

ΔΙΚΤΥΑΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΓΚΙΩΚΑΣ ΠΑΝΑΓΙΩΤΗΣ | 711171060

ΤΕΛΙΚΗ ΑΣΚΗΣΗ ΕΞΑΜΗΝΟΥ

ΠΕΡΙΕΧΟΜΕΝΑ

Contents

ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	2
Εντολές δημιουργίας βάσης:	2
ΥΛΟΠΟΙΗΣΗ ΚΩΔΙΚΑ	3
Κλάση DBManager:	3
Κλάση Book και BookDao:	3
index.jsp:	4
Servlet AddBookServlet και InsertBook.jsp:	5
DeleteBookServlet και DeleteBook.jsp:	7
UpdateBookServlet, UpdateBook.jsp και UpdateBook2.jsp	8
OrderBookServlet και OrderBook.jsp:	10
ViewBookServlet και ViewBook.jsp:	12

ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Στην άσκηση χρησιμοποιήθηκε μια mysql βάση δεδομένων. Αποτελείτε από τον πίνακα **"Book"** με τα στοιχεία **"ID"**, **"Title"**, **"Author"**, **"Pages"**, **"Copies"**. Επίσης δημιουργήθηκε ο χρήστης **"gkiokas"** με κωδικό **"12345"** για την διαχείριση της.

Εντολές δημιουργίας βάσης:

```
mysql> create database BookManagement;
```

```
mysql> use BookManagement;
```

```
mysql> create table Book;
```

```
mysql> create table Book(ID int NOT NULL, Title  
varchar(50) NOT NULL, Author varchar(30), Pages int,  
Copies int NOT NULL, primary key ID);
```

```
mysql> create table Book(ID int NOT NULL, Title  
varchar(50) NOT NULL, Author varchar(30), Pages int,  
Copies int NOT NULL, primary key (ID));
```

```
mysql> CREATE USER 'gkiokas'@'localhost' IDENTIFIED  
BY '12345';
```

```
mysql> grant all privileges on * . * to  
'gkiokas'@'localhost';
```

ΥΛΟΠΟΙΗΣΗ ΚΩΔΙΚΑ

Για την υλοποίηση του κώδικα έχουν επιλεγθεί servlets και χρησιμοποιούνται σελίδες jsp για την εισαγωγή και εμφάνιση των αποτελεσμάτων λόγω της δυνατότητας που παρέχουν για δυναμικό περιεχόμενο. Ο server που χρησιμοποιείται για το deploy της εφαρμογής είναι ο Tomcat.

Κλάση DBManager:

Στην κλάση αυτή πραγματοποιείται η σύνδεση στη βάση μέσω του χρήστη "gkiokas".

Κλάση Book και BookDao:

Για την επίτευξη της αντικειμενοστρέφειας στον κώδικα έχει δημιουργηθεί η κλάση "**Book**" που περιγράφει το αντικείμενο και του δίνει τα χαρακτηριστικά:

"id": Μοναδικό χαρακτηριστικό του κάθε βιβλίου.

"title": Τίτλος βιβλίου.

"author": Συγγραφέας βιβλίου.

"pages": Σελίδες βιβλίου.

"copies": Αντίτυπα βιβλίου.

Για τον διαχωρισμό της λογικής της εφαρμογής και την απλοποίηση των servlet χρησιμοποιείται η κλάση "**BookDao**". Στην κλάση αυτή περιέχονται οι μέθοδοι για όλες τις λειτουργίες που προσφέρει η εφαρμογή μας.

index.jsp:

Η πρώτη σελίδα που βλέπει ο χρήστης η οποία και την οδηγεί στις υπόλοιπες λειτουργίες.

Welcome to Book Manager

You have 6 options

[View list of books](#)

List of all books in the database.

[View book](#)

More details on specific book in the database.

[Insert book](#)

Insert a book in the database.

[Delete book](#)

Delete a book from the database.

[Update book](#)

Update a book in the database.

[Order book](#)

Order a book.

Για τις περισσότερες λειτουργίες γίνεται κλήση των αντίστοιχων σελίδων jsp εκτός από την λειτουργία “**ViewList**” για την οποία καλείται το servlet απευθείας.

Servlet AddBookServlet και InsertBook.jsp:

Χρησιμοποιείται για την εισαγωγή νέων βιβλίων στη βάση. Έχει υλοποιηθεί η συνάρτηση **“doPost”** καθώς παρέχει περισσότερη ασφάλεια, η οποία είναι απαραίτητη, καθώς τις παραμέτρους που παίρνουμε τις βάζουμε στη βάση. Για τον ίδιο λόγο χρησιμοποιούνται **PreparedStatements** για την εκτέλεση των **queries** στη βάση.

Insert book

ID:
Title:
Author:
Pages:
Copies:

[Home](#)

Μέσα από την **“InsertBook.jsp”** παίρνουμε τις παραμέτρους που χρειαζόμαστε για την εισαγωγή στην βάση. Μέσω της html τα πεδία **“ID”**, **“Pages”**, **“Copies”** παίρνουν μόνο αριθμητικές τιμές και γλυτώνουμε ελέγχους. Όταν έρθει η φόρμα στο servlet ελέγχονται οι τιμές των παραμέτρων για τυχών κενά πεδία ή για μία σύγκρουση δεδομένων λόγω ίδιου **“ID”**. Για όλες τις περιπτώσεις επιστρέφονται στην ίδια σελίδα ενημερωτικά μηνύματα ανάλογα με την έκβαση της εκτέλεσης της λειτουργίας. Για την αναπαράσταση των πεδίων έχει χρησιμοποιηθεί ένας πίνακας μέσα σε μια φόρμα.

Insert book

ID:
Title:
Author:
Pages:
Copies:

All boxes must be filled.

[Home](#)

Insert book

ID:
Title:
Author:
Pages:
Copies:

Insertion failed. ID already exists.

[Home](#)

Insert book

ID:
Title:
Author:
Pages:
Copies:

Insertion succeeded.

[Home](#)

```
mysql> select * from book;
```

ID	Title	Author	Pages	Copies
1	Java Servlets	Gkiokas	250	40
2	Student Life	Gkiokas	60	25
3	Hey	Gkiokas	1	1

```
3 rows in set (0.00 sec)
```

Τα μηνύματα αυτά προστίθενται μέσω μεταβλητών μέσω της jsp και εμφανίζονται μόνο αν υπάρχει η αντίστοιχη μεταβλητή.

```
21 | | |
22 | | | <p>${error}</p>
23 | | | <p>${success}</p>
24 | | | <p>${fail}</p>
```

DeleteBookServlet και DeleteBook.jsp:

Χρησιμοποιείται για την διαγραφή βιβλίων από την βάση. Έχει υλοποιηθεί η συνάρτηση **“doGet”** καθώς δε μας νοιάζει τόσο η ασφάλεια. Το servlet παίρνει το επιθυμητό **“ID”** και μετά από ελέγχους, για το αν έχουν συμπληρωθεί τα κενά και για το αν υπάρχει το βιβλίο, γίνεται η εκτέλεση του αιτήματος.

Delete book

Fill in the ID of the book you want to be deleted

ID:

[Home](#)

Πάλι επιστρέφονται μηνύματα με τον αντίστοιχο τρόπο.

Delete book

Fill in the ID of the book you want to be deleted

ID:

All boxes must be filed.

[Home](#)

Delete book

Fill in the ID of the book you want to be deleted

ID:

Book deleted successfully

[Home](#)

Delete book

Fill in the ID of the book you want to be deleted

ID:

Book could not be deleted. Check the ID again.

[Home](#)

```
mysql> select * from book;
+----+-----+-----+-----+-----+
| ID | Title      | Author | Pages | Copies |
+----+-----+-----+-----+-----+
| 1  | Java Servlets | Gkiokas | 250   | 40     |
| 2  | Student Life  | Gkiokas | 60    | 25     |
| 3  | Hey          | Gkiokas | 1     | 1      |
+----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from book;
+----+-----+-----+-----+-----+
| ID | Title      | Author | Pages | Copies |
+----+-----+-----+-----+-----+
| 1  | Java Servlets | Gkiokas | 250   | 40     |
| 2  | Student Life  | Gkiokas | 60    | 25     |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```


UpdateBookServlet, UpdateBook.jsp και UpdateBook2.jsp

Χρησιμοποιείται για την ενημέρωση των στοιχείων ενός βιβλίου σε μια βάση. Μέσα από την **“UpdateBook.jsp”** ο χρήστης διαλέγει το βιβλίο που θέλει να τροποποιήσει.

Update book

Provide the ID of the book you want to update.

ID:

[Home](#)

Η πληροφορία αυτή στέλνεται στο servlet το οποίο παράγει με την σειρά του την σελίδα **“UpdateBook2.jsp”** στην οποία συμπληρώνει τα υπάρχοντα στοιχεία του βιβλίου σε μια φόρμα, πάνω στην οποία ο χρήστης κάνει τις αλλαγές που θέλει, πλην του **“ID”**, και τις υποβάλλει. Έχει και την δυνατότητα επίσης να επιστρέψει στην προηγούμενη σελίδα.

Update book

Change the attributes you want to update.

ID:
Title:
Author:
Pages:
Copies:

[Return](#)

[Home](#)

Πραγματοποιούνται οι ίδιοι έλεγχοι όπως και στα παραπάνω, αν κατά την διαδικασία της τελικής υποβολής διαπιστωθεί κάποιο κενό πεδίο το πρόγραμμα στέλνει τον χρήστη πίσω στην προηγούμενη σελίδα, ώστε να ξαναφορτωθούν τα στοιχεία του βιβλίου στην φόρμα υποβολής.

Update book

Provide the ID of the book you want to update.

ID:

All boxes must be filled.

[Home](#)

Update book

Provide the ID of the book you want to update.

ID:

Updated successfully.

[Home](#)

Update book

Provide the ID of the book you want to update.

ID:

ID does not exists. Please give a valid ID.

[Home](#)

```
mysql> select * from book;
+----+-----+-----+-----+-----+
| ID | Title       | Author | Pages | Copies |
+----+-----+-----+-----+-----+
| 1  | Java Servlets | Gkiokas | 250   | 40      |
| 2  | Student Life  | Gkiokas | 60    | 25      |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from book;
+----+-----+-----+-----+-----+
| ID | Title       | Author | Pages | Copies |
+----+-----+-----+-----+-----+
| 1  | Java       | Gkiokas | 250   | 80      |
| 2  | Student Life | Gkiokas | 60    | 25      |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

OrderBookServlet και OrderBook.jsp:

Χρησιμοποιείται για την παραγγελία ενός βιβλίου από τον χρήστη. Ο χρήστης διαλέγει μέσω μιας φόρμας στην “**OrderBook.jsp**” το βιβλίο και την ποσότητα που θέλει να παραγγείλει.

Order book

ID:

Amount:

[Home](#)

Με την πληροφορία αυτή, το servlet ελέγχει την διαθεσιμότητα του βιβλίου και προχωράει την λειτουργία.

Order book

ID:

Amount:

All boxes must be filled.

[Home](#)

Order book

ID:

Amount:

Sorry, this ID does not exist.

[Home](#)

Order book

ID:

Amount:

Sorry, this book is out of stock.

[Home](#)

Order book

ID:

Amount:

Sorry, we only have 40 books in stock.

[Home](#)

Order book

ID:

Amount:

Order placed successfully

[Home](#)

```
mysql> select * from book;
+----+-----+-----+-----+-----+
| ID | Title       | Author | Pages | Copies |
+----+-----+-----+-----+-----+
| 1  | Java Servlets | Gkiokas | 250   | 50     |
| 2  | Student Life  | Gkiokas | 60    | 25     |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from book;
+----+-----+-----+-----+-----+
| ID | Title       | Author | Pages | Copies |
+----+-----+-----+-----+-----+
| 1  | Java Servlets | Gkiokas | 250   | 40     |
| 2  | Student Life  | Gkiokas | 60    | 25     |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

ViewListServlet και ViewList.jsp:

Χρησιμοποιείται για την εμφάνιση της λίστας με τα διαθέσιμα βιβλία από τη βάση δεδομένων. Καλείται από την **“index.jsp”** το servlet το οποίο παράγει μια λίστα από βιβλία την οποία δίνει στη jsp σελίδα η οποία κάνοντας χρήση των **“JSTL tags”** παράγει τον πίνακα στον οποίο τοποθετούνται τα δεδομένα των βιβλίων. Τα tags αυτά είναι μια συλλογή από jsp tag τα οποία δίνουν στη σελίδα αρκετές νέες δυνατότητες. Αυτή που χρησιμοποιούμε είναι η επανάληψη κώδικα. Έτσι μπορούμε να εισάγουμε επαναληπτικά τα στοιχεία της λίστας στον πίνακα.

```
<%-- JSTL function for for loop in order to create the table --%>
<c:forEach items="${list}" var="item">
  <tr>
    <td>${item.id}</td>
    <td>${item.title}</td>
    <td>${item.author}</td>
  </tr>
</c:forEach>
```

Επίσης δίνεται η επιλογή στον χρήστη να πάει απευθείας στην σελίδα **“ViewBook.jsp”** για να πάρει περισσότερες πληροφορίες για το βιβλίο που τον ενδιαφέρει.

List of all books.

View specific book for more details by using its ID.

Id	Title	Author
1	Java Servlets	Gkiokas
2	Student Life	Gkiokas

[View Book](#)

[Home](#)

ViewBookServlet και ViewBook.jsp:

Χρησιμοποιείται για την εμφάνιση περισσότερων πληροφοριών για ένα βιβλίο. Ο χρήστης συμπληρώνει το “ID” του βιβλίου που θέλει στην φόρμα που βρίσκετε στην “**ViewBook.jsp**” και το servlet αφού ελέγξει τις παραμέτρους επιστρέφει στην σελίδα το βιβλίο. Η σελίδα έπειτα δημιουργεί τον πίνακα για να εμφανισθούν τα δεδομένα. Η δημιουργία αυτή θα γίνει μόνο αν έχει επιστραφεί κάποιο βιβλίο από το servlet. Ο έλεγχος αυτός γίνεται μέσα από “**JSTL tags**” και ειδικότερα αυτό του ελέγχου.

```
21 <!-- JSTL if statement to check if table must be shown-->
22 <c:if test = "${flag == 1}">
23     <table style="border:1px solid black; padding:5px; border-spacing:2px;">
24         <tr>
25             <th>Id</th>
26             <th>Title</th>
27             <th>Author</th>
28             <th>Pages</th>
29             <th>Copies</th>
30         </tr>
31         <tr>
32             <td>${book.id}</td>
33             <td>${book.title}</td>
34             <td>${book.author}</td>
35             <td>${book.pages}</td>
36             <td>${book.copies}</td>
37         </tr>
38     </table>
39 </c:if>
```

Επίσης δίνεται η επιλογή στον χρήστη να πάει απευθείας στην σελίδα “**OrderBook.jsp**” για να παραγγείλει κάποιο βιβλίο που τον ενδιαφέρει.

View book

ID:

[Order book](#)

[Home](#)

View book

ID:

Please give an ID.

[Order book](#)

[Home](#)

View book

ID:

ID does not exists. Please give a valid ID.

[Order book](#)

[Home](#)

View book

ID:

Id	Title	Author	Pages	Copies
1	Java Servlets	Gkiokas	250	50

[Order book](#)

[Home](#)