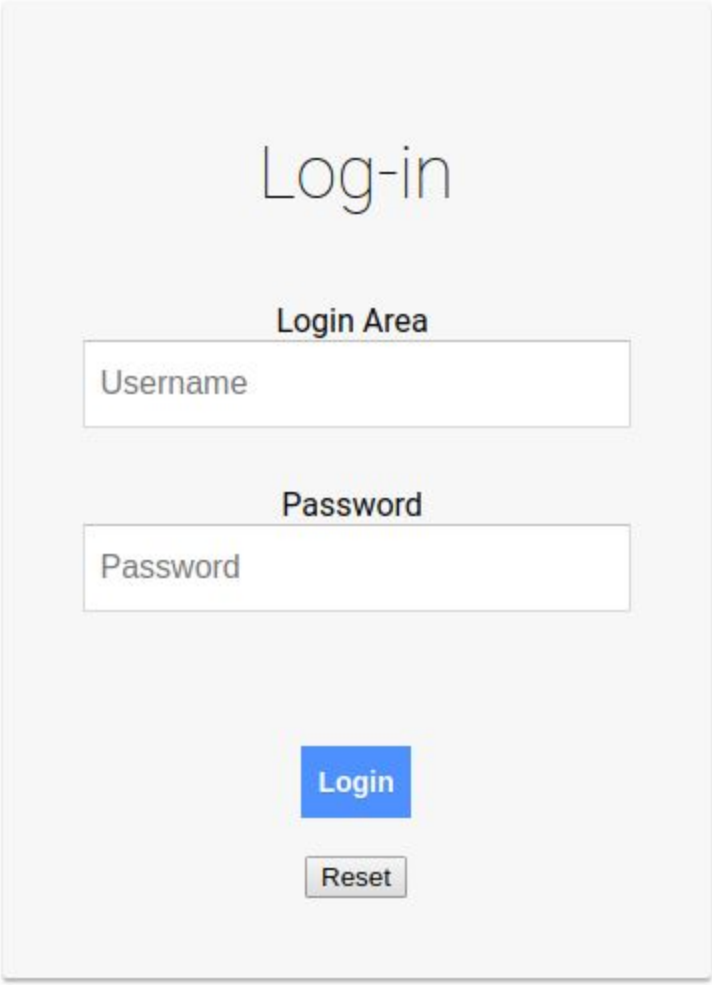
Web: Xoring

In this challenge we are asked to bypass the authentication interface, without knowing any info about the target user. However, the description says that this is not “essential”. So, let’s have a look on the app:



If we try to insert random credentials we receive an error message, as expected. We can study the sources of the page, hoping to find any clue. Inside we can find a file called “script.js”.

However, it's not clear at all what it is doing. This is a common practice on the web, called *javascript obfuscation*. Let's use an online tool in order to obtain a clearer version.

```
'use strict';
/**
 * @param {!Object} form
 * @return {undefined}
 */
function pasuser(form) {
  if (form.id.value == "admin") {
    if (x(form.pass.value, "6") == "\u007fNeAM+bh_saaES_mFlSYYu}nYw\u001d}") {
      /** @type {string} */
      location = "success.html";
    } else {
      alert("Invalid password/ID");
    }
  } else {
    alert("Invalid UserID");
  }
}
/** @type {!Array} */
var _0x8d99 = ["", "fromCharCode", "length", "substr"];
/**
 * @param {?} g
 * @param {string} o
 * @return {?}
 */
function x(g, o) {
  /** @type {!Array} */
  var key = [];
  var ret = _0x8d99[0];
  /** @type {number} */
  z = 1;
  for (; z <= 255; z++) {
    /** @type {number} */
    key[String[_0x8d99[1]](z)] = z;
  }
  /** @type {number} */
  j = z = 0;
  for (; z < g[_0x8d99[2]]; z++) {
    ret = ret + String[_0x8d99[1]](key[g[_0x8d99[3]](z, 1)] ^ key[o[_0x8d99[3]](j, 1)]);
    /** @type {number} */
    j = j < o[_0x8d99[2]] ? j + 1 : 0;
  }
  return ret;
}
;
```

If we use the value **admin** as a username, the code checks the password that we insert with a specific password (ciphered). However, our inserted password goes first through a function called **x**, with a value **6**. It seems like an encryption algorithm. Here we have two choices:

1. Hope that this is a symmetric encryption;
2. Define a reversing algorithm.

The first option seems faster, so we can try it. The idea is that, given the fact that we know the encryption key (6), if it is symmetric we can just use the same function with the ciphered password. Go on the console and write:

```
x("_NeAM+bh_saaES_mFlSYYu}nYw\u001d}", "6")
```

The output is:

"iNSA{+ThisWasSimpleYouKnow+}"

Luckily our first consideration was correct, and it saved some time.