//server database, username, password

mysql\_connect('localhost','root','') or die('Cannot connect to server.');

//database name

mysql\_select\_db('members') or die('Cannot connect to database.');

//update user score everytime it increases

//function cardio\_points(){

//mysql\_query("UPDATE `users` SET `` = '' WHERE `user\_id` = '$user\_id'");

//manual inputs to store and stores the information in a .csv file

function inputData($cardio\_points, $user\_id){

global $user\_data;

$cardio\_points = $user\_data['cardio\_points'] + $cardio\_points;

$cardio\_points = round($cardio\_points);

echo 'Your user score is now: ' . $cardio\_points;

mysql\_query("UPDATE `users` SET `cardio\_points` = '" . $cardio\_points . "' WHERE `user\_id` = " . (int)$user\_id);

//create and store manually inputed data into a .csv file

//move .csv file to user\_data folder

}

//returns user tips

function daily\_tips($result){

$result = date('w');

//echo $result;

if($result == 0){

$random = rand(1,3);

switch ($random){

case 1:

$result = 'Find a smart and motivated training partner.';

break;

case 2:

$result = 'Schedule exact times for all workouts.';

break;

case 3:

$result = 'Keep a food and training log, and learn from it!';

break;

}

}

if($result == 1){

$random = rand(1,3);

switch ($random){

case 1:

$result = 'Don\'t train in pain. Pain is your body\'s way of telling you that something\'s wrong. Listen and act upon these messages!';

break;

case 2:

$result = 'Frame goals around behaviors rather than outcomes.';

break;

case 3:

$result = 'Continue doing what works. (Many times we tend to discontinue doing things which have worked in the past.)';

break;

}

}

if($result == 2){

$random = rand(1,3);

switch ($random){

case 1:

$result = 'Identify and fortify weak links. This could refer to habit patterns, muscle groups, motor qualities, etc.';

break;

case 2:

$result = 'In most things, the truth tends to be in the middle (rather than the extremes.)';

break;

case 3:

$result = 'If you\'re under 40, hold your stretches for 30 seconds. If you\'re over 40, hold them for 60 seconds.';

break;

}

}

if($result == 3){

$random = rand(1,3);

switch ($random){

case 1:

$result = 'Count your repetitions backward.';

break;

case 2:

$result = 'Exhale forcefully at the top of the movement when you do abdominal crunches.';

break;

case 3:

$result = 'If you don\'t like an exercise, start doing it. \'You are probably avoiding it because you are weak at it.\'';

break;

}

}

if($result == 4){

$random = rand(1,3);

switch ($random){

case 1:

$result = 'Use dumbbells, barbells, and machines in that order.';

break;

case 2:

$result = 'Keep your weight workouts under an hour.';

break;

case 3:

$result = 'Ensure your morning is seamless and quick.';

break;

}

}

if($result == 5){

$random = rand(1,3);

switch ($random){

case 1:

$result = 'Envision waking up early, working out and enjoying it.';

break;

case 2:

$result = 'To maximize your workout, good form is a must.';

break;

case 3:

$result = 'We\'re all capable of much more than we think. Believe yourself and you can accomplish it.';

break;

}

}

if($result == 6){

$random = rand(1,3);

switch ($random){

case 1:

$result = 'Do more work in the same amount of time (workout to workout) and muscle will grow.';

break;

case 2:

$result = 'In resistance training, we tend to focus too much on what to lift and not enough on how or why. Think about it.';

break;

case 3:

$result = 'Try going outside an hour a day, even if its just sitting in the park or walking around.';

break;

}

}

return $result;

}

//stores user score

function add\_cardio\_points($cardio\_points, $user\_id){

global $user\_data;

$cardio\_points = $user\_data['cardio\_points'] + $cardio\_points;

$cardio\_points = round($cardio\_points);

echo 'Your user score is now: ' . $cardio\_points;

mysql\_query("UPDATE `users` SET `cardio\_points` = '" . $cardio\_points . "' WHERE `user\_id` = " . (int)$user\_id);

}

//upload file, generate name, update user database with path of image

function uploadDeviceData($user\_id, $file\_temp, $file\_extn){

$file\_path = '../user\_data/files/' . $\_FILES['activity\_file']['name'];

move\_uploaded\_file($file\_temp, "~./user\_data/files/"."{$\_FILES['activity\_file']['name']}");

//insert to database

mysql\_query("UPDATE `users` SET `user\_file` = '" . mysql\_real\_escape\_string($file\_path) . "' WHERE `user\_id` = " . (int)$user\_id);

}

//checks if user is admin

//NOTE: if changing type, need to also change 'admin\_protect()' in general.php

function has\_access($user\_id, $type){

$user\_id = (int)$user\_id;

$type = (int)$type;

return (mysql\_result(mysql\_query("SELECT COUNT(`user\_id`) FROM `users` WHERE `user\_id` = $user\_id AND `type` = $type"), 0) == 1) ? true : false;

}

//recover password or username by email

function recover($mode, $email){

//sanitize the data to prevent injection

$mode = sanitize($mode);

$email = sanitize($email);

$user\_data = user\_data(user\_id\_from\_email($email), 'user\_id', 'first\_name', 'username');

if($mode == 'username'){

//recover username

email($email, 'Your username', "Hello " . $user\_data['first\_name'] . ",\n\nYour username is: " . $user\_data['username'] . "\n\n-something");

}else if($mode == 'password'){

//recover password

$generated\_password = substr(md5(rand(999, 999999)), 0, 8);

change\_password($user\_data['user\_id'], $generated\_password);

//force user to update new password after temporary password

update\_user($user\_data['user\_id'], array('password\_recover' => '1'));

email($email, 'Your password recovery', "Hello " . $user\_data['first\_name'] . ",\n\nYour new password is: " . $generated\_password . "\n\n-something");

}

}

//update user information

function accountSettings($user\_id, $update\_data){

$update = array();

//checks every element in array is valid

array\_walk($update\_data, 'array\_sanitize');

foreach($update\_data as $field=>$data){

$update[] = '`' . $field . '` = \'' . $data . '\'';

}

//for testing only

//print\_r($update);

//die();

mysql\_query("UPDATE `users` SET " . implode(', ', $update) . " WHERE `user\_id` = $user\_id");

}

//activating the account through email

function activate($email, $email\_code){

$email = mysql\_real\_escape\_string($email);

$email\_code = mysql\_real\_escape\_string($email\_code);

//check if email codes match

if(mysql\_result(mysql\_query("SELECT COUNT(`user\_id`) FROM `users` WHERE `email` = '$email' AND `email\_code` = '$email\_code' AND `active` = 0"), 0) == 1){

//query to update user active status

mysql\_query("UPDATE `users` SET `active` = 1 WHERE `email` = '$email'");

return true;

}else{

return false;

}

}

//change user password

function change\_password($user\_id, $password){

//sanitize the data

$user\_id = (int)$user\_id;

$password = md5($password);

mysql\_query("UPDATE `users` SET `password` = '$password', `password\_recover` = 0 WHERE `user\_id` = $user\_id");

}

//store the amount of active users

function user\_count(){

return mysql\_result(mysql\_query("SELECT COUNT(`user\_id`) FROM `users` WHERE `active` = 1"), 0);

}

//register new user data

function registerUser($register\_data){

//for testing only

//print\_r($register\_data);

//die();

//checks every element in array is valid

array\_walk($register\_data, 'array\_sanitize');

$register\_data['password'] = md5($register\_data['password']);

$fields = '`' . implode('`, `', array\_keys($register\_data)) . '`';

$data = '\'' . implode('\', \'', $register\_data) . '\'';

//for testing only

//echo $data;

//echo $fields;

//die();

mysql\_query("INSERT INTO `users` ($fields) VALUES ($data)");

//needs to change link and url to whatever link you are using

//send email of verification code

//email($register\_data['email'], 'Activate your account', "

//Hello " . $register\_data['first\_name'] . ",\n\nYou need to active your account, use the link below to activate your account:\n\nhttp://se1.engr.rutgers.edu/~13group12/activate.php?email=" . $register\_data['email'] . "&email\_code=" . $register\_data['email\_code'] . " \n\n-whoever");

}

//retrieves user data for usage

function user\_data($user\_id){

$data = array();

//remove characters from the id, prevents sql injection

$user\_id = (int)$user\_id;

$func\_num\_args = func\_num\_args();

$func\_get\_args = func\_get\_args();

//for testing only

//print\_r($func\_get\_args);

if($func\_num\_args > 1){

unset($func\_get\_args[0]);

$fields = '`' . implode('`, `', $func\_get\_args) . '`';

$data = mysql\_fetch\_assoc(mysql\_query("SELECT $fields FROM `users` WHERE `user\_id` = $user\_id"));

//return data in array

return $data;

}

}

//sorts the leader in cardio\_points

function leader\_list($user\_id) {

$sql = "SELECT \* FROM `users` ORDER BY `cardio\_points` DESC";

$result = mysql\_query($sql);

$i = 1;

echo "<table>

<tr><th colspan=\"5\">Top 5 Leaders</th></tr><tr><th>Number</th><th>Username</th><th>Age</th><th>Cardio Points</th><th>Gender</th></tr>";

while ($row = mysql\_fetch\_array($result)) {

//printf("Name: %s Score: %s", $row["username"], $row["cardio\_points"]);

echo '<tr><td>' . $i . '</td><td>' . $row["username"] . '</td><td>' . $row["age"] . '</td><td>' . $row["cardio\_points"] . '</td><td>' . $row["gender"] . '</td></tr>';

$i++;

if($i == 6){

break;

}

}

echo "</table>";

return $result;

}

//checks if the user is logged in

function logged\_in(){

return (isset($\_SESSION['user\_id'])) ? true : false;

}

//checks if user exists

function user\_exists($username){

//removes unnecessary keys, prevents sql injection

$username = sanitize($username);

//checks if user is in database From table

$query = mysql\_query("SELECT COUNT(`user\_id`) FROM `users` WHERE `username` = '$username'");

return(mysql\_result($query, 0) == 1) ? true : false;

}

//checks if user email exists

function email\_exists($email){

//removes unnecessary keys, prevents sql injection

$email = sanitize($email);

//checks if user is in database From table

return(mysql\_result(mysql\_query("SELECT COUNT(`user\_id`) FROM `users` WHERE `email` = '$email'"), 0) == 1) ? true : false;

}

function user\_active($username){

//removes unnecessary keys, prevents sql injection

$username = sanitize($username);

//checks if user is in database

return(mysql\_result(mysql\_query("SELECT COUNT(`user\_id`) FROM `users` WHERE `username` = '$username' AND `active` = 1"), 0) == 1) ? true : false;

}

function user\_id\_from\_username($username){

$username = sanitize($username);

return mysql\_result(mysql\_query("SELECT `user\_id` FROM `users` WHERE `username` = '$username'"), 0, 'user\_id');

}

function user\_id\_from\_email($email){

$email = sanitize($email);

return mysql\_result(mysql\_query("SELECT `user\_id` FROM `users` WHERE `email` = '$email'"), 0, 'user\_id');

}

function login($username, $password){

$user\_id = user\_id\_from\_username($username);

//secure the password and username

$username = sanitize($username);

//convert password

$password = md5($password);

return (mysql\_result(mysql\_query("SELECT COUNT(`user\_id`) FROM `users` WHERE `username` = '$username' AND `password` = '$password'"), 0) == 1) ? $user\_id : false;

}

//prevent register page to appear when logged in

function logged\_in\_redirect(){

if(logged\_in() === true){

header('Location: home.php');

exit();

}

}

//redirect user if not logged in

function protect\_page(){

if(logged\_in() === false){

header('Location: protected.php');

exit();

}

}

//keeps users away from page if not an admin

//NOTE: if changing type, need to change 'is\_admin()' in users.php

function admin\_protect(){

global $user\_data;

if(has\_access($user\_data['user\_id'], 3) === false){

header('Location: index.php');

exit();

}

}

//loop through elements to prevent sql injections

//NOTE: must have & sign before $item

//strips tags off first, then turns into html code

function array\_sanitize(&$item){

$item = htmlentities(strip\_tags(mysql\_real\_escape\_string($item)));

}

//removes escape characters to prevent sql injections

function sanitize($data){

return htmlentities(strip\_tags(mysql\_real\_escape\_string($data)));

}

//report error from array in a list

function output\_errors($errors){

return '<ul><li>' . implode('</li><li>', $errors) . '</li></ul>';

}