DAM2/ASIX2 - PHP: UF1

UF1 - Entrega #1

05 de Novembre de 2020

Versió 1.0

# Control de Versions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versió | Comentaris | Professor | Curs | Data |
| 1.0 | Primera versió | David González | Alumnes DAM2/ASIX2 | 05/11/2020 |

# Índex

[Control de Versions 2](#_Toc55542153)

[Índex 3](#_Toc55542154)

[1 Pràctica #1 – Formulari de Login Hardcoded 4](#_Toc55542155)

[1.1 Enunciat 4](#_Toc55542156)

[1.2 Requeriments 4](#_Toc55542157)

[1.3 Entrega 5](#_Toc55542158)

# Pràctica #1 – Formulari de Login Hardcoded

## Enunciat

Enhorabona, finalment has aconseguit entrar i avui és el teu primer dia com a desenvolupador junior PHP en una important multinacional del sector de les telecomunicacions. Diuen que els inicis no són fàcils, però de ben segur que el teu ... no serà una excepció. El teu responsable no dubte el més mínim de la professionalitat dels responsables de RRHH, però com ja saps, més val prevenir que curar, així que t’ha encarregat una sèrie de tasques les quals supervisarà directament per tal de validar les teves aptituds i així confirmar la idoneïtat del teu perfil per la vacant que ara ocupes.

La teva missió és crear un login d’usuaris que compleixi amb els següents requeriments tècnics.

## Requeriments

Contindrà el formulari de login considerant el següent:

1. Cal incloure els camps d’email i password.
2. Cal incloure un enllaç per recuperar la contrasenya (però no cal implementar res de moment).
3. Cal que la pàgina sigui web responsive.
4. El teu cap espera un estil visual sorprenent i creatiu, però com ja se sap, no cal tampoc reinventar la roda, així que et recomana visitar 🡪 <https://codepen.io/> i inspirar-te en alguna de les solucions existents.
5. Cal que la validació del formulari es realitzi dins del propi index.php
6. Cal que adoptis unes mínimes mesures de seguretat que evitin el XSS.
7. Et demana que transformis l’email i el password a BASE64 mitjançant la funció **btoa()** de Javascript abans d’enviar el formulari. Posteriorment, el codi PHP de validació caldrà que els decodifiqui via la funció **base64\_decode()**: <https://www.php.net/manual/es/function.base64-decode.php>
8. La validació del password es farà contra els valors de hash desats al servidor invocant la funció **password\_verify()**: <https://www.php.net/manual/es/function.password-verify.php>
9. Per la validació, caldrà fer ús d’un array de constants amb les següents parelles clau-valor:
   1. admin@educem.com 🡪 “iloveu“
   2. donald@educem.com 🡪 ”m4k3Am3r1caGr3atAg41n!”
   3. gilete@educem.com 🡪 “ErF4ryS1empr3”
   4. gon@educem.com 🡪 “Fatality!”
10. Com pots observar se’t donen les contrasenyes en clar, però únicament les pots fer servir per calcular el hash que hardcodejaràs en el teu PHP per validar-les. Per tal d’obtenir els valors de hash equivalents per cadascuna d’elles, t’hauràs de fer un petit script en PHP que t’ho generi i que faci ús de la funció **password\_hash**(): <https://www.php.net/manual/es/function.password-hash.php>
11. Si l’autenticació falla, caldrà mostrar una alerta mitjançant javascript a l’usuari i resetejar els valors dels camps email i password en el formulari als seus valors inicials.
12. Si l’autenticació és correcta, caldrà redirigir l’usuari a la web de l’educem 🡪 <https://educem.com>
13. **BONUS TRACK:** Evita que l’usuari percebi la codificació que fas a base64 del camps email i password ni que sigui per un segon mentre el formulari s’envia.

## Entrega

Com a bon professional, una vegada completada la pràctica, pujaràs tots els fitxers necessaris a un repositori de GitHub públic i en facilitaràs l’enllaç al teu responsable a través del Moodle.

És important que hi sigui tot, o del contrari, quan ens baixem el repositori no ho podrem posar en funcionament!