Password Protection

1. Encrypt with SHA-2
2. Auto complete will be off
3. <?php  
   $hashed\_password = crypt('mypassword'); // let the salt be automatically generated  
     
   /\* You should pass the entire results of crypt() as the salt for comparing a  
      password, to avoid problems when different hashing algorithms are used. (As  
      it says above, standard DES-based password hashing uses a 2-character salt,  
      but MD5-based hashing uses 12.) \*/  
   if (hash\_equals($hashed\_password, crypt($user\_input, $hashed\_password))) {  
      echo "Password verified!";  
   }  
   ?>

Form Protection

1. Validate image (mime type)
2. Validate Input fields (example : text box , area box , hidden field)
3. Validate SQL Injection
4. Validate JavaScript Injection(Cross site Scripting)
5. Filter metacharacters
6. FILTER\_SANITIZE\_FULL\_SPECIAL\_CHARS , FILTER\_SANITIZE\_STRING
7. is\_numeric(),filter\_var()
8. CSRF protection (page token)
9. Generate Token

session\_start();

if (empty($\_SESSION['token'])) {

$\_SESSION['token'] = bin2hex(random\_bytes(32));

}

$token = $\_SESSION['token'];

1. Validate website url for redirection – (Arabic / English ) – User can update other website url
2. JavaScript libraries
3. Be alert **Login page password-guessing attack**
4. **Broken Links**
5. **Content type**
6. **Done display IP address of website**

Cross Site Scripting \*\*\*  
SQL Injection\*\*  
File Upload\*

<https://www.wordfence.com/learn/how-to-prevent-cross-site-scripting-attacks/>