

# Προγραμματισμός & Συστήματα στον Παγκόσμιο Ιστό

Εργαστηριακή Άσκηση Ακαδημαϊκού Έτους 2022 – 2023

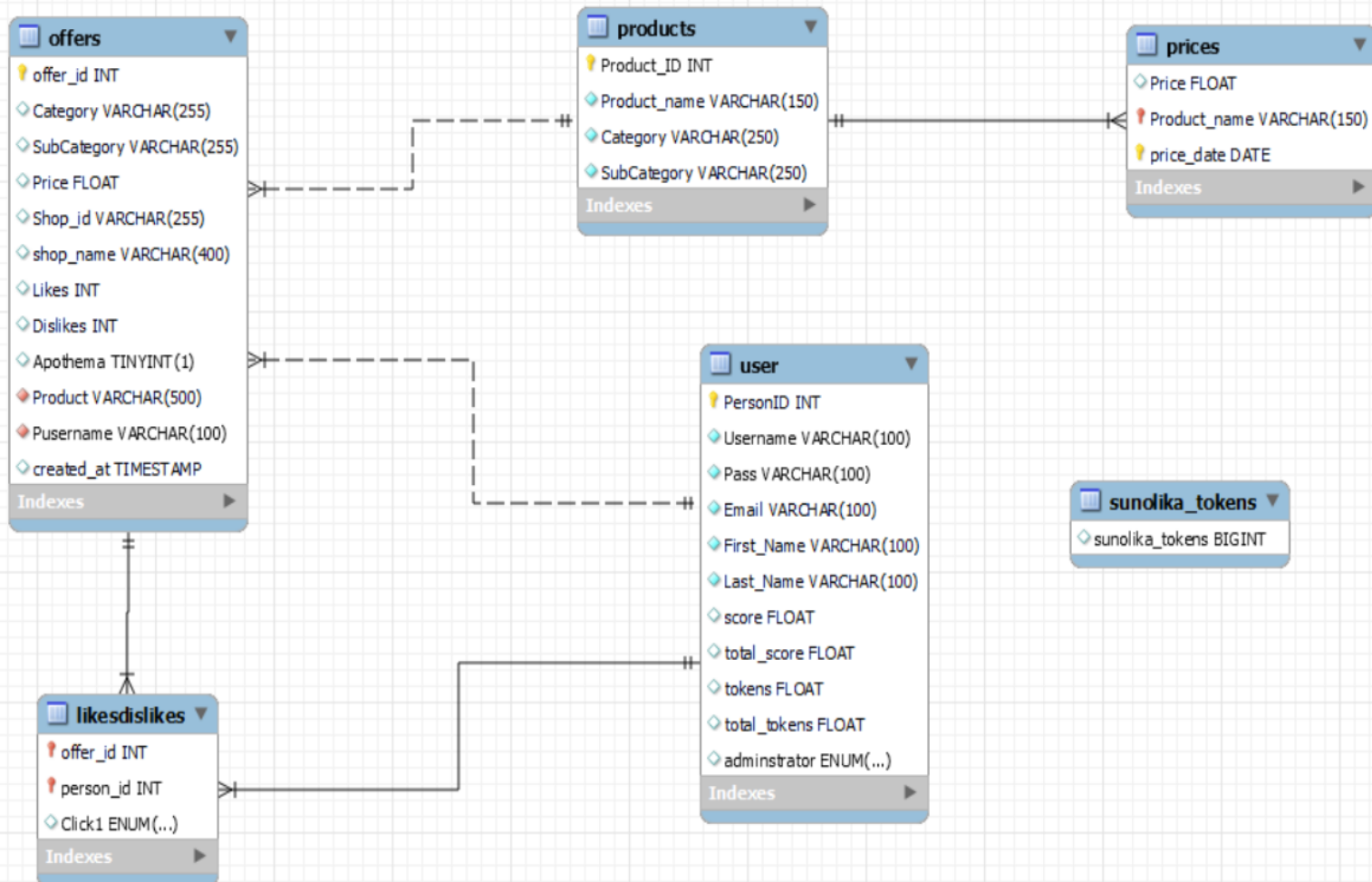
Αμίσης Ελευθέριος, AM: 1072464

Πίσσουρος Χάρης, AM: 1069673

Παπαχριστοφίλου Σαράντης, AM: 1072600

## Σχεδιασμός Βάσης

Για τη βάση δεδομένων χρησιμοποιήσαμε την MySQL και παρακάτω παρουσιάζεται το σχήμα της βάσης.



**user:** Πίνακας ο οποίος αφορά τους χρήστες της ιστοσελίδας και περιέχει όλα τα βασικά στοιχεία όπως το Username, ο κωδικός, το email και το ονοματεπώνυμο. Επίσης, περιέχει τα score και tokens των χρηστών, ενώ με τη χρήση του **administrator enum('admin','user') DEFAULT 'user'** διευκρινίζεται το εάν πρόκειται για απλό χρήστη ή για διαχειριστή.

**products:** Στον πίνακα αυτόν αποθηκεύονται τα προϊόντα, τα οποία ο διαχειριστής ανεβάζει στην ιστοσελίδα μέσω του json “products\_and\_categories.json”. Περιέχει το id το προϊόντος, το όνομά του, καθώς και τα ids της κατηγορίας και της υποκατηγορίας του.

**prices:** Στον πίνακα αυτόν αποθηκεύονται τα ονόματα των προϊόντων μαζί με τις τιμές τους σε συγκεκριμένες ημερομηνίες. Η εισαγωγή στοιχείων στον πίνακα prices γίνεται μέσω του διαχειριστή, ο οποίος ανεβάζει το αρχείο "prices.json".

**offers:** Στην ιστοσελίδα όλοι οι χρήστες έχουν το δικαίωμα να υποβάλλουν προσφορές. Κατά την υποβολή μίας προσφοράς τα στοιχεία του προϊόντος, του καταστήματος που έγινε η προσφορά, αλλά και του χρήστη που την υπέβαλλε αποθηκεύονται στον πίνακα αυτόν.

**likesdislikes:** Οι χρήστες έχουν το δικαίωμα να κάνουν Like ή Dislike σε μία προσφορά. Η δραστηριότητα αυτή αποθηκεύεται για κάθε χρήστη στον πίνακα LikesDislikes προκειμένου να γίνει η ανάκτηση τους και να προβάλλονται στο προφίλ του.

**sunolika\_tokens:** Πίνακας στον οποίο αποθηκεύεται ο συνολικός αριθμός tokens στο τέλος κάθε μήνα, με σκοπό να γίνει η διανομή του 80% αυτών στους χρήστες.

## Σύνδεση Βάσης με το σύστημα

Για την σύνδεση της βάσης με το υπόλοιπο σύστημα χρησιμοποιήσαμε το παρακάτω php αρχείο db.php

```
<?php

if($_SERVER ["REQUEST_METHOD"]=="POST" || "GET"){
$servername = "localhost";
$username = "root";
$password = "sarantis159263";
$dbname = "WEB";

$conn = mysqli_connect($servername,$username,$password,$dbname);
if ($conn)
{
    //echo "Connection Established! <br>";
}
else
{
    die("Connection failed");
}
}

?>
```

Το παραπάνω αρχείο κάθε φορά που χρειάζεται να έχουμε πρόσβαση στην βάση δεδομένων όπως φαίνεται παρακάτω:

```
include '../get_from_db/db.php';
```

- Όσον αφορά τις αλλαγές που πρέπει να γίνονται στη βάση έπειτα από συγκεκριμένο χρονικό διάστημα, αξιοποιήθηκαν τα EVENTS της MySQL. Συγκεκριμένα, για τον μηδενισμό των tokens και του score υπάρχουν τα events Zero\_Tokens και Zero\_Score αντίστοιχα. Η διαγραφή των προσφορών έπειτα από την παρέλευση μίας εβδομάδας γίνεται μέσω του Expired\_Offers, ενώ ο υπολογισμός των tokens και στη συνέχεια η διανομή τους στους χρήστες βάσει score γίνεται μέσω των Υpologise\_Tokens και Moirase Tokens.

## Ρυθμίσεις SERVER

- Το configuration που έγινε στο server για κατάλληλη χρήση της cache memory έγινε μέσω εντολών headers(). Συγκεκριμένα, χρησιμοποιήθηκαν οι παρακάτω εντολές:
  1. `header("Content-Type: application/json");` ==> Η εντολή αυτή χρησιμοποιήθηκε προκειμένου να διαπιστώσει ο client ότι το response που δέχτηκε είναι σε μορφή json.
  2. `header("Cache-Control: max-age:1800");` ==> Η εντολή αυτή χρησιμοποιήθηκε στο `getOffer.php` και `retrieve_products.php` με διαφορετικά max-ages, ανάλογα το χρονικό διάστημα που κρίναμε καταλληλότερο, προκειμένου να διατηρούνται τα responses στην cache του client και να γίνεται συντομότερη η φόρτωση του περιεχομένου.

Σκοπός της χρήσης των εντολών ήταν να διευκολύνουμε την φόρτωση των δεδομένων στο client για δεδομένα που δεν αλλάζουν τόσο συχνά(π.χ. προϊόντα) ή δεδομένα που δεν μας ενδιαφέρει η άμεση εμφάνιση τους. Παραθέτουμε σχετικό σκρίνσοτ, στο οποίο φαίνεται το max-age που έχουμε ορίσει:

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Headers	Cookies	Request	Response	Timings	Stack Trace
200	GET	localhost	GetOffer.php	client_map.js:52 (xhr)	json	2.87 kB	2.58 kB	Filter Headers					
200	GET	localhost	GetOffer.php	client_map.js:293 (xhr)	json	cached	2.58 kB	Response Headers (284 B)					
									Cache-Control: max-age=1800				
									Connection: Keep-Alive				
									Content-Length: 2584				
									Content-Type: application/json				
									Date: Tue, 19 Sep 2023 11:51:12 GMT				
									Expires: Thu, 19 Nov 1981 08:52:00 GMT				
									Keep-Alive: timeout=5, max=97				
									Pragma: no-cache				
									Server: Apache/2.4.52 (Ubuntu)				
									Request Headers (486 B)				

- Χρήση SESSION: Για την διατήρηση των στοιχείων ενός χρήστη μεταξύ των σελίδων, όπως π.χ. από το αρχικό log in στην σελίδα υποβολής προσφορών και εν συνεχεία στο προφίλ του αξιοποιήσαμε την λειτουργία SESSION. Έτσι, αποθηκεύαμε στοιχεία μεταξύ των σελίδων και ήταν πιο εύκολη η ανάκτηση δεδομένων από τη MySQL. Το SESSION εντάχθηκε στα αρχεία PHP όπως φαίνεται παρακάτω:

```
• <?php  
• session_start();
```

- Χωρίς να αποτελεί ρύθμιση του server, αλλά επηρεάζει την πλοήγηση του χρήστη στην ιστοσελίδα, προσθέσαμε τις παρακάτω εντολές:

```
var timeoutDuration = 600000; //10 minutes  
var timeout = setTimeout(function() {  
    window.location.href = "../login_register/log_in.php";  
}, timeoutDuration);  
  
window.addEventListener('mousemove', resetTimer);  
window.addEventListener('click', resetTimer);  
  
function resetTimer() {  
    clearTimeout(timeout);  
    timeout = setTimeout(function() {  
        window.location.href = "../login_register/log_in.php";  
    }, timeoutDuration);  
}
```

Στην ουσία τοποθετείται ένας χρονοδιακόπτης, οποίος μετρά το χρονικό διάστημα κατά το οποίο ο χρήστης έχει παραμείνει αδρανής, δηλαδή δεν έχει κουνήσει το ποντίκι και δεν έχει κάνει κλικ σε κάποιο σημείο της σελίδας. Κάθε φορά που ο χρήστης κάνει κάποια ενέργεια ο timer μηδενίζεται, ωστόσο εάν περάσουν 10 λεπτά στα οποία ο χρήστης είναι αδρανής, το σύστημα αυτόματα τον κάνει log out και τον μεταφέρει στην αρχική σελίδα log in, ώστε να συνδεθεί ξανά με τα στοιχεία του εφόσον το επιθυμεί.

## Τεχνολογίες που χρησιμοποιήθηκαν

- Javascript , MySQL
- PHP , JSON
- Html
- CSS