Validators.php

<?php

function isNumber($number, $min = 0, $max = 100): bool

{

return ($number >= $min and $number <= $max);

}

function validateText(string $theStr, string $theID){

$settings['options']['regexp'] = '/^[A-z]{2,10}$/';

// $result = filter\_input(INPUT\_POST, $theID, FILTER\_VALIDATE\_REGEXP, $settings);

// return $result;

//OR

return(filter\_input(INPUT\_POST, $theID, FILTER\_VALIDATE\_REGEXP, $settings))?true:false;

//ORIGINAL VERSION

// FILTER\_VALIDATE\_REGEXP, $settings)

// $validationFilters[$theID]['filter'] = FILTER\_VALIDATE\_REGEXP;

// $validationFilters['first\_name']['options']['regexp'] = '/^[A-z]{2,10}$/';

}

function isText(string $string, int $min = 0, int $max = 1000): bool

{

$length = mb\_strlen($string);

return ($length >= $min and $length <= $max);

}

function isEmail($email): bool

{

return (filter\_var($email, FILTER\_VALIDATE\_EMAIL)) ? true : false;

}

function isPassword($password)

{

if ( mb\_strlen($password) >= 8 // Length 8 or more chars

and preg\_match('/[A-Z]/', $password) // Contains uppercase A-Z

and preg\_match('/[a-z]/', $password) // Contains lowercase a-z

and preg\_match('/[0-9]/', $password) // Contains 0-9

) {

return true; // Passed all tests

}

return false; // Invalid password

}

function sanitizeData(&$userData){

print("sanitizing<br>");

$userData['first\_name']=filter\_var($userData['first\_name'], FILTER\_SANITIZE\_FULL\_SPECIAL\_CHARS);

$userData['last\_name']=filter\_var($userData['last\_name'], FILTER\_SANITIZE\_FULL\_SPECIAL\_CHARS);

$userData['user\_name']=filter\_var($userData['user\_name'], FILTER\_SANITIZE\_FULL\_SPECIAL\_CHARS);

$userData['email']=filter\_var($userData['email'], FILTER\_SANITIZE\_EMAIL);

//no need to sanitize password but before inserting password into DB store its hash and insert that into the DB

}

?>