Τεχνολογίες Διαδικτύου

Εισαγωγή στη php,
Βάση δεδομένων,
Προσωρινή αποθήκευση πληροφοριών,
Θέματα ασφάλειας

Καθηγητής Χρήστος Δουληγέρης
Δρ. Ρόζα Μαυροπόδη

Αποθήκευση δεδομένων/πληροφοριών

Στην πλευρά του εξυπηρετητή:

1. ~~Αρχεία (τα είδαμε)~~
2. Βάση δεδομένων
3. Σύνοδοι

Αποθήκευση στη πλευρά του πελάτη:

1. Cookies
2. Προσωρινή αποθήκευση περιεχομένου (cashe)
3. WebStorage: localStorage, sessionStorage
4. Οι τεχνολογίες Web SQL database και indexedDB

Βάση Δεδομένων

Μια βάση δεδομένων αποτελεί μια **δομημένη** συλλογή από αρχεία ή δεδομένα που αποθηκεύονται σε ένα υπολογιστικό σύστημα και **οργανώνονται** με τέτοιο τρόπο ώστε να μπορεί να **αναζητηθούν** γρήγορα και οι πληροφορίες να μπορεί να **ανακτηθούν** γρήγορα.

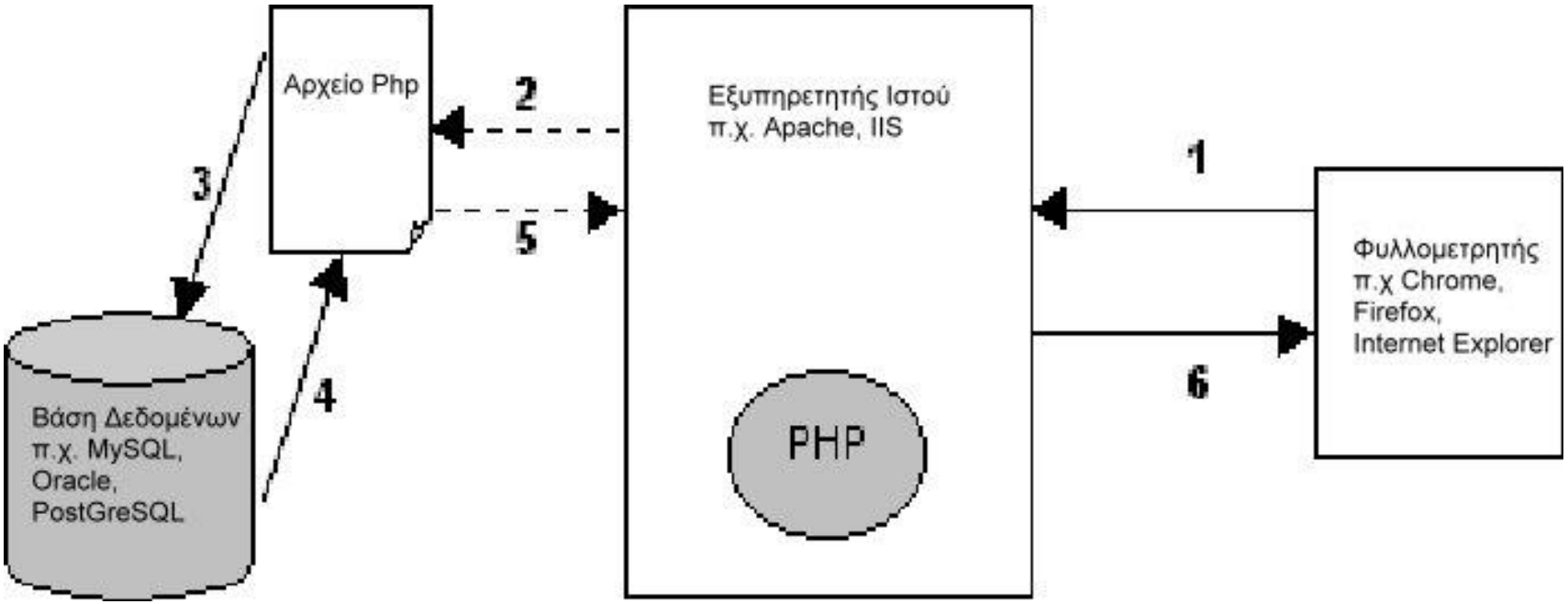
Βάση Δεδομένων

Αποθήκευση σε βάση δεδομένων όταν χρειάζεται:

- προβολή εξατομικευμένου περιεχομένου
- διαθεσιμότητά του περιεχομένου για σχετικά μεγάλο χρονικό διάστημα

Πώς γίνεται

Η λειτουργία της βάσης δεομένων στηρίζεται στο μοντέλο πελάτη/εξυπηρετητή



Επιπλέον

- Όταν χρησιμοποιούνται βάσεις αποθήκευσης οι πληροφορίες ενός δεδομένου τύπου αποθηκεύονται μία και μόνο φορά.
- ACID (atomicity, consistency, isolation, and durability) δηλαδή εξασφαλίζουν την ακεραιότητα, τη συνέπεια, τη διακριτότητα και τη διάρκεια των συναλλαγών
- Η ταυτόχρονη χρήση από πολλούς χρήστες δεν καταστρέφει τα δεδομένα
- Αποτελούν μηχανισμούς ανθεκτικούς σε σφάλματα, κυρίως λόγω του βαθμού ωριμότητας της εφαρμοζόμενης τεχνολογίας
- Οι βάσεις μπορεί να διαχειριστούν μεγάλου μεγέθους δεδομένα με αποδοτικό και γρήγορο τρόπο.

Γιατί MySQL

- Είναι ελεύθερο προς χρήση λογισμικό
- Πάνω από 10 εκατομμύρια εγκαταστάσεις η MySQL είναι ίσως το πιο δημοφιλές σύστημα διαχείρισης βάσης δεδομένων για εξυπηρετητές ιστού.
- Εξαιρετικά ένα ισχυρό και γρήγορο σύστημα διαχείρισης βάσεων δεδομένων

Php και MySQL

- Μια βάση, όπως η MySQL, χρησιμοποιείται για την αποθήκευση των πληροφοριών
- Η SQL για τη δημιουργία ερωτημάτων σε αυτήν,
- Η γλώσσα PHP χρησιμοποιείται για την μορφοποίηση των αποτελεσμάτων και την προβολή τους σε κάποια ιστοσελίδα.

Php και MySQL - Βήματα

Η διασύνδεση ενός προγράμματος PHP με μία βάση περιλαμβάνει τα παρακάτω βήματα:

1. Αρχικά πραγματοποιείται σύνδεση στην πλατφόρμα MySQL και επιλογή της επιθυμητής βάσης.
2. Δημιουργείται το ερώτημα που θα σταλεί στη βάση.
3. Εκτελείται το ερώτημα.
4. Ανακτώνται τα αποτελέσματα και μορφοποιούνται, ώστε να προβληθούν στο Διαδίκτυο
5. Επαναλαμβάνονται τα βήματα 2 έως 4 μέχρι να ανακτηθούν τα επιθυμητά δεδομένα
6. Διακόπτεται η σύνδεση με τη βάση.

Php και MySQL - Σύνδεση

Διάφορες ομάδες/ οικογένειες συναρτήσεων διαχείρισης βάσεων δεδομένων:

PHP's MySQL Extension: Οι συναρτήσεις είναι διαδικαστικές (procedural) και χρειάζονται προγραμματιστική διαφυγή (manual escaping). Εγκαταλείφθηκε από την έκδοση PHP 5 και μετά.

PHP's mysqli Extension: Αποτελεί μια εξελιγμένη αντικειμενοστρεφή έκδοση της προηγούμενης μεθόδου.

PHP Data Objects (PDO): Αποτελεί έναν πιο αφαιρετικό τρόπο αλληλεπίδρασης με κάποια βάση δεδομένων. Υποστηρίζει την MySQL, καθώς και άλλες βάσεις δεδομένων.

PHP's mysqli Extension

```
<?PHP
$user_name = "root";
$password = "";
$database = "addressbook";
$server = "127.0.0.1";
mysqli_connect($server, $user_name, $password ,
$database);
print "Σύνδεση με βάση";
?>
```

```php
<?php

//MySQL Extension //εγκαταλήφθηκε σπό την έκδοση 5.5
mysql_connect($server, $user_name, $password);
mysql_select($database);

//Data Objects (PDO)
PDO('mysql:host=example.com;dbname=database', 'user', 'password');

?>
```

Διαπιστευτήρια σύνδεσης - credentials

Τα διαπιστευτήρια αυτά περιλαμβάνουν: όνομα χρήστη της βάσης, συνθηματικό, όνομα της βάσης καθώς και το όνομα του μηχανήματος που είναι αποθηκευμένα τα δεδομένα.

Επειδή είναι συνηθισμένο ένας ιστότοπος να διαθέτει πολλαπλές σελίδες, οι οποίες με τη σειρά τους χρειάζονται να συνδέονται με τη βάση δεδομένων, αποτελεί καλή πρακτική τα διαπιστευτήρια αυτά να αποθηκεύονται σε κάποιο αρχείο, π.χ. στο login.php,

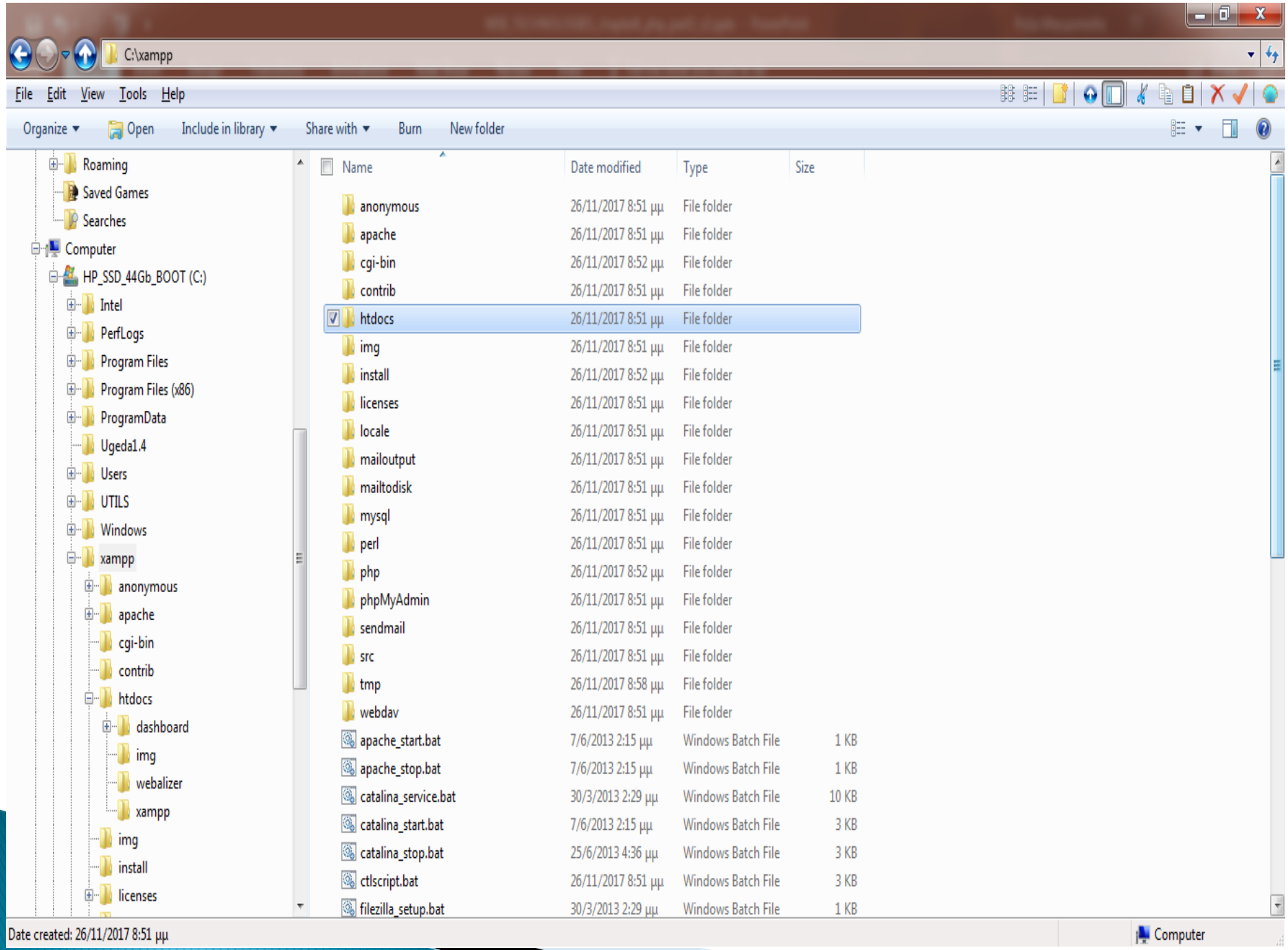
Ασφάλεια διαπιστευτηρίων

Αποτελεί καλή πρακτική οι μεταβλητές των διαπιστευτηρίων να έχουν οριστεί ως στατικές, οι οποίες δεν μπορούν να αλλάξουν τιμή παρά μόνον όταν ορίζονται. Οπότε έστω ότι το αρχείο login.php έχει τα παρακάτω περιεχόμενα. Το αρχείο αυτό αποθηκεύεται μακριά από το WEB_ROOT :

```
define('DB_PASS', 'password');
define('DB_USER', 'user_name');
define('DB_BASE', 'database');
define('DB_SERVER', 'server_url');
```

Αρχείο διαπιστευτηρίων

```
<?php // login.php
/*Το αναγνωριστικό του εξυπηρετητή, στον οποίο είναι αποθηκευμένη η βάση. Όταν πρόκειται για το τοπικό μηχάνημα, δίνεται το localhost ή 127.0.0.1 */
define('DB_SERVER', 'localhost');
/*Το όνομα της βάσης που θα χρησιμοποιηθεί */
define('DB_BASE', 'publications');
/*Το όνομα και το συνθηματικό του χρήστη που θα πραγματοποιήσει τη σύνδεση */
define('DB_USER', 'username');
define('DB_PASS', 'password');
?>
```


C:\xampp
File Edit View Tools Help
Organize Open Include in library Share with Burn New folder
Roaming
Saved Games
Searches
Computer
HP_SSD_44Gb_BOOT (C:)
Intel
PerfLogs
Program Files
Program Files (x86)
ProgramData
Ugeda1.4
Users
UTILS
Windows
xampp
anonymous
apache
cgi-bin
contrib
htdocs
dashboard
img
webalizer
xampp
img
install
licenses
Name
Date modified
Type
Size
anonymous 26/11/2017 8:51 μμ File folder
apache 26/11/2017 8:51 μμ File folder
cgi-bin 26/11/2017 8:52 μμ File folder
contrib 26/11/2017 8:51 μμ File folder
htdocs 26/11/2017 8:51 μμ File folder
img 26/11/2017 8:51 μμ File folder
install 26/11/2017 8:52 μμ File folder
licenses 26/11/2017 8:51 μμ File folder
locale 26/11/2017 8:51 μμ File folder
mailoutput 26/11/2017 8:51 μμ File folder
mailtodisk 26/11/2017 8:51 μμ File folder
mysql 26/11/2017 8:51 μμ File folder
perl 26/11/2017 8:51 μμ File folder
php 26/11/2017 8:52 μμ File folder
phpMyAdmin 26/11/2017 8:51 μμ File folder
sendmail 26/11/2017 8:51 μμ File folder
src 26/11/2017 8:51 μμ File folder
tmp 26/11/2017 8:58 μμ File folder
webdav 26/11/2017 8:51 μμ File folder
apache_start.bat 7/6/2013 2:15 μμ Windows Batch File 1 KB
apache_stop.bat 7/6/2013 2:15 μμ Windows Batch File 1 KB
catalina_service.bat 30/3/2013 2:29 μμ Windows Batch File 10 KB
catalina_start.bat 7/6/2013 2:15 μμ Windows Batch File 3 KB
catalina_stop.bat 25/6/2013 4:36 μμ Windows Batch File 3 KB
ctlscript.bat 26/11/2017 8:51 μμ Windows Batch File 3 KB
filezilla_setup.bat 30/3/2013 2:29 μμ Windows Batch File 1 KB
Date created: 26/11/2017 8:51 μμ
Computer

Σύνδεση με βάση

```php
<?php
//Κλήση αρχείου διαπιστευτηρίων
require_once 'login.php';
//Δημιουργία σύνδεσης
$conn = mysqli_connect(DB_SERVER,DB_USER,DB_PASS,DB_BASE);
//Έλεγχος σύνδεσης
if (!$conn) {
        die("Connection failed: " .mysqli_connect_error());
}
/*ορισμός charset της σύνδεσης ώστε να παρουσιάζονται τα ελληνικά σωστά*/
mysqli_set_charset($conn, "utf8");
?>
```

Εκτέλεση ερωτήματος

```php
<?php
// Δημιουργία ερωτήματος αναζήτηση
$query = "SELECT * FROM classics";
// Εκτέλεση ερωτήματος στη βάση
$result = mysqli_query($conn, $query);
//Έλεγχος αποτελεσμάτων
if (!$result)
                die(mysqli_connect_error());
?>
```

Ανάγνωση Δεδομένων

Υπάρχουν τέσσερεις διαφορετικοί τρόποι ανάγνωσης δεδομένων.

- `mysqli_fetch_array()`
- `mysqli_fetch_row()`
- `mysqli_fetch_assoc()`
- `mysqli_fetch_object()`

Και οι τέσσερεις θα φέρουν/διαβάσουν μια γραμμή από τη βάση στη συνέχεια την επόμενη κ.ο.κ. Έτσι συνδυάζονται με κάποιον βρόχο, ώστε να εμφανιστούν όλα τα δεδομένα ενός πίνακα.

mysql_fetch_row()

```
$query = mysqli_query ("select * from table_name");
while ($fetch=mysqli_fetch_row($query)) {
echo $fetch[0];
//Εκτυπώνει την πρώτη στήλη της γραμμής
echo $fetch[1];
//Εκτυπώνει τη δεύτερη στήλη της γραμμής
}
```

mysql_fetch_assoc()

```
$query = mysqli_query ("select * from table_name");

while ($fetch=mysqli_fetch_assoc($query)){
echo $fetch['name_of_fisrt_col'];
//Εκτυπώνει την πρώτη στήλη της γραμμής
echo $fetch['name_of_second_col'];
 //Εκτυπώνει τη δεύτερη στήλη της γραμμής
}
```

mysql_fetch_array()

```
$query = mysqli_query ("select * from table_name");
while ($fetch=mysqli_fetch_array($query)) {
echo $fetch['name_of_fisrt_col'];
//Εκτυπώνει την πρώτη στήλη της γραμμής
echo $fetch[1];
//Εκτυπώνει τη δεύτερη στήλη της γραμμής }
```

mysql_fetch_object()

```
$query = mysqli_query ("select * from table_name");

while($fetch=mysqli_fetch_object($query)) {
echo $fetch->'name_of_fisrt_col';
//Εκτυπώνει την πρώτη στήλη της γραμμής
echo $fetch->'name_of_second_col';
//Εκτυπώνει τη δεύτερη στήλη της γραμμής }
```

mysqli_num_rows()

Μια χρήσιμη συνάρτηση είναι και η **mysqli_num_rows** η οποία επιστρέφει τον αριθμό των γραμμών που περιλαμβάνονται σε ένα σύνολο αποτελεσμάτων.

mysqli_num_rows()

```php
<?php
//Ελέγχει εάν τα αποτελέσματα περιέχουν κάποια γραμμή περιεχομένων
if (mysqli_num_rows($result) > 0) {
// Εκτύπωση των αποτελεσμάτων ανά γραμμή
while($row=mysqli_fetch_assoc($result)){
        echo "Συγγραφέας: " . $row["author"]. " – Τίτλος: " .
        $row["title"]. " " . $row["year"]. " " . $row["isbn"]."<br>";
} else {
        echo "0 εγγραφές βρέθηκαν";
} }
?>
```

Αποσύνδεση

```
<?php
/* απελευθέρωση μεταβλητής
αποτελεσμάτων */
mysqli_free_result($result);
//αποδέσμευση μεταβλητής σύνδεσης
mysqli_close($conn);
?>
```

Αποθήκευση δεδομένων/πληροφοριών

Στην πλευρά του εξυπηρετητή:

1. ~~Αρχεία (τα είδαμε)~~
2. ~~Βάση δεδομένων (τα είδαμε)~~
3. Σύνοδοι

Αποθήκευση στη πλευρά του πελάτη:

1. Cookies
2. Προσωρινή αποθήκευση περιεχομένου (cashe)
3. WebStorage: localStorage, sessionStorage
4. Οι τεχνολογίες Web SQL database και indexedDB

Σύνοδοι- Sessions

Η ανάγκη αποθήκευσης εξατομικευμένων πληροφοριών του χρήστη σε ασφαλές περιβάλλον δημιούργησαν την τεχνική των συνόδων.

Το εγγενές πρόβλημα της αποθήκευσης οποιασδήποτε ευαίσθητης πληροφορίας σε τοπικό μηχάνημα είναι ότι μπορεί να παραβιαστεί κατά τη βούληση του χρήστη.

Σύνοδοι - Πώς γίνεται

Η χρήση των συνόδων περιλαμβάνει τα παρακάτω βήματα:

1. Αρχικά στον εξυπηρετητή δημιουργείται ένα αναγνωριστικό συνόδου και αποστέλλεται στο χρήστη.
2. Το αναγνωριστικό αποθηκεύεται στο τοπικό μηχάνημα του χρήστη
3. Αυτό το αναγνωριστικό αποστέλλεται στον εξυπηρετητή κάθε φορά που ο συγκεκριμένος χρήστης έχει κάποιο αίτημα
4. Ο εξυπηρετητής χρησιμοποιεί αυτό το αναγνωριστικό ώστε να ανακτήσει τις ατομικές πληροφορίες του χρήστη

Σύνοδοι λειτουργικά χαρακτηριστικά

- Το αναγνωριστικό της συνόδου αποθηκεύεται στο τοπικό μηχάνημα.
- Διαχειρίζεται ως cookie και καλείται προκαθορισμένα PHPSESSIONID.
- Εναλλακτικά, μπορεί να μεταβιβάζεται ενσωματωμένο στο URL, το οποίο δεν εφαρμόζεται συχνά στις μέρες μας.
- Το αναγνωριστικό έχει χρονικό όριο λήξης. Συνήθως το κλείσιμο του παραθύρου του φυλλομετρητή
- Οι πληροφορίες της συνόδου στον εξυπηρετητή αποθηκεύονται είτε σε κάποιο αρχείο ή σε πίνακα σε βάση δεδομένων.
- Η php χρησιμοποιεί την υπερ-καθολική μεταβλητή πίνακα $_SESSION ώστε να ανακτήσει πληροφορίες συνόδου.

Σύνοδοι πλεονεκτήματα

Το γεγονός ότι δε μεταδίδονται οι πληροφορίες μέσω του Διαδικτύου αλλά το αναγνωριστικό τους, κάνει τη χρήση των συνόδων σχετικά ασφαλή.

Επίσης δεν υπάρχει περιορισμός στο μέγεθος των πληροφοριών που μπορεί να αποθηκευτούν, επειδή η αποθήκευση λαμβάνει χώρα στη πλευρά του εξυπηρετητή

Σύνοδοι παράδειγμα

```
<?php
session_start();
/*εκτύπωση αναγνωριστικού συνόδου */
echo "Your session id is " . session_id();
//αποθήκευση τιμής μεταβλητής
$_SESSION['var'] = $val;
// αποθήκευση άμεσων δεδομένων
$_SESSION['FirstName'] = "Jim";
$temp=$_SESSION['FirstName']; // ανάγνωση τιμής
print $temp; //θα εκτυπώσει την τιμή "Jim"
// έλεγχος ύπαρξης στοιχείου της συνόδου
if (isset($_SESSION['FirstName'])) { // κάνε κάτι }
// καταστροφή του στοιχείου FirstName και των δεδομένων που περιέχει.
unset($_SESSION['FirstName']);
                    // κλείσιμο της συνόδου
                    session_destroy( );
                                                    ?>
```

Αποθήκευση δεδομένων/πληροφοριών

Στην πλευρά του εξυπηρετητή:

1. ~~Αρχεία (τα είδαμε)~~
2. ~~Βάση δεδομένων (τα είδαμε)~~
3. ~~Σύνοδοι~~

Αποθήκευση στη πλευρά του πελάτη:

1. Cookies
2. Προσωρινή αποθήκευση περιεχομένου (cashe)
3. WebStorage: localStorage, sessionStorage
4. Οι τεχνολογίες Web SQL database και indexedDB

Cookie

Τα cookies αποτελούν μικρές ‘ποσότητες’ δεδομένων (έως 4KB) τα οποία δημιουργούνται από την εφαρμογή και αποθηκεύονται στο φυλλομετρητή του χρήστη.

Η δημιουργία τους προέκυψε από την ανάγκη να αποθηκεύονται τοπικά πληροφορίες, για τον χρήστη και την επίσκεψή του. Με τον τρόπο αυτό, ο ιστότοπος θυμάται τις ενέργειες και τις προτιμήσεις (όπως κωδικός σύνδεσης, γλώσσα, μέγεθος γραμματοσειράς και άλλες προτιμήσεις απεικόνισης) για ένα χρονικό διάστημα, έτσι δεν χρειάζεται να εισάγονται οι προτιμήσεις αυτές κάθε φορά που επισκέπτεται ο χρήστης τον ιστότοπο.

Ποιος έχει πρόσβαση στις πληροφορίες τους

Τα cookies απέκτησαν άσχημη φήμη επειδή επιτρέπουν την καταγραφή πληροφοριών του επισκέπτη, π.χ. πόσες φορές έχει επισκεφτεί έναν ιστοτόπο, τι τον ενδιέφερε, τι αγοράζει συχνά κλπ.

Εξαιτίας της φύσης των πληροφοριών που περιέχουν και τις συνέπειές τους στην ιδιωτική ζωή και προσπαθώντας να αποτρέψουν τη χρήση/ανάγνωσή τους από μη εξουσιοδοτημένους τομείς, υλοποιήθηκε η πολιτική της ιδίας προέλευσης (**Same Origin Policy** – SOP), σύμφωνα με την οποία τα cookies μπορεί να διαβαστούν μόνο από τον τομέα που τα εξέδωσε.

Ποιος έχει πρόσβαση στις πληροφορίες τους

Με άλλα λόγια, εάν ένα cookie εκδίδεται από, για παράδειγμα, unipi.gr, μπορεί να ανακτάται μόνο από τον εξυπηρετητή ιστού χρησιμοποιώντας αυτόν τον τομέα. Αυτό αποτρέπει άλλους δικτυακούς τόπους από το να αποκτήσουν πρόσβαση σε πληροφορίες για τις οποίες δεν έχουν τα κατάλληλα δικαιώματα.

Πότε στέλνονται

Τα cookies ανταλλάσσονται κατά τη διάρκεια της μεταφοράς των επικεφαλίδων του πρωτοκόλλου HTTP και πριν από την αποστολή οποιαδήποτε τμήματος HTML μιας ιστοσελίδας. Είναι αδύνατο να σταλεί ένα cookie από τη στιγμή που έχει μεταφερθεί ένα τμήμα HTML της ιστοσελίδας. Ως εκ τούτου, είναι σημαντικός ο προσεκτικός σχεδιασμός της χρήσης των cookies.

Η διαχείρισή τους

- Τοπικά με javascript.
- Απομακρυσμένα με php.

Ο ορισμός τους πραγματοποιείται με τη συνάρτηση setcookie

```
setcookie(name, value, expire, path,
domain, secure, httponly);
```

```
setcookie('username', 'rosa', time()
+ 60 * 60 * 24 * 7, '/');
```

Παράμετρος	Περιγραφή
name	Το όνομα του cookie. Κάθε μελλοντική αναφορά του εξυπηρετητή σε αυτό το cookie θα πραγματοποιείται με τη χρήση αυτού του αναγνωριστικού.
value	Η τιμή/περιεχόμενο του. Μπορεί να είναι έως 4 KB ανά cookie και οποιοδήποτε αλφαριθμητικό κείμενο. Συνηθέστερες τιμές είναι το username και η ημερομηνία τελευταία επίσκεψης.
Προαιρετικές παράμετροι	
expire	Μια αριθμητική τιμή σε δευτερόλεπτα, αρχίζοντας από 00:00:00 GMT την 1η Ιανουαρίου 1970. Μετά από αυτή τη χρονική στιγμή το cookie είναι απροσπέλαστο. Εάν δεν οριστεί, αυτή η παράμετρος, τότε ως λήξη ορίζεται το κλείσιμο του παραθύρου του φυλλομτρητή.
path	Καθορίζει τους καταλόγους, για τους οποίους είναι έγκυρο το cookie. Μια απλή / (κάθετος) επιτρέπει στο cookie να ισχύει για όλους τους καταλόγους. Η προκαθορισμένη τιμή είναι ο τρέχων κατάλογος, μέσα στον οποίο έχει δημιουργηθεί το cookie.
domain	Ο τομέας του Διαδικτύου του cookie. Για παράδειγμα εάν αυτό είναι .unipi.gr τότε το cookie είναι διαθέσιμο για: www.unipi.gr, images.unipi.gr. Εάν, για παράδειγμα έχει οριστεί για το images.unipi.gr τότε είναι διαθέσιμο για τους π.χ. sub.images.unipi.gr, αλλά όχι και για τον www.unipi.gr.
secure	Καθορίζει τον τρόπο μεταφοράς του cookie πάνω από το πρωτόκολλο επικοινωνίας http ή https. Εφόσον είναι ‘αληθής’ (1), η μεταφορά θα πρέπει να συμβεί μόνο με https. Η προεπιλεγμένη τιμή είναι ‘ψευδής’ (0) .
httponly	Ισχύει από την έκδοση PHP 5.2.0 και μετέπειτα. Καθορίζει εάν θα είναι προσβάσιμο με στις τεχνολογίες στην πλευρά του επισκέπτη, π.χ. javascript. Εάν είναι ‘αληθής’ τότε το cookie είναι απροσπέλαστο στην javascript. Η προεπιλεγμένη τιμή είναι ‘ψευδής’. Δεν υποστηρίζεται από όλους τους φυλλομετρητές.

Εφόσον έχει δημιουργηθεί κάποιο cookie δεν μπορεί να διαβαστεί/προσπελαστεί η τιμή του, παρά μόνον στην περίπτωση όπου έχει πραγματοποιηθεί ανανέωση της ιστοσελίδας, οπότε και ο φυλλοετρητής θα το αποστείλει πίσω στον εξυπηρετητή.

Όπως έχει προαναφερθεί όλα τα cookie περιέχονται στην υπερκαθολική μεταβλητή πίνακα $_COOKIE.

Έλεγχος:

```
if (isset($_COOKIE['username'])) $username
= $_COOKIE['username'];
```

Καταστροφή:

```
setcookie('username', 'rosa', time()
- 2592000, '/');
```

Αποθήκευση δεδομένων/πληροφοριών

Στην πλευρά του εξυπηρετητή:

1. ~~Αρχεία (τα είδαμε)~~
2. ~~Βάση δεδομένων (τα είδαμε)~~
3. ~~Σύνοδοι~~

Αποθήκευση στη πλευρά του πελάτη:

1. ~~Cookies~~
2. Προσωρινή αποθήκευση περιεχομένου (cashe)
3. WebStorage: localStorage, sessionStorage
4. Οι τεχνολογίες Web SQL database και indexedDB

Προσωρινή αποθήκευση- cashe

Η προσωρινή αποθήκευση cashe του φυλλομετρητή είναι μια προσωρινή τοποθεσία αποθήκευσης στον υπολογιστή σας για αρχεία που έχουν ληφθεί από το πρόγραμμα περιήγησής σας κατά την προβολή ιστότοπων.

Τα αρχεία που αποθηκεύονται προσωρινά σε τοπικό επίπεδο περιλαμβάνουν όλα τα έγγραφα που απαρτίζουν έναν ιστότοπο, όπως **αρχεία** html, CSS, JavaScript, καθώς και εικόνες, γραφικά και άλλο περιεχόμενο πολυμέσων.

cashe αντί storage

Η διαφορά με την αποθήκευση (storage) είτε σε τοπικό επίπεδο είτε στον εξυπηρετητή είναι ότι στην cashe αποθηκεύονται ολόκληρα αρχεία, απαραίτητα για τη λειτουργία της ιστοσελίδας.

Στην τεχνική της storage αποθηκεύονται δεδομένα, πληροφορίες σχετικές με τον κάθε χρήστη, μικρής, συνήθως, έκτασης

Γιατί ;

Γιατί χρησιμοποιείται:

- Η διαθεσιμότητα των εφαρμογών εκτός σύνδεσης.
- Η σύνδεση με το Διαδίκτυο μπορεί να είναι ασταθής.
- Παλαιότερα ο χρήστης θα έπρεπε να αποθηκεύει κάθε ιστοσελίδα μεμονωμένα.

Η HTML5 δημιούργησε και χρησιμοποιεί ένα javascript API (application programming interface) το Application Cache

Πώς λειτουργεί

Αρχικά ορίζονται οι κανόνες αποθήκευσης, δηλαδή ποια αρχεία θα αποθηκεύονται και ποια όχι.

Στη συνέχεια συνδέεται το αρχείων των κανόνων αυτών με το αρχείο HTML της ιστοσελίδας όπως φαίνεται παρακάτω.

```
<html manifest="manifest.appcache">
.
.
.</html>
```

Αρχείο .appcache

CACHE MANIFEST
CACHE:
εισαγωγή σχολίων στο αρχείο manifest.
index.html
stylesheet.css
images/masthead.png
scripts/misc.js
NETWORK:
search.php
login.php
/api

FALLBACK:
images/dynamic.php static_image.png

Αποθήκευση δεδομένων/πληροφοριών

Στην πλευρά του εξυπηρετητή:

1. ~~Αρχεία (τα είδαμε)~~
2. ~~Βάση δεδομένων (τα είδαμε)~~
3. ~~Σύνοδοι~~

Αποθήκευση στη πλευρά του πελάτη:

1. ~~Cookies~~
2. ~~Προσωρινή αποθήκευση περιεχομένου (cashe)~~
3. WebStorage: localStorage, sessionStorage
4. Οι τεχνολογίες Web SQL database και indexedDB

WebStorage (: localStorage, sessionStorage)

- Η τεχνολογία Web Storage παρέχει έναν απλό τρόπο αποθήκευσης πληροφοριών στη μορφή κλειδί=>τιμή.
- Η αποθήκευση είναι ‘ανθεκτική’ (persistent) και αυτό διότι οι πληροφορίες λήγουν ή καταστρέφονται μόνον εφόσον διαγραφούν προγραμματιστικά από την εφαρμογή ή κατ’ επιλογήν από το χρήστη του φυλλομετρητή.
- Το μέγεθος των πληροφοριών που μπορεί να αποθηκευτούν είναι περίπου 5 MB για κάθε τομέα (domain)
- Οι πληροφορίες αποθηκεύονται τοπικά και εάν και εφόσον υπάρχει δίκτυο, είναι δυνατός ο συγχρονισμός τους με τον εξυπηρετητή.

localStorage sessionStorage

Οι δύο αυτοί τύποι διαφέρουν ως προς τη διάρκεια διατήρησης και το εύρος πρόσβασης των αποθηκευμένων πληροφοριών.

- Στη localStorage η αποθήκευση είναι ‘ανθεκτική’ (persistent). Στη sessionStorage τα αποθηκευμένα δεδομένα διαγράφονται όταν κλείσει η καρτέλα ή το παράθυρο του φυλλομετρητή.
- Στην τεχνική localStorage, επιτρέπει την πρόσβαση σε κάθε παράθυρο ή καρτέλα του ίδιου φυλλομετρητή. Στη sessionStorage περιορίζει το εύρος πρόσβασης των αποθηκευμένων δεδομένων στο τρέχον παράθυρο του φυλλομετρητή.

localStorage

```
localStorage.setItem('fullname',
'rosa louki'); //αποθήκευση
alert("Your name is: " +
localStorage.getItem('fullname'));
//ανάκτηση
alert("Hello " +
localStorage.fullname);
// ανάκτηση εναλλακτικός τρόπος
localStorage.removeItem('fullname');
// διαγραφή συγκεκριμένης εγγραφής
```

sessionStorage

```
sessionStorage.setItem('fullname', '
rosa louki'//αποθήκευση
alert("Your name is: " +
sessionStorage.getItem('fullname'));
//ανάκτηση
alert("Hello " +
sessionStorage.fullname);
// ανάκτηση εναλλακτικός τρόπος
sessionStorage.removeItem('fullname')
; // διαγραφή συγκεκριμένης εγγραφής
```

Αποθήκευση δεδομένων/πληροφοριών

Στην πλευρά του εξυπηρετητή:

1. ~~Αρχεία (τα είδαμε)~~
2. ~~Βάση δεδομένων (τα είδαμε)~~
3. ~~Σύνοδοι~~

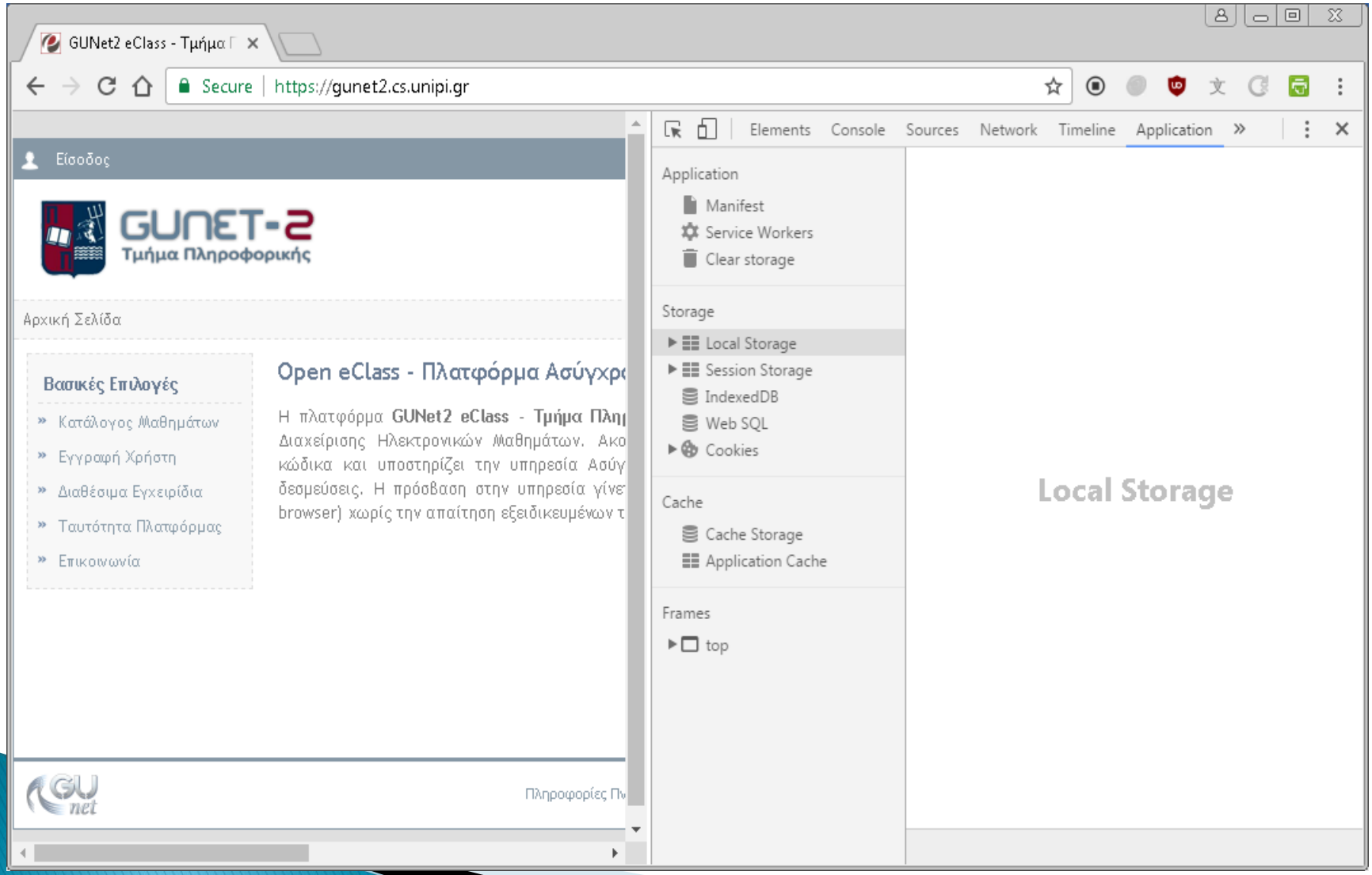
Αποθήκευση στη πλευρά του πελάτη:

1. ~~Cookies~~
2. ~~Προσωρινή αποθήκευση περιεχομένου (cashe)~~
3. ~~WebStorage: localStorage, sessionStorage~~
4. Οι τεχνολογίες Web SQL database και indexedDB

Web SQL database και indexedDB

Οι προηγούμενες τεχνολογίες (WebStorage) βασίζονται στην αποθήκευση των πληροφοριών με τη μορφή κλειδί->τιμή. Όταν απαιτείται η διαχείριση μεγάλου όγκου πληροφοριών, τότε είναι προτιμότερο η αποθήκευση τους να γίνει με έναν δομημένο τρόπο και η τυχαία ευρετηρίασή τους. Για το σκοπό αυτό χρησιμοποιούνται **βάσεις δεδομένων τοπικά εγκατεστημένες** στο μηχάνημα του επισκέπτη.

Προσωρινά αποθηκευμένες πληροφορίες



PHP και Ασφάλεια

Μερικές ιδέες που μπορεί να εφαρμοστούν ώστε να ενισχυθεί η ασφάλεια των διαδικτυακών εφαρμογών

Αποφυγή σύντομων εναρκτήριων ετικετών

Καλό θα είναι να αποφεύγεται η χρήση των σύντομων ετικετών καθώς ο κώδικας ερμηνευόταν εσφαλμένα ως προτάσεις XML.

```
<? echo "Να αποφεύγεται. Προκαλεί σύγχυση με κώδικα XML" ?>
<?php echo "Σωστός τρόπος" ?>
```

Επικύρωση δεδομένων

Κατά την 'παραλαβή' των δεδομένων μιας φόρμας θα πρέπει πάντα να ελέγχονται ότι είναι του αναμενόμενου τύπου. Δηλαδή εφόσον μια τιμή αναμένεται να είναι αλφαριθμητική τότε δε θα πρέπει να περιέχει σύμβολα !@#$%^&* κλπ

Εφόσον αναμένεται να είναι τύπου email τότε να είναι του τύπου user@domain.com κλπ.

Εφόσον πρόκειται για URL τότε να μην περιέχει σύμβολα όπως %.

Επίθεση Έγχυσης ερωτημάτων SQL - SQL injection

Είναι ένας τύπος επίθεσης ο οποίος προσπαθεί να εκτελέσει ενέργειες/αιτήματα σε μια βάση δεδομένων η οποία χρησιμοποιείται από τον εξυπηρετητή. Έστω ότι ισχύει:

```
$id = $_GET['id'];
mysqli_query(
"SELECT * FROM pages WHERE id = '$id'");
```

τότε θα μπορούσε να εκτελεστεί και το:

http://www.site.com/path/page.php?id=5; DROP TABLE members;

XSS (Cross site scripting)

Η επίθεση XSS (Cross site scripting) αποτελεί την προσπάθεια ενός κακόβουλου χρήστη να εισαγάγει τμήματα κώδικα, HTML ή Javascript, ώστε να επηρεάσει τη λειτουργία της ιστοσελίδας.

XSS (Cross site scripting)

```
<?php
$term = $_GET['term']; ?>
...
Προβολή στην οθόνη του :
<?php echo $term; ?>
```

τότε θα μπορούσε να εκτελεστεί και το:

http://...../search.php?term=<script>alert('hi');</script>

Προστασία αρχείων, συνθηματικών και συνόδων

- Ονομασία αρχείων ως .php και
- Κρυπτογράφηση συνθηματικών

```
$hash = md5($password);
```

ή ακόμα καλύτερα

```
$salt = 'οποιοδήποτε κείμενο';
$hash = md5($password . $salt);
```

- Αποθήκευση των συνόδων σε βάση δεδομένων

CSRF (Cross Site Request Forgery)

Η επίθεση CSRF (Cross Site Request Forgery) μοιάζει με την XSS, η οποία αναφέρθηκε νωρίτερα στο κεφάλαιο. Αποτελεί μια προσπάθεια εισαγωγής κώδικα στην πλευρά του πελάτη/χρήστη. Ο κώδικας είναι συνήθως Javascript.

```
<a href='http://www.original-application.com/delete_record.php?id=123' target='_blank' rel='noreferrer' >
<img src='http://www.original-application.com/update_record.php?id=123'>
</a>
```

Είστε σίγουροι ότι θέλετε να διαγράψετε .... κλπ

Αναδημιουργία του αναγνωριστικού της συνόδου (session ID)

Αποτελεί καλή πρακτική να γίνεται αναδημιουργία του αναγνωριστικού της συνόδου σύνδεσης σε συγκεκριμένα χρονικά διαστήματα ή μετά από ορισμένο αριθμό αιτημάτων. Η τακτική αυτή μπορεί να περιορίσει την 'ζημιά' στην περίπτωση όπου το αρχικό αναγνωριστικό της συνόδου έχει εκτεθεί.

```
session_regenerate_id();
//ή
session_regenerate_id(true);
```

Μοναδική σύνοδος

Η ύπαρξη μιας μοναδικής συνόδου ανά χρήστη επιτρέπει την αποτελεσματικότερη επίβλεψη και ασφάλισή της. Αρχικά ελέγχεται εάν ο χρήστης έχει πραγματοποιήσει συνδέσεις από διαφορετικές τοποθεσίες. Στη συνέχεια αποδεσμεύεται η μία από τις δύο, συνήθως η προηγούμενη/παλαιότερη. Η τακτική αυτή είναι χρήσιμη σε εφαρμογές οι οποίες ανταλλάσσουν εμπιστευτικά δεδομένα, για παράδειγμα σε έναν ιστότοπο ηλεκτρονικών αγορών.

Διαφορετικός χρήστης ανά εφαρμογή για σύνδεση με τη βάση δεδομένων

Κάθε εφαρμογή προκειμένου να συνδεθεί σε μία βάση δεδομένων χρειάζεται να χρησιμοποιήσει τα διαπιστευτήρια ενός νόμιμου χρήστη της βάσης. Ο χρήστης της βάσης δεδομένων, τον οποίο χρησιμοποιεί η εφαρμογή ώστε να συνδεθεί στη βάση, δεν πρέπει να έχει δικαιώματα ώστε να εκτελέσει εντολές ή να γράψει στο τοπικό σύστημα αρχείων του εξυπηρετητή.

Ανάγνωση του καταλόγου των φακέλων

Όταν ένας φυλλομετρητης προβάλλει έναν κατάλογο ιστοτόπου, ο οποίος δεν περιέχει αρχείο του είδους index.html (ή οποιοδήποτε άλλο αρχείο ευρετηρίου), τότε εμφανίζονται τα αρχεία του καταλόγου αυτού.