


Beadandó feladat

Készíts MySQL adatbázist a következő szerkezettel (adatbázis neve: „adatok” tábla neve: „tabla”):

Sor 	Username	Titkos
1	katika@gmail.com	piros
2	arpi40@freemail.hu	zold
3	zsanettka@hotmail.com	sarga
4	hatizsak@protonmail.com	kek
5	terpeszterez@citromail.hu	fekete
6	nagysanyi@gmail.hu	feher

(nem használtam ékezeteket, hogy véletlenül se legyen kódolási probléma)

Készíts egy **esztétikus** PHP oldalt, amely egy formban bekéri a felhasználói nevet (Username) és a jelszót (echo nélkül). A bevitt információkat GET vagy POST tömb segítségével juttasd át a szerver oldalra.

A szerver oldalon található egy password.txt állomány (mellékelve). A fileban az adatbázisban is megtalálható Username található egy * karakter majd a hozzá tartozó jelszó. A sorok szabványos Linux EOL -al vannak lezárva azaz csak egy 0A byte-al (azaz 13-as karakter nincs) pl.:

```
katika@gmail.com*jelszó1
arpi40@freemail.hu*jelszó2
zsanettka@hotmail.com*jelszó3
...
```

Természetesen a valódi, titkos jelszavak vannak az állományban.

A password.txt-ben azért nem látjuk a fenti sorokat közvetlenül, mert le van titkosítva az egész állomány. Egy jelszó állományt illik ugyanis titkosítani. Így néz ki:

password.txt ✖	
00000000	70 53 93 60 6E 66 32 86 64 64 6E 5E 4D 5A 72 72 1C 8A 58 77
00000014	6E 55 80 2F 38 0A 66 64 8F 60 37 35 32 85 69 68 6A 5F 80 60
00000028	6F 33 5A 94 21 73 74 5E 88 67 0A 7F 65 80 65 68 79 66 8A 58
0000003c	43 6D 61 93 64 64 6E 5E 4D 5A 72 72 1C 82 6A 6C 71 5E 80 5E
00000050	34 37 0A 6D 53 93 60 7D 78 53 8A 37 73 77 61 93 66 71 72 53
00000064	88 63 31 68 61 8C 21 77 77 53 82 62 6C 73 59 0A 79 57 91 6F
00000078	68 78 6C 93 5C 75 6A 6C 5F 5A 6C 79 64 8E 64 64 6E 5E 4D 5F
0000008c	78 2F 55 94 62 72 77 5D 80 0A 73 53 86 70 76 66 60 98 60 43
000000a0	6C 5F 80 60 6F 33 5A 94 21 6D 7A 5E 82 6A 6C 70 53 0A

Látod benne a 0A bájtokat ?! Mindig az a sor vége jel.

A titkosítási algoritmus nagyon egyszerű, mégis hatékony. A karakterek bájtonként egy-egy számértékkel vannak eltolva. A megoldó kulcs ez az öt szám: **5,-14,31,-9,3** amely értékekkel el lettek tolva karakterek kódjai azaz ezek a számok lettek sorba hozzáadva az állomány karakter kódjaihoz. Ezekkel a számokkal kell „visszatekerni” a karaktereket sorba az 1. - 5. számig, majd ismét az 1. -től kezdve körbe – körbe. Az EOL (A0) természetesen nincs kódolva. Minden sorban

előről kezdődik a kódolás , tehát minden sor első karaktere az 5 számmal van eltolva a második karakter pedig a -14 el és így tovább.

Tehát. Az első bájt 70 hexában (az ábrán láthatod) azaz 112 decimálisan. A megoldókulcs első bájtja 5 . Ha 5-öt kivonok a 112 -ből akkor 107 lesz ami a "k" betű az ASCII kódtáblázatban. És lám a feladatban is a "katika@gmail.com" -al kezdődik az állomány aminek az első betűje "k". Tehát tuti egyszerű a kikódolás. Nehogy kézzel csináld! Erre kell a php program. (Képzeld el , hogy több ezer felhasználó is lehet)

A PHP program tehát azonosítsa, hogy a form-ba beolvasott felhasználói névhez a (password.txt alapján) megfelelő jelszót adták-e meg. Ha igen akkor az adatbázisból olvassa ki hogy az illetőnek mi a kedvenc színe és a megfelelő színű korong vagy egyéb kép állomány jelenjen meg, vagy a háttér legyen olyan színű, mindegy. Amennyiben rossz felhasználói nevet adott meg a „nincs ilyen felhasználó” hibaüzenetet írja ki. Ha rossz jelszót adott meg akkor „hibás jelszó” üzenetet adjon és 3 másodperc múlva dobja át a felhasználót a police.hu -ra.

A helyes jelszavakat felesleges megadnom, hiszen benne vannak a password.txt-ben és ha a felhasználó nevet ki tudta kódolni, akkor a jelszavakat is. Így tudod tesztelni is a programod helyességét.

Nem nagy kódú, egyszerű, gyorsan megírható mégis sok kompetenciát és hasznos technológiákat igénylő feladat. Ha nincs teljesen kész akkor is értékelhető és alkalmas a megfelelő differenciálásra (1-5 jegy).

A megoldást töltsd fel bármilyen freeweb szerverre működőképesen. (A shrek mysql-éhez külön engedélyt kell kérni, sajnos nem jár automatikusan, erre nem lesz most idő szerintem).

Jó munkát!