<?*php*use mikehaertl\wkhtmlto\Pdf;  
  
/\* Database connection settings \*/  
$host = "omitted";  
$user = "admin";  
$pass = "omitted";  
$db = "innodb";  
$mysqli = new mysqli($host,$user,$pass,$db) or die($mysqli->error);  
  
  
  
function check\_login() {  
 // Allows errors to display on all pages  
 error\_reporting(E\_ALL);  
 error\_reporting(-*1*);  
 ini\_set('display\_errors', *true*);  
  
 // Checks login  
 if ( $\_SESSION['logged\_in'] != *true* ) {  
 if ($\_SESSION['attempt']) {  
 $\_SESSION['rmessage'] = "Confirmation link has been sent. Please verify your account by clicking on the link in the message!";  
 $\_SESSION['goodbad'] = 'good';  
  
 }  
 else {  
 $\_SESSION['rmessage'] = "You must log in before viewing your profile page";  
 $\_SESSION['goodbad'] = 'bad';  
 }  
 header("location: ../accounts/registration.php");  
 }  
 else {  
 $\_SESSION['rmessage'] = '';  
 $\_SESSION['goodbad'] = 'good';  
 }  
}  
  
  
function safe\_query($mysqli, $query\_string) {  
 $result = $mysqli -> query($query\_string);  
 if(! empty( $mysqli->error ) ){  
 console\_log($query\_string);  
 console\_log($mysqli->error);  
 }  
// console\_log('safe query'.$query\_string);  
 return $result;  
}  
  
  
function make\_equality\_string($eq\_dict, $conj=' and ') {  
 $keys = array\_keys($eq\_dict);  
 $equality\_string = '';  
 foreach($keys as $key) {  
 if ($equality\_string == '') {  
 $conjunction = '';  
 }  
 else {  
 $conjunction = $conj;  
 }  
 if ($eq\_dict[$key] == *null*) {  
 $equality\_string = $equality\_string.$conjunction.$key." IS null ";  
 }  
 else {  
 $equality\_string = $equality\_string.$conjunction.$key." = '".$eq\_dict[$key]."'";  
  
 }  
 }  
 return $equality\_string;  
}  
  
  
function dict\_sanitize($mysqli, $dict) {  
 foreach ($dict as $key => $element) {  
 $element = $mysqli->escape\_string($element);  
 $element = addcslashes($element, '%\_');  
 $dict[$key] = $element;  
 }  
  
 return $dict;  
}  
  
  
  
function select\_query($mysqli, $table\_name, $eq\_dict, $desc\_field=*Null*) {  
 # make the query  
 $eq\_dict = dict\_sanitize($mysqli, $eq\_dict);  
  
  
 $equality\_string = make\_equality\_string($eq\_dict);  
 $qp1 = "select \* from ".$table\_name." where ".$equality\_string;  
 $qp2 = ";";  
  
 if ($desc\_field != *NULL*) {  
 $qp2 = " ORDER BY ".$desc\_field." DESC;";  
 }  
  
 $query = $qp1.$qp2;  
  
 $output = safe\_query($mysqli, $query);  
 console\_log($query);  
 console\_log("Select Query Executed --> Num rows: ".$output->num\_rows);  
 return $output;  
}  
  
  
function insert\_query($mysqli, $table\_name, $insert\_dict) {  
 $insert\_dict = dict\_sanitize($mysqli, $insert\_dict);  
  
 $fields = array\_keys($insert\_dict);  
 $values = $insert\_dict;  
 $f\_string = implode(", ", $fields);  
 $v\_string = '';  
  
  
 foreach ($values as $value) {  
 if ($value == *null*) {  
 $v\_string = $v\_string." null,";  
 }  
 else {  
 $v\_string = $v\_string."'".$value."',";  
 }  
 }  
// $v\_string = str\_replace(',);', ');', $v\_string);  
  
 $v\_string = rtrim($v\_string, ',);');  
  
 $query = "insert into ".$table\_name." (".$f\_string.") VALUES (".$v\_string.");";  
 console\_log($query);  
 return safe\_query($mysqli, $query);  
// if wait == False:  
// return make\_query(query)  
// else:  
// global wait\_string  
// wait\_string = wait\_string + query  
  
}  
  
  
function update\_query($mysqli, $table\_name, $eq\_dict, $change\_dict) {  
 $eq\_dict = dict\_sanitize($mysqli, $eq\_dict);  
 $change\_dict = dict\_sanitize($mysqli, $change\_dict);  
  
  
 $equality\_string = make\_equality\_string($eq\_dict);  
 $change\_string = make\_equality\_string($change\_dict, $conj=', ');  
 $query = 'UPDATE '.$table\_name.' SET '.$change\_string.' WHERE '.$equality\_string.';';  
 console\_log($query);  
 return safe\_query($mysqli, $query);  
}  
  
  
function console\_log($message, $priority=*1*) {  
 $curr\_priority = *1*;  
 if ($priority <= $curr\_priority) {  
 $STDERR = fopen("php://stderr", "w");  
 fwrite($STDERR, "*\n*".$message."*\n\n*");  
 fclose($STDERR);  
 }  
}  
  
  
// Post, Redirect, Get --> stops from resubmission on page refresh  
function prg() {  
 if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {  
 header("Location: {$\_SERVER['REQUEST\_URI']}", *true*, *303*);  
 }  
}  
  
  
function make\_time($kids) {  
 $num\_pages = *0*;  
  
 if ($kids == *1*) {  
 $wait = *1444*;  
 return $wait;  
 }  
  
 foreach ($kids as $kid) {  
 if ($kid['level'] == *0*) {  
 $num\_pages += *16*;  
 }  
 else {  
 $num\_pages += *4*;  
 }  
 }  
  
 console\_log("Nummpagess:".$num\_pages);  
 $wait = *1000* + ($num\_pages \* *111*);  
 return $wait;  
}  
  
  
  
function make\_pdf($html\_string, $kids) {  
 require\_once '../vendor/autoload.php';  
  
 $wait = make\_time($kids);  
  
 console\_log("Wait time:".$wait);  
 $options = array('javascript-delay' => $wait, 'margin-top' => '50mm');  
 $pdf = new Pdf;  
 $pdf->setOptions($options);  
 $pdf->addPage($html\_string);  
  
 if (!$pdf->send('wksht\_test.pdf')) {  
 console\_log('Error: pdf not sent');  
 $error = $pdf->getError();  
 console\_log($error);  
 }  
}  
  
  
function echo\_json($array) {  
 $myJSON = json\_encode($array);  
  
 echo $myJSON;  
}  
  
  
function python\_slice($string, $start, $end) {  
 $string = substr($string, $start, (strlen($string) - $start));  
}  
  
  
function unique\_multidim\_array($array, $key) {  
 $temp\_array = array();  
 $i = *0*;  
 $key\_array = array();  
  
 foreach($array as $val) {  
 if (!in\_array($val[$key], $key\_array)) {  
 $key\_array[$i] = $val[$key];  
 $temp\_array[$i] = $val;  
 }  
 $i++;  
 }  
 return $temp\_array;  
}  
  
  
function e($string) {  
 $ciphering = "AES-256-CTR";  
 $encryption\_iv = '1234567891011121';  
 $encryption\_key = "UCkey";  
 $options = *0*;  
  
 $encryption = openssl\_encrypt($string, $ciphering, $encryption\_key, $options, $encryption\_iv);  
 $encryption = urlencode($encryption);  
  
 return $encryption;  
}  
  
  
function d($encryption) {  
 $ciphering = "AES-256-CTR";  
 $decryption\_iv = '1234567891011121';  
 $decryption\_key = "UCkey";  
 $options = *0*;  
  
 $encryption = urldecode($encryption);  
 $decryption = openssl\_decrypt($encryption, $ciphering, $decryption\_key, $options, $decryption\_iv);  
  
 return $decryption;  
}  
  
  
?>