

**ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ ΛΑΡΙΣΑΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**



**ΤΜΗΜΑ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ  
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΜΕ ΘΕΜΑ:**

**ΥΛΟΠΟΙΗΣΗ ΔΙΑΔΙΚΤΥΑΚΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ SUPER  
MARKET (e-shop), ΜΕ ΧΡΗΣΗ ΤΩΝ ΤΕΧΝΟΛΟΓΙΩΝ JAVA (JSP,  
servlets, JDBC) & MYSQL**

**ΟΙ ΣΠΟΥΔΑΣΤΕΣ:**

**Αθανασίου Βάνα-Βαΐα T-1080**

**Αθανασίου Δημήτριος T-752**

**ΕΠΙΒΛΕΠΩΝ**

**Χρήστος Σωμαράς**

## ΛΑΡΙΣΑ 2010

# ΕΥΧΑΡΙΣΤΙΕΣ

Ξεκινώντας θα θέλαμε να ευχαριστήσουμε θερμά όλους εκείνους που συνέβαλαν στη δημιουργία, ολοκλήρωση και τελειοποίηση της πτυχιακής εργασίας. Ευχαριστούμε θερμά τον καθηγητή μας κο. Χρήστο Σωμαρά για την ουσιαστική καθοδήγησή του πάνω στο σκελετό που θα έπρεπε να στηριχτεί η πτυχιακή μας εργασία, όπως και τους υπόλοιπους καθηγητές αλλά και συμφοιτητές που μας υποστήριξαν και μας βοήθησαν καθ' όλα τα χρόνια των σπουδών μας.

Με εκτίμηση

Αθανασίου Βάνα - Βαΐα

Αθανασίου Δημήτριος

## ΑΦΙΕΡΩΜΕΝΗ

Στους γονείς μας, για την υπομονή και την πίστη που υπέδειξαν καθ' όλη τη διάρκεια της φοίτησής μας καθώς και κατά τη διάρκεια της προετοιμασίας και ολοκλήρωσης αυτής της εργασίας!

# ΠΕΡΙΕΧΟΜΕΝΑ

## ΠΡΟΛΟΓΟΣ

Η επιλογή του θέματος «ΥΛΟΠΟΙΗΣΗ ΔΙΑΔΙΚΤΥΑΚΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ SUPER MARKET (e-shop), ΜΕ ΧΡΗΣΗ ΤΩΝ ΤΕΧΝΟΛΟΓΙΩΝ JAVA (JSP, servlets, JDBC) & MYSQL» αποτέλεσε μεγάλη πρόκληση στο να γράψουμε μια μελέτη η οποία να αναφέρεται σε ζητήματα των ημερών, όπως οι online αγορές.

Πιστεύουμε ότι διαβάζοντας αυτή την μελέτη, ενημερώνεται ο αναγνώστης σε γενικά θέματα που αφορούν στο τρόπο λειτουργίας ενός ηλεκτρονικού καταστήματος. Επίσης, για το λόγο του ότι η εργασία αυτή ειδικεύεται σε ένα online Super Market, μπορούμε να δούμε μέχρι πια σημεία μπορεί να φτάσει η τεχνολογία των online αγορών, όπως για παράδειγμα η αποστολή των αγαθών που βρίσκουμε σε ένα Super Market, στο σπίτι μας.

Οι πληροφορίες που αποκομίζει ο αναγνώστης διαβάζοντας αυτή την εργασία είναι σημαντικές και μπορούν να τον βοηθήσουν στο να απαντήσει σε ερωτήματα και απορίες που τον απασχολούν και σχετίζονται με το συγκεκριμένο θέμα.

## ΕΙΣΑΓΩΓΗ

Η υλοποίησή της πτυχιακής μας εργασίας έγινε σταδιακά. Αρχικά εργαστήκαμε στο θεωρητικό κομμάτι, δηλαδή ποιο θα είναι το θέμα του καταστήματος και τι δυνατότητες θα έχει. Μετά από έρευνά μας αποφασίσαμε την ενσωμάτωση βασικών λειτουργιών που βρήκαμε σε σελίδες διάφορων ηλεκτρονικών καταστημάτων, εκτός από την χρήση mail server, κάτι που θα ξέφευγε από τους στόχους της παρούσας μελέτης. Η ανάλυση της θεωρητικής προσέγγισης της χρήσης των ηλεκτρονικών καταστημάτων αναγράφεται στο 1<sup>ο</sup> Κεφάλαιο. Κατά την διάρκεια των εργασιών παρουσιάστηκαν διάφορα προβλήματα τα οποία ξεπεράστηκαν ύστερα από πολύωρες αναζητήσεις στο Διαδίκτυο και στα βιβλία της βιβλιογραφίας μας. Όπως γίνεται κατανοητό λόγω του γεγονότος ότι η φύση της μελέτης είναι καθαρά πρακτική, η πλειοψηφία των προβλημάτων ξεπεράστηκε ύστερα από πολλές δοκιμές και τροποποιήσεις του κώδικα. Επομένως δεν είναι εφικτή η αναφορά μας σε συγκεκριμένα προβλήματα και σε τρόπους επίλυσης.

Στην πτυχιακή μας μελέτη χρησιμοποιήθηκαν οι τεχνολογίες JavaServlets, JavaServer Pages και μέσω του JDBC είχαμε επικοινωνία μεταξύ της εφαρμογής και της βάσης δεδομένων MySQL που κρατούσε πληροφορίες για τους χρήστες και τα προϊόντα. Η χρήση και ανάλυση αυτών των τεχνολογιών γίνεται στο 2<sup>ο</sup> Κεφάλαιο. Για την καλή εμφάνιση των περιεχομένων της εφαρμογής χρησιμοποιήθηκαν οι τεχνολογίες HTML, JavaScript, καθώς και τα προγράμματα Photoshop CS3 και LogoCreator 5.

Η ιστοσελίδα του ηλεκτρονικού μας καταστήματος εκτός από πληροφόρηση των επισκεπτών – πελατών, παρέχει υπηρεσίες εγγραφής χρήστη με δυνατότητα παραγγελίας προϊόντων επιλέγοντας συγκεκριμένη ημερομηνία και ώρα παράδοσης. Οι δυνατότητες του διαχειριστή είναι να εισάγει καθώς και να διαγράψει προϊόντα από τη βάση. Στα κεφάλαια που ακολουθούν δηλαδή 3<sup>ο</sup> και 4<sup>ο</sup> γίνεται η λεπτομερής ανάλυση της σχεδίασης και του κώδικα της εφαρμογής. Τέλος στο 5<sup>ο</sup> κεφάλαιο αναγράφονται τα συμπεράσματα του project.

# **ΚΕΦΑΛΑΙΟ 1<sup>ο</sup>: ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ**

## **ΚΑΤΑΣΤΗΜΑ**

Το "ηλεκτρονικό κατάστημα" είναι ένα σύνολο από ιστοσελίδες και άλλους μηχανισμούς στο διαδίκτυο, που κάνει την ίδια δουλειά με το παραδοσιακό κατάστημα: δηλαδή παρουσιάζει και πουλάει τα προϊόντα!

Το ηλεκτρονικό κατάστημα έχει κάποια σημαντικά πλεονεκτήματα σε σχέση με το παραδοσιακό κατάστημα: είναι ανοιχτό όλο το εικοσιτετράωρο, είναι διαθέσιμο σε όλη την Ελλάδα και τον κόσμο, δε χρειάζεται προσωπικό και έχει πολύ λιγότερα έξοδα.

### **1.1. Η Ανάπτυξη Διαδικτυακών Εφαρμογών Σήμερα**

Ανάπτυξη Διαδικτυακών Εφαρμογών (Web Development) είναι κάθε δραστηριότητα που σχετίζεται με την ανάπτυξη ενός ιστοχώρου (web site) ή με εφαρμογές (web application) για το παγκόσμιο ιστό (World Wide Web). Αυτή μπορεί να περιλαμβάνει εφαρμογές, όπως το e-Commerce (e-shop, e-banking, B2B, B2C), Συστήματα Διαχείρισης Περιεχομένου (Content Management System - CMS), Συστήματα Διαχείρισης Πελατειακών Επαφών (Customer Relation Management - CRM) ή ακόμα πιο εξειδικευμένες εφαρμογές όπως Online Booking Συστήματα για ξενοδοχειακές μονάδες, κρατήσεις εισιτηρίων κλπ.

Η έννοια της ανάπτυξης δικτυακών εφαρμογών (web development) είναι μια παρεξηγημένη έννοια. Οποιοσδήποτε δημιουργεί μια ιστοσελίδα "κατακτά" χωρίς στην ουσία να είναι και το τίτλο του web developer. Ακόμη, πολύ συχνά υπάρχει η σύγχυση των εννοιών του web design αλλά και του web development.

Η ανάπτυξη δικτυακών εφαρμογών είναι μια διαδικασία που εμπεριέχει το web design και καθορίζεται από τα στάδια του Σχεδιασμού δικτυακών εφαρμογών. Η λέξη και μόνο της ανάπτυξης υποδηλώνει τη χρήση γλώσσας προγραμματισμού για διαδικτυακές εφαρμογές όπως η PHP, Python, Java, ruby, perl. Δηλαδή, με τη χρήση μιας από τις προηγούμενες γλώσσες προγραμματισμού καθορίζονται οι λειτουργίες του δικτυακού

τόπου και ο τρόπος με τον οποίο ο web server θα συνθέτει και θα στέλνει στον web browser του χρήστη το αποτέλεσμα εμφάνισης του δικτυακού τόπου.

Η διαδικασία αυτής της σύνθεσης και εμφάνισης του περιεχομένου και του γραφιστικού κομματιού του δικτυακού τόπου για κάθε ιστοσελίδα του, γίνεται μέσα από τον κώδικα μιας από τις παραπάνω γλώσσες προγραμματισμού με τη συνεργασία της Βάσης Δεδομένων στην οποία υπάρχει το περιεχόμενο του δικτυακού τόπου και των λειτουργιών του web server (apache, tomcat, IIS), που υφίστανται για αυτό το σκοπό (λήψη αιτήματος, σύνθεση, αποστολή απάντησης στο χρήστη).

Επειδή, λοιπόν υπάρχει η παραπάνω εμπλοκή των γλωσσών προγραμματισμού αλλά και της Βάσης Δεδομένων του περιεχομένου, γι αυτό και πλέον μιλάμε για δυναμικές ιστοσελίδες οι οποίες ανάλογα του τι υπάρχει στη βάση δεδομένων μπορούν και ενημερώνονται κατά την ενημέρωση αυτής (ΒΔ). [16]

## 1.2. Το Ηλεκτρονικό Εμπόριο



Στο πρόσφατο παρελθόν οι συναλλαγές και οι αγορές των καταναλωτών και αντίστοιχα οι πωλήσεις των εμπόρων γίνονταν με καθαρά συμβατικά μέσα. Οι καταναλωτές προκειμένου να αγοράσουν αυτό που επιθυμούσαν ή να δεχτούν μία υπηρεσία έπρεπε να μεταβούν στην έδρα του προμηθευτή των αγαθών ή των υπηρεσιών. Στις μέρες μας ο τρόπος διεξαγωγής των συναλλαγών έχει αλλάξει ριζικά.

Ένας από τους νέους και τάχιστους τρόπους εξυπηρέτησης των καταναλωτών είναι το Ηλεκτρονικό Εμπόριο το οποίο αναπτύσσεται ραγδαία στο εξωτερικό αλλά και στην Ελλάδα (με πιο αργούς ρυθμούς).

Ενδεικτικό της καθυστερημένης ανάπτυξης του ηλεκτρονικού εμπορίου στην Ελλάδα είναι οι δύο υπουργικές αποφάσεις 3035/Β2-48.2001 και 7681/Β2-255.2001 που προωθούν τη διενέργεια δοκιμαστικής έρευνας για το ηλεκτρονικό εμπόριο. Οι αποφάσεις αυτές είναι του 2001, χρονιά που σε άλλες ευρωπαϊκές χώρες ανθούσε το ηλεκτρονικό εμπόριο. Αλλά και οι υπουργικές αποφάσεις 4708/2003, 36/2003 και 10220/Γ3-571/2004 που καταδεικνύουν το ίδιο πράγμα.[24]



Ως ηλεκτρονικό εμπόριο ορίζεται το εμπόριο που πραγματοποιείται με ηλεκτρονικά μέσα βασίζεται δηλαδή στην ηλεκτρονική μετάδοση δεδομένων. Το ηλεκτρονικό εμπόριο αποτελεί έκφραση των λεγόμενων υπηρεσιών εξ αποστάσεως (ΠΔ 39.2001).

Το ηλεκτρονικό εμπόριο αποτελεί μια ολοκληρωμένη συναλλαγή που πραγματοποιείται μέσω του διαδικτύου χωρίς να είναι απαραίτητη η φυσική παρουσία των συμβαλλομένων μερών, δηλαδή του πωλητή και του αγοραστή, οι οποίοι μπορούν να βρίσκονται ακόμα και σε διαφορετικές χώρες. Είναι οποιαδήποτε συναλλαγή που ενέχει διαδικτυακή δέσμευση για αγορά ή πώληση αγαθών ή υπηρεσιών. [13]

Ηλεκτρονικό εμπόριο θεωρούνται επίσης και οι συναλλαγές μέσω τηλεφώνου και φαξ. Το ηλεκτρονικό εμπόριο διακρίνεται σε έμμεσο και άμεσο. Ο πρώτος όρος χρησιμοποιείται όταν πρόκειται για την ηλεκτρονική παραγγελία υλικών αγαθών που μπορούν να παραδοθούν μόνο με παραδοσιακούς τρόπους όπως είναι το ταχυδρομείο.

Άμεσο είναι το ηλεκτρονικό εμπόριο που περιλαμβάνει παραγγελία, πληρωμή και παράδοση αυτών αγαθών και υπηρεσιών. Η πληρωμή των υπηρεσιών αυτών γίνεται είτε με πιστωτικές κάρτες είτε με ηλεκτρονικό χρήμα με την αρωγή πάντα και τη σύμπραξη των τραπεζών.

Ο χρόνος συνιστά εξαιρετικά πολύτιμο αγαθό στο ηλεκτρονικό εμπόριο. Η εξοικονόμηση του αγαθού αυτού (το Internet είναι από τη φύση του το ταχύτερο κανάλι συναλλαγών) αποτελεί για αρκετούς καταναλωτές τόσο σημαντική προτεραιότητα, που συχνά είναι διατεθειμένοι να πληρώσουν περισσότερα για ένα προϊόν με αντάλλαγμα την ταχύτητα διεκπεραίωσης και παράδοσής του. Εξ ου και η διατήρηση της τιμολογιακής διασποράς που είδαμε παραπάνω, ως ένα βαθμό τουλάχιστον. [13]

Επίσης, στοιχεία όπως προφίλ καταναλωτών, συνήθειες και προτιμήσεις, προσθέτουν αξία στις ηλεκτρονικές συναλλαγές, πέραν της βραχυπρόθεσμης/χρηματικής, καθώς συμβάλλουν στην παραγωγή εξατομικευμένων αγαθών, υπηρεσιών και διαφημίσεων από τους προμηθευτές. Κατά κάποιον τρόπο, οι καταναλωτές συμμετέχουν στη διαδικασία τόσο ως πελάτες όσο και ως "παραγωγοί" πληροφοριών που ενσωματώνονται στην προμήθεια αγαθών και υπηρεσιών. Στις θετικές συνέπειες μιας τέτοιας αλληλεπίδρασης εστιάζει και ο D. Quah στο *Weightless Economy in Economic Development* (1999). Την αρχική έμφαση στην αγοραστική μόνο διάσταση του ηλεκτρονικού εμπορίου διαδέχθηκε η ευρύτερη αναγνώριση της σημασίας των εφαρμογών

business-to-business. Γι' αυτό και η συνειδητοποίηση της δυναμικής επίδρασης των online τεχνολογιών στις επιχειρηματικές διαδικασίες "πίσω από τη βιτρίνα" δεν άργησε να επικρατήσει. Αλλά και από πλευράς τζίρου, οι συναλλαγές B2B αντιπροσωπεύουν μακράν το μεγαλύτερο κομμάτι της αγοράς του ηλεκτρονικού εμπορίου. [1]

Από τα μέσα της δεκαετίας του '60, η εφαρμογή των τεχνολογιών πληροφορικής και επικοινωνίας στη διοίκηση των επιχειρήσεων αναπτύσσεται συνεχώς, παράλληλα με την εξέλιξη της τεχνολογίας των ηλεκτρονικών υπολογιστών. Οι ΤΠΕ γίνονται ολοένα πιο απαραίτητες στην επιχείρηση και στη διοίκησή της, όσο η πολυπλοκότητα των διαδικασιών και των σχέσεων που τις διέπουν αυξάνεται. Η πολυπλοκότητα αυτή προέρχεται από τον ιστό των σχέσεων αλληλεξάρτησης και των οικονομικών δεσμών, στο πλαίσιο των οποίων λειτουργεί συνήθως μια τυπική επιχείρηση. Όταν οι τεχνολογίες του Internet συνδυάζονται με τις ΤΠΕ, το ηλεκτρονικό εμπόριο επιτρέπει στις επιχειρήσεις να "κάνουν περισσότερα με λιγότερα" ("do more with less"), μειώνοντας τα κόστη διαχείρισης των λειτουργικών διαδικασιών που πλαισιώνουν τις δραστηριότητές τους στην αλυσίδα αξίας. [1]

Ορίζοντας γενικά την έννοια «ηλεκτρονικό εμπόριο» μπορούμε να πούμε ότι πρόκειται για κάθε είδος εμπορικής συναλλαγής μεταξύ προσώπων (φυσικών και μη) που πραγματοποιείται με ηλεκτρονικά μέσα. Είναι η διάθεση και αγοραπωλησία προϊόντων ηλεκτρονικά, η διεκπεραίωση εμπορικών λειτουργιών και συναλλαγών χωρίς τη χρήση χαρτιού, συνήθως μέσω δικτύων ηλεκτρονικών υπολογιστών. Πρόκειται για την αγοραπωλησία αγαθών, πληροφοριών και υπηρεσιών μέσα από δίκτυα ηλεκτρονικών υπολογιστών. [1]

### Το ηλεκτρονικό εμπόριο μπορεί να οριστεί από τέσσερις διαφορετικές οπτικές γωνίες:

- **Επιχειρήσεις:** Ως εφαρμογή νέων τεχνολογιών προς την κατεύθυνση του αυτοματισμού των συναλλαγών και της ροής εργασιών.
- **Υπηρεσίες:** Ως μηχανισμός που έχει στόχο να ικανοποιήσει την κοινή επιθυμία προμηθευτών και πελατών για καλύτερη ποιότητα υπηρεσιών, μεγαλύτερη ταχύτητα εκτέλεσης συναλλαγών και μικρότερο κόστος.

- **Απόσταση:** Ως δυνατότητα αγοραπωλησίας προϊόντων και υπηρεσιών μέσω του Internet ανεξάρτητα από τη γεωγραφική απόσταση.
- **Επικοινωνία:** Ως δυνατότητα παροχής πληροφοριών, προϊόντων ή υπηρεσιών, και πληρωμών μέσα από δίκτυα ηλεκτρονικών υπολογιστών. [14]

**Το ηλεκτρονικό εμπόριο σε πρακτικό επίπεδο, μπορεί να πάρει πολλές μορφές:**

- **Εσωτερικό εμπόριο:** Στόχος είναι η αποτελεσματικότερη λειτουργία των δραστηριοτήτων μιας επιχείρησης, ώστε να μπορεί να προσφέρει καλύτερα προϊόντα και υπηρεσίες στους πελάτες της. Οι εφαρμογές του συνήθως εντάσσονται στη λειτουργία ενός τοπικού δικτύου (Intranet) και μπορούν να είναι: επικοινωνία μεταξύ ομάδων εργασίας, ηλεκτρονική δημοσίευση (άμεση διανομή πληροφοριών) κτλ.
- **Συναλλαγές μεταξύ επιχειρήσεων (Business-to-Business - B2B):** Το ηλεκτρονικό εμπόριο επιτρέπει σε επιχειρήσεις να βελτιώσουν τη μεταξύ τους συνεργασία, απλοποιώντας τις διαδικασίες και το κόστος των προμηθειών, την ταχύτερη αποστολή των προμηθειών και τον αποτελεσματικότερο έλεγχο του επιπέδου αποθεμάτων. Επιπλέον καθιστά ευκολότερη την αρχειοθέτηση των σχετικών εγγράφων και ποιοτικότερη την εξυπηρέτηση πελατών. Η δυνατότητα ηλεκτρονικής σύνδεσης με προμηθευτές και διανομείς καθώς και η πραγματοποίηση ηλεκτρονικών πληρωμών βελτιώνουν ακόμη περισσότερο την αποτελεσματικότητα: οι ηλεκτρονικές πληρωμές περιορίζουν το ανθρώπινο σφάλμα, αυξάνουν την ταχύτητα και μειώνουν το κόστος των συναλλαγών. Το ηλεκτρονικό εμπόριο προσφέρει τη δυνατότητα αυξημένης πληροφόρησης σχετικά με τα προσφερόμενα προϊόντα - είτε από τους προμηθευτές είτε από ενδιάμεσους οργανισμούς που προσφέρουν υπηρεσίες ηλεκτρονικού εμπορίου. [14]

## Λιανικές πωλήσεις - Ηλεκτρονικό εμπόριο μεταξύ επιχείρησης και καταναλωτών (Business-to-Consumer - B2C):

Πρόκειται για την πιο διαδεδομένη μορφή ηλεκτρονικού εμπορίου.

Ο καταναλωτής έχει πρόσβαση σε μια τεράστια ποικιλία προϊόντων σε δικτυακούς κόμβους-καταστήματα, βλέπει, επιλέγει, αν επιθυμεί να αγοράσει είδη ένδυσης μπορεί ενίοτε και να τα δοκιμάζει (μέσω ειδικών προγραμμάτων), ανακαλύπτει προϊόντα τα οποία δεν θα μπορούσε να βρει εύκολα στη χώρα του, συγκρίνει τιμές και τέλος αγοράζει. Κι όλα αυτά χωρίς να βγει από το σπίτι του, κερδίζοντας πολύτιμο χρόνο και κόπο. [16]

### Εμπορικές λειτουργίες

Προκειμένου ένας δικτυακός τόπος να χαρακτηριστεί ηλεκτρονική αγορά, θα πρέπει να παρέχει τουλάχιστον μία από τις παρακάτω εμπορικές λειτουργίες:

- Δημοπρασίες
- Αντίστροφες Δημοπρασίες
- Δυναμική τιμολόγηση
- Προτυποποίηση
- Ηλεκτρονική παρακολούθηση της ροής εργασιών
- Δημιουργία και διαχείριση καταλόγων
- Καταλόγους με δυνατότητα online παραγγελίας
- Ανταλλαγή πληροφορίας σε διάφορους κλάδους και αγαθά [16]

### Ποιο είναι το κέρδος μιας ηλεκτρονικής αγοράς:

Οι περισσότερες πλατφόρμες ηλεκτρονικού εμπορίου κερδίζουν μέσω διαφόρων χρεώσεων. Οι χρεώσεις συνήθως ποικίλλουν, ανάλογα με το ρόλο του χρήστη (αγοραστή ή προμηθευτή) και την έκταση της χρήσης. Τα έξοδα των συναλλαγών άλλοτε επιβαρύνουν μόνο τον αγοραστή ή τον πωλητή, και σε κάποιες περιπτώσεις και τους δύο. Οι κύριες πηγές εσόδων είναι:

- Συνδρομή μελών
- Προμήθεια επί της αξίας των συναλλαγών
- Σταθερή χρέωση για κάθε συναλλαγή που ολοκληρώνεται
- Φιλοξενία εταιρικών καταλόγων προϊόντων
- Διευρυμένη εταιρική προβολή, πέραν της βασικής πληροφορίας για τα προϊόντα
- Διαφήμιση (banners και σύνδεσμοι)
- Ενοικίαση ή πώληση λογισμικού ηλεκτρονικού εμπορίου
- Πρόσθετες υπηρεσίες, όπως logistics και οικονομικές υπηρεσίες
- Παροχή συμβουλευτικών υπηρεσιών

Αρχικά είχε εκτιμηθεί ότι το κύριο έσοδο των ηλεκτρονικών αγορών θα ήταν οι προμήθειες επί των συναλλαγών. Σήμερα οι περισσότερες κερδίζουν από τις συνδρομές των μελών τους.

Κάποιες πλατφόρμες δεν λειτουργούν τόσο ως κερδοσκοπικές εταιρίες, αλλά περισσότερο ως συνεργάτες για την ομαλή διεξαγωγή συναλλαγών. Οι ιδιοκτήτες των ηλεκτρονικών αυτών αγορών συναλλάσσονται οι ίδιοι στην πλατφόρμα και εύλογα επιθυμούν να διατηρήσουν το κόστος στα χαμηλότερα δυνατά επίπεδα.

Ωστόσο, από το 2001 και μετά πολλές ηλεκτρονικές αγορές τείνουν όλο και περισσότερο να μετεξελιχθούν σε παρόχους λύσεων και ειδικών εφαρμογών ηλεκτρονικού εμπορίου, στοχεύοντας κυρίως σε μεγάλες εταιρίες που σκοπεύουν να δημιουργήσουν τη δική τους "ιδιωτική" πλατφόρμα εμπορίου. [16]

### 1.3. Προδιαγραφές Ενός Ηλεκτρονικού Καταστήματος

Παρακάτω παρατίθενται τα βασικότερα χαρακτηριστικά που πρέπει να διαθέτει ένα ηλεκτρονικό κατάστημα και που υπάρχουν στο μέτρο του δυνατού και στην εφαρμογή που αναπτύχθηκε για αυτή την πτυχιακή εργασία, τόσο από την πλευρά του διαχειριστή όσο και από την πλευρά χρηστών.

- Διαχείριση αποκλειστικά μέσω Internet. Δεν χρειάζεται εγκατάσταση κάποιου επιπλέον προγράμματος στον υπολογιστή.
- Δυνατότητα προσθήκης, αλλαγής και διαγραφής των παραγγελιών σε άμεσο χρόνο.
- Μεγάλος αριθμός κατηγοριών.
- Δυνατότητα προσθήκης πολλαπλών εικόνων (μικρή και μεγάλη φωτογραφία) για κάθε προϊόν ξεχωριστά.
- Δυνατότητα περιήγησης ανά κατηγορία (Οπωροπωλείο, Κρεοπωλείο κτλ.).
- Αναλυτική προβολή όλων των στοιχείων της παραγγελίας.
- Προβολή στοιχείων λογαριασμού των μελών.
- Εύκολη διαχείριση μελών.
- On-line προσθήκη προϊόντων
- Εισαγωγή και αποσύνδεση από το σύστημα.[13]

## **1.4. Πλεονεκτήματα Ενός E-Shop (Ηλεκτρονικού Καταστήματος)**

### **1.4.1. Πλεονεκτήματα για την επιχείρηση**

Όσον αφορά την επιχείρηση το διαδίκτυο έχει συμβάλει σημαντικά στην ανάπτυξή της. Πρώτα από όλα μέσα από το διαδίκτυο η επιχείρηση έχει τη δυνατότητα διαφήμισής της με ελάχιστο κόστος. Παράλληλα το αγοραστικό κοινό στο οποίο απευθύνεται, ξεπερνάει τα σύνορα της χώρας με πρόσβαση ακόμα και σε απομακρυσμένες περιοχές.

Επίσης δίνει τη δυνατότητα για την ύπαρξη νέων προϊόντων και υπηρεσιών καθώς και την πραγματοποίηση τεχνολογικών επιτευγμάτων. Εξίσου σημαντική είναι η συμβολή του διαδικτύου, όσον αφορά την εύκολη διακίνηση των προϊόντων και τη μείωση στα λειτουργικά έξοδα της εταιρείας για την πραγματοποίηση των αγορών.

Παρέχεται η δυνατότητα για μια πληρέστερη ικανοποίηση των πελατειακών αναγκών, καθώς στοχεύονται συγκεκριμένες αγορές με την εφαρμογή των κατάλληλων αγοραστικών μεθόδων. Χαρακτηριστικό των διαδικτυακών αγορών είναι τις περισσότερες φορές η απουσία μεσαζόντων.

Τα πλεονεκτήματα για τις επιχειρήσεις, είναι η γεωγραφική επέκταση των δραστηριοτήτων τους, καθώς και η ελαχιστοποίηση του κόστους αυτής της επεκτατικής τους πολιτικής. Ακόμη σημαντικό είναι να αναφέρουμε τη δυνατότητα για γρήγορες αποφάσεις σε καταστάσεις αδιεξόδου, αλλά και την καλυτέρευση της εικόνας της εταιρείας, ως προς τους πελάτες της, τις νέες επιχειρηματικές συνεργασίες και επενδυτικά ανοίγματά της, που της δίνουν τη δυνατότητα να τα πραγματοποιήσει μέσω της διαδικτυακής της δραστηριοποίησης.

Τέλος γίνεται καλύτερος ποιοτικός έλεγχος των προϊόντων, αλλά και υποστηρίζονται όλες οι ενδοεπιχειρηματικές δραστηριότητες της εταιρείας, καθιστώντας το διαδίκτυο ως την καλύτερη δυνατή λύση για την e-επιχείρηση. Επίσης γίνεται ταχύτατη παραγγελία και προώθηση των αγαθών στο καταναλωτικό κοινό, σε χρόνο και σε σημεία που αυτό επιθυμεί.

Σύμφωνα με τον καθηγητή του Οικονομικού Πανεπιστημίου Αθηνών Γ. Δουκίδη (2006) για μια εταιρεία που δραστηριοποιείται διαδικτυακά υπάρχουν τρεις παράγοντες, που είναι συσχετισμένοι με διάφορα οφέλη.

Όσο μεγαλύτερη είναι μια εταιρεία, που δραστηριοποιείται διαδικτυακά, τόσο μεγαλύτερος θα είναι και ο αριθμός των παραγγελιών της, το ύψος των αποθεμάτων της, καθώς και η πιθανή συνεργασία της με αρκετούς πελάτες και προμηθευτές. Η χρονική διάρκεια της σε διαδικτυακές αγοραπωλησίες της παρέχουν αναγνωρισιμότητα και φήμη. [13, 16]

#### **1.4.2. Πλεονεκτήματα για τους πελάτες**

Όσον αφορά τους καταναλωτές μέσα από τις διαδικτυακές αγορές, έχουν τη δυνατότητα επιλογής περισσότερων αγορών και πραγματοποίηση αυτών σε συμφέρουσες τιμές. Επίσης τους δίνεται η δυνατότητα εύρεσης ποιοτικών προϊόντων και υπηρεσιών, αλλά και απόκτηση προϊόντων αποκλειστικής χρήσης. Η ενημέρωση για νέα προϊόντα είναι ευκολότερη, καθώς η παροχή πληροφοριών είναι αρκετή και κάθε χρήστης μπορεί να την χρησιμοποιήσει για προσωπικό του όφελος. Βρίσκουν, λοιπόν, φθηνά προϊόντα από ότι στις φυσικές αγορές και με ελάχιστο κόστος. Υπάρχει ακόμη και η παρουσία εναλλακτικών προϊόντων άλλα και η δυνατότητα εικοσιτετράωρων αγορών.

Τα πλεονεκτήματα του e-εμπορίου για τους καταναλωτές είναι εξίσου σπουδαία, διότι έχουν την δυνατότητα ποικιλίας ηλεκτρονικών αγορών παγκόσμιας εμβέλειας, καθώς και εξοικονόμησης χρόνου για την πραγματοποίηση των αγορών αυτών.

Ο χρήστης έχει την δυνατότητα να αξιοποιήσει την ύπαρξη προσφορών και εκπτώσεων στα προϊόντα. Στο ηλεκτρονικό εμπόριο υπάρχουν αρκετοί προμηθευτές για την καλύτερη εύρεση των επιθυμητών προϊόντων. Ακόμη ακολουθείται ανταγωνιστική πολιτική από τις επιχειρήσεις, γεγονός που οδηγεί στη μείωση τιμών, κάτι που είναι συμφέρον για τους χρήστες του διαδικτύου [13].

Πρέπει να σημειωθεί πως οι περισσότεροι από τους μισούς ηλεκτρονικούς αγοραστές χρησιμοποιούν την πιστωτική τους κάρτα για την διεκπεραίωση της συναλλαγής.



Πάντως από την άλλη πλευρά θετικό είναι, ότι, όσοι εκ των χρηστών έχουν κάνει ηλεκτρονικές αγορές, δηλώνουν κατά πλειοψηφία (94,3%) αρκετά ικανοποιημένοι.

Σύμφωνα με έρευνα της VPRC το έτος 2005, τα πλεονεκτήματα του ηλεκτρονικού εμπορίου που προσελκύουν τον Έλληνα καταναλωτή, είναι οι χαμηλότερες τιμές των προϊόντων (35,3%), οι αυξημένες δυνατότητες εξυπηρέτησης (28,5%), όπως η ευκολία επίσκεψης πολλαπλών ηλεκτρονικών καταστημάτων, η διατήρηση του ελέγχου της συναλλαγματικής διαδικασίας, η διεκπεραίωση ταχύτατων ηλεκτρονικών παραγγελιών, η πραγματοποίηση αγορών όλη την διάρκεια του 24ώρου καθώς και η ποσοτική ποικιλία και διαθεσιμότητα των αγαθών, σε ποσοστό (33,1%) (VPRC,2004).

### **1.4.3. Πλεονεκτήματα για την κοινωνία**

Δυνατότητα για περισσότερα άτομα να εργάζονται εξ αποστάσεως και να πραγματοποιούν αγορές χωρίς να μετακινούνται με συνακόλουθη μείωση της κυκλοφορίας και της ρύπανσης. Επίσης επιτυγχάνεται η μείωση των τιμών, η οποία ωφελεί τους σε δύσκολη οικονομική θέση πολίτες. Γίνεται εφικτή η πρόσβαση για τους κατοίκους απομακρυσμένων περιοχών σε νέα αγαθά & υπηρεσίες.

Παρατηρείται βελτιστοποίηση της ποιότητας των υπηρεσιών του δημόσιου και μείωση του κόστους αυτών, με ταυτόχρονη εξασφάλιση χρόνου και χρήματος για τις ίδιες τις υπηρεσίες και τους πολίτες της χώρας λειτουργώντας μέσα σε ένα περιβάλλον εκσυγχρονισμού. Επιτυγχάνεται έλεγχος της υπερκατανάλωσης, και διασφάλιση της οικονομικής πολιτικής, για την διεξαγωγή δημοσίων διαγωνισμών σε οργανισμούς, φορείς και επιχειρήσεις κοινής ωφέλειας. [13, 16]

## 1.5. Μειονεκτήματα Ενός E-Shop (Ηλεκτρονικού Καταστήματος)

Τα αδύνατα σημεία των δικτυακών αγορών παρουσιάζονται στο γεγονός ότι δεν υπάρχει ποιοτικός έλεγχος στην αποστολή αγαθών μικρού βάρους και αντοχής όπως, τρόφιμα και προϊόντα π.χ. (βιβλία CD, software, hardware κλπ.). Ακόλουθη είναι βέβαια και η αδυναμία πώλησης προϊόντων μεγάλου μεγέθους λόγω του βάρους και όγκου τους.

Εξίσου σημαντικό είναι και το υψηλό κόστος δραστηριοποίησης μιας επιχείρησης διαδικτυακά, λόγω υψηλού κόστους λειτουργίας και τεχνολογικών μέσων που χρησιμοποιεί, καθώς και η χρονοβόρα διαδικασία για αναγνώριση από το καταναλωτικό κοινό. Δεν παραλείπουμε και την δυσκολία λειτουργίας καταχώρησης, σε βάσεις δεδομένων, των στοιχείων των πελατών, καθώς και τις δυσκολίες στη διαχείριση αυτών των δεδομένων. Επίσης παράγοντας δυσκολίας αποτελεί και η καταβολή των απαραίτητων κεφαλαίων για την ενεργοποίηση της εταιρείας, τα όποια θα πρέπει να βρίσκονται σε άμεση διαθεσιμότητα . [13, 16]

Από την άλλη πλευρά (επιχειρηματικός κόσμος) μεγάλο μειονέκτημα για τις επιχειρήσεις που δραστηριοποιούνται στον ελλαδικό χώρο, είναι η μειωμένη ταχύτητα μεταφοράς δεδομένων πολλών megabytes, που αποτελούν μερικά από τα σημεία τα οποία καλείται να αντιμετωπίσει το Ιντερνέτ 2, το Ιντερνέτ του μέλλοντος.

Πολλά πανεπιστημιακά ιδρύματα στην Αμερική κάνουν χρήση αυτού του δικτύου, του νέου Ιντερνέτ 2. Το νέο ιντερνέτ χρησιμοποιεί μια μέθοδο που ενώνει ένα ενισχυμένο δίκτυο κόμβου με γειτονικά δίκτυα και έχει τη δυνατότητα μεταφοράς αρχείων έως και 10 Gigabyte/sec. Στόχος του νέου αυτού δικτύου είναι η ανάπτυξη ενός πανίσχυρου δικτύου, που θα παρέχει μεγάλες ταχύτητες σύνδεσης και μεταφοράς δεδομένων με νέες επαναστατικές μεθόδους, για τη χρήση της σύγχρονης τεχνολογίας και ποιοτικές διαδικτυακές υπηρεσίες, γεγονός που θα προσφέρει ικανοποίηση στους συνεχώς αυξανόμενους χρηστές του διαδικτύου.

Για τους ηλεκτρονικούς καταναλωτές σημαντικές είναι οι παγίδες από την αλόγιστη χρήση του, που μπορεί να προκαλέσει αύξηση των τηλεφωνικών λογαριασμών αλλά και ανασφάλεια στη χρήση των πιστωτικών καρτών. Το ηλεκτρονικό έγκλημα στο διαδίκτυο εξαπλώνεται χωρίς όρια, καθώς και οι αξιόποινες πράξεις, οι οποίες

καταδοκούν τα θύματα τους, όπως μικρά παιδιά και έφηβες, με την προβολή και διακίνηση ανήθικου οπτικοακουστικού υλικού, από άτομα υπεράνω υποψίας, που έχουν ως εργαλείο του εγκλήματος τους το διαδίκτυο. [24]

Πρωτεύοντα ρόλο παίζει και ο κίνδυνος επαφής και επικοινωνίας με άγνωστα άτομα, τα οποία δεν γνωρίζονται μεταξύ τους, ούτε ξέρουν κάποιο στοιχείο τις περισσότερες φορές από την προσωπικότητα, και τον χαρακτήρα τους, παρά μόνο λίγες προτάσεις γραμμένες από αυτούς, στην οθόνη του υπολογιστή τους από το άτομο με το οποίο επικοινωνούν, κατά την διάρκεια της διαδικτυακής συνομιλίας. Επίσης πολύ συχνή είναι η απειλή από κακόβουλους ιούς που προσβάλλουν τις αρχές του διαδικτύου, καθώς γίνονται παραβιάσεις στα προσωπικά δεδομένα των χρηστών και όχι μόνο. Δεν παύει να υφίσταται βέβαια και η κλοπή πνευματικών δικαιωμάτων προσωπικής ιδιοκτησίας, κάτι που τείνει να πλήξει καλλιτέχνες, δημιουργούς αλλά και δισκογραφικές εταιρείες.

Το σίγουρο είναι ότι κυριαρχεί ο αθέμιτος ανταγωνισμός, σε κάθε είδους διαδικτυακή δραστηριότητα.

Ένα ακόμη μεγάλο μειονέκτημα του διαδικτύου είναι και η ανάπτυξη του καταναλωτισμού, καθώς οι καταναλωτές δέχονται βομβαρδισμό πληροφοριών από υπηρεσίες και προϊόντα, τα οποία τις περισσότερες φορές δεν τους είναι χρήσιμα, με αποτέλεσμα να προβαίνουν σε άσκοπες αγορές. Αυτό που πρέπει να προέχει στο διαδίκτυο είναι η επιφύλαξη, η σύνεση και η σωστή χρήση του από κάθε χρήστη – καταναλωτή, χωρίς να απειλείται η προσωπική του περιούσια, τα πνευματικά του δικαιώματα αλλά και η ηθική του υπόσταση στο σύνολο της.[24]

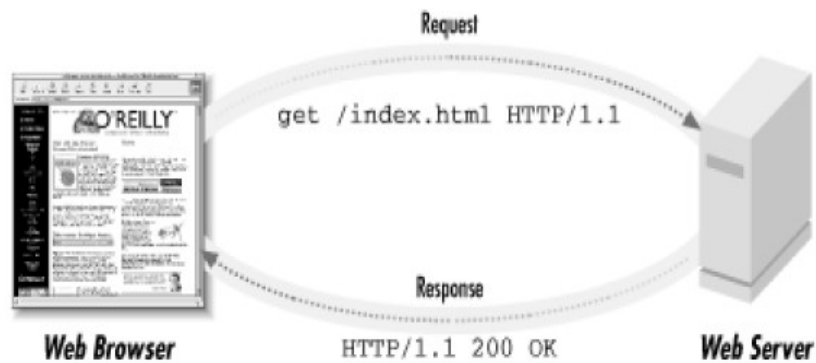
# ΚΕΦΑΛΑΙΟ 2<sup>ο</sup>: ΛΟΓΙΣΜΙΚΟ ΚΑΙ ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΔΥΝΑΜΙΚΩΝ ΙΣΤΟΣΕΛΙΔΩΝ

Αυτό το κεφάλαιο έχει ως σκοπό την αναφορά και την παρουσίαση όλων των προγραμμάτων που αποτελούν τα απαραίτητα εργαλεία για τη δημιουργία της εφαρμογής. Τα προγράμματα χωρίζονται σε δύο κατηγορίες, όπου κάποια από αυτά είναι κύρια, χρησιμοποιούνται ευρέως και πάνω σε αυτά βασίζεται ο διαδικτυακός τόπος, ενώ κάποια άλλα, τα βοηθητικά προγράμματα, δεν χρησιμοποιούνται σε μεγάλο βαθμό.

## **2.1. Tomcat Server**

Ο Tomcat είναι μια open-source εφαρμογή των προδιαγραφών των τεχνολογιών Java Servlets και JavaServer Pages, που αναπτύσσεται κάτω από το Jakarta project στο Apache Software Foundation. Ο Tomcat εφαρμόζει ένα νέο Servlet framework (ονομαζόμενο Catalina) που στηρίζεται επάνω σε μια εντελώς νέα αρχιτεκτονική με βάση τις Servlet 2.3 και JSP 1.2 προδιαγραφές. Περιλαμβάνει πολλές νέες δυνατότητες που τον ανάγουν σε μια χρήσιμη πλατφόρμα για την δημιουργία αλλά και την ανάπτυξη εφαρμογών αλλά και των υπηρεσιών web. Με λίγα λόγια, ο Tomcat είναι ένας διακομιστής εφαρμογών γραμμένος σε 100% καθαρή Java.

Ο Tomcat χρησιμοποιείται για πολλούς σκοπούς, και δεν περιορίζεται μόνο στη χρήση του ως διακομιστής εφαρμογών. Παρέχει μια ανοιχτή πλατφόρμα για την δημιουργία επεκτάσιμων υπηρεσιών web αλλά και υπηρεσιών content management. Η χρήση του Tomcat σε συνδυασμό με ένα βελτιστοποιημένο FreeBSD σύστημα, μπορεί να παρέχει πολύ συνεπείς αλλά και υψηλών ρυθμών εξυπηρέτησης υπηρεσίες. [4]



Εικόνα 2.1 - Αρχή Λειτουργίας ενός Web Server

## 2.2. Η Java

### 2.2.1. Εισαγωγή στη JAVA

Η Java είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου η οποία έχει προέλθει από την εταιρεία Sun Microsystems. Χρησιμοποιείται τόσο σε εφαρμογές διαδικτύου όσο και σε αυτόνομες εφαρμογές. Μερικά από τα βασικά της χαρακτηριστικά είναι η αντικειμενοστρέφεια, η επαναχρησιμοποίηση κώδικα και η φορητότητα. Αντικειμενοστρέφεια είναι μία προσέγγιση στην ανάπτυξη λογισμικού που οργανώνει τόσο το πρόβλημα όσο και τη λύση του ως μία συλλογή από διακριτά αντικείμενα. Η επαναχρησιμοποίηση κώδικα επιτυγχάνεται με τη χρησιμοποίηση-εισαγωγή ήδη υπάρχοντων προγραμμάτων των λεγόμενων κλάσεων οι οποίες επιτρέπουν στον προγραμματιστή να φτάσει γρήγορα στο επιθυμητό τελικό αποτέλεσμα. Η φορητότητα της Java έγκειται στην ανεξαρτησία λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα.

Άλλα ιδιαίτερα χαρακτηριστικά της Java είναι ο πολυνηματισμός, η ιδιαίτερη τεχνολογία δικτύου καθώς και το χαρακτηριστικό της ότι μπορεί να χρησιμοποιηθεί σε οποιοδήποτε μεγέθους εφαρμογή. Ένα άλλο δυνατό σημείο της Java είναι οι δυνατότητες που παρέχει για την δημιουργία ιστοσελίδων και μάλιστα δυναμικών, οι οποίες χρησιμοποιώντας το πρωτόκολλο μεταφοράς HTTP και την γλώσσα υπερκειμένου HTML προσφέρουν στους χρήστες του διαδικτύου πλείστες δυνατότητες. [10, 15]

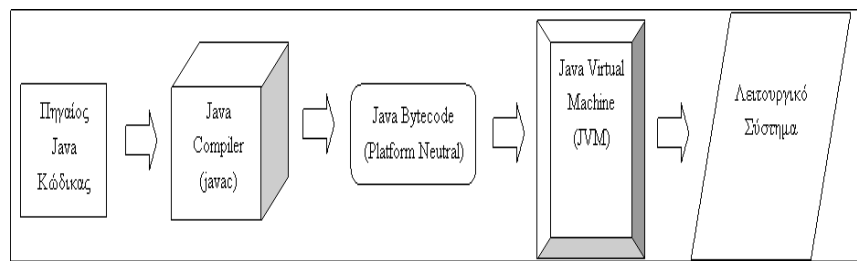
## 2.2.2. Πλεονεκτήματα της java

Τί κάνει όμως τη Java τόσο δημοφιλή; Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh (σύντομα θα τρέχουν και σε Playstation καθώς και σε άλλες παιχνιδιομηχανές) χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή ([Intel](#) x86, [IBM](#), Sun SPARC, [Motorola](#)) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας «κατανοεί» διαφορετικό assembly κώδικα. Ο συμβολικός (assembly) κώδικας που “τρέχει” σε Windows είναι διαφορετικός από αυτόν που “τρέχει” σε Macintosh. Η λύση δόθηκε με την ανάπτυξη της Εικονικής Μηχανής (Virtual Machine ή VM ή EM στα ελληνικά). [10, 15]

### 2.2.2.1. Η εικονική μηχανή της Java

Τι κάνει λοιπόν το Java Virtual Machine; Αφού γραφτεί κάποιο πρόγραμμα σε Java τότε μεταγλωττίζεται μέσω του Java compiler (javac), ο οποίος παράγει ένα αριθμό από .class αρχεία (=bytecode. Το bytecode είναι η μορφή που παίρνει ο πηγαίος κώδικας της Java όταν μεταγλωττιστεί) αναλόγως με το πόσες κλάσεις έχουν γραφτεί για την εφαρμογή. Όταν προσπαθήσουμε λοιπόν να εκτελέσουμε την εφαρμογή μας το Java Virtual Machine που πρέπει να είναι εγκατεστημένο στο μηχάνημά μας, θα αναλάβει να διαβάσει τα .class αρχεία και να τα μεταφράσει σε γλώσσα και εντολές (assembly) που υποστηρίζει το λειτουργικό μας και ο επεξεργαστής μας, έτσι ώστε να εκτελεστεί (να σημειώσουμε εδώ ότι αυτό συμβαίνει με την παραδοσιακή Εικονική Μηχανή (Εικόνα 2.2)). Πιο σύγχρονες εφαρμογές της EM μπορούν και αποδελτιώνουν τα πολύχρηστα τμήματα bytecode απ' ευθείας σε εγγενή κώδικα (native code) με αποτέλεσμα να βελτιώνεται η ταχύτητα. Χωρίς αυτό δε θα ήταν δυνατή η εκτέλεση λογισμικού γραμμένο σε Java. Πρέπει να πούμε ότι η Virtual Machine είναι λογισμικό εξαρτημένο από τη πλατφόρμα, δηλαδή για κάθε είδος λειτουργικού συστήματος και αρχιτεκτονικής επεξεργαστή, υπάρχει

διαφορετική έκδοση! Έτσι υπάρχουν διαφορετικές JVM για Windows, Linux, Unix, Macintosh, κινητά τηλέφωνα, παιχνιδιομηχανές κλπ!



**Εικόνα 2.2.** - Παραδοσιακή εικονική μηχανή java

Οτιδήποτε θέλει να κάνει ο προγραμματιστής (ή ο χρήστης) γίνεται μέσω της εικονικής μηχανής. Αυτό βοηθάει στο να υπάρχει μεγαλύτερη ασφάλεια στο σύστημα γιατί η εικονική μηχανή είναι υπεύθυνη για την επικοινωνία χρήστη - υπολογιστή. Ο προγραμματιστής δεν μπορεί να γράψει κώδικα ο οποίος θα έχει καταστροφικά αποτελέσματα για τον υπολογιστή γιατί η εικονική μηχανή θα τον ανιχνεύσει και δε θα επιτρέψει να εκτελεστεί. Από την άλλη μεριά ούτε ο χρήστης μπορεί να κατεβάσει «κακό» κώδικα από το δίκτυο και να τον εκτελέσει. Αυτό είναι ιδιαίτερα χρήσιμο για μεγάλα καταναμημένα συστήματα όπου πολλοί χρήστες χρησιμοποιούνε το ίδιο πρόγραμμα συγχρόνως. [10, 15]

#### **2.2.2.2. Ο συλλέκτης απορριμμάτων (Garbage Collector)**

Ακόμα μία ιδέα που βρίσκεται πίσω από τη Java είναι η ύπαρξη του συλλέκτη απορριμμάτων (Garbage Collector). Η σύλλεξη απορριμμάτων είναι μία κοινή ονομασία που χρησιμοποιείται στον τομέα της πληροφορικής για να δηλώσει την ελευθέρωση τμημάτων μνήμης από δεδομένα που δε χρειάζονται και δε χρησιμοποιούνται άλλο. Αυτή η απελευθέρωση μνήμης στη Java είναι αυτόματη και γίνεται μέσω του συλλέκτη απορριμμάτων. Υπεύθυνη για αυτό είναι και πάλι η εικονική μηχανή η οποία μόλις «καταλάβει» ότι ο σωρός (heap) της μνήμης (στη Java η συντριπτική πλειοψηφία των αντικειμένων αποθηκεύονται στο σωρό σε αντίθεση με τη C++ που αποθηκεύονται κυρίως σε στοίβα - stack) κοντεύει να γεμίσει, ενεργοποιεί το συλλέκτη απορριμμάτων. Έτσι ο προγραμματιστής δε χρειάζεται να ανησυχεί για το πότε και αν θα ελευθερώσει ένα συγκεκριμένο τμήμα της μνήμης, ούτε και για δείκτες (pointers) που εστιάζουν σε άδεια

θέση μνήμης. Αυτό είναι ιδιαίτερα σημαντικό αν σκεφτούμε ότι ένα μεγάλο ποσοστό κατάρρευσης των προγραμμάτων οφείλονται σε λάθος χειρισμό της μνήμης. [10, 15]

## **2.3. Jsp, Servlets Και Κλάσεις**

### **2.3.1. Servlets**

Οι ιστοσελίδες που περιέχουν κώδικα JSP (JavaServer Pages), δηλ. ανάμιξη κώδικα HTML με κώδικα Java, μετατρέπονται (μεταφράζονται ή μεταγλωττίζονται) σε Servlets πριν εκτελεστούν στον server. Η κατανόηση αυτής της μετάφρασης (απόδοσης) σε κώδικα Servlet θα μας βοηθήσει να καταλάβουμε καλύτερα τις δραστηριότητες των JSPs στο παρασκήνιο. Ένα Servlet, στη γενική του μορφή, είναι μια τάξη (class) της Java που υλοποιεί (implements) το interface Servlet και δέχεται αιτήσεις (requests) και παράγει (δημιουργεί) αποκρίσεις (responses). Οι αιτήσεις μπορεί να προέρχονται από τάξεις της Java, από Web clients ή και από άλλα Servlets.

Όταν υλοποιούμε ένα interface λέμε ότι η τάξη μάς παρέχει υλοποιήσεις για τις μεθόδους που είναι δηλωμένες στο interface. Συνεπώς, όταν υλοποιούμε το interface Servlet δηλώνουμε ότι ο κώδικάς μας θα παρέχει υλοποιήσεις για τις μεθόδους που βρίσκονται στο interface Servlet. Θα ασχοληθούμε μ' ένα μόνο συγκεκριμένο είδος Servlet, το HttpServlet, το οποίο δέχεται HTTP requests και παράγει HTTP responses. Όταν γράφουμε το δικό μας HttpServlet, δεν υλοποιούμε το interface Servlet απευθείας, αλλά επεκτείνουμε (extend) την τάξη HttpServlet. [10, 7, 15]

### **2.3.2. Σελίδες JSP**

Μία συμπληρωματική τεχνολογία για την κατασκευή διαδικτυακών εφαρμογών με την Java είναι οι Σελίδες Εξυπηρετητή Java (Java Server Pages ή JSP). Οι JSP επιτρέπουν την εύκολη μίξη κώδικα HTML και κώδικα Java στην ίδια σελίδα.

Όπως είδαμε με τα Java Servlets ο χρήστης θα πρέπει να γράψει αρκετές εντολές `out.print` για να σχηματίσει την απόκριση του servlet στον χρήστη (δηλ. το αποτέλεσμα). Αυτό είναι κουραστικό όταν η απόκριση είναι πολύπλοκη (π.χ. έχει πίνακες ή άλλα σύνθετα HTML στοιχεία). Είναι πιο βολικό η απόκριση να σχηματίζεται απευθείας γράφοντας HTML και όπου χρειάζεται να χρησιμοποιείται κώδικας Java. [9, 12, 15]



Αυτή ακριβώς την ανάγκη έρχονται να καλύψουν οι JSP σελίδες.

### **2.3.3. Μετατροπή των JSPs σε Servlets**

Αφού τα JSPs (ιστοσελίδες με JSP κώδικα) μετατρέπονται σε Servlets, τότε θα μπορεί να ρωτήσει κάποιος γιατί να πρέπει να μάθουμε και να χρησιμοποιούμε και τα δύο; Ο βασικότερος λόγος για το γιατί πρέπει να χρησιμοποιούμε τα JSPs είναι ότι μπορούμε να γράψουμε ευκολότερα τον κώδικα, ενώ αν ασχοληθούμε με το γράψιμο του κώδικα σε Servlet, θα πρέπει να μάθουμε να δημιουργούμε τάξεις (classes) και κληρονομικότητα (inheritance) σε γλώσσα Java, κάτι όχι ιδιαίτερα επιθυμητό από τους περισσότερους.

Θα πρέπει επίσης να μάθουμε πώς είναι ένας κώδικας γραμμένος σε Servlet, για να γνωρίζουμε τι βρίσκεται πίσω από έναν κώδικα που είναι γραμμένος σε JSP.

Οι μέθοδοι που χρησιμοποιούνται στον κύκλο ζωής ενός JSP ή ενός Servlet είναι οι εξής :

- `new()`, για τη δημιουργία ενός JSP ή Servlet.
- `init()`, για την απόδοση αρχικών τιμών (αρχικοποίηση).
- `service()` ή `doGet()` ή `doPost()` για τον χειρισμό των αιτήσεων των clients.
- `destroy()`, για τη διαγραφή ενός JSP ή Servlet.
- `finalize()`, για τον καθαρισμό της μνήμης από τα διαγραμμένα JSPs ή Servlets.

Όσον αφορά τώρα τη διαδικασία μετατροπής (μετάφρασης), το JSP μεταφράζεται (translated) στον αντίστοιχο κώδικα Servlet, το οποίο είναι ένα αρχείο με επέκταση `.java`. Αυτό μετά μεταγλωττίζεται (compiled) στο αντίστοιχο bytecode αρχείο με επέκταση `.class`. Το αρχείο `.class` εκτελείται, δημιουργείται η έξοδος σε μορφή HTML εγγράφου και στέλνεται πίσω στον client. Μετά από την πρώτη κλήση ενός JSP, τα επόμενα requests δεν θα ακολουθήσουν όλη τη φάση της μετάφρασης αλλά θα πάνε κατευθείαν στο ήδη μεταγλωττισμένο αρχείο `.class`. [9, 12, 15]

### **2.3.3.1. Η Βασική Δομή ενός HttpServlet**

Υπάρχουν πολλές μέθοδοι που υπερκαλύπτουμε (override) όταν γράφουμε ένα HttpServlet, όπως :

- void init(ServletConfig sc) throws ServletException;
- void Service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException;
- void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException;
- void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException;

Η μέθοδος init() καλείται για να αποδώσει αρχικές τιμές σε ορισμένες παραμέτρους. Η μέθοδος Service() χρησιμοποιείται για να ελέγξουμε τις καταχωρήσεις του χρήστη όταν δεν μας ενδιαφέρει αν η αίτησή (request) του προήλθε από μια μέθοδο GET ή μια μέθοδο POST. Αν, όμως, έχει σημασία το αν η HTML φόρμα στάλθηκε με μια μέθοδο GET ή μια μέθοδο POST, τότε θα πρέπει να υπερκαλύψουμε μια από τις μεθόδους doGet() ή doPost().

Μια HTML φόρμα διαθέτει την ιδιότητα (attribute) METHOD η οποία ορίζει τον τρόπο με τον οποίο θα σταλούν τα δεδομένα στον server. Η μέθοδος GET προσαρτά τα δεδομένα στο URL και τα στέλνει στον server μ' αυτόν τον τρόπο, ενώ η μέθοδος POST συσκευάζει τα δεδομένα σ' ένα πακέτο και στέλνει το πακέτο στον server. [7, 10, 15]

### **2.3.3.2. Ένα Παράδειγμα Μετατροπής JSP σε Servlet**

Θα δούμε τώρα ένα παράδειγμα κώδικα σε JSP και το ισοδύναμό του σε κώδικα Servlet. Ο κώδικας Servlet παράγει την ίδια ακριβώς έξοδο με το JSP, αλλά με πολύ περισσότερη δουλειά. Ο κώδικας JSP που θα δούμε δημιουργεί έναν πίνακα δύο στηλών που περιέχει στην πρώτη στήλη τους αριθμούς 1 έως 10 και στη δεύτερη στήλη το κείμενο Άρτιος ή Περιττός, ανάλογα.

### EvenOddNumbers.jsp

```
<html>
  <head>
    <title> Οι Άρτιοι και οι Περιττοί Αριθμοί </title>
  </head>
  <body>
    <center>
      <h2> Οι Άρτιοι και οι Περιττοί Αριθμοί </h2>
    </center>
    <table border="2" align="center">
      <th>Αριθμός</th><th>Άρτιος/Περιττός</th>
      <% for (int i=0; i<10; i++) { %>
        <tr><td><%=i%></td>
          <% if(Math.IEEEremainder(i, 2)==0) { %>
            <td>Ο Αριθμός είναι Άρτιος</td>
          </tr>
          <% } else { %>
            <td>Ο Αριθμός είναι Περιττός</td>
          </tr>
          <% } // end if
        } // end for %>
      </table>
    </body>
```

</html>

Η έξοδος της παραπάνω σελίδας θα είναι ένας πίνακας με δύο στήλες, όπου στη μια στήλη θα είναι ένας από τους ακέραιους αριθμούς 1-10 και στη δεύτερη στήλη το κείμενο Άρτιος ή Περιττός, ανάλογα. Ο πίνακας θα έχει περίγραμμα (border) ίσο με 2 pixels και θα είναι κεντραρισμένος στη σελίδα. Με τα tags <th> δημιουργούμε τις επικεφαλίδες του πίνακα.

Το σύμβολο <%, που κλείνει με το σύμβολο %>, αποκαλείται scriptlet tag και χρησιμοποιείται για να προσθέσουμε κώδικα Java μέσα σε HTML κώδικα. Το tag <%= αποκαλείται expression tag και μας επιτρέπει να εισάγουμε μια έκφραση της Java η οποία θα εκτιμηθεί (υπολογισθεί) και το αποτέλεσμα της θα εισαχθεί στον JSP κώδικα σαν κανονικό κείμενο.

Θα δούμε τώρα τον ισοδύναμο κώδικα Servlet, ο οποίος θα παράγει το ίδιο ακριβώς αποτέλεσμα. Αυτό σημαίνει ότι ο παραπάνω κώδικας JSP θα μεταφρασθεί στον επόμενο κώδικα Servlet.

```
import java.io.*;
```

```
import javax.servlet.*;
```

```
import javax.servlet.http.*;
```

```
public class EvenOddNumbers extends HttpServlet
```

```
{
```

```
    public void Service(HttpServletRequest request,
```

```
                        HttpServletResponse response)
```

```
        throws IOException, ServletException
```

```
{
```

```
    response.setContentType("text/html");
```

```
    ServletOutputStream out=response.getOutputStream();
```

```
    out.print("<html>");
```

```
    out.print("<head><title>Οι Άρτιοι και οι Περιττοί Αριθμοί
```

```

        </title></head>");

    out.print("<body>");

    out.print("<center>");

    out.print("<h2>Οι Άρτιοι και οι Περιττοί Αριθμοί</h2>");

    out.print("</center>");

    out.print("<table border='2' align='center'>");

    out.print("<th>Αριθμός</th><th>Άρτιος/Περιττός</th>");

    for (int i=0; i<10; i++) {

        out.print("<tr><td>" + i + "</td>");

        if(Math.IEEEremainder(i, 2)==0) {

            out.print("<td>Ο Αριθμός είναι Άρτιος</td></tr>");

        }

        else {

            out.print("<td>Ο Αριθμός είναι Περιττός</td></tr>");

        } // τέλος εντολής if-else

    } // τέλος βρόχου for

    out.print("</table></body></html>");

    out.close();

}

}

```

Βλέπουμε ότι αν και το αποτέλεσμα είναι το ίδιο, απαιτείται πολύ περισσότερη δουλειά για να γράψουμε κώδικα για Servlets και επίσης πολύ καλή γνώση της Java, κάτι που δεν είναι απαραίτητο αν θέλουμε να γράψουμε σε JSP κώδικα

### 2.3.4. Κλάσεις

Η κλάση είναι μία γενική φόρμα για την κατασκευή αντικειμένων που έχουν παρόμοια χαρακτηριστικά. Αφού ορίσουμε τη κλάση, μετά μπορούμε να χρησιμοποιήσουμε τον ορισμό της για να δημιουργήσουμε αντικείμενα. Κάθε αντικείμενο λέγεται και στιγμιότυπο της κλάσης. Στη Java υπάρχουν έτοιμες κλάσεις, αλλά μπορεί ο χρήστης να δημιουργήσει και τις δικές του κλάσεις. Δύο είναι τα στοιχεία που περιλαμβάνονται στο ορισμό μιας κλάσης.

A) Οι μεταβλητές (variables), οι οποίες διαφοροποιούν το ένα αντικείμενο της κλάσης από το άλλο και

B) Οι μέθοδοι (methods), οι οποίες καθορίζουν τι μπορούμε να κάνουμε στα αντικείμενα μιας κλάσης.

Για να ορίσουμε μια κλάση, χρησιμοποιούμε τη λέξη class ακολουθούμενη από το όνομα που επιλέγουμε για τη κλάση. Πχ :

*Class circle*

```
{  
  
}
```

Την αποθηκεύουμε με το όνομα circle.java και όταν τη μεταγλωττίσουμε δημιουργείται το αρχείο circle.class. Ακολουθεί ένα παράδειγμα για να κατανοήσουμε το ορισμό της κλάσης :

*Class circle*

```
{
```

```
    Static double pi = 3.14;
```

```
    Static int count = 0;
```

```
    Double radius;
```

```
    Double x;
```

```
    Double y;
```

}

Οι μεταβλητές της κλάσης ανήκουν σε δύο κατηγορίες :

- Οι μεταβλητές `pi` και `count` είναι μεταβλητές κλάσης και δηλώνονται με τη λέξη `static` . Μια μεταβλητή κλάσης υπάρχει έστω και εάν δεν έχει δημιουργηθεί ακόμη κανένα αντικείμενο. Όταν το αντικείμενο δημιουργηθεί, τότε περιέχει τη τιμή του `pi` και της `count`. Εάν οι τιμές αυτές αλλάξουν, τότε οι νέες τιμές τοποθετούνται σε όλα τα αντικείμενα της κλάσης.
- Οι μεταβλητές `radius`, `x`, `y` λέγονται μεταβλητές στιγμιότυπου. Σε κάθε αντικείμενο της κλάσης που δημιουργείται, οι μεταβλητές αυτές υπάρχουν, αλλά έχουν διαφορετική τιμή από αντικείμενο σε αντικείμενο.

Ανάλογα με τις μεταβλητές, υπάρχουν και δύο κατηγορίες μεθόδων, οι μέθοδοι κλάσης και οι μέθοδοι στιγμιότυπου.

- A) Οι μέθοδοι κλάσης δηλώνονται με τη λέξη `static` και μπορεί να εκτελεστούν ακόμη και αν δεν έχει δημιουργηθεί κανένα αντικείμενο της κλάσης.
- B) Οι μέθοδοι στιγμιότυπου για να εκτελεστούν πρέπει να υπάρχουν αντικείμενα. Αν δεν υπάρχει αντικείμενο, δεν έχει νόημα να εκτελεστεί μία τέτοια μέθοδος.

**Η μεταβλητή `this`** : Κάθε μέθοδος στιγμιότυπου διαθέτει μία αναφορά με το όνομα `this`, η οποία αναφέρεται στο τρέχον αντικείμενο, για το οποίο έχει κληθεί η μέθοδος. Κάθε φορά στην μνήμη υπάρχει μόνο ένα αντίγραφο της μεθόδου. Χρησιμοποιώντας λοιπόν την λέξη `this`, μπορούμε να κάνουμε την μέθοδο να εργάζεται κάθε φορά για διαφορετικό αντικείμενο. Επειδή το `this` αναφέρεται στο τρέχον αντικείμενο της κλάσης, χρησιμοποιείται μόνο για τις μεθόδους στιγμιότυπου και όχι για τις μεθόδους κλάσης.

### 2.3.5. Java Scripts

Η JavaScript είναι γλώσσα προγραμματισμού η οποία έχει σαν σκοπό την παραγωγή δυναμικού περιεχομένου σε ιστοσελίδες. Έχει τις ρίζες της στην ECMAScript της οποίας ουσιαστικά αποτελεί επέκταση με μερικές πρόσθετες δυνατότητες.

#### Τρόπος σύνταξης

Όπως και η PHP, η Javascript έχει βασιστεί όσον αφορά τον τρόπο σύνταξης του κώδικά της στη γλώσσα προγραμματισμού C, με την οποία παρουσιάζει πολλές ομοιότητες. Όμως ενώ η PHP είναι μια server side γλώσσα προγραμματισμού, η Javascript είναι client side.

Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στον server, αλλά στο πρόγραμμα περιήγησης των επισκεπτών. Αυτή η διαφορά έχει και πλεονεκτήματα και μειονεκτήματα για καθεμιά από τις δύο γλώσσες. Συγκεκριμένα, η Javascript δεν έχει καμία απαίτηση από πλευράς δυνατοτήτων του server για να εκτελεστεί (επεξεργαστική ισχύ, συμβατό λογισμικό διακομιστή), αλλά βασίζεται στις δυνατότητες του browser των επισκεπτών. Επίσης μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Παρόλα αυτά, οι δυνατότητές της είναι σημαντικά μικρότερες από αυτές της PHP και δεν παρέχει συνδεσιμότητα με βάσεις δεδομένων.]

#### Javascript και Java

Η Javascript δεν θα πρέπει να συγχέεται με τη Java, που είναι διαφορετική γλώσσα προγραμματισμού και με διαφορετικές εφαρμογές. Τονίζεται ότι ο σωστός τρόπος γραφής της είναι "Javascript" και όχι 'Java script' σαν δύο λέξεις, όπως λανθασμένα γράφεται ορισμένες φορές. ]



## 2.4. Η MySQL

### 2.4.1. Εισαγωγή στη MySQL

#### **Η MySQL είναι ένα σύστημα διαχείρισης βάσης δεδομένων.**

Μια βάση δεδομένων είναι μια δομημένη συλλογή των δεδομένων. Μπορεί να είναι οτιδήποτε, από μια απλή λίστα με ψώνια σε super market ή τα τεράστια ποσά των πληροφοριών σε ένα εταιρικό δίκτυο. Για να προστεθεί η πρόσβαση και η επεξεργασία των δεδομένων σε μία ηλεκτρονική βάση δεδομένων, θα χρειαστεί ένα σύστημα διαχείρισης βάσεων δεδομένων όπως MySQL Server. Δεδομένου ότι οι ηλεκτρονικοί υπολογιστές είναι πολύ καλοί στο χειρισμό μεγάλου όγκου δεδομένων, τα συστήματα διαχείρισης βάσεων δεδομένων διαδραματίζουν κεντρικό ρόλο στην πληροφορική, ως standalone κοινής ωφελείας, ή και τμήματα των άλλων εφαρμογών. ]

#### **Η MySQL είναι ένα σχεσιακό σύστημα διαχείρισης βάσης δεδομένων.**

Μια σχεσιακή βάση δεδομένων αποθηκεύει δεδομένα σε χωριστούς πίνακες αντί να θέτει όλα τα δεδομένα σε μια μεγάλη αποθήκη. Αυτό προσθέτει ταχύτητα και ευελιξία. Το μέρος SQL του "MySQL" αντιπροσωπεύει "Structured Query Language". Η SQL είναι η πιο κοινή τυποποιημένη γλώσσα που χρησιμοποιείται για πρόσβαση σε βάσεις δεδομένων και ορίζεται από την ANSI / ISO SQL Πρότυπο. Το πρότυπο SQL έχει εξελιχθεί από το 1986 και υπάρχουν διάφορες εκδόσεις.]

Προς το παρόν ο MySQL Administrator και ο MySQL Query Browser είναι τα μόνα προγράμματα της σουίτας των γραφικών εργαλείων της MySQL. ]

## 2.4.2. MySQL Administrator

Ο MySQL Administrator είναι ένα πρόγραμμα για την εκτέλεση των διοικητικών πράξεων, όπως τη ρύθμιση, την παρακολούθηση, την έναρξη και την παύση ενός διακομιστή MySQL, τη διαχείριση των χρηστών και των συνδέσεων, την εκτέλεση αντιγράφων ασφαλείας, καθώς και διάφορα άλλα διοικητικά καθήκοντα.

Οι περισσότερες εργασίες μπορούν να εκτελεστούν χρησιμοποιώντας μια εντολή πελάτη της γραμμής, όπως `mysqladmin`, ή `mysql`. Ο MySQL Administrator έχει τα εξής πλεονεκτήματα:

- Γραφική διεπαφή με τον χρήστη που τον καθιστά πιο διαισθητικό.
- Παρέχει μια καλύτερη εικόνα από τις ρυθμίσεις που είναι ζωτικής σημασίας για τις επιδόσεις, την αξιοπιστία και την ασφάλεια του διακομιστή MySQL.
- Εμφανίζει τους δείκτες απόδοσης γραφικά, καθιστώντας έτσι πιο εύκολο να προσδιοριστεί και να συντονιστούν οι ρυθμίσεις του διακομιστή.

Ο MySQL Administrator Browser είναι σχεδιασμένος, ώστε να συνεργάζεται με τις εκδόσεις 4.0 και άνω της MySQL. ]

## 2.4.3. MySQL Query Browser

Ο MySQL Query Browser είναι ένα γραφικό εργαλείο, το οποίο παρέχεται από την MySQL AB, για την δημιουργία, εκτέλεση και βελτιστοποίηση αιτημάτων σε γραφικό περιβάλλον. Ενώ ο MySQL Administrator έχει σχεδιαστεί για την διαχείριση ενός MySQL εξυπηρετητή, ο MySQL Query Browser έχει σχεδιαστεί για να βοηθήσει να θέτονται αιτήματα και να αναλύονται δεδομένα, τα οποία είναι αποθηκευμένα στην MySQL βάση δεδομένων.

Ενώ όλα τα αιτήματα, που μπορούν να εκτελεστούν με τον MySQL Query Browser, μπορούν επίσης να εκτελεστούν με την εφαρμογή γραμμής εντολών `mysql`, ο MySQL Query Browser επιτρέπει την υποβολή αιτημάτων και την επεξεργασία των δεδομένων με ένα πιο διαισθητικό, γραφικό τρόπο.

Ο MySQL Query Browser είναι σχεδιασμένος, ώστε να συνεργάζεται με τις εκδόσεις 4.0 και άνω της MySQL.

Ο MySQL Query Browser είναι σε μεγάλο βαθμό, το αποτέλεσμα απόψεων και ανάδρασης, τα οποία έχει δεχθεί η MySQL AB, από πολλούς χρήστες σε μια περίοδο πολλών ετών. ]

#### **2.4.4. Πλεονεκτήματα της MySQL**

Στην πραγματικότητα, είναι δύσκολο να πει κανείς για πιο λόγο προτιμάται το ένα ή το άλλο σύστημα βάσης δεδομένων. Παρόλα αυτά, κάποιες μικρές λεπτομέρειες και διαφορές, καθορίζουν τελικά ποιο θα χρησιμοποιηθεί για την ανάπτυξη web εφαρμογών. Σε τελικό στάδιο, η απόφαση για το ποιο λογισμικό θα χρησιμοποιηθεί εξαρτάται από τις απαιτήσεις του έργου που πρόκειται να υλοποιηθεί.

Η απάντηση στο παραπάνω ερώτημα είναι αδύνατον να δοθεί μονολεκτικά ή μέσα σε μια φράση. Υπάρχουν πολλοί λόγοι που συγκλίνουν στην άποψη υπέρ της MySQL, μερικοί από τους οποίους είναι και οι εξής :

- **Αξιοπιστία και επιδόσεις.** Η MySQL διαθέτει τις πιο ανταγωνιστικές και αξιόπιστες επιδόσεις, λόγο του ότι διαθέτει λογισμικό ανοικτού πηγαίου κώδικα.
- **Ευκολία χρήσης και ανάπτυξης.** Η MySQL αρχιτεκτονική καθιστά εξαιρετικά γρήγορα αποτελέσματα και είναι ευπροσάρμοστη. Η μοναδική πολλαπλής αποθήκευσης αρχιτεκτονική της MySQL δίνει την ευελιξία που χρειάζονται με ένα DBMS σύστημα ακόμα και σε μεγάλες εταιρείες, διότι παρέχει μεγάλη σταθερότητα, ταχύτητα και ευκολία εγκατάστασης.
- **Ελευθερία από πλατφόρμα Lock-in.** Με την δυνατότητα άμεσης πρόσβασης στον πηγαίο κώδικα MySQL, διασφαλίζεται η ελεύθερη προσέγγιση, εμποδίζοντας έτσι την αποκλειστικότητα σε ένα και μόνο προμηθευτή ή πλατφόρμα.
- **Cross-πλατφόρμα υποστήριξης.** Η MySQL είναι διαθέσιμη σε περισσότερες από είκοσι διαφορετικές πλατφόρμες, συμπεριλαμβανομένων όλων των μεγάλων διανομέων, Linux, Unix, Microsoft Windows και Mac OS X.

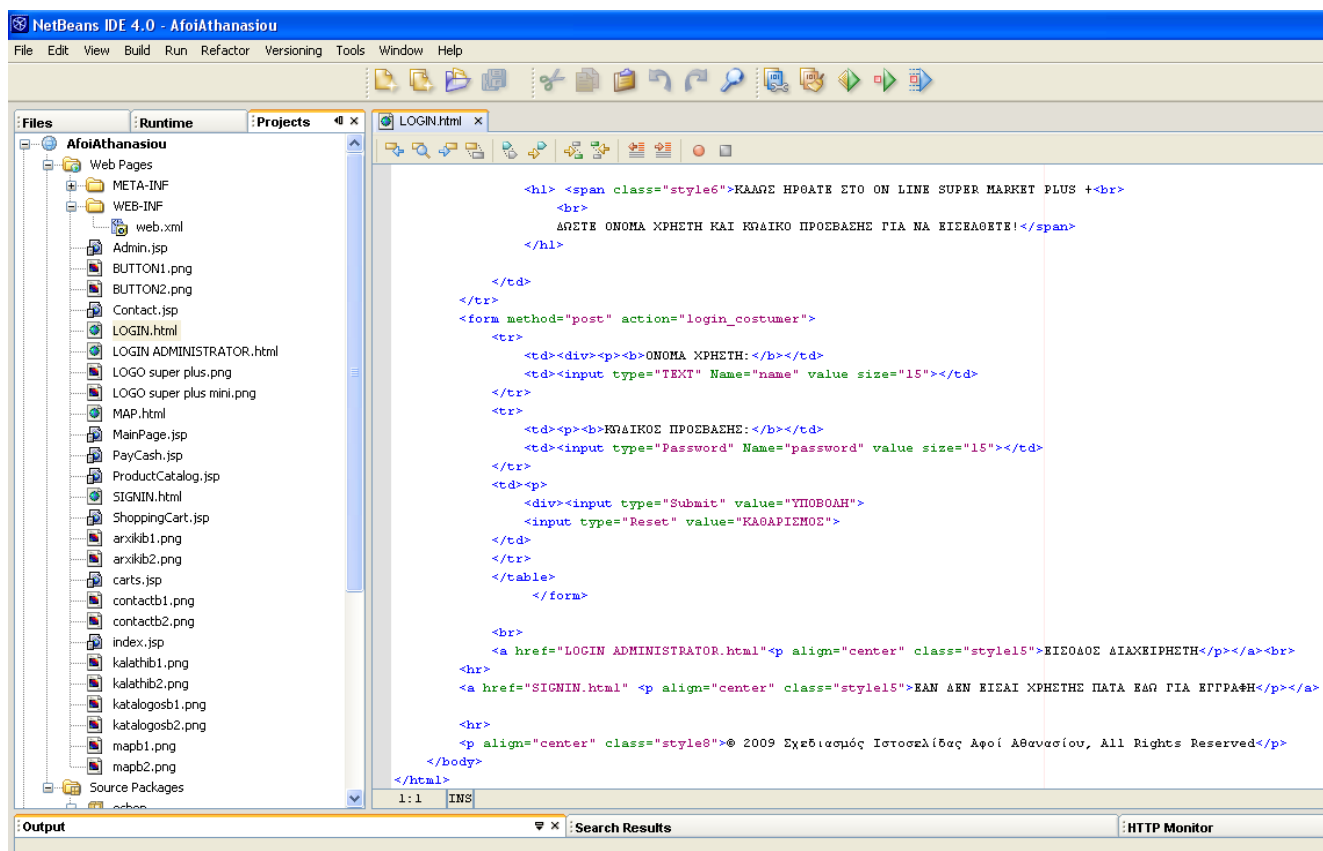
- **Εκατομμύρια εκπαιδευμένων και πιστοποιημένων Developers.** Η MySQL είναι η πιο δημοφιλής βάση δεδομένων ανοικτού κώδικα.
- **Απόδοση :** Η MySQL ηγείται σε αυτό τον τομέα, κυρίως λόγω της μορφής με την οποία αποθηκεύονται οι πίνακες. Αυτή η μορφή, με όνομα MyISAM, έχει πολύ μικρές απαιτήσεις σε επεξεργαστική ισχύ, μνήμη και χωρητικότητα.
- **Διάθεση πηγαίου κώδικα :** Όπως ήδη έχει αναφερθεί, ο πηγαίος κώδικας της MySQL είναι διαθέσιμος σε οποιονδήποτε θέλει να το μελετήσει ή να τον τροποποιήσει.
- **Κόστος άδειας χρήσης :** Είναι γεγονός πως η MySQL διανέμεται σε δυο εκδόσεις, όπου η μια είναι χωρίς κανένα κόστος και παρέχεται στο Internet, ενώ η δεύτερη, αν και επί πληρωμή, δεν έχει υπερβολικό κόστος. Αντίθετα, το κόστος της άδειας χρήσης άλλων συστημάτων βάσεων δεδομένων, είναι ένα σημαντικό στοιχείο που πρέπει να ληφθεί υπόψη.
- **Πρόσβαση από πολλούς χρήστες :** Η MySQL, με ευκολία μπορεί να διαχειρίζεται ταυτόχρονα πολλούς χρήστες, αφού από την αρχή είχε σχεδιαστεί να είναι ένα πολυχρηστικό (multi – user) σύστημα.
- **Διαχείριση μεγάλων βάσεων δεδομένων :** Όπως έχει ήδη αναφερθεί, η MySQL μπορεί να διαχειριστεί τεράστιες βάσεις δεδομένων, με εκατομμύρια εγγραφές και χιλιάδες πίνακες.
- **Ασφάλεια :** Η ασφάλεια σε άλλα συστήματα δεν αποτελεί δυνατό σημείο και αυτό οφείλεται στο ότι οι πίνακες, εφόσον αποθηκεύονται σε τοπικούς δίσκους, μπορούν να ανοιχτούν και να διαχειριστούν με απευθείας πρόσβαση στον υπολογιστή που βρίσκονται. Επιπλέον τη διαχείριση της ασφάλειας σε MySQL, την αναλαμβάνει ο ίδιος ο MySQL Server. Έτσι, οποιοσδήποτε προσπαθήσει να έχει πρόσβαση στα δεδομένα χωρίς τα σωστά στοιχεία (όνομα χρήστη και κωδικό πρόσβασης), θα αποτυγχάνει διαρκώς. ]

## 2.5. NetBeans IDE 4.0

Ωστόσο, για την ολοκλήρωση της εργασίας, είναι αναγκαία και η χρήση και κάποιων άλλων προγραμμάτων. Το **NetBeans** αποτελεί βοηθητικό πακέτο λογισμικού. Πρόκειται για πρόγραμμα δημιουργίας και επεξεργασίας κώδικα java και html.

Το NetBeans IDE είναι ένα ανοικτού κώδικα ολοκληρωμένο περιβάλλον ανάπτυξης το οποίο γράφτηκε εξ ολοκλήρου στην Java χρησιμοποιώντας την πλατφόρμα NetBeans. Το NetBeans IDE υποστηρίζει την ανάπτυξη όλων των τύπων εφαρμογής της Java (J2SE, WEB, EJB και κινητές εφαρμογές).

Το πρόγραμμα υποστηρίζει χρωματική μορφοποίηση του κειμένου με βάση τον κώδικα, εμφάνιση tips και οδηγιών κατά τη διάρκεια της πληκτρολόγησης του κώδικα κ.ά. (εικόνα 2.3.)



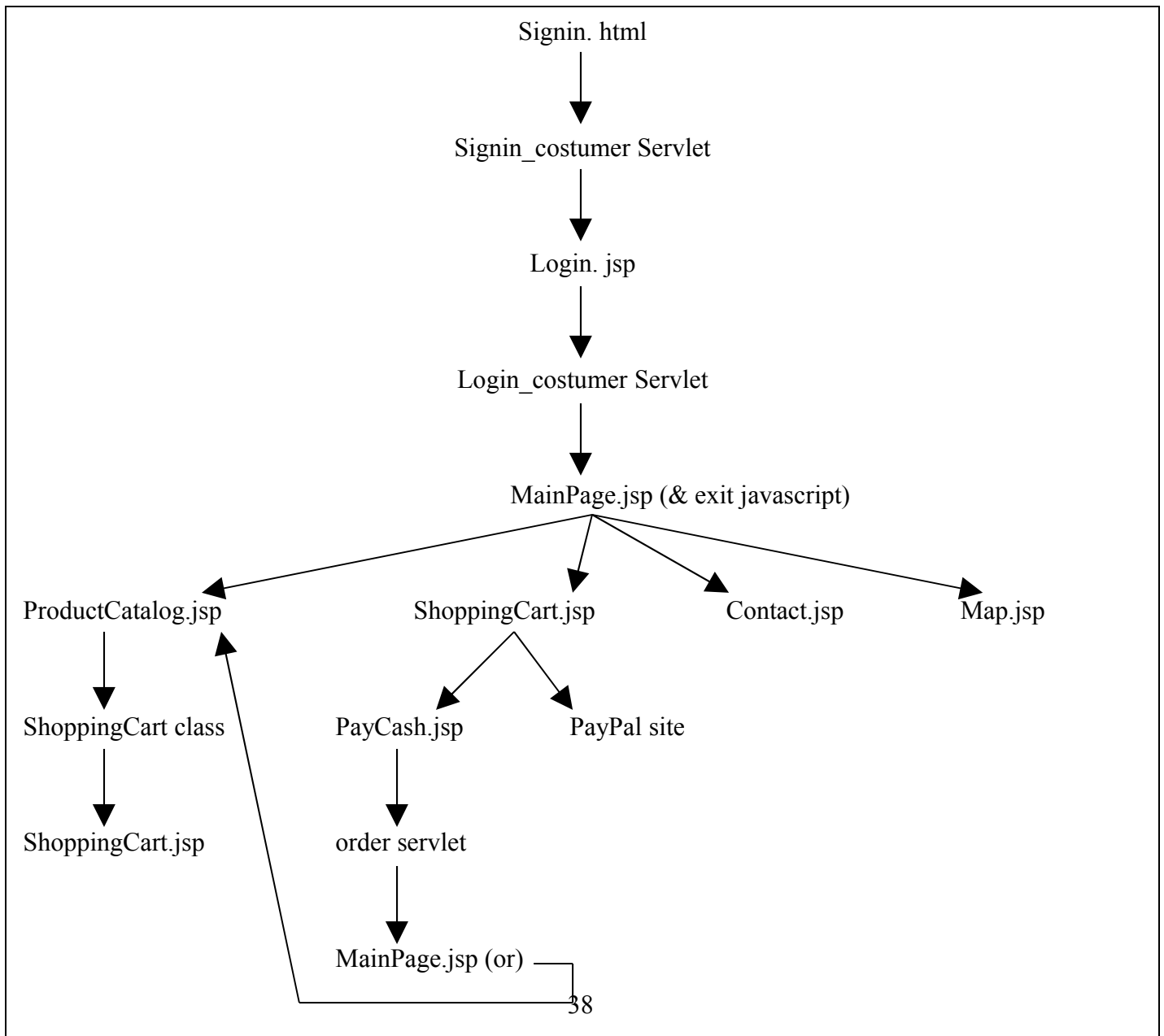
Εικόνα 2.3. Γραφικό περιβάλλον του NetBeans

# ΚΕΦΑΛΑΙΟ 3<sup>ο</sup>: ΣΧΕΔΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ - ΔΙΑΓΡΑΜΜΑΤΑ

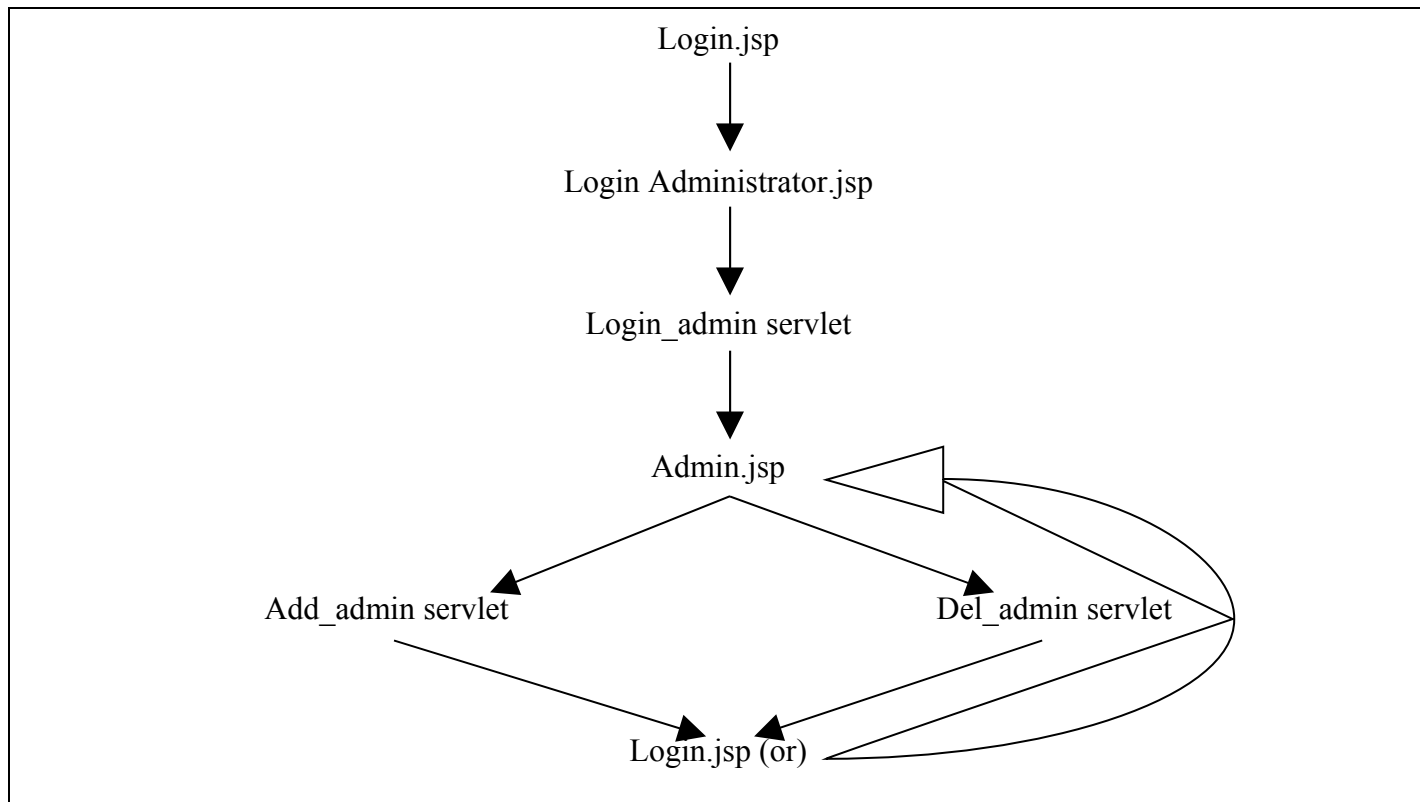
Σε αυτό το κεφάλαιο εμφανίζεται η πλήρης σχεδίαση και λειτουργία της εφαρμογής μέσα από διαγράμματα.

## **3.1. Ολοκληρωμένο Διάγραμμα Λειτουργίας**

**Διάγραμμα1.** Διάγραμμα Χρήστη

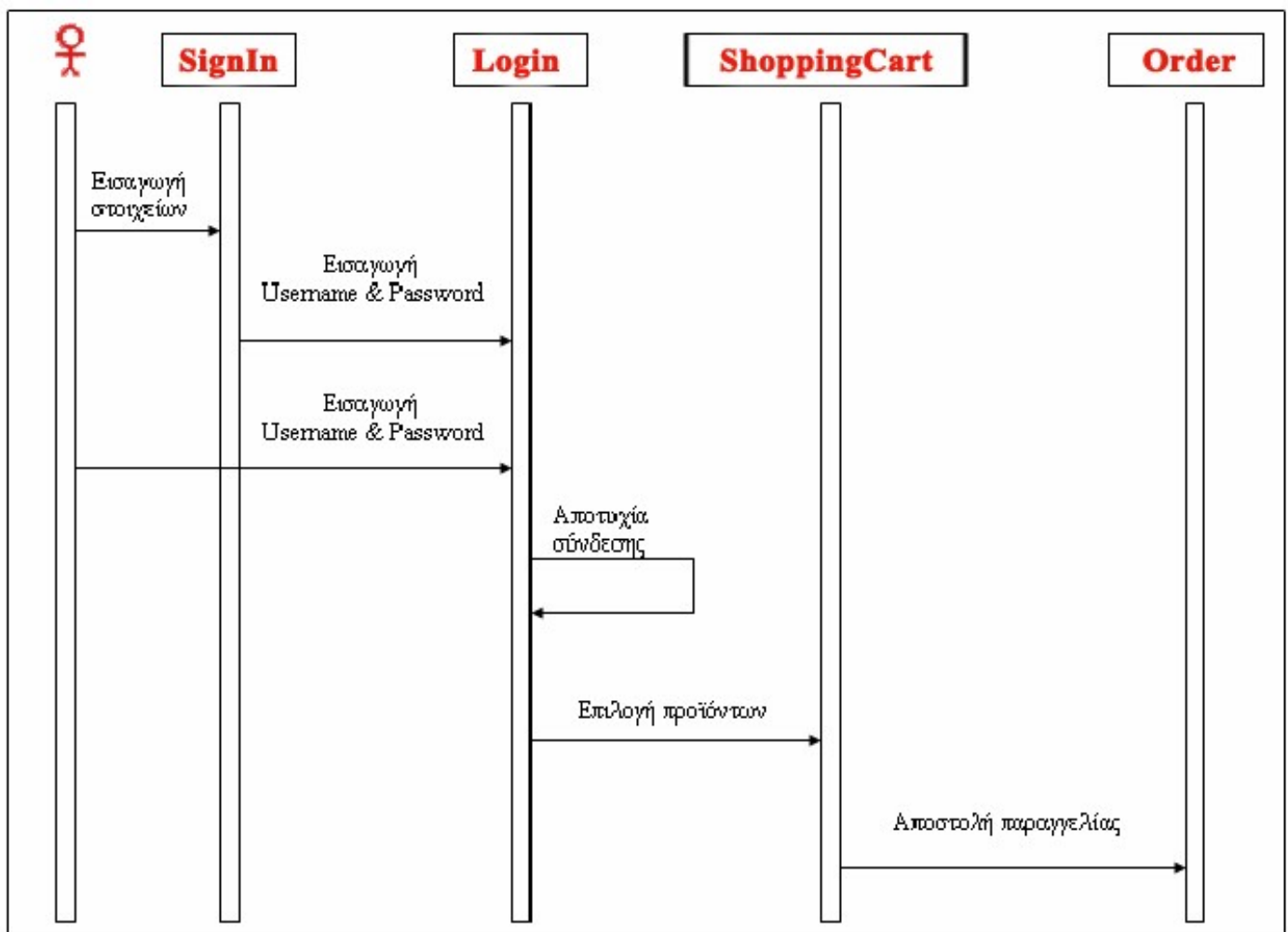


**Διάγραμμα2.** Διάγραμμα Διαχειριστή



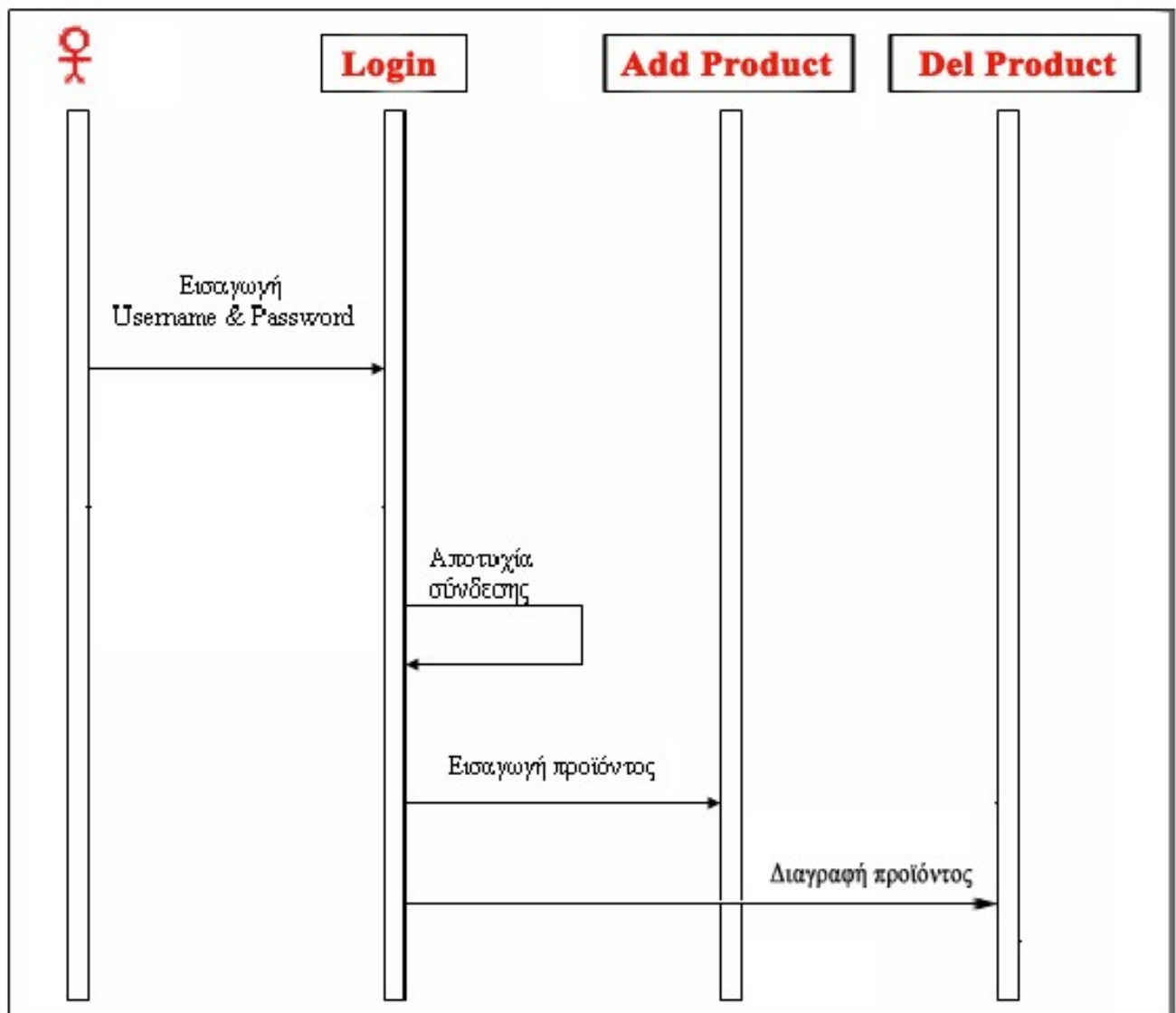
## 3.2. Διαγράμματα Ακολουθίας

Διάγραμμα3. Διάγραμμα ακολουθίας για την παραγγελία προϊόντων από εγγεγραμμένο ή μελλοντικό χρήστη





**Διάγραμμα4.** Διάγραμμα ακολουθίας για την προσθήκη προϊόντων στη βάση από το διαχειριστή



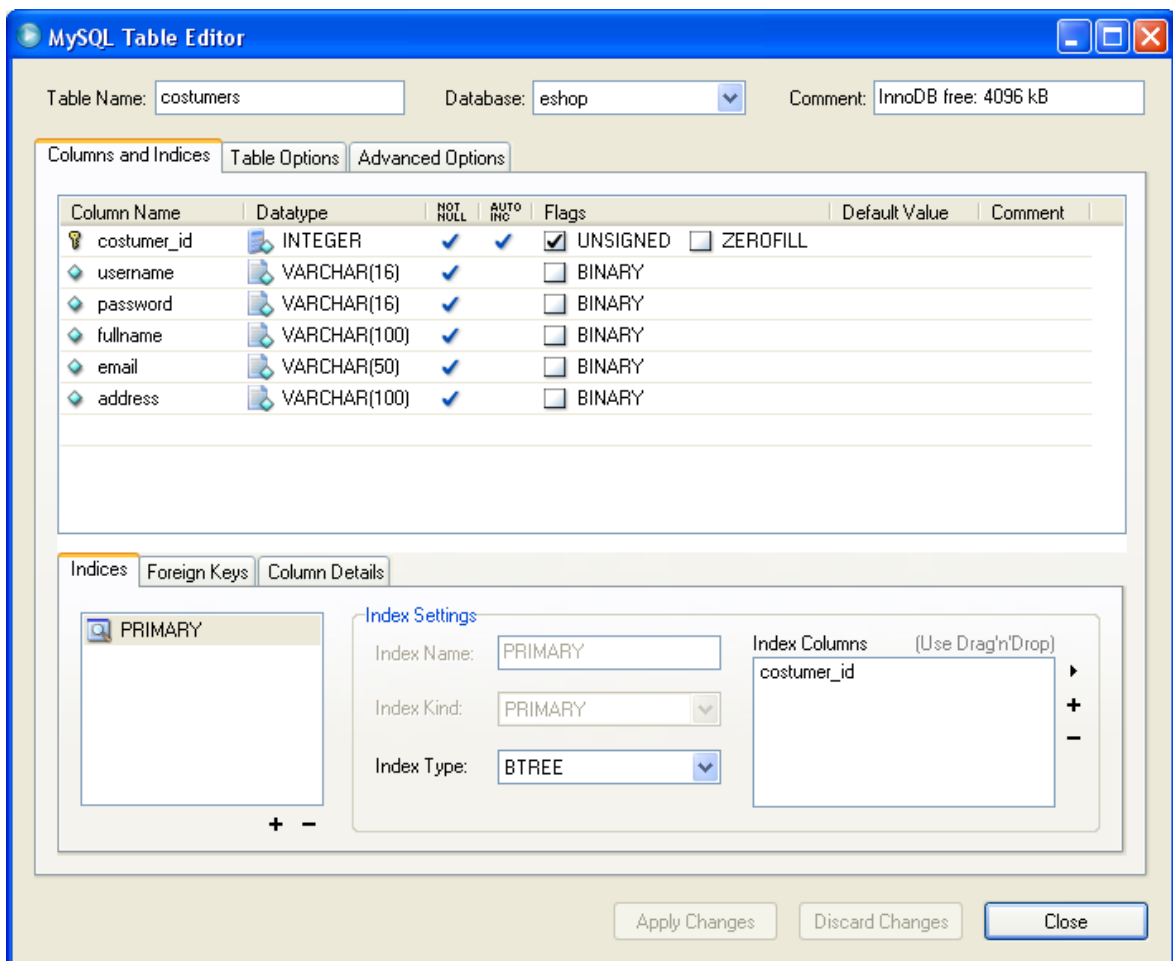


# ΚΕΦΑΛΑΙΟ 4<sup>ο</sup>: Η ΤΕΧΝΟΛΟΓΙΑ ΤΩΝ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

## 4.1. Περιγραφή Της Βάσης Δεδομένων

Δημιουργούμε την Βάση Δεδομένων με όνομα eshop, η οποία αποτελείται από τρεις πίνακες. Τον πίνακα των προϊόντων (product), των χρηστών-πελατών (costumers), και των παραγγελιών (orders). Για την δημιουργία τους χρησιμοποιούμε τον MySQL Administrator όπως φαίνεται στις εικόνες που ακολουθούνε.

### Δημιουργία του πίνακα costumers



## Δημιουργία του πίνακα product

The screenshot shows the MySQL Table Editor interface. At the top, the window title is "MySQL Table Editor". Below the title bar, there are fields for "Table Name: product", "Database: eshop", and "Comment: InnoDB free: 4096 kB".

The main area is divided into three tabs: "Columns and Indices" (selected), "Table Options", and "Advanced Options".

Under the "Columns and Indices" tab, there is a table with the following columns: Column Name, Datatype, NOT NULL, AUTO INC, Flags, Default Value, and Comment.

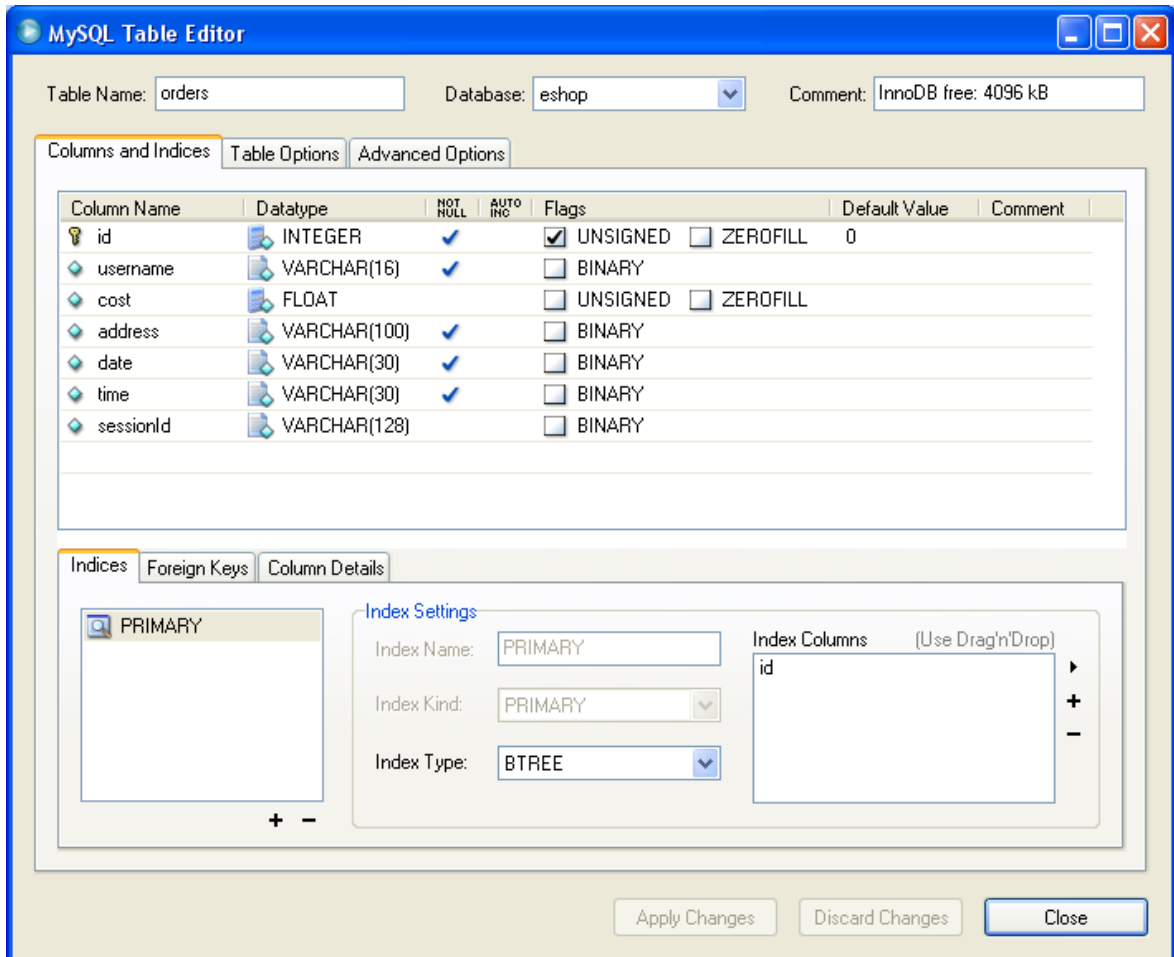
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
product_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		
description	VARCHAR(50)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
price	FLOAT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	0	
category	VARCHAR(50)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		

Below the columns table, there are tabs for "Indices", "Foreign Keys", and "Column Details". The "Indices" tab is selected, showing a list of indices with "PRIMARY" selected. To the right, the "Index Settings" section includes:

- Index Name: PRIMARY
- Index Kind: PRIMARY
- Index Type: BTREE
- Index Columns (Use Drag'n'Drop): product\_id

At the bottom of the window, there are three buttons: "Apply Changes", "Discard Changes", and "Close".

## Δημιουργία του πίνακα orders



Για να πραγματοποιηθεί η σύνδεση με την Βάση Δεδομένων (MySQL) χρησιμοποιείται από τον connector 'mysql-connector-java-3.0.9-stable-bin.jar' που είναι αποθηκευμένος στο φάκελο WEB-INF\lib η τάξη Driver. Στη συνέχεια δημιουργείται ένα αντικείμενο τύπου Connection όπου καλεί τη βάση μας 'eshop' ως χρήστης 'root'.

```
Connection con;
```

```
private String message;
```

```
public void init(ServletConfig config)
```

```
throws ServletException {
```

```
    super.init(config);
```

```

    try {
        Class.forName("org.gjt.mm.mysql.Driver");
        con=DriverManager.getConnection("jdbc:mysql://localhost/eshop?
user=root&password=root");
    } catch (Exception e) {
        message=e.getMessage();
    }
}

```

Ο κώδικας για να κλείσει η σύνδεση της βάσης με την εφαρμογή είναι ο παρακάτω:

```

public void destroy(){
    if(con!=null) {
        try{
            con.close();
        } catch(SQLException e) {
            log("Could't close connection with MySQL",e);
        }
    }
}
}

```

## 4.2. Περιγραφή Των JSP Και HTML Σελίδων

Αρχικά καθορίζεται η κωδικοποίηση της σελίδας ώστε να εμφανίζονται χωρίς προβλήματα οι ελληνικοί χαρακτήρες.

```
<%@page contentType="text/html"%>  
<%@page pageEncoding="UTF-8"%>
```

Σε πολλές από τις σελίδες είναι απαραίτητα και τα ακόλουθα πακέτα τάξεων.

```
<%@page import="java.io.*"%>  
<%@page import="java.net.*"%>  
<%@page import="java.util.*"%>  
<%@page import="javax.servlet.*"%>  
<%@page import="javax.servlet.http.*"%>  
<%@page import="java.sql.*"%>  
<%@page import="eshop.*"%>  
<%@page import="use.*"%>
```

Επίσης στις jsp και html σελίδες εισάγεται το στυλ των γραμμάτων

```
...  
<style type="text/css" Encoding="UTF-8">
```

```
    body {  
        background-color: #6666E3  
    }  
    h1 {background-color: #FCE047}  
    h1 {color: #000000}  
    h1 {text-align: center}  
    div {color: #000000}  
    tr {text-align: left}  
    input {align: right}  
    div {text-align: center}  
    .style8 {color: #FCE047; font-size: 12px;}
```

```

#news {
font-family:"times new roman";
font-size:18;
position:absolute;
top:5px;
left:255px;
height:146px;
text-align:center;
padding-left:10;
padding-right:10;
width: 514px;
font-weight: bold;
color: #FCE047;
}
.style14 {
font-size: large;
font-weight: bold;
color: #FCA000;
}
.style15 {
font-size: medium;
font-weight: bold;
color: #FCE047;
}
.style16 {font-size: large}
</style>

```

Τέλος στις σελίδες του χρήστη αλλά και του διαχειριστή εμφανίζεται στην αρχή ο παρακάτω κώδικας:

```

<% response.setHeader("Pragma", "no-cache");%>
<% response.setHeader("Cache-Control", "no-store");%>
<% response.setDateHeader("Expires",-1);%>

```



Αυτός ο κώδικας έχει εισαχθεί λόγω του ότι σε κάθε σελίδα πρέπει να κρατείται το session της εισόδου του χρήστη και του διαχειριστή αντιστοίχως, οι οποίες τιμές μπορούμε να πούμε ότι είναι τιμές που κάθε φορά θα πρέπει να «ανανεώνονται» μεταβαίνοντας από τη μια σελίδα στην άλλη, έτσι λοιπόν καταφέρνουμε να κρατούνται τα sessions σε κάθε ανανέωση της ιστοσελίδα ή κάθε φορά που ο χρήστης επιλέγει να μετακινηθεί με τα κουμπιά «εμπρός» και «πίσω» χωρίς να χάνεται το session.

#### 4.2.1. Εγγραφή χρήστη (SIGNIN.html)

Μέσω της σελίδας SIGNIN.html ο επισκέπτης μπορεί να εγγραφεί χρήστης. Με την παρακάτω φόρμα ο χρήστης εισάγοντας τα προσωπικά του δεδομένα μπορεί να κάνει την εγγραφή του, η φόρμα καλεί το servlet `signin_costumer` (σελ.62), για να περάσει μέσω αυτού τα στοιχεία του νέου χρήστη στη βάση και συγκεκριμένα στον πίνακα `costumer`. Επίσης εδώ καλείτε το javascript `ParseRegCost` (σελ 75) το οποίο αποτρέπει την εισαγωγή περισσότερων ή λιγότερων χαρακτήρων στα ανάλογα πεδία, όπως επίσης ελέγχει για τη σωστή επαναπληκτρολόγηση του κωδικού πρόσβασης και για την μη αποδεκτή τιμή των 'root' ως όνομα χρήστη, λόγω του ότι αντιστοιχεί στον διαχειριστή.

```
<h3 class="style1"><u> ΣΥΜΠΛΗΡΩΣΕ ΤΑ ΣΤΟΙΧΕΙΑ ΤΗΣ ΦΟΡΜΑΣ ΜΕ ΤΑ ΑΝΤΙΣΤΟΙΧΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ</u></h3>
<form method="post" action="signin_costumer" onSubmit="javascript:return ParseRegCost(this)">
  <p><span class="style1">ΟΝΟΜΑ ΧΡΗΣΤΗ (3-16 chars):</span>
  <input type="text" name="username" size="16" maxlength="16"></p>
  <p><span class="style1">ΚΩΔΙΚΟΣ ΠΡΟΣΒΑΣΗΣ (6-16 chars) :</span>
  <input type="text" name="password" size="16" maxlength="16"></p>
  <p><span class="style1">ΕΠΑΝΑΠΛΗΚΤΡΟΛΟΓΙΣΕ ΚΩΔΙΚΟ ΠΡΟΣΒΑΣΗΣ (Confirm):</span>
  <input type="text" name="password2" size="16" maxlength="16"></p>
  <p><span class="style1">ΟΝΟΜΑΤΕΠΩΝΥΜΟ:</span>
  <input type="text" name="fullname" size="30"></p>
  <p><span class="style1">E-mail:</span>
  <input type="text" name="email" size="20"></p>
  <p><span class="style1">ΔΙΕΥΘΥΝΣΗ:</span>
  <input type="text" name="address" size="50"></p>

  <font color="#000000"></font>
  <p><input type="submit" value="ΥΠΟΒΟΛΗ"> <input type="Reset" value="ΚΑΘΑΡΙΣΜΟΣ" ></p>
</form>
```

## 4.2.2. Είσοδος χρήστη (Login.jsp)

Η σελίδα Login.jsp είναι η σελίδα που καλείται για τη σύνδεση του χρήστη. Μέσω της παρακάτω φόρμας ο χρήστης συνδέεται εισάγοντας το ‘Όνομα χρήστη’ και τον ‘Κωδικό’ του. Η ταυτοποίηση των τιμών που εισάγονται γίνεται στο servlet login\_costumer (σελ.64).

```
<form method="post" action="login_costumer">
  <tr>
    <td><div><p><b>ΟΝΟΜΑ ΧΡΗΣΤΗ:</b></td>
    <td><input type="TEXT" Name="name" value size="15"></td>
  </tr>
  <tr>
    <td><p><b>ΚΩΔΙΚΟΣ ΠΡΟΣΒΑΣΗΣ:</b></td>
    <td><input type="Password" Name="password" value size="15"></td>
  </tr>
  <td><p>
    <div><input type="Submit" value="ΥΠΟΒΟΛΗ">
    <input type="Reset" value="ΚΑΘΑΡΙΣΜΟΣ">
  </td>
</tr>
</table>
</form>
```

## 4.2.3. Είσοδος διαχειριστή (LoginAdministrator.jsp)

Η σελίδα LoginAdministrator.jsp είναι η σελίδα που καλείται για τη σύνδεση του διαχειριστή. Μέσω της παρακάτω φόρμας ο διαχειριστής συνδέεται εισάγοντας το ‘ADMIN\_NAME’ και τον ‘ADMIN\_PASSWORD’ του. Η ταυτοποίηση των τιμών που εισάγονται γίνεται στο servlet login\_admin (σελ.65)..

```
---
<form method="post" action="login_admin">
  <tr>
    <td><div><p><b>ADMIN_NAME:</b></td>
    <td><input type="TEXT" Name="aname" value size="15"></td>
  </tr>
  <tr>
    <td><p><b>ADMIN_PASSWORD:</b></td>
    <td><input type="Password" Name="apassword" value size="15"></td>
  </tr><br>
  <br>
  <td><p>
    <div><input type="Submit" value="ΥΠΟΒΟΛΗ">
    <input type="Reset" value="ΚΑΘΑΡΙΣΜΟΣ">
  </td>
</tr>
</table>
</form>
```

#### 4.2.4. Η σελίδα του διαχειριστή (Admin.jsp)

The screenshot shows the Admin.jsp page for Super Market PLUS+. The page has a blue background with a yellow banner at the top containing the text "ΣΕΛΙΔΑ ΔΙΑΧΕΙΡΗΣΤΗ ON LINE SUPER MARKET PLUS +" and "ΚΑΛΩΣ ΗΡΘΕΣ!". Below the banner, there are two main sections: "ΠΡΟΣΘΗΚΗ ΠΡΟΪΟΝΤΩΝ" (Add Product) and "ΔΙΑΓΡΑΦΗ ΠΡΟΪΟΝΤΩΝ" (Delete Product). The "ΠΡΟΣΘΗΚΗ ΠΡΟΪΟΝΤΩΝ" section includes a text input for "Δώσε περιγραφή:", a text input for "Δώσε τιμή:", a dropdown menu for "Επίλεξε Κατηγορία:" with "Επίλεξε" selected, and a "ΠΡΟΣΘΗΚΗ" button. The "ΔΙΑΓΡΑΦΗ ΠΡΟΪΟΝΤΩΝ" section includes a text input for "Δώσε τον Κωδικό του Προϊόντος που θέλεις να διαγράψεις:" and a "ΔΙΑΓΡΑΦΗ" button. At the bottom of the page, there is a small copyright notice: "© 2009 Σχεδιασμός Ιστοσελίδας Αφοι Αθανασίου, All Rights Reserved".

Η σελίδα αυτή εμφανίζεται εφόσον ο διαχειριστής κάνει login από την σελίδα LoginAdministrator.jsp.

Ο διαχειριστή από την σελίδα του μπορεί να προβεί σε 2 ενέργειες :

##### **1. Προσθήκη νέου προϊόντος στη βάση**

Η οποία πραγματοποιείται από την παρακάτω φόρμα καλώντας το servlet add\_admin (σελ. 66).

```

<form method="post" action="add_admin">
<br>
<br>
<table width="1136" border="1">
  <tr>
    <td width="383"><p align="center"><a align="center" class="style14">ΠΡΟΣΘΗΚΗ ΠΡΟΪΟΝΤΩΝ</a></p>
    <p><a align="center" class="style15">Δώσε περιγραφή:
      <input type="text" name="prde" size=20>
    </a> <br>
    <br>
    <a align="center" class="style15"> Δώσε τιμή:
      <input type="text" name="prp" size=10>
    </a> <br>
    <br>
    <a align="center" class="style15"> Επίλεξε Κατηγορία: </a>
    <select name="prkat">
      <option>Επίλεξε</option>
      <option>ΟΠΗΡΟΠΩΡΑΕΙΟ</option>
      <option>ΚΡΕΟΠΩΡΑΕΙΟ</option>
      <option>ΙΧΘΥΟΠΩΡΑΕΙΟ</option>
      <option>ΓΑΛΑΚΤΟΚΟΜΙΚΑ-ΤΥΡΟΚΟΜΙΚΑ</option>
      <option>ΑΛΛΑΝΤΙΚΑ</option>
      <option>ΕΙΔΗ ΠΑΝΤΟΠΩΡΑΕΙΟΥ-ΤΡΟΦΙΜΑ</option>
      <option>ΧΑΡΤΙΚΑ-ΑΤΟΜΙΚΗ ΥΓΙΕΙΝΗ</option>
    </select>
    <br>
    <br>
    <input type="Submit" value="ΠΡΟΣΘΗΚΗ">
  </p></td></form>

```

## 2. Διαγραφή προϊόντος από τη βάση

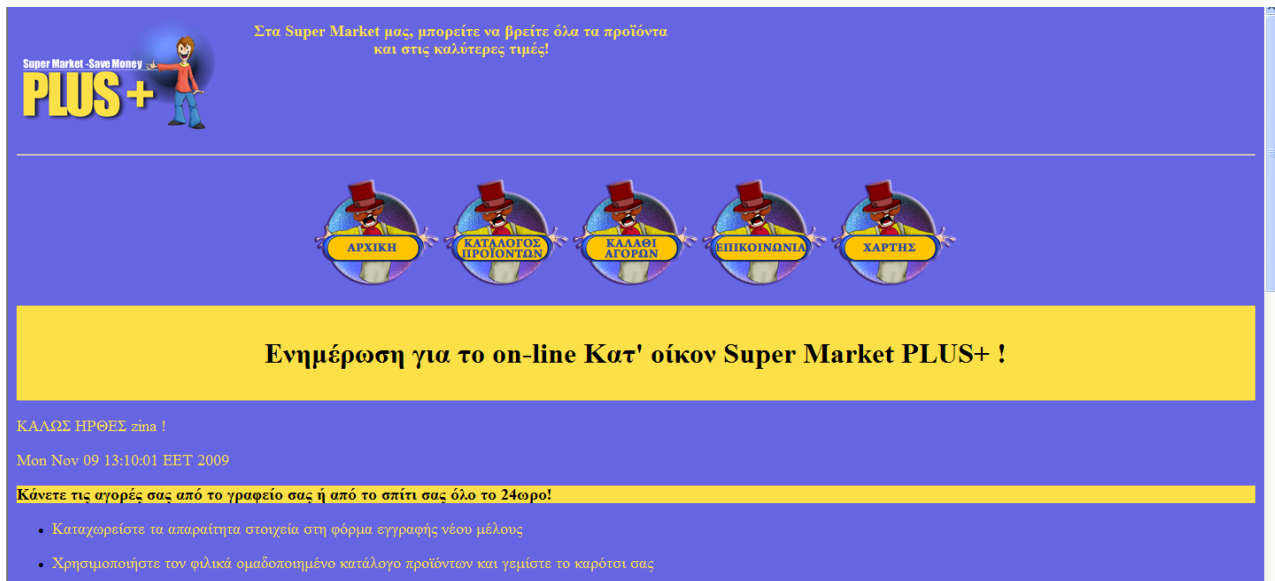
Η οποία πραγματοποιείται από την παρακάτω φόρμα καλώντας το servlet del\_admin (σελ67).

```

<form method="post" action="del_admin" >
<td width="366"><p align="center"><a align="center" class="style14">ΔΙΑΓΡΑΦΗ ΠΡΟΪΟΝΤΩΝ</a></p>
<a align="center" class="style15"> Δώσε τον Κωδικό του Προϊόντος που θέλεις να διαγράψεις: </a>
<input type="text" name="delID" size=10>
<br>
<br>
<input type="Submit" value="ΔΙΑΓΡΑΦΗ" >
</p></td></form>

```

## 4.2.5. Η σελίδα του χρήστη (MainPage.jsp)



Η MainPage.jsp είναι η σελίδα του χρήστη. Ο χρήστης μπορεί να την καλέσει από οποιαδήποτε σελίδα επιλέγοντας το αντίστοιχο menu button (Αρχική) . Στη συγκεκριμένη σελίδα επιστρέφουμε από τη βάση την τιμή του ονόματος του χρήστη χρησιμοποιώντας sessions, επίσης εμφανίζουμε την τρέχουσα ημερομηνία. Αυτός ο κώδικας εμφανίζεται σε όλες τις jsp σελίδες του χρήστη.

```
<%  
String username=(String) session.getAttribute("login.isDone");  
if(username==null) username="";  
%>  
  
<p align="justify" class="style15">ΚΑΛΩΣ ΗΡΘΕΣ <%=username%> !</p>  
  
<p align="justify" class="style15"> <%= new java.util.Date() %></p>
```

Επίσης στο τέλος των σελίδων του χρήστη εμφανίζεται ο κώδικας: `a href="Login.jsp?logout=true"><p align="justify" class="style15">ΕΞΟΔΟΣ ΑΠΟ ΤΟ ΚΑΤΑΣΤΗΜΑ!</p></a>` για την έξοδο του από το κατάστημα.

Από τη σελίδα του ο χρήστης μπορεί να παραγγείλει προϊόντα, να δει το καλάθι των αγορών του, να επικοινωνήσει με τους υπεύθυνους και να δει το χάρτη του website. Οι παραπάνω ενέργειες θα αναπτυχθούν παρακάτω.

Εδώ εμφανίζεται ο κώδικας για τα menu buttons

```
<p align="center">
    <a href="MainPage.jsp" onMouseOut="MM_swapImgRestore()" on-
MouseOver="MM_swapImage('Image2','arxikib2.png',1)"></a>
    <a href="ProductCatalog.jsp" onMouseOut="MM_swapImgRestore()" on-
MouseOver="MM_swapImage('Image3','katalogosb2.png',1)"></a>
    <a href="ShoppingCart.jsp" onMouseOut="MM_swapImgRestore()" on-
MouseOver="MM_swapImage('Image4','kalathib2.png',1)"></a>
    <a href="Contact.jsp" onMouseOut="MM_swapImgRestore()" on-
MouseOver="MM_swapImage('Image5','contactb2.png',1)"></a>
    <a href="MAP.html" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('Image6','mapb2.png',1)"></a> </p>
```

#### 4.2.6. Ο κατάλογος προϊόντων (ProductCatalog.jsp)

Η σελίδα αυτή εμφανίζει τα προϊόντα ανά κατηγορία με δυνατότητα επιλογής προϊόντος ενός ή περισσότερων για την προσθήκη τους στο καλάθι. Εδώ εισάγεται η κλάση ShoppingCart.java που θα προσθέτει τα προϊόντα στο καλάθι με τη βοήθεια των βοηθητικών συναρτήσεων που περιέχει σε συνεργασία κάποιων από αυτών με τα ανάλογα java scripts (σελ. 75).

```
.....
<jsp:useBean id="cart" scope="session" class="eshop.ShoppingCart" />
.....
Στο πάνω μέρος της σελίδας εμφανίζεται πάντα η υπάρχουσα ποσότητα των προϊόντων που
είναι στο καλάθι καθώς και το συνολικό κόστος
...
    <a href="ShoppingCart.jsp"><p align="justify" class="style15">Η ποσότητα στο
καλάθι αγορών σας είναι: <%=cart.getNumOfItems() %></p>
        <p align="justify" class="style15">Το συνολικό ποσό σας είναι: <
%=cart.getCost() %> €</p>
...

```

Όλα τα δεδομένα αυτής της σελίδας περικλείονται σε μια φόρμα ξεκινώντας από τις κλείσεις και το πέρασμα παραμέτρων για τα javascripts (σελ. 75).

```

.....
<form>
    <input type="hidden" name="product_id_opor" value="<%=request.getParameter("product_id_opor")%>">
        <input type="hidden" name="des_opor" value="<
%=request.getParameter("des_opor")%>">
            <input type="hidden" name="pr_opor" value="<
%=request.getParameter("pr_opor")%>">

    <%
String product_id_opor = request.getParameter("product_id_opor");
String quantity1 = request.getParameter("quantity1");
String des_opor = request.getParameter("des_opor");
String price_opor = request.getParameter("pr_opor");

System.out.println("product_id"+ product_id_opor);
if ( product_id_opor != null && !product_id_opor.equals("")) {
    try{
        float price=Float.parseFloat(price_opor);
        int quantity=Integer.parseInt(quantity1);
        cart.addItem(product_id_opor, des_opor, price, quantity );
    }catch(Exception e1){
    }
}
}
%>
.....

```

Το παρακάτω κομμάτι κώδικα μας εμφανίζει τα προϊόντα κατηγοριοποιημένα, ανακτώντας τα από τον πίνακα product της βάσης δεδομένων.

```

.....
<%! Connection con;%>
<%
    try {

        Class.forName("org.gjt.mm.mysql.Driver");

        con = DriverManager.getConnection("jdbc:mysql://localhost/eshop?
user=root&password=root");
    }

    catch (ClassNotFoundException e) {
        log("Couldn't find mm.mysql driver");
    }
    catch (SQLException e) {
        log("SQL Exception while connecting to MySQL", e);
    }

    if (con!=null){

```

```

try {
    Statement stmt = con.createStatement();%>

    <table width=716 border=1 align="center">
        <tr bgcolor="#FFCC99">
            <td width=179><div
align="center"><strong>ΟΠΩΡΟΠΩΛΕΙΟ</strong></div></td>
            <td width=179><div
align="center"><strong>ΚΡΕΟΠΩΛΕΙΟ</strong></div></td>
            <td width=179><div
align="center"><strong>ΙΧΘΥΟΠΩΛΕΙΟ</strong></div></td>
            <td width=179><div align="center"><strong>ΓΑΛΑΚΤΟΚΟΜΙΚΑ-
ΤΥΠΟΚΟΜΙΚΑ</strong></div></td>

        </tr>
        <tr>
            <td bgcolor="#FFFFCC">
                <% ResultSet crs1 = stmt.executeQuery("SELECT * FROM product order by de-
scription");
                out.println("<select name=category1 class=form1>"+
                    "<option>Επιλέξε</option>");
                String des1 = "null";
                float pr1 = 0;
                String cat1 = "ΟΠΩΡΟΠΩΛΕΙΟ";
                while (crs1.next()) {
                    String description = crs1.getString("description");
                    float price=Float.parseFloat(crs1.getString("price"));
                    int product_id=Integer.parseInt(crs1.getString("product_id"));

                    if (cat1.equals(crs1.getString("category"))){
                        if( des1.equals(description) ){
                            crs1.next();
                            crs1.previous();
                        }
                        else {
                            out.println("<option value="+product_id+">"+description+",
"+price+" €</option>");
                            des1=description;
                            pr1=price;
                        }
                    }
                }
                out.println("</select>"+
                    "</td>"+
                    "<td bgcolor=#FFFFCC>");
                .....

```



Στο παρακάτω κομμάτι κώδικα εμφανίζεται η εισαγωγή της ποσότητας των προϊόντων στο τέλος της οποίας υπάρχει ένα button για να καλεί το ανάλογο Javascript (σελ. 75) για να γίνεται η προσθήκη των προϊόντων , όπως αναφέρθηκε παραπάνω.

```
<tr>
    <td bgcolor="#FFFFCC">Ποσότητα :
        <input name="quantity1" size=3>
    </td>
.....
.....

    <tr>
    <td bgcolor="#FFFFCC">
        <input type="button" name="Button1" value="ΠΡΟΣΘΗΚΗ"
onclick="javascript:opor();">
        <input type="hidden" name="quantity1">
    </td>
.....
.....
```

Τέλος υποβάλουμε τη φόρμα

```
.....
<div> <input type="submit" value="ΠΡΟΣΘΗΚΗ ΣΤΟ ΚΑΛΑΘΙ"></div>
    </form>
.....
```

Το βασικό μέρος της εμφάνιση της σελίδας ProductCatalog.jsp φαίνεται παρακάτω.

## ΚΑΤΑΛΟΓΟΣ ΠΡΟΪΟΝΤΩΝ

### Οδηγίες για τα ψώνια σας:

1. Επιλέξτε το προϊόν που θέλετε απο την αναδυόμενη λίστα κάθε κατηγορίας
2. Πληκτρολογήστε την ποσότητα που επιθυμείτε
3. Πατήστε το κουμπί 'ΠΡΟΣΘΗΚΗ' για κάθε προϊόν που επιλέγετε
4. Τέλος πατήστε το κουμπί 'ΠΡΟΣΘΗΚΗ ΣΤΟ ΚΑΛΑΘΙ'

Η ποσότητα στο καλάθι αγορών σας είναι: 0

Το συνολικό ποσό σας είναι: 0.0 €

ΟΠΩΡΟΠΩΛΕΙΟ	ΚΡΕΟΠΩΛΕΙΟ	ΙΧΘΥΟΠΩΛΕΙΟ	ΓΑΛΑΚΤΟΚΟΜΙΚΑ-ΤΥΡΟΚΟΜΙΚΑ
Επίλεξε <input type="text"/>	Επίλεξε <input type="text"/>	Επίλεξε <input type="text"/>	Επίλεξε <input type="text"/>
Ποσότητα : <input type="text"/>	Ποσότητα : <input type="text"/>	Ποσότητα : <input type="text"/>	Ποσότητα : <input type="text"/>
<input type="button" value="ΠΡΟΣΘΗΚΗ"/>	<input type="button" value="ΠΡΟΣΘΗΚΗ"/>	<input type="button" value="ΠΡΟΣΘΗΚΗ"/>	<input type="button" value="ΠΡΟΣΘΗΚΗ"/>

ΑΛΛΑΝΤΙΚΑ	ΕΙΔΗ ΠΑΝΤΟΠΩΛΕΙΟΥ-ΤΡΟΦΙΜΑ	ΧΑΡΤΙΚΑ-ΑΤΟΜΙΚΗ ΥΓΙΕΙΝΗ
Επίλεξε <input type="text"/>	Επίλεξε <input type="text"/>	Επίλεξε <input type="text"/>
Ποσότητα : <input type="text"/>	Ποσότητα : <input type="text"/>	Ποσότητα : <input type="text"/>
<input type="button" value="ΠΡΟΣΘΗΚΗ"/>	<input type="button" value="ΠΡΟΣΘΗΚΗ"/>	<input type="button" value="ΠΡΟΣΘΗΚΗ"/>

© 2009 Σχεδιασμός Ιστοσελίδας Αρτσι Αθνασιου. All Rights Reserved

### 4.2.7. Το καλάθι αγορών (ShoppingCart.jsp)

Σε αυτή τη σελίδα ο χρήστης μπορεί να δει την παραγγελία του αναλυτικά χρησιμοποιώντας την συνάρτηση `getEnumeratino` της κλάσης `cart`, η οποία έχει δηλωθεί στην αρχή της σελίδας:

```
<jsp:useBean id="cart" scope="session" class="eshop.ShoppingCart" />
```

Και επιπλέον μπορεί να διαγράψει προϊόντα από το καλάθι του μέσω της χρήσης javascript (σελ. 76).

...

```
</form>
```

```
<input type="hidden" id="wichRow" name="wichRow" />
<table width="300" border="1" cellspacing="0"
  cellpadding="2" border="0">
  <h1>Περιεχόμενα Καλαθιού</h1>
  <tr>
    <th align="center" class="style15">ΚΩΔ. ΠΡΟΪΟΝΤΟΣ</th>
    <th align="center" class="style15">ΠΕΡΙΓΡΑΦΗ</th>
    <th align="center" class="style15">ΤΙΜΗ</th>
    <th align="center" class="style15">ΠΟΣΟΤΗΤΑ</th>
    <th align="center" class="style15">ΔΙΑΓΡΑΦΗ</th>
  </tr>
```

```

<%

String row = request.getParameter("wichRow");
if ( row != null )
    cart.removeItem(row);
Enumeration e = cart.getEnumeration();
String[] tmpItem;
// Iterate over the cart

int i=0;
while (e.hasMoreElements()) {
    tmpItem = (String[])e.nextElement();
    %>
        <tr>
            <td align="center"><b><%=tmpItem[0] %></b></td>
            <td align="center"><b><%=tmpItem[1] %></b></td>
            <td align="center"><b><%=tmpItem[2] %> €</b></td>
            <td align="center"><b><%=tmpItem[3] %></b></td>
            <td align="center">
                <input type="button" name="but" value="Delete" onClick="javascript:
deleteRow(<%=tmpItem[0] %>); form.submit();"/>
            </td>
        </tr>

    <%
        i++;
    }
    %>

```

Επιστροφή στον κατάλογο

### Περιεχόμενα Καλαθιού

ΚΩΔ. ΠΡΟΪΟΝΤΟΣ	ΠΕΡΙΓΡΑΦΗ	ΤΙΜΗ	ΠΟΣΟΤΗΤΑ	ΔΙΑΓΡΑΦΗ
24	Γιαούρτι (τμχ.)	1.3 €	2	Delete
44	Σοκολάτα ρόφημα (100 gr)	4.25 €	2	Delete
8	Κοτόπουλο (τμχ.)	7.2 €	1	Delete

ΣΥΝΟΛΟ ΠΡΟΪΟΝΤΩΝ: 5, ΣΥΝΟΛΙΚΟ ΚΟΣΤΟΣ: 18.3 €

ΕΠΙΛΕΞΤΕ ΤΡΟΠΟ ΓΙΑ ΟΛΟΚΛΗΡΩΣΗ ΠΑΡΑΤΗΛΙΑΣ

© 2009 Συστήμας, Ινστιτούτος Αφοι Αθηνών, All Rights Reserved

Το σύνολο των προϊόντων όπως και το συνολικό κόστος της αγοράς εμφανίζονται χρησιμοποιώντας της συναρτήσεις `getNumOfItems()` και `getCost()` της κλάσης `ShoppingCart` (σελ. 71).

Επίσης μπορεί μέσω του ανάλογου συνδέσμου να επιστρέψει στον κατάλογο για να συνεχίσει την παραγγελία προσθέτοντας νέα προϊόντα σε αυτή

Τέλος μπορεί να επιλέξει εάν θα πληρώσει με αντικαταβολή ή με πιστωτική κάρτα. Στην περίπτωση που θα επιλέξει να πληρώσει με πιστωτική κάρτα θα εμφανιστεί η σελίδα του paypal και αυτό λειτουργεί μέσω του ανάλογου javascript (σελ. 77).

...

```
<p><input type="button" value="ΑΝΤΙΚΑΤΑΒΟΛΗ" name="cash" onClick="javascript: PayCash(this); form.submit();">
```

```
<input type="button" value="ΠΙΣΤΩΤΙΚΗ ΚΑΡΤΑ (PayPal)" name="cart" onClick="javascript: PayPal(this); form.submit();"></p>
```

...

#### 4.2.8. Σελίδα παραγγελίας με αντικαταβολή (PayCash.jsp)

Στην περίπτωση που η πληρωμή γίνει με αντικαταβολή θα εμφανιστεί η παρακάτω σελίδα

Συμπληρώνοντας τα ανάλογα πεδία στη φόρμα της σελίδας καλείτε το servlet order και μέσω αυτού η παραγγελία καταχωρείτε στον πίνακα orders της βάσης.

....

```
<form method="post" action="order">
```

```
<input type="hidden" id="cost" name="cost" value="<%=cart.getCost() %>" />
```

```
<br>
```

```
<br>
```

```

        <a align="center" class="style14"> ΣΥΝΟΛΟ ΠΡΟΪΟΝΤΩΝ: <%=cart.getNu-
mOfItems() %>, </a>
        <a align="center" class="style14"> ΣΥΝΟΛΙΚΟ ΚΟΣΤΟΣ: <%=cart.getCost()
%> €</a> <br>
        <br>
        <br>
        <a align="center" class="style15"> Δώσε Διεύθυνση:
        <input name="ADDRESS" size=20> </a> <br>
        <br>
        <a align="center" class="style15"> Δώσε ημερομηνία παράδοσης
(dd/mm/yyyy):
        <input name="DATE" size=3> </a> <br>
        <br>
        <a align="center" class="style15"> Δώσε ώρα παράδοσης:
        <select name="TIME">
            <option value="0000">Αδιάφορη</option>
            <option value="0800">08:00</option>
            <option value="0830">08:30</option>
            <option value="0900">09:00</option>
            <option value="0930">09:30</option>
            <option value="1000">10:00</option>
            <option value="1030">10:30</option>
            <option value="1100">11:00</option>
            <option value="1130">11:30</option>
            <option value="1200">12:00</option>
            <option value="1230">12:30</option>
            <option value="1300">13:00</option>
            <option value="1330">13:30</option>
            <option value="1400">14:00</option>
            <option value="1430">14:30</option>
            <option value="1500">15:00</option>
            <option value="1530">15:30</option>
            <option value="1600">16:00</option>
            <option value="1630">16:30</option>
            <option value="1700">17:00</option>
            <option value="1730">17:30</option>
            <option value="1800">18:00</option>
            <option value="1830">18:30</option>
            <option value="1900">19:00</option>
            <option value="1930">19:30</option>
            <option value="2000">20:00</option>
            <option value="2030">20:30</option>
            <option value="2100">21:00</option>
            <option value="2130">21:30</option>
            <option value="2200">22:00</option>
        </select></a> <br>
        <br>
        <br>
        <br>

```

```
<p><input type="submit" value="ΑΠΟΣΤΟΛΗ ΠΑΡΑΓΓΕΛΙΑΣ"> <input type="Reset" value="ΚΑΘΑΡΙΣΜΟΣ"></p>
```

```
</form> .....
```

Στην περίπτωση της ώρας προτιμήθηκε να εμφανίζονται συγκεκριμένες ώρες παράδοσης της παραγγελίας στον χρήστη χρησιμοποιώντας την ανάλογη select

### 4.3. Περιγραφή Των Servlets και Java Class

Πολλά στοιχεία από τα παρακάτω servlet που θα αναλυθούν είναι κοινά. Για παράδειγμα η εισαγωγή συγκεκριμένων βιβλιοθηκών όπως :

```
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
η χρησιμοποίηση των μεθόδων doGet() και doPost()
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
.
..
}
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
.
..
}
```

Επίσης όταν από ένα servlet κριθεί αναγκαία η σύνδεση με τη βάση, η σύνδεση αυτή επιτυγχάνεται εντός της μεθόδου init() με τη κλήση της μεθόδου getConnection() του αντικειμένου MySqlConnection της βοηθητικής τάξης MySqlConnection . Τα οποία αναλύονται στη σελίδα 44.

Τέλος, σε όλα τα servlet καλείται η μέθοδος getSession() του αντικειμένου request το οποίο είναι τύπου HttpServletRequest. Επειδή το HttpSession κατά την πρώτη κλήση δεν υπάρχει, δημιουργείται.

```
HttpSession session = request.getSession();
```

Όπως αναφέραμε στην `init()` δημιουργείται η σύνδεση με τη βάση. Αντίστοιχα στη `destroy()` η σύνδεση κλείνει (σελ. 45).

#### 4.3.1. Signin Servlet

Το `signin_costumer servlet` έχει ως στόχο την αποθήκευση των στοιχείων που εισάγει ο χρήστης, στην βάση δεδομένων. Η εισαγωγή των στοιχείων γίνεται στην φόρμα της σελίδας `SIGNIN.html`

Επειδή τα στοιχεία στέλνονται μέσω φόρμας χρησιμοποιούμε την μέθοδο `doPost`. Με την μέθοδο `request.getParameter("όνομα πεδίου")` παίρνουμε τις τιμές των πεδίων της φόρμας τις οποίες αποθηκεύουμε σε μεταβλητές τύπου `String` μιας και η συγκεκριμένη μέθοδο επιστρέφει τιμές τύπου `String`, εκτός του κωδικού του χρήστη το οποίο εισάγεται αυτόματα στη βάση και είναι τύπου `integer` και για αυτό το λόγο δηλώνεται ως δημόσιο μέλος: `public int cosID;`

Σε περίπτωση αποτυχίας εμφανίζεται σελίδα με το ανάλογο μήνυμα, όπως και στην περίπτωση επιτυχής εισαγωγής εμφανίζονται τα στοιχεία του νέου χρήστη καθώς και links για 'Νέα Εγγραφή' και 'Επιστροφή στην αρχική σελίδα'

```
// Members
```

```
public int cosID;
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)  
throws ServletException, IOException {
```

```
response.setContentType("text/html;charset=UTF-8");  
PrintWriter out=response.getWriter();
```

```
String cosname=request.getParameter("username");  
String pas1=request.getParameter("password");  
String fname=request.getParameter("fullname");  
String mail=request.getParameter("email");  
String addre=request.getParameter("address");
```

```
if(con!=null) {  
try{
```

```
Statement stmt=con.createStatement();
```

```

        int updateCount=stmt.executeUpdate("INSERT INTO costumers
VALUES("+cosID+", '"+cosname+"', '"+pas1+"', '"+fname+"', '"+mail+"', '"+adre+"");")
;
        if (updateCount !=0) {
            out.println("<HTML><HEAD><TITLE>ΕΠΙΤΥΧΗΣ/ΑΝΕΠΙΤΥΧΗΣ
ΕΓΓΡΑΦΗ</TITLE>" +
                "<style type=text/css>" +

                " body {background-color: #6666E3}" +
                " h1 {background-color: #FCE047}" +
                " h1 {color: #000000}" +
                " h1 {text-align: center}" +
                " div {color: #000000}" +
                " tr {text-align: left}" +
                " input {align: right}" +
                " div {text-align: center}" +
                ".style8 {color: #FCE047; font-size: 12px;}" +
                ".style14 {font-size: large;font-weight: bold;color: #FCE047; }" +
                " </style></HEAD>");
            out.println("<body>");
            out.println("<h1>Η ΒΑΣΗ ΕΝΗΜΕΡΩΘΗΚΕ!!!</h1><br>");
            out.println("<p align=center class=style14>Στοιχεία Χρήστη
:</p><center><b>");
                ResultSet rs=stmt.executeQuery("SELECT * FROM costumers WHERE user-
name='"+cosname+"";");

                while(rs.next()) { //epomeni egrafi
                    String line=rs.getInt(1)+"', '"+rs.getString(2)+"', '"+rs.getString(3)+"', '"+rs.-
getString(4)+"', '"+rs.getString(5)+"', '"+rs.getString(6);
                    out.println(line+"<br></center></b>");
                    out.println("<p align=center class=style14>Ο χρήστης προστέθηκε
επιτυχώς στη Βάση Δεδομένων </p>");
                }
            } else {
                out.println("<h1>ΑΠΟΤΥΧΙΑ ΕΝΗΜΕΡΩΣΗΣ ΤΗΣ ΒΑΣΗΣ
ΔΕΔΟΜΕΝΩ</h1>");
            }
            out.println("<a align=center class=style14 href='SIGNIN.html'>"+ "Νέα
Εγγραφή</a>");
            out.println("<br>");
            out.println("<a align=center class=style14 href='Login.jsp'>"+ "<b><i>
Επιστροφή στην αρχική σελίδα</b></i></a><br>");
            out.println("<hr><p align=center class=style8>© 2009 Σχεδιασμός
Ιστοσελίδας Αφοί Αθανασίου, All Rights Reserved</p>");
            out.println("</body></html>");
        } catch (SQLException e) {
            log("SQL exception" , e);
        }
    }
} else {
    out.println("could't connect to DB eshop");
}

```



```
}  
}
```

### 4.3.2. Login Servlet

Το login\_costumer καλείτε μέσω της φόρμας της σελίδας Login.jsp, και η χρήση του είναι να μας κατευθύνει στη σελίδα του χρήστη MainPage.jsp. Αυτό επιτυγχάνεται με την ταυτοποίηση των τιμών που έχει εισάγει ο χρήστης στα πεδία 'username' και 'password' της Login.jsp με τα δεδομένα που είναι αποθηκευμένα στη βάση μας.

```
...  
public void doPost(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
    response.setContentType("text/html;charset=utf-8");  
    PrintWriter out=response.getWriter();  
    String name= request.getParameter("name");  
    String password =request.getParameter("password");  
    if(!allowUser(name,password,request)){  
        out.println("<HTML><HEAD><TITLE>ΑΝΕΠΙΒΕΒΑΙΩΤΟΣ  
ΚΩΔΙΚΟΣ</TITLE>" +  
            "<style type=text/css>" +  
  
            " body {background-color: #6666E3}" +  
            "h1 {background-color: #FCE047}" +  
            " h1 {color: #000000}" +  
            " h1 {text-align: center}" +  
            "div {color: #000000}" +  
            "tr {text-align: left}" +  
            " input {align: right}" +  
            " div {text-align: center}" +  
            ".style8 {color: #FCE047; font-size: 12px;}" +  
            ".style14 {font-size: large;font-weight: bold;color: #FCE047; }" +  
            " </style></HEAD>");  
        out.println("<body>");  
        out.println("<h1>ΤΟ ΟΝΟΜΑ ΣΑΣ ΚΑΙ Ο ΚΩΔΙΚΟΣ ΠΡΟΣΒΑΣΗΣ ΕΙΝΑΙ  
ΑΚΥΡΑ!!!</h1><br>");  
        out.println("<p align=center class=style14>Προσπαθήστε ξανά</p><center><a  
href=\"LOGIN.html\"><br> <b>κάντε κλικ εδώ</b> </a></center><br>");  
        out.println("<hr><p align=center class=style8>© 2009 Σχεδιασμός Ιστοσελίδας  
Αφοί Αθανασίου, All Rights Reserved</p>");  
        out.println("</body></html>");  
    } else{ //an einai swsta  
        HttpSession session=request.getSession();  
        session.setAttribute("login.isDone",name);  
  
        if ( cosid != null ){  
            session.setAttribute("curUser",cosid);  
        }  
        String target=(String) session.getAttribute("LOGIN.target"); //anakatefthinsi
```

```

        if(target!=null) {
            response.sendRedirect(target);
            return;
        }
        response.sendRedirect("MainPage.jsp");
    }
}
protected boolean allowUser(String user,String password, HttpServletRequest request){
    try {
        HttpSession session=request.getSession();
        Statement st = con.createStatement();

        ResultSet rs = st.executeQuery(
            "SELECT costumer_id " +
            "FROM costumers " +
            "WHERE (username = " + user + ") " +
            "AND (password = " + password + ")");
        while (rs.next()) {
            if (rs.getString("costumer_id")!=null){
                cosid = rs.getString("costumer_id");
                session.setAttribute("curUser",cosid);
                return true ;
            }
        }
    } catch (SQLException e) {
        log("SQL Exception", e);
        System.out.println("SQL Error");
    }

    return false;
}
}
}

```

Το servlet για το login του διαχειριστή είναι πιο απλό διότι ελέγχει μόνο το username και password του διαχειριστή και όταν είναι σωστά ανοίγει η σελίδα του διαχειριστή Admin.jsp.

```

...
public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=utf-8");
    PrintWriter out=response.getWriter();
    String name= request.getParameter("aname");
    String password =request.getParameter("apassword");
    if(!allowUser(name,password,request)){
        out.println("<HTML><HEAD><TITLE>ΑΝΕΠΙΒΕΒΑΙΩΤΟΣ
ΚΩΔΙΚΟΣ</TITLE>" +
            "<style type=text/css>" +

```

```

        " body {    background-color: #6666E3    }"+
        " h1 {background-color: #FCE047}"+
        " h1 {color: #000000}"+
        " h1 {text-align: center}"+
        " div {color: #000000}"+
        " tr {text-align: left}"+
        " input {align: right}"+
        " div {text-align: center}"+
        ".style8 {color: #FCE047; font-size: 12px;}"+
        ".style14 {font-size: large;font-weight: bold;color: #FCE047; }"+
        " </style></HEAD>");
    out.println("<body>");
    out.println("<h1>ΤΟ ΟΝΟΜΑ ΣΑΣ ΚΑΙ Ο ΚΩΔΙΚΟΣ ΠΡΟΣΒΑΣΗΣ ΕΙΝΑΙ
    ΑΚΥΡΑ!!!</h1><br>");
    out.println("<p align=center class=style14>Προσπαθήστε ξανά</p><center><a
    href='\"LOGIN ADMINISTRATOR.html\"'><br> <b>κάντε κλικ εδώ</b> </a></cen-
    ter><br>");
    out.println("<hr><p align=center class=style8>© 2009 Σχεδιασμός Ιστοσελίδας
    Αφοί Αθανασίου, All Rights Reserved</p>");
    out.println("</body></html>");
} else{ //an einai swsta
    HttpSession session=request.getSession();
    session.setAttribute("loginadmin.isDone",name);

    String target=(String) session.getAttribute("Admin.target"); //anakatefthinsi
    if(target!=null) {
        response.sendRedirect(target);
        return;
    }
    response.sendRedirect("Admin.jsp");
}
}
}

```

### 4.3.3. Add product Servlet

Με τη χρήση του servlet add\_admin ο διαχειριστής της εφαρμογής εισάγει νέα προϊόντα στη βάση. Οι παράμετροι που δίνει ( δηλαδή κωδικό, περιγραφή, τιμή και είδος ) θα αποθηκευτούν σε μεταβλητές τύπου String εκτός από τον κωδικό που είναι integer και αριθμείτε αυτόματα και θα ελεγχθεί η τιμή του κωδικού ώστε να μην είναι κενή. Στη συνέχεια αν δεν παρουσιαστεί εξαίρεση τα δεδομένα θα αποθηκευτούν στη βάση.

```

...
// Members

```

```

public    int pid;

```

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out=response.getWriter();

    String pde=request.getParameter("prde");
    float pp=Float.parseFloat(request.getParameter("prp"));
    String pkat=request.getParameter("prkat");

    if (con!=null){

        try {
            Statement stmt = con.createStatement();

            int updateCount=stmt.executeUpdate("INSERT INTO product
VALUES("+pid+", "+pde+", "+pp+", "+pkat+"");");
            if (updateCount !=0) {
                out.println("<HTML><HEAD><TITLE>ΕΠΙΤΥΧΗΣ/ΑΝΕΠΙΤΥΧΗΣ
ΠΡΟΣΘΗΚΗ</TITLE>" +
                    "<style type=text/css>" +
                    " body {background-color: #6666E3}" +
                    " h1 {background-color: #FCE047}" +
                    " h1 {color: #000000}" +
                    " h1 {text-align: center}" +
                    " div {color: #000000}" +
                    " tr {text-align: left}" +
                    " input {align: right}" +
                    " div {text-align: center}" +
                    ".style8 {color: #FCE047; font-size: 12px;}" +
                    ".style14 {font-size: large;font-weight: bold;color: #FCE047;}" +
                    " </style></HEAD>");
                out.println("<body>");
                out.println("<h1>Η ΒΑΣΗ ΕΝΗΜΕΡΩΘΗΚΕ!!!</h1><br>");
                out.println("<p align=center class=style14> Στοιχεία Προϊόντος :</p>>");
                ResultSet rs=stmt.executeQuery("SELECT * FROM product WHERE descrip-
tion="+pde+"");
                while(rs.next()) { //epomeni egrafi
                    String line=rs.getInt(1)+" "+rs.getString(2)+" "+rs.getFloat(3)+" "+rs.-
getString(4);
                    out.println(line+"<br>");
                    out.println("<p align=center class=style14>Η ΠΡΟΣΘΗΚΗ ΕΓΙΝΕ
ΕΠΙΤΥΧΩΣ </p>");
                }
            } else {
                out.println("Η ΠΡΟΣΘΗΚΗ ΔΕΝ ΕΓΙΝΕ <br>");
            }
        }
    }
}

```

```

        out.println("<a align=center class=style14 href='Admin.jsp'>"+ "Νέα
Προσθήκη</a>");
        out.println("<br>");
        out.println("<a align=center class=style14 href='LOGIN.html'>"+ "<b><i>
ΕΞΟΛΟΣ</b></i></a>");
        out.println("<hr><p align=center class=style8>© 2009 Σχεδιασμός
Ιστοσελίδας Αφοί Αθανασίου, All Rights Reserved</p>");
        out.println("</body></html>");

    } catch(SQLException ex) {
        log("SQL εξαίρεση", ex);
    }
}

else
    out.println("Δεν μπόρεσα να συνδεθώ με την ΒΔ eshop στην MySQL");

}

}

```

#### 4.3.4. Delete product Servlet

Εδώ θα εισάγουμε id του προϊόντος που θέλουμε να διαγράψουμε και θα διαγραφεί το αντίστοιχο προϊόν.

```

...
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out=response.getWriter();

    int delpr=Integer.parseInt(request.getParameter("delID"));

    if (con!=null){

        try {
            Statement stmt = con.createStatement();
            int updateCount=stmt.executeUpdate("DELETE FROM product WHERE
product_id='"+delpr+"'");
            if (updateCount !=0) {
                out.println("<HTML><HEAD><TITLE>ΕΠΙΤΥΧΗΣ/ΑΝΕΠΙΤΥΧΗΣ
ΔΙΑΓΡΑΦΗ ΠΡΟΪΟΝΤΟΣ</TITLE>"+
                    "<style type=text/css>"+
                    " body {background-color: #6666E3}"+

```

```

        "h1 {background-color: #FCE047}" +
        " h1 {color: #000000}" +
        " h1 {text-align: center}" +
        "div {color: #000000}" +
        "tr {text-align: left}" +
        " input {align: right}" +
        " div {text-align: center}" +
        ".style8 {color: #FCE047; font-size: 12px;}" +
        ".style14 {font-size: large;font-weight: bold;color: #FCE047; }" +
        " </style></HEAD>");
    out.println("<body>");
    out.println("<h1>Η ΒΑΣΗ ΕΝΗΜΕΡΩΘΗΚΕ!!!</h1><br>");
    out.println("<p align=center class=style14> Το προϊόν διαγράφηκε:</p>");
    /* ResultSet rs=stmt.executeQuery("SELECT * FROM product WHERE de-
description='"+pde+"'");
    while(rs.next()) { //epomeni egrafi
        String line=rs.getInt(1)+" "+rs.getString(2)+" "+rs.getFloat(3)+" "+rs.-
getString(4);
        out.println(line+"<br>");
        out.println("<p align=center class=style14>Η ΠΡΟΣΘΗΚΗ ΕΓΙΝΕ
ΕΠΙΤΥΧΩΣ </p>");
        }*/
    } else {
        out.println("Η ΔΙΑΓΡΑΦΗ ΔΕΝ ΕΓΙΝΕ<br>");
    }
    out.println("<a href='Admin.jsp'>"+ "<p align=center class=style14 >Νέα
Διαγραφή</p></a>");
    out.println("<br>");
    out.println("<a align=center class=style14 href='LOGIN.html'>"+ "<p
align=center class=style14 > ΕΞΟΔΟΣ</p></a>");
    out.println("<hr><p align=center class=style8>© 2009 Σχεδιασμός
Ιστοσελίδας Αφοί Αθανασίου, All Rights Reserved</p>");
    out.println("</body></html>");

    } catch(SQLException ex) {
        log("SQL εξαίρεση", ex);
    }
}

else
    out.println("Δεν μπόρεσα να συνδεθώ με την ΒΔ eshop στην MySQL");

}

}

```

### 4.3.5. Order Servlet

Μέσω αυτού του servlet η παραγγελία του πελάτη καταχωρείται στη βάση δεδομένων μας.

```
....
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out=response.getWriter();

    String cost = request.getParameter("cost");
    String addr=request.getParameter("ADDRESS");
    String d=request.getParameter("DATE");
    String t=request.getParameter("TIME");

    if (con!=null){

        try {
            Statement stmt = con.createStatement();
            HttpSession session=request.getSession();
            String usName=(String) session.getAttribute("login.isDone");
            String sessionId = request.getSession().getId();
            //=====
            String orID = null;
            ResultSet rs=stmt.executeQuery("SELECT max(id) FROM orders");
            while(rs.next()) { //epomeni egrafi
                int id=rs.getInt(1);
                orID = id+1+"";
            }
            //=====
            int updateCount=stmt.executeUpdate("INSERT INTO orders
VALUES("+orID+", ""+usName+", ""+cost+", ""+addr+", ""+d+", ""+t+", ""+sessionId+"")
;");
            if (updateCount !=0) {
                out.println("<HTML><HEAD><TITLE>ΕΠΙΤΥΧΗΣ/ΑΝΕΠΙΤΥΧΗΣ
ΠΑΡΑΓΓΕΛΙΑ</TITLE>" +
                "<style type=text/css>" +

                " body {background-color: #6666E3}" +
                "h1 {background-color: #FCE047}" +
                " h1 {color: #000000}" +
                " h1 {text-align: center}" +
                "div {color: #000000}" +
                "tr {text-align: left}" +
```

```

        " input {align: right}"+
        " div {text-align: center}"+
        ".style8 {color: #FCE047; font-size: 12px;}"+
        ".style14 {font-size: large;font-weight: bold;color: #FCE047; }"+
        " </style></HEAD>");
    out.println("<body>");
    out.println("<h1>Η ΒΑΣΗ ΕΝΗΜΕΡΩΘΗΚΕ!!!</h1><br>");
    out.println("<p align=center class=style14> Στοιχεία Παραγγελίας
: </p><center><b>");
        rs=stmt.executeQuery("SELECT * FROM orders WHERE username='"+us-
Name+"' and sessionId='"+sessionId+"'");
        while(rs.next()) { //epomeni egrafi
            String line=rs.getInt(1)+" "+rs.getString(2)+" "+rs.getFloat(3)+" "+rs.-
getString(4)+" "+rs.getString(5)+" "+rs.getString(6);
            out.println(line+"<br></center></b>");
            out.println("<p align=center class=style14>Η παραγγελία σας
καταχωρήθηκε επιτυχώς </p>");
        }

        //clearing the cart
        Hashtable cart_items =
        (Hashtable)request.getSession().getAttribute("cart_list");
        cart_items.clear();

    } else {
        out.println("<p align=center class=style14>Η ΠΑΡΑΓΓΕΛΙΑ ΣΑΣ ΔΕΝ
ΜΠΟΡΕΣΕ ΝΑ ΚΑΤΑΧΩΡΗΘΕΙ</p> <br>");
    }
    out.println("<a href='ProductCatalog.jsp'>"+ "<p align=center class=style14
>Νέα Παραγγελία</p></a>");
    out.println("<br>");
    out.println("<a href='MainPage.jsp'>"+ "<p align=center class=style14
>Επιστροφή στην αρχική σελίδα</p></a>");
    out.println("<br>");
    out.println("<a href='Login.jsp?logout=true'>"+ "<p align=center
class=style14 > ΕΞΟΔΟΣ</p></a>");
    out.println("<hr><p align=center class=style8>© 2009 Σχεδιασμός
Ιστοσελίδας Αφοί Αθανασίου, All Rights Reserved</p>");
    out.println("</body></html>");
    } catch(SQLException e) {
        log("SQL exception" , e);
    }
    } else {
        out.println("could't connect to DB eshop");
    }
}

```



#### 4.3.6. Κλάση Shopping Cart

Με τη βοηθητική αυτή τάξη μπορούν να γίνουν ενέργειες όπως της προσθήκης των προϊόντων στο καλάθι, το συνολικό κόστος της παραγγελίας και το πλήθος των προϊόντων της παραγγελίας.

```
public class ShoppingCart {

    protected Hashtable items = new Hashtable();

    public void addItem(String itemId, String desc, float price, int quantity) {
        String[] item = {itemId, desc, Float.toString(price), Integer.toString(quantity)};

        if (items.containsKey(itemId)) {

            String[] tmpItem = (String[])items.get(itemId);
            int tmpQuant = Integer.parseInt(tmpItem[3]);
            quantity += tmpQuant;
            tmpItem[3] = Integer.toString(quantity);
        } else {

            items.put(itemId, item);
        }
    }

    public void removeItem(String itemId) {
        if (items.containsKey(itemId)) {
            items.remove(itemId);
        }
    }

    public void updateQuantity(String itemId, int quantity) {
        if (items.containsKey(itemId)) {
            String[] tmpItem = (String[])items.get(itemId);
            tmpItem[3] = Integer.toString(quantity);
        }
    }

    public Enumeration getEnumeration() {
        return items.elements();
    }

    public float getCost() {

        Enumeration e = items.elements();
        String[] tmpItem;
        float totalCost = 0.0f;
    }
}
```

```

while (e.hasMoreElements()) {
    tmpItem = (String[])e.nextElement();
    totalCost += (Integer.parseInt(tmpItem[3]) * Float.parseFloat(tmpItem[2]));
}
return totalCost;
}

public int getNumOfItems() {
    Enumeration e = items.elements();
    String[] tmpItem;
    int numOfItems = 0;
    while (e.hasMoreElements()) {
        tmpItem = (String[])e.nextElement();
        numOfItems += Integer.parseInt(tmpItem[3]);
    }
    return numOfItems;
}

```

#### 4.3.7. Κλάση SetCharacterEncodingFilter

Η βοηθητική αυτή τάξη του πακέτου filters, χρησιμοποιείται για την ανάγνωση και εισαγωγή των ελληνικών χαρακτήρων στη βάση, παρακάτω εμφανίζεται ολόκληρος ο κώδικας της κλάσης.

```

package filters;

import java.io.IOException;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.UnavailableException;

public class SetCharacterEncodingFilter implements Filter {
    public void destroy() {

        this.encoding = null;
        this.filterConfig = null;
    }
    protected String selectEncoding(ServletRequest request) {
        return (this.encoding);
    }
}

```

Για να λειτουργήσει σωστά η παραπάνω κλάση φιλτραρίσματος θα πρέπει να εισαχθεί στην αρχή του αρχείου web.xml ο παρακάτω κώδικας:

```
<filter>
  <filter-name>Set Character Encoding</filter-name>
  <filter-class>filters.SetCharacterEncodingFilter</filter-class>
</init-param>
  <param-name>encoding</param-name>
  <param-value>UTF-8</param-value>
</init-param>
</filter>
<filter-mapping>
  <filter-name>Set Character Encoding</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

## 4.4. Περιγραφή Των JavaScripts

Το παρακάτω script εμφανίζεται σε όλες τις σελίδες που έχει πρόσβαση ο χρήστης και μπορούμε απλά να πούμε πως είναι ένα διακοσμητικό στοιχείο και στοχεύει περισσότερο στο marketing της ιστοσελίδας.

```
<SCRIPT LANGUAGE="JavaScript">
  arr=new Array();
  arr[0]="Στα Super Market μας, μπορείτε να βρείτε όλα τα προϊόντα και στις
καλύτερες τιμές!";
  arr[1]="Τα PLUS + εγγυώνται για την ποιότητα και την φρεσκάδα των
προϊόντων τους!";
  arr[2]="Χρησιμοποιήστε τον κατάλογο των προϊόντων και γεμίστε το καρότσι
σας..."
  arr[3]="Μη ξεχάσετε να ελέγξετε την παραγγελίας σας.";
  arr[4]="Η εξυπηρέτησή σας, είναι προτεραιότητά μας!";
  arr[5]="Το On-Line KAT' OIKON κατάστημα λειτουργεί όλο το 24ωρο";
  arr[6]="ΣΑΣ ΕΥΧΑΡΙΣΤΟΥΜΕ ΠΟΥ ΕΠΙΣΚΕΦΤΗΚΑΤΕ ΤΟ ON-LINE
ΚΑΤΑΣΤΗΜΑ PLUS +";
  pause = 400
```

```

scroll_speed = 150;
i=0;
setTimeout("begin()",400);
function begin()
{
news.innerHTML=arr[i];
news.style.paddingTop=news.offsetHeight - 20;
i++
if (i==arr.length) i=0;
setTimeout("begin2()",1000);
}
function begin2()
{
a=parseInt(news.style.paddingTop)-2;
if (a < 5) {clearTimeout("t"); setTimeout("begin()",pause); return false};
news.style.paddingTop=a;
t=setTimeout("begin2()",scroll_speed);
}
</script>

```

#### 4.4.1. javascript ParseRegCost

Σε αυτή τη συνάρτηση γίνονται οι έλεγχοι κατά την καταχώριση των στοιχείων ενός νέου χρήστη, η οποία αποτρέπει την εισαγωγή διαφορετικών στοιχείων από τα ζητούμενα στα πεδία του ονόματος χρήστη και του κωδικού πρόσβασης και αντιστοιχεί στην σελίδα SIGNIN.html

```

<script language="JavaScript" type="text/JavaScript">

    function ParseRegCost(f) {

        if (f.username.value.length<3 || f.username.value.length>16) {
            alert('Το όνομα χρήστη πρέπει να είναι μεταξύ 3 και 16 χαρακτήρων!');
            return false;
        }

        if (f.password.value.length<6 || f.password2.value.length<6) {
            alert('Ο κωδικός πρόσβασης πρέπει να είναι μεταξύ 6 και 16 χαρακτήρων!');
            return false;
        }

        if (f.password.value!=f.password2.value) {
            alert('Ο κωδικός πρόσβασης δεν ταιριάζει! Παρακαλώ δοκιμάστε ξανά!');
            return false;
        }

    }

</script>

```

#### 4.4.2. Javascript Product Catalog

Εδώ έχουνε περαστεί οι συναρτήσεις στη σελίδα ProductCatalog.jsp οι οποίες αντιστοιχούνε στην εισαγωγή των προϊόντων στο καλάθι αγορών, για να εμφανιστούνε στη συνέχεια στη σελίδα ShoppingCart. Σαν παράδειγμα εμφανίζεται το script της κατηγορίας του σπωροπωλείου, με ανάλογο τρόπο γίνονται και τα script για της υπόλοιπες κατηγορίες του καταλόγου των προϊόντων.

```
function opor(){
```

```

    var prId=document.getElementById('category1');
    var select_list_selected_index = prId.selectedIndex;
    var text = prId.options[select_list_selected_index].text;

```

```

    var quant = document.getElementById('quantity1');
    var desc = text.substring(0,text.indexOf(','));
    var price = text.substring(text.indexOf(', ')+2, text.indexOf(' €'));

```

```

    if (select_list_selected_index == 0 ){
        alert('Επίλεξε Προϊόν');
        return;
    }

```

```

    }

    if ( quant.value == "" ){
        alert('Δώσε Ποσότητα');
        return;
    }

    alert("Ψώνισες απο το Οπωροπωλείο");

    document.getElementById('product_id_opor').value=prId.value;
    document.getElementById('des_opor').value=desc;
    document.getElementById('pr_opor').value=price;
}

```

#### 4.4.3. Javascripts deleteRow

Το συγκεκριμένο java script διαγράφει ένα προϊόν από το καλάθι αγορών πριν την ολοκλήρωση της παραγγελίας στη σελίδα ShoppingCart.jsp.

```

function deleteRow(index){

document.getElementById('wichRow').value=index;
}

```

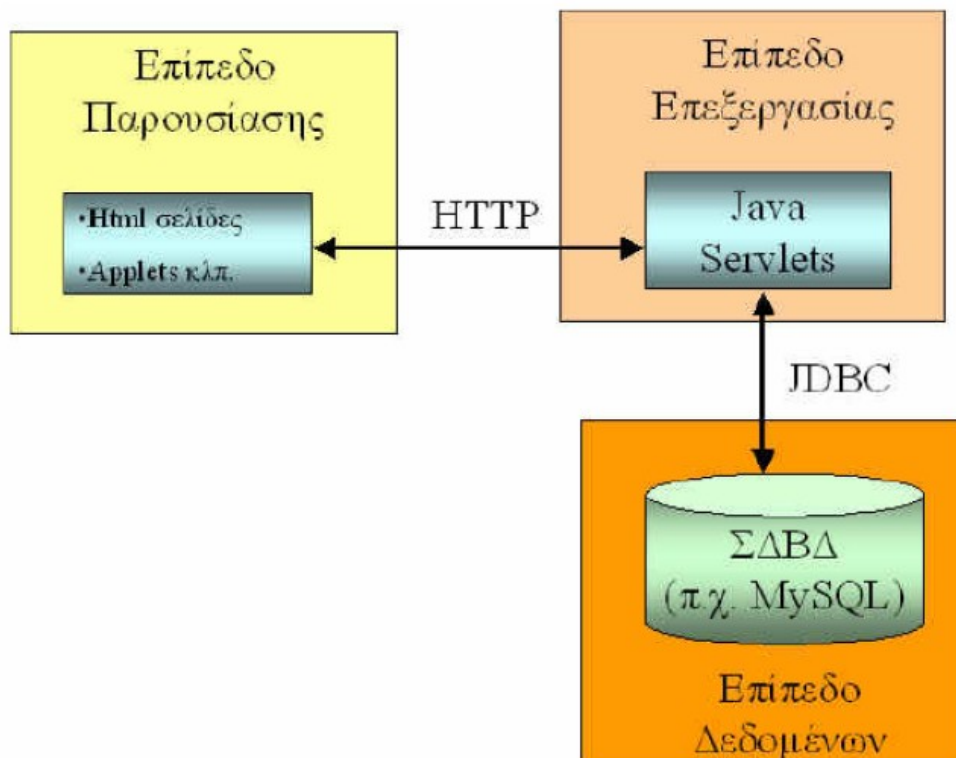
#### 4.4.4. Javascripts PayPal & PayCash

Οι παρακάτω συναρτήσεις είναι δύο απλές συναρτήσεις οι οποίες μας μεταβαίνουν στην σελίδα του paypal και στην σελίδα PayCash, αντιστοίχως. Καλούνται μέσω των δύο submit buttons στην σελίδα ShoppingCart.jsp.

```
function PayPal(p) {  
  
window.open('https://www.paypal.com/gr');  
}  
  
function PayCash(c) {  
  
window.open('PayCash.jsp');  
}
```

## ΚΕΦΑΛΑΙΟ 5<sup>ο</sup>: ΣΥΜΠΕΡΑΣΜΑΤΑ

Μετά τη ενδεκάμηνη ενασχόλησή μας με την πτυχιακή μας μελέτη, αποκτήσαμε μεγάλο εύρος γνώσεων σχετικά με την κατασκευή δυναμικών web εφαρμογών και συγκεκριμένα με χρήση Java τεχνολογιών. Την λειτουργία – φιλοσοφία του ακόλουθου μοντέλου την κατανοήσαμε πλήρως και πήραμε μια ιδέα από τις δυνατότητες που προσφέρουν τα JavaServlets και οι JSP σελίδες.



Πηγή: Γ.Κακαρόντζας, "Προγραμματισμός Internet με την τεχνολογία των JavaServlets."

Συγκεκριμένα, με τη χρήση του session πετύχαμε την 'σύνδεση' των σελίδων οι οποίες, λόγω του γεγονότος ότι το HTTP πρωτόκολλο είναι stateless, συμπεριφέρονται ως ανεξάρτητες. Εκτός από την χρήση του session για την επικοινωνία σελίδων και Servlets χρησιμοποιήσαμε και τα hidden πεδία που αποδείχτηκαν εξίσου λειτουργικά. Εν κατακλείδι, μπορούμε να πούμε ότι οι τεχνολογίες που χρησιμοποιήσαμε ανταποκρίθηκαν στις αρχικές μας απαιτήσεις. Οι υπηρεσίες που προσφέρει η εφαρμογή μας υλοποιήθηκαν χάρη στις πολλές δυνατότητες (κατάλληλα 'εργαλεία', ευελιξία) που διαθέτουν. Αντίθετα υστέρησαν σε σχεδιαστική ευκολία αφού θα ήταν προτιμότερη η οπτική σχεδίαση της



εφαρμογής με την αποφυγή δηλαδή της συγγραφής HTML κώδικα. Το τελευταίο βέβαια μπορεί να ξεπεραστεί με χρήση επιπλέον Java τεχνολογιών ή και προγραμμάτων. Πιστεύουμε λοιπόν ότι αξίζει να τις χρησιμοποιήσει κάποιος για να υλοποιήσει web εφαρμογές.

Ως πλεονεκτήματα για τη χρήση τεχνολογιών Java μπορούμε να αναφέρουμε τα κάτωθι :

- Χρησιμοποιούν μια πολλή δυνατή γλώσσα προγραμματισμού, τη Java
- Για τη χρήση τους απαιτείται μηδενικό κόστος
- Μέσω του JDBC μπορούν να επικοινωνούν με Βάσεις Δεδομένων
- Είναι ανεξάρτητες από Λειτουργικά Συστήματα αφού εκτελούνται από τη Java Virtual Machine

Μειονεκτήματα είναι :

- Το επίπεδο δυσκολίας εκμάθησης της Java, αφού υπάρχουν πολύ πιο εύκολες γλώσσες που υλοποιούν Διαδικτυακές εφαρμογές
- Η μη υποστήριξη τους από τους servers των περισσότερων εταιρειών web hosting της αγοράς

# **ΒΙΒΛΙΟΓΡΑΦΙΑ**

## **Ξένη**

1. D. Quah "Weightless Economy in Economic Development", 2001, Oxford University Press.

## **Ελληνική - Μεταφρασμένη**

2. E. Kastro, "Εισαγωγή στην HTML και τα CSS.", 2009, εκδ. Κλειδάριθμος.
3. E. Kramer, "Οπτικός οδηγός της HTML 4.", 2001, εκδ. Γκιούρδας,
4. J. C. Meloni, "Μάθετε PHP, MySQL και APACHE σε 24 ώρες.", 2003, εκδ. M. Γκιούρδας.
5. K. Binder, C. Rose, "Μάθετε το Adobe Photoshop CS3 σε 24 ώρες", 2007, εκδ. M. Γκιούρδας.
6. M. Fowler, K. Scott "Εισαγωγή στη uml.", 2006, εκδ. Κλειδάριθμος
7. M. Webb, "Πλήρες εγχειρίδιο της JavaScript.", 2001, Γκιούρδας M.
8. Γ. Γκαράνη, "Εργαστήριο - Ειδικά θέματα Βάσεων Δεδομένων." 2003, Τμήμα Τεχν. Πληροφορικής & Τηλ/νιών ΤΕΙ Λάρισας
9. Γ. Κακαρόντζας, "Προγραμματισμός Internet με την τεχνολογία των JSP." 2003, Τμήμα Τεχν. Πληροφορικής & Τηλ/νιών ΤΕΙ Λάρισας
10. Γ. Κακαρόντζας, "Προγραμματισμός Internet με την τεχνολογία των JavaServlets." 2003, Τμήμα Τεχν. Πληροφορικής & Τηλ/νιών ΤΕΙ Λάρισας
11. Γ. Κακαρόντζας, "Εργαστήρια τεχνολογίας λογισμικού." 2003, Τμήμα Τεχν. Πληροφορικής & Τηλ/νιών ΤΕΙ Λάρισας
12. Γ. Λιακέας, "Η γλώσσα JavaScript για δυναμικές σελίδες στον Παγκόσμιο Ιστό." 2002, εκδ. Κλειδάριθμος
13. Γ. Σιώμοκος & Ι. Τσιάμης, "Στρατηγικό Ηλεκτρονικό Μάρκετινγκ", 2004, εκδ. Σταμούλη
14. Ν. Γεωργόπουλος, "Ηλεκτρονικό Επιχειρείν, Προγραμματισμός & Σχεδίαση.", 2005, εκδ. Μπένου Ευγ

## Διαδίκτυο

15. M. Hall, Larry Brown, "Core Servlets and JavaServer Pages.", Free Online Version, ανάκτηση δεδομένων από: <http://pdf.coreservlets.com/>
16. «Επιστημονικό Μάρκετινγκ», 2006, Ανάκτηση δεδομένων από <http://www.morax.gr/index.php?issueYear=2006&issueMonth=9>,
17. Ανάκτηση δεδομένων από: <http://java.sun.com/j2ee/index.jsp>
18. Ανάκτηση δεδομένων από: [http://www.jspinsider.com/content/jsp/javamail/jspjavamail\\_6.jsp](http://www.jspinsider.com/content/jsp/javamail/jspjavamail_6.jsp)
19. Ανάκτηση δεδομένων από: <http://java.ittoolbox.com/code/d.asp?d=2378&a=s>
20. Ανάκτηση δεδομένων από: <http://javaalmanac.com/egs/javax.mail/SendApp.html>
21. Ανάκτηση δεδομένων από: [http://www.unix.org.ua/oreilly/java-ent/servlet/ch13\\_02.htm](http://www.unix.org.ua/oreilly/java-ent/servlet/ch13_02.htm)
22. Ανάκτηση δεδομένων από: [http://www.unix.org.ua/oreilly/java-ent/servlet/ch04\\_04.htm#ch04-34777](http://www.unix.org.ua/oreilly/java-ent/servlet/ch04_04.htm#ch04-34777)
23. Ανάκτηση δεδομένων από: <http://rollerjm.free.fr/pro/Struts11.html>
24. Ανάκτηση δεδομένων από: <http://www.go-online.gr>