {
592     // Check if profile already exists
593     /*    $result = selectUser($ucid);
594         if (!is_null($result['profile'])) {
595             return null;
596         }*/
597     $profileId = initProfile($firstname, $lastname);
598     $result = updateUser($ucid,"","","","",$profileId);
```

```
599     if (is_null($result)) {
600         return null;
601     }
602     $fields = "opcode=7&profileID={$profileId}";
603     if (!empty($lastname)) {
604         $lastname = addslashes($lastname);
605         $fields .= "&lastname={$lastname}";
606     }
607     if (!empty($firstname)) {
608         $firstname = addslashes($firstname);
609         $fields .= "&firstname={$firstname}";
610     }
611     if (!empty($relationshipId)) {
612         $fields .= "&search_relationshipID={$relationshipId}";
613     }
614     if (!empty($genderId)) {
615         $fields .= "&search_genderID={$genderId}";
616     }
617     if (!empty($classId)) {
618         $fields .= "&search_gradeID={$classId}";
619     }
620     if (!empty($status)) {
621         $status = addslashes($status);
622         $fields .= "&status={$status}";
623     }
624     if (!empty($image)) {
625         $fields .= "&profilePicPath={$image}";
626     }
627     $result = postToDatabase($fields);
628     $message = $result['message'];
629     if (str_compare($message, 'updated')) {
630         foreach ($interests as $key => $value) {
631             insertInterest($profileId, $value);
632         }
633         return selectUserOptions($ucid, ['profile' => true, 'interests' => true]);
634     } else {
635         return null;
636     }
637 }
638 
639 function deleteUser($ucid) {
640     $fields = "opcode=4&ucid=$ucid";
641     $result = postToDatabase($fields);
642     return str_compare($result['message'], "Deleted");
643 }
644 
645 function updateProfile($ucid, $firstname, $lastname, $relationshipId, $classId, $genderId,
    $status, $image, $interests) {
646     $profileId = getProfileId($ucid);
647     if (is_null($profileId)) {
648         return null;
649     }
650 
651     $fields = "opcode=7&profileID={$profileId}";
652     if (!empty($lastname)) {
653         $lastname = addslashes($lastname);
654         $fields .= "&lastname={$lastname}";
655     }
656     if (!empty($firstname)) {
657         $firstname = addslashes($firstname);
658         $fields .= "&firstname={$firstname}";
659     }
660     if (!empty($relationshipId)) {
661         $fields .= "&search_relationshipID={$relationshipId}";
662     }
663     if (!empty($genderId)) {
664         $fields .= "&search_genderID={$genderId}";
665     }
666     if (!empty($classId)) {
667         $fields .= "&search_gradeID={$classId}";
```

```
668     }
669     if (!empty($status)) {
670         $status = addslashes($status);
671         $fields .= "&status={$status}";
672     }
673     if (!empty($image)) {
674         $image = addslashes($image);
675         $fields .= "&profilePicPath={$image}";
676     }
677     foreach ($interests as $key => $value) {
678         insertInterest($profileId, $value);
679     }
680 
681     if (!empty($firstname) || !empty($lastname) || !empty($classId) || !empty($genderId)
    || !empty($relationshipId) || !empty($status) || !empty($image)) {
682         $result = postToDatabase($fields);
683         $message = $result['message'];
684         if (str_compare($message, 'updated')) {
685             return selectUserOptions($ucid, ['profile' => true, 'interests' => true]);
686         } else {
687             return null;
688         }
689     }
690     return selectUserOptions($ucid, ['profile' => true, 'interests' => true]);
691 
692 }
693 
694 function initProfile($firstname, $lastname) {
695     $fields = "opcode=5&firstname={$firstname}&lastname={$lastname}";
696     $result = postToDatabase($fields);
697     $profileId = $result['profileID'];
698     return $profileId;
699 }
700 
701 function selectProfile($profileId) {
702     $fields = "opcode=6&profileID={$profileId}";
703     $result = postToDatabase($fields);
704     if (!isset($result['profileID'])) return null;
705 
706     $grade = getGrade($result['search_gradeID']);
707     $relationship = getRelationship($result['search_relationshipID']);
708     $gender = getGender($result['search_genderID']);
709 
710     if (is_null($result['profilePicPath'])) {
711         $result['profilePicPath'] = "http://i.imgur.com/cIiHMjg.png";
712     }
713     $json = ['profile_id' => $profileId, 'first_name' => $result['firstName'],'last_name'
    => $result['lastName'],
714         'class_level' => $grade, 'relationship' => $relationship, 'gender' => $gender, '
    about' => $result['status'],
715         'image' => $result['profilePicPath']];
716     return $json;
717 }
718 
719 function selectSmallProfile($profileId) {
720     $fields = "opcode=6&profileID={$profileId}";
721     $result = postToDatabase($fields);
722     if (!isset($result['profileID'])) return null;
723     if (is_null($result['profilePicPath'])) {
724         $result['profilePicPath'] = "http://i.imgur.com/cIiHMjg.png";
725     }
726     $json = ['profile_id' => $profileId, 'ucid' => getUcid($profileId), 'first_name' =>
    $result['firstName'],'last_name' => $result['lastName'], 'image' => $result['
    profilePicPath']];
727     return $json;
728 }
729 
730 function createUser($user, $pass, $email, $ucid) {
731     $pass = password_hash($pass, PASSWORD_DEFAULT);
732     $fields = "opcode=1&username={$user}&password={$pass}&ucid={$ucid}&email={$email}";
```

```
733     $result = postToDatabase($fields);
734     if (str_compare(strtolower($result['message']), 'exists')) {
735         return null;
736     }
737     return selectUser($ucid);
738 }
739 
740 function checkPassword($user, $pass){
741     $fields = "opcode=2&ucid={$user}";
742     $result = postToDatabase($fields);
743     if (isset($result['password']) && password_verify($pass, $result['password'])) {
744         return true;
745     } else {
746         return false;
747     }
748 }
749 
750 function postToDatabase($fields) {
751     //var_dump($fields);
752     $result = postRequest('https://web.njit.edu/~maz9/DB/P4/', [], $fields);
753     //var_dump($result);
754     return json_decode($result, true);
755 }
756 
757 function loginNjit($user, $pass) {
758     $headers = array(
759         'Origin' => 'https://www.njit.edu',
760         'Accept-Encoding' => 'gzip, deflate',
761         'Accept-Language' => 'en-US,en;q=0.8',
762         'Upgrade-Insecure-Requests' => '1',
763         'User-Agent' => 'Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML,
    like Gecko) Chrome/48.0.2564.116 Safari/537.36',
764         'Content-Type' => 'application/x-www-form-urlencoded',
765         'Accept' => 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q
    =0.8',
766         'Cache-Control' => 'max-age=0',
767         'Referer' => 'https://www.njit.edu/cp/login.php',
768         'Connection' => 'keep-alive',
769         'DNT' => '1',
770     );
771     $fields = "user={$user}&pass={$pass}&uuid=0xACA021";
772     return postRequest('https://cp4.njit.edu/cp/home/login', $headers, $fields);
773 }
774 
775 
776 /*
777  * 1 Freshman
778  * 2 Sophomore
779  * 3 Junior
780  * 4 Senior
781  * */
782 
783 function getGrade($gradeID) {
784     switch ($gradeID) {
785         case 1:
786             return "Freshman";
787         case 2:
788             return  "Sophomore";
789         case 3:
790             return "Junior";
791         case 4:
792             return "Senior";
793         default:
794             return "Error getting grade, should not happen";
795     }
796 }
797 
798 
799 /*
800  * 1 Single
```

```
 * 2 Dating
 * 3 Married
 * 4 Complicated
 * */

function getRelationship($relationshipID) {
    switch ($relationshipID) {
        case 1:
            return "Single";
        case 2:
            return "Dating";
        case 3:
            return "Married";
        case 4:
            return "Complicated";
        default:
            return "Error getting relationship, should not happen";
    }
}

function getGender($genderID) {
    switch ($genderID) {
        case 1:
            return "Male";
        case 2:
            return "Female";
        case 3:
            return "Other";
        default:
            return "Error getting gender, should not happen";
    }
}

function encode_json($value) {
    return json_encode($value, JSON_PRETTY_PRINT | JSON_UNESCAPED_SLASHES);
}

function str_compare($str1, $str2) {
    return (strcmp(strtolower($str1), strtolower($str2) ) == 0);
}

function getUploaded( ) {
    if (!empty($_FILES["file"]["name"])) {
        $target_dir = "uploads/";
        if (strcmp($_SERVER['SERVER_SOFTWARE'], 'Apache')) {
            $target_dir = $_SERVER["CONTEXT_DOCUMENT_ROOT"] . '/api/profile/uploads/';
        }
        $file_name = $_SERVER['REQUEST_TIME'] * 1000 . '_' . basename($_FILES["file"]["name
"]);
        $target_file = $target_dir . $file_name;

        $uploadOk = 1;
        $imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));
        // Check if image file is a actual image or fake image
        if (isset($_POST["submit"])) {
            $check = getimagesize($_FILES["file"]["tmp_name"]);
            if ($check !== false) {
                $uploadOk = 1;
            } else {
                http_response_code(400);
                die(encode_json(['message' => "Bad request - File provided was not a png,
    jpeg or gif. " . $check["mime"] . ".", 'error' => true]));
            }
        }

        // Check file size
        if ($_FILES["file"]["size"] > 5000000) {
            http_response_code(413);
            die(encode_json(['message' => "Payload too large - File provided was too large
    . Must be less than 5MB", 'error' => true]));
```

```
868         }
869 
870         // Allow certain file formats
871         if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
872             && $imageFileType != "gif"
873         ) {
874             http_response_code(400);
875             die(encode_json(['message' => "Bad request - File provided was not a png, jpeg
    or gif.", 'error' => true]));
876         }
877 
878         if (move_uploaded_file($_FILES["file"]["tmp_name"], $target_file)) {
879             return 'https://web.njit.edu/~tj76/api/profile/uploads/' . $file_name;
880         } else {
881             return "";
882         }
883     }
884 }
885 
886 function checkAuth() {
887     if (!isset($_SERVER['PHP_AUTH_USER'])) {
888         header('WWW-Authenticate: Basic realm="My Realm"');
889         header('HTTP/1.0 401 Unauthorized');
890         die(encode_json(['message' => "Must supply authorization header. http://
    stackoverflow.com/a/11960692/2238427", 'error' => true]));
891     } else {
892         if (!checkPassword($_SERVER['PHP_AUTH_USER'], $_SERVER['PHP_AUTH_PW'])) {
893             die(encode_json(['message' => "UCID or Password is incorrect", 'error' => true]
    ));
894         }
895     }
896 }
897 
898 
```