

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

Τμήμα Ηλεκτρολόγων & Ηλεκτρονικών Μηχανικών



UNIVERSITY of WEST ATTICA

FACULTY OF ENGINEERING

Department of Electrical & Electronics Engineering

Πρόγραμμα Μεταπτυχιακών Σπουδών

Διαδικτυωμένα Ηλεκτρονικά Συστήματα

Master of Science in

*Internetworked Electronic Systems*

## ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Εφαρμογή μοντέλου client-server με δυνατότητα αξιοποίησης βάσης  
δεδομένων σε υπολογιστικό σύστημα



Μεταπτυχιακός Φοιτητής : ΛΕΟΝΤΑΡΗΣ ΕΥΑΓΓΕΛΟΣ, ΑΜ: 0018

Επιβλέπων Καθηγητής: ΔΗΜΗΤΡΙΟΣ ΜΕΤΑΦΑΣ, Επικ. Καθηγητής

ΑΙΓΑΛΕΩ, ΣΕΠΤΕΜΒΡΙΟΣ 2018

---



## ΠΕΡΙΛΗΨΗ

Αντικείμενο της παρούσας εργασίας αποτελεί η ανάλυση και η περιγραφή ενός συστήματος τύπου ERP (Enterprise Resource Planning) που δημιουργήθηκε με σκοπό να καλύψει τις ανάγκες μιας μικροζυθοποιίας.

Στο 1<sup>ο</sup> κεφάλαιο γίνεται μια ανάλυση των βασικών εννοιών των συστημάτων ERP και αναλύονται τα πλεονεκτήματα και τα μειονεκτήματα αυτών των συστημάτων.

Στο 2<sup>ο</sup> κεφάλαιο γίνεται η περιγραφή της προτεινόμενης ζυθοποιίας δηλαδή του περιβάλλοντος στο οποίο θα εφαρμοστεί η προτεινόμενη λύση. Επιπλέον, αιτιολογείται η επιλογή της συγκεκριμένης εφαρμογής ως η πιο συμφέρουσα ( από άποψη κυρίως κόστους και χρόνου) για τις ανάγκες της μικροζυθοποιίας.

Στο 3<sup>ο</sup> κεφάλαιο παρουσιάζεται η προτεινόμενη λύση. Γίνεται διεξοδική ανάλυση του κώδικα που χρησιμοποιήθηκε και παρουσιάζονται όλα τα αρχεία που χρησιμοποιήθηκαν ούτως ώστε να συνθέσουν το εν λόγω σύστημα.

Στο 4<sup>ο</sup> και τελευταίο κεφάλαιο αναλύονται τα συμπεράσματα που προέκυψαν από την υλοποίηση του συστήματος καθώς επίσης και προτείνονται κάποιες ιδέες για περαιτέρω εξέλιξη και βελτίωση του συστήματος

**ΛΕΞΕΙΣ – ΚΛΕΙΔΙΑ:** ERP Συστήματα, Βάσεις δεδομένων, Περιβάλλον Ανάπτυξης Διαδικτυακών Υπηρεσιών, Στοιβά Λογισμικού

## ΠΙΝΑΚΑΣ ΣΥΜΒΟΛΩΝ-ΑΚΡΩΝΥΜΙΩΝ- ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ

IS	Information Systems
CBIS	Computer Based Information System
ERP	Enterprise Resource Planning
LAMP	Linux – Apache – MySQL- Php
WAMP	Windows – Apache – MySQL- Php
CRM	Customer Relationship Management
HRM	Human Resource Management
BBT	Bright Beer Tank
PHP	Hypertext Preprocessor
CSS	Cascading Style Sheets
HTML	HyperText Markup Language

# Πίνακας περιεχομένων

## Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1	Εισαγωγή – Θεωρητικό Υπόβαθρο.....	7
1.1	Ορισμός Πληροφοριακού Συστήματος.....	8
1.2	Τι είναι το πληροφοριακό σύστημα ERP;.....	10
1.3	Λόγοι που υπαγορεύουν την ανάγκη εγκατάστασης ERP συστημάτων.....	12
1.4	Πλεονεκτήματα των ERP συστημάτων.....	14
1.5	Μειονεκτήματα των ERP συστημάτων.....	15
ΚΕΦΑΛΑΙΟ 2	Περιγραφή Προτεινόμενης Ζυθοποιίας για την χρήση του λογισμικού.....	17
2.1	Πρώτες ύλες.....	18
2.2	Είδη μύρας.....	20
2.3	Η διαδικασία Παραγωγής.....	20
2.4	Γιατί η προτεινόμενη λύση είναι η πιο συμφέρουσα για την εν λόγω ζυθοποιία;.....	24
ΚΕΦΑΛΑΙΟ 3	Σχεδίαση λογισμικού για την προτεινόμενη εταιρεία.....	26
3.1	Η ιστοσελίδα.....	27
3.2	Το τμήμα private.....	27
3.2.1	Το τμήμα shared.....	27
3.2.1.1	Το αρχείο public_header.....	28
3.2.1.2	Το αρχείο public_footer.....	29
3.2.1.3	Το αρχείο staff_header.....	29
3.2.1.4	Το αρχείο staff_footer.....	30
3.2.2	Τα απαραίτητα custom αρχεία.....	30
3.2.2.1	Το αρχείο auth_functions.....	31
3.2.2.2	Το αρχείο database.....	32
3.2.2.3	Το αρχείο db_credentials.....	33
3.2.2.4	Το αρχείο functions.....	33
3.2.2.5	Το αρχείο initialize.....	35
3.2.2.6	Το αρχείο query_functions.....	36
3.2.2.7	Το αρχείο validation_functions.....	42

3.3. Το τμήμα public .....	44
3.3.1. Το αρχείο index.php.....	44
3.3.2. Το αρχείο contact.php .....	45
3.3.3. Το αρχείο breweryProcess.php .....	46
3.4. Ο φάκελος staff .....	47
3.4.1 Ο φάκελος subject.....	47
3.4.1.1 Το αρχείο create.php .....	47
3.4.1.2 Το αρχείο delete.php .....	48
3.4.1.3 Το αρχείο edit.php .....	50
3.4.1.4 Το αρχείο index.php.....	53
3.4.1.5 Το αρχείο new.php .....	54
3.4.1.6 Το αρχείο show.php.....	56
3.4.2 Ο φάκελος admins.....	57
3.4.2.1 Το αρχείο delete.php .....	57
3.4.2.2 Το αρχείο edit.php .....	58
3.4.2.3 Το αρχείο index.php.....	60
3.4.2.4 Το αρχείο new.php .....	62
3.4.2.5 Το αρχείο show.php.....	63
3.4.3 Το αρχείο index.php.....	64
3.4.4 Το αρχείο log_in.php .....	65
3.4.5 Το αρχείο log_out.php .....	67
3.5 Ο φάκελος stylesheet .....	67
3.5.1 Το αρχείο public.css.....	67
3.5.2 Το αρχείο staff.css.....	71
ΚΕΦΑΛΑΙΟ 4            Συμπεράσματα - Προτάσεις.....	75
ΒΙΒΛΙΟΓΡΑΦΙΑ - ΠΗΓΕΣ .....	78
Αναφορές.....	78

## ΚΕΦΑΛΑΙΟ 1 Εισαγωγή – Θεωρητικό Υπόβαθρο

---

Είναι αναμφίβολο ότι ο 21<sup>ος</sup> αιώνας θα μπορούσε να χαρακτηριστεί ως ο αιώνας της τεχνολογίας, της πληροφορικής και των πληροφοριακών συστημάτων, ο αιώνας της ταχείας μετάδοσης των πληροφοριών, όπου η πληροφορία παίζει σημαντικό ρόλο τόσο στο χώρο των επιχειρήσεων όσο και στη καθημερινή μας ζωή.

Όσον αφορά τον χώρο των επιχειρήσεων οι παγκόσμιες ανταγωνιστικές συνθήκες και οι συνεχείς καινοτομίες, αναγκάζουν πολλές επιχειρήσεις να επανεξετάσουν το τρόπο με τον οποίο συναλλάσσονται και να ανασχεδιάζουν τις ενδοεπιχειρησιακές τους διαδικασίες. Τέτοιοι ανασχεδιασμοί απαιτούν στις περισσότερες περιπτώσεις, την υποστήριξη ολοκληρωμένων πληροφοριακών συστημάτων. Τέτοιου είδους συστήματα είναι τα λεγόμενα Enterprise resource planning (ERP) συστήματα που αποσκοπούν στη μείωση της ενέργειας, τη βελτιστοποίηση της ποιότητας, την ευελιξία, τη μεγιστοποίηση της παραγωγής και της εκμετάλλευσης των αποθεμάτων, όλα τα στοιχεία δηλαδή που απαιτεί μια σύγχρονη ανταγωνιστική επιχείρηση. Τα συστήματα αυτά είναι σύγχρονα προϊόντα λογισμικού που διαχειρίζονται μηχανογραφικά τις κυριότερες εμπορικές και παραγωγικές δραστηριότητες των επιχειρήσεων και οργανισμών, παρέχοντας την πλήρη αξιοποίηση των διαθέσιμων πόρων αλλά και την αποτελεσματικότερη οργάνωση.

Όλες αυτές οι καινοτομίες και τα τεχνολογικά άλματα δεν θα μπορούν να αφήσουν ανεπηρέαστο και τον κλάδο της ζυθοποιίας. Πρόκειται άλλωστε για έναν κλάδο όπου παρατηρείται ιδιαίτερα έντονος και σκληρός ανταγωνισμός τόσο μεταξύ εταιρειών κολοσσών, όπως οι πολυεθνικές εταιρείες της Αθηναϊκής και Ολυμπιακής Ζυθοποιίας, όσο και μεταξύ εγχώριων μικροζυθοποιών. Για να ανταπεξέλθουν λοιπόν στον ανταγωνισμό οι εταιρείες που ασχολούνται με την παραγωγή μπίρας αποφασίζουν να επενδύσουν σε business λογισμικά προκειμένου να ελέγχουν τη ροή των παραγωγικών τους λειτουργιών κι ως εκ τούτου να αυξήσουν τα κέρδη τους.

## 1.1 Ορισμός Πληροφοριακού Συστήματος

Ακούγοντας κανείς την φράση πληροφοριακά συστήματα, (Information Systems, IS) εύλογα σκέπτεται πως αυτά σχετίζονται με τους ηλεκτρονικούς υπολογιστές, την επικοινωνία, την πληροφόρηση (προφορική, γραπτή, ψηφιακή, αναλογική) και γενικότερα με την ραγδαία ανάπτυξη των δυνατοτήτων του διαδικτύου και των περί αυτού τεχνολογιών.

Προσπαθώντας να ορίσουμε τι είναι πληροφοριακό σύστημα θα λέγαμε ότι είναι ένα οργανωμένο σύνολο διαδικασιών για την συλλογή, αποθήκευση, μετάδοση και επεξεργασία δεδομένων για την παροχή χρήσιμων, ολοκληρωμένων και έγκυρων πληροφοριών όπου είναι αυτό απαραίτητο. Επιπρόσθετα, θα λέγαμε ότι πληροφοριακό σύστημα είναι ένα σύνολο διαδικασιών, τόσο ανθρώπινου δυναμικού, όσο και αυτοματοποιημένων υπολογιστικών συστημάτων, που προορίζονται για τη συλλογή, εγγραφή, ανάκτηση, επεξεργασία, αποθήκευση και ανάλυση πληροφοριών. Από το παραπάνω καταλαβαίνει κανείς ότι ο ανθρώπινος παράγοντας λειτουργεί περισσότερο ως αναπόσπαστο κομμάτι του πληροφοριακού συστήματος και λιγότερο ως χρήστης του. «Δεν υπάρχει πληροφοριακό σύστημα χωρίς ανθρώπινη συμμετοχή ενώ είναι δυνατόν να υπάρξει πληροφοριακό σύστημα χωρίς ούτε έναν υπολογιστή» (Palmius, 2007). Επομένως, Πληροφοριακό σύστημα ονομάζεται κάθε περιβάλλον στο οποίο οι πόροι (πχ άνθρωποι και ηλεκτρονικοί υπολογιστές) συντονίζονται για την εκπλήρωση ενός ευρύτερου σκοπού. Επιμέρους και ανεξάρτητα υποσυστήματα λοιπόν, συνεργάζονται ώστε να μετατρέψουν αρχικά δεδομένα υπό τη μορφή στοιχείων εισόδου σε πληροφορίες – στοιχεία εξόδου, που θα είναι χρήσιμες για την επίτευξη των στόχων μια οικονομικής μονάδας.

Σε γενικές γραμμές οι βασικές λειτουργίες που ένα πληροφοριακό σύστημα επιτελεί είναι:

- Ικανοποίηση των απαιτήσεων των δρώντων του οργανισμού (οι οποίοι αποτελούν χρήστες του πληροφοριακού συστήματος) σε πληροφορίες για την εκτέλεση των καθηκόντων τους. Η ικανοποίηση αυτή είναι αναγκαίο να είναι έγκυρη και έγκαιρη.
- Αναζήτηση, διαλογή και ανάκτηση των απαραίτητων δεδομένων από το εσωτερικό του οργανισμού ή από το περιβάλλον του για την εκτέλεση των διαδικασιών που υποστηρίζει.



- Η επεξεργασία των δεδομένων που τελικά ανακτώνται και η παραγωγή πληροφοριών αναγκαίων για τις διαδικασίες λήψης αποφάσεων.
- Αφού παραχθεί η ζητούμενη πληροφορία θα πρέπει να διανεμηθεί στους ενδιαφερόμενους προκειμένου να αξιοποιηθεί.

Όσον αφορά στα στοιχεία που πρέπει να έχει κάθε πληροφοριακό σύστημα προκειμένου να καλύπτει τις ανάγκες όλων των υπολογιστικών μονάδων (Computer Based Information System -CBIS) είναι τα εξής:

- *Λογισμικό*: Το λογισμικό είναι το σύνολο των προγραμμάτων που χρησιμοποιούν οι υπολογιστές με στόχο να διεκπεραιώσουν τις εργασίες που ζητά ο χρήστης και να εξυπηρετήσουν τις κατάλληλες δομές δεδομένων μαζί δε με την σχετική τεκμηρίωσή τους. Το λογισμικό υλοποιεί τη λογική της μεθοδολογίας, της διαδικασίας ή του ελέγχου που απαιτείται, προς εξυπηρέτηση, άμεσα του χρήστη και έμμεσα ολόκληρου του παραγωγικού συστήματος. Παράγει με τον τρόπο αυτό πληροφορίες, δηλαδή επεξεργασμένα δεδομένα σε ωφέλιμη μορφή για αυτόν που τα αποκτά
- *Υλικό*: Είναι όλες οι υλικές μονάδες που χρησιμοποιούνται από το πληροφοριακό σύστημα όπως οι ηλεκτρονικοί υπολογιστές, οι εκτυπωτές, οι scanners, οι δικτυακές συσκευές και οποιεσδήποτε άλλες συσκευές που λειτουργούν στο πληροφοριακό σύστημα τα λεγόμενα περιφερειακά.
- *Ανθρώπινο δυναμικό*: Ο κύριος στόχος ενός πληροφοριακού συστήματος είναι να παρέχει πολύτιμες πληροφορίες στους διαχειριστές του και στους χρήστες του. Οι χρήστες χωρίζονται σε τρεις κατηγορίες:
  - End-Users, που αποτελείται από τους υπαλλήλους, τους πελάτες, τους προμηθευτές και τους άλλους που επικοινωνούν με το σύστημα πληροφοριών.
  - Εσωτερικοί Χρήστες, συμπεριλαμβανομένων των διαχειριστών, οι τεχνικοί, οι αντιπρόσωποι πωλήσεων και τα εταιρικά στελέχη.
  - Εξωτερικοί Χρήστες, που αποτελούνται από τους πελάτες που χρησιμοποιούν τις εταιρίες συστημάτων για την εκτέλεση των συναλλαγών, οι προμηθευτές που χρησιμοποιούν το σύστημα για τον προγραμματισμό των πωλήσεων,

καθώς και το προσωπικό που χρησιμοποιεί το σύστημα εκτός ωρών γραφείου.

- *Δεδομένα*: Τα στοιχεία αυτά αναφέρονται σε πράγματα ή οντότητες και μπορεί να βρίσκονται υπό επεξεργασία μέχρι να πάρουν την τελική μορφή τους. Μπορεί να είναι εικόνα, κείμενο, ήχος, σύμβολα κ.α.
- *Δίκτυο Διακίνησης Δεδομένων και Πληροφοριών*: Είναι η υποδομή που υποστηρίζει την ροή δεδομένων και πληροφοριών.



**Εικόνα 1 Βασικά στοιχεία πληροφοριακού συστήματος**

## 1.2 Τι είναι το πληροφοριακό σύστημα ERP;

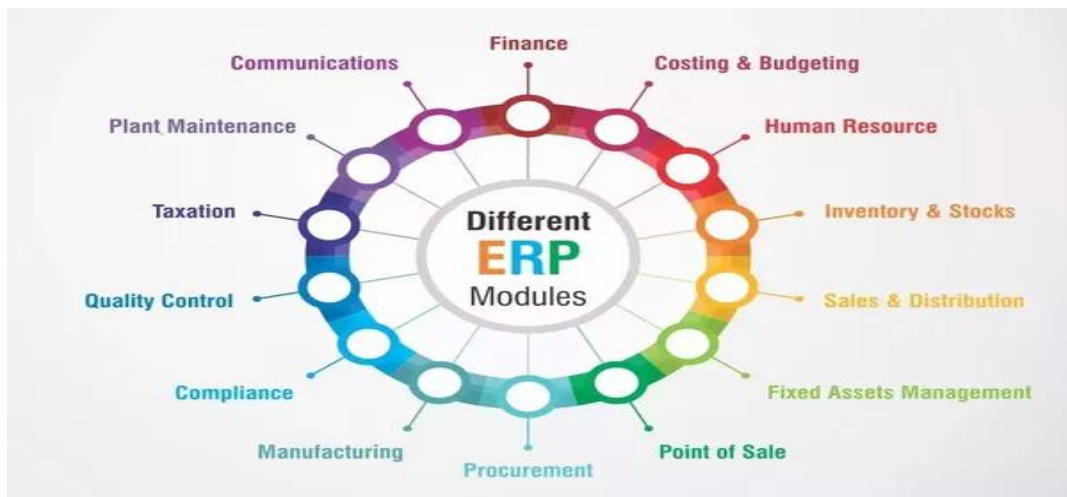
Το ERP σημαίνει Enterprise Resource Planning (Διαχείριση Επιχειρησιακών Πόρων) και πρόκειται για ένα ολοκληρωμένο πληροφοριακό σύστημα που έχει στόχο την υποστήριξη επιχειρησιακών δραστηριοτήτων. Τα συστήματα λογισμικού ERP θα μπορούσαν να οριστούν ως επιχειρησιακά συστήματα που μπορούν να επεξεργαστούν δεδομένα από το εσωτερικό και εξωτερικό περιβάλλον της επιχείρησης και να παρέχουν τις κατάλληλες πληροφορίες στη διοίκηση της με σκοπό να ληφθούν γρήγορα σωστές και έγκυρες αποφάσεις για την αύξηση του *business performance*.

Αξίζει να σημειωθεί ότι ένα σύστημα ERP επεξεργάζεται και αξιοποιεί όλα τα δεδομένα της επιχείρησης, υποστηρίζοντας όλες τις λειτουργίες της, που μπορεί να εκτείνονται από τις

οικονομικές υπηρεσίες και τη διοίκηση ανθρώπινων πόρων, μέχρι την εφοδιαστική αλυσίδα και τις σχέσεις με τους πελάτες. Στόχος επομένως του ERP δεν είναι η εξυπηρέτηση των απαιτήσεων ενός τομέα στην επιχείρηση, όπως λ.χ. του λογιστηρίου, της παραγωγής, των πωλήσεων κ.λπ., αλλά η εξυπηρέτηση των διαδικασιών μέσα στην επιχείρηση, στις οποίες διαδικασίες εμπλέκονται οι διάφοροι τομείς, έτσι ώστε να μπορεί αυτή να διεκπεραιώνει τις κύριες επιχειρηματικές δραστηριότητές της .

Στην ουσία ένα σύστημα ERP, αποτελείται από επιμέρους ενσωματωμένες εφαρμογές (γνωστές ως modules), τις οποίες η επιχείρηση αξιοποιεί για να συγκεντρώνει, να αποθηκεύει, να διαχειρίζεται και σε τελική ανάλυση, να ερμηνεύει δεδομένα για πολλές (ιδανικά όλες) από τις δραστηριότητές της. Από τη στιγμή που τα δεδομένα εισαχθούν σε κάποια μονάδα (module) του ERP, αυτά είναι διαθέσιμα σε οποιαδήποτε μονάδα του ERP τα χρειαστεί. Με τον τρόπο αυτό, επιτυγχάνεται μία λογική ενοποίηση των διαδικασιών μεταξύ των τμημάτων της επιχείρησης. Μερικά από τα modules ενός συστήματος ERP είναι:

- Εμπορική Διαχείριση
- Οικονομική Διαχείριση
- Έσοδα-Εξοδα
- Γενική Λογιστική
- Διαχείριση Παγίων
- Διαχείριση Αποθήκης
- Διαχείριση Ανθρώπινων Πόρων (HRM)
- Μισθοδοσία
- Διαχείριση Διανομής
- Διαχείριση Εφοδιαστικής Αλυσίδας
- Διαχείριση Πελατειακών Σχέσεων (CRM)
- Διαχείριση Λιανικής
- Προϊοντικός Σχεδιασμός
- Διαχείριση Παραγωγής
- Διαχείριση Υπηρεσιών
- Χρηματοοικονομική Λογιστική
- Πωλήσεις
- Marketing
- Τιμολόγηση



**Εικόνα 2 Modules ERP Συστήματος**

### **1.3 Λόγοι που υπαγορεύουν την ανάγκη εγκατάστασης ERP συστημάτων**

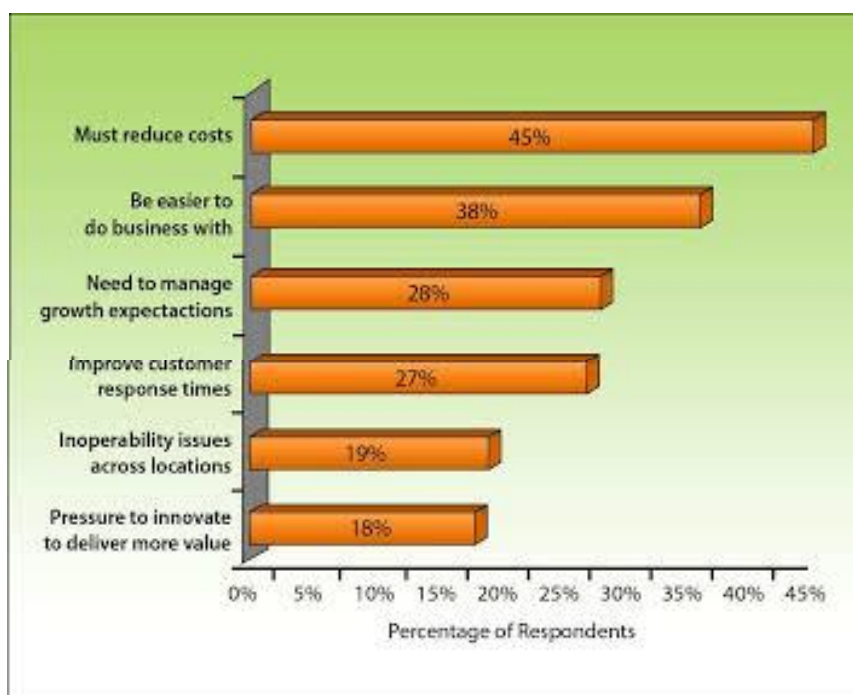
Είναι κοινώς αποδεκτό στις μέρες μας ότι όταν μια επιχείρηση επαναπροσδιορίζει τον τρόπο λειτουργίας της μέσα από τις νέες τεχνολογίες αξιοποιώντας τις δυναμικά, αυτή αποκτά συγκριτικό πλεονέκτημα σε σχέση με αυτές που δεν τις χρησιμοποιούν. Αντίθετα, αυτές οι επιχειρήσεις που ακόμα ανθίστανται στην χρήση νέων τεχνολογιών φαίνεται να μην μπορούν να ανταπεξέλθουν στον σκληρό ανταγωνισμό καθώς δεν μπορούν ούτε να μειώσουν τα κόστη τους, ούτε να αποκτήσουν αποδοτικότερη διαχείριση και λειτουργία.

Σε γενικές γραμμές ένα σύστημα Enterprise Resource Planning (ERP) είναι ένα ολοκληρωμένο πληροφοριακό σύστημα που παρέχει τη δυνατότητα επικοινωνίας μεταξύ όλων των τμημάτων μιας επιχείρησης. Ένα σύστημα ERP, επιτελεί ουσιαστικά προσομοίωση της πραγματικότητας των καθημερινών πρακτικών. Οι λόγοι που οι επιχειρήσεις προμηθεύονται συστήματα ERP είναι οι εξής (Πολλάλης, Γιαννακόπουλος, & Παπουτσή, 2004) :

- ❖ Συγκρότηση όλων των μηχανογραφικών διαδικασιών κάτω από ένα ενιαίο μηχανογραφικό σύστημα.
- ❖ Αύξηση της παραγωγικότητας και της αποδοτικότητας.
- ❖ Βελτίωση της ποιότητας.
- ❖ Ολοκλήρωση επιχειρησιακών διαδικασιών.
- ❖ Μείωση κόστους σε όλη την εφοδιαστική αλυσίδα (αποθεμάτων, προμηθειών, logistics, πληροφοριακών πόρων).

- ❖ Ακεραιότητα και ακρίβεια πληροφοριών
- ❖ Ταχύτητα
- ❖ Βασική υποδομή για διευρυμένη επιχείρηση (extended enterprise) και ebusiness.

Σύμφωνα με έρευνα του 2010 που διεξήχθη από την Ομάδα Aberdeen για Μικρές και Μεσαίες Επιχειρήσεις (MME) ως προς τους κύριους λόγους που οδήγησαν τις εταιρίες να εφαρμόσουν μια λύση ERP, σχεδόν το 50% ανέφερε την ανάγκη να μειωθεί το κόστος για να βελτιωθούν τα περιθώρια της κερδοφορίας. Η βελτίωση της επαφής με τους πελάτες ακολουθούσε σαν την δεύτερη πιο συχνά αναφερόμενη αιτία. Η κατανομή των απαντήσεων σε αυτή την έρευνα φαίνεται παρακάτω στο γράφημα.



**Εικόνα 3 Παράγοντες απόφασης για την εγκατάσταση ενός ERP συστήματος**

Συνοπτικά θα λέγαμε ότι οι βασικοί λόγοι που καθιστούν αναγκαία την εγκατάσταση ενός ERP συστήματος είναι i) είτε η επίλυση υπαρχόντων επιχειρησιακών προβλημάτων, ii) είτε η προσπάθεια για βελτίωση των διαδικασιών στην επιχείρηση. Σε κάθε περίπτωση όμως το ERP είναι απλώς το μέσο, η δυνατότητα για την επιχείρηση να βελτιώσει τις λειτουργίες της. Από εκεί και πέρα, χρειάζεται δημιουργική ενσωμάτωση του συστήματος ERP μέσα στην επιχείρηση, ώστε να είναι παραγωγική.

## 1.4 Πλεονεκτήματα των ERP συστημάτων

Η εφαρμογή των συστημάτων ERP στις επιχειρήσεις δημιουργεί πολλές ευκαιρίες για τη βελτίωση της ανταγωνιστικότητας. Τα συστήματα ERP προσφέρουν τα παρακάτω πλεονεκτήματα (Beheshti, 2006, Gupta, 2000) :

- Καλύτερος συντονισμός σε όλη τη ροή των εμπορικών και παραγωγικών λειτουργιών της επιχείρησης, από τη μείωση των αποθεμάτων και τη φειδώ στις παραγγελίες των πρώτων υλών, ως την καλύτερη εκμετάλλευση του παραγωγικού εξοπλισμού, με αποτέλεσμα τη μείωση των χρόνων διεκπεραίωσης & παράδοσης των προϊόντων στους πελάτες και την ταυτόχρονη αποφυγή νεκρών χρόνων ή άσκοπων υπερωριών.
- Κατάργηση πολλαπλής εισαγωγής των ίδιων δεδομένων σε ποικίλα πληροφοριακά συστήματα με ενοποίηση όλων των λειτουργιών εμπορίας (πωλήσεων και αγορών), προγραμματισμού και παρακολούθησης παραγωγής, οικονομικών όπως επίσης και των υπόλοιπων διοικητικών λειτουργιών σε ένα μοναδικό σύστημα.
- Μείωση των λειτουργικών εξόδων της μηχανογραφικής υποστήριξης της επιχείρησης, λόγω της χρήσης ενός και μόνο συστήματος.
- Ακριβέστερος προγραμματισμός & καλύτερη αξιοποίηση της χρήσης των πόρων της επιχείρησης.
- Τυποποίηση και κεντρικός έλεγχος των βασικών εργασιών της επιχείρησης.
- Αυτοματοποίηση διαδικασιών & αύξηση παραγωγικότητας στο γραφείο, λόγω της επιτάχυνσης διεκπεραιωτικών εργασιών ρουτίνας.
- Εξοικονόμηση χρόνου των στελεχών διοίκησης εξαιτίας της έγκυρης και ολοκληρωμένης πληροφόρησης, με αποτέλεσμα τη λήψη καλύτερων και ταχύτερων αποφάσεων και τη βελτίωση της αποδοτικότητας.
- Καλύτερη εξυπηρέτηση πελατών, με περισσότερη ακρίβεια και συνέπεια στους χρόνους παράδοσης, με συνέπεια τη βελτίωση της συνολικής εικόνας της επιχείρησης.
- Βελτίωση της προσαρμοστικότητας της επιχείρησης και της απόκρισής της στις μεταβολές της αγοράς και τις αλλαγές προτεραιοτήτων από τους πελάτες, μέσα από τη δυνατότητα ταχύτερης αναπροσαρμογής των προγραμμάτων και των παραγγελιών.

## 1.5 Μειονεκτήματα των ERP συστημάτων

Παρόλο που τα συστήματα ERP σχεδιάζονται με στόχο τη βελτίωση της λειτουργίας της επιχείρησης ως σύνολο σε ένα διαρκώς εξελισσόμενο εξωτερικό περιβάλλον, παρουσιάζουν μειονεκτήματα που είναι σημαντικό να αναφερθούν. Η εγκατάσταση λοιπόν ενός ERP μπορεί να ενέχει και κινδύνους λόγω του ότι :

- το πληροφοριακό σύστημα και η οργανωτική δομή της επιχείρησης δεν είναι συμβατές (οργανωτικός κίνδυνος),
- δεν έχουν εκπαιδευτεί οι χρήστες στη διαχείριση του συστήματος (κίνδυνος εφαρμογής). Ίσως επίσης υπάρξει απροθυμία των εργαζομένων προς εκπαίδευση και χρήση του συστήματος,
- Οι συνεχείς απαιτήσεις για εκπαίδευση του προσωπικού, δεσμεύουν ανθρώπινους πόρους που είναι χρήσιμοι για την καθημερινή λειτουργία της επιχείρησης,
- οι λειτουργίες που προσφέρονται από το λογισμικό είναι διαφορετικές από τις αναμενόμενες λειτουργίες και συνεπώς δεν ανταποκρίνονται στις προσδοκίες. (λειτουργικός κίνδυνος),
- η σύμβαση μεταξύ του εκδότη του πληροφοριακού συστήματος και της επιχείρησης είναι ατελής (τεχνικός κίνδυνος)
- η επιχείρηση έχει υπερεκτιμήσει τα οφέλη από το σύστημα (οικονομικός κίνδυνος),
- η εφαρμογή του συστήματος μπορεί να είναι χρονοβόρα και να υπάρξει υπέρβαση του κόστους.

Σε κάθε περίπτωση τα αναμενόμενα οφέλη από ένα σύστημα ERP είναι τεράστια αλλά και οι παράγοντες που μπορεί να προκαλέσουν την αποτυχία πολλοί, με κυριότερους την έλλειψη εμπειρίας για τέτοιου εύρους έργα και την αντίδραση στην αλλαγή που εκ των πραγμάτων επιφέρεται με την υλοποίησή τους. Με τη δημιουργία της σωστής ομάδας και τη συμμετοχή όλων τα προβλήματα θα ξεπεραστούν και άπαντες τότε θα βιώσουν ένα περιβάλλον έγκαιρης και έγκυρης πληροφόρησης με όλα τα πλεονεκτήματα που έχει αυτό για την εταιρεία αλλά και τους εργαζόμενους.

## Συστήματα, πριν και μετά τα ERPs

	<i>Προ- ERP</i>	<i>Μετα- ERP</i>
Πληροφοριακά συστήματα	Μεμονωμένα συστήματα	Ολοκληρωμένα συστήματα
Συντονισμός	Έλλειψη συντονισμού μεταξύ επιχειρησιακών λειτουργιών (πχ. παραγωγή και πωλήσεις)	Υποστήριξη συντονισμού μεταξύ επιχειρησιακών λειτουργιών
Βάσεις Δεδομένων	Μη ολοκληρωμένα δεδομένα, διαφορετική σημασία δεδομένων (πχ. πελάτης), ασυνέπεια στον ορισμό δεδομένων	Ολοκληρωμένα δεδομένα, ίδια σημασία δεδομένων σε όλες τις λειτουργίες
Συντήρηση	Τμηματική συντήρηση, ασυνέπειες, το κόστος συντήρησης μεμονωμένων legacy systems είναι μεγάλο	Ομοιόμορφη συντήρηση, οι μεταβολές επηρεάζουν πολλαπλά συστήματα
Interfaces	Δυσκολία στη διαχείριση των Interfaces μεταξύ συστημάτων	Κοινά Interfaces μεταξύ συστημάτων
Πληροφορία	Περιττή, ασυνεπής πληροφορία	Συνεπής πληροφορία σε πραγματικό χρόνο (πχ. για πελάτες, προμηθευτές)
Αρχιτεκτονική συστήματος	Μπορεί να μην είναι η πιο προηγμένη	Βασίζεται στο μοντέλο n-tier
Διαδικασίες	Μη συμβατές διαδικασίες	Συνεπείς επιχειρησιακές διαδικασίες που βασίζονται σε πληροφοριακό μοντέλο
Εφαρμογές	Ανόμοιες εφαρμογές (πχ. πολλά διαφορετικά συστήματα αγορών)	Μοναδικές εφαρμογές (πχ. ένα σύστημα αγορών)

13

**Εικόνα 4 Η κατάσταση στην επιχείρηση προ και μετά την εγκατάσταση ενός ERP**



## ΚΕΦΑΛΑΙΟ 2 Περιγραφή Προτεινόμενης Ζυθοποιίας για την χρήση του λογισμικού

---

Το πληροφοριακό σύστημα που σχεδιάστηκε απευθύνεται σε μικρές/μεσαίες ζυθοποιίες οι οποίες έχουν ως στόχο την αποδοτικότερη διαχείριση των πρώτων υλών(πόρων) τους, την επίβλεψη και την καταγραφή τους και ως εκ τούτου την αποτελεσματικότερη διαχείριση των παραγγελιών και των πελατών τους. Προτού όμως παρουσιαστεί το λογισμικό θεωρείται αναγκαίο να αναφερθούν γενικά πληροφοριακά στοιχεία και βασικά χαρακτηριστικά του κλάδου της ζυθοποιίας στην οποία προτείνεται η χρήση του εν λόγω λογισμικού.

Σήμερα οι ελληνικές ζυθοποιίες αριθμούνται γύρω στις 20. Το μεγαλύτερο μερίδιο αγοράς κατέχει η Αθηναϊκή ζυθοποιία, ενώ η Ολυμπιακή Ζυθοποιία και η Ζυθοποιία Μακεδονίας – Θράκης είναι κάποιες από τις μεγαλύτερες του κλάδου. Αξίζει επίσης να σημειωθεί ότι το τελευταίο χρονικό διάστημα ιδιαίτερα κερδισμένες στην αγοράς μπίρας εμφανίζονται οι ελληνικές ζυθοποιίες, με την Έζα και την ΖΜΘ που παρουσιάζουν σημαντική αύξηση του τζίρου τους.

Η ζήτηση του συγκεκριμένου προϊόντος επηρεάζεται από παράγοντες όπως το εισόδημα των καταναλωτών, τη τιμή του προϊόντος, τη τουριστική κίνηση κατά τη διάρκεια του καλοκαιριού και η διαφήμιση, η οποία επηρεάζει σε ένα βαθμό τις προτιμήσεις του κοινού. Με την εποχικότητα ως κύριο χαρακτηριστικό του κλάδου, τα brands μπίρας επενδύουν στην ευρηματική διαφήμιση και στις προωθητικές ενέργειες. Σημαντική ανάπτυξη καταγράφουν και οι μικροζυθοποιίες, ενώ η προτίμηση των καταναλωτών αρχίζει να στρέφεται στην εγχώρια παραγωγή.

Στόχος όλων, είναι πλέον και σύμφωνα με τις υπάρχουσες οικονομικές συνθήκες που επικρατούν στη χώρα μας, η προσπάθεια ολοένα και μεγαλύτερου μεριδίου της εγχώριας αγοράς ,για μεγαλύτερες εξαγωγές προϊόντος στην Ευρώπη αλλά και παγκοσμίως. Γι' αυτό το σκοπό άλλωστε εφαρμόζουν ιδιαίτερα δαπανηρά επενδυτικά προγράμματα- όπως για παράδειγμα το σύστημα Enterprise Resource Planning- με απώτερο στόχο την αύξηση

κερδών μύρας μέσα από την αύξηση παραγωγικότητας την αποτελεσματικότερη διαχείριση και λειτουργία.

## 2.1 Πρώτες ύλες

Για την παραγωγή της μύρας είναι απαραίτητα τέσσερα βασικά συστατικά : το νερό, η βύνη, ο λυκίσκος και η μαγιά. Από αυτές τις τέσσερις παραδοσιακές φυσικές πρώτες ύλες πρέπει αποκλειστικά να παρασκευάζεται η μύρα, σύμφωνα με τον γερμανικό νόμο, γνωστό και ως «νόμος περί αγνότητας της μύρας» (Reinheitsgebot). Παρακάτω παρουσιάζονται αναλυτικά οι πρώτες ύλες για μια τέτοιου είδους εταιρεία:

- **Νερό:** Είναι, ίσως το σημαντικότερο συστατικό καθώς αποτελεί το 95% της μύρας. Απαιτείται η χρήση καθαρού νερού τόσο σε βιολογικό επίπεδο όσο και σε φυσικοχημικό ενώ σημαντικό ρόλο έχει και η περιεκτικότητα σε άλατα ασβεστίου και μαγνησίου τα οποία επηρεάζουν το PH της μύρας, τη δράση των ενζύμων και του λυκίσκου. Η σημαντικότητα του νερού γίνεται περισσότερο κατανοητή αν αντιληφθεί κανείς ότι πριν αρκετά χρόνια, το προφίλ του νερού προσδιόριζε σε μεγάλο βαθμό και τον τύπο της μύρας. Η εξέλιξη της τεχνολογίας έχει οδηγήσει στην αυτοματοποιημένη επεξεργασία του νερού ώστε να επιτευχθεί το επιθυμητό προφίλ ανάλογα με το είδος της παραγόμενης μύρας.
- **Βύνη κριθαριού:** Το κριθάρι ευδοκμεί σε χώρες της Βόρειας Αμερικής και της Ευρώπης. Η ποιότητα του κριθαριού εξαρτάται σε μεγάλο βαθμό από:
  - τις ποικιλίες
  - τις εδαφολογικές συνθήκες
  - τις κλιματολογικές συνθήκες
  - την τεχνογνωσία του καλλιεργητή

Για την παραγωγή της μύρας χρησιμοποιείται κυρίως δίστιχο κριθάρι, το οποίο είναι φτωχό σε πρωτεΐνες και πλούσιο σε άμυλο. Πριν την αξιοποίηση του στην παραγωγική διαδικασία, το κριθάρι υποβάλλεται στη διαδικασία της βυνοποίησης ώστε να παραχθεί η βύνη. Στόχος, της βυνοποίησης είναι η δημιουργία των

απαιτούμενων ενζύμων, εντός του κριθαριού, ώστε να γίνει διάσπαση και να μετατραπεί το περιεχόμενο του κριθαριού από μη διαλυτό σε διαλυτό. Η πρωτεϊνική σύνθεση του κριθαριού, από το οποίο προέρχεται, αποτελεί έναν από τους πιο καθοριστικούς παράγοντες της τελικής ποιότητας της βύνης, η οποία με τη σειρά της επηρεάζει καθοριστικά το σώμα και τον αφρό. Η βύνη αποτελείται κυρίως από άμυλο (60-65%), υδατάνθρακες (7-10%), πρωτεΐνες (9-12%), καθώς και από μεταλλικά άλατα, βιταμίνες, πολυφαινόλες, κυτταρίνη και άλλους πολυσακχαρίτες.

- **Λυκίσκος:** Υπάρχουν πολλά και διαφορετικά είδη λυκίσκου. Ωστόσο, μόνο ο «*Humulus Lupulus*», και συγκεκριμένα τα θηλυκά άνθη του αξιοποιείται κατά τη ζυθοποίηση. Ο λυκίσκος περιέχει έλαια και ρητίνες που καθορίζουν τη γεύση, το άρωμα και την πικράδα της μύρας ενώ, έχει αντιβακτηριδιακή δράση, περιέχει ενώσεις που δρουν ως συντηρητικά, όπως είναι οι τανίνες και τα οργανικά οξέα. Ο λυκίσκος ευδοκιμεί σε χώρες των ζωνών που φαίνονται στο χάρτη, (μεταξύ των οποίων η Κεντρική Ευρώπη, η Βόρειος Αμερική και η Δυτική Ασία) ενώ οι περισσότερες ζυθοποιίες χρησιμοποιούν το Λυκίσκο σε μορφή εκχυλίσματος ή πελλετών.
- **Μαγιά:** Η μαγιά είναι ένας μικροσκοπικός μονοκύτταρος οργανισμός. Από τις εκατοντάδες ποικιλίες και είδη μαγιάς, μόνο 2 βασικές κατηγορίες αξιοποιούνται στη διαδικασία της ζυθοποίησης.
  - Βυθοζύμες (bottom fermenting) χρησιμοποιούνται για την παραγωγή μύρας τύπου Lager και καθιζάνουν στον πυθμένα του δοχείου.
  - Αφροζύμες (top fermenting) χρησιμοποιούνται για την παραγωγή μύρας τύπου Ale και κατά τη διάρκεια της ζύμωσης συγκεντρώνονται στην επιφάνεια του γλεύκους.

Βασικό ρόλο της μαγιάς αποτελεί η μετατροπή των ζυμώσιμων σακχάρων του γλεύκους σε αλκοόλη και διοξείδιο του άνθρακα. Παράλληλα κατά την αλκοολική ζύμωση παράγονται και πλήθος άλλων στοιχείων που καθορίζουν την ποιότητα του τελικού προϊόντος. Η μαγιά έχει και έναν επιπλέον ρόλο στη ζυθοποίηση, καθώς

προσδίδει ιδιαίτερο άρωμα και γεύση στην μύρα και προσδιορίζοντας τον τελικό χαρακτήρα της μύρας.

## **2.2 Είδη μύρας**

Στην παγκόσμια αγορά διατίθενται χιλιάδες μάρκες μύρας. Οι μάρκες αυτές ανταποκρίνονται σε διαφορετικές καταναλωτικές ανάγκες και κατατάσσονται σε διαφορετικού τύπου μύρας οι οποίοι με το πέρασμα των χρόνων έχουν εξελιχθεί και συνδεθεί με διαφορετικές γεωγραφικές περιοχές. Ωστόσο, μια πρώτη κατηγοριοποίηση είναι εφικτή βάση του είδους μαγιάς και της διαδικασίας ζύμωσης:

- 1) Μύρες που έχουν ζυμωθεί με αφοζύμες και
- 2) Μύρες που έχουν ζυμωθεί με βυθοζύμες.

## **2.3 Η διαδικασία Παραγωγής**

Η παραγωγική διαδικασία της μύρας αποτελείται από τα 3 στάδια:  
Βυνοποίηση | Ζυθοποίηση | Εμφιάλωση και Αποθήκευση



Εικόνα 5 Τα στάδια παραγωγής της μύρας

## 1. ΒΥΝΟΠΟΙΗΣΗ

Αποτελεί τη διαδικασία μετατροπής του κριθαριού σε βύνη. Η βυνοποίηση αποτελείται από 3 στάδια:

### - Διαβροχή (Steeping)

Στάδιο κατά το οποίο λαμβάνει χώρα η διοχέτευση νερού/αέρα μέχρι το νερό να αποτελεί το 45% του βάρους του κριθαριού.

### - Βλάστηση (Germination)

Διοχέτευση αέρα και παραμονή για 5 ημέρες σε θερμοκρασία 12-16°C. Σ' αυτό το στάδιο παράγονται τα χρήσιμα κατά τη ζυθοποίηση ένζυμα όπως Αμυλάσες και Πρωτεάσες.

### - Φρύξη (Kilning)

Διοχέτευση θερμού αέρα 80°C με στόχο την απενεργοποίηση των ενζύμων και την παράλληλα αύξηση του χρώματος και αρώματος.

## 2. ΖΥΘΟΠΟΙΗΣΗ

## **Ανάμιξη**

Διαδικασία κατά την οποία ποσότητα βύνης αλέθεται και αναμιγνύεται με νερό σε ειδική δεξαμενή που φέρει θερμομανδύα για τη ρύθμιση της θερμοκρασίας. Το σκαχαροδιάλυμα που προκύπτει από την ανάμιξη ονομάζεται γλεύκος.

## **Διήθηση**

Σε συνέχεια της ανάμιξης το γλεύκος διηθείται με στόχο την επίτευξη διαύγασης. Η διήθηση πραγματοποιείται σε ειδικές δεξαμενές με διάτρητο πυθμένα και σύστημα ανάδευσης αποτελούμενο από περιστρεφόμενα μαχαίρια.

## **Βρασμός**

Μετά τη διήθηση του γλεύκους ακολουθεί ο βρασμός του, ο οποίος αποτελεί μία σημαντικότερη διεργασία κατά την οποία λαμβάνουν χώρα τα ακόλουθα:

- Αποστείρωση του ζυθογλεύκους
- Απενεργοποίηση της δράσης των ενζύμων
- Ανάπτυξη του χρώματος, καθώς και του αρωματικού και γευστικού προφίλ (flavor)
- Συμπύκνωση του ζυθογλεύκους λόγω εξάτμισης νερού
- Ισομερίωση και εκχύλιση ουσιών που περιέχονται στο λυκίσκο (ισο-α- οξέα)
- Σχηματισμός και καταβύθιση συμπλεγμάτων πρωτεϊνών – πολυφαινολών
- Απομάκρυνση πτητικών ανεπιθύμητων ουσιών (π.χ.DMS) που έχει ως αποτέλεσμα τη σταθεροποίηση της γεύσης και του αρώματος

Τα υπολείμματα του λυκίσκου και οι κροκιδωμένες κατά τον βρασμό πρωτεΐνες απομακρύνονται από το βρασμένο γλεύκος με περιδίνηση και κατακάθιση, ενώ το διαυγές γλεύκος ψύχεται, περνώντας από εναλλάκτη θερμότητας και οδηγείται προς τη δεξαμενή ζυμώσεως.

## **Ζύμωση**

- Κατά την πορεία προς τη δεξαμενή ζυμώσεως, προστίθεται στο γλεύκος μαγιά και διοχετεύεται αέρας.

- Στις δεξαμενές ζυμώσεως γίνεται η αλκοολική ζύμωση υπό ελεγχόμενες συνθήκες θερμοκρασίας και πίεσης. Η διαδικασία, διαρκεί από 5 έως 9 ημέρες, επιτυγχάνοντας τον μεταβολισμό των σακχάρων του γλεύκους από τη μαγιά με την ταυτόχρονη παραγωγή αιθανόλης και διοξειδίου του άνθρακα.

### **Σίτευση - Ωρίμαση**

- Το στάδιο της ωρίμασης ή σίτευσης της μύρας προσδίδει στην μύρα το τελικό γευστικό προφίλ της.

- Κατά την ωρίμαση η μύρα παγώνει στους 0 με -1°C.

### **Φιλτράρισμα**

Πριν τη συσκευασία, απαιτείται το φιλτράρισμα της μύρας ώστε να απομακρυνθούν υπολείμματα μαγιάς καθώς και διάφορες ουσίες που υπονομεύουν την ποιότητα της μύρας με την πάροδο του χρόνου (πρωτεΐνες, τανίνες). Κατά το στάδιο του φιλτραρίσματος ρυθμίζεται και η ακριβής συγκέντρωση του ανθρακικού και επίσης γίνεται η τελική ρύθμιση της πυκνότητας, και η διαυγής μύρα (bright beer) αποθηκεύεται σε ειδικές δεξαμενές BBT (bright beer tanks). Τέλος, η μύρα οδηγείται προς το τμήμα εμφιαλώσεως και συσκευασίας σε τρεις τύπους: φιάλη, αλουμινένιο κουτί και βαρέλι.

## **3. ΕΜΦΙΑΛΩΣΗ ΚΑΙ ΑΠΟΘΗΚΕΥΣΗ**

Η τεχνολογία σήμερα έχει προχωρήσει πολύ και υπάρχουν σε πολλές ζυθοποιίες αυτόματες μηχανές εμφιάλωσης που ξεπλένουν και στη συνέχεια, γεμίζουν τα μπουκάλια, προσθέτουν τις ετικέτες και τα πώματα. Σε μικροζυθοποιίες αυτό γίνεται πολλές φορές χειρωνακτικά και αν όχι όλο σίγουρα κάποια στάδια. Πριν την εμφιάλωση συνήθως γίνεται παστερίωση (θέρμανση για την θανάτωση των διάφορων μικροοργανισμών) ο βαθμός της οποίας προσδιορίζεται ανάλογα με τον τρόπο αποθήκευσης της μύρας και το χρονικό επιτρεπόμενο διάστημα κατανάλωσης της. Μην ξεχνάμε ότι οι μύρες εμφιαλώνονται είτε σε βαρέλια, είτε σε μπουκάλια διαφορετικών λίτρων, είτε τέλος σε κουτάκια. Στα κουτάκια που κατά κύριο λόγο η παραμονή είναι μεγαλύτερη γίνεται και πιο έντονη παστερίωση. Όσο αφορά στην τελική αποθήκευση η μύρα, σαν ποτό με χαμηλό αλκοόλ είναι ευαίσθητη, θέλει λοιπόν προσοχή και δροσερό περιβάλλον μακριά από τον ήλιο και την υψηλή θερμοκρασία.

Αξίζει να σημειωθεί ότι η διαδικασία παραγωγής ενός είδους μπίρας αν και είναι σχετικά αργή διαδικασία χρήζει συνεχούς παρακολούθησης διότι το τελικό προϊόν είναι αρκετά ευαίσθητο και οι διορθωτικές ενέργειες σε περίπτωση λάθους είναι περιορισμένες. Συνεπώς ένα σύστημα που θα καταγράφει τις πρώτες ύλες θα τις ελέγχει και θα τις έχει διαθέσιμες ανά πάσα ώρα και στιγμή είναι αναγκαίο για την διάθεση πιο ποιοτικού προϊόντος που παρέχει στην αγορά.

## **2.4 Γιατί η προτεινόμενη λύση είναι η πιο συμφέρουσα για την εν λόγω ζυθοποιία;**

Τα συστήματα ERP αποδεδειγμένα αυξάνουν την παραγωγικότητα μιας επιχείρησης μειώνοντας τις δαπάνες της σε εργασιακό χρόνο και οικονομικό κόστος. Αυτή η αποτελεσματικότητά τους όμως συνεπάγεται κάποιες συνέπειες τις οποίες η εν λόγω λύση αποφεύγει.

Αρχικά, σημαντικό ρόλο παίζει η ετοιμότητα της επιχείρησης και συγκεκριμένα το επίπεδο της εσωτερικής οργάνωσης, τα στελέχη και το προσωπικό που θα πρέπει να είναι έτοιμα στην αρχή να προσαρμοστούν και στη συνέχεια να απορροφήσουν ένα τέτοιο πρόγραμμα. Συνήθως για μία επαγγελματική προσέγγιση τύπου ERP δαπανώνται αρκετές εργατοώρες για εκπαιδεύσεις και για ρυθμίσεις του συστήματος. Στατιστικά έχει υπολογισθεί ένας μέσος όρος ενός έτους, γεγονός που αναλογεί σε αρκετές χιλιάδες ευρώ. Στην προτεινόμενη λύση αυτό το πρόβλημα κοστίζει σίγουρα πολύ - πολύ λιγότερο διότι ουσιαστικά δεν απαιτείται σχεδόν καμία εκπαίδευση αφού το όλο σύστημα είναι στημένο με μορφή ιστοσελίδας.

Επίσης, δεν χρειάζεται ιδιαίτερος χρόνος για να στηθεί το σύστημα επειδή είναι ήδη φτιαγμένο ούτως ώστε να καλύψει τις βασικές ανάγκες σε πληροφόρηση και καταγραφή πρώτων υλών, έτοιμων προϊόντων και ποιοτικών χαρακτηριστικών. Επί της ουσίας η απλότητα του όλου εγχειρήματος μας δίνει το πλεονέκτημα μιας γρήγορης εγκατάστασης με μηδαμινό κόστος για την εταιρεία και τους εργαζόμενους της, χωρίς όμως να στερούμαστε σε λειτουργικότητα και αποδοτικότητα συστήματος.

Ακόμη αξίζει να σημειωθεί ότι για μια επαγγελματική λύση τύπου ERP απαιτείται συγκεκριμένος εξοπλισμός (H/Y) με συγκεκριμένες δυνατότητες. Επίσης, πολλές φορές απαιτείται συγκεκριμένο λειτουργικό σύστημα ούτως ώστε να εγκατασταθεί το λογισμικό.



Ακόμα κι όμως αν δεν απαιτείται συγκεκριμένο λειτουργικό, απαιτούνται σίγουρα άδειες λειτουργίας του προγράμματος. Η προτεινόμενη λύση επειδή υλοποιείται σε μορφή ιστοσελίδας πρακτικά μπορεί να γίνει προσπελάσιμη από οποιαδήποτε ηλεκτρονική συσκευή που μπορεί να συνδεθεί στο διαδίκτυο. Αυτό σημαίνει ότι δεν είναι αναγκαία η αγορά πρόσθετου εξοπλισμού. Για τον ίδιο λόγο δεν απαιτείται αγορά αδειών λειτουργίας διότι ουσιαστικά δεν επιβαρύνουμε τον εξοπλισμό μας με κάποιο λογισμικό. Ενδεικτικά αναφέρουμε ότι μια άδεια λειτουργίας για 3-6 Η/Υ φτάνει κάποιες χιλιάδες ευρώ ανάλογα με την εταιρεία που παρέχει το ERP σύστημα.

Συμπερασματικά, λοιπόν αντιλαμβάνεται κανείς ότι η πολύχρονη διαδικασία στησίματος ενός ERP προγράμματος αποφεύγεται αφού ο χρόνος στησίματος της προτεινόμενης λύση είναι μηδαμινός. Επίσης, τα έξοδα εγκατάστασης, συντήρησης και εκπαίδευσης σχεδόν παραλείπονται λόγω της φύσης της εφαρμογής.

## ΚΕΦΑΛΑΙΟ 3 Σχεδίαση λογισμικού για την προτεινόμενη εταιρεία

---

Στο παρόν κεφάλαιο παρουσιάζεται το λογισμικό που σχεδιάστηκε εξ αρχής ώστε να προσαρμοστεί στα δεδομένα και στις ανάγκες της εν λόγω εταιρείας. Πιο συγκεκριμένα, η λύση που προτείνεται είναι μια ιστοσελίδα στην οποία έχουν πρόσβαση οι εργαζόμενοι της εταιρείας από τα τμήματα της παραγωγής, του ποιοτικού ελέγχου και της αποθήκης καθώς επίσης και απλοί επισκέπτες της ιστοσελίδας. Ειδικότερα, η ιστοσελίδα επιτελεί δύο λειτουργίες:

- α) να συνδέει τα τμήματα παραγωγής, ποιοτικού ελέγχου και αποθήκης στηριζόμενη σε μία κοινή βάση δεδομένων
- β) να λειτουργεί σαν απλός ιστότοπος παρέχοντας γενικές πληροφορίες (όπως πληροφορίες σχετικά με την τοποθεσία και την παραγωγική διαδικασία της εταιρείας).

Η διάρθρωση των ενότητων στο εν λόγω κεφάλαιο γίνεται ως εξής:

- Στην ενότητα 3.1 γίνεται μια προεπισκόπηση της ιστοσελίδας. Η ιστοσελίδα χωρίζεται σε δύο μέρη τα οποία αναλύονται εκτενέστερα στα κεφάλαια 3.2 και 3.3.
- Στην ενότητα 3.2 γίνεται η παρουσίαση και η επεξήγηση του τμήματος της ιστοσελίδας του οποίου τα αρχεία δεν είναι εμφανή από κανέναν χρήστη αλλά έχουν όλες τις απαραίτητες συναρτήσεις ώστε να δημιουργηθεί η ιστοσελίδα.
- Στην ενότητα 3.3 γίνεται η παρουσίαση του τμήματος της ιστοσελίδας του οποίου οι χρήστες μπορούν να περιηγηθούν και να επιλέξουν τις ενέργειες που τους διατίθενται. Το τμήμα αυτό της ιστοσελίδας αφορά τόσο τους εξουσιοδοτημένους χρήστες, όσο και τους απλούς επισκέπτες της.

- Στην ενότητα 3.4 γίνεται η ανάλυση του τμήματος της ιστοσελίδας στο οποίο μπορούν οι χρήστες με πρόσβαση και μόνο να κάνουν αλλαγές,
- ενώ στην ενότητα 3.5 γίνεται η επεξήγηση των αρχείων της ιστοσελίδας τα οποία είναι απαραίτητα για την μορφοποίησή της.

### 3.1 Η ιστοσελίδα

Η ιστοσελίδα χωρίζεται σε δύο μέρη. Πιο συγκεκριμένα, διακρίνεται στο τμήμα *public* όπου παρέχονται στον χρήστη δύο δυνατότητες. Αρχικά διατίθενται γενικές πληροφορίες για την εταιρεία, όπως συμβαίνει σε μία τυπική σελίδα. Αυτό συμβαίνει μέσω των επιλογών Brewery Process και Contact, ενώ μέσω της επιλογής Tanks ο χρήστης μπορεί να διαχειρίζεται και να επεξεργάζεται τα δεδομένα στα οποία έχει πρόσβαση ανάλογα με το τμήμα της ζυθοποιίας στο οποίο ανήκει. Ο χρήστης μπορεί να ανήκει είτε στο τμήμα της παραγωγής, είτε στο τμήμα του ποιοτικού ελέγχου, είτε στο τμήμα της αποθήκης. Για παράδειγμα ο χρήστης «Ποιοτικός Έλεγχος» μπορεί να δει και να επεξεργαστεί τα δεδομένα που αφορούν το τμήμα του, όπως το LOT της μαγιάς, λυκίσκου κ.α.

Το δεύτερο τμήμα της ιστοσελίδας είναι το τμήμα *private* όπου δεν έχει κανένας χρήστης πρόσβαση παρα μόνο ο δημιουργός/συντηρητής της ιστοσελίδας. Το τμήμα αυτό περιέχει όλα τα απαραίτητα αρχεία με συναρτήσεις έτοιμες και μη τα οποία συνθέτουν την ιστοσελίδα.

### 3.2 Το τμήμα private

Το τμήμα private όπως αναφέρθηκε νωρίτερα είναι η βάση της δημιουργίας της ιστοσελίδας. Διακρίνεται στον υποφάκελο shared και στα custom αρχεία που δημιουργήθηκαν για να εξυπηρετούν λειτουργίες της ιστοσελίδας χωρίς να εκθέτουν κώδικα στον απλό χρήστη.

#### 3.2.1. Το τμήμα shared

Στον υποφάκελο shared περιλαμβάνονται τα αρχεία public\_header, public\_footer ,staff\_header και staff\_footer.

### 3.2.1.1 Το αρχείο public\_header

Το αρχείο public\_header όπως υποδηλώνει το όνομα του είναι η αρχή της κεντρικής σελίδας (homepage). Ο σκοπός της ύπαρξής του όπως και των υπολοίπων αρχείων είναι να μπορεί εύκολα να μεταβάλλεται από το αρχείο public.css για το οποίο θα μιλήσουμε αργότερα αλλά και να χρησιμοποιείται όποτε κρίνεται αναγκαίο. Με αυτόν τον τρόπο μπορούμε εύκολα να κάνουμε μικρο-αλλαγές στην ιστοσελίδα χωρίς να υπάρχει κίνδυνος να πέσει ολόκληρο το site. Το αρχείο public\_header είναι το παρακάτω:

```
<!doctype html>

<html lang="en">
  <head>
    <title>ies2018 <?php if(isset($page_title)) { echo '- ' . h($page_title); }
?></title>
    <meta charset="utf-8">

    <link rel="stylesheet" media="all" href="<?php echo
url_for('/stylesheets/public.css'); ?>" />
  </head>

  <body>

    <header>
      <h1>
        Brewing Project 2018
      </h1>
    </header>
```

Αξίζει να σημειωθεί στο αρχείο public.css ορίζεται η εμφάνιση όλων των tags της html σελίδας όπως του title του header κ.α. Σε αυτό το αρχείο χρησιμοποιείται η συνάρτηση url\_for η οποία δημιουργήθηκε για να μας απαλλάσσει να γράφουμε κάθε φορά το ακριβές path του αρχείου που χρησιμοποιούμε.

### 3.2.1.2 Το αρχείο public\_footer

Το αρχείο public\_footer χρησιμοποιείτε όπως και το public\_header. Για αυτό το λόγο το tag <html> και το tag <body> ανοίγουν με το public\_header και κλείνουν με το public\_footer. Στο τέλος χρησιμοποιούμε και μια custom συνάρτηση την db\_disconnect για να απελευθερώσουμε την βάση δεδομένων μας.

```
<footer>
    <p>Copyright <?php echo date('Y'); ?>, IES_2018 </p>
</footer>

</body>
</html>

<?php
    db_disconnect($db);
?>
```

### 3.2.1.3 Το αρχείο staff\_header

Το αρχείο staff\_header χρησιμοποιείτε για να διαμορφώσει τις html σελίδες που έχουν πρόσβαση οι χρήστες με δικαιώματα admin οι οποίοι μπορούν να διαμορφώσουν τις βάσεις δεδομένων της εφαρμογής. Υπάρχουν τρία κουμπιά που παραπέμπουν στις αντίστοιχες html σελίδες, το Home, Menu και Logout. Επίσης υπάρχει και η ένδειξη User για να παρατηρούμε αυτή την στιγμή ποιος χρήστης είναι συνδεδεμένος.

```
<?php

if(!isset($page_title)) { $page_title = 'Staff Area';}
?>

<!doctype html>

<html lang="en">
    <head>
        <title>IES_thesis-<?php echo $page_title; ?></title>
        <meta charset="utf-8">
```

```

    <link rel="stylesheet" media="all" href="<?php echo
url_for('/stylesheets/staff.css');?>" />
</head>

<body>

    <header>
    <h1>Brewery Project</h1>
    </header>

    <navigation>
    <ul>
        <li>User: <?php echo isset($_SESSION['username']) ? $_SESSION['username']
: ''; ?></li>
        <li><a href="<?php echo url_for('/staff/index.php'); ?>">Menu</a></li>
        <li><a href="<?php echo url_for('/staff/log_out.php'); ?>">Logout</a></li>
        <li><a href="<?php echo url_for('/index.php'); ?>">Home</a></li>
    </ul>
    </navigation>

```

#### 3.2.1.4 Το αρχείο staff\_footer

Το αρχείο staff\_footer κλείνει την html σελίδα παρουσιάζοντας την ημερομηνία τροποποίησης και καλεί την συνάρτηση db\_disconnect.

```

<footer>
    &copy; <?php echo date('Y'); ?> ies
</footer>

</body>
</html>

<?php
    db_disconnect($db);
?>

```

#### 3.2.2. Τα απαραίτητα custom αρχεία

Τα απαραίτητα custom αρχεία είναι αρχεία php και χρησιμοποιούνται από όλες τις html σελίδες που συνθέτουν το σύστημα τύπου ERP. Δημιουργήθηκαν με σκοπό να καλούνται από τις html σελίδες και να διαθέτουν σε αυτές όλες τις custom συναρτήσεις. Η συγγραφή των συγκεκριμένων αρχείων βοήθησε αρκετά ούτως ώστε να υπάρχει

καλύτερη οργάνωση του πηγαίου κώδικα αλλά και πολύ μεγαλύτερη ευελιξία στην συντήρηση ολόκληρου του συστήματος.

### 3.2.2.1 Το αρχείο `auth_functions`

Το αρχείο `auth_functions` χρησιμοποιείται κυρίως από τις html σελίδες για `log in` και `logout`. Υπάρχουν ακόμα δυο πρόσθετες συναρτήσεις οι οποίες καλούνται από όλες τις html σελίδες που έχουν πρόσβαση οι χρήστες-μέλη. Αυτές είναι οι `require_login` και `is_logged_in` όπου η πρώτη καλεί την δεύτερη και ελέγχει το αποτέλεσμα. Αν υπάρχει ο λογαριασμός συνεχίζει την ανάγνωση της html σελίδας στην οποία βρίσκεται, αλλιώς κάνει `redirect` στην `log_in.php`

`<?php`

```
// Performs all actions necessary to log in an admin
function log_in_admin($admin) {
// Regenerating the ID protects the admin from session fixation.
    session_regenerate_id();
    $_SESSION['admin_id'] = $admin['id'];
    $_SESSION['last_login'] = time();
    $_SESSION['username'] = $admin['username'];

    return true;
}

// Performs all actions necessary to log out an admin
function log_out_admin() {
    unset($_SESSION['admin_id']);
    unset($_SESSION['last_login']);
    unset($_SESSION['username']);
    // session_destroy(); // optional: destroys the whole session
    return true;
}

// is_logged_in() contains all the logic for determining if a
// request should be considered a "logged in" request or not.
// It is the core of require_login() but it can also be called
// on its own in other contexts (e.g. display one link if an admin
// is logged in and display another link if they are not)
function is_logged_in() {
    // Having a admin_id in the session serves a dual-purpose:
    // - Its presence indicates the admin is logged in.
    // - Its value tells which admin for looking up their record.
    return isset($_SESSION['admin_id']);
}

// Call require_login() at the top of any page which needs to
// require a valid login before granting access to the page.
function require_login() {
    if(!is_logged_in()) {
```

```

        redirect_to(url_for('/staff/log_in.php'));
    } else {
        // Do nothing, let the rest of the page proceed
    }
}
?>

```

Να σημειώσουμε ότι χρησιμοποιούνται τα Sessions αντί για τα Cookies διότι με τα Sessions υπάρχει μεγαλύτερος χώρος αποθήκευσης στην μνήμη, γίνεται συχνή αναζήτηση στην βάση δεδομένων και είναι κατάλληλα για user authentication και redirect. Αντιθέτως η υστέρηση στην προβολή της εκάστοτε σελίδας είναι ελάχιστη και δεν δημιουργεί πρόβλημα καθώς δεν γίνεται αντιληπτή από τον χρήστη.

### 3.2.2.2 Το αρχείο database

Το αρχείο database.php είναι υπεύθυνο για την σύνδεση, την αποσύνδεση, την επιβεβαίωση σύνδεσης με την βάση δεδομένων και την επιβεβαίωση αποτελέσματος από την βάση δεδομένων. Χρησιμοποιήθηκαν οι συναρτήσεις που υπάρχουν στο πακέτο της php όπως οι mysqli\_connect, mysqli\_connect\_error κλπ. Επίσης η συνάρτηση db\_escape δημιουργήθηκε χάριν συντομίας γιατί χρησιμοποιείται αρκετά συχνά. Να αναφέρουμε ότι με την συγκεκριμένη συνάρτηση αποφεύγεται η γνωστή επίθεση sql injection καθώς αυτό που θα μπει σαν input στο query θα είναι απαλλαγμένο από special characters οπότε κάποιο query που θα είναι τύπου : something or 1=1; Θα εκτελεστεί ως something or 11; και εκ των πραγμάτων θα χτυπήσει σφάλμα.

```

<?php

require_once('db_credentials.php');

function db_connect(){
    $connection = mysqli_connect(DB_SERVER, DB_USER, DB_PASS, DB_NAME);
    confirm_db_connect();
    return $connection;
}

function db_disconnect($connection){
    if(isset($connection)){
        mysqli_close($connection);
    }
}

```



```

}
}

function db_escape($connection,$string){
    return mysqli_real_escape_string($connection,$string);
}

function confirm_db_connect(){
    if(mysqli_connect_errno()) {
        $msg = "Database connection failed: ";
        $msg .= mysqli_connect_error();
        $msg .= " (" . mysqli_connect_errno() . ")";
        exit($msg);
    }
}

function confirm_result_set($result_set){
    if (!$result_set) {
        exit("Database query failed.");
    }
}
?>

```

### 3.2.2.3 Το αρχείο db\_credentials

Το αρχείο db\_credentials.php περιέχει τα στοιχεία που χρειάζεται για να συνδεθούμε με την βάση δεδομένων.

```

<?php

define("DB_SERVER","localhost");
define("DB_USER","webuser");
define("DB_PASS","ies2018");
define("DB_NAME","brewing");

?>

```

### 3.2.2.4 Το αρχείο functions

Το αρχείο functions.php περιέχει συναρτήσεις που μας διευκολύνουν σχετικά με το path του αρχείου, με το redirect των σελίδων, την εμφάνιση μηνυμάτων λαθών σχετικά με την σελίδα αλλά και τον έλεγχο του τρόπου αποστολής δεδομένων (POST REQUEST ή GET REQUEST). Επίσης κατασκευάστηκαν κάποιες συναρτήσεις που βοηθούν στο να γίνονται queries στην βάση δεδομένων χωρίς κίνδυνο όπως οι u,h και raw\_u.

```

<?php

```

```

function url_for($script_path) {
    // add the leading '/' if not present
    if($script_path[0] != '/') {
        $script_path = "/" . $script_path;
    }
    return WWW_ROOT . $script_path;
}

function u($string=""){
    return urlencode($string);
}

function raw_u($string=""){
    return rawurlencode($string);
}

function h($string=""){
    return htmlspecialchars($string);
}

function error_404(){
    header($_SERVER["SERVER_PROTOCOL"] . " 404 Not Found");
    exit();
}

function error_500(){
    header($_SERVER["SERVER_PROTOCOL"] . " 500 Internal Server Error");
    exit();
}

function redirect_to($location){
    header("Location: " . $location);
    exit;
}

function is_post_request(){
    return $_SERVER['REQUEST_METHOD'] == 'POST';
}

function is_get_request(){
    return $_SERVER['REQUEST_METHOD'] == 'GET';
}

function display_errors($errors=array()) {
    $output = '';
    if(!empty($errors)) {
        $output .= "<div class=\"errors\">";
        $output .= "Please fix the following errors:";
        $output .= "<ul>";
        foreach($errors as $error) {
            $output .= "<li>" . h($error) . "</li>";
        }
    }
}

```

```

        $output .= "</ul>";
        $output .= "</div>";
    }
    return $output;
}
?>

```

### 3.2.2.5 Το αρχείο initialize

Το αρχείο initialize.php χρησιμοποιείται από πολλές άλλες html σελίδες καθώς είναι το αρχείο που γίνεται η αρχικοποίηση όλων. Αρχικά ενεργοποιούμε τον output\_buffer που μας βοηθά στο να μην υπάρχουν προβλήματα με τα redirect. Ύστερα ενεργοποιούμε τα sessions στα οποία αποθηκεύουμε δεδομένα παρόμοια με το να αποθηκεύαμε δεδομένα με την χρήση cookies. Οι επόμενες γραμμές είναι για να οριστούν τα paths και να καλέσουμε όλα τα custom αρχεία. Στην συνέχεια γίνεται σύνδεση στην βάση δεδομένων και ορίζεται ο πίνακας λαθών. Από τα παραπάνω συμπεραίνουμε ότι το αρχείο initialize.php είναι ίσως το κυριότερο αρχείο διότι αρχικοποιεί τα πάντα.

```

<?php
ob_start();//output_buffering is on

session_start();//turn on sessions
// Assign file paths to PHP constants
// __FILE__ returns the current path to this file
// dirname() returns the path to the parent directory
define("PRIVATE_PATH", dirname(__FILE__));
define("PROJECT_PATH", dirname(PRIVATE_PATH));
define("PUBLIC_PATH", PROJECT_PATH . '/public');
define("SHARED_PATH", PRIVATE_PATH . '/shared');

// Assign the root URL to a PHP constant
// * Do not need to include the domain
// * Use same document root as webserver
// * Can set a hardcoded value:
// define("WWW_ROOT", '~/ies_thesis/brewery_project/public');
// define("WWW_ROOT", '');
// * Can dynamically find everything in URL up to "/public"
$public_end = strpos($_SERVER['SCRIPT_NAME'], '/public') + 7;
$doc_root = substr($_SERVER['SCRIPT_NAME'], 0, $public_end);
define("WWW_ROOT", $doc_root);

require_once('functions.php');

```

```

require_once('database.php');
require_once('query_functions.php');
require_once('auth_functions.php');
require_once('validation_functions.php');

$db = db_connect();
$errors = [];
?>

```

### 3.2.2.6 Το αρχείο query\_functions

Το αρχείο query\_functions.php χρησιμοποιείται για να κάνουμε όλα τα queries στην βάση δεδομένων. Τα queries έχουν φτιαχτεί με την μορφή συναρτήσεων και καλούνται μέσα σε άλλες html σελίδες επανειλημμένα. Όλες οι συναρτήσεις επιστρέφουν ένα αντικείμενο τύπου πίνακα associative. Ακόμη αυτό που αξίζει να σημειωθεί είναι ότι όταν χρησιμοποιείται ο κωδικός χρήστη καλούμε την συνάρτηση password\_hash η οποία κρυπτογραφεί τον κωδικό με κρυπτογράφηση bcrypt. Αυτό γίνεται με σκοπό να μην υπάρχει σε απλή μορφή ο κωδικός χρήστη στην βάση δεδομένων.

```

<?php
function find_all_tanks(){
    global $db;

    $sql = "SELECT * FROM tanks ";
    $sql .= "ORDER BY tanks_id ASC";
    $result = mysqli_query($db, $sql);
    confirm_result_set($result);
    return $result;
}

function find_tanks_by_id($id){
    global $db;
    $sql = "SELECT * FROM tanks ";
    $sql .= "WHERE tanks_id=" . db_escape($db,$id);
    //echo $sql; //in order to check sql query
    $result = mysqli_query($db,$sql);
    confirm_result_set($result);
    $tanks= mysqli_fetch_assoc($result);
    mysqli_free_result($result);
    return $tanks; //returns an assoc. array
}

function insert_subject($tank_number, $malt, $water, $yeast, $hops, $recipe,
    $timestamp){
    global $db;

    $errors = validate_subject($tanks);

```

```

if(!empty(errors)){
    return $errors;
}

$sql = "INSERT INTO tanks ";
$sql .= "(tank_number, malt, water, yeast, hops, recipe, timestamp) ";
$sql .= "VALUES (";
$sql .= "" . db_escape($db, $tank_number) . ",";
$sql .= "" . db_escape($db, $malt) . ",";
$sql .= "" . db_escape($db, $water) . ",";
$sql .= "" . db_escape($db, $yeast) . ",";
$sql .= "" . db_escape($db, $hops) . ",";
$sql .= "" . db_escape($db, $recipe) . ",";
$sql .= "" . db_escape($db, $timestamp) . """;
$sql .= ")";
$result = mysqli_query($db,$sql);
// For INSERT statements $result is True/False
if($result){
    return true;
}
else{
    //INSERT failed
    echo mysqli_error($db);
    db_disconnect($db);
    exit;
}
}

function update_subject($tanks){

    global $db;

    $errors = validate_subject($tanks);
    if(!empty($errors)){
        return $errors;
    }
    $sql = "UPDATE tanks SET ";
    $sql .= "tanks_id='" .db_escape($db, $tanks['tanks_id']) . "', ";
    $sql .= "tank_number='" .db_escape($db, $tanks['tank_number']) . "', ";
    $sql .= "malt='" .db_escape($db, $tanks['malt']) . "', ";
    $sql .= "water='" .db_escape($db, $tanks['water']) . "', ";
    $sql .= "yeast='" .db_escape($db, $tanks['yeast']) . "', ";
    $sql .= "hops='" .db_escape($db, $tanks['hops']) . "', ";
    $sql .= "recipe='" .db_escape($db, $tanks['recipe']) . "', ";
    $sql .= "timestamp='" .db_escape($db, $tanks['timestamp']) . "' ";
    $sql .= "WHERE tanks_id='" .db_escape($db, $tanks['id']) . "' ";
    $sql .= "LIMIT 1";

    $result = mysqli_query($db, $sql);
    //For UPDATE statements, $result is true/false
    if($result){
        return true;
    }
}

```

```

    }
    else{
        echo mysqli_error($db);
        db_disconnect($db);
        exit;
    }
}

function delete_subject($id){
    global $db;
    $sql = "DELETE FROM tanks ";
    $sql .= "WHERE tanks_id='" . db_escape($db, $id) ."' ";
    $sql .= "LIMIT 1";

    $result = mysqli_query($db, $sql);

    //For DELETE statements, $result is true/false
    if($result){
        return true;
    }else {
        //DELETE Failed
        echo mysqli_error($db);
        db_disconnect($db);
        exit;
    }
}
}

```

```

function validate_subject($tanks) {

    $errors = [];
    /*
    // tank_number
    if(is_blank($tanks['tank_number'])) {
        $errors[] = "Tank number should be greater than 0 and less than 999.";
    }
    if(!has_length($tanks['tank_number'], ['min' => 2, 'max' => 999])) {
        $errors[] = "Tank number should be greater than 0 and less than 999";
    }
    */
    // position
    // Make sure we are working with an integer
    $tank_number_int = (int) $tanks['tank_number'];
    if($tank_number_int <= 0) {
        $errors[] = "Tank number must be greater than zero.";
    }
    if($tank_number_int > 999) {
        $errors[] = "Tank number must be less than 999.";
    }
    /*
    // visible

```

```

    // Make sure we are working with a string
    $visible_str = (string) $subject['visible'];
    if(!has_inclusion_of($visible_str, ["0","1"])) {
        $errors[] = "Visible must be true or false.";
    }
    */
    return $errors;
}

// Admins

// Find all admins, ordered last_name, first_name
function find_all_admins() {
    global $db;

    $sql = "SELECT * FROM admins ";
    $sql .= "ORDER BY id ASC, username ASC";
    $result = mysqli_query($db, $sql);
    confirm_result_set($result);
    return $result;
}

function find_admin_by_id($id) {
    global $db;

    $sql = "SELECT * FROM admins ";
    $sql .= "WHERE id='" . db_escape($db, $id) . "' ";
    $sql .= "LIMIT 1";
    $result = mysqli_query($db, $sql);
    confirm_result_set($result);
    $admin = mysqli_fetch_assoc($result); // find first
    mysqli_free_result($result);
    return $admin; // returns an assoc. array
}

function find_admin_by_username($username) {
    global $db;

    $sql = "SELECT * FROM admins ";
    $sql .= "WHERE username='" . db_escape($db, $username) . "' ";
    $sql .= "LIMIT 1";
    $result = mysqli_query($db, $sql);
    confirm_result_set($result);
    $admin = mysqli_fetch_assoc($result); // find first
    mysqli_free_result($result);
    return $admin; // returns an assoc. array
}

function validate_admin($admin) {

    if(is_blank($admin['username'])) {
        $errors[] = "Username cannot be blank.";
    }
}

```

```

} elseif (!has_length($admin['username'], array('min' => 8, 'max' => 255))) {
    $errors[] = "Username must be between 8 and 255 characters.";
} elseif (!has_unique_username($admin['username'], $admin['id'] ?$admin['id']
: 0)) {
    $errors[] = "Username not allowed. Try another.";
}

if(is_blank($admin['password'])) {
    $errors[] = "Password cannot be blank.";
} elseif (!has_length($admin['password'], array('min' => 12))) {
    $errors[] = "Password must contain 12 or more characters";
} elseif (!preg_match('/[A-Z]/', $admin['password'])) {
    $errors[] = "Password must contain at least 1 uppercase letter";
} elseif (!preg_match('/[a-z]/', $admin['password'])) {
    $errors[] = "Password must contain at least 1 lowercase letter";
} elseif (!preg_match('/[0-9]/', $admin['password'])) {
    $errors[] = "Password must contain at least 1 number";
} elseif (!preg_match('/[^A-Za-z0-9\s]/', $admin['password'])) {
    $errors[] = "Password must contain at least 1 symbol";
}

if(is_blank($admin['confirm_password'])) {
    $errors[] = "Confirm password cannot be blank.";
} elseif ($admin['password'] !== $admin['confirm_password']) {
    $errors[] = "Password and confirm password must match.";
}

return $errors;
}

function insert_admin($admin) {
    global $db;

    $errors = validate_admin($admin);
    if (!empty($errors)) {
        return $errors;
    }

    $hashed_password = password_hash($admin['password'], PASSWORD_BCRYPT);

    $sql = "INSERT INTO admins ";
    $sql .= "( username, hashed_password) ";
    $sql .= "VALUES (";
    $sql .= "'" . db_escape($db, $admin['username']) . "',";
    $sql .= "'" . db_escape($db, $hashed_password) . "'";
    $sql .= ")";
    $result = mysqli_query($db, $sql);

    // For INSERT statements, $result is true/false
    if($result) {
        return true;
    } else {
        // INSERT failed
        echo mysqli_error($db);
    }
}

```



```

        db_disconnect($db);
        exit;
    }
}

function update_admin($admin) {
    global $db;

    $errors = validate_admin($admin);
    if (!empty($errors)) {
        return $errors;
    }

    $hashed_password = password_hash($admin['password'], PASSWORD_BCRYPT);

    $sql = "UPDATE admins SET ";
    $sql .= "hashed_password='" . db_escape($db, $hashed_password) . "',";
    $sql .= "username='" . db_escape($db, $admin['username']) . "' ";
    $sql .= "WHERE id='" . db_escape($db, $admin['id']) . "' ";
    $sql .= "LIMIT 1";
    $result = mysqli_query($db, $sql);

    // For UPDATE statements, $result is true/false
    if($result) {
        return true;
    } else {
        // UPDATE failed
        echo mysqli_error($db);
        db_disconnect($db);
        exit;
    }
}

function delete_admin($admin) {
    global $db;

    $sql = "DELETE FROM admins ";
    $sql .= "WHERE id='" . db_escape($db, $admin['id']) . "' ";
    $sql .= "LIMIT 1";
    $result = mysqli_query($db, $sql);

    // For DELETE statements, $result is true/false
    if($result) {
        return true;
    } else {
        // DELETE failed
        echo mysqli_error($db);
        db_disconnect($db);
        exit;
    }
}

?>

```

### 3.2.2.7 Το αρχείο validation\_functions

Το αρχείο validation\_functions.php περιέχει συναρτήσεις που ελέγχουν τα δεδομένα που βάζει ο χρήστης στις φόρμες ούτως ώστε να μην βάζει δεδομένα εκτός προδιαγραφών. Για παράδειγμα στην συμπλήρωση username τα δεδομένα δεν πρέπει να ξεπερνούν τους 20 χαρακτήρες.

```
<?php

// is_blank('abcd')
// * validate data presence
// * uses trim() so empty spaces don't count
// * uses === to avoid false positives
// * better than empty() which considers "0" to be empty
function is_blank($value) {
    return !isset($value) || trim($value) === '';
}

// has_presence('abcd')
// * validate data presence
// * reverse of is_blank()
// * I prefer validation names with "has_"
function has_presence($value) {
    return !is_blank($value);
}

// has_length_greater_than('abcd', 3)
// * validate string length
// * spaces count towards length
// * use trim() if spaces should not count
function has_length_greater_than($value, $min) {
    $length = strlen($value);
    return $length > $min;
}

// has_length_less_than('abcd', 5)
// * validate string length
// * spaces count towards length
// * use trim() if spaces should not count
function has_length_less_than($value, $max) {
    $length = strlen($value);
    return $length < $max;
}

// has_length_exactly('abcd', 4)
// * validate string length
// * spaces count towards length
// * use trim() if spaces should not count
function has_length_exactly($value, $exact) {
```

```

    $length = strlen($value);
    return $length == $exact;
}

// has_length('abcd', ['min' => 3, 'max' => 5])
// * validate string length
// * combines functions_greater_than, _less_than, _exactly
// * spaces count towards length
// * use trim() if spaces should not count
function has_length($value, $options) {
    if(isset($options['min']) && !has_length_greater_than($value, $options['min']
- 1)) {
        return false;
    } elseif(isset($options['max']) && !has_length_less_than($value,
$options['max'] + 1)) {
        return false;
    } elseif(isset($options['exact']) && !has_length_exactly($value,
$options['exact'])) {
        return false;
    } else {
        return true;
    }
}

// has_inclusion_of( 5, [1,3,5,7,9] )
// * validate inclusion in a set
function has_inclusion_of($value, $set) {
    return in_array($value, $set);
}

// has_exclusion_of( 5, [1,3,5,7,9] )
// * validate exclusion from a set
function has_exclusion_of($value, $set) {
    return !in_array($value, $set);
}

// has_string('nobody@nowhere.com', '.com')
// * validate inclusion of character(s)
// * strpos returns string start position or false
// * uses !== to prevent position 0 from being considered false
// * strpos is faster than preg_match()
function has_string($value, $required_string) {
    return strpos($value, $required_string) !== false;
}

// has_valid_email_format('nobody@nowhere.com')
// * validate correct format for email addresses
// * format: [chars]@[chars].[2+ letters]
// * preg_match is helpful, uses a regular expression
//   returns 1 for a match, 0 for no match
//   http://php.net/manual/en/function.preg-match.php
function has_valid_email_format($value) {
    $email_regex = '/\A[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,}\Z/i';
    return preg_match($email_regex, $value) === 1;
}

```

```

}

function has_unique_username($username, $current_id="0") {
    global $db;

    $sql = "SELECT * FROM admins ";
    $sql .= "WHERE username='" . db_escape($db, $username) . "' ";
    $sql .= "AND id != '" . db_escape($db, $current_id) . "'";

    $result = mysqli_query($db, $sql);
    $admin_count = mysqli_num_rows($result);
    mysqli_free_result($result);

    return $admin_count === 0;
}
?>

```

### 3.3. Το τμήμα public

Το τμήμα public διακρίνεται στους φακέλους: images, staff, stylesheets και στις html σελίδες: index.php, contact.php και breweryProcess.php. Αυτό το τμήμα ουσιαστικά είναι το ενεργό κομμάτι του εν λόγω συστήματος. Από εδώ ο χρήστης είτε ως απλός επισκέπτης μπορεί να περιηγηθεί στην ιστοσελίδα και να συλλέξει γενικές πληροφορίες για την ζυθοποιία είτε ως συνδεδεμένος χρήστης να επεξεργαστεί τα δεδομένα που έχουν καταχωρηθεί στο σύστημα.

#### 3.3.1. Το αρχείο index.php

Το αρχείο index.php αποτελεί την home page του συστήματος τύπου erp. Αρχικά φορτώνονται στην σελίδα το αρχείο initialize.php και το header.php. Στην συνέχεια φορτώνεται από τον φάκελο images η εικόνα που εμφανίζεται και με το tag navigation φορτώνονται οι σύνδεσμοι που παραπέμπουν τον χρήστη στις υπόλοιπες html σελίδες. Όλα τα κομμάτια της συγκεκριμένης σελίδας διαμορφώνονται σύμφωνα με το αρχείο public.css

```

<?php require_once('../private/initialize.php'); ?>

<?php include(SHARED_PATH . '/public_header.php'); ?>

<div id="main">

```

```

<div id="page">
    

</div>

<navigation>
<ul class="subjects">
    <li><a href="<?php echo
url_for((' /staff/index.php '));?>">Tanks</a></li>
    <li><a href="<?php echo url_for((' /breweryProcess.php '));?>">Brewery
        Process</a></li>
    <li><a href="<?php echo
url_for((' /contact.php '));?>">Contact</a></li>
</ul>
</navigation>
</div>

<?php include(SHARED_PATH . '/public_footer.php'); ?>

```

### 3.3.2. Το αρχείο *contact.php*

Το αρχείο *contact.php* περιέχει έναν χάρτη από το google maps όπου ο χρήστης μπορεί να δει την ακριβή τοποθεσία της ζυθοποιίας και να χρησιμοποιήσει όλες τις λειτουργίες που του διατίθενται από την εφαρμογή όπως zoom in, zoom out προβολή από δορυφόρο, να δει εικόνες από το φυσικό περιβάλλον κλπ. Να σημειώσουμε ότι παραμετροποιήθηκε το μέγεθος του χάρτη έτσι ώστε να συμβαδίζει με το μέγεθος της html σελίδας και δηλώθηκε το εν λόγω γεωγραφικό μήκος και πλάτος με απόκλιση.

```

<!DOCTYPE html>
<html>
  <head>
    <style>
      /* Set the size of the div element that contains the map */
      #map {
        height: 400px; /* The height is 400 pixels */
        width: 80%;/* The width is the width of the web page */
        margin-left: 150px;
      }
    </style>
  </head>
  <body>

    <h3> Google Maps Location</h3>

    <!--The div element for the map -->
    <div id="map"></div>
    <script>

```

```

// Initialize and add the map
function initMap() {
  // The location
  var industrialZone = {lat: 38.332552, lng: 23.614108};
  // The map, centered at location
  var map = new google.maps.Map(
    document.getElementById('map'), {zoom: 6, center: industrialZone});
  // The marker, positioned at location
  var marker = new google.maps.Marker({position: industrialZone, map: map});
}
</script>
<!--Load the API from the specified URL
* The async attribute allows the browser to render the page while the API
loads
* The key parameter will contain your own API key
* The callback parameter executes the initMap() function
-->
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCP4WhWoeF9Vqn2qqqqhR5RB6qpA
LZ50M&callback=initMap">
</script>
</body>
</html>

```

### 3.3.3. Το αρχείο *breweryProcess.php*

Το αρχείο *breweryProcess.php* περιέχει ένα βίντεο το οποίο εξηγεί επακριβώς την διαδικασία με την οποία παρασκευάζεται, εμφιαλώνεται και αποθηκεύεται η μύρα. Χρησιμοποιήθηκε αυτού του είδους το πολυμέσο γιατί αφενός μεν είναι πολύ κατατοπιστικό και αφετέρου διότι δεν υπάρχει σε καμία άλλη html σελίδα του συστήματος πολυμέσο

```

<?php require_once('../private/initialize.php'); ?>
<!DOCTYPE html>
<html>
<body>

<video width="100%" height="40%" controls>
  <source src="/ies_thesis/brewery_project/public/images/The Brewing Process.mp4"
  type="video/mp4">
  <source src="/ies_thesis/brewery_project/public/images/The Brewing Process.ogg"
  type="video/ogg">

</video>

</body>
</html>

```

### 3.4. Ο φάκελος staff

Ο φάκελος staff περιέχει τον φάκελο admins, τον φάκελο subjects καθώς και τα αρχεία index.php, log\_in.php και log\_out.php. Σε αυτήν την εότητα έχουν πρόσβαση μόνο οι εγγεγραμμένοι χρήστες. Για τον τρόπο με τον οποίο εμφανίζονται οι html σελίδες της συγκεκριμένης ενότητας ευθύνεται το αρχείο staff.css από τον φάκελο stylesheets.

#### 3.4.1 Ο φάκελος subject

Ο φάκελος subject περιέχει τα αρχεία που αφορούν όλη την διαχείριση της βάσης δεδομένων των προϊόντων. Τα αρχεία που τον συνθέτουν αναλύονται εκτενέστερα παρακάτω.

##### 3.4.1.1 Το αρχείο create.php

Το αρχείο create.php από τον φάκελο subjects αρχικά φορτώνει όλα τα απαραίτητα αρχεία από το initialize.php και στην συνέχεια πριν εμφανίσει οτιδήποτε στην html σελίδα καλεί την συνάρτηση require\_login() όπου σε περίπτωση που δεν έχει γίνει login κάνει redirect στην log\_in.php αλλιώς συνεχίζει κανονικά. Ελέγχει αν υπάρχει post request και αν υπάρχει καταχωρεί τα νέα δεδομένα με την συνάρτηση insert\_subjects() και την mysqli\_insert-id(). Τέλος κάνει redirect στην show.php ή αν υπάρχει οποιοδήποτε πρόβλημα με το request ξαναφορτώνει την new.php

```
<?php
require_once('../private/initialize.php');
require_login();

if(is_post_request()){
// Handle form values sent by new.php

$tank_number =isset($_POST['tank_number']) ? $_POST['tank_number'] : '';
$malt =isset($_POST['malt']) ? $_POST['malt'] : '';
$water =isset($_POST['water']) ? $_POST['water'] : '';
$yeast =isset($_POST['yeast']) ? $_POST['yeast'] : '';
$hops =isset($_POST['hops']) ? $_POST['hops'] : '';
$recipe =isset($_POST['recipe']) ? $_POST['recipe'] : '';
$timestamp =isset($_POST['timestamp']) ? $_POST['timestamp'] : '';
```

```

$result = insert_subject($tank_number, $malt, $water, $yeast, $shops, $recipe,
$timestamp);
$new_id = mysqli_insert_id($db);
redirect_to(url_for('/staff/subjects/show.php?id=' . $new_id));

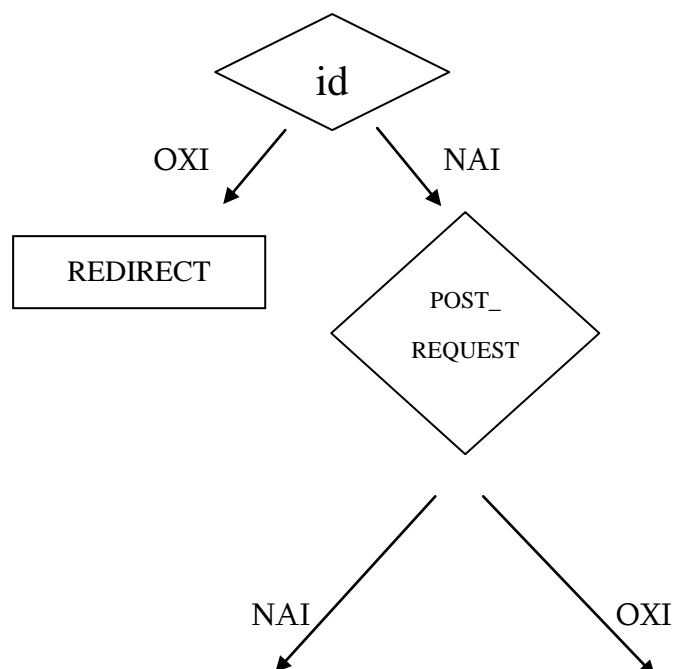
}
else{
    redirect_to(url_for('/staff/subjects/new.php'));
}

?>

```

### 3.4.1.2 Το αρχείο delete.php

Το αρχείο delete.php από τον φάκελο subjects ελέγχει αν υπάρχει ή όχι το id από το request. Αν δεν υπάρχει id κάνει redirect, αλλιώς για το συγκεκριμένο id ελέγχει αν έχει γίνει post\_request. Αν έχει γίνει post\_request το διαγράφει με την συνάρτηση delete\_subject. Αν όμως δεν έχει γίνει post\_request εμφανίζει τα στοιχεία της συγκεκριμένης καταχώρησης. Στη συνέχεια αν πατηθεί το κουμπί Delete Subject τότε κάνει post\_request στην ίδια την html σελίδα και το διαγράφει από την βάση δεδομένων. Ευκρινέστερα απεικονίζεται στο παρακάτω διάγραμμα:





DELETE  
SUBJECT

ΕΜΦΑΝΙΣΗ HTML

```
<?php

require_once('../private/initialize.php');
require_login();

if(!isset($_GET['id'])) {
    redirect_to(url_for('/staff/subjects/index.php'));
}
$id = $_GET['id'];

$tanks = find_tanks_by_id($id);

if(is_post_request()) {
    $result = delete_subject($id);
    redirect_to(url_for('/staff/subjects/index.php'));
}

?>

<?php $page_title = 'Delete Subject'; ?>
<?php include(SHARED_PATH . '/staff_header.php'); ?>

<div id="content">

    <a class="back-link" href="<?php echo url_for('/staff/subjects/index.php');
?>">&laquo; Back to List</a>

    <div class="subject delete">
        <h1>Delete Record</h1>
        <p>Are you sure you want to delete this record?</p>

    <div class="subject show">

        <p>Record: <?php echo h($tanks['tanks_id']); ?></p>

<div class="attributes">
    <dl>
        <dt>Tanks_id</dt>
        <dd><?php echo h($tanks['tanks_id']); ?></dd>
    </dl>
    <dl>
        <dt>Tank_number</dt>
        <dd><?php echo h($tanks['tank_number']); ?></dd>
    </dl>
    <dl>
        <dt>Malt</dt>
        <dd><?php echo h($tanks['malt']); ?></dd>
    </dl>
</div>
```

```

<dl>
  <dt>Water</dt>
  <dd><?php echo h($tanks['water']); ?></dd>
</dl>
<dl>
  <dt>Yeast</dt>
  <dd><?php echo h($tanks['yeast']); ?></dd>
</dl>
<dl>
  <dt>Hops</dt>
  <dd><?php echo h($tanks['hops']); ?></dd>
</dl>
<dl>
  <dt>Recipe</dt>
  <dd><?php echo h($tanks['recipe']); ?></dd>
</dl>
<dl>
  <dt>Timestamp</dt>
  <dd><?php echo h($tanks['timestamp']); ?></dd>
</dl>
</div>
</div>
<form action="<?php echo url_for('/staff/subjects/delete.php?id=' .
h(u($tanks['tanks_id']))); ?>" method="post">
  <div id="operations">
    <input type="submit" name="commit" value="Delete Subject" />
  </div>
</form>
</div>

<?php include(SHARED_PATH . '/staff_footer.php'); ?>

```

### 3.4.1.3 Το αρχείο edit.php

Το αρχείο edit.php από τον φάκελο subjects λειτουργεί με την ίδια λογική όπως το αρχείο delete.php. Αρχικά ελέγχει αν υπάρχει ή όχι το id. Εφόσον υπάρχει ελέγχει αν έχει γίνει post\_request. Αν έχει γίνει ανανεώνει τα στοιχεία της βάσης δεδομένων και τελικά κάνει redirect στην show.php. Αν δεν έχει γίνει post\_request ψάχνει να βρει την καταχώρηση που αντιστοιχεί στην βάση δεδομένων με το δοσμένο id. Τέλος όταν βρεί την καταχώρηση, την απεικονίζει με μορφή φόρμας.

```
<?php
```

```

require_once('../../private/initialize.php');
require_login();

if(!isset($_GET['id'])){
    redirect_to(url_for('/staff/subjects/index.php'));
}
$id = $_GET['id'];

//Form territories initialized

if(is_post_request()){
    // Handle form values sent by new.php
    $tanks=[];
    $tanks['id'] = $id;
    $tanks['tanks_id'] =isset($_POST['tanks_id']) ? $_POST['tanks_id'] : '';
    $tanks['tank_number'] =isset($_POST['tank_number']) ? $_POST['tank_number'] :
'';
    $tanks['malt'] =isset($_POST['malt']) ? $_POST['malt'] : '';
    $tanks['water'] =isset($_POST['water']) ? $_POST['water'] : '';
    $tanks['yeast'] =isset($_POST['yeast']) ? $_POST['yeast'] : '';
    $tanks['hops'] =isset($_POST['hops']) ? $_POST['hops'] : '';
    $tanks['recipe'] =isset($_POST['recipe']) ? $_POST['recipe'] : '';
    $tanks['timestamp'] =isset($_POST['timestamp']) ? $_POST['timestamp'] : '';

    $result = update_subject($tanks);
    if($result === true){
        redirect_to(url_for('/staff/subjects/show.php?id=' . $id));
    }else {
        $errors = $result;
        // var_dump($errors);
    }

    echo "Form parameters<br />";
    echo "Tanks_id: " . $tanks['tanks_id'] . "<br />";
    echo "Tank_number: " . $tanks['tank_number'] . "<br />";
    echo "Malt: " . $tanks['malt'] . "<br />";
    echo "Water: " . $tanks['water'] . "<br />";
    echo "Yeast: " . $tanks['yeast'] . "<br />";
    echo "Hops: " . $tanks['hops'] . "<br />";
    echo "Recipe: " . $tanks['recipe'] . "<br />";
    echo "Timestamp: " . $tanks['timestamp'] . "<br />";

}
else{
    $tanks = find_tanks_by_id($id);
}
?>

<?php $page_title = 'Edit Subject'; ?>
<?php include(SHARED_PATH . '/staff_header.php'); ?>

<div id="content">

```

```

    <a class="back-link" href="<?php echo url_for('/staff/subjects/index.php');
?>">&laquo; Back to List</a>

    <div class="subject edit">
        <h1>Edit Subject</h1>

<?php echo display_errors($errors); ?>

    <form action="<?php echo url_for('/staff/subjects/edit.php?id=' . h(u($id)));
?>" method="post">
        <dl>
            <dt>Tanks_id</dt>
            <dd><input type="text" name="tanks_id" value="<?php echo
$tanks['tanks_id']; ?>" /></dd>
        </dl>
        <dl>
            <dt>Tank_number</dt>
            <dd><input type="text" name="tank_number" value="<?php echo
$tanks['tank_number']; ?>" /></dd>
        </dl>
        <dl>
            <dt>Malt</dt>
            <dd><input type="text" name="malt" value="<?php echo $tanks['malt']; ?>"
/></dd>
        </dl>
        <dl>
            <dt>Water</dt>
            <dd><input type="text" name="water" value="<?php echo $tanks['water']; ?>"
/></dd>
        </dl>
        <dl>
            <dt>Yeast</dt>
            <dd><input type="text" name="yeast" value="<?php echo $tanks['yeast']; ?>"
/></dd>
        </dl>
        <dl>
            <dt>Hops</dt>
            <dd><input type="text" name="hops" value="<?php echo $tanks['hops']; ?>"
/></dd>
        </dl>
        <dl>
            <dt>Recipe</dt>
            <dd><input type="text" name="recipe" value="<?php echo $tanks['recipe'];
?>" /></dd>
        </dl>
        <dl>
            <dt>Timestamp</dt>
            <dd><input type="text" name="timestamp" value="<?php echo
$tanks['timestamp']; ?>" /></dd>
        </dl>
        <div id="operations">
            <input type="submit" value="Edit Subject" />
        </div>

```

```

        </form>

    </div>

</div>

<?php include(SHARED_PATH . '/staff_footer.php'); ?>

```

#### 3.4.1.4 Το αρχείο index.php

Το αρχείο index.php από τον φάκελο subjects συγκεντρώνει όλα τα δεδομένα που έχει βρει στην βάση δεδομένων χρησιμοποιώντας την συνάρτηση find\_all\_tanks(). Στην συνέχεια χρησιμοποιώντας την συνάρτηση mysqli\_fetch\_assoc() για τα δεδομένα που έχει, τα απεικονίζει σε μορφή πίνακα όπου στην άκρη του παρέχει τις επιλογές View, Edit και Delete. Αξίζει να σημειωθεί ότι σε όσα σημεία της html σελίδας γίνεται αλληλεπίδραση με τα δεδομένα της βάσης εκείνα φιλτράρονται από τις συναρτήσεις h και u και στη συνέχεια χρησιμοποιούνται.

```

<?php require_once('../private/initialize.php'); ?>

<?php
require_login();
$tanks_set = find_all_tanks();

?>

<?php $page_title = 'Subjects'; ?>
<?php include(SHARED_PATH . '/staff_header.php'); ?>

<div id="content">
    <div class="subjects_listing">
        <h1>Subjects</h1>

        <div class="actions">
            <a class="action" href="<?php echo
url_for('/staff/subjects/new.php'); ?>">Create New Subject</a>
        </div>

<table class="list">
    <tr>
        <th>tanks_id</th>
        <th>tank_number</th>
        <th>malt</th>
        <th>water</th>
        <th>yeast</th>

```

```

        <th>hops</th>
        <th>recipe</th>
        <th>timestamp</th>
        <th>&nbsp;</th>
        <th>&nbsp;</th>
    <th>&nbsp;</th>
</tr>

<?php while($tanks = mysqli_fetch_assoc($tanks_set)) { ?>
    <tr>
        <td><?php echo h($tanks['tanks_id']); ?></td>
        <td><?php echo h($tanks['tank_number']); ?></td>
        <td><?php echo h($tanks['malt']); ?></td>
        <td><?php echo h($tanks['water']); ?></td>
        <td><?php echo h($tanks['yeast']); ?></td>
        <td><?php echo h($tanks['hops']); ?></td>
        <td><?php echo h($tanks['recipe']); ?></td>
        <td><?php echo h($tanks['timestamp']); ?></td>
        <td><a class="action" href="<?php echo
url_for('/staff/subjects/show.php?id=' . h(u($tanks['tanks_id'])));
?>">View</a></td>
        <td><a class="action" href="<?php echo
url_for('/staff/subjects/edit.php?id=' . h(u($tanks['tanks_id'])));
?>">Edit</a></td>
        <td><a class="action" href="<?php echo
url_for('/staff/subjects/delete.php?id=' . h(u($tanks['tanks_id'])));
?>">Delete</a></td>
    </tr>
    <?php } ?>
</table>
<?php
    mysqli_free_result($tanks_set);
?>

</div>

</div>

<?php include(SHARED_PATH . '/staff_footer.php'); ?>

```

### 3.4.1.5 Το αρχείο new.php

Το αρχείο new.php από τον φάκελο subjects είναι η html σελίδα που εμφανίζει μια φόρμα με όλα τα στοιχεία μιας καταχώρησης στα οποία μπορεί να επέμβει ο χρήστης. Πατώντας το κουμπί create subject τα δεδομένα μεταφέρονται στην html σελίδα create.php με την μέθοδο post.

```

<?php

require_once('../../private/initialize.php');
require_login();

<?php $page_title = 'Create Subject'; ?>
<?php include(SHARED_PATH . '/staff_header.php'); ?>

<div id="content">

    <a class="back-link" href="<?php echo url_for('/staff/subjects/index.php');
?>">&laquo; Back to List</a>

    <div class="subject new">
        <h1>Create Subject</h1>

        <form action="<?php echo url_for('/staff/subjects/create.php'); ?>"
method="post">
            <dl>
                <dt>Tank number</dt>
                <dd><input type="text" name="tank_number" value="" /></dd>
            </dl>
            <dl>
                <dt>Malt</dt>
                <dd><input type="text" name="malt" value="" /></dd>
            </dl>
            <dl>
                <dt>Water</dt>
                <dd><input type="text" name="water" value="" /></dd>
            </dl>
            <dl>
                <dt>Yeast</dt>
                <dd><input type="text" name="yeast" value="" /></dd>
            </dl>
            <dl>
                <dt>Hops</dt>
                <dd><input type="text" name="hops" value="" /></dd>
            </dl>
            <dl>
                <dt>Recipe</dt>
                <dd><input type="text" name="recipe" value="" /></dd>
            </dl>
            <dl>
                <dt>Timestamp</dt>
                <dd><input type="text" name="timestamp" value="" /></dd>
            </dl>
            <div id="operations">
                <input type="submit" value="Create Subject" />
            </div>
        </form>

    </div>

```

```
</div>
```

```
<?php include(SHARED_PATH . '/staff_footer.php'); ?>
```

### 3.4.1.6 Το αρχείο show.php

Το αρχείο show.php από τον φάκελο subjects ελέγχει το δοσμένο id και στην συνέχεια καλεί την συνάρτηση find\_tanks\_by\_id(). Αφού καταχωρήσει στον πίνακα tanks τα δεδομένα που επέστρεψε η find\_tanks\_by\_id(), παρουσιάζει σε μορφή πίνακα τα δεδομένα του tanks πίνακα.

```
<?php require_once('../private/initialize.php'); ?>
require_login();
```

```
<?php
$id = isset($_GET['id']) ? $_GET['id'] : '1';
//$id = $_GET['id'] ?? '1'; // PHP > 7.0
```

```
$tanks = find_tanks_by_id($id);
?>
```

```
<?php $page_title = 'Show Tank'; ?>
<?php include(SHARED_PATH . '/staff_header.php'); ?>
```

```
<div id="content">
```

```
    <a class="back-link" href="<?php echo url_for('/staff/subjects/index.php');
?>">&laquo; Back to List</a>
```

```
    <div class="subject show">
```

```
        <h1>Record: <?php echo h($tanks['tanks_id']); ?></h1>
```

```
<div class="attributes">
```

```
    <dl>
        <dt>Tanks_id</dt>
        <dd><?php echo h($tanks['tanks_id']); ?></dd>
    </dl>
```

```
    <dl>
        <dt>Tank_number</dt>
        <dd><?php echo h($tanks['tank_number']); ?></dd>
    </dl>
```

```
    <dl>
        <dt>Malt</dt>
        <dd><?php echo h($tanks['malt']); ?></dd>
    </dl>
```

```
    <dl>
        <dt>Water</dt>
```



```

        <dd><?php echo h($tanks['water']); ?></dd>
    </dl>
    <dl>
        <dt>Yeast</dt>
        <dd><?php echo h($tanks['yeast']); ?></dd>
    </dl>
    <dl>
        <dt>Hops</dt>
        <dd><?php echo h($tanks['hops']); ?></dd>
    </dl>
    <dl>
        <dt>Recipe</dt>
        <dd><?php echo h($tanks['recipe']); ?></dd>
    </dl>
    <dl>
        <dt>Timestamp</dt>
        <dd><?php echo h($tanks['timestamp']); ?></dd>
    </dl>
</div>
</div>
</div>

```

### 3.4.2 Ο φάκελος *admins*

Ο φάκελος *admins* περιέχει τα αρχεία που αφορούν όλη την διαχείριση της βάσης δεδομένων των χρηστών με εξουσιοδότηση (εργαζόμενων). Τα αρχεία που τον συνθέτουν αναλύονται εκτενέστερα παρακάτω.

#### 3.4.2.1 Το αρχείο *delete.php*

Το αρχείο *delete.php* από τον φάκελο *admins* αρχικά ελέγχει το δοσμένο *id*. Αν δεν υπάρχει κάνει *redirect* στην *index.php*. Στην συνέχεια ελέγχει αν έχει πατηθεί το κουμπί *Delete Admin*. Αν έχει πατηθεί, έχει γίνει δηλαδή *post\_request* από την σελίδα *delete.php* τότε καλεί την *delete\_admin()* και διαγράφει την καταχώρηση από την βάση δεδομένων. Τέλος εμφανίζει το μήνυμα *Admin deleted* και κάνει *redirect* στην *index.php*.

```

<?php
require_once('../private/initialize.php');
require_login();

```

```

if(!isset($_GET['id'])) {
    redirect_to(url_for('/staff/admins/index.php'));
}
$id = $_GET['id'];

if(is_post_request()) {
    $result = delete_admin($id);
    $_SESSION['message'] = 'Admin deleted.';
    redirect_to(url_for('/staff/admins/index.php'));
} else {
    $admin = find_admin_by_id($id);
}

?>

<?php $page_title = 'Delete Admin'; ?>
<?php include(SHARED_PATH . '/staff_header.php'); ?>

<div id="content">

    <a class="back-link" href="<?php echo url_for('/staff/admins/index.php');
?>">&laquo; Back to List</a>

    <div class="admin delete">
        <h1>Delete Admin</h1>
        <p>Are you sure you want to delete this admin?</p>
        <p class="item"><?php echo h($admin['username']); ?></p>

        <form action="<?php echo url_for('/staff/admins/delete.php?id=' .
h(u($admin['id']))); ?>" method="post">
            <div id="operations">
                <input type="submit" name="commit" value="Delete Admin" />
            </div>
        </form>
    </div>

</div>

<?php include(SHARED_PATH . '/staff_footer.php'); ?>

```

### 3.4.2.2 Το αρχείο edit.php

Το αρχείο edit.php από τον φάκελο admins παρουσιάζει μια φόρμα όπου ο χρήστης εισάγει το username, το password και επιβεβαιώνει το password. Πατώντας το κουμπί Edit Admin γίνεται redirect στην ίδια την σελίδα με την μέθοδο post\_request. Αφού έχει γίνει post\_request καλείται η συνάρτηση update\_admin() όπου αναλαμβάνει να ενημερώσει την βάση δεδομένων. Εφόσον τα καταφέρει εμφανίζει το μήνυμα Admin

updated και κάνει redirect στην show.php. Σε διαφορετική περίπτωση ψάχνει ξανά να βρει το σετ δεδομένων από την βάση χρησιμοποιώντας την συνάρτηση find\_admin\_by\_id().

```
<?php

require_once('../private/initialize.php');
require_login();

if(!isset($_GET['id'])) {
    redirect_to(url_for('/staff/admins/index.php'));
}
$id = $_GET['id'];

if(is_post_request()) {
    $admin = [];
    $admin['id'] = $id;
    $admin['username'] = $_POST['username'] ? $_POST['username'] : '';
    $admin['password'] = $_POST['password'] ? $_POST['password'] : '';
    $admin['confirm_password'] = $_POST['confirm_password'] ?
$_POST['confirm_password'] : '';

    $result = update_admin($admin);
    if($result === true) {
        $_SESSION['message'] = 'Admin updated.';
        redirect_to(url_for('/staff/admins/show.php?id=' . $id));
    } else {
        $errors = $result;
    }
} else {
    $admin = find_admin_by_id($id);
}

?>

<?php $page_title = 'Edit Admin'; ?>
<?php include(SHARED_PATH . '/staff_header.php'); ?>

<div id="content">

    <a class="back-link" href="<?php echo url_for('/staff/admins/index.php');
?>">&laquo; Back to List</a>

    <div class="admin edit">
        <h1>Edit Admin</h1>

        <?php echo display_errors($errors); ?>

        <form action="<?php echo url_for('/staff/admins/edit.php?id=' . h(u($id)));
?>" method="post">
```

```

        <dl>
            <dt>Username</dt>
            <dd><input type="text" name="username" value="<?php echo
h($admin['username']); ?>" /></dd>
        </dl>

        <dl>
            <dt>Password</dt>
            <dd><input type="password" name="password" value="" /></dd>
        </dl>

        <dl>
            <dt>Confirm Password</dt>
            <dd><input type="password" name="confirm_password" value="" /></dd>
        </dl>
        <p>
            Passwords should be at least 12 characters and include at least one
            uppercase letter, lowercase letter, number, and symbol.
        </p>
        <br />

        <div id="operations">
            <input type="submit" value="Edit Admin" />
        </div>
    </form>

</div>

</div>

<?php include(SHARED_PATH . '/staff_footer.php'); ?>

```

### 3.4.2.3 Το αρχείο index.php

Το αρχείο index.php από τον φάκελο admins αποθηκεύει στον πίνακα \$admin\_set το αποτέλεσμα της συνάρτησης find\_all\_admins(). Στην συνέχεια μπαίνει σε ένα βρόχο επανάληψης όπου για κάθε στοιχείο του πίνακα \$admin\_set θα παρουσιάσει σε έναν πίνακα το id, το username και τα κουμπιά View, Edit, Delete. Τέλος καλεί την συνάρτηση mysqli\_free\_result() για να απελευθερώσει την μνήμη του συστήματος.

```

<?php
require_once('../private/initialize.php');
require_login();

```

```

$admin_set = find_all_admins();

?>

<?php $page_title = 'Admins'; ?>
<?php include(SHARED_PATH . '/staff_header.php'); ?>

<div id="content">
  <div class="admins listing">
    <h1>Admins</h1>

    <div class="actions">
      <a class="action" href="<?php echo url_for('/staff/admins/new.php');
?>">Create New Admin</a>
    </div>

    <table class="list">
      <tr>
        <th>ID</th>
        <!-- <th>First</th>
        <th>Last</th>
        <th>Email</th> -->
        <th>Username</th>
        <th>&nbsp;</th>
        <th>&nbsp;</th>
        <th>&nbsp;</th>
      </tr>

      <?php while($admin = mysqli_fetch_assoc($admin_set)) { ?>
        <tr>
          <td><?php echo h($admin['id']); ?></td>

          <td><?php echo h($admin['username']); ?></td>
          <td><a class="action" href="<?php echo
url_for('/staff/admins/show.php?id=' . h(u($admin['id'])); ?>">View</a></td>
          <td><a class="action" href="<?php echo
url_for('/staff/admins/edit.php?id=' . h(u($admin['id'])); ?>">Edit</a></td>
          <td><a class="action" href="<?php echo
url_for('/staff/admins/delete.php?id=' . h(u($admin['id'])); ?>">Delete</a></td>
        </tr>
      <?php } ?>
    </table>

    <?php
      mysqli_free_result($admin_set);
    ?>
  </div>
</div>

<?php include(SHARED_PATH . '/staff_footer.php'); ?>

```

### 3.4.2.4 Το αρχείο new.php

Το αρχείο new.php από τον φάκελο admins λειτουργεί με την ίδια λογική με το αρχείο new.php από τον φάκελο subjects με την διαφορά ότι όταν πατηθεί το κουμπί create Admin και γίνει post\_request τότε καλείται η συνάρτηση insert\_admin() όπου αναλαμβάνει να ενημερώσει την βάση δεδομένων. Επίσης με την συνάρτηση mysqli\_insert\_id καταχωρεί στην μεταβλητή \$new\_id το καινούριο id. Τέλος κάνει redirect στην show.php με το καινούριο id. Σε περίπτωση που κάτι έχει πια στραβά εμφανίζει τα λάθη.

```
<?php

require_once('../../private/initialize.php');
require_login();

if(is_post_request()) {
    $admin = [];

    $admin['username'] = $_POST['username'] ? $_POST['username'] : '';
    $admin['password'] = $_POST['password'] ? $_POST['password'] : '';
    $admin['confirm_password'] = $_POST['confirm_password']
    ? $_POST['confirm_password'] : '';

    $result = insert_admin($admin);
    if($result === true) {
        $new_id = mysqli_insert_id($db);
        $_SESSION['message'] = 'Admin created.';
        redirect_to(url_for('/staff/admins/show.php?id=' . $new_id));
    } else {
        $errors = $result;
    }
} else {
    // display the blank form
    $admin = [];

    $admin["username"] = '';
    $admin['password'] = '';
    $admin['confirm_password'] = '';
}

?>

<?php $page_title = 'Create Admin'; ?>
<?php include(SHARED_PATH . '/staff_header.php'); ?>

<div id="content">
```

```

<a class="back-link" href="<?php echo url_for('/staff/admins/index.php');
?>">&laquo; Back to List</a>

<div class="admin new">
  <h1>Create Admin</h1>

  <?php echo display_errors($errors); ?>

  <form action="<?php echo url_for('/staff/admins/new.php'); ?>" method="post">

    <dl>
      <dt>Username</dt>
      <dd><input type="text" name="username" value="<?php echo
h($admin['username']); ?>" /></dd>
    </dl>

    <dl>
      <dt>Password</dt>
      <dd><input type="password" name="password" value="" /></dd>
    </dl>

    <dl>
      <dt>Confirm Password</dt>
      <dd><input type="password" name="confirm_password" value="" /></dd>
    </dl>
    <p>
      Passwords should be at least 12 characters and include at least one
      uppercase letter, lowercase letter, number, and symbol.
    </p>
    <br />

    <div id="operations">
      <input type="submit" value="Create Admin" />
    </div>
  </form>

</div>

</div>

<?php include(SHARED_PATH . '/staff_footer.php'); ?>

```

### 3.4.2.5 Το αρχείο show.php

Το αρχείο show.php από τον φάκελο admins χρησιμοποιεί την συνάρτηση find\_admin\_by\_id() και για το δοσμένο id καταχωρεί στον πίνακα \$admin τα δεδομένα.

Στην συνέχεια παρουσιάζει το username του admin και παρέχει στον χρήστη τις επιλογές edit και delete.

```
<?php

require_once('../private/initialize.php');
require_login();

$id = $_GET['id'] ? $_GET['id'] : '1';
$admin = find_admin_by_id($id);

?>

<?php $page_title = 'Show Admin'; ?>
<?php include(SHARED_PATH . '/staff_header.php'); ?>

<div id="content">

    <a class="back-link" href="<?php echo url_for('/staff/admins/index.php');
?>"&laquo; Back to List</a>

    <div class="admin show">

        <h1>Admin: <?php echo h($admin['username']); ?></h1>

        <div class="actions">
            <a class="action" href="<?php echo url_for('/staff/admins/edit.php?id=' .
h(u($admin['id']))); ?>">Edit</a>
            <a class="action" href="<?php echo url_for('/staff/admins/delete.php?id=' .
h(u($admin['id']))); ?>">Delete</a>
        </div>

        <div class="attributes">
            <dl>
                <dt>Username</dt>
                <dd><?php echo h($admin['username']); ?></dd>
            </dl>
        </div>

    </div>

</div>
```

### 3.4.3 Το αρχείο index.php

Το αρχείο index.php του φάκελου staff είναι η κεντρική σελίδα για τους εξουσιοδοτημένους χρήστες (εργαζόμενοι εταιρείας) από την οποία έχουν την δυνατότητα να διαχειριστούν τα δεδομένα στην βάση δεδομένων των προϊόντων καθώς



επίσης και να διαχειριστούν τα προσωπικά τους δεδομένα(username,password) . Επιλέγοντας από το μενού την επιλογή subjects διαχειρίζονται τα δεδομένα στην βάση δεδομένων των προϊόντων, ενώ από την επιλογή admins διαχειρίζονται τα προσωπικά τους δεδομένα.

```
<?php require_once('../private/initialize.php'); ?>

<?php require_login();?>
<?php $page_title = 'Staff Menu'; ?>
<?php include(SHARED_PATH . '/staff_header.php'); ?>

<div id="content">
  <div id="main-menu">
    <h2>Main Menu</h2>
    <ul>
      <li><a href="<?php echo url_for('/staff/subjects/index.php');?>">Subjects</a>
      </li>

      <li><a href="<?php echo url_for('/staff/admins/index.php');?>">Admins</a>
      </li>
    </ul>
  </div>
</div>

<?php include(SHARED_PATH . '/staff_footer.php'); ?>
```

#### 3.4.4 Το αρχείο log\_in.php

Το αρχείο log\_in.php του φάκελου staff είναι υπεύθυνο για τον έλεγχο πρόσβασης στη βάση δεδομένων προϊόντων και χρηστών. Όταν καλεστεί από μια html σελίδα ελέγχει αν έχει γίνει post request. Αν έχει γίνει post request κάνει τους απαραίτητους ελέγχους πιστοποίησης και κάνει redirect εφόσον πιστοποιηθεί ο χρήστης. Σε περίπτωση που δεν πιστοποιηθεί ο χρήστης εμφανίζει το αντίστοιχο μήνυμα. Επίσης σε περίπτωση που δεν έχει γίνει post request εμφανίζει μια φόρμα που ζητά από τον επισκέπτη της ιστοσελίδας username και password.

```
<?php
require_once('../private/initialize.php');

$errors = [];
$username = '';
$password = '';

if(is_post_request()) {
```

```

$username = $_POST['username'] ? $_POST['username'] : '';
$password = $_POST['password'] ? $_POST['password'] : '';

//Validations
if(is_blank($username)){
    $errors[] = "Username cannot be blank.";
}
if(is_blank($password)){
    $errors[] = "Password cannot be blank.";
}

//if there were no errors, try to login
if(empty($errors)){
    $admin = find_admin_by_username($username);
    if($admin){
        if(password_verify($password, $admin['hashed_password'])){
            //password matches
            log_in_admin($admin);
            redirect_to(url_for('/staff/index.php'));
        }else{
            //username found, but password does not match
            $errors[] = "Log in was unsuccessful.";
        }
    }
    }else {
        //no username found
        $errors[] = "Log in was unsuccessful.";
    }
}

}

?>

<?php $page_title = 'Log in'; ?>
<?php include(SHARED_PATH . '/staff_header.php'); ?>

<div id="content">
    <h1>Log in</h1>

    <?php echo display_errors($errors); ?>

    <form action="log_in.php" method="post">
        Username:<br />
        <input type="text" name="username" value="<?php echo h($username); ?>" /><br
    />
        Password:<br />
        <input type="password" name="password" value="" /><br />
        <input type="submit" name="submit" value="Submit" />
    </form>

</div>

```

```
<?php include(SHARED_PATH . '/staff_footer.php'); ?>
```

### 3.4.5 Το αρχείο *log\_out.php*

Το αρχείο *log\_out.php* του φάκελου *staff* καλεί την συνάρτηση *log\_out\_admin()* και διαγράφει από το συγκεκριμένο *session* ότι δεδομένο αφορά τον χρήστη. Στην συνέχεια κάνει *redirect* στην *log\_in.php*.

```
<?php
require_once('../private/initialize.php');

log_out_admin();
redirect_to(url_for('/staff/log_in.php'));

?>
```

## 3.5 Ο φάκελος *stylesheet*

Ο φάκελος *stylesheet* περιέχει τα αρχεία *public.css* και *staff.css*. Το αρχείο *public.css* είναι υπεύθυνο για την διαμόρφωση των *html* σελίδων που δεν απαιτούν πρόσβαση και είναι διαθέσιμες σε όλους. Αντίθετα το αρχείο *staff.css* απευθύνεται στην διαμόρφωση των *html* σελίδων που είναι απαραίτητος ο έλεγχος πρόσβασης. Ο κώδικας και για τα δύο αρχεία παρατίθεται στη συνέχεια.

### 3.5.1 Το αρχείο *public.css*

```
/* Site Colors:
#e03e52 - pink
#c8ced2 - grey
#f07b27 - orange
#786758 - brown
*/

html {
  height: 100%;
  width: 100%;
}

body {
  width: 1100px;
  height: 100%;
  margin: auto;
```

```

border: 0;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 13px;
line-height: 15px;
background: #c8ced2;
}

header {
height: 100px;
width: 1100px;
margin: 0;
padding: 0;
text-align: left;
background: #1a207c;
color: #FFFFFF;
}

header h1 {
padding: 1em;
margin: 0;
text-align: center;
}

#main {
width: 1100px;
height: 100%;
min-height: 800px;
margin: 0;
padding: 0;
background: #c8ced2;
}

footer {
clear: both;
height: 3em;
margin: 0;
padding: 1em;
text-align: center;
background: #e03e52;
color: #FFFFFF;
}

/* Navigation */

navigation {
float: left;
width: 200px;
height: 100%;
background: #7a6855;
}

navigation a {
color: #FFFFFF;
}

```

```

    text-decoration: none;
}

navigation a:hover {
    color: #e03e52;
    text-decoration: underline;
}

a {
    color: #E03E52;
    text-decoration: none
}

a:hover {
    text-decoration: underline
}

ul.subjects {
    margin-top: 20px;
    margin-left: 20px;
    padding: 0;
    list-style: none;
}

ul.subjects > li {
    margin-bottom: 1em;
}

ul.pages {
    margin-left: 15px;
    padding: 0;
    list-style: none;
    font-weight: normal;
}

.selected > a {
    color: #f07b27;
    font-style: normal;
}

navigation li {
    font-size: 16px;
    line-height: 1.5em;
}

/* Page Content */

#hero-image {
    float: left;
    margin-bottom: 40px;
    padding: 0;
    height: 200px;
    width: 900px;
}

```

```

}

#page {
  float: right;
  width: 900px;
  height: 100%;
  padding-left: 0;
  vertical-align: top;
  background: #ffffff;
}

#page h1 {
  font-size: 16px;
  color: #000000;
  line-height: normal;
}

#page h2 {
  font-size: 14px;
  color: #000000;
  line-height: normal;
}

#page p {
  font-family: "Gill Sans", "Gill Sans MT", "Myriad Pro", "DejaVu Sans Condensed",
  Helvetica, Arial, "sans-serif" font-size: 12px;
  color: #000000;
  padding: 0px;
  line-height: normal;
}

#page h3 {
  font-size: 14px;
  color: #e03e52;
}

#content {
  margin-top: 0px;
  margin-left: 10px;
  margin-right: 10px;
}

#content p {
  font-size: 16px;
  line-height: normal;
}

#content li {
  padding: 5px;
  font-size: 16px;
}

#disclaimer {

```

```

margin-left: 10px;
font-size: 14px;
line-height: normal;
clear: both;
padding: 1em;
}

/* Service blocks on homepage */

#service-blocks {
margin-top: 50px;
}

#service-blocks .service {
float: left;
width: 250px;
margin-right: 40px;
}

/* Learn More links */

a.Learnmore {
color: #E03E52;
font-size: 14px;
font-weight: bold;
text-decoration: none
}

a.Learnmore:hover {
text-decoration: underline
}

```

### 3.5.2 Το αρχείο staff.css

```

html {
height: 100%;
width: 100%;
}

body {
height: 980px;
width: 100%;
margin: 0;
padding: 0;
}

header {
width: 980px;
height: 40px;
margin: 0 0 10px 0;
padding: 0;
background: #0055DD;
color: white;
}

```

```

}

header h1 {
  margin: 0;
  padding: 5px;
  font-size: 20px;
  font-family: Verdana, Arial, Helvetica, sans-serif;
  text-align: center;
}

navigation {
  width: 980px;
  margin: 0;
  padding: 0;
  text-align: center;
}

navigation ul {
  width: 980px;
  list-style: none;
  padding: 0;
  margin: 0;
}

navigation ul li {
  display: inline;
  margin-right: 1em;
}

#content {
  width: 920px;
  min-height: 500px;
  padding: 0 30px;
}

footer {
  width: 940px;
  height: 20px;
  margin: 0 0 10px 0;
  padding: 10px 20px;
  background: #0055DD;
  color: white;
}

/* Actions */

.actions {
  margin-bottom: 1em;
}

/* Lists */

table {

```



```

border-collapse: collapse;
vertical-align: top;
}

table.List {
width: 100%;
}

table.List tr td {
border: 1px solid #999999;
}

table.List tr th {
border: 1px solid #0055DD;
background: #0055DD;
color: white;
text-align: left;
}

/* Forms */

dl {
clear: both;
overflow: hidden;
margin: 0.5em 0;
}

dt {
float: left;
font-weight: bold;
width: 125px;
}

dd {
float: left;
margin-left: 1em;
}

#operations {
clear: both;
margin: 1em 0 1em 75px;
}

/* Errors */

.errors {
display: inline-block;
border: 2px solid red;
color: red;
padding: 1em;
margin-bottom: 1em;
}

.errors ul {

```

```
margin-bottom: 0;  
padding-left: 1em;  
}
```

Στην παρούσα διπλωματική εργασία αναπτύχθηκε μια εφαρμογή τύπου ERP που απευθύνεται σε μικροζυθοποιία. Ο στόχος της εργασίας επετεύχθη καθώς δημιουργήθηκε ένα σύστημα ικανό να καλύψει τις ανάγκες επεξεργασίας και διάθεσης των δεδομένων που είναι ζωτικής σημασίας για την ζυθοποιία. Η εφαρμογή τύπου ERP είναι ουσιαστικά ένα website το οποίο μπορούν να επισκέπτονται τόσο απλοί χρήστες για να περιηγηθούν όπως ακριβώς και σε ένα συνηθισμένο ιστότοπο, όσο και εξουσιοδοτημένοι χρήστες (εργαζόμενοι της μικροζυθοποιίας) για να επεξεργαστούν τη βάση δεδομένων των προϊόντων της μικροζυθοποιίας.

Η εφαρμογή διαθέτει τρεις λειτουργίες, εκ των οποίων μία λειτουργία αποκλειστικά για εξουσιοδοτημένους χρήστες και δύο λειτουργίες για απλούς χρήστες αλλά και εξουσιοδοτημένους.

**Επιλογή ιστοσελίδας 1:** Η επιλογή για εξουσιοδοτημένους χρήστες αφορά τη διαχείριση της βάσης δεδομένων των προϊόντων (επεξεργασία μιας καταχώρησης , δημιουργία μιας νέας καταχώρησης, διαγραφή μιας καταχώρησης) . Με αυτή την λειτουργία οι εργαζόμενοι της ζυθοποιίας από τα τμήματα της παραγωγής, της αποθήκης και του ποιοτικού ελέγχου μπορούν να ενημερώνονται σχετικά με το πόσες φιάλες μύρας έχουν παραχθεί (τμήμα παραγωγής), πόσες έχουν εγκριθεί από τον ποιοτικό έλεγχο (τμήμα ποιοτικού ελέγχου), πόσες είναι διαθέσιμες προς πώληση (τμήμα αποθήκης). Η λειτουργία αυτή είναι διαθέσιμη μέσω της επιλογής Tanks στην κεντρική σελίδα.

**Επιλογή ιστοσελίδας 2:** Η επιλογή αυτή πέρα από τους εξουσιοδοτημένους χρήστες αφορά και τους απλούς επισκέπτες της ιστοσελίδας. Μέσω αυτής παρέχονται πληροφορίες σχετικά με την τοποθεσία της μικροζυθοποιίας. Η λειτουργία αυτή είναι διαθέσιμη μέσω της επιλογής Contact στην κεντρική σελίδα.

**Επιλογή ιστοσελίδας 3 :** Αντίστοιχα και η εν λόγω επιλογή είναι διαθέσιμη τόσο από τους εξουσιοδοτημένους χρήστες, όσο κι από τους απλούς επισκέπτες. Μέσω αυτής παρέχονται

πληροφορίες σχετικά με τη διαδικασία της παραγωγής της μπύρας μέσω της προβολής ενός αντίστοιχου video. Η λειτουργία αυτή είναι διαθέσιμη μέσω της επιλογής Brewery Process στην κεντρική σελίδα.

Για την υλοποίηση όλων των παραπάνω λειτουργιών έγινε συγγραφή πηγαίου κώδικα και χρησιμοποιήθηκε παραμετροποιήσιμο λογισμικό. Ειδικότερα, χρησιμοποιήθηκε λογισμικό open source για το στήσιμο του server και επιλέχθηκαν τα windows 10 σαν λειτουργικό σύστημα. Το λογισμικό open source που χρησιμοποιήθηκε είναι:

- Apache web server
- Php
- MySQL

Η τριάδα Apache, Php, MySQL συναντάται συνήθως μαζί με κάποια έκδοση Linux λειτουργικού και αναφέρεται ως σύστημα LAMP (Linux – Apache – MySQL- Php). Στην περίπτωσή μας όμως χρησιμοποιήθηκαν τα windows 10 αντί για κάποια έκδοση Linux, οπότε το σύστημά μας είναι WAMP (Windows - Apache – MySQL- Php). Η επιλογή των windows σαν λειτουργικό σύστημα θεωρήθηκε ουσιαστικά αυτονόητη διότι ελάχιστες βιομηχανίες παγκοσμίως χρησιμοποιούν λειτουργικό σύστημα διαφορετικό των windows.

Έχοντας υπόψιν τις ανάγκες διάθεσης δεδομένων για μια μικροζυθοποιία υλοποιήθηκε το εν λόγω σύστημα. Είναι ένα πλήρως λειτουργικό σύστημα που εξυπηρετεί όλους τους εργαζόμενους και μη (όπως οι επισκέπτες της ιστοσελίδας). Χαρακτηρίζεται ως ασφαλές σύστημα διότι η πρόσβαση στη βάση δεδομένων απαιτεί εξουσιοδότηση και έχει γίνει όσο το δυνατόν μεγαλύτερη προσπάθεια στο να αποφευχθεί η παραβίαση της βάσης δεδομένων. Για αυτό τον λόγο έχουν χρησιμοποιηθεί συναρτήσεις που αποτρέπουν επιθέσεις *SQL Injection* και κάνουν *sanitize data* για την βάση δεδομένων. Ακόμα όμως και αν παραβιασθεί η βάση δεδομένων, τα passwords των χρηστών (admins) έχουν υποστεί κρυπτογράφηση, οπότε δεν αποκαλύπτονται.

Οι προτάσεις-βελτιώσεις που θα μπορούσαν να χρησιμοποιηθούν είναι:

- Πρόσθετα μέτρα προστασίας με μορφή hardware και software . Για παράδειγμα θα μπορούσαν να υπάρχουν αισθητήρες δακτυλικού αποτυπώματος, αναγνώρισης προσώπου ή ακόμα και φωνής ούτως ώστε σε συνεργασία με τον κωδικό πρόσβασης

και το username να καθιστούσαν την πρόσβαση σε μη εξουσιοδοτημένα άτομα πρακτικά αδύνατη. Το παραπάνω μέτρο σε καμία περίπτωση δεν αποτελεί ακραίο μέτρο διότι το ζήτημα της ασφάλειας δεδομένων σε επιχειρήσεις είναι ζωτικής σημασίας καθότι μπορεί να εκτεθούν πληροφορίες στο ευρύ κοινό, τόσο σε προσωπικό επίπεδο ,όσο και σε επιχειρηματικό.

- Αντίγραφο Ασφαλείας των δεδομένων. Η βάση δεδομένων που χρησιμοποιήθηκε έχει στοιχεία πολύτιμα για την εύρυθμή λειτουργία της επιχείρησης, για αυτό το λόγο θα ήταν καλό ανά τακτά χρονικά διαστήματα το σύστημα να δημιουργούσε ένα αντίγραφο ασφαλείας αυτόματα και να αποθηκευόταν και στον server τοπικά και σε μία υπηρεσία cloud.
- Logging Tools. Τα εν λόγω συστήματα αποτελούν προγράμματα τα οποία μπορούν να ενσωματωθούν στον πηγαίο κώδικα και να καταγράφουν events εισόδου και εξόδου χρηστών, όπως επίσης και να απεικονίζουν σε real time την κατάσταση της ιστοσελίδας. Με αυτό τον τρόπο ο διαχειριστής έχει την δυνατότητα να παρακολουθεί την ιστοσελίδα και να διαπιστώσει αν υπάρχουν τυχόν κενά ασφαλείας και αστοχίες του συστήματος.
- Επέκταση ιστοσελίδας ως προς την διασύνδεση όλων των τμημάτων που απαρτίζουν την ζυθοποιία. Ακόμη θα μπορούσε να βελτιωθεί η λειτουργικότητα της ιστοσελίδας διαθέτοντας στον χρήστη παραπάνω δυνατότητες διαχείρισης της βάσης δεδομένων, όπως για παράδειγμα εύρεση μιας συγκεκριμένης καταχώρησης.

## ΒΙΒΛΙΟΓΡΑΦΙΑ - ΠΗΓΕΣ

---

### Αναφορές

- Beheshti, H. (2006). What managers should know about ERP/ERP II. *Management Research News*, σσ. 184-193.
- Duckett, J. (2011). *HTML & CSS : Design and Build Websites*. Εκδόσεις John Wiley & Sons
- Gupta, A. (2000). Enterprise resource planning: the emerging organizational value. *Industrial Management & Data Systems*, σσ. 114-118.
- Julie C. Meloni, M. (2004). *Μάθετε PHP,MySQL και Apache - Όλα σε ένα*. Εκδόσεις Γκιούρδας.
- Thomson, L. W. (2006). *Ανάπτυξη Web Εφαρμογών με PHP και MySQL* (3η εκδ.). (Γ. Μαρία, Μεταφρ.) Εκδόσεις Γκιούρδας.
- Ullman, L. (2005). *Εισαγωγή στην PHP για τον παγκόσμιο ιστό* (2η εκδ.). Εκδόσεις Κλειδάριθμος.
- Πολλάλης, Γ. Α., Γιαννακόπουλος, Δ. Ι., & Παπουτσής, Ι. (2004). *Πληροφοριακά Συστήματα*. Αθήνα: Σταμούλη.

### Πηγές από το Διαδίκτυο

- 1.<http://www.open-source.gr>
- 2.<http://httpd.apache.org>
- 3.<http://www.w3schools.com>
- 4.<http://www.php.net>
- 5.<http://www.mysql.com>