

---

**Εφαρμογή ηλεκτρονικού εμπορίου με  
χρήση jsp, java servlet, apache tomcat  
και MySQL .**

---

<b>Τίτλος</b>	Εφαρμογή ηλεκτρονικού εμπορίου με χρήση jsp, java servlet, apache tomcat και MySQL.
<b>Στοιχεία Συντάκτη</b>	Ελένογλου Ευστάθιος, Μαρκόπουλος Παναγιώτης
<b>Εισηγητής</b>	Αλεξανδρής Γεώργιος

Σχολή: Σ.Δ.Ο

Τμήμα: Ε.Π.Δ.Ο

Μεβολόγιο: 2006

# Πίνακας Περιεχομένων

<b>ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....</b>	<b>5</b>
<b>1. ΕΙΣΑΓΩΓΗ.....</b>	<b>6</b>
JSP ΚΑΙ WEB APPLICATION DEVELOPMENT.....	6
HTML FORMS.....	7
ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ.....	9
SQL.....	12
JDBC.....	13
<b>2. SERVLETS, JAVASERVER PAGES ΚΑΙ JAVABEANS.....</b>	<b>16</b>
ΤΙ ΕΙΝΑΙ ΕΝΑ SERVLET;.....	16
ΚΥΚΛΟΣ ΖΩΗΣ ΕΝΟΣ SERVLET.....	17
ΤΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΩΝ SERVLET.....	18
JAVA SERVER PAGE (JSP).....	19
ΤΙ ΕΙΝΑΙ ΕΝΑ JAVA BEAN.....	21
JAVA SERVER PAGE ΚΑΙ JAVA BEANS.....	23
MYSQL.....	24
<b>3. ΟΔΗΓΟΣ ΓΙΑ ΤΗΝ ΕΓΚΑΤΑΣΤΑΣΗ ΛΟΓΙΣΜΙΚΟΥ.....</b>	<b>26</b>
ΕΙΣΑΓΩΓΗ.....	26
ΕΓΚΑΤΑΣΤΑΣΗ APACHE.....	27
ΕΓΚΑΤΑΣΤΑΣΗ PHP.....	33
ΕΓΚΑΤΑΣΤΑΣΗ JSP.....	37
ΕΓΚΑΤΑΣΤΑΣΗ MYSQL.....	51
ΕΓΚΑΤΑΣΤΑΣΗ ODBC.....	62
ΕΓΚΑΤΑΣΤΑΣΗ PHP MYADMIN.....	68
<b>4. JUMP START CYCLES.....</b>	<b>69</b>
<b>5. ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>100</b>

## 1. Εισαγωγή.

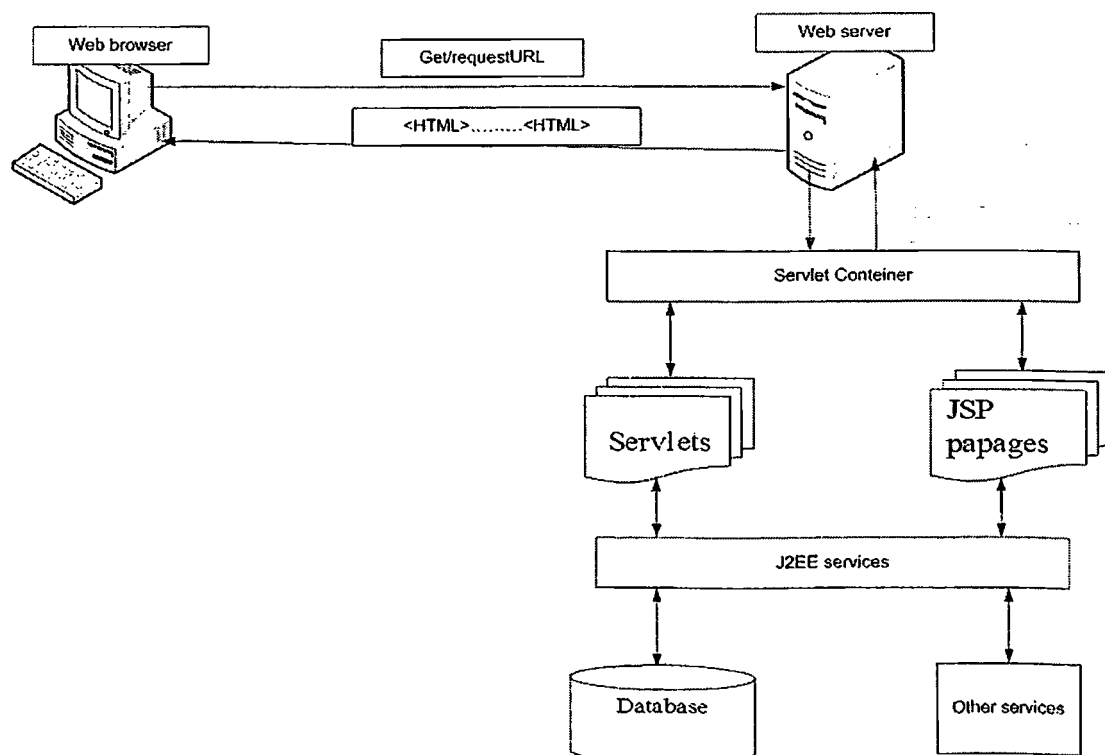
### JSP και Web Application development

---

Σήμερα, οι εφαρμογές Web επεκτείνονται στο internet και των επιχειρήσεων εξερχόμενων και εισερχόμενων. Έχουν τη δυνατότητα να αυξήσουν την παραγωγικότητα και την αλλαγή του τρόπου με τον οποίο οι επιχειρήσεις, μικρές και μεγάλες, λειτουργούν. Ο Server-side web application development είναι παλιός σε σχέση με το internet. Για χρόνια, οι υπεύθυνοι για την ανάπτυξη εφαρμογών στο internet χρησιμοποιούσαν άλλα εργαλεία όπως η CGI (Common Gateway Interface) για να παρέμβουν στη λογική ενός Web server. Με αποτέλεσμα να χρησιμοποιούν συνήθως περισσότερους πόρους από τη μνήμη και τον επεξεργαστή επίσης ήταν δύσκολο από το πρόγραμμα να τα αποκωδικοποιήσει. Μια σημαντική βελτίωση ήρθε με την απελευθέρωση του Java Servlet το 1997 και ακολούθησε η απελευθέρωση της JSP. Τα Java Servlet και η JSP είναι αλληλοεξαρτώμενα προγράμματα.

Τα Java Servlet και η JSP φέρνουν την πλήρες δυνατότητες της Java στον κεντρικό υπολογιστή δικτύου, στις βάσεις δεδομένων, στη πρόσβαση στο internet, και σε πολύπλοκες διαδικασίες με διαφορετικά πρότυπα επεξεργασίας. Java Servlet και JSP διαχειρίζονται μαζί παραμένουν στη μνήμη και χρησιμοποιούνται αποτελεσματικά εκεί που υπάρχουν πολλαπλές αιτήσεις υπηρεσιών ταυτόχρονα.

Στο σχήμα παρουσιάζετε, πως με Java Servlet και JSP μπορούν να χρησιμοποιηθούν μαζί με την Java 2 Enterprise Edition (J2EE)



## HTML Forms

---

Αυτό το πρόγραμμα θα χρησιμοποιήσει JSP ως κύρια γλώσσα ανάπτυξης. Η JSP ενσωματώνεται μέσα στην HTML. Ας δούμε λίγο τη μορφή της HTML. Τι είναι HTML; HTML (HyperText Markup Language) είναι ένα είδος περιγραφικής γλώσσας. Το HTML χρησιμοποιείται για να γράψουμε ένα έγγραφο για έναν browser και δίνει επίσης τη δυνατότητα στο σχεδιαστή ιστοσελίδων την ευκαιρία να ανακτήσει εισαγωγές από τους ανθρώπους που διαβάζουν την ιστοσελίδα του. Η κοινοπραξία World Wide Web (W3C) διακηρύττει για την HTML το ρητό "για να δημοσιεύσουμε τις πληροφορίες, μας χρειάζεται μια παγκοσμίως κατανοητή γλώσσα. Ένα είδος μητρικής γλώσσας που όλοι οι υπολογιστές να μπορούν να την καταλάβουν. Η γλώσσα αυτή είναι η HTML."

Το HTML είναι η δημοφιλέστερη γλώσσα στο World Wide Web. Η νεώτερη έκδοσή είναι η 4,0. Στην HTML, χρησιμοποιούμε τις ετικέτες(tags) για να ελέγξουμε το κείμενο που θα είναι ορατό και για τη παραγωγή των συνδέσμων υπέρ-κειμένου για να το συνδέσουμε με ένα άλλο έγγραφο HTML. Χρησιμοποιούμε HTML forms για να εξετάσουμε τις εισαγωγές των χρηστών. HTML forms περιλαμβάνουν τα στοιχεία διεπαφών που αφήνουν τους χρήστες να εισαγάγουν το κείμενο και να κάνουν τις επιλογές τους. Όλα τα στοιχεία διεπαφών στην HTML πρέπει να καθοριστούν με ετικέτες < ΜΟΡΦΗ > και </ΜΟΡΦΗ >.

Υπάρχουν τρεις σημαντικές ιδιότητες μέσα στα στοιχεία form. Ο πρώτος είναι η "ACTION", η οποία καθορίζει ένα Uniform Resource Identifier (URI) της HTTP όταν υποβάλλονται οι πληροφορίες. Η δεύτερη είναι η "METHOD", η οποία καθορίζει ένα HTTP GET ή POST method. Ένα HTTP GET ή POST request ερμηνεύεται συνήθως από το Web server ως αίτημα να ανακτηθεί το έγγραφο που ονομάζεται στο URL.

Η διαφορά μεταξύ GET και POST methods όταν χρησιμοποιούνται σε μια HTML form είναι πώς παρέχουν τα δεδομένα εισόδου στον κεντρικό υπολογιστή για να τα επεξεργαστεί.

(1) GET: Οι τιμές μορφής επισυνάπτονται στο URL ως σειρά ερώτησης.

(2) POST: Οι τιμές μορφής παρέχονται στο ρεύμα εισαγωγής.

Η POST HTTP είναι συνήθως καλύτερη επιλογή για τη μέθοδο αιτήματος. Τελευταία ιδιότητα είναι η "ENCTYPE". Αυτή η μέθοδος κωδικοποιεί τις τιμές σε ένα ρεύμα στοιχείων και διευκρινίζεται στις ιδιότητες enctype όταν η POST μέθοδος χρησιμοποιείται.

Το HTML παρέχει ένα βασικό σύνολο στοιχείων διεπαφών ή ελέγχων εισαγωγής που μπορεί να βρσκεται στις forms. Το σχήμα επεξηγεί μια χαρακτηριστική μορφή.

Home \ Join Member

[Traditional Chinese Culture in Taiwan](#)

[Philosophy](#)

[Chinese Macrame](#)

[Folk Arts](#)

[Craftsman](#)

[S.S.J](#)

[Edit Member Data](#)

### Join Member

\*Account:

\*Create Password:

\*Verify Password:

\*Email Address:

Answer These Quick Questions To Join

\*Name: First Name:  Last Name:

BirthDay: January  01  ,  (Month Day, Year)

Gender:  MALE  FEMALE

\*Address:

City:

State:

Zip/Postal Code:

Country:

Το σύνολο στοιχείων περιλαμβάνει:

1. Text entry elements: ενιαία γραμμή ή πολλαπλάσια εισαγωγή γραμμών.
2. Selection menus: κατάλογοι επιλογών που επιδεικνύονται σε ένα εξελισσόμενο παράθυρο επιλογών.
3. Buttons: μιμηθείτε ένα κουμπί επιλογής σε ένα control panel. Αυτός είναι ο συχνότερα χρησιμοποιημένος τρόπος για να αρχίσει μια εντολή, όπως για να υποβάλει μια φόρμα ή να καθαρίσει την εισαγωγή σε τομείς.
4. Check boxes: μικρά τετράγωνα που μπορούν είτε να είναι ανοιχτά είτε κλειστά.
5. Radio buttons: παρόμοια με τα παράθυρα ελέγχου, δείχνουν αν οι τιμές είναι αληθής ή όχι.
6. File selection elements: έλεγχοι που επιτρέπουν στο χρήστη να διευκρινίσει το όνομα του αρχείου που θέλει να φορτώσει. Χαρακτηριστικά, αυτός ο έλεγχος περιλαμβάνει ένα Browse κουμπί που αναγκάζει ένα πλαίσιο διαλόγου επιλογής αρχείων να ανοίξει.
7. Hidden elements: Στοιχεία που χρησιμοποιούνται μη οπτικά για να δημιουργήσουν παραμέτρους με σταθερές τιμές.

## Βάσεις δεδομένων

---

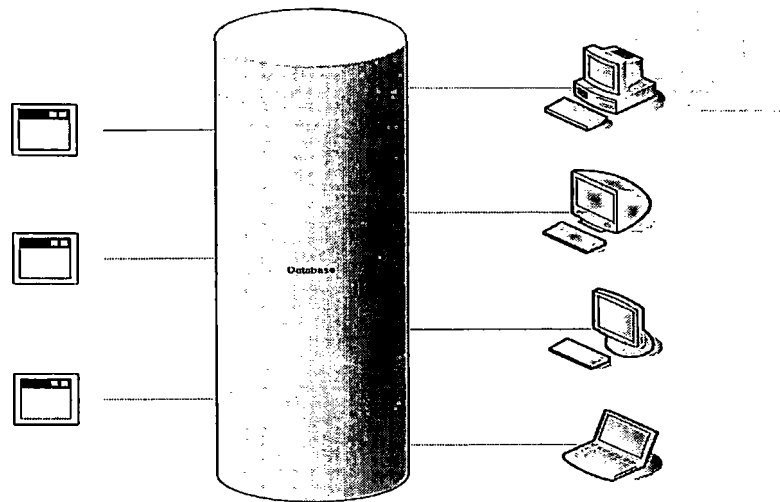
Οι βάσεις δεδομένων έχουν γίνει πια ένα μεγάλο μέρος των εφαρμογών που αφορούν το Internet .Τι είναι μια βάση δεδομένων ; Η βάση δεδομένων μπορεί να θεωρηθεί ως ένα είδος ηλεκτρονικής αρχειοθήκης , με αλλά λόγια ,ένας χώρος για την αποθήκευση μιας συλλογής ηλεκτρονικών αρχείων δεδομένων. Ο χρήστης του συστήματος έχει στη διάθεση του ορισμένα βοηθήματα , για να εκτελεί σε αυτά τα αρχεία διαφορετικές εργασίες , στις οποίες συγκαταλέγονται ,ανάμεσα σε άλλες , και οι εξής:

- ✚ Η προσθήκη κενών αρχείων στη βάση δεδομένων
- ✚ Η εισαγωγή νέων αρχείων σε υπάρχοντα αρχεία
- ✚ Η ανάκληση δεδομένων από υπάρχοντα αρχεία
- ✚ Η ενημέρωση δεδομένων σε υπάρχοντα αρχεία
- ✚ Η διαγραφή δεδομένων σε υπάρχοντα αρχεία
- ✚ Η αφαίρεση υπάρχόντων αρχείων , κενών η όχι, από τη βάση δεδομένων

Η αποθήκευση των δεδομένων μπορεί να είναι μεγάλη ως μια δημόσια βιβλιοθήκη η μικρή ως μια επιχειρησιακοί κάρτα που βρίσκεται σε κάποιο γραφείο.

Στην ηλεκτρονική αποθήκευση των βάσεων δεδομένων, υπάρχει ένα λογισμικό υπολογιστών που επιτρέπει σε έναν χρήστη να υλοποίηση, διατηρήσει και να διαχειριστή μια βάση δεδομένων .Αυτό το λογισμικό καλείται Σύστημα Διαχειρίσεις Βάσεων Δεδομένων (DBMS- Database Management System).

Σύστημα Διαχειρίσεις Βάσεων Δεδομένων είναι βασικά ένα ηλεκτρονικό σύστημα τήρησης εγγραφών. Με απλά λόγια είναι ένα σύστημα, για υπολογιστές, που ο γενικός σκοπός του είναι να τηρεί πληροφορίες και να δίνει αυτές τις πληροφορίες όταν του ζητούνται. Οι πληροφορίες που δίνονται σε ένα τέτοιο σύστημα μπορεί να οτιδήποτε έχει σημασία για το άτομο η τον οργανισμό που εξυπηρετεί το συγκεκριμένο σύστημα – με αλλά λόγια, οτιδήποτε χρειάζεται την υποβοήθηση των εργασιών αυτού του ατόμου η οργανισμού.



Απλοποιημένη εικόνα ενός συστήματος βάσης δεδομένων

Η εικόνα παρουσιάζει μια άποψη ενός συστήματος βάσης δεδομένων ,απλοποιημένη σε μεγάλο βαθμό. Σκοπός της εικόνας είναι να δείξει ότι ένα σύστημα βάσης δεδομένων απαρτίζεται από τέσσερα βασικά στοιχεία : τα δεδομένα ,το υλικό(hardware),το λογισμικό(software),και τους χρήστες(users).

Υπάρχουν συστήματα βάσεων δεδομένων που εκτείνονται σε όλο το φάσμα – από τους μικροϋπολογιστές (ακόμα και τα φορητά PC) μέχρι και τα μεγάλα συστήματα mainframe.Οι υπηρεσίες που παρέχει το κάθε δεδομένο εξαρτώνται ως ένα βαθμό και από την ισχύ του μηχανήματος που είναι εγκατεστημένο . Ειδικότερα τα "τα μεγάλα συστήματα" είναι συστήματα **πολλών χρηστών** ενώ εκείνα των "μικρών συστημάτων" είναι συνήθως συστήματα **ενός χρήστη**.

**Σύστημα ενός χρήστη** (single user system) είναι ένα σύστημα όπου μόνο ένας χρήστης έχει πρόσβαση στην βάση δεδομένων οποιαδήποτε στιγμή .

**Σύστημα πολλών χρηστών** (multi-user system) είναι ένα σύστημα όπου πολλοί χρηστές έχουν πρόσβαση στη βάση δεδομένων ταυτόχρονα.



Τα μέρη του συστήματος που έχουν να κάνουν με το υλικό (hardware) είναι εξής :

- ✓ Τα δευτερεύουσα μέσα αποθήκευσης όπως οι μαγνητικοί δίσκοι με κινούμενες κεφαλές που χρησιμοποιούνται για την αποθήκευση δεδομένων , καθώς και οι αντίστοιχες συσκευές εισόδου / εξόδου , όπως μονάδες δίσκων(disk drives) , οι ελεγκτές συσκευών (device controllers).
- ✓ Ο επεξεργαστής ή οι επεξεργαστές (processor) και η κύρια μνήμη (main memory) που χρησιμοποιούνται για την εκτέλεση του λογισμικού του συστήματος βάσεων δεδομένων.

Οι χρήστες που χρησιμοποιούν τα συστήματα διαχείρισης βάσεων δεδομένων μπορούν να διακριθούν σε τρεις κατηγορίες:

- Οι **προγραμματιστές εφαρμογών** (application programmers) οι οποίοι είναι υπεύθυνοι για το γράψιμο προγραμμάτων εφαρμογών που χρησιμοποιούν τη βάση δεδομένων .Τα προγράμματα είναι γραμμένα σε κάποια γλώσσα προγραμματισμού όπως η C,C++,JAVA και άλλες.
- Υπάρχουν οι **τελικοί χρήστες** οι οποίοι αλληλεπιδρούν με το σύστημα μέσω συνδεδεμένων σταθμών εργασίας ή τερματικών. Ένας τελικός χρήστης μπορεί να προσπελάζει τη βάση δεδομένων μέσω κάποιας από τις εφαρμογές άμεσης επεξεργασίας η να χρησιμοποιεί κάποια διασύνδεση (Interface) που είναι οργανικό μέρος του συστήματος βάσης δεδομένων.
- Στην τρίτη κατηγορία υπάρχει ο **υπεύθυνος διαχείρισης βάσεων δεδομένων** (database administrator , DBA)

Υπάρχουν πολλά είδη Συστημάτων βάσεων δεδομένων σήμερα στην αγορά . Πολλά μεγάλα εμπορικά web site χρησιμοποιούν προϊόντα μεγάλων εταιριών όπως η Oracle.Υπάρχουν όμως και μερικά δωρεάν για τα μικρομεσαία web site. Ένα τέτοιο προϊόν και η MySQL.Η MySQL είναι δημοφιλής με τους ISPs και τους web application developers για την ταχύτητα, την αξιοπιστία και τη ευελιξία του συστήματος έλεγχου πρόσβασης της. Αρχικά η MySQL σχεδιάστηκε για διαχειρίζεται μεγάλες βάσεις δεδομένων με μεγάλη ταχύτητα.

Η MySQL ανήκει στο σχεσιακό μοντέλο συστημάτων βάσεων δεδομένων .Τι είναι το σχεσιακό μοντέλο; Το σχεσιακό μοντέλο είναι το θεμέλιο της σύγχρονης τεχνολογίας των βάσεων δεδομένων, είναι αυτό που κάνει επιστήμη το πεδίο των βάσεων δεδομένων.Η σχεσιακή βάση δεδομένων οργανώνει τα στοιχεία σε πίνακες .Ένας πίνακας έχει ένα όνομα και έχει σειρές και στήλες.

Κάθε σειρά περιέχει τα στοιχεία για κάθε ένα αντικείμενο από τις στήλες .Ένα σχεσιακό σύστημα βάσεων δεδομένων οργανώνει τα δεδομένα σε πίνακες . Ως αποτέλεσμα και οι χρήστες βλέπουν τα δεδομένα σε μορφή πινάκων.

Taiwan Tradition Craftwork   Database Member											
Home   Administrator   Select Database   Member											Sign Out
DELETE											
No	Last Name	First Name	DELETE								
No	Last Name	First Name	Gender	Birthday	E-mail	Address	City	State	Zip	Country	Join Time
4	Lee	CK	male	1957-01-05	gg@yaleva.com	3320 Fremont Ave.	SKOKIE	CA	60039	USA	2001-07-24 10:45:43
3	Tsun	Tung-Lin	male	1971-07-20	tsuntunglin@yahoo.com	2478 Fremont Ave.	SKOKIE	IL	60076	USA	2001-07-19 23:41:30
2	Tsun	Tung-Lin	male	1971-07-28	tsuntong0228@yahoo.com	3326 Fremont Ave.	SKOKIE	IL	60075	USA	2001-07-19 23:27:19
1	Tsun	Tung-Lin	male	1971-07-27	tsuntong0328@yahoo.com	3328 Fremont Ave.	SKOKIE	IL	60075	USA	2001-08-07 12:37:20

Ένας τυπικός πίνακας

## SQL

Η SQL (Structured Query Language - Δομημένη γλώσσα ερωτημάτων) είναι μια γλώσσα υπολογιστών που χρησιμοποιείται για εκφράσει διαδικασίες βάσεων δεδομένων για δεδομένα που είναι οργανωμένα σε σχεσιακή φόρμα (π.χ σε πίνακες). Η SQL υιοθετήθηκε ως πρότυπο της βιομηχανίας το 1986.

Για να εργαστούν μαζί οι άνθρωποι και οι βάσεις δεδομένων πρέπει να μιλούν μια κοινή γλώσσα. Αν και σε γενικές γραμμές κάθε κατασκευαστής βάσεων δεδομένων θα μπορούσε να καθορίσει μια δικιά του τέτοια γλώσσα, αυτό όμως θα προκαλούσε προβλήματα και για το χρήστη και για τους προμηθευτές βάσεων δεδομένων.

Αντ' αυτού, οριστική μια σταθερή γλώσσα η SQL, την οποία υποστηρίζουν όλοι οι προμηθευτές βάσεων δεδομένων.

Η SQL διαφέρει από τις διαδικαστικές γλώσσες στο ότι επικεντρώνεται στον ορισμό και το χειρισμό των δεδομένων και οι εντολές της είναι πολύ κατανοητές μια και γράφονται σε απλά αγγλικά.

Η SQL επιτρέπει στους χρηστές να έχουν πρόσβαση σε δεδομένα σχεσιακών μοντέλων όπως αυτά της Oracle,MySQL,Microsoft SQL Server,Access και αλλά επιτρέποντας στους χρηστές να επιλέγουν ποια δεδομένα θέλουν να δουν.

Η SQL επιτρέπει στους χρηστές να ορίσουν τα δεδομένα και να τα χειριστούν. Αυτή εργασία θα χρησιμοποιήσει SQL εντολές για χειριστεί το σύστημα. Η SQL έχει δυο κύρια συστατικά:

(1) Data Manipulation Language - Γλώσσα χειρισμού δεδομένων (DML):συμπεριλαμβάνοντας εντολές όπως SELECT, INSERT

UPDATE και DELETE.

(2) Data Definition Language - Γλώσσα καθορισμού στοιχείων (DDL): συμπεριλαμβάνοντας εντολές όπως CREATE, DROP και ALTER

## **JDBC**

---

Πριν μερικά χρόνια , για να αντιμετωπίσει η Microsoft το πρόβλημα των διαφορετικών τύπων βάσεων δεδομένων , ανέπτυξε το πρότυπο ODBC(Open Data Base Connectivity), ένα πρότυπο για την προσπέλαση βάσεων δεδομένων. Το ODBC καθορίζει ένα συγκεκριμένο τύπο συμπεριφοράς των εφαρμογών στην περίπτωση που αυτές καλούν βάσεις δεδομένων , και έναν άλλον όταν μια συγκεκριμένη βάση δεδομένων ανταποκρίνεται σε κλήσεις τύπου ODBC από μια εφαρμογή .

Η εταιρεία Sun συμπεριέλαβε στην Java τη διασύνδεση JDBC η οποία καθορίζει τον τρόπο επικοινωνίας μιας εφαρμογής με σχεσιακές βάσεις δεδομένων . Η γλώσσα με αυτόν τον τρόπο κατορθώνει να υποστηρίζει σχεσιακές βάσεις δεδομένων που δημιουργούνται από διαφορετικά DBMS ,όπως κατορθώνει να υποστηρίζει και τα διαφορετικά λειτουργικά συστήματα.

Το JDBC επιτρέπει στον προγραμματιστή να γράψει εφαρμογές βάσεων δεδομένων με τη χρήση αποκλειστικά και μόνο της γλώσσας Java. Παρότι η ίδια η εταιρεία Sun δηλώνει ότι το JDBC δεν είναι ακρώνυμο , πολλοί θεωρούν ότι είναι το ακρώνυμο των λέξεων Java Data Base Connectivity. Όπως και το ODBC της Microsoft , έτσι και το JDBC καθορίζει το πώς οι εφαρμογές θα επικοινωνούν με τις βάσεις δεδομένων και ταυτόχρονα το πώς θα κατασκευάζονται οι Β.Δ. ώστε να μπορούν να ανταποκρίνονται στις κλήσεις των εφαρμογών. Η διάφορα είναι ότι το ODBC είναι γραμμένο σε γλώσσα C , ενώ το JDBC

είναι αποτελείται αποκλειστικά από κλάσεις και διασυνδέσεις αποκλειστικά γραμμένες σε Java.

Το JDBC δίνει στον προγραμματιστή τις εξής δυνατότητες:

- Να συνδέσει μια εφαρμογή με μια υπάρχουσα βάση δεδομένων.
- Να δημιουργήσει εντολές SQL και να τις περάσει στην βάση δεδομένων προκειμένου να εκτελέσει διάφορες εργασίες.
- Να πάρει τα αποτελέσματα των επεξεργασιών των εντολών SQL από τη βάση δεδομένων.
- Να επεξεργαστεί περαιτέρω τα αποτελέσματα που έλαβε.

Η JSP , η κύρια γλώσσα που θα χρησιμοποιήσουμε σε αυτήν την εργασία , χρησιμοποιεί τις βιβλιοθήκες JDBC για να επικοινωνεί με την βάση δεδομένων έτσι ώστε τα δεδομένα της βάσης να μπορούν να ενημερωθούν, να ανακτηθούν, ή να διαγραφούν χρησιμοποιώντας JSP φόρμες .

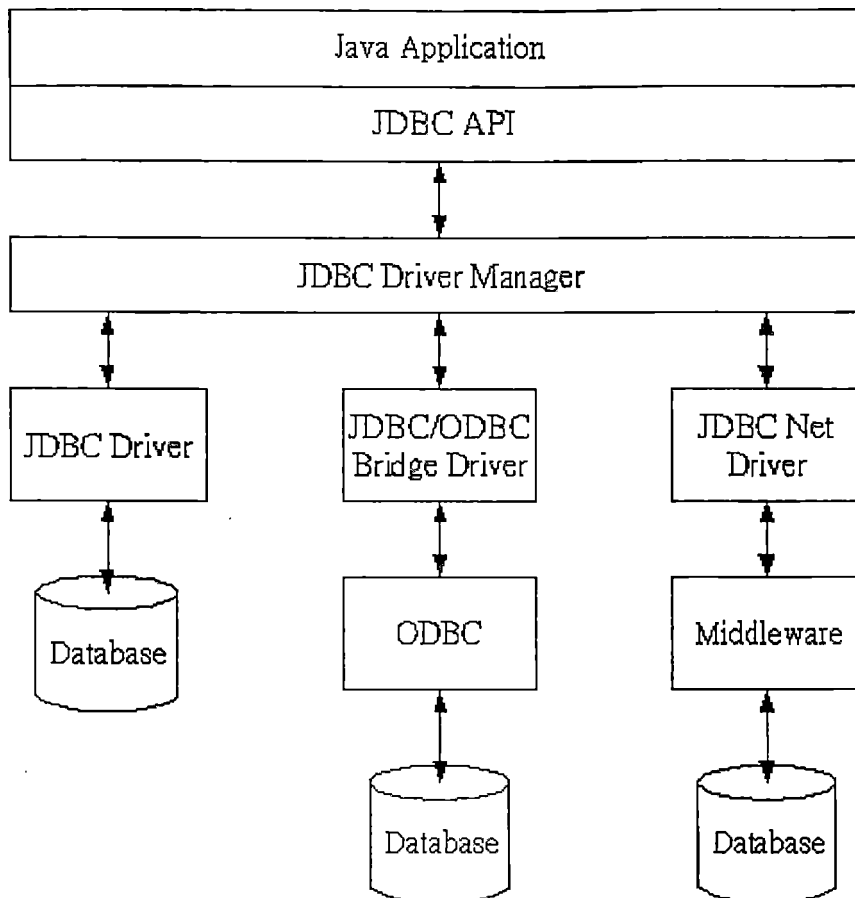
Ένα πρόγραμμα που κάνει χρήση του JDBC , για να ανοίξει και να χρησιμοποιήσει μια βάση δεδομένων ακολουθεί τα παρακάτω βήματα:

- Εισάγει τις απαραίτητες κλάσεις
- Φορτώνει τον κατάλληλο οδηγό JDBC
- Προσδιορίζει την πηγή των δεδομένων
- Δημιουργεί ένα αντικείμενο Connection
- Δημιουργεί ένα αντικείμενο Statement
- Κάνει ερώτηση (query) στη Β.Δ. με την βοήθεια του αντικειμένου Statement
- Παίρνει δεδομένα από το αντικείμενο ResultSet που επιστρέφεται
- Κλείνει το αντικείμενο ResultSet
- Κλείνει το αντικείμενο Statement
- Κλείνει το αντικείμενο Connection

Η όλη διεργασία ξεκινάει με την κλάση Driver Manager . Αυτός συνδέει τον κατάλληλο οδηγό JDBC (που αντιστοιχεί στη βάση δεδομένων με την οποία θέλουμε να επικοινωνήσουμε) με την εφαρμογή μας. Μόλις αποκατασταθεί η σύνδεση , ο Driver Manager διεκπεραιώνει την επικοινωνία μεταξύ της εφαρμογής και του οδηγού.

Στη συνέχεια η σύνδεση γίνεται με τρεις τρόπους :

- I. Με τον JDBC Driver και μετά την βάση δεδομένων
- II. Τη γέφυρα JDBC-ODBC .Η γέφυρα επιτρέπει σε μια εφαρμογή συμβατή με JDBC να επικοινωνεί με βάση δεδομένων συμβατή με ODBC. Η γέφυρα μεταφέρει εντολές και αποτελέσματα μέσω του οδηγού ODBC.
- III. Και τον JDBC NET Driver.



#### Η ΔΟΜΗ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ JDBC

Οι JDBC Drivers είναι κατηγοριοποιημένοι σε τέσσερις τύπους:

- 1) Τύπος 1-Η γέφυρα JDBC-ODBC
- 2) Τύπος 2- Εγγενής δυαδικός κώδικας. Παρόμοιος με την γέφυρα JDBC-ODBC μόνο που μπαίνει μέσα σε κώδικα Java.
- 3) Τύπος 3-Καθαρή Java στο middleware(υλικολογισμικό) της βάσης δεδομένων. Οι οδηγοί τύπου 3 επικοινωνούν χρησιμοποιώντας πρωτόκολλα δικτύων με middleware servers ,οι οποίοι επικοινωνούν με ένα η παραπάνω Σύστημα Διαχείρισης Βάσεων Δεδομένων.( JDBC NET Driver)
- 4) Τύπος 4-Καθαρή Java απ' ευθείας στη βάση δεδομένων. Αυτός ο τύπος οδηγού μπαίνει κατευθείαν στο εγγενές πρωτόκολλο που χρησιμοποιείται από σύστημα διαχείρισης βάσεων δεδομένων.

Και τύπος 1 και ο τύπος 2 απαιτούν κώδικα να εγκατασταθεί και συντεθεί ανάλογα στα συστήματα των πελατών.

Ο τύπος 3 χρησιμοποιεί ένα πρωτόκολλο δικτύων που επικοινωνεί μεταξύ του κεντρικού υπολογιστή και του πελάτη.

Οι οδηγοί τύπου 4 μπορεί να μην είναι κατάλληλοι για συστήματα διαχείρισης βάσεων δεδομένων που είναι πίσω από firewall αλλά είναι αρκετά κατάλληλα για java servlets και JSP(java server page) που είναι συνδεδεμένα σε μια database server που είναι στο ίδιο hardware η στο ίδιο τοπικό δίκτυο.

## **2. Servlets, JavaServer Pages και JavaBeans**

### **Τι είναι ένα Servlet;**

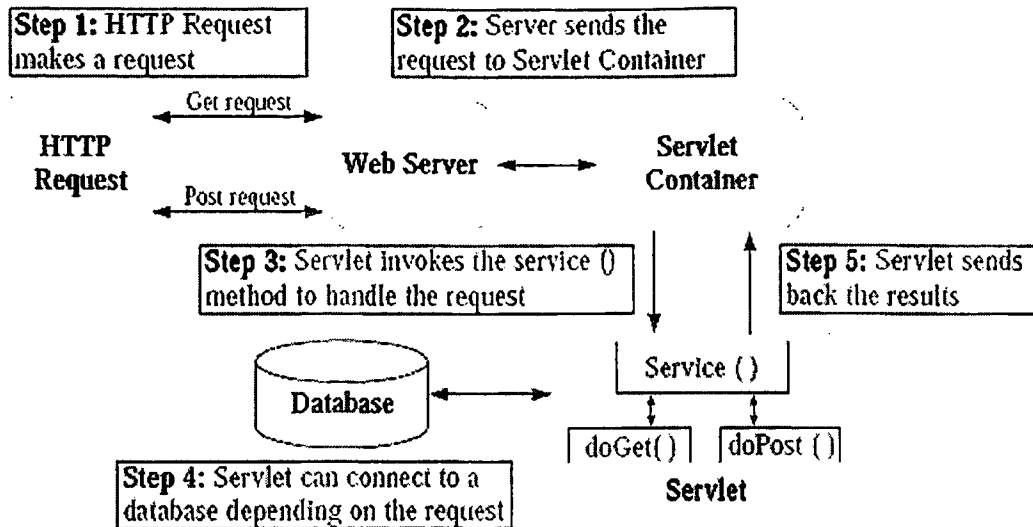
---

**Με τον όρο servlet (μικροεφαρμογή διακομιστή) εννοούμε μια εφαρμογή ή ένα σενάριο (script) γραμμένο σε γλώσσα Java το οποίο εκτελείται σε έναν εξυπηρετητή (π.χ Ιστού) σε αντίθεση με τα προγράμματα applet που εκτελούνται σε υπολογιστές-πελάτες.**

Servlet είναι μια δυναμική κλάση που εξυπηρετεί αιτήματα από το internet ή από έναν server. Τα Servlet είναι σημείο κλειδί για την ανάπτυξη της java ως προς τους server. Ένα Servlet είναι μια μικρή επέκταση σε ένα server που εμπλουτίζει την λειτουργικότητα του server. Τα Servlet επιτρέπουν στους προγραμματιστές να επεκτείνουν η να τροποποιήσουν τύπο server (web server, mail server η οποιονδήποτε άλλο server ) που είναι βασισμένος στη java με ένα άγνωστο μέχρι τώρα βαθμό ευκαμψίας , μεταφερσιμότητας και ευκολίας.

Επειδή τα Servlet «τρέχουν» σε server , δεν εξαρτώνται από την συμβατότητα με τους browser.

Η εικόνα δείχνει την ροή ενός μηνύματος ενός http αιτήματος σε ένα servlet , το οποίο ξεκινάει όταν ο χρήστης μπαίνει στο site.



Ροή μηνύματος σε ένα web server με servlet container

## Κύκλος ζωής ενός servlet

Κάθε servlet container έχει έναν ορισμένο επακριβώς κύκλο ζωής.

Γεννιέται είτε όταν servlet container ξεκινάει , στο πρώτο αίτημα , είτε πάνω στο πρώτο αίτημα μετά από κάποια αλλαγή στην κλάση (αρχείο) του servlet .

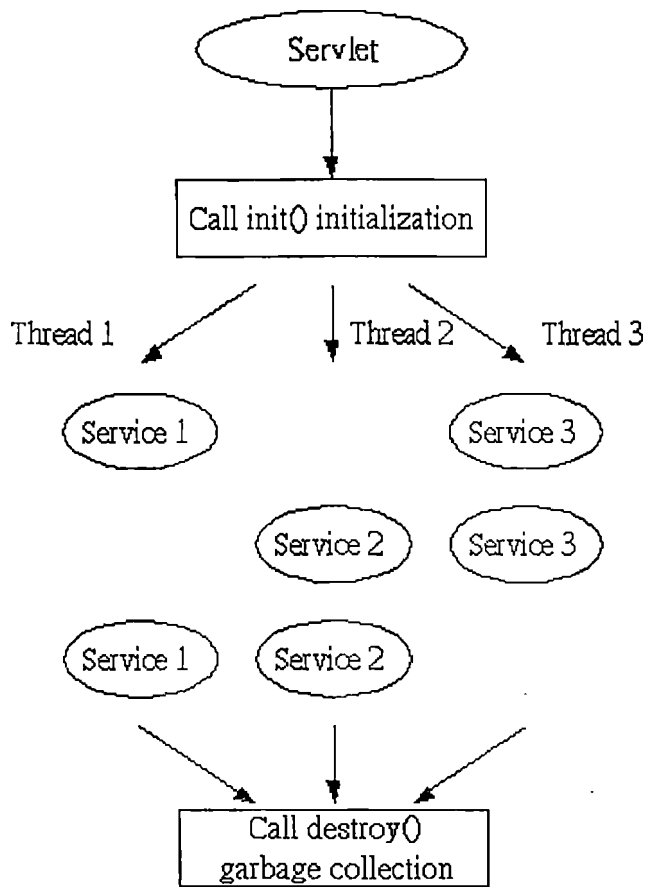
Όλα τα servlet υπάρχουν για να διαχειρίζονται αιτήματα και λήγουν όταν βρεθούν εξαιρέσεις (σφάλματα) , πάνω στην διαδικασία φόρτωσης από το servlet container , η όταν το servlet container σταματήσει να λειτουργεί .

Για την επεξεργασία του αιτήματος , κάθε servlet πρέπει να εκτελεί τρεις «στάνταρ» μεθόδους : την init ( ) , service ( ) και την destroy ( ) .

Το container ενός servlet κάνει τα παρακάτω :

- 1) Ο server φορτώνει το servlet όταν αρχικά ζητηθεί
- 2) Ο server εκτελεί την μέθοδο init ( ) του servlet
- 3) Το servlet χρησιμοποιεί πολύ-επεξεργασία για να διαχειριστή τα αιτήματα καλώντας την μέθοδο service ( ) .

- 4) Όταν τελειώνει η διαδικασία καλείται η μέθοδος `destroy ( )` του κάθε `servlet` για «καταστρέψει» το `servlet`.



Ο κύκλος ζωής ενός `servlet`

### **Τα πλεονεκτήματα των `servlet`**

---

Τα `servlet` προσφέρουν έναν αριθμό πλεονεκτημάτων, περιλαμβάνοντας: μεταφερσιμότητα, δύναμη, αποδοτικότητα, ασφάλεια, ενσωμάτωσης, και ευελιξίας. Τα κυριότερα πλεονεκτήματα πάνω στα περιβάλλοντα προγραμματισμού για `server` :



- 1) Έχουν καλύτερη απόδοση επειδή μπορούν να παραμείνουν εσωτερικά και να διαχειριστούν πολλαπλά νήματα ταυτόχρονα.
- 2) Απλότητα επειδή δεν απαιτούν κάποια εγκατάσταση λογισμικού στους χρήστες εκτός από έναν web browser.
- 3) Αν και οι http servers δεν έχουν κάποια δυνατότητα για να θυμούνται λεπτομέρειες κάποιου προηγούμενου αιτήματος από τον ίδιο χρήστη , τα servlet χορηγούν ένα μέσο που παρακολουθεί την API που ξεπερνά αυτήν την οριοθέτηση .

Η πρόσβαση στη τεχνολογία της JAVA περιλαμβάνοντας τα νήματα , την δουλειά πάνω στα δίκτυα και την επικοινωνία με τις βάσεις δεδομένων

## **Java Server Page (JSP)**

---

Η Java, έχει εξελιχθεί πολύ από την πρώτη της παρουσίαση από την Sun. Μετά τα applets που αποτέλεσαν τη βασική μέθοδο αξιοποίησης των (πολλών) δυνατοτήτων της γλώσσας, ακολούθησε το server side scripting με την τεχνολογία Java Server Pages και σήμερα η πλατφόρμα ωριμάζει, και πλέον προσφέρει ένα ολοκληρωμένο περιβάλλον για την ανάπτυξη Enterprise εφαρμογών που αξιοποιούν πλήρως τις δυνατότητες που προσφέρει το Internet, την Java 2 Enterprise Edition.

Η Sun παρουσίασε την τεχνολογία JSP για ν' ανταγωνιστεί την Microsoft στο χώρο των Server-side applications. Μια Java Server Page (JSP) είναι ένα πρότυπο για ιστοσελίδες που χρησιμοποιεί τον κώδικα της Java για να παράγει ένα έγγραφο HTML δυναμικά. Η JSP τρέχει σε ένα server-side component γνωστό ως JSP container, το οποίο μεταφράζει τις σελίδες σε Java Servlets. Την πρώτη φορά που έχουμε πρόσβαση σε ένα JSP μπορούμε να παρατηρήσουμε ότι παίρνει λίγο χρόνο για να ανταποκριθεί.

### **Πλεονεκτήματα / μειονεκτήματα**

Η JSP χρησιμοποιεί component objects, tags που επιτρέπουν την ενσωμάτωση server side κώδικα στην HTML σελίδα, session tracking και διάδραση με βάσεις δεδομένων. Επίσης η JSP χρησιμοποιεί JavaBeans για να υλοποιήσει την component αρχιτεκτονική.

Η τεχνολογία JSP δεν "δένει" αναγκαστικά με κάποιο συγκεκριμένο Web server ή λειτουργικό, και εκτός Microsoft είναι ήδη ένα ευρέως αποδεκτό standard.

Ένας παράγοντας δυσκολίας είναι η εξοικείωση πολλών developers με τη χρήση των RAD εργαλείων της Microsoft (Visual Studio/InterDev), αλλά πλέον στην αγορά κυκλοφορούν ισάξια / καλύτερα περιβάλλοντα για Java /JSP/J2EE (Borland JBuilder).  
Ακόμα η αντικειμενοστραφής φύση της Java, προσθέτει ακόμα περισσότερο στο learning

curve, για τους developers που δεν είναι εξοικειωμένοι με τον object oriented προγραμματισμό.

Η Java είναι μια γλώσσα με πολύ περισσότερες δυνατότητες από την Visual Basic/VBScript. Ένα ακόμα σημαντικό πλεονέκτημα της τεχνολογίας JSP είναι τα extensible tags. Με αυτό τον μηχανισμό, ο developer μπορεί να δημιουργήσει custom tags επιτρέποντας με αυτό τον τρόπο την επέκταση της tag γλώσσας των JSP.

### **Επέκταση / δυνατότητες**

Η τεχνολογία JSP, είναι στην ουσία μια υλοποίηση του Servlet API της Sun. Ακριβώς για αυτό το λόγο, αποτελεί κομμάτι της J2EE αρχιτεκτονικής, πράγμα που της επιτρέπει να χρησιμοποιεί όλα τα J2EE components όπως JavaBeans, Enterprise JavaBeans components & Java servlets. Στην ουσία οι JSP σελίδες γίνονται compiled σαν servlets, οπότε έχουν όλα τα χαρακτηριστικά μιας Java server side εφαρμογής. Επίσης μέσα από JSP σελίδες, έχουμε πρόσβαση σε όλα τα services της J2EE όπως JDBC για πρόσβαση σε βάσεις δεδομένων, JavaMail Java Message Service (JMS) κ.α. Επίσης οι JSP σελίδες μπορούν να επικοινωνούν με enterprise applications χρησιμοποιώντας CORBA-compliant τεχνολογίες όπως: Java IDL και RMI-POP.

### **Πως λειτουργεί**

Ο κώδικας για μία JSP σελίδα, είναι στην ουσία HTML που περιέχει εκτός από τα γνωστά html tags και τα ειδικά JSP tags ή / και Java κώδικα σε JSP tags. Η κατάληξη του αρχείου είναι .jsp και αυτό ενημερώνει τον server ότι αυτή η σελίδα απαιτεί "ειδική μεταχείριση".

Η "ειδική" αυτή μεταχείριση, που επιτυγχάνετε μέσα από server extention ή plugin (ανάλογα με τον server) και αποτελείται από 4 στάδια:

- Η JSP engine κάνει parse τη σελίδα και δημιουργεί ένα Java source file.
- Κάνει compile το java αρχείο σε java class. Το class αυτό είναι ένα Java servlet, και από αυτό το σημείο η servlet engine το αντιμετωπίζει σαν τέτοιο
- Η servlet engine φορτώνει το servlet προς εκτέλεση
- Το servlet εκτελείται και επιστρέφει τ' αποτελέσματα του στον web server που τα παρουσιάζει με τη μορφή απλού html κώδικα στον browser του χρήστη

Παρότι η διαδικασία αυτή μπορεί να ακούγεται χρονοβόρα, είναι πιο αποτελεσματική απ ότι φαίνεται. Τα δύο πρώτα στάδια εκτελούνται μόνο μία φορά, όταν κάνουμε deploy ή update τις σελίδες. Το 3ο στάδιο εκτελείται μόνο στο πρώτο request για το servlet, από τη στιγμή που ξεκίνησε ο server. Μετά απ αυτό ο class loader φορτώνει την class μια φορά και είναι διαθέσιμο για όλη τη "ζωή" της Java Virtual Machine. Επίσης, πολλοί application servers παρέχουν page caching, που μπορεί να βελτιώσει ακόμα περισσότερο την αποτελεσματικότητα. Με page caching, ακόμα και το 4ο στάδιο μπορεί να εκτελεστεί μόνο μια φορά, πράγμα που εξαρτάται φυσικά από το πόσο δυναμικά είναι τα data που περιέχει η σελίδα.

### **Βασικά JSP tags**

Οι JSP σελίδες μπορούν να περιέχουν πολλών ειδών tags, ακόμα και κάποια που ορίζει ο developer, όμως 4 είναι τα κύρια tags που αποτελούν το βασικό "οπλοστάσιο" για να ξεκινήσει κανείς την ανάπτυξη σελίδων JSP:

- Declaration tag (<%! %>) Το tag αυτό χρησιμοποιείται για την δήλωση μεταβλητών ή μεθόδων π.χ: <%! int k = 0 ; %>
- Specification tag (<% %><) Το tag αυτό χρησιμοποιείται για ενσωμάτωση Java κώδικα. π.χ.  

```
<% for ( i= 0 ; k < user_array.count; k++) {
out.println( user_array[i] ) ; }
%>
```
- Expression tag (<%= %>) Το tag αυτό χρησιμοποιείται για να εμφανίσουμε μία Java expression, κι είναι shorthand για την μέθοδο out.println(). π.χ <%= user\_array[2] %>
- JavaBean tags(<jsp:useBean >, <jsp:setProperty>) Για την ενσωμάτωση JavaBeans. Το tag εντοπίζει τα beans μέσω των "scope" και "class" attributes που περιέχονται σ' αυτό. Εάν δεν εντοπιστεί το bean, τότε δημιουργείται ένα νέο από την class που έχουμε ορίσει στο tag.

### **Εν κατακλείδι**

Με αυτή την παρουσίαση των JSP απλά αγγίξαμε την επιφάνεια αυτής της τεχνολογίας. Οι σελίδες JSP είναι πολύ περισσότερα από απλές HTML με ενσωματωμένη Java. Αποτελούν ένα μόνο component της J2EE enterprise server αρχιτεκτονικής. Αξιοποιώντας αυτή τη τεχνολογία που περιλαμβάνει JSP, servlets, EJB, JNDI, XML, κ.α. γίνετε αντιληπτό πως αποτελεί μια πολύ ανταγωνιστική αρχιτεκτονική για τη δημιουργία distributed applications και σίγουρα αποτελεί μια καλή εναλλακτική λύση γι αυτούς που δεν θέλουν να περιοριστούν στις αρχιτεκτονικές της Microsoft.

---

### **ΤΙ ΕΙΝΑΙ ΕΝΑ JAVA BEAN**

Ένα java bean είναι 100% συστατικό της java που δουλεύει σε κάθε java virtual machine .Με τη χρήση αυτής της τεχνολογίας ο προγραμματιστής μπορεί να γράψει έναν «κόκκο» ( Bean ), που είναι ένα κομμάτι επαναχρησιμοποιημένου κώδικα ( reusable code ), το οποίο σε άλλες περιπτώσεις έχει εξωτερική μορφή γραφικών και σε άλλες όχι.

Με την τεχνολογία Java Beans η Sun θέλησε να δημιουργήσει μια αρχιτεκτονική βασισμένη σε συστατικά (component base) για τη δημιουργία, οπτικών η μη, τμημάτων επαναχρησιμοποιημένου κώδικα για την κατασκευή εφαρμογών .

Όταν προσθέτουμε και άλλο java κώδικα σε μια jsp σελίδα γίνεται όλο και ποιο δύσκολο για να διαχειριστή κάποιος την σελίδα. Είναι ακόμα ποιο δύσκολο όταν ο κώδικας java μπερδευτεί με την html. Ο τρόπος με τον οποίο μπορούμε να κρατήσουμε την java ξεχωριστή είναι να χρησιμοποιήσουμε java bean.

Τα παρακάτω είναι οι ελάχιστες απαιτήσεις που ένα συστατικό σε java bean :

- 1) Πρέπει οι constructor να έχουν μηδενική αντιδικία.
- 2) Πρέπει να χρησιμοποιεί getters και setters για να εκθέτει τις ιδιότητες.
- 3) Πρέπει να υποστηρίζει το JDK 1.1

Με αυτούς τους περιορισμούς, οι πολλές κλάσεις είναι ήδη «κόκκοι» ( beans ) η μπορούν με λίγη προσπάθεια να μετατραπούν σε bean . Δεν χρειάζεται ένα ξεχωριστό περιβάλλον για να τρέχουν τα bean εκτός από κάποια Java Virtual Machine.

Για να χρησιμοποιήσουμε ένα Java Bean , πρώτα , πρέπει να καταλάβουμε τις ιδιότητες του . Οι πιο πολλοί κόκκοι έχουν ιδιότητες.

Ένας κόκκος περιέχει :

- Γεγονότα
- Ιδιότητες
- Μεθόδους

Τα γεγονότα ( events ) είναι αντικείμενα που χρησιμοποιούνται για να δηλώσουν μια αλλαγή στην κατάσταση ενός αντικειμένου-πηγής του γεγονότος .

Όταν συμβεί ένα γεγονός σε έναν κόκκο , παράγεται ένα αντικείμενο γεγονός το οποίο ειδοποιεί μέσω της κατάλληλης μεθόδου τον ακροατή που έχει καταχωρηθεί για να παρακολουθεί το γεγονός .

Οι ιδιότητες που καθορίζουν τα χαρακτηριστικά ενός κόκκου είναι κάτι ανάλογο με τις ιδιότητες ενός τυπικού συστατικού Java π.χ. ενός button . Ορισμένες από τις ιδιότητες αυτές αντιπροσωπεύουν ορατά του κόκκου , ενώ άλλες μη ορατά .

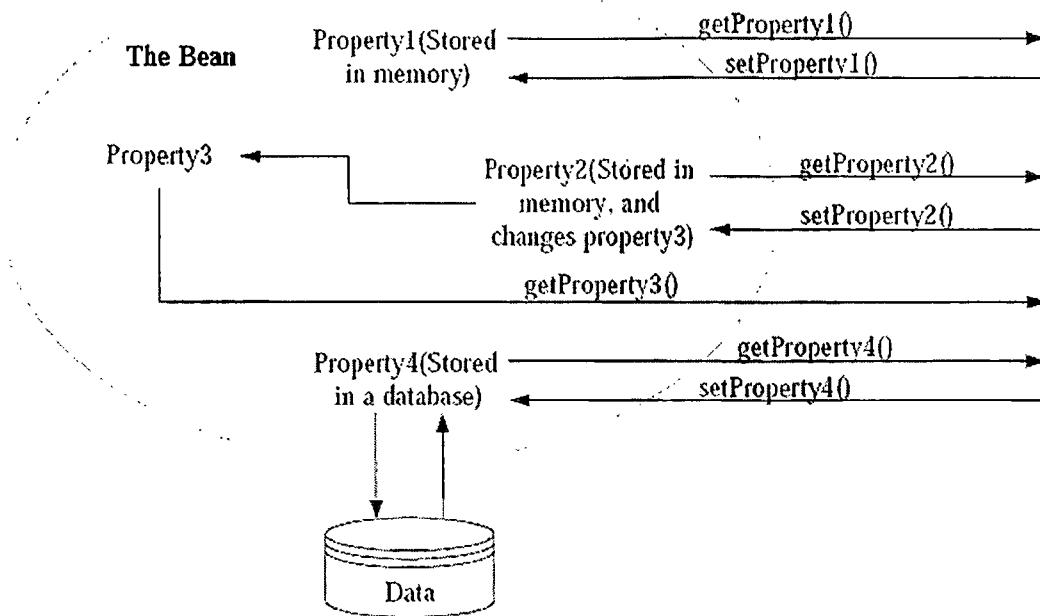
Για να δημιουργηθεί ή να ανακτηθεί μια ιδιότητα χρησιμοποιούνται μέθοδοι όπως οι παρακάτω :

```
Public void set < όνομα_ιδιότητας > (<τύπος> < τιμή >);
```

```
Public < τύπος > get < όνομα_ιδιότητας > ( ) ;
```

Μια ιδιότητα μπορεί να είναι μόνο για εγγραφή (λείπει η μέθοδος get ) , μόνο για ανάγνωση ( λείπει η set ) , ή για εγγραφή και ανάγνωση ( υπάρχουν και οι δυο μέθοδοι ).

Ένα μοντέλο Java Bean φαίνεται στην παρακάτω φωτογραφία .



Μοντέλο ενός Java Bean

### ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΩΝ JAVA BEAN

Ένα πλεονέκτημα των Java Bean είναι ότι δεν χρειάζεται να ξέρεις κάτι από πριν ( εκτός από Java ) για να χρησιμοποιήσεις .

Άλλο ένα είναι ότι δεν χρειάζεται να κάνεις κάτι για να αναζητήσεις η να αλλάξεις μια ιδιότητα , απλά να ψάξεις κάποιες πληροφορίες σε μια βάση δεδομένων .

Επίσης σύμφωνα με την γνώμη πολλών ειδικών , η τεχνολογία JavaBeans υπερτερεί μια και οι ανάλογες τεχνολογίες της Microsoft λειτουργούν κυρίως στα WINDOWS ή μόνο στα WINDOWS.

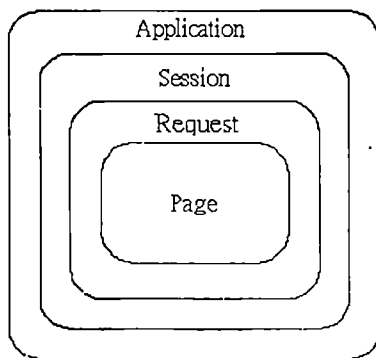
### JAVA SERVER PAGE ΚΑΙ JAVA BEANS

Για να χρησιμοποιήσεις JavaBeans μέσα σε JSP , τότε οι προδιαγραφές του JSP απαιτούν τρία tags : ένα είναι να εντοπίσει η να δημιουργήσει ένα Java Bean , άλλο ένα είναι να θέσει( set ) μια ιδιότητα σε κάποιο Java Bean και τέλος να πάρει ( get ) μια ιδιότητα από ένα Java Bean .

Purpose	Syntax
To load a JavaBean	<code>&lt;jsp:useBean id="name" scope="page request session application" class="className" /&gt;</code>
To set a property	<code>&lt;jsp:setProperty name="beanName" prop_expr /&gt;</code>
To get a property	<code>&lt;jsp:getProperty name="beanName" property="propertyName" &gt;</code>

### Σύνταξη για τη χρησιμοποίηση ενός JavaBean σε JSP

Οι ιδιότητες πεδίου για τη φόρτωση ενός JavaBean καθορίζει το αντικείμενο με το οποίο εμείς συνδέουμε το bean. Αυτό το πεδίο του JavaBean φαίνεται στο σχήμα



Το πεδίο ενός JavaBean

- (1) Page: Έγκυρη όσο διαρκεί αυτή η σελίδα JSP. Αυτό είναι το πεδίο προεπιλογής.
- (2) Request: Έγκυρη για το υπόλοιπο της σελίδας JSP και για οποιουσδήποτε άλλους πόρους συντήρησης αυτού του αιτήματος.
- (3) Session: Έγκυρη κατά τη διάρκεια της εκτέλεσης οποιασδήποτε σελίδας JSP ή Java Servlet σε αυτή το HTTP session.
- (4) Application: Έγκυρος σε οποιαδήποτε σελίδα JSP ή Java Servlet σε αυτήν την WEB εφαρμογή.

---

## MySQL

### ΕΙΣΑΓΩΓΗ

Η MySQL είναι το πιο γνωστό σύστημα διαχείρισης βάσεων δεδομένων (ανοιχτού κώδικα), αναπτύχθηκε , διανέμεται και υποστηρίζεται από την MySQL AB . Η MySQL AB είναι μια εταιρεία που ενώνει το πρόγραμμα ανοιχτού κώδικα με εμπορικούς σκοπούς .

Το site της MySQL είναι (<http://www.mysql.com/>) και έχει τις τελευταίες πληροφορίες που αφορούν το λογισμικό της MySQL.

Παρακάτω παραθέτουμε κάποιες πληροφορίες για την MySQL:

- Η MySQL είναι ένα σύστημα διαχείρισης βάσεων δεδομένων
- Είναι βασισμένη πάνω στο σχεσιακό μοντέλο
- Είναι πρόγραμμα ανοιχτού κώδικα
- Ο MySQL Database Server είναι πολύ γρήγορος , αξιόπιστος και εύκολος στη χρήση
- Ο MySQL Server δουλεύει σε συστήματα που βασίζονται στο μοντέλο πελάτη / εξυπηρετητή
- Πολλά προγράμματα σχετικά με την MySQL είναι διαθέσιμα

## 2. ΙΣΤΟΡΙΚΑ

Ξεκινώντας υπήρχε η πρόθεση να χρησιμοποιηθεί το σύστημα βάσεων δεδομένων *mysql* για να ενωθούν πίνακες χρησιμοποιώντας κάποιες ρουτίνες . Όμως μετά από κάποιο πειραματισμό, καταλήξανε στο συμπέρασμα ότι η *mysql* δεν ήταν γρήγορη αρκετά ούτε ευέλικτη αρκετά για της ανάγκες που υπήρχαν.

Ο προσδιορισμός του ονόματος MySQL δεν είναι ξεκάθαρος. Τα βασικά αρχεία και πολλές βιβλιοθήκες που υπάρχουν έχουν το πρόθεμα *my* για πάνω από 10 χρόνια. Όμως η κόρη του συν-ιδρυτή Monty Widenius ονομάζεται και αυτή *My*. Τι από τα δυο έδωσε το όνομα της στη MySQL παραμένει ακόμα ένα μυστήριο.

### ΚΥΡΙΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Οι παρακάτω γραμμές παραθέτουν κάποια απο τα πιο σημαντικά χαρακτηριστικά της MySQL :

Εσωτερικά της MySQL και μεταφερσιμότητα :

- Είναι γραμμένη σε C και C++
- Είναι δοκιμασμένη με μια μεγάλη γκάμα διαφορετικών compilers
- Δουλεύει σε πολλές διάφορες πλατφόρμες
- Χρησιμοποιεί GNU Automake, Autoconf, και Libtool για τη μεταφερσιμότητα
- Πλήρως πολυ-επεξεργαστική χρησιμοποιώντας πυρήνα νημάτων
- Πολύ γρήγορο σύστημα μνήμης
- Εύκολο να προστεθούν άλλες συσκευές αποθήκευσης
- Ο κώδικας της MySQL είναι δοκιμασμένος σε πολλά εργαλεία

- Ο server είναι διαθέσιμος και ως πρόγραμμα αλλά και ως βιβλιοθήκη

#### Τύποι δεδομένων

- Έχει πολλούς τύπους δεδομένων ( FLOAT,DOUBLE κτλ)
- Πολλαπλές εγγραφές

#### Συνδεσιμότητα

- Clients μπορούν να συνδεθούν στον MySQL server χρησιμοποιώντας TCP/IP socket σε κάθε πλατφόρμα.
- Ο connector/ODBC (MyODBC ) προσφέρει υποστήριξη για MySQL προγράμματα που χρησιμοποιούν ODBC (Open DataBase Connectivity ).
- Ο connector/j προσφέρει υποστήριξη για πλατφόρμες Java

### **3. Οδηγός για την εγκατάστασης λογισμικού.**

#### **Εισαγωγή.**

---

Τώρα θα παρουσιάσουμε έναν οδηγό για την εγκατάσταση του Apache web server με υποστήριξη php και jsp σε ένα μηχάνημα με Windows xp pro service pack 2 .

Παρακάτω παραθέτουμε τις version που χρησιμοποιήσαμε εμείς :

Windows XP PRO service pack 2

Apache 2.0.52

PHP 5.0.3

Tomcat 5.5.4

Java 2 Standard Edition 5.0

MySQL 4.1.7

MyODBC Connector 3.51.10

PHP MyAdmin 2.6.0 patch level 3



Χρειαζόμαστε τα εξής αρχεία :

apache\_2.0.52-win32-x86-no\_ssl.msi

php-5.0.3-Win32.zip

Jakarta-tomcat-connectors-jk2.0.4-win32-apache2.0.49.zip

Jre-1\_5\_0-windows-i586.exe

Mysql-4.1.7-win.zip

MyODBC-3.51.10-x86-win-32bit.msi

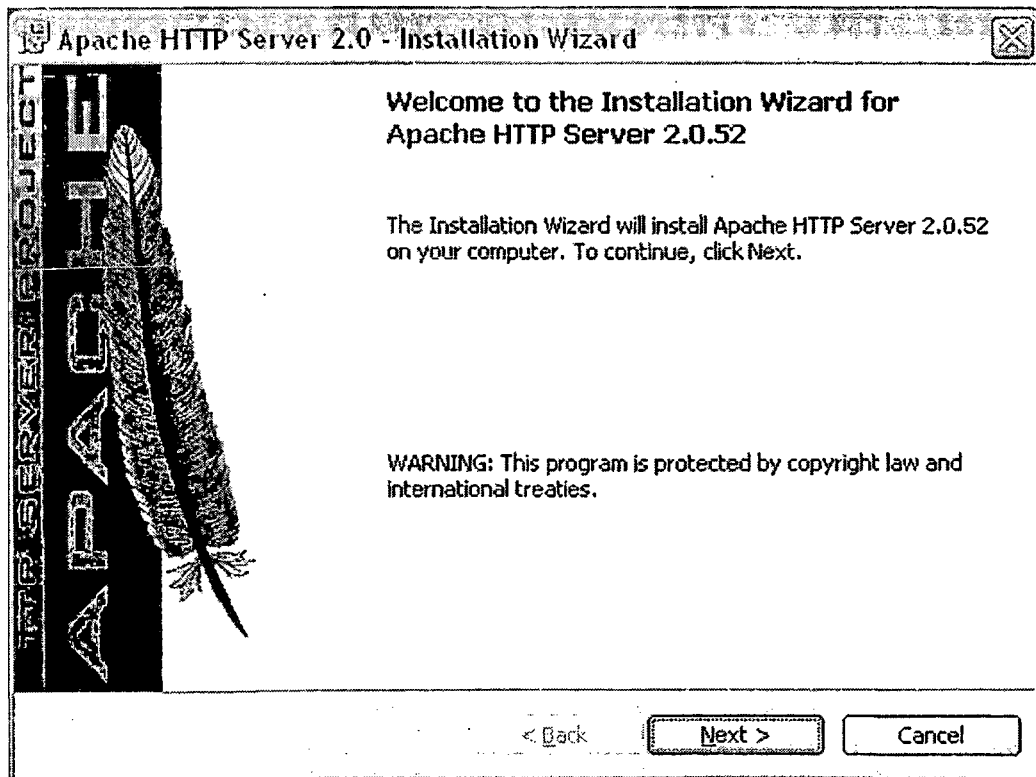
phpMyAdmin-2.6.0-pl3.zip

## Εγκατάσταση Apache

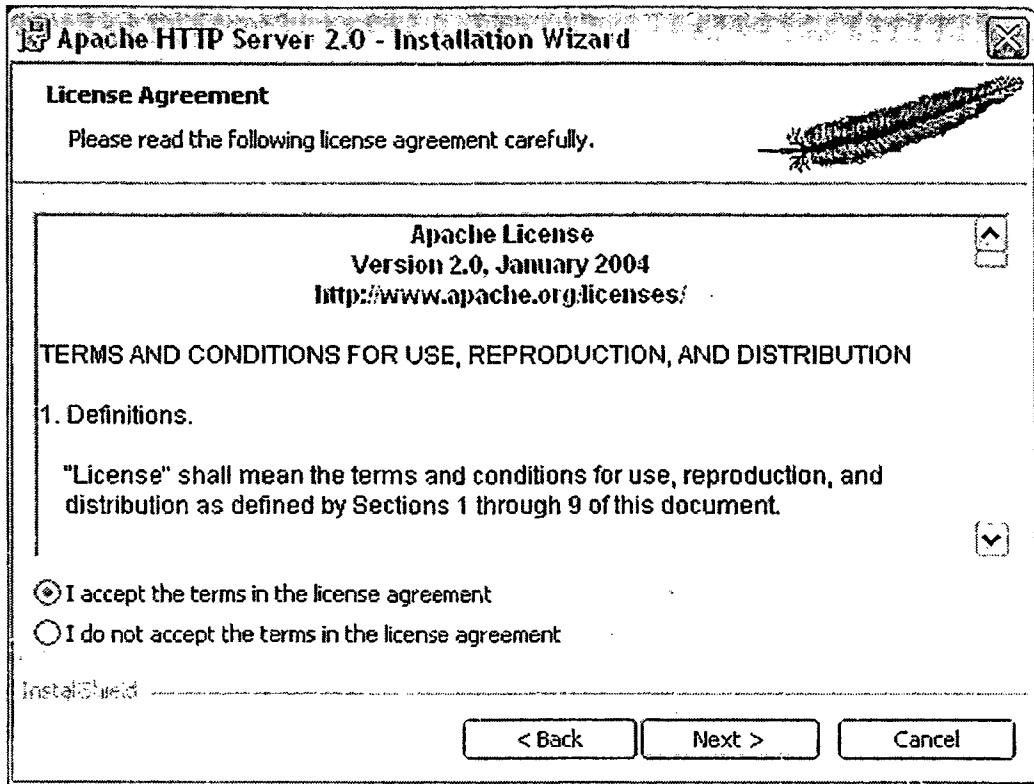
---

Τρέχουμε το αρχείο *apache\_2.0.52-win32-x86-no\_ssl.msi* .

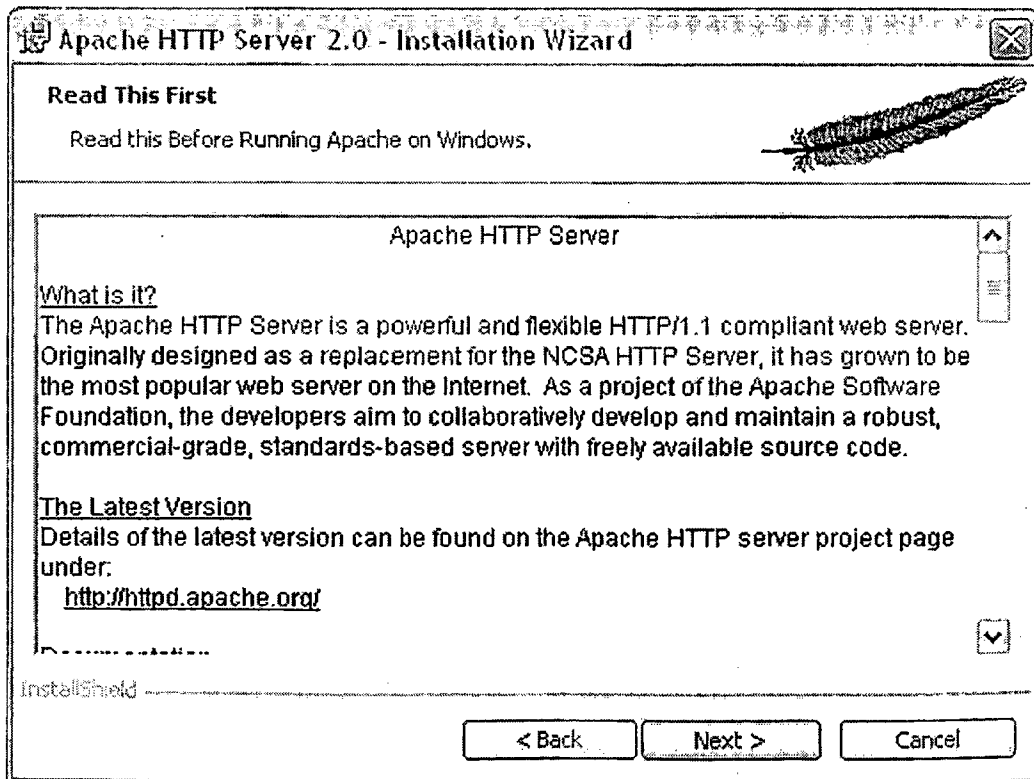
Εμφανίζεται η παρακάτω εικόνα :



Πατάμε next



Επιλέγουμε αυτό που φαίνεται και πατάμε next ..



Πατάμε next.

**Apache HTTP Server 2.0 - Installation Wizard**

**Server Information**  
Please enter your server's information.

Network Domain (e.g. somenet.com)

Server Name (e.g. www.somenet.com):

Administrator's Email Address (e.g. webmaster@somenet.com):

Install Apache HTTP Server 2.0 programs and shortcuts for:

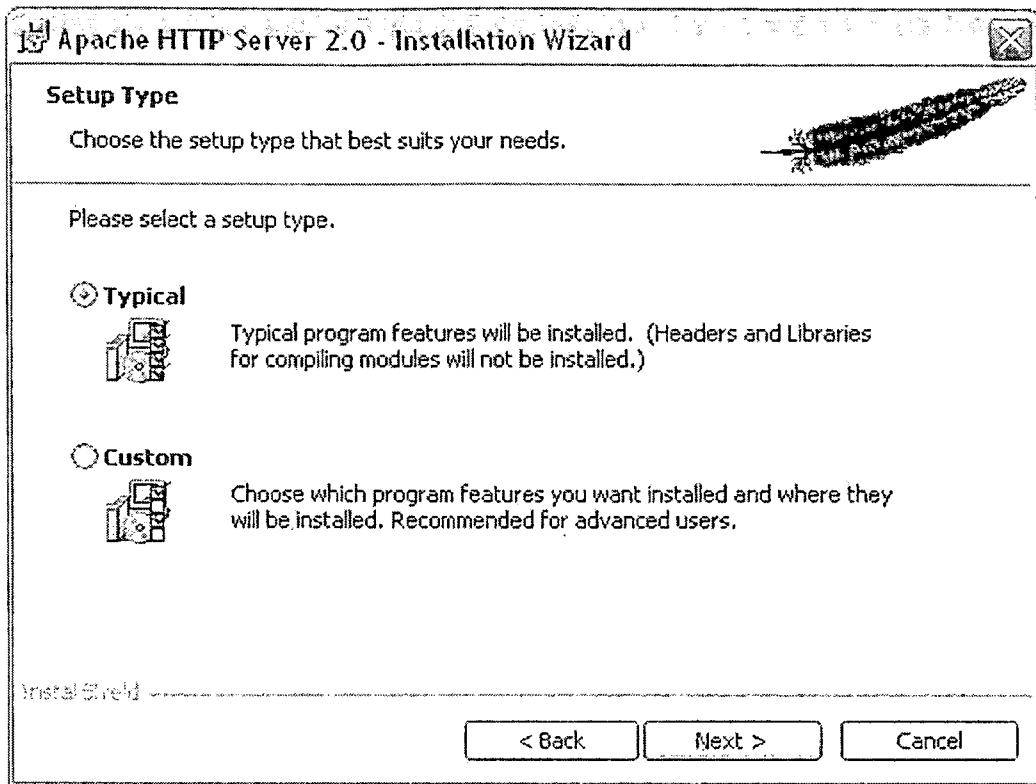
for All Users, on Port 80, as a Service -- Recommended.

only for the Current User, on Port 8080, when started Manually.

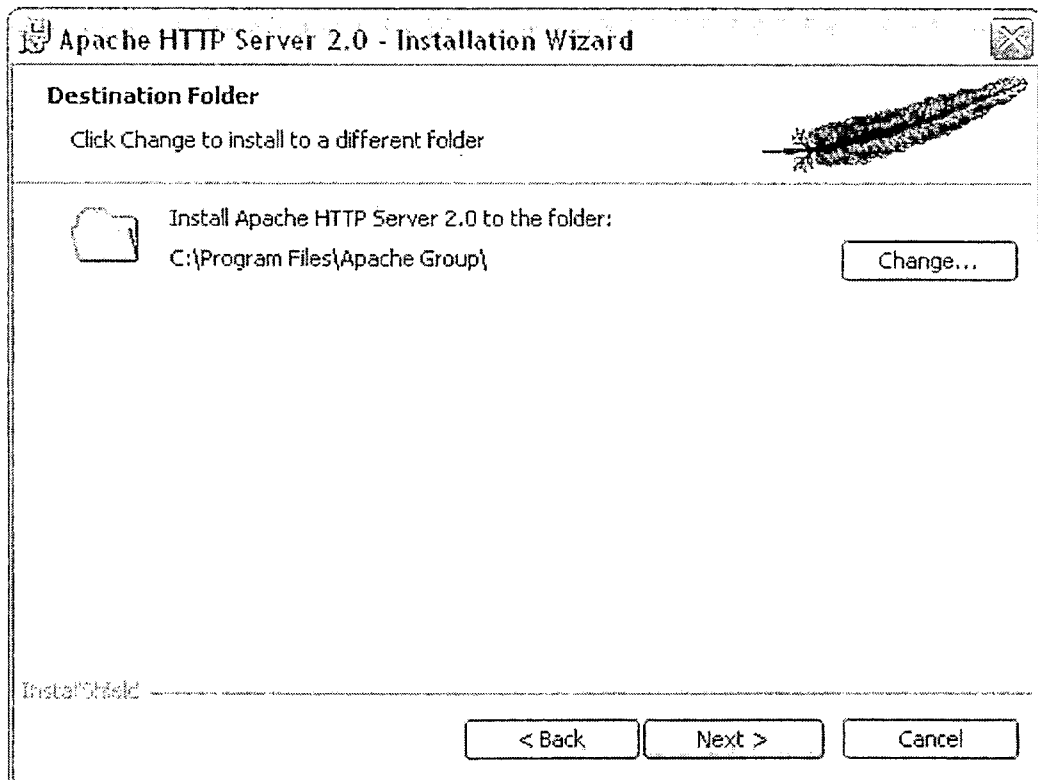
InstallShield

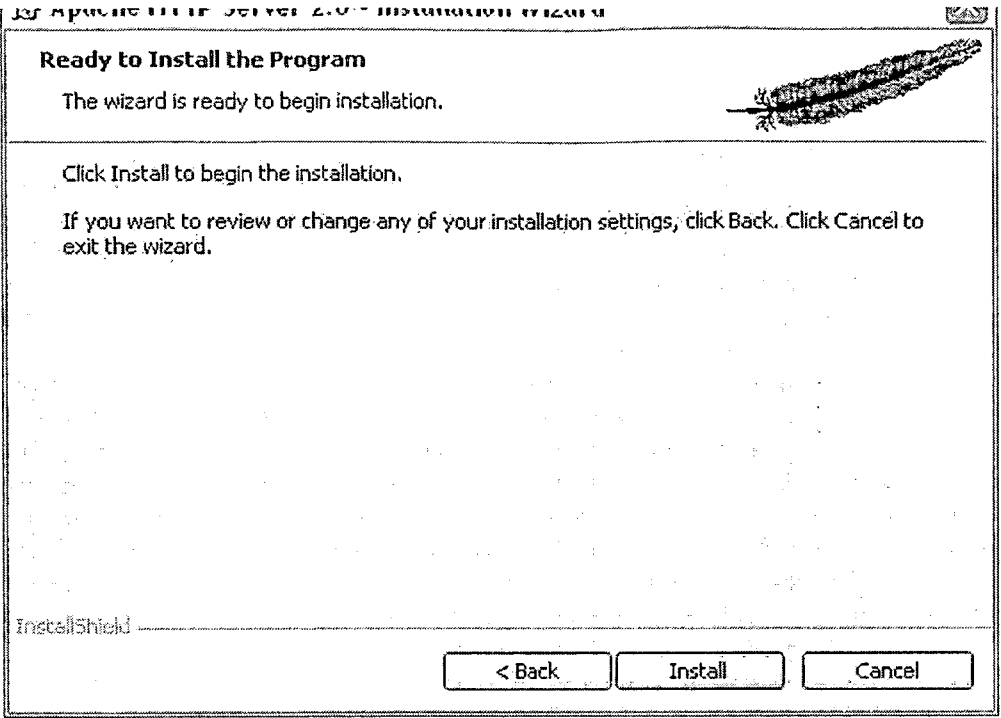
< Back    Next >    Cancel

Επιλέγουμε for All Users, on port 80, as a service --Recommended και πατάμε next .  
Επίσης βάζουμε τα ονόματα της αρεσκείας μας.



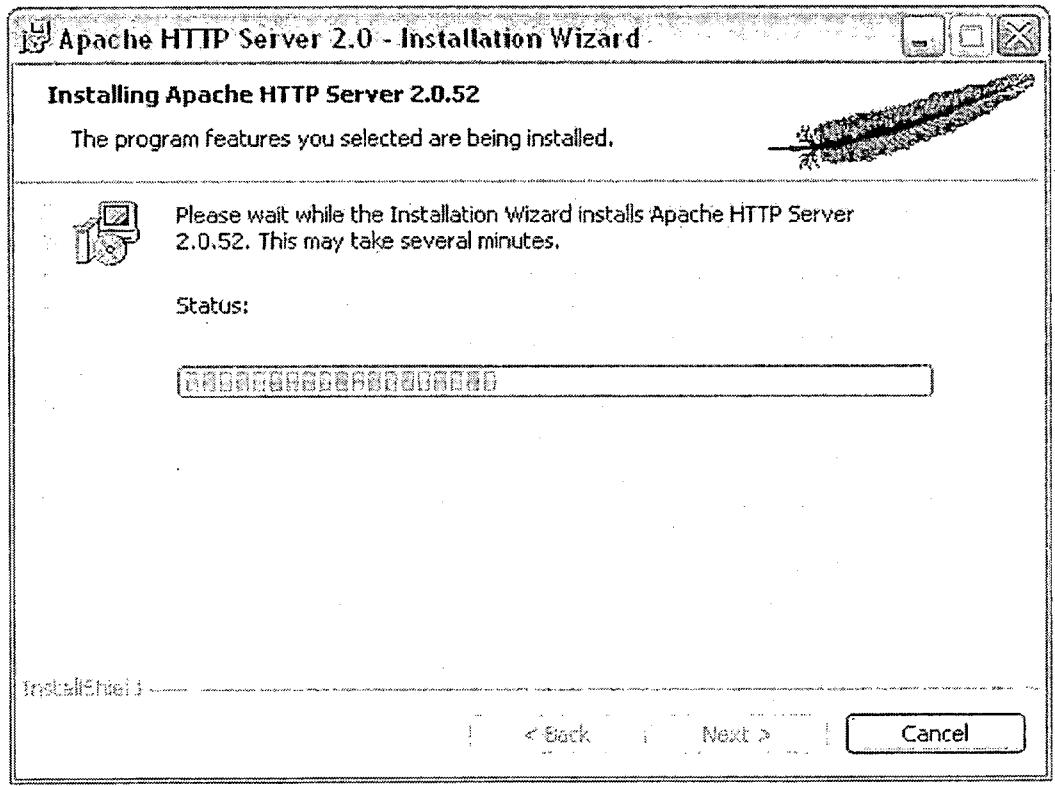
Επιλέγουμε Typical και πατάμε next .

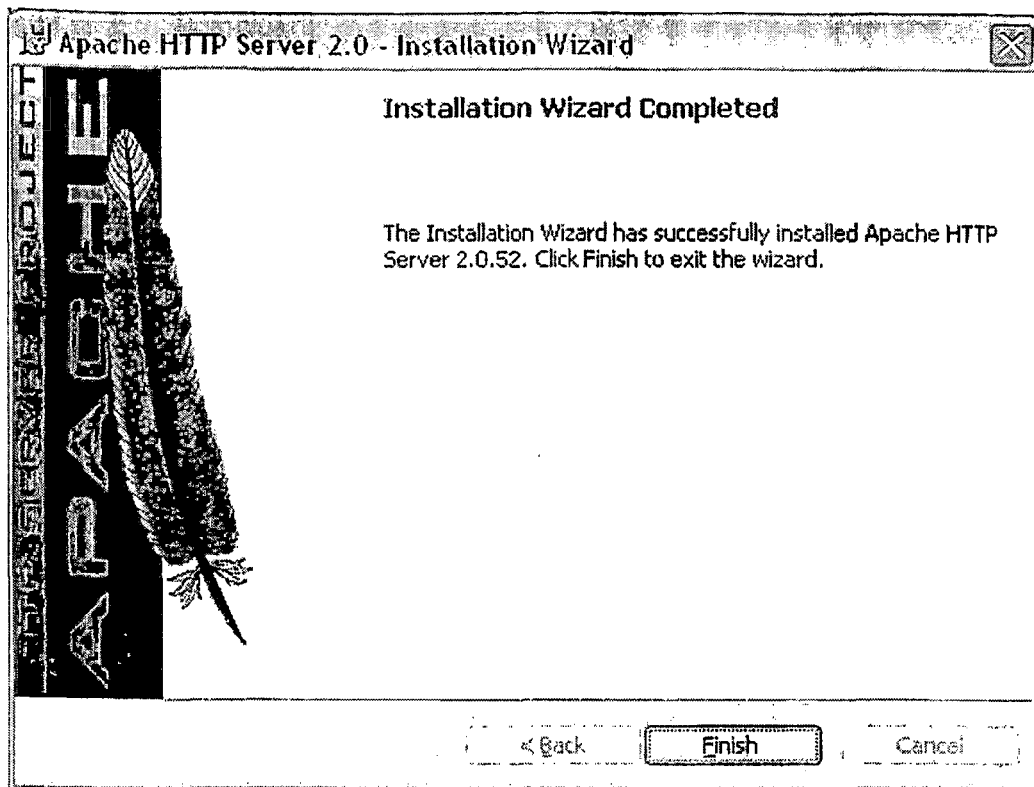




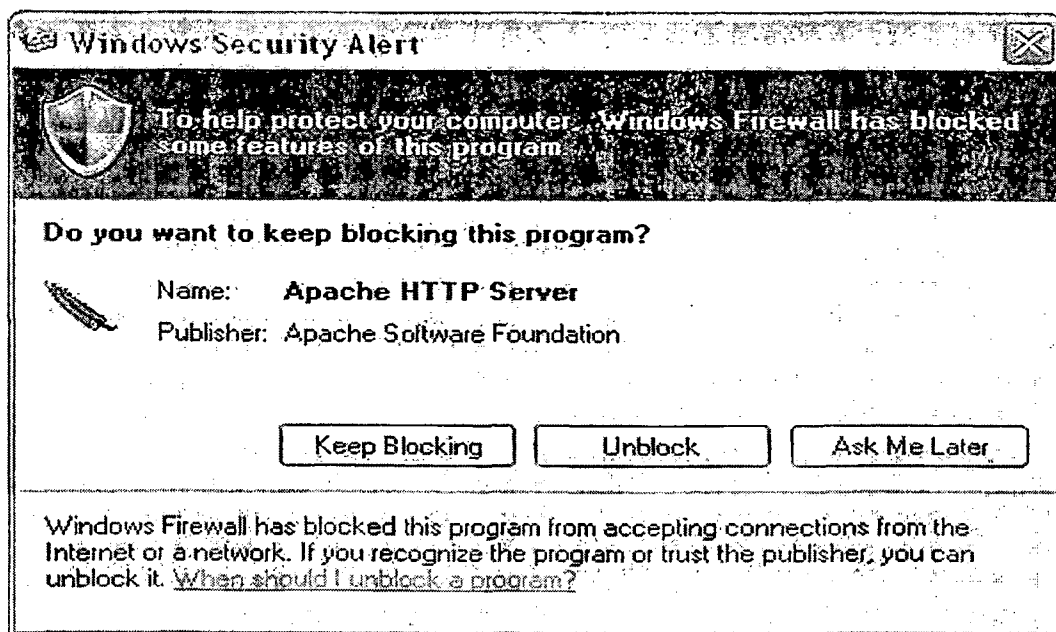
Επιλέγουμε next .

Πατάμε install για να αρχίσει η εγκατάσταση.





Πατάμε finish για να τελειώσει η εγκατάσταση.



Αν δούμε το παραπάνω σημαίνει ότι κάποιο firewall είναι ανοιχτό και δεν αφήνει τον apache να λειτουργήσει . Αν έχουμε το firewall των windows τότε στο παραπάνω πατάμε Unblock.



Αντιγράφουμε και στη συνέχεια μετονομάζουμε το αρχείο **c:\php\php.ini-recommended** σε **c:\php\php.ini**.

Στη συνέχεια ανοίγουμε το αρχείο php.ini αλλάζουμε τα παρακάτω και μετά σώζουμε το αρχείο.

```
include_path          =          ".;C:\php\includes\"
session.save_path     =          "C:\Temp"
session.cookie_path   =          \
doc_root              =          "C:\Program Files\Apache Group\Apache2\htdocs"
extension_dir         =          "C:\php\ext\"
extension=php_mysql.dll
extension=php_mysql.dll
mysql.default_port    =          3306
mysql.default_host    =          localhost
mysql.default_user    =          root
mysqli.default_port = 3306

mysqli.default_host = localhost

mysqli.default_user = root
```

Μετά θα προσθέσουμε το C:/php στο PATH των System Enviromental Variables.

Κάνουμε δεξί κλικ στο My Computer επιλέγουμε properties.

Στη συνέχεια πηγαίνουμε στην καρτέλα Advanced .

Εκεί πατάμε Enviromental Variables.

Στο System Variables επιλέγουμε PATH και πηγαίνουμε στο edit

Πηγαίνουμε τον κέρσορα στο τέλος αυτού που υπάρχει και προσθέτουμε το  ;C:/php

Θα πρέπει να μοιάζει με αυτό :

```
%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\ph
```

␣



**Σημείωση :** Αν δεν θέλουμε να προσθέσουμε το C:/php απλά να αντιγράψουμε το *libmysql.dll*

στο *C:/Windows/System* .

Μετά προσθέτουμε τις ακόλουθες γραμμές στο *httpd.conf* .

**Σημείωση :** Το *httpd.conf* βρίσκεται στο *C:\Program Files\Apache Group\Apache2\conf\*

```
LoadModule php5_module "c:/php/php5apache2.dll"
```

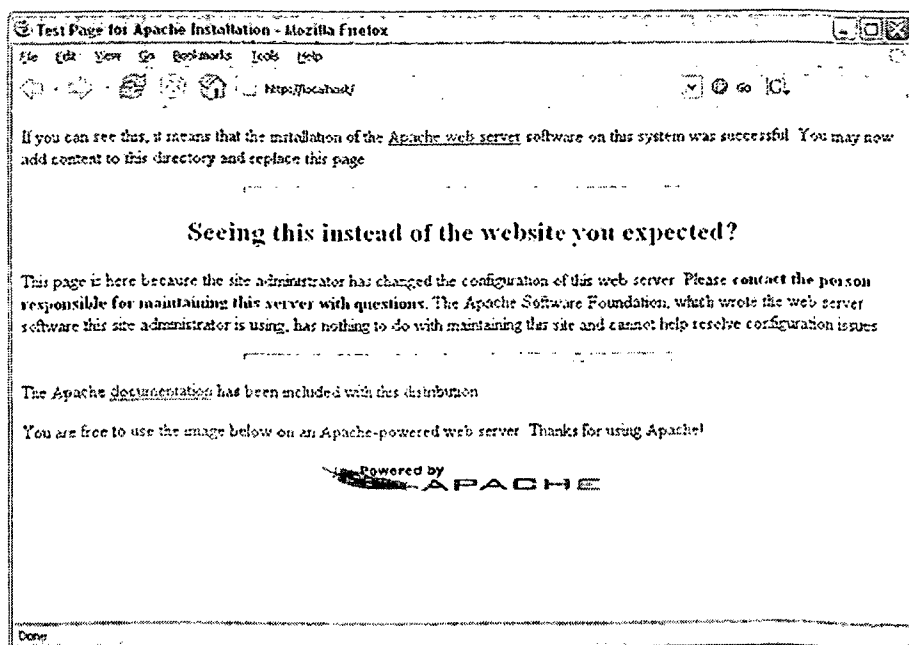
```
AddType application/x-httpd-php .php
```

```
PHPIIniDir "C:/php"
```

Μετά αλλάζουμε την ακόλουθη γραμμή στο *httpd.conf* .

1. Από:  
*DirectoryIndex index.html index.html.var*
2. Σε:  
*DirectoryIndex index.php index.jsp index.html index.html.var*

Τώρα θα κάνουμε ένα τεστ . Κάνουμε restart στον apache και ανοίγουμε σε έναν web browser το http://localhost ελπίζοντας να δεις ότι και την προηγούμενη φορά.



Τώρα θα τεστάρουμε αν δουλεύει η php που εγκαταστήσαμε.

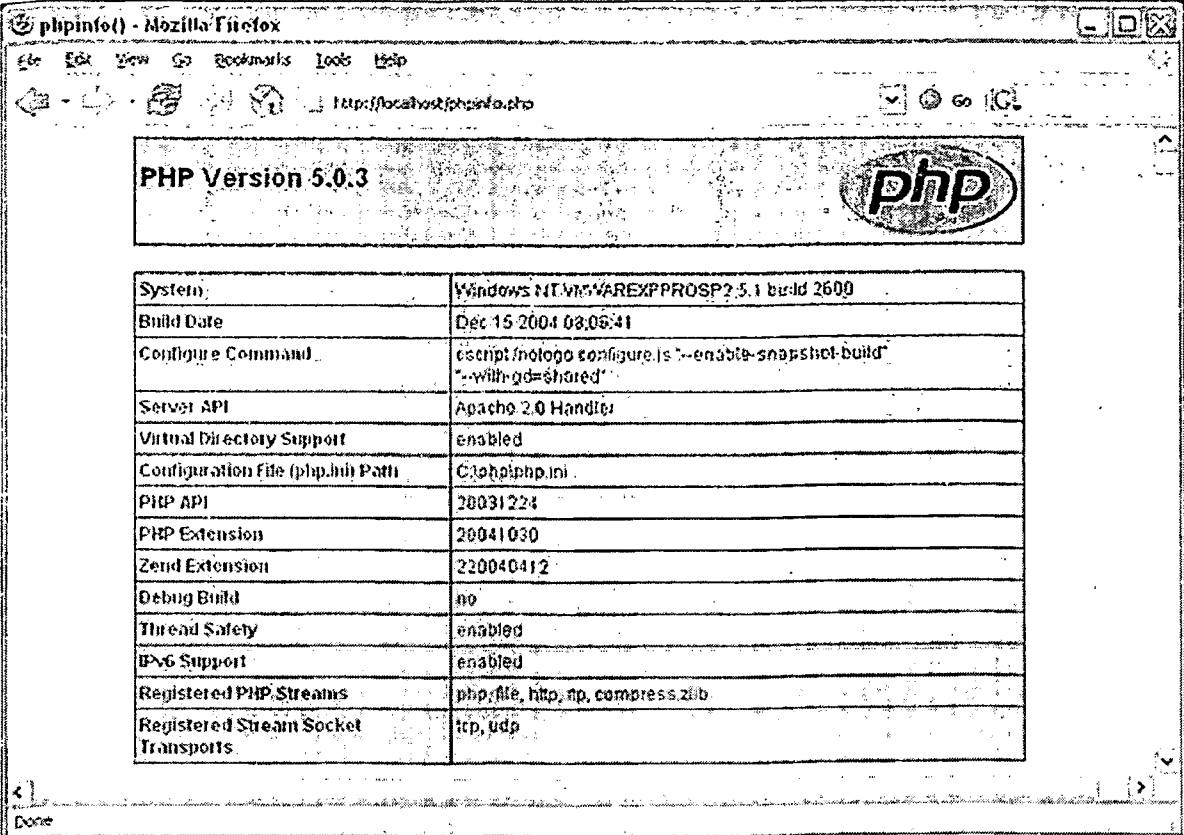
Δημιουργούμε λοιπόν ένα αρχείο txt και το ονομάζουμε **phpinfo.php**

Γράφουμε τα παρακάτω μέσα στο αρχείο : **<?php phpinfo();?>**.

Σώζουμε το αρχείο μέσα στο **C:\Program Files\Apache Group\Apache2\htdocs**.

Τώρα ανοίγουμε σε ένα web browser το παρακάτω : <http://localhost/phpinfo.php> .

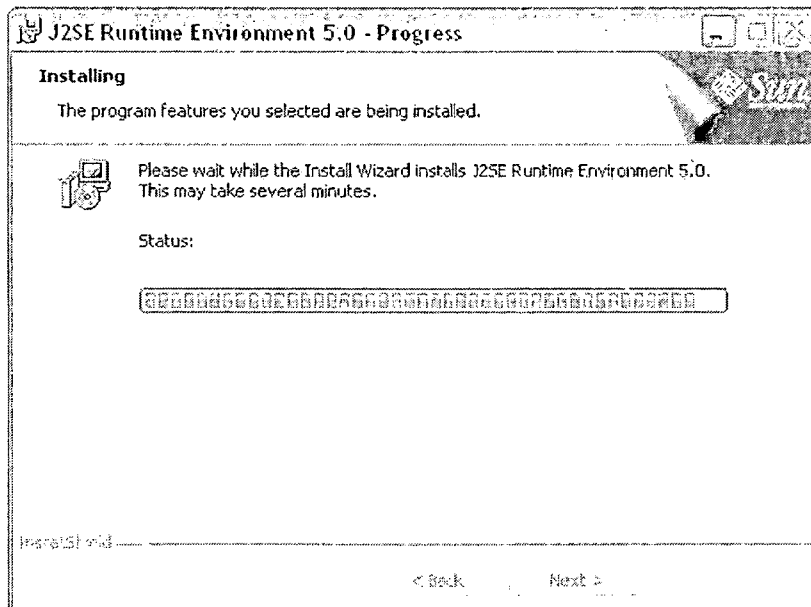
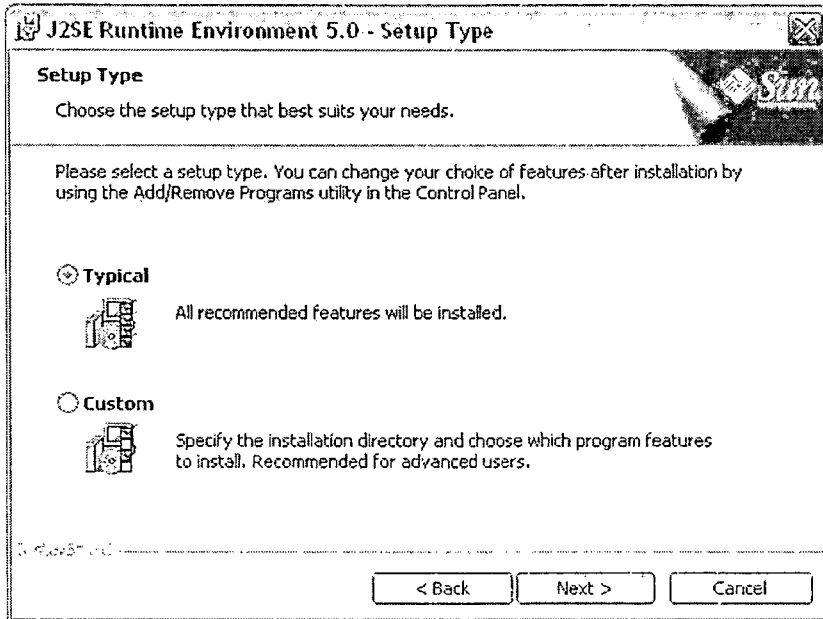
Θα πρέπει να εμφανίσει κάτι σαν το παρακάτω :



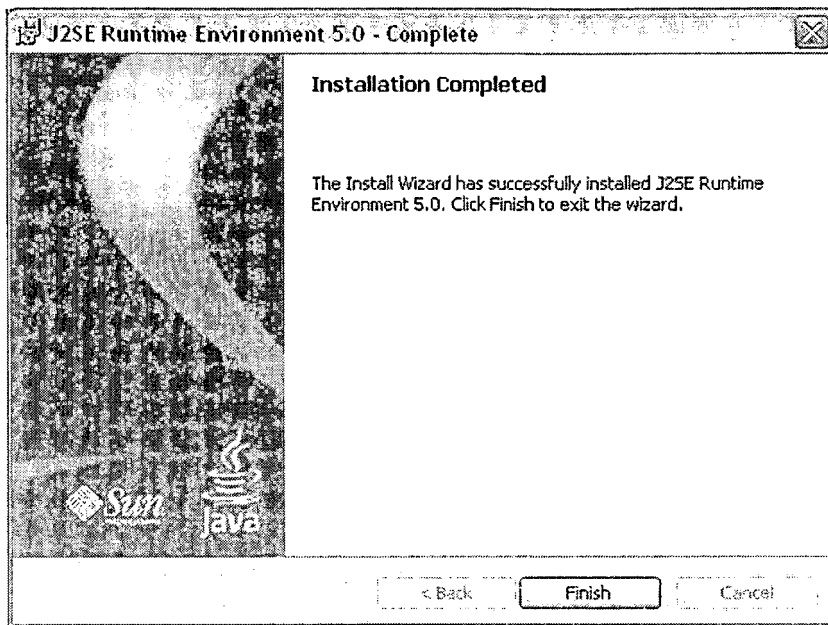
System	Windows NT/MS/AREXP/PROSP? 5.1 build 2600
Build Date	Dec 15 2004 08:08:41
Configure Command	escript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration file (php.ini path)	C:\php\php.ini
PHP API	20031224
PHP Extension	20041030
Zend Extension	220040412
Debug Build	no
Thread Safety	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, http, ftp, compress, zlib
Registered Stream Socket Transports	tcp, udp



Πατάμε next.

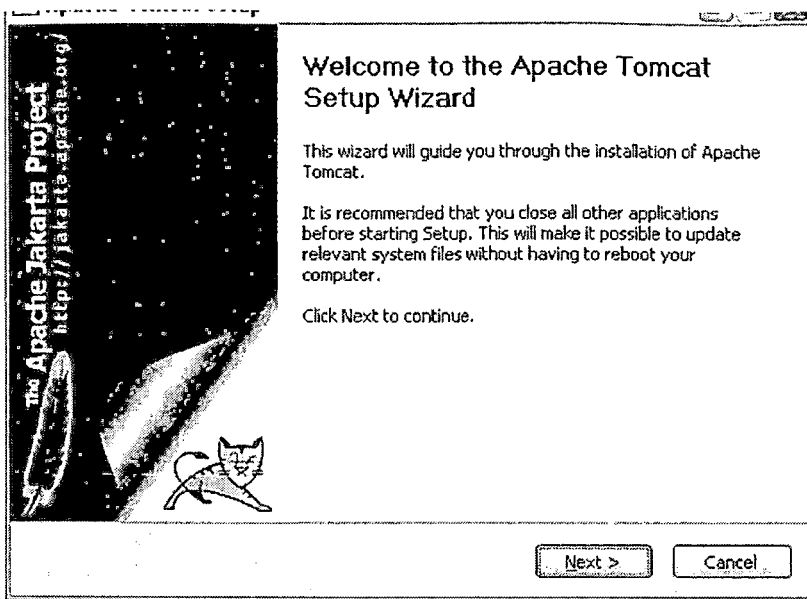


Επιλέγουμε Typical και πατάμε next.

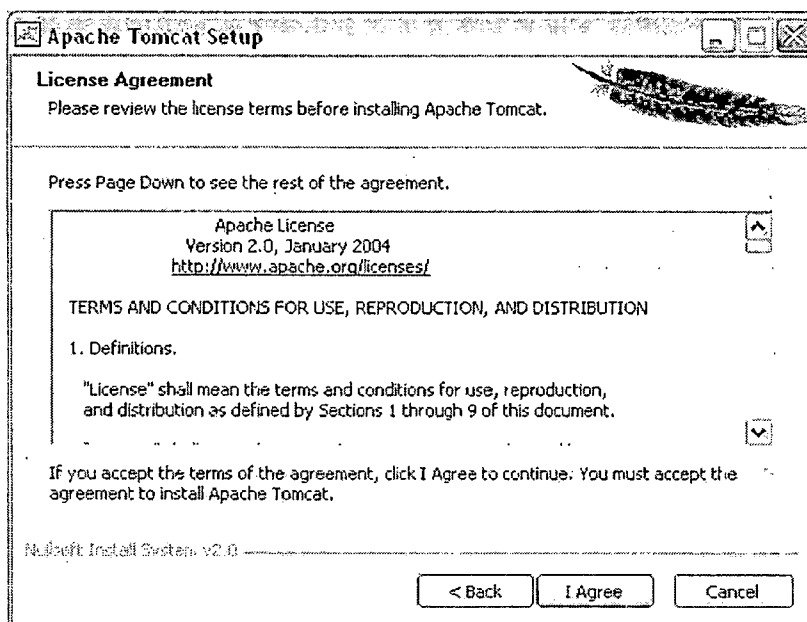


Πατάμε finish για να τελειώσει η εγκατάσταση.

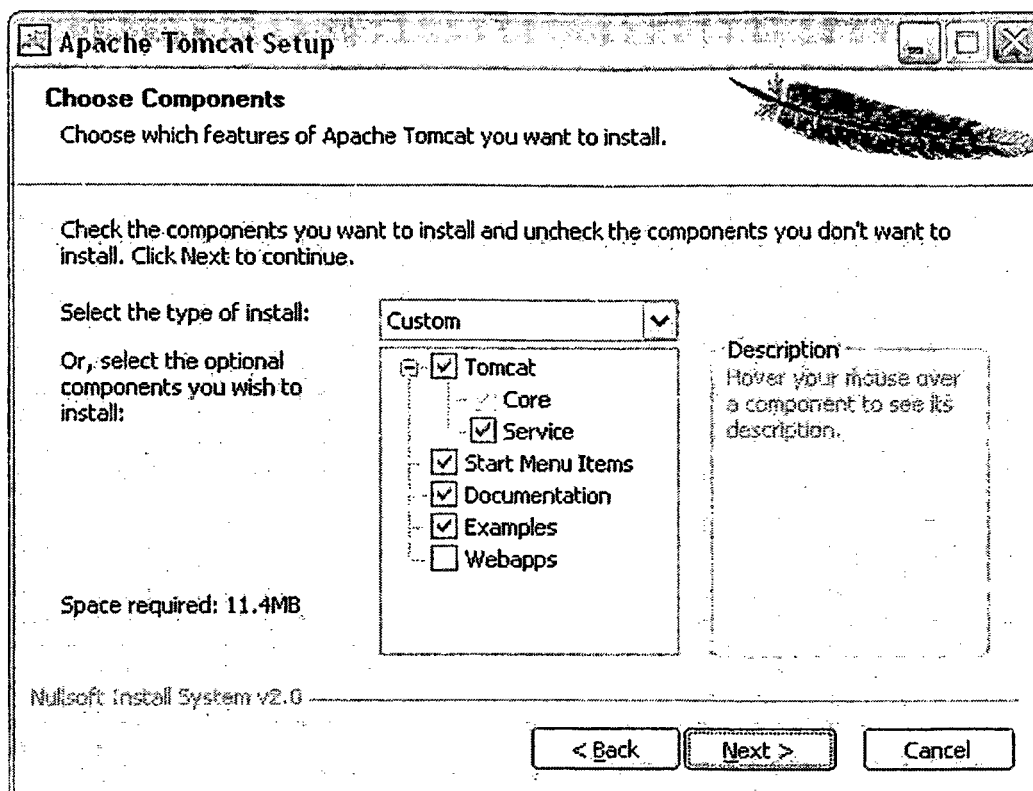
Μετά τρέχουμε το αρχείο *jakarta-tomcat-5.5.4.exe*



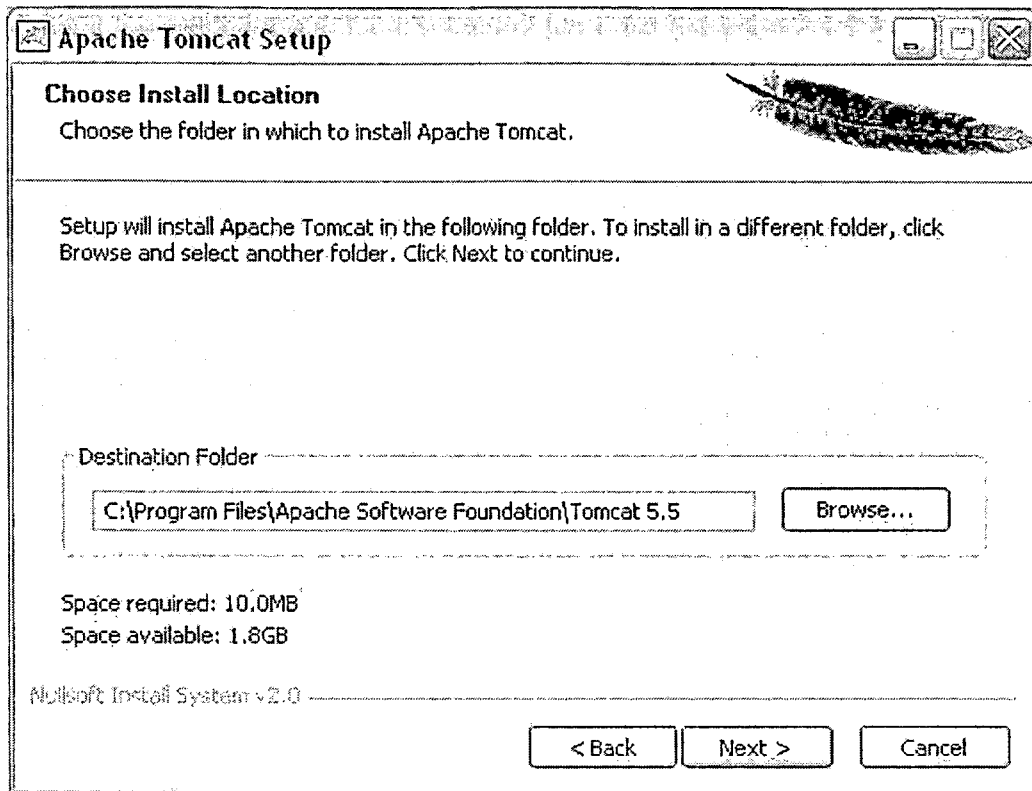
Πατάμε next.



Πατάμε I Agree.



Εδώ επιλέγουμε αυτά που φαίνονται το service για ξεκινάει ο tomcat μαζί με τα windows και το examples για να κάνουμε μετά κάποια τεστ ώστε να βεβαιωθούμε αν όλα όσα κάναμε είναι σωστά. Μετά πατάμε Next .



Πατάμε Next .



Apache Tomcat Setup: Configuration Options

**Configuration**  
Tomcat basic configuration.

HTTP/1.1 Connector Port

Administrator Login

User Name

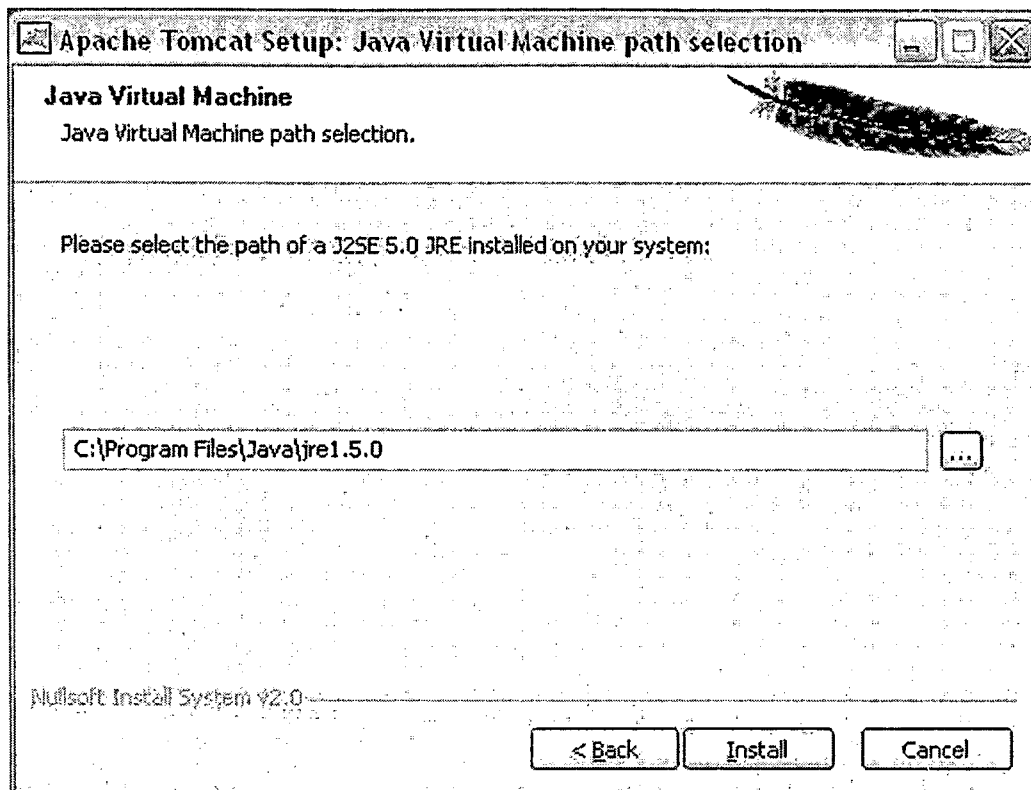
Password

Nullsoft Install System v2.0

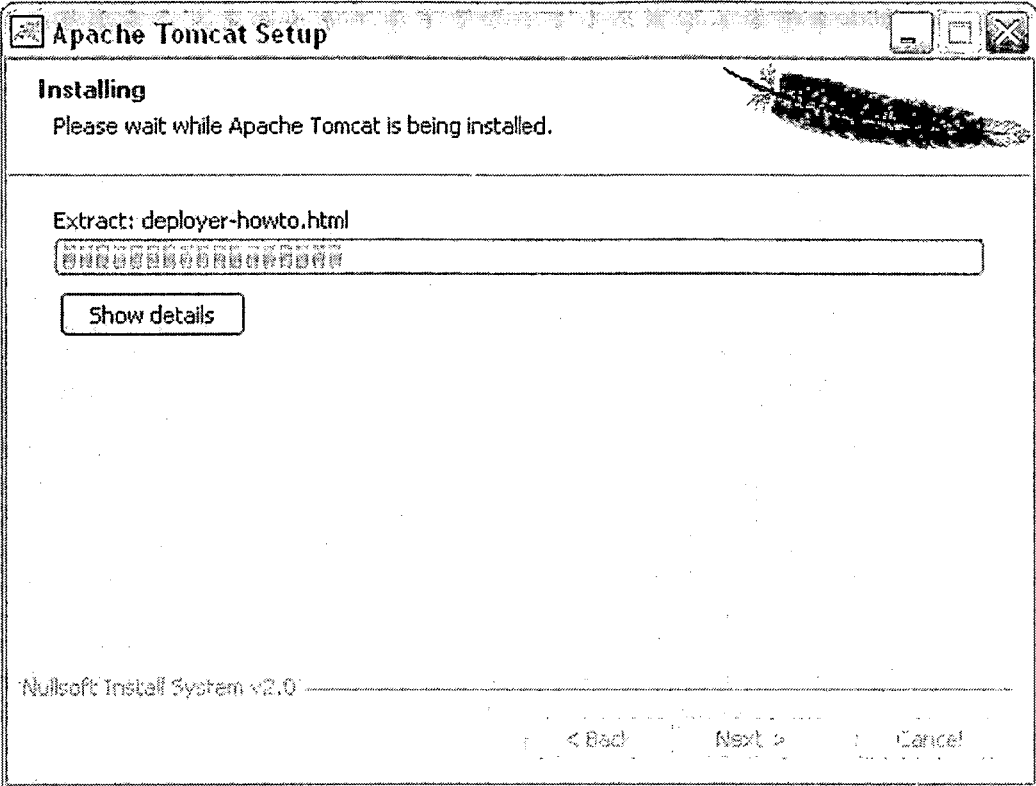
< Back    Next >    Cancel

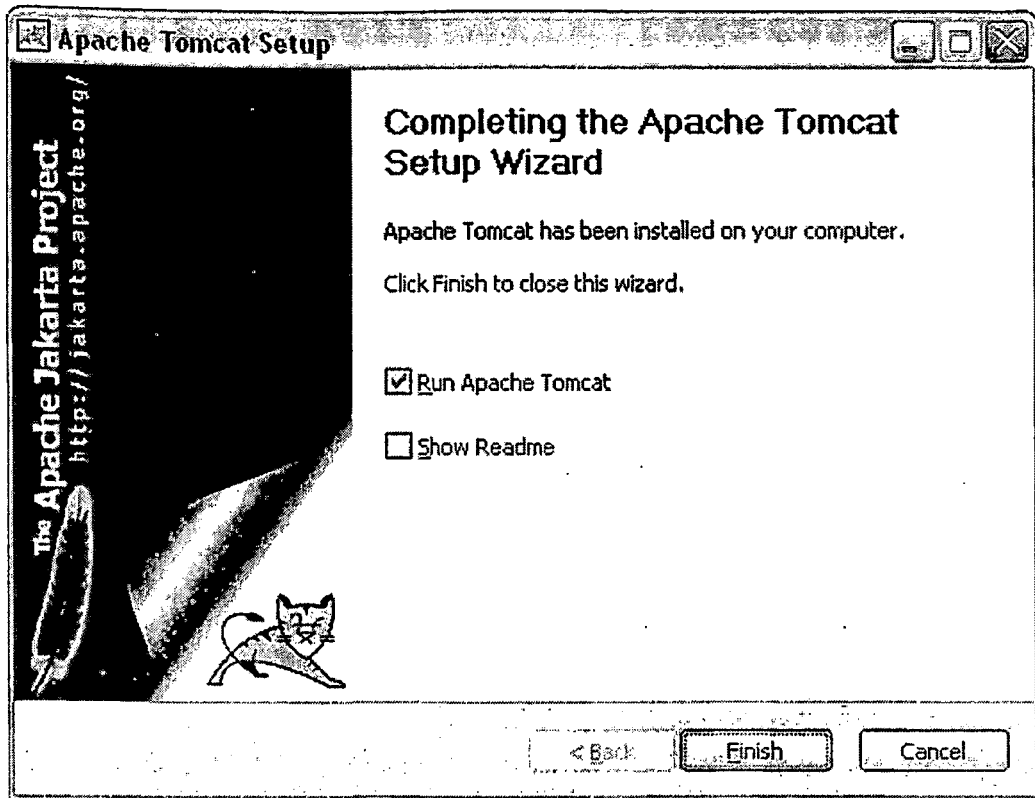
Αφήνουμε το port και το user name έτσι ως έχουν και απλά βάζουμε ένα κωδικό στο password .

Πατάμε Next .



Πατάμε install για να ξεκινήσει η εγκατάσταση .

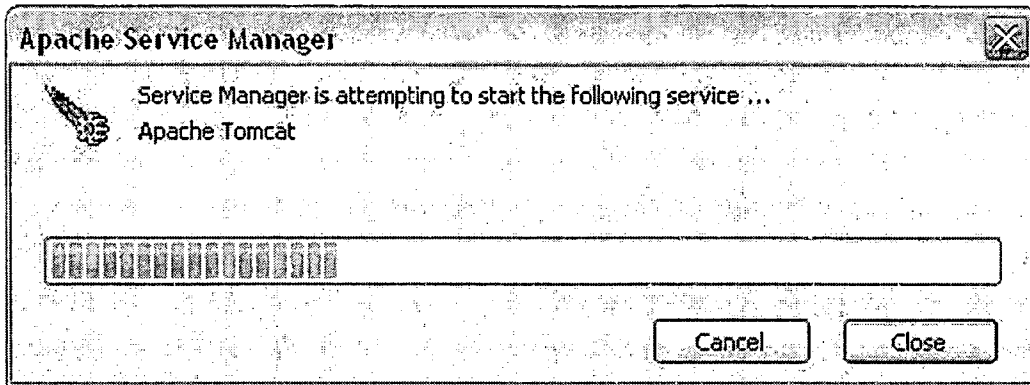




Πατάμε Finish .

Σειρά έχει ένα τεστ .

Ξεκινάμε τον tomcat εκτός και αν έχουμε πατήσει το Run Apache Tomcat οπότε θα ξεκινήσει αυτόματα .

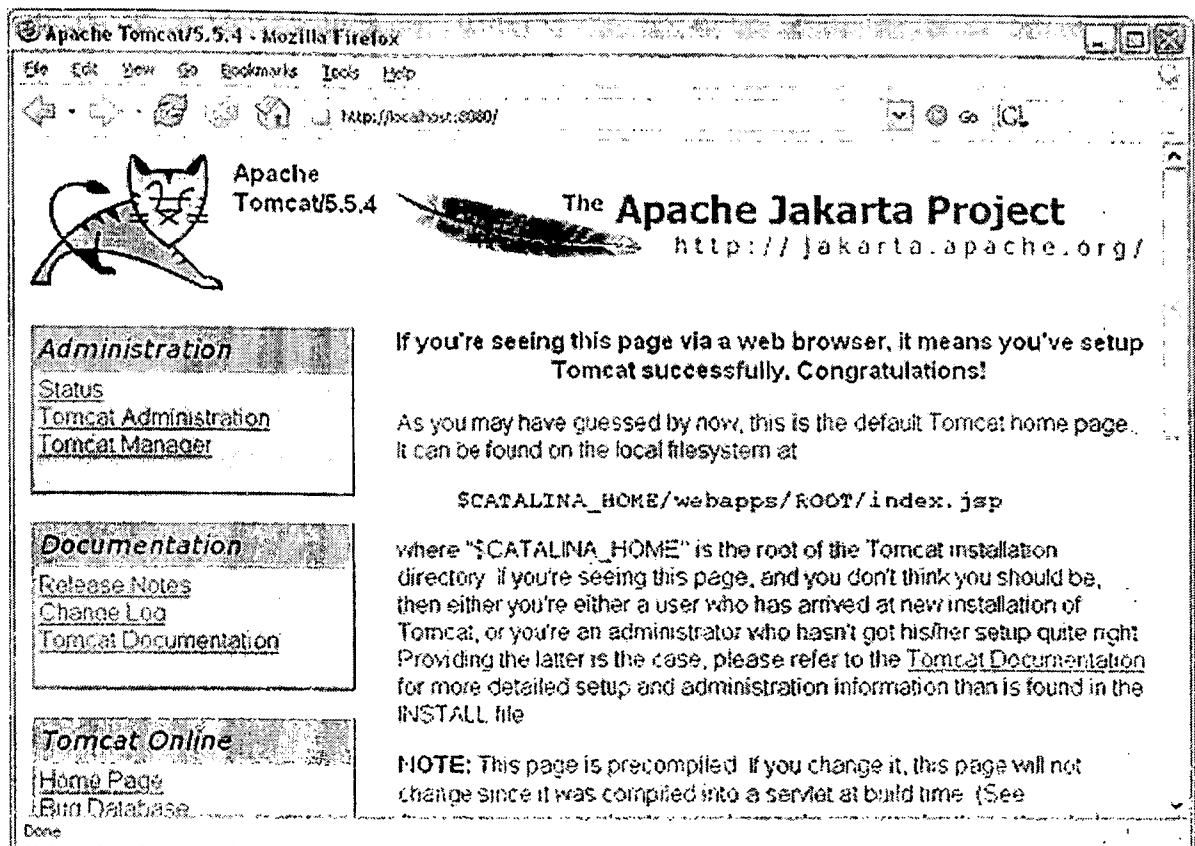


Θα πρέπει να βλέπουμε το παρακάτω στο system tray :



δηλαδή ένα ο apache ένα ο tomcat .

Μετά ανοίγουμε σε ένα web browser το ακόλουθο <http://localhost:8080> και θα πρέπει να δούμε το παρακάτω :



Αποσυμπιέζουμε το `mod_jk2.so` από το `jakarta-tomcat-connectors-jk2.0.4-win32-apache2.0.49.zip` στο `C:\Program Files\Apache Group\Apache2\modules` .

Μετά στο αρχείο `http.conf` προσθέτουμε το παρακάτω στο section "Load Modules" :

**`LoadModule jk2_module modules/mod_jk2.so`**

Στη συνέχεια δημιουργούμε ένα νέο text αρχείο και το ονομάζουμε **`workers2.properties`**

και έπειτα το σώζουμε στο `C:\Program Files\Apache Group\Apache2\conf` .

Βάζουμε το ακόλουθο κείμενο μέσα στο αρχείο :

[shm]

file=C:\Program

Files\Apache

Group\Apache2\logs\jk2.shm

size=1048576

[channel.socket:localhost:8009]

port=8009

host=127.0.0.1

[ajp13:localhost:8009]

channel=channel.socket:localhost:8009

[status:status]

[uri:/jsp-examples/\*]

worker=ajp13:localhost:8009

[uri:/status/\*]

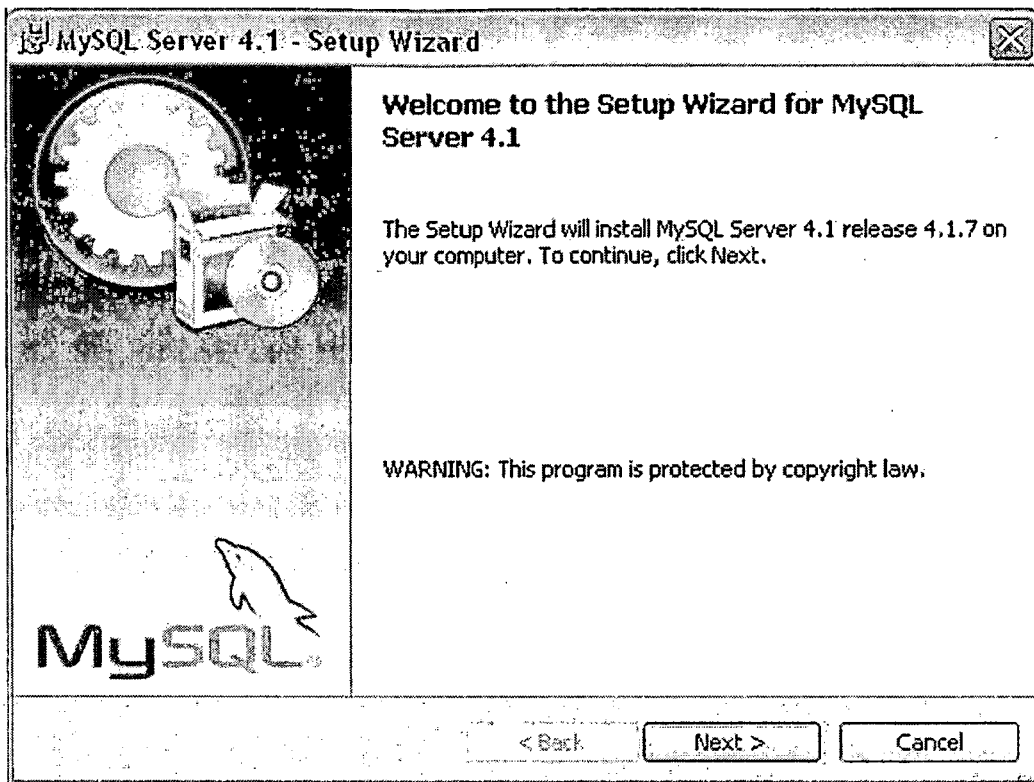
worker=status:status

Μετά στο αρχείο **jk2.properties** που βρίσκεται στο **C:\Program Files\Apache Software Foundation\Tomcat 5.5\conf\** προσθέτουμε τα ακόλουθα στο τέλος του αρχείου :

handler.list=channelSocket,request

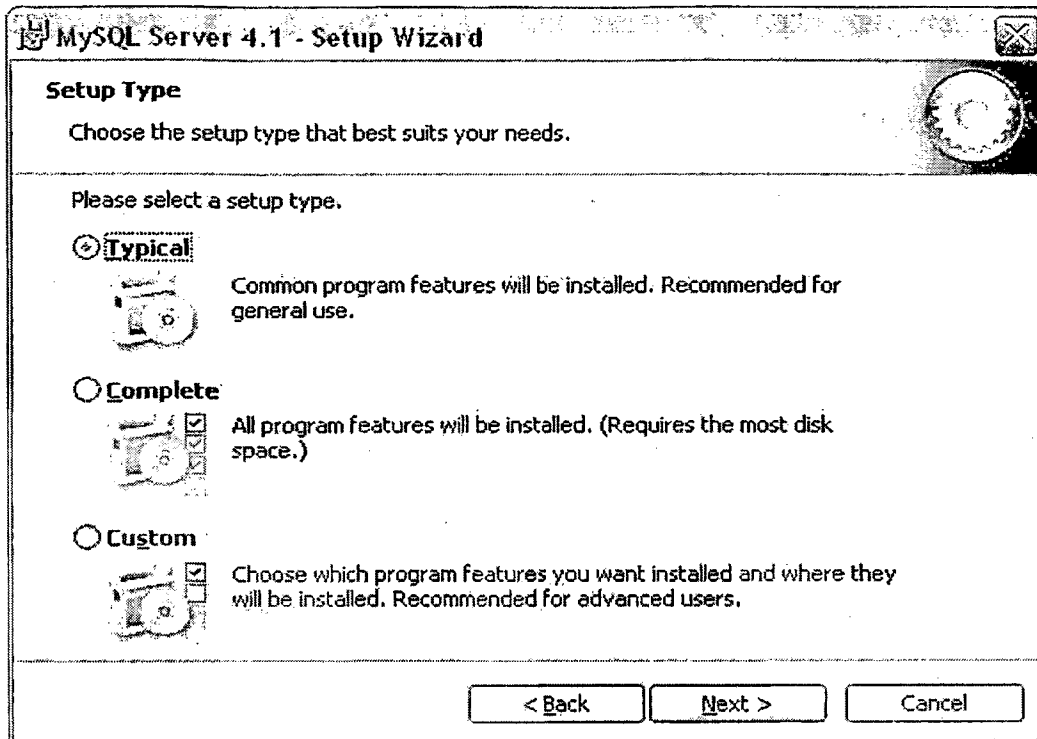
channelSocket.port=8009

channelSocket.address=127.0.0.1

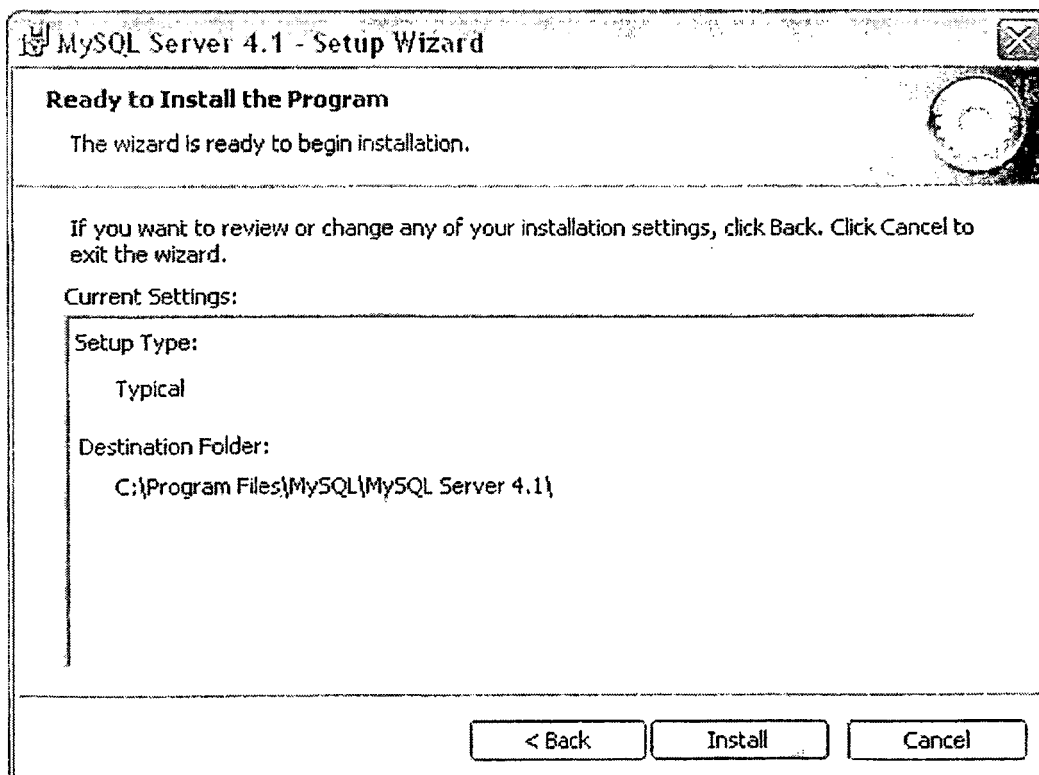


Πατάμε next .

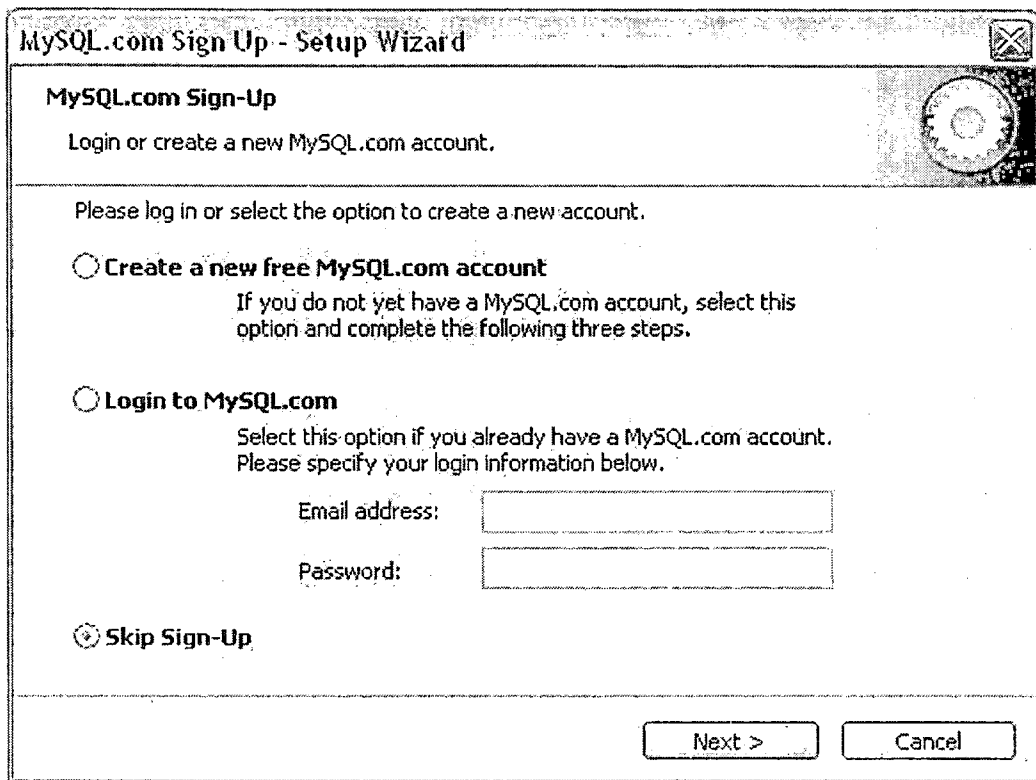




Επιλέγουμε typical και πατάμε next .

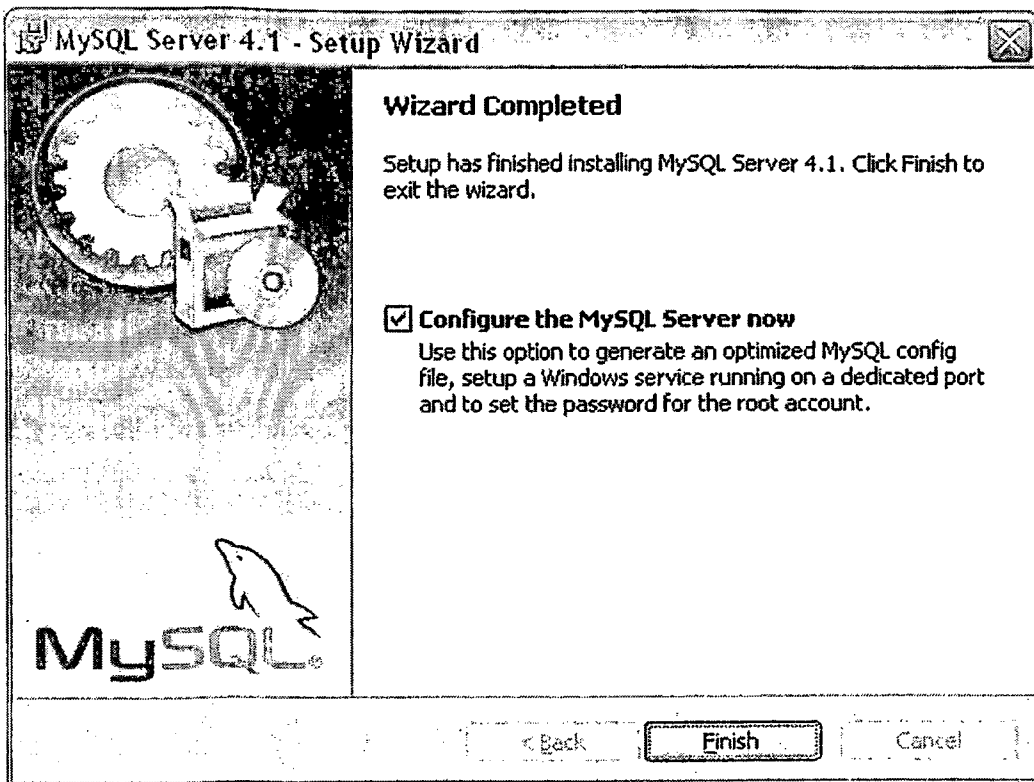


Πατάμε install .

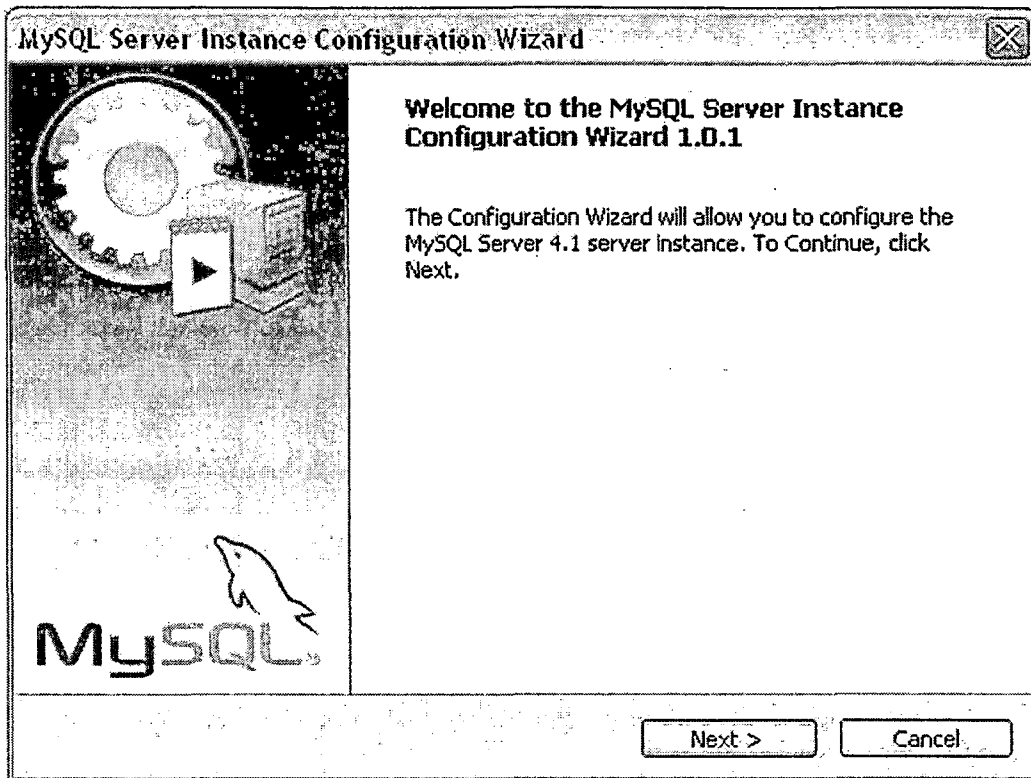


The image shows a Windows-style dialog box titled "MySQL.com Sign Up - Setup Wizard". The main heading is "MySQL.com Sign-Up" with the instruction "Login or create a new MySQL.com account." Below this, it says "Please log in or select the option to create a new account." There are three radio button options: "Create a new free MySQL.com account" (with a sub-instruction: "If you do not yet have a MySQL.com account, select this option and complete the following three steps."), "Login to MySQL.com" (with a sub-instruction: "Select this option if you already have a MySQL.com account. Please specify your login information below."), and "Skip Sign-Up" (which is selected). The "Login to MySQL.com" option has two text input fields: "Email address:" and "Password:". At the bottom right, there are two buttons: "Next >" and "Cancel".

Επιλέγουμε Skip και πατάμε Next .

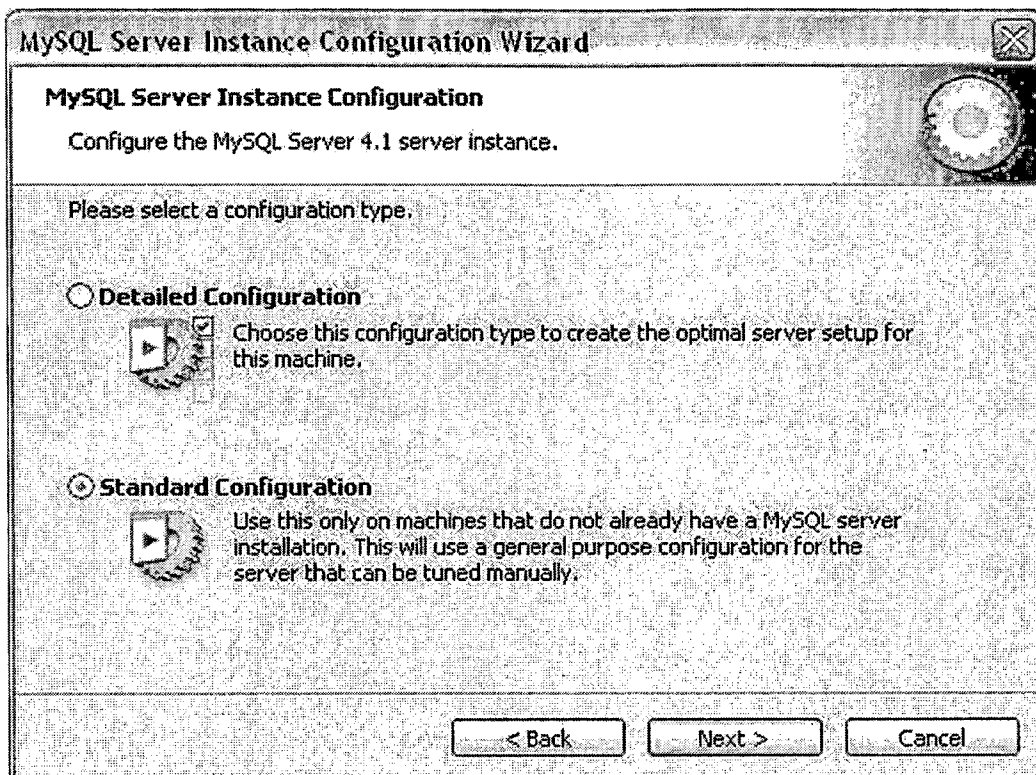


Επιλέγουμε το configure και πατάμε finish .

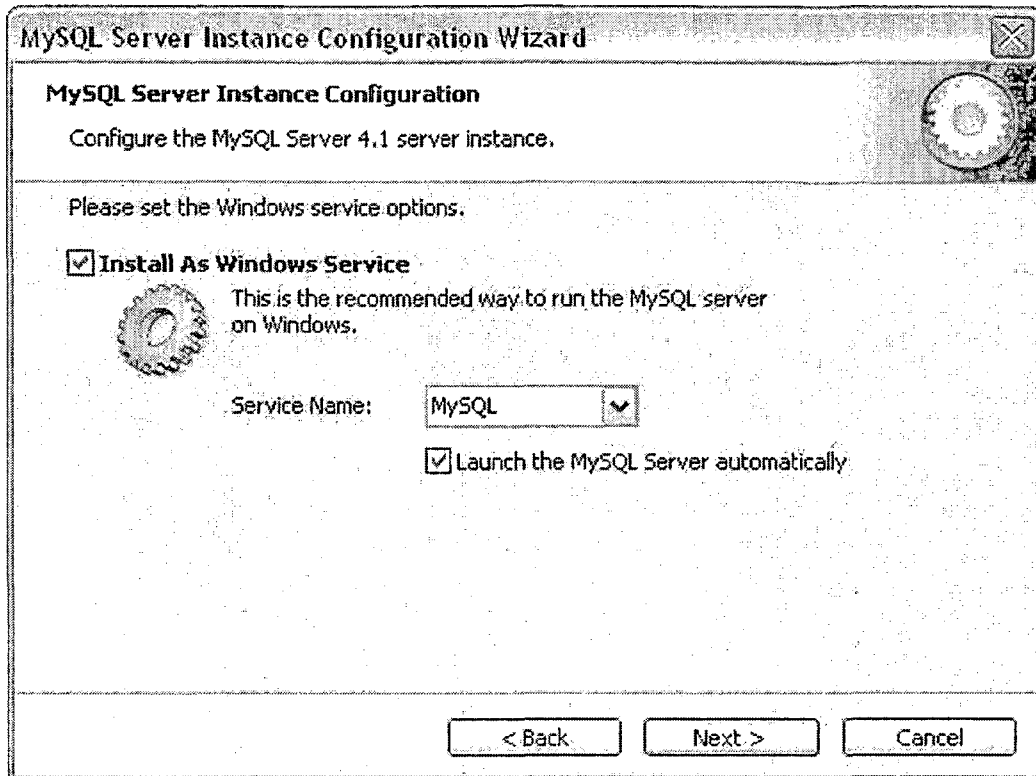


Αρχίζουμε την ρύθμιση του MySQL server .

Πατάμε Next ..



Επιλέγουμε Standard και πατάμε Next .



Κάνουμε τα πάντα όπως στη φωτογραφία και πατάμε Next .


**MySQL Server Instance Configuration Wizard**

**MySQL Server Instance Configuration**

Configure the MySQL Server 4.1 server instance.

Please set the security options.


**Modify Security Settings**

 **root** New root password:  Enter the root password.

Confirm:  Retype the password.

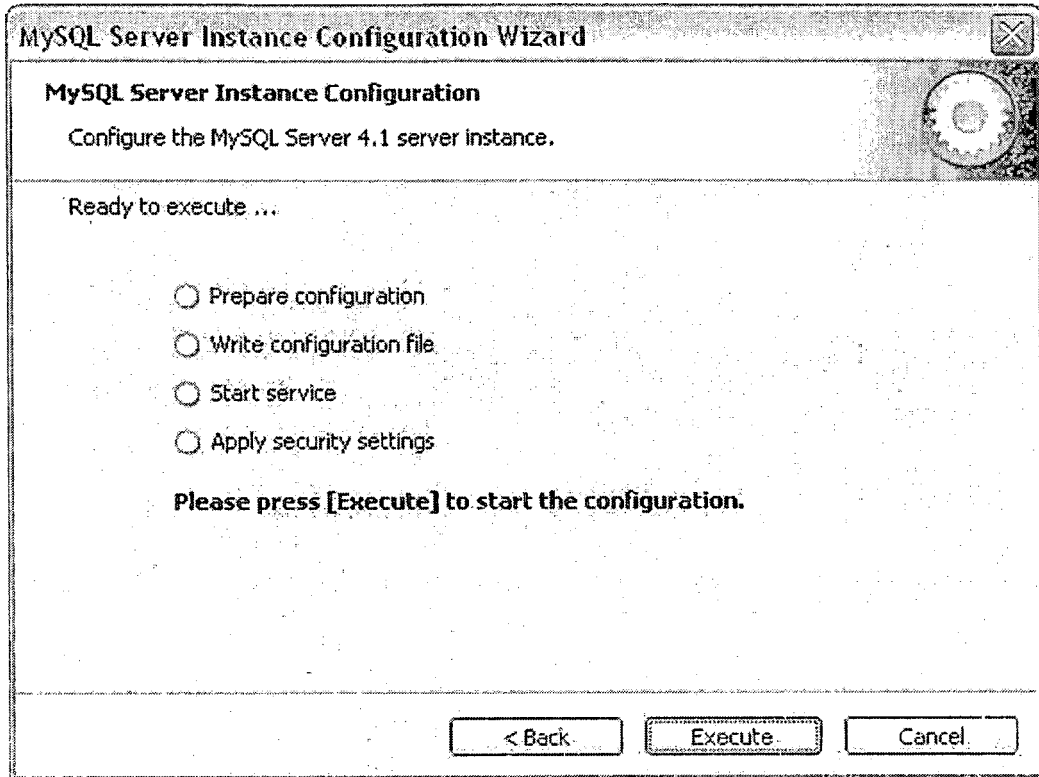
Root may only connect from localhost

**Create An Anonymous Account**

 This option will create an anonymous account on this server. Please note that this can lead to an insecure system.

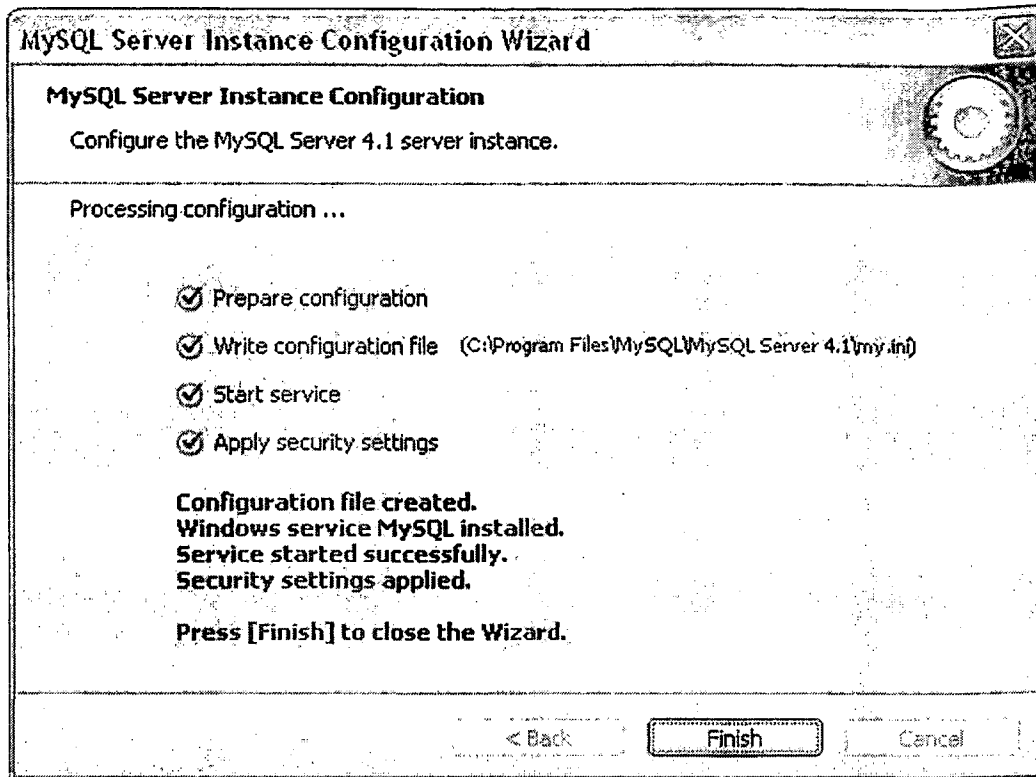
< Back    Next >    Cancel

Βάζουμε ένα κωδικό στο root και πατάμε next .



Πατάμε execute για να αρχίσει η ρύθμιση του server .





Πατάμε finish .

Τεστ : Ανοίγουμε τη γραμμή εντολών και φτάνουμε μέχρι το επίπεδο **C:\Program Files\MySQL\MySQL Server 4.1\bin\** .

Εκεί βάζουμε την εντολή **mysqlshow -u root -p** .

Βάζουμε τον κωδικό που δώσαμε πριν και πρέπει να μας εμφανίσει τις βάσεις δεδομένων που υπάρχουν :

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Jeff>cd C:\Program Files\MySQL\MySQL Server 4.1\bin
C:\Program Files\MySQL\MySQL Server 4.1\bin>mysqlshow -u root -p
Enter password: ****
+-----+
| Databases |
+-----+
| mysql    |
| test     |
+-----+

C:\Program Files\MySQL\MySQL Server 4.1\bin>_
```

## Εγκατάσταση ODBC

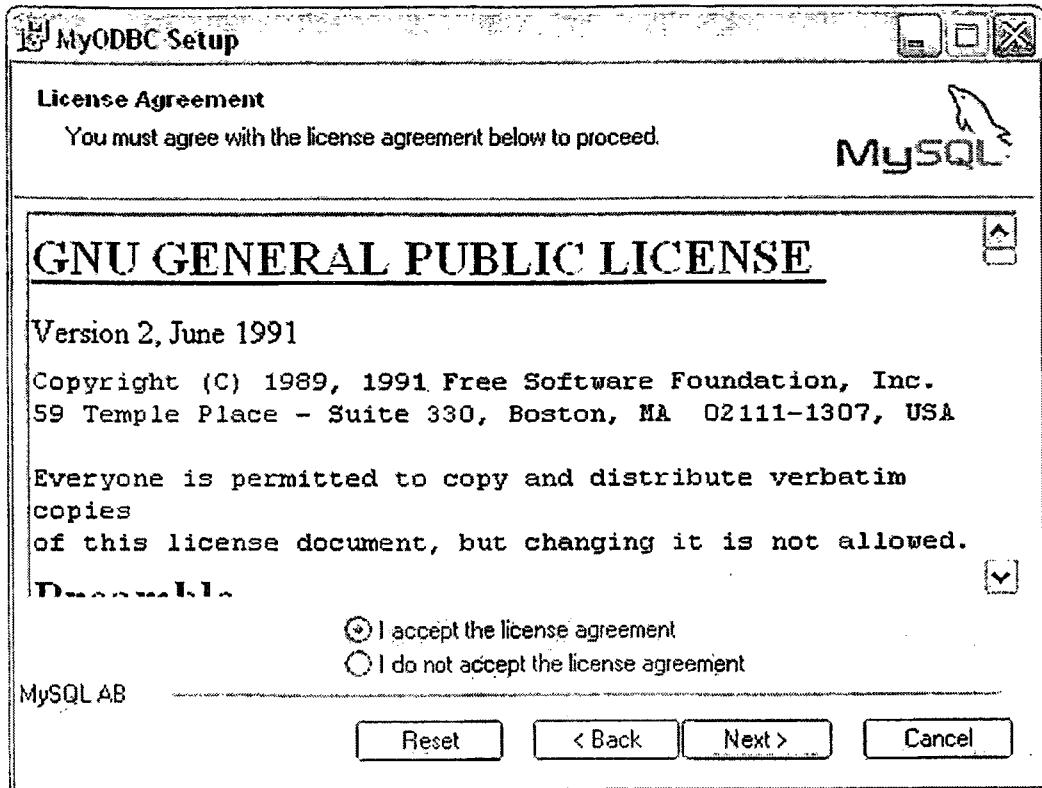
---

Τρέχουμε το **MyODBC-3.51.06.exe** .

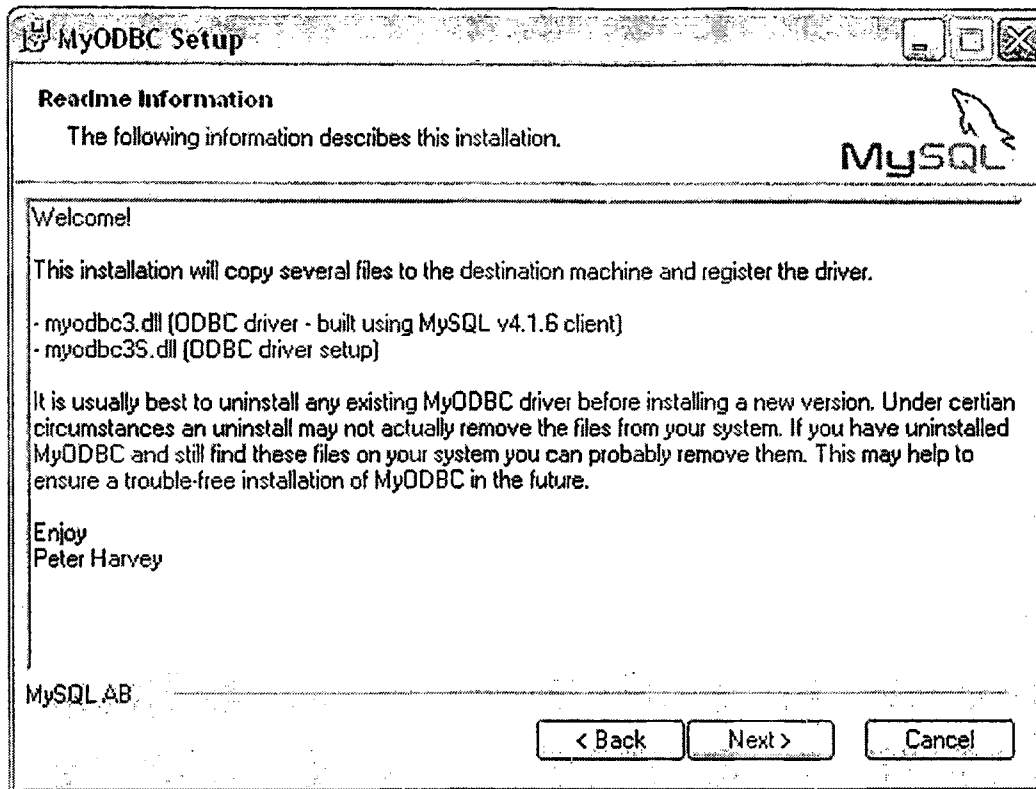
**Εδώ να πούμε ότι η εγκατάσταση του MyODBC-3.51.06.exe είναι επιλογή του καθενός πως θέλει να ενώσει τη βάση δεδομένων του με το site .**



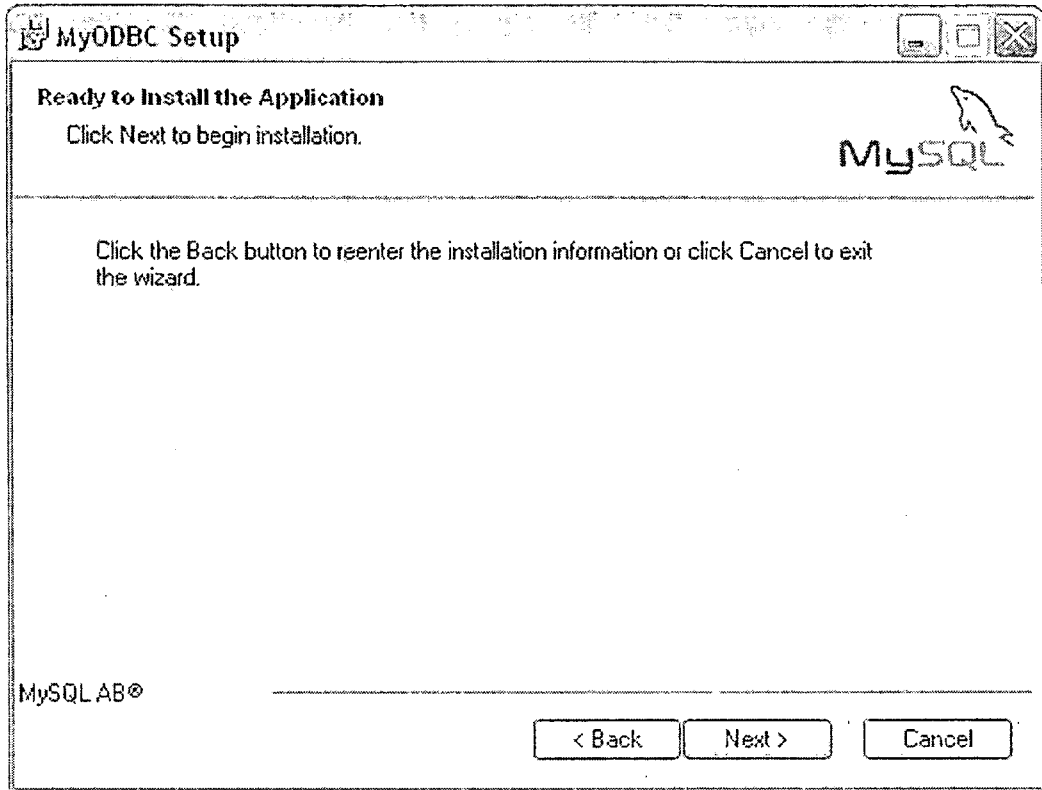
Πατάμε next .



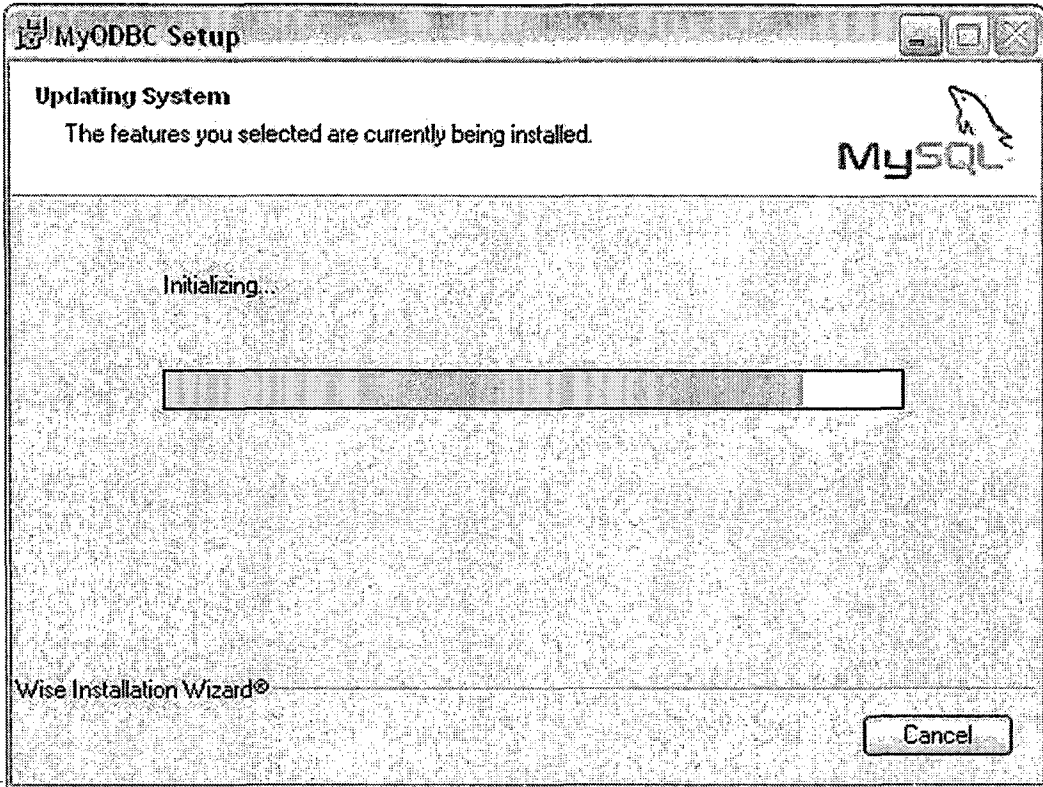
Επιλέγουμε I accept και πατάμε Next .

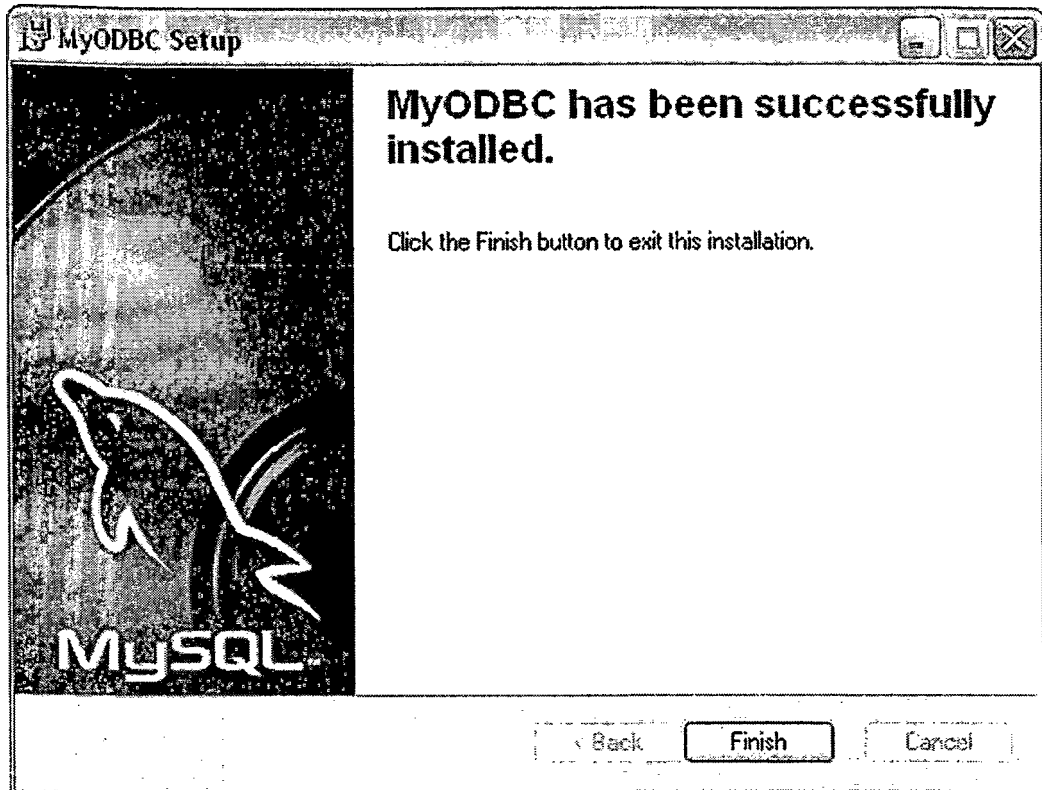


Πατάμε next .



Πατάμε next .





Πατάμε finish για τελειώσει η εγκατάσταση .

### Εγκατάσταση PHP MyADMIN

---

Αποσυμπιέζουμε το **phpMyAdmin-2.6.0-pl3.zip** στα **htdocs** και το μετονομάζουμε σε κάτι σαν και αυτό : **C:\Program Files\Apache Group\Apache2\htdocs\phpmyadmin .**

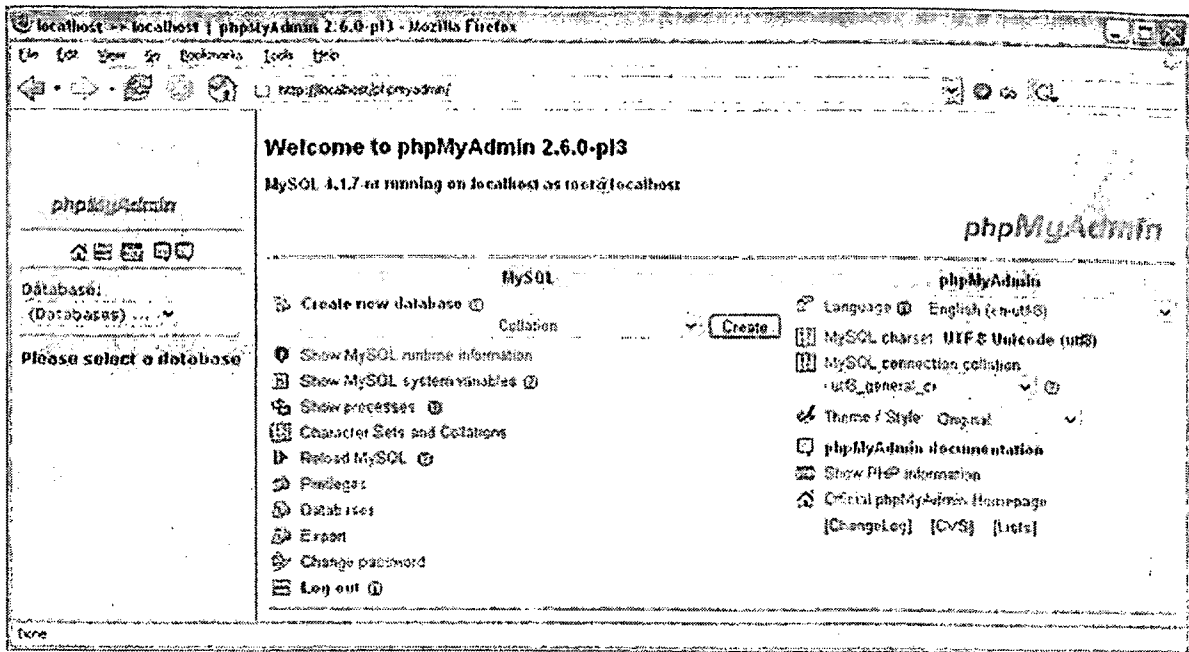
Μετά ανοίγουμε το **config.inc.php** (μέσα στο φάκελο του **phpmyadmin** ) σε ένα text editor και αλλάζουμε τις ακόλουθες μεταβλητές :

```
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpmyadmin/';  
$cfg['Servers'][$i]['extension'] = 'mysqli';  
$cfg['Servers'][$i]['auth_type'] = 'http';
```



Και τέλος ανοίγουμε σε ένα web browser το <http://localhost/phpmyadmin/> χρησιμοποιώντας ως user name το root και password αυτό που δώσαμε πρίν .

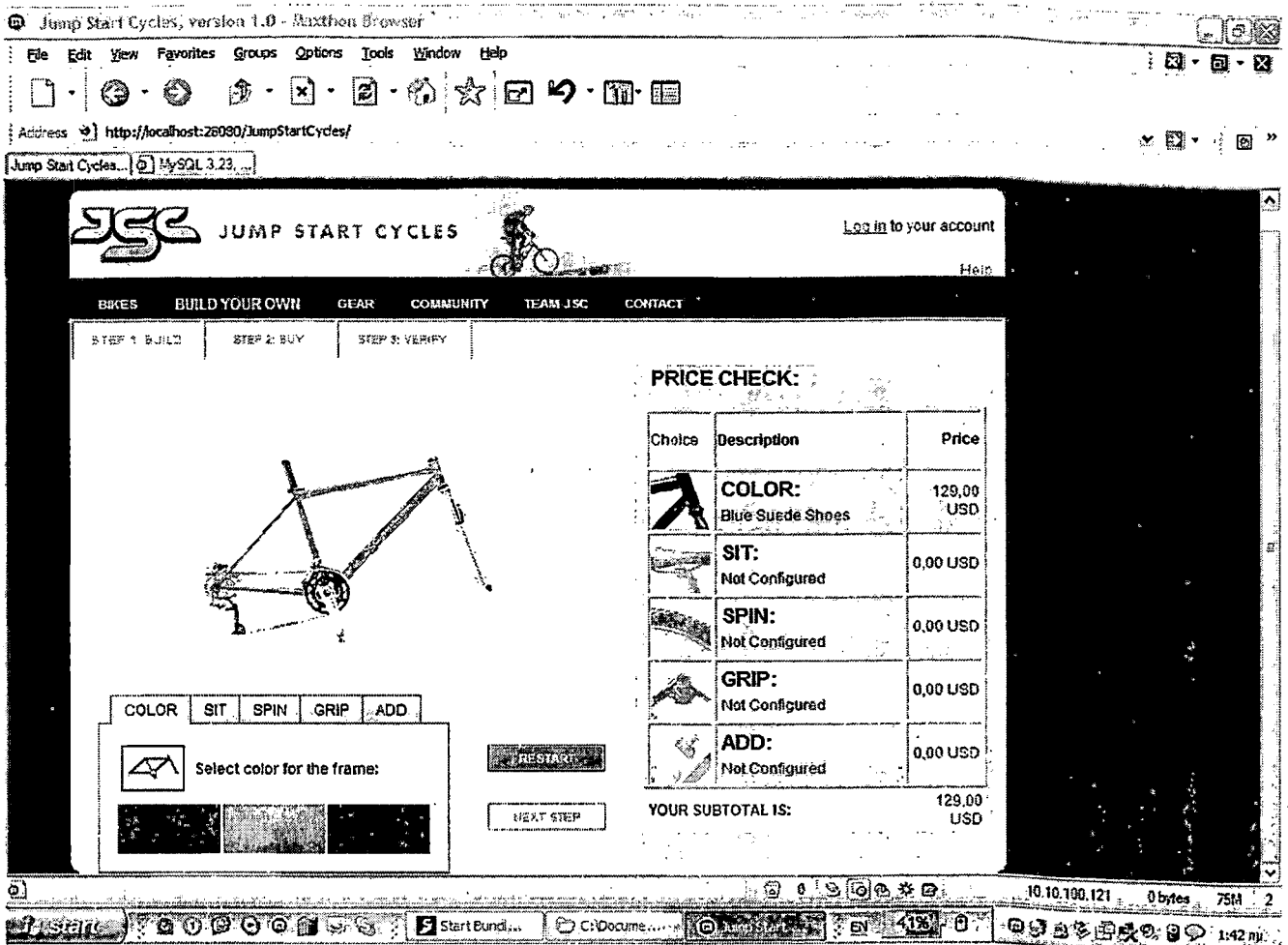
Θα πρέπει να δούμε αυτό :



Όλο το τρίτο κεφάλαιο είναι στην ουσία η βάση για να γίνει οτιδήποτε γιατί αν δεν έχουμε κάνει σωστά όλα τα προηγούμενα τότε δεν θα μπορούσαμε να προχωρήσουμε παρακάτω στην υλοποίηση και εφαρμογή του site που θέλουμε να κατασκευάσουμε .

#### 4. Jump Start Cycles

Εδώ είναι μια φωτογραφία απο το site.



Και πιο κάτω παραθέτουμε τον κώδικα jsp της συγκεκριμένης σελίδας.

```

<jsp:directive.page contentType="text/html;charset=UTF-8" pageEncoding="UTF-
8"/>

<f:view>

  <ui:page binding="#{Build.page1}" id="page1">
    <ui:html binding="#{Build.html1}" id="html1">
      <ui:head binding="#{Build.head1}" id="head1" title="Jump Start Cycles,
      version 1.0">
        <ui:link binding="#{Build.link1}" id="link1"
        url="/resources/styleSheet.css"/>
      </ui:head>
    </ui:html>
  </ui:page>

```

```

<ui:body binding="#{Build.body1}" id="body1" style="background-color:
rgb(51, 51, 51); -rave-layout: grid">
    <ui:form binding="#{Build.form1}" id="form1">
        <div style="left: 48px; top: 24px; position: absolute; width:
750px">
            <jsp:directive.include file="TopMenu.jspf"/>
        </div>
        <ui:tabSet binding="#{Build.optionTabSet}" id="optionTabSet"
lite="true" mini="true" selected="tabColor" style="height: 118px; left: 72px; top:
432px; position: absolute">
            <ui:tab binding="#{Build.tabColor}" id="tabColor"
text="COLOR">
                <h:panelGrid binding="#{Build.gridColorOuter}" columns="1"
id="gridColorOuter">
                    <h:panelGrid binding="#{Build.gridColorTop}"
cellpadding="3" columns="2" id="gridColorTop">
                        <ui:image binding="#{Build.iconColor}" height="35"
id="iconColor" style="border: 1px solid"
                            url="resources/icn_color_down.gif" width="48"/>
                        <ui:label binding="#{Build.labelColor}" id="labelColor"
labelLevel="2" text="Select color for the frame:"/>
                    </h:panelGrid>
                    <h:panelGrid binding="#{Build.gridColorBottom}"
cellpadding="0" columns="3" id="gridColorBottom">
                        <ui:imageHyperlink
action="#{Build.selectColor1_action}" binding="#{Build.selectColor1}" height="40"
id="selectColor1"
                            imageURL="resources/img_opt_color_1.png"
                            tooltip="Blue Suede Shoes" width="82"/>

```

```

        <ui:imageHyperlink
action="#{Build.selectColor2_action}" binding="#{Build.selectColor2}" height="40"
id="selectColor2"

        imageURL="resources/img_opt_color_2.png"
toolTip="Commando Green" width="82"/>

        <ui:imageHyperlink
action="#{Build.selectColor3_action}" binding="#{Build.selectColor3}" height="40"
id="selectColor3"

        imageURL="resources/img_opt_color_3.png"
toolTip="Bubblegum Pink" width="82"/>
    </h:panelGrid>
</h:panelGrid>
</ui:tab>
<ui:tab binding="#{Build.tabSit}" id="tabSit" text="SIT">
    <h:panelGrid binding="#{Build.gridSitOuter}"
id="gridSitOuter">
        <h:panelGrid binding="#{Build.gridSitTop}" cellpadding="3"
columns="2" id="gridSitTop">
            <ui:image binding="#{Build.iconSit}" height="35"
id="iconSit" style="border: 1px solid" url="resources/icn_sit_down.gif" width="48"/>
            <ui:label binding="#{Build.labelSit}" id="labelSit"
labelLevel="2" text="Select seat type:"/>
        </h:panelGrid>
        <h:panelGrid binding="#{Build.gridSitBottom}"
cellpadding="0" columns="3" id="gridSitBottom">
            <ui:imageHyperlink action="#{Build.selectSit1_action}"
binding="#{Build.selectSit1}" height="40" id="selectSit1"

            imageURL="resources/img_opt_sit_1.png"
toolTip="Bananas Forever" width="82"/>
            <ui:imageHyperlink action="#{Build.selectSit2_action}"
binding="#{Build.selectSit2}" height="40" id="selectSit2"

```

```

        imageURL="resources/img_opt_sit_2.png"
toolTip="Cushy Tushy" width="82"/>
        <ui:imageHyperlink action="#{Build.selectSit3_action}"
binding="#{Build.selectSit3}" height="40" id="selectSit3"
        imageURL="resources/img_opt_sit_3.png"
toolTip="Hillclimber" width="82"/>
    </h:panelGrid>
</h:panelGrid>
</ui:tab>
<ui:tab binding="#{Build.tabSpin}" id="tabSpin" text="SPIN">
    <h:panelGrid binding="#{Build.gridSpinOuter}"
id="gridSpinOuter">
        <h:panelGrid binding="#{Build.gridSpinTop}"
cellpadding="3" columns="2" id="gridSpinTop">
            <ui:image binding="#{Build.iconSpin}" height="35"
id="iconSpin" style="border: 1px solid"
                url="resources/icn_spin_down.gif" width="48"/>
            <ui:label binding="#{Build.labelSpin}" id="labelSpin"
labelLevel="2" text="Select your favorite wheels:"/>
        </h:panelGrid>
    <h:panelGrid binding="#{Build.gridSpinBottom}"
columns="3" id="gridSpinBottom">
        <ui:imageHyperlink action="#{Build.selectSpin1_action}"
binding="#{Build.selectSpin1}" height="40" id="selectSpin1"
            imageURL="resources/img_opt_spin_1.png"
toolTip="Heavy Tread" width="82"/>
        <ui:imageHyperlink action="#{Build.selectSpin2_action}"
binding="#{Build.selectSpin2}" height="40" id="selectSpin2"
            imageURL="resources/img_opt_spin_2.png"
toolTip="Skinny Street" width="82"/>

```

```

        <ui:imageHyperlink action="#{Build.selectSpin3_action}"
binding="#{Build.selectSpin3}" height="40" id="selectSpin3"
        imageURL="resources/img_opt_spin_3.png"
toolTip="Super Fat Boys" width="82"/>
    </h:panelGrid>
</h:panelGrid>
</ui:tab>
<ui:tab binding="#{Build.tabGrip}" id="tabGrip" text="GRIP">
    <h:panelGrid binding="#{Build.gridGripOuter}"
id="gridGripOuter">
        <h:panelGrid binding="#{Build.gridGripTop}"
cellpadding="3" columns="2" id="gridGripTop">
            <ui:image binding="#{Build.iconGrip}" height="35"
id="iconGrip" style="border: 1px solid"
                url="resources/icn_grip_down.gif" width="48"/>
            <ui:label binding="#{Build.labelGrip}" id="labelGrip"
labelLevel="2" text="Add your handlebar grips:"/>
        </h:panelGrid>
        <h:panelGrid binding="#{Build.gridGripBottom}"
columns="3" id="gridGripBottom">
            <ui:imageHyperlink action="#{Build.selectGrip1_action}"
binding="#{Build.selectGrip1}" height="40" id="selectGrip1"
                imageURL="resources/img_opt_grip_1.png"
toolTip="Jumpstart Extreme" width="82"/>
            <ui:imageHyperlink action="#{Build.selectGrip2_action}"
binding="#{Build.selectGrip2}" height="40" id="selectGrip2"
                imageURL="resources/img_opt_grip_2.png"
toolTip="Bar-B's" width="82"/>
            <ui:imageHyperlink action="#{Build.selectGrip3_action}"
binding="#{Build.selectGrip3}" height="40" id="selectGrip3"

```

```

        imageURL="resources/img_opt_grip_3.png"
toolTip="Hog Bars" width="82"/>
    </h:panelGrid>
</h:panelGrid>
</ui:tab>
<ui:tab binding="#{Build.tabAdd}" id="tabAdd" text="ADD">
    <h:panelGrid                binding="#{Build.gridAddOuter}"
id="gridAddOuter">
        <h:panelGrid                binding="#{Build.gridAddTop}"
cellpadding="3" columns="2" id="gridAddTop">
            <ui:image    binding="#{Build.iconAdd}"    height="35"
id="iconAdd" style="border: 1px solid" url="resources/icn_add_down.gif" width="48"/>
            <ui:label    binding="#{Build.labelAdd}"    id="labelAdd"
labelLevel="2" text="Finish with an accessory pack:"/>
        </h:panelGrid>
    <h:panelGrid                binding="#{Build.gridAddBottom}"
columns="3" id="gridAddBottom">
        <ui:imageHyperlink action="#{Build.selectAdd1_action}"
binding="#{Build.selectAdd1}" height="40" id="selectAdd1"
            imageURL="resources/img_opt_add_1.png"
toolTip="Diva Package" width="82"/>
        <ui:imageHyperlink action="#{Build.selectAdd2_action}"
binding="#{Build.selectAdd2}" height="40" id="selectAdd2"
            imageURL="resources/img_opt_add_2.png"
toolTip="All-Terrain Package" width="82"/>
        <ui:imageHyperlink action="#{Build.selectAdd3_action}"
binding="#{Build.selectAdd3}" height="40" id="selectAdd3"
            imageURL="resources/img_opt_add_3.png"
toolTip="Search and Rescue Package" width="82"/>
    </h:panelGrid>
</h:panelGrid>

```

```

        </ui:tab>
    </ui:tabSet>
    <ui:image      binding="#{Build.overlaySpin}"      height="255"
id="overlaySpin" style="position: absolute; left: 72px; top: 168px" width="411"/>
    <ui:image      binding="#{Build.overlayColor}"      height="255"
id="overlayColor" style="position: absolute; left: 72px; top: 168px" width="411"/>
    <ui:image      binding="#{Build.overlaySit}"        height="255"
id="overlaySit" style="position: absolute; left: 72px; top: 168px" width="411"/>
    <ui:image      binding="#{Build.overlayGrip}"       height="255"
id="overlayGrip" style="position: absolute; left: 72px; top: 168px" width="411"/>
    <ui:image      binding="#{Build.overlayAdd}"        height="255"
id="overlayAdd" style="left: 72px; top: 168px; position: absolute" width="411"/>
    <ui:panelGroup binding="#{Build.pricePanel}"      block="true"
id="pricePanel" style="height: 406px; position: absolute; background-color: #eee88e;
width: 286px; left: 504px; top: 168px"/>
    <div style="left: 512px; top: 204px; position: absolute">
        <jsp:directive.include file="PriceTable.jspf"/>
    </div>
    <ui:label      binding="#{Build.priceHeader}"      id="priceHeader"
labelLevel="1" style="position: absolute; left: 518px; top: 170px" text="PRICE
CHECK:"/>
    <h:panelGrid  binding="#{Build.totals}"            border="0"
columnClasses="totals-colum-description,totals-column-price" columns="2" id="totals"
style="left: 512px; top: 516px; position: absolute"
styleClass="totals-table">
        <ui:label      binding="#{Build.totalsSubtotalDescription}"
id="totalsSubtotalDescription" labelLevel="2" text="YOUR SUBTOTAL IS:"/>
        <ui:staticText binding="#{Build.totalsSubtotalPrice}"
converter="#{Build.subtotalConverter}" id="totalsSubtotalPrice"/>
    </h:panelGrid>

```



```

                <ui:button                                action="{Build.restart_action}"
binding="{Build.restart}" id="restart"
                style="left: 384px; top: 480px; position: absolute; width: 95px"
text="RESTART"/>
                <ui:button                                action="{Build.nextStep_action}"
binding="{Build.nextStep}" id="nextStep" primary="true"
                style="left: 384px; top: 528px; position: absolute; width: 95px"
text="NEXT STEP"/>
                <ui:panelGroup                            binding="{Build.loginPromptPanel}"
id="loginPromptPanel" style="left: 672px; top: 48px; position: absolute">
                <ui:hyperlink                            action="{Build.loginPromptLink_action}"
binding="{Build.loginPromptLink}" id="loginPromptLink" text="Log in"/>
                <ui:staticText                            binding="{Build.loginPromptText}"
id="loginPromptText" text=" to your account"/>
                </ui:panelGroup>
                <ui:staticText                            binding="{Build.welcomeMessage}"
id="welcomeMessage" style="left: 528px; top: 48px; position: absolute; text-align:
right; width: 264px"/>
                <ui:panelGroup                            binding="{Build.logoutPromptPanel}"
id="logoutPromptPanel" style="left: 672px; top: 72px; position: absolute">
                <ui:hyperlink                            action="{Build.logoutPanelLink_action}"
binding="{Build.logoutPanelLink}" id="logoutPanelLink" text="Log out"/>
                <ui:staticText                            binding="{Build.logoutPanelText}"
id="logoutPanelText" text=" of your account"/>
                </ui:panelGroup>
                <h:panelGrid bgcolor="#daeef6" binding="{Build.stepPanel}"
                columnClasses="step-selected,step-unselected,step-
unselected,step-unselected" columns="4" id="stepPanel" rules="cols" style="border-
width: 0px; border-style: solid; height: 36px; left: 50px; top: 130px; position: absolute;
width: 750px">
                <ui:panelGroup                            binding="{Build.stepPanel1}"    block="true"
id="stepPanel1" style="width: 100px">

```

```

        <h:outputText                binding="#{Build.outputText1}"
id="outputText1"
                style="background-color: white; font-family: sans-serif;
font-size: 10px" value="STEP 1: BUILD"/>
    </ui:panelGroup>
    <ui:panelGroup binding="#{Build.stepPanel2}" block="true"
id="stepPanel2" style="width: 100px">
        <h:outputText                binding="#{Build.outputText2}"
id="outputText2" style="font-family: sans-serif; font-size: 9px" value="STEP 2: BUY"/>
    </ui:panelGroup>
    <ui:panelGroup binding="#{Build.stepPanel3}" block="true"
id="stepPanel3" style="width: 100px">
        <h:outputText                binding="#{Build.outputText3}"
id="outputText3" style="font-family: sans-serif; font-size: 9px" value="STEP 3:
VERIFY"/>
    </ui:panelGroup>
    <ui:panelGroup binding="#{Build.stepPanel4}" block="true"
id="stepPanel4" style="height: 21px; width: 420px"/>
</h:panelGrid>
</ui:form>
</ui:body>
</ui:html>
</ui:page>
</f:view>
</jsp:root>

```

Choice	Description	Price
<Image>	<b>COLOR:</b> Static Text	Static Text
<Image>	<b>SIT:</b> Static Text	Static Text
<Image>	<b>SPIN:</b> Static Text	Static Text
<Image>	<b>GRIP:</b> Static Text	Static Text
<Image>	<b>ADD:</b> Static Text	Static Text

Εδώ έχουμε τον πίνακα τιμών της ιστοσελίδας μας αυτός υλοποιείται σε καθαρή java και προσαρμόζεται μέσα στον jsp κώδικα . Παρακάτω παραθέτουμε αυτόν τον κώδικα .

```
package jumpstartcycles;
```

```
import com.sun.rave.web.ui.appbase.AbstractPageBean;
```

```
import java.math.BigDecimal;
```

```
import javax.faces.FacesException;
```

```
import javax.faces.component.html.HtmlPanelGrid;
```

```
import com.sun.rave.web.ui.component.StaticText;
```

```
import com.sun.rave.web.ui.component.Label;
```

```
import com.sun.rave.web.ui.component.ImageComponent;
```

```
import javax.faces.convert.NumberConverter;
```

```

public class PriceTable extends AbstractPageBean {
    // <editor-fold defaultstate="collapsed" desc="Creator-managed Component
Definition">

    private int __placeholder;

    private HtmlPanelGrid priceTable = new HtmlPanelGrid();

    public HtmlPanelGrid getPriceTable() {
        return priceTable;
    }

    public void setPriceTable(HtmlPanelGrid hpg) {
        this.priceTable = hpg;
    }

    private StaticText priceHeaderIcon = new StaticText();

    public StaticText getPriceHeaderIcon() {
        return priceHeaderIcon;
    }

    public void setPriceHeaderIcon(StaticText st) {
        this.priceHeaderIcon = st;
    }

    private Label priceHeaderDescription = new Label();

```

```
public Label getPriceHeaderDescription() {  
    return priceHeaderDescription;  
}
```

```
public void setPriceHeaderDescription(Label l) {  
    this.priceHeaderDescription = l;  
}
```

```
private Label priceHeaderPrice = new Label();
```

```
public Label getPriceHeaderPrice() {  
    return priceHeaderPrice;  
}
```

```
public void setPriceHeaderPrice(Label l) {  
    this.priceHeaderPrice = l;  
}
```

```
private ImageComponent priceColorIcon = new ImageComponent();
```

```
public ImageComponent getPriceColorIcon() {  
    return priceColorIcon;  
}
```

```
public void setPriceColorIcon(ImageComponent ic) {  
    this.priceColorIcon = ic;  
}
```

```
private HtmlPanelGrid priceColorPanel = new HtmlPanelGrid();
```

```
public HtmlPanelGrid getPriceColorPanel() {  
    return priceColorPanel;  
}
```

```
public void setPriceColorPanel(HtmlPanelGrid hpg) {  
    this.priceColorPanel = hpg;  
}
```

```
private Label priceColorLabel = new Label();
```

```
public Label getPriceColorLabel() {  
    return priceColorLabel;  
}
```

```
public void setPriceColorLabel(Label l) {  
    this.priceColorLabel = l;  
}
```

```
private StaticText priceColorName = new StaticText();
```

```
public StaticText getPriceColorName() {  
    return priceColorName;  
}
```

```
public void setPriceColorName(StaticText st) {
```

```
        this.priceColorName = st;
    }

    private StaticText priceColorPrice = new StaticText();

    public StaticText getPriceColorPrice() {
        return priceColorPrice;
    }

    public void setPriceColorPrice(StaticText st) {
        this.priceColorPrice = st;
    }

    private ImageComponent priceSitIcon = new ImageComponent();

    public ImageComponent getPriceSitIcon() {
        return priceSitIcon;
    }

    public void setPriceSitIcon(ImageComponent ic) {
        this.priceSitIcon = ic;
    }

    private HtmlPanelGrid priceSitPanel = new HtmlPanelGrid();

    public HtmlPanelGrid getPriceSitPanel() {
        return priceSitPanel;
    }
}
```

```

public void setPriceSitPanel(HtmlPanelGrid hpg) {
    this.priceSitPanel = hpg;
}

private StaticText priceSitPrice = new StaticText();

public StaticText getPriceSitPrice() {
    return priceSitPrice;
}

public void setPriceSitPrice(StaticText st) {
    this.priceSitPrice = st;
}

private Label priceSitLabel = new Label();

public Label getPriceSitLabel() {
    return priceSitLabel;
}

public void setPriceSitLabel(Label l) {
    this.priceSitLabel = l;
}

private StaticText priceSitName = new StaticText();

public StaticText getPriceSitName() {

```



```

    return priceSitName;
}

public void setPriceSitName(StaticText st) {
    this.priceSitName = st;
}

private ImageComponent priceSpinIcon = new ImageComponent();

public ImageComponent getPriceSpinIcon() {
    return priceSpinIcon;
}

public void setPriceSpinIcon(ImageComponent ic) {
    this.priceSpinIcon = ic;
}

private HtmlPanelGrid priceSpinPanel = new HtmlPanelGrid();

public HtmlPanelGrid getPriceSpinPanel() {
    return priceSpinPanel;
}

public void setPriceSpinPanel(HtmlPanelGrid hpg) {
    this.priceSpinPanel = hpg;
}

private StaticText priceSpinPrice = new StaticText();

```

```
public StaticText getPriceSpinPrice() {
    return priceSpinPrice;
}

public void setPriceSpinPrice(StaticText st) {
    this.priceSpinPrice = st;
}

private Label priceSpinLabel = new Label();

public Label getPriceSpinLabel() {
    return priceSpinLabel;
}

public void setPriceSpinLabel(Label l) {
    this.priceSpinLabel = l;
}

private StaticText priceSpinName = new StaticText();

public StaticText getPriceSpinName() {
    return priceSpinName;
}

public void setPriceSpinName(StaticText st) {
    this.priceSpinName = st;
}
```

```
private ImageComponent priceGripIcon = new ImageComponent();
```

```
public ImageComponent getPriceGripIcon() {  
    return priceGripIcon;  
}
```

```
public void setPriceGripIcon(ImageComponent ic) {  
    this.priceGripIcon = ic;  
}
```

```
private HtmlPanelGrid priceGripPanel = new HtmlPanelGrid();
```

```
public HtmlPanelGrid getPriceGripPanel() {  
    return priceGripPanel;  
}
```

```
public void setPriceGripPanel(HtmlPanelGrid hpg) {  
    this.priceGripPanel = hpg;  
}
```

```
private StaticText priceGripPrice = new StaticText();
```

```
public StaticText getPriceGripPrice() {  
    return priceGripPrice;  
}
```

```
public void setPriceGripPrice(StaticText st) {
```

```

        this.priceGripPrice = st;
    }

    private Label priceGripLabel = new Label();

    public Label getPriceGripLabel() {
        return priceGripLabel;
    }

    public void setPriceGripLabel(Label l) {
        this.priceGripLabel = l;
    }

    private StaticText priceGripName = new StaticText();

    public StaticText getPriceGripName() {
        return priceGripName;
    }

    public void setPriceGripName(StaticText st) {
        this.priceGripName = st;
    }

    private ImageComponent priceAddIcon = new ImageComponent();

    public ImageComponent getPriceAddIcon() {
        return priceAddIcon;
    }

```

```
public void setPriceAddIcon(ImageComponent ic) {
    this.priceAddIcon = ic;
}

private HtmlPanelGrid priceAddPanel = new HtmlPanelGrid();

public HtmlPanelGrid getPriceAddPanel() {
    return priceAddPanel;
}

public void setPriceAddPanel(HtmlPanelGrid hpg) {
    this.priceAddPanel = hpg;
}

private StaticText priceAddPrice = new StaticText();

public StaticText getPriceAddPrice() {
    return priceAddPrice;
}

public void setPriceAddPrice(StaticText st) {
    this.priceAddPrice = st;
}

private Label priceAddLabel = new Label();

public Label getPriceAddLabel() {
```

```

        return priceAddLabel;
    }

    public void setPriceAddLabel(Label l) {
        this.priceAddLabel = l;
    }

    private StaticText priceAddName = new StaticText();

    public StaticText getPriceAddName() {
        return priceAddName;
    }

    public void setPriceAddName(StaticText st) {
        this.priceAddName = st;
    }

    private NumberConverter priceColorConverter = new NumberConverter();

    public NumberConverter getPriceColorConverter() {
        return priceColorConverter;
    }

    public void setPriceColorConverter(NumberConverter nc) {
        this.priceColorConverter = nc;
    }

    private NumberConverter priceSitConverter = new NumberConverter();

```

```
public NumberConverter getPriceSitConverter() {  
    return priceSitConverter;  
}
```

```
public void setPriceSitConverter(NumberConverter nc) {  
    this.priceSitConverter = nc;  
}
```

```
private NumberConverter priceSpinConverter = new NumberConverter();
```

```
public NumberConverter getPriceSpinConverter() {  
    return priceSpinConverter;  
}
```

```
public void setPriceSpinConverter(NumberConverter nc) {  
    this.priceSpinConverter = nc;  
}
```

```
private NumberConverter priceGripConverter = new NumberConverter();
```

```
public NumberConverter getPriceGripConverter() {  
    return priceGripConverter;  
}
```

```
public void setPriceGripConverter(NumberConverter nc) {  
    this.priceGripConverter = nc;  
}
```

```

private NumberConverter priceAddConverter = new NumberConverter();

public NumberConverter getPriceAddConverter() {
    return priceAddConverter;
}

public void setPriceAddConverter(NumberConverter nc) {
    this.priceAddConverter = nc;
}
// </editor-fold>
public PriceTable() {
    // <editor-fold defaultstate="collapsed" desc="Creator-managed Component
Initialization">
    try {
        priceColorConverter.setType("currency");
        priceColorConverter.setCurrencyCode("USD");
        priceColorConverter.setCurrencySymbol("$");
        priceSitConverter.setType("currency");
        priceSitConverter.setCurrencyCode("USD");
        priceSitConverter.setCurrencySymbol("$");
        priceSpinConverter.setType("currency");
        priceSpinConverter.setCurrencyCode("USD");
        priceSpinConverter.setCurrencySymbol("$");
        priceGripConverter.setType("currency");
        priceGripConverter.setCurrencyCode("USD");
        priceGripConverter.setCurrencySymbol("$");
        priceAddConverter.setType("currency");

```



```

        priceAddConverter.setCurrencyCode("USD");
        priceAddConverter.setCurrencySymbol("$");
    } catch (Exception e) {
        log("PriceTable Initialization Failure", e);
        throw e instanceof FacesException ? (FacesException) e: new
FacesException(e);
    }
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected ApplicationBean1 getApplicationBean1() {
    return (ApplicationBean1)getBean("ApplicationBean1");
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */
protected RequestBean1 getRequestBean1() {
    return (RequestBean1)getBean("RequestBean1");
}

/**
 * <p>Return a reference to the scoped data bean.</p>
 */

```

```

protected SessionBean1 getSessionBean1() {
    return (SessionBean1)getBean("SessionBean1");
}

public void init() {

    super.init();

    try {
        _init();
    } catch (Exception e) {
        log("PriceTable Initialization Failure", e);
        throw e instanceof FacesException ? (FacesException) e: new
FacesException(e);
    }

}

/**
 * <p>Callback method that is called after the component tree has been
 * restored, but before any event processing takes place. This method
 * will <strong>only</strong> be called on a postback request that
 * is processing a form submit. Customize this method to allocate
 * resources that will be required in your event handlers.</p>
 */
public void preprocess() {

```

```
}
```

```
/**
```

```
* <p>Callback method that is called just before rendering takes place.  
* This method will <strong>only</strong> be called for the page that  
* will actually be rendered (and not, for example, on a page that  
* handled a postback and then navigated to a different page). Customize  
* this method to allocate resources that will be required for rendering  
* this page.</p>
```

```
*/
```

```
public void prerender() {
```

```
}
```

```
/**
```

```
* <p>Callback method that is called after rendering is completed for  
* this request, if <code>init()</code> was called (regardless of whether  
* or not this was the page that was actually rendered). Customize this  
* method to release resources acquired in the <code>init()</code>,  
* <code>preprocess()</code>, or <code>prerender()</code> methods (or  
* acquired during execution of an event handler).</p>
```

```
*/
```

```
public void destroy() {
```

```
}
```

```
/**
```

```
* <p>Automatically managed component initialization.  
<strong>WARNING:</strong>
```

```

* This method is automatically generated, so any user-specified code inserted
* here is subject to being replaced.</p>
*/
private void _init() throws Exception {
}

/**
* <p>Return <code>>true</code> if the specified bike has been
* configured completely enough to order.</p>
*
* @param bike The current bike configuration
*/
public boolean configured(Bike bike) {

    return (bike.getColor().getDesc() != null) &&
           (bike.getSit().getDesc() != null) &&
           (bike.getSpin().getDesc() != null) &&
           (bike.getGrip().getDesc() != null);

}

/**
* <p>Set up the price table based upon the current bike configuration.</p>
*
* @param bike The current bike configuration
*/

```

```

public void setup(Bike bike) {

    // Set properties for the price table
    priceColorIcon.setUrl(bike.getColor().getSquareURL());
    priceColorName.setText(bike.getColor().getName());
    priceColorPrice.setText(bike.getColor().getPrice());
    priceSitIcon.setUrl(bike.getSit().getSquareURL());
    priceSitName.setText(bike.getSit().getName());
    priceSitPrice.setText(bike.getSit().getPrice());
    priceSpinIcon.setUrl(bike.getSpin().getSquareURL());
    priceSpinName.setText(bike.getSpin().getName());
    priceSpinPrice.setText(bike.getSpin().getPrice());
    priceGripIcon.setUrl(bike.getGrip().getSquareURL());
    priceGripName.setText(bike.getGrip().getName());
    priceGripPrice.setText(bike.getGrip().getPrice());
    priceAddIcon.setUrl(bike.getAdd().getSquareURL());
    priceAddName.setText(bike.getAdd().getName());
    priceAddPrice.setText(bike.getAdd().getPrice());

}

/**
 * <p>Calculate and return the current subtotal.</p>
 */
public BigDecimal subtotal(Bike bike) {

    BigDecimal subtotal = new BigDecimal((double) 0.0);

```

```
        subtotal = subtotal.add((BigDecimal) bike.getColor().getPrice());
        subtotal = subtotal.add((BigDecimal) bike.getSit().getPrice());
        subtotal = subtotal.add((BigDecimal) bike.getSpin().getPrice());
        subtotal = subtotal.add((BigDecimal) bike.getGrip().getPrice());
        subtotal = subtotal.add((BigDecimal) bike.getAdd().getPrice());
        return subtotal;
    }
}
```



JUMP START CYCLES



Εδώ είναι το πάνω μενού και αυτό λειτουργεί με αυτή τη λογική υλοποιείται σε java και προσαρμόζεται στον jsp κώδικα . Παρακάτω παραθέτουμε μέρος του κώδικα .

```
public String topMenuBikes_action() {  
    topMenu.setSelected("topMenuBuild");  
    return "case1";  
}
```

```
public String topMenuBuild_action() {  
    topMenu.setSelected("topMenuBuild");  
    return null;  
}
```

```
public String topMenuGear_action() {  
    topMenu.setSelected("topMenuBuild");  
    return "case1";  
}
```

```
public String topMenuCommunity_action() {  
    topMenu.setSelected("topMenuBuild");  
    return "case1";  
}
```

```
public String topMenuTeam_action() {  
    topMenu.setSelected("topMenuBuild");  
    return null;  
}
```

```

public String topMenuContact_action() {
    topMenu.setSelected("topMenuBuild");
    return null;
}

public String topMenuHelp_action() {
    topMenu.setSelected("topMenuHelp");
    return null;
}
}

```

## **5. Βιβλιογραφία**

Αντικειμενοστρεφής προγραμματισμός JAVA Κλεάνθης Θραμπουλίδης Εκδόσεις Τζιολα

Tomcat The Definitive Guide Jason Brittain & Ian F. Darwin Εκδόσεις O'Reilly

Java How to Program Deitel & Deitel

Εισαγωγή στη Java Γιώργος Λιακέας Εκδόσεις Κλειδάριθμος

<http://de.wikipedia.org/wiki/MySQL>

[www.mysql.org](http://www.mysql.org)

<http://java.sun.com/>

[www.java.com](http://www.java.com)

<http://forum.java.sun.com>



<http://tomcat.apache.org/>

<http://java.sun.com/products/jsp/tomcat/>

<http://jakarta.apache.org/site/downloads/>

<http://www.phpmyadmin.net/>