

ΥΠΗΡΕΣΙΕΣ ΣΤΟ ΥΠΟΛΟΓΙΣΤΙΚΟ
ΝΕΦΟΣ ΚΑΙ ΤΗΝ ΟΜΙΧΛΗ (ΠΛΗ 513)
ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2022-2023

1ο Μέρος Προγραμματιστικής Άσκησης

ΑΝΑΠΤΥΞΗ WEB ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΤΗ ΔΙΑΧΕΙΡΙΣΗ ΣΥΣΤΗΜΑΤΟΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Επιμέλεια Εργασίας: Τζαβάρας Αιμίλιος (email: atzavaras@tuc.gr)

ΠΕΡΙΓΡΑΦΗ ΑΣΚΗΣΗΣ

Στόχος της άσκησης είναι η εξοικείωση με τεχνολογίες Web για την ανάπτυξη μιας Web εφαρμογής (Web application). Θα χρησιμοποιηθούν κάποιες από τις πιο συνηθισμένες και ευρέως διαδεδομένες τεχνολογίες για την ανάπτυξη web εφαρμογής όπως HTML, CSS, JS, AJAX, PHP, MySQL.



1. ΣΕΝΑΡΙΟ CLIENT-SERVER ΕΦΑΡΜΟΓΗΣ

Η εφαρμογή αποτελεί ένα σύστημα για την καταχώριση προϊόντων που προορίζονται για πώληση, καθώς και την αναζήτηση, την επιλογή και την αποθήκευση των προϊόντων από απλούς χρήστες (καταναλωτές) με σκοπό την μετέπειτα παραγγελία αυτών των προϊόντων (όπως σε ένα e-shop).

Οι χρήστες εγγράφονται στην εφαρμογή με τη συμπλήρωση μιας φόρμας στην οποία συμπληρώνουν και το ρόλο που θέλουν να έχουν σε αυτήν (τύπος λογαριασμού).

Οι χρήστες της εφαρμογής μπορούν να έχουν έναν από τους **3 παρακάτω ρόλους**:

Οι **Admins** (Διαχειριστές) είναι οι μόνοι που διαχειρίζονται τον πίνακα χρηστών. Πιο συγκεκριμένα, μπορούν να αποδεχτούν και να διαγράψουν χρήστες και να ενημερώσουν το ρόλο και τα στοιχεία των ήδη υπαρχόντων (προσοχή στο ποια στοιχεία μπορούν να δουν και να ενημερώσουν).

Οι **Product Sellers** (Πωλητές Προϊόντων) είναι άτομα ή επιχειρήσεις/εταιρείες που επιθυμούν να πουλήσουν τα προϊόντα τους. Μπορούν να κάνουν προσθήκη, ενημέρωση και διαγραφή προϊόντων. Πρέπει να φροντίσετε ώστε να έχουν δικαίωμα να διαχειριστούν μόνο τα δικά τους προϊόντα.

Τέλος, οι απλοί **Users** (Χρήστες), δηλαδή οι καταναλωτές, έχουν τη δυνατότητα να αναζητούν προϊόντα με βάση συγκεκριμένα κριτήρια (Είδος/Κατηγορία προϊόντος, Τιμή προϊόντος, Όνομα προϊόντος, Όνομα πωλητή προϊόντος π.χ. κάποια γνωστή εταιρεία, κλπ) και να προσθέτουν στο Καλάθι τους (“Προσθήκη στο Καλάθι” / “Add to Cart”) αυτά που τους ενδιαφέρουν. Θα μπορούν φυσικά ανά πάσα στιγμή να δουν και να διαχειριστούν τη δική τους αποθηκευμένη λίστα προϊόντων στο Καλάθι τους.

Για την υλοποίηση της εφαρμογής απαιτείται η δημιουργία 3 πινάκων σε βάση δεδομένων MySQL:

-Πίνακας: Users

- ID: int (AUTO_INCREMENT)
- NAME: string
- SURNAME: string
- USERNAME: string
- PASSWORD: string
- EMAIL: string
- ROLE: ENUM [“ADMIN”, “PRODUCTSELLER”, “USER”]
- CONFIRMED: BOOLEAN

-Πίνακας: Products

- ID: int (AUTO_INCREMENT)
- NAME: string
- PRODUCTCODE: string
- PRICE: float
- DATEOFWITHDRAWAL: datetime ή string
- SELLERNAME: string
- CATEGORY: string

-Πίνακας: Carts

- ID: int (AUTO_INCREMENT)
- USERID: string
- PRODUCTID: string
- DATEOFINSERTION: datetime ή string

2. ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ

Οι βασικές λειτουργίες που πρέπει να υποστηρίζει το σύστημα που θα φτιάξετε είναι:

2.1 Σύνδεση (Login)

Στην πρώτη σελίδα της εφαρμογής **index.php** θα πρέπει να γίνεται η αυθεντικοποίηση του χρήστη. Ο χρήστης θα πρέπει να δίνει το username και το password μέσα σε μία φόρμα. Πατώντας το κουμπί “login”, θα πρέπει να ελέγχεται η ορθότητα των στοιχείων από τη βάση δεδομένων και, εφόσον είναι σωστά, να του επιτρέπεται η είσοδος στην επόμενη σελίδα, διαφορετικά να εμφανίζεται μήνυμα για λανθασμένο username ή password. Πρέπει να φροντίσετε να μην επιτρέπεται σε καμία σελίδα η πρόσβαση χωρίς authentication, ακόμα και αν κάποιος χρήστης προσπαθεί να αποκτήσει πρόσβαση μέσω του URL. Σε τέτοιες περιπτώσεις, θα πρέπει να γίνεται ανακατεύθυνση (redirection) στη σελίδα index.php. Επίσης, μετά το επιτυχές login, κάθε σελίδα της εφαρμογής θα πρέπει να είναι προσβάσιμη μόνο στους χρήστες που αφορά. Π.χ. μόνο οι πωλητές προϊόντων θα μπορούν να μπουν στη σελίδα προσθήκης νέων προϊόντων στην εφαρμογή ή μόνο οι Admins θα μπορούν να μπουν στη σελίδα ενημέρωσης του ρόλου των χρηστών, κ.ο.κ.

Σημείωση: Παρότι χρησιμοποιείται authentication για τη σύνδεση του χρήστη, δεν υπάρχει ασφάλεια γενικά στη μετάδοση της πληροφορίας. Δηλαδή τα δεδομένα (username, password) δεν προστατεύονται επαρκώς κατά τη μετάδοσή τους με HTTP. Σε επόμενο μέρος της άσκησης (θα ανακοινωθεί αργότερα μέσα στο εξάμηνο) πιθανότατα θα ζητηθεί κάποιος πιο ασφαλής τρόπος/πρωτόκολλο για τη μετάδοση της πληροφορίας στην εφαρμογή.

2.2 Εγγραφή (Sign up)

Στην πρώτη σελίδα της εφαρμογής **index.php** θα πρέπει επίσης να δίνεται η δυνατότητα μέσω ειδικού button για εγγραφή (registration) κάποιου χρήστη στο σύστημα. Με το πάτημα του κουμπιού θα πρέπει να εμφανίζεται άλλη ειδική φόρμα και με την συμπλήρωση στοιχείων (ID, name, surname, username, password, email, role) και το πάτημα ειδικού κουμπιού να καταχωρείται ο χρήστης και να επανεμφανίζεται η αρχική φόρμα για login. Η φόρμα του login θα πρέπει να επανεμφανίζεται και όταν ακυρώνεται κάποια εγγραφή με ειδικό button. Ας γίνει σε πρώτη φάση με redirection σε άλλη σελίδα κάθε φορά (signup.php και index.php). Για να μπορεί ο χρήστης να εισέλθει στην εφαρμογή μετά την εγγραφή, θα πρέπει να έχει γίνει πρώτα αποδεκτός (confirmed) από τον Administrator.

2.3 Είσοδος στο σύστημα

2.3.1. welcome.php

Μετά την επιτυχή είσοδο του χρήστη στο σύστημα, θα γίνεται ανακατεύθυνση στη σελίδα **welcome.php** της εφαρμογής (για όλους τους ρόλους χρηστών). Εκεί θα υπάρχει η δυνατότητα για *αποσύνδεση* (logout) με ανακατεύθυνση στην index.php και το ονοματεπώνυμο του χρήστη και το ρόλο (σε παρένθεση) στο πάνω δεξιό μέρος της σελίδας. Μέσα από κατάλληλο μενού (μπάρα στο αριστερό ή στο πάνω μέρος της σελίδας), ο χρήστης θα μπορεί να προσπελάσει κάθε σελίδα της εφαρμογής. Το μενού θα πρέπει να εμφανίζεται *μόνο* όταν ο χρήστης τοποθετεί τον κέρσορα του ποντικιού πάνω από συγκεκριμένη περιοχή.

Επιπλέον, από το μενού ο χρήστης θα μπορεί να μεταβεί στις σελίδες **products.php** και **cart.php** (για τους Users), **seller.php** (για τους Product Sellers) και **administration.php** (για τους Admins). Σε περίπτωση που ο ρόλος ενός χρήστη δεν επιτρέπει την πρόσβαση σε κάποιες από τις παραπάνω σελίδες, θα πρέπει να μεταβαίνει σε ειδική σελίδα στην οποία θα ενημερώνεται για την απαγόρευση πρόσβασης και θα υπάρχει ειδικό κουμπί για επιστροφή στην αρχική σελίδα.

2.3.2. products.php

Προβολή όλων των προϊόντων και δυνατότητα αναζήτησης με βάση το όνομα του προϊόντος, τον πωλητή (το όνομα ή το username του), το είδος/κατηγορία προϊόντος, την τιμή του προϊόντος ή την ημερομηνία απόσυρσής του (date of withdrawal) από τον πωλητή. Δυνατότητα προσθήκης κάθε προϊόντος στο Καλάθι. Σύνδεσμος για πρόσβαση στη σελίδα του Καλαθιού (Cart) από την οποία θα μπορούν να διαγραφούν (αφαιρεθούν) απ' το Καλάθι προϊόντα, εφόσον ο χρήστης το επιθυμεί.

2.3.3. cart.php

Προβολή όλων των αγαπημένων προϊόντων (όσων είναι στο Καλάθι), της τιμής τους και της ημερομηνίας εισαγωγής τους στο Καλάθι (date of insertion), καθώς και δυνατότητα διαγραφής κάποιου προϊόντος από το Καλάθι. **Προαιρετικά (bonus)**, μπορείτε να υπολογίζετε και να προβάλετε στη σελίδα αυτή το άθροισμα των τιμών όλων των προϊόντων που έχουν εισαχθεί από το χρήστη στο Καλάθι του (συνολικό κόστος). Η τιμή του προϊόντος αποθηκεύεται στο πεδίο PRICE στην MySQL, στον πίνακα Products.

2.3.4. seller.php

Προβολή όλων των καταχωρημένων προϊόντων (με τα στοιχεία τους) και δυνατότητα προσθήκης νέων προϊόντων, διαγραφής τους και ενημέρωσης των στοιχείων των προϊόντων. Προσέξτε ώστε ο κάθε χρήστης να μπορεί να βλέπει και να διαχειρίζεται μόνο τα δικά του προϊόντα.

2.3.5. administration.php

Εμφάνιση όλων των χρηστών σε μορφή πίνακα και δυνατότητα διαγραφής τους και ενημέρωσης των στοιχείων τους στη βάση. Ο κωδικός πρόσβασης του κάθε χρήστη δεν πρέπει προφανώς να είναι ορατός και ούτε να υπάρχει δυνατότητα να ενημερωθεί με κάποιο κουμπί από τον administrator.

3. ΕΠΙΠΛΕΟΝ ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ

3.1 Ασύγχρονη επικοινωνία και ενημέρωση περιεχομένου σελίδας διεπαφής

Για κάθε δημιουργία, επεξεργασία (ενημέρωση), διαγραφή στη βάση δεδομένων, το ανανεωμένο περιεχόμενο θα πρέπει να εμφανίζεται *δυναμικά* στην αντίστοιχη σελίδα μετά από κάθε ενέργεια. Δηλαδή να μη χρειάζεται να ανανεωθεί η εκάστοτε σελίδα για να δει κάποιος χρήστης τις αλλαγές που πραγματοποιήσε. Για παράδειγμα, ο admin διαγράφει κάποιο χρήστη πατώντας ένα κουμπί και ο πίνακας με τους χρήστες που βλέπει ανανεώνεται δυναμικά (αφού έχει ολοκληρωθεί επιτυχώς η διαγραφή στην MySQL και έχει ληφθεί επιβεβαίωση γι' αυτό), χωρίς την ανανέωση (refresh) της σελίδας.

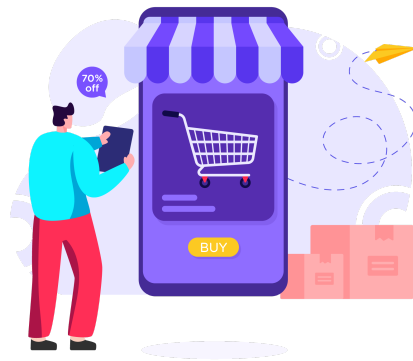
ΣΗΜΕΙΩΣΗ:

- Φροντίστε ώστε η εφαρμογή να είναι όσο γίνεται ευχάριστη και εύκολη στη χρήση (εικόνες, εύχρηστο μενού, γραμματοσειρές, χρώματα, έλεγχος εγκυρότητας δεδομένων εισόδου στις φόρμες, κτλ.).
- Υποθέστε ότι υπάρχει ένας αρχικός Admin στο σύστημα (καταχωρείται κατά την αρχικοποίηση της εφαρμογής).
- Μπορείτε να δημιουργήσετε επιπλέον σελίδες της εφαρμογής όπου αυτό σας εξυπηρετεί αρκεί να ελέγχεται το δικαίωμα πρόσβασης σε αυτές ανάλογα με το ρόλο των χρηστών και να αποτρέπεται η πρόσβαση σε αυτές αν προηγουμένως δεν έχουν αυθεντικοποιηθεί οι χρήστες στο σύστημα (login procedure).
- Για την υλοποίηση της άσκησης απαιτείται η εγκατάσταση και χρήση Apache Server & MySQL Database (*χρήσιμος σύνδεσμος για all in one solution: <https://www.apachefriends.org/index.html>*).
- Είστε ελεύθεροι να χρησιμοποιήσετε οποιοδήποτε Web Framework σας εξυπηρετεί τόσο στη δημιουργία του γραφικού περιβάλλοντος (Frontend) όσο και στην υλοποίηση της λογικής της εφαρμογής (Backend).

ΠΑΡΑΔΟΤΕΑ:

- Scripts κώδικα με επεξηγηματικά σχόλια.

- Ένα αρχείο .sql (backup αρχείο της βάσης MySQL).



// happy coding!