EECS 4413. LAB 06: Data persist: JDBC. DAO design pattern

A. IMPORTANT REMINDERS

- Lab6 is due on **Wednesday (Aug 2) at 11pm**. No late submission will be accepted.
- For this lab, you are welcome to attend the lab sessions on ~~this and~~ next Wednesday. TAs or instructor will be available to help you. The location is LAS1002. Attendance is optional. You can also ask questions on Monday (July 31) after class.
- Feel free to signal a TA for help if you stuck on any of the steps below. Yet, note that TAs would need to help other students too.
- You can submit your lab work any time before the specified deadline.

B. IMPORTANT PRE-LAB WORKS YOU NEED TO DO BEFORE GOING TO THE LAB

- Download this lab description and the associated files and read it completely.

C. GOALS/OUTCOMES FOR LAB

- On JDBC, both in MySQL, and SQLite.
- DAO design pattern
- JSP with EL, JSTL.

D. TASKS

Part1A: JDBC connection -- MYSQL

Part1B: DAO and MVC model

Part 2: JDBC connection -- SQLite

E. SUBMISSIONS

- eClass submission. More information can be found at the end of this document.

Part I JDBC, DAO model

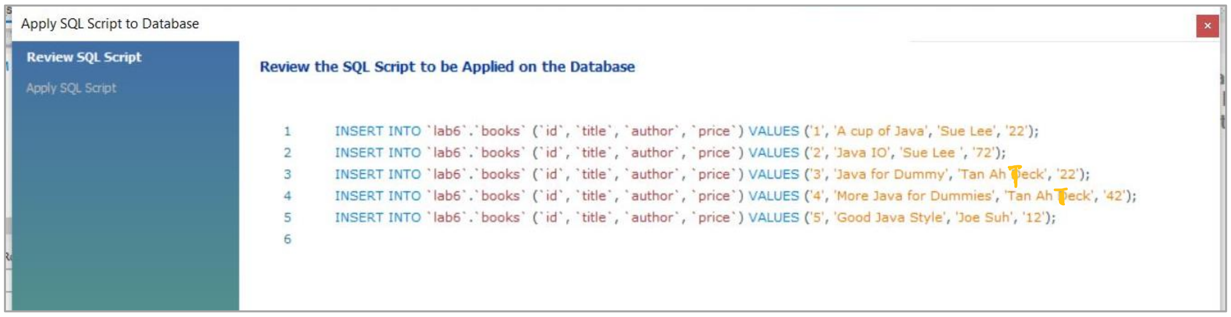
In this exercise, we extend the bookshop program you did in lab3~5, but with book data in a database. In this exercise we use MySQL database. In Part 2 we use another database SQLite.

In the previous versions, book is entered manually and maintained in class Table. Now, we put the data in database.

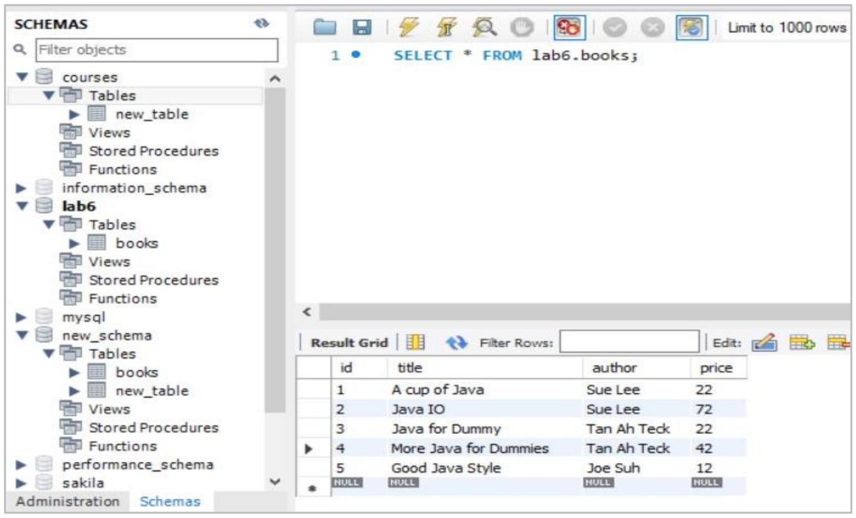
Part 1A

- Download MySQL from https://www.mysql.com/ and install on your computer. Choose MySQL Community Server. You can install as Service or Application.   During installation or configuration, when asked, use **4413** as the password for **root** user. This allows the TA to run your program without modifying the password.
- You can manipulate MySQL with command line, using the shell it provides or your terminal, but it is easier to use a GUI tool. Download a GUI for MySQL. One popular one is MySQL WorkBench. If the installation already has the WorkBench GUI tool, you can use it. Otherwise, you can download MySQL WorkBench separately or download other GUI tools that can connect to MySQL (e.g., Heidi SQL).
- To operate on the database from Java, you should also have a jar file (the database driver). If your installation already has a folder Connector J, that you can find the jar file there. Otherwise download the from Connector J for your version of MySQL. Two versions of the connector jar files are also provided for you.
- Start the MySQL server, and use command line or a GUI tool such as MySQL WorkBench to create a Schema/Database **lab6** and then create a table **books** in the schema/database, with the structure and data as follows:

```
CREATE TABLE `books` (
 `id` int NOT NULL,
 `title` varchar(45) DEFAULT NULL,
 `author` varchar(45) DEFAULT NULL,
 `price` double DEFAULT NULL,
 PRIMARY KEY (`id`)
```


Apply SQL Script to Database
Review SQL Script
Apply SQL Script
Review the SQL Script to be Applied on the Database

```
INSERT INTO `lab6`.`books` (`id`, `title`, `author`, `price`) VALUES ('1', 'A cup of Java', 'Sue Lee', '22');
INSERT INTO `lab6`.`books` (`id`, `title`, `author`, `price`) VALUES ('2', 'Java IO', 'Sue Lee ', '72');
INSERT INTO `lab6`.`books` (`id`, `title`, `author`, `price`) VALUES ('3', 'Java for Dummy', 'Tan Ah Teck', '22');
INSERT INTO `lab6`.`books` (`id`, `title`, `author`, `price`) VALUES ('4', 'More Java for Dummies', 'Tan Ah Teck', '42');
INSERT INTO `lab6`.`books` (`id`, `title`, `author`, `price`) VALUES ('5', 'Good Java Style', 'Joe Suh', '12');
```



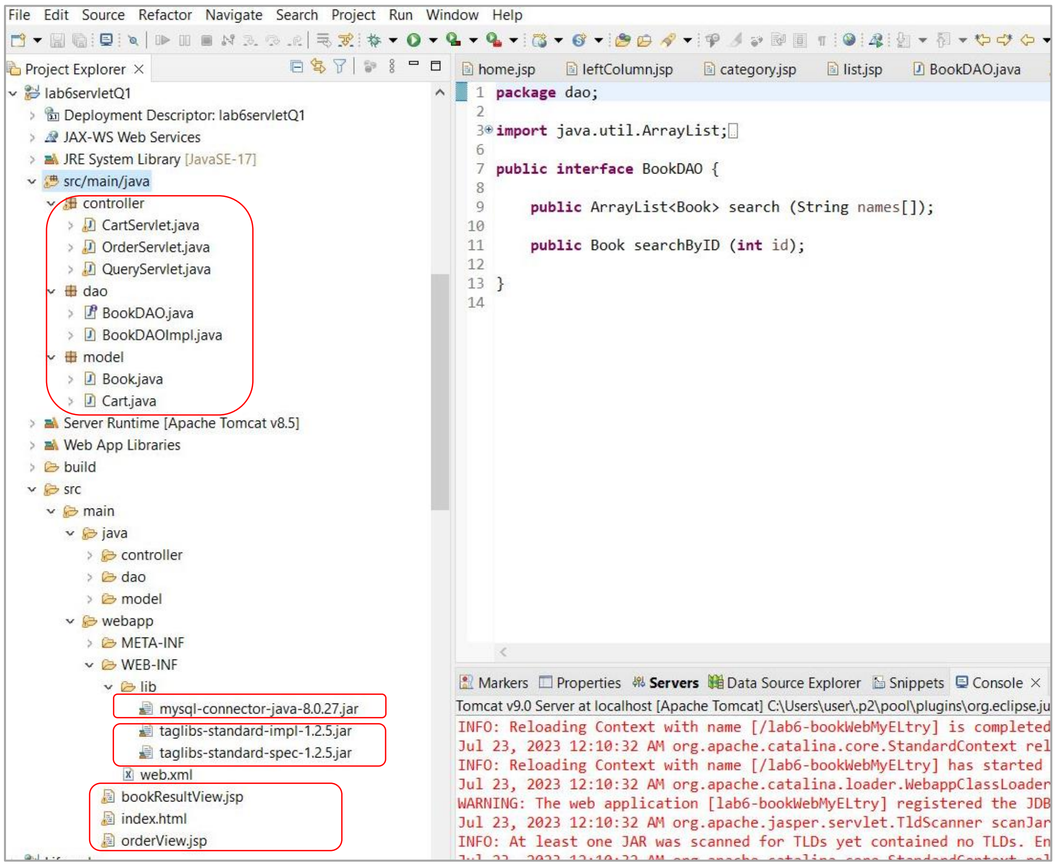
```
SELECT * FROM lab6.books;
```

id	title	author	price
1	A cup of Java	Sue Lee	22
2	Java IO	Sue Lee	72
3	Java for Dummy	Tan Ah Teck	22
4	More Java for Dummies	Tan Ah Teck	42
5	Good Java Style	Joe Suh	12
NULL	NULL	NULL	NULL

- Now import the **lab5servletQ2B.war** you submitted for lab5, import as **lab6servletQ1**.
- Remove the class Table.
- Modify the code of **QueryServlet** so that it given author names, loads the list of books from the database.
- If you searched for book in **CartServlet**, then modify the code of it so that it loads from the database.
- Make sure MySQL server is running. Put the connector Jar file into the lib directory of your project, as demonstrated in class and the video.
- Run the program. If implemented correctly, your program should give the same output as in lab5.

Part 1B DAO design pattern (as well as MVC model)

- The drawback of part 1 is that the database connect code is intermixed with the business logic code. It is good to have clear separation of business logic and low level database connection and retrieval. Moreover, if database connection information needs to be changed (e.g., using another database jar file), then all the connection code needs to be modified. A cleaner approach is the DAO pattern where database connection is handled by the class.
- If you haven't done in part 1A, create a package **controller**, put the servlet files into the package. Create **model** package, put Book and Cart class into there.
- Create a new package **dao**. Inside, create an interface **BookDAO**, which defines methods **public ArrayList<Book> search (String authors[])** and **public Book searchByID(int id)**
- Created class **BookDAOImpl**, which implements the interface, handling all database connection and retrieve books by authors or id. The structure of the project is given in the figure below.
- In **QueryServlet**, remove the database connection and data retrieval code that you did in part 1A. Instead, to get a list of books, create a instance of **BookBAOImpl** and call method on it (similar to call **Table** in previous labs). Following the principle of 'programming to interface', assign the instance to type **BookDAO**
- Make the similar change if you do book search in **CartServlet**.
- If implemented correctly, your program should get the same output.



```
lab6servletQ1
  Deployment Descriptor: lab6servletQ1
  JAX-WS Web Services
  JRE System Library [JavaSE-17]
  src/main/java
    controller
      CartServlet.java
      OrderServlet.java
      QueryServlet.java
    dao
      BookDAO.java
      BookDAOImpl.java
    model
      Book.java
      Cart.java
  Server Runtime [Apache Tomcat v8.5]
  Web App Libraries
  build
  src
    main
      java
        controller
        dao
        model
      webapp
        META-INF
        WEB-INF
          lib
            mysql-connector-java-8.0.27.jar
            taglibs-standard-impl-1.2.5.jar
            taglibs-standard-spec-1.2.5.jar
          web.xml
        bookResultView.jsp
        index.html
        orderView.jsp
```

```java
package dao;

import java.util.ArrayList;

public interface BookDAO {

    public ArrayList<Book> search (String names[]);

    public Book searchByID (int id);

}
```

```
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\user\.p2\pool\plugins\org.eclipse.ju
INFO: Reloading Context with name [/lab6-bookWebMyELtry] is completed
Jul 23, 2023 12:10:32 AM org.apache.catalina.core.StandardContext rel
INFO: Reloading Context with name [/lab6-bookWebMyELtry] has started
Jul 23, 2023 12:10:32 AM org.apache.catalina.loader.WebappClassLoader
WARNING: The web application [lab6-bookWebMyELtry] registered the JDB
Jul 23, 2023 12:10:32 AM org.apache.jasper.servlet.TldScanner scanJar
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. En
```

- Submit: export as was file with default name **lab6servletQ1.war**. Make sure to check 'export source files'

Part 2 JDBC, SQLite

In the exercise, you implement a Book search application, using SQLite, with more tables.

This application maintains a database of books. It allows users to search all books, search book by category, and can search by keyword.
In this application we use another database system: SQLite. SQLite is a serverless embedded database. Database is stored in a .db disk file. You can access the database by accessing the file.

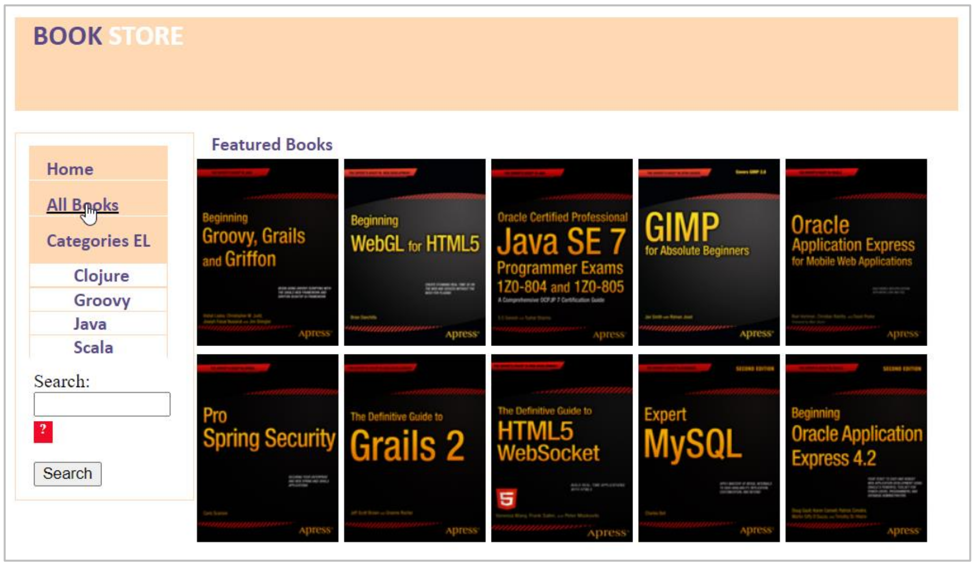
- You are given **Books.db**, which is the SQLite database file used by the application. It contains 3 tables. Download the file to your computer. You can use command line to access manipulate the database, but you can download a GUI for it. Recommend downloading **DB browser for SQLite**. With this tool you can open the database and view the tables there. The 3 tables are: BOOK, AUTHOR, CATEGORY. Study the structure of the tables.

- You are given the partially implemented program. Import the provided war file, as **lab6serveltQ2.** The program implements MVC and DAO pattern. The program contains Servlet, JSP, CSS and JS files. Study how the different languages implement different functionalities, which may help you for your project.
- Read the code and complete the code.
  - Complete the DAO class **BookDAOImpl**. Study the database and you will need to join 3 tables to get the needed information from the database. You need to point to the path of the database file **Books.db**.
  - Complete the controller file **BookController**, which uses DAO to retrieve the books.
  - Complete JS file **list.jsp** to output list of books. Use JSTL and EL. Don't use scriptlets to embed java code

- Put the provided SQLite JDBC jar file to the lib directory of your project.
- If implemented correctly, you will get the following results.

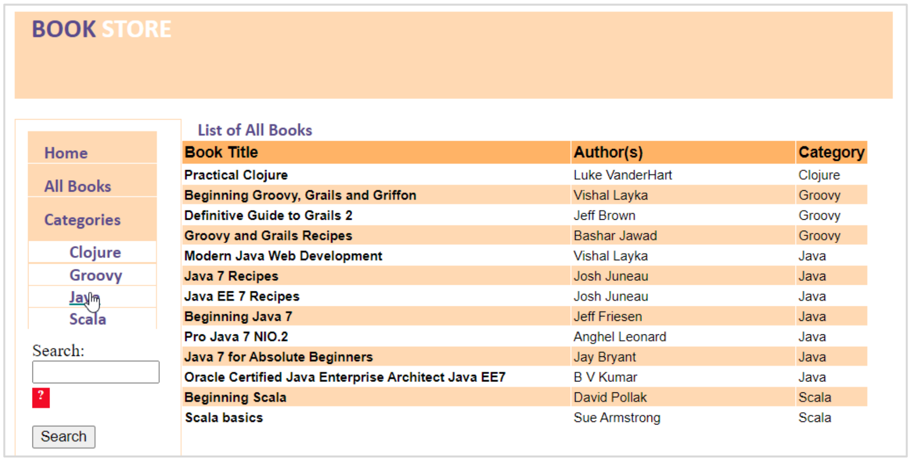
Home page.

Then click All Books


BOOK STORE
Featured Books
Home
All Books
Categories EL
Clojure
Groovy
Java
Scala
Search:
?
Search
Beginning Groovy, Grails and Griffon
Beginning WebGL for HTML5
Oracle Certified Professional Java SE 7 Programmer Exams 1Z0-804 and 1Z0-805
GIMP for Absolute Beginners
Oracle Application Express for Mobile Web Applications
Pro Spring Security
The Definitive Guide to Grails 2
The Definitive Guide to HTML5 WebSocket
Expert MySQL
Beginning Oracle Application Express 4.2

All Books search result. Totally 13 books.

Then, click Java Category


BOOK STORE

Home
All Books
Categories
Clojure
Groovy
Java
Scala

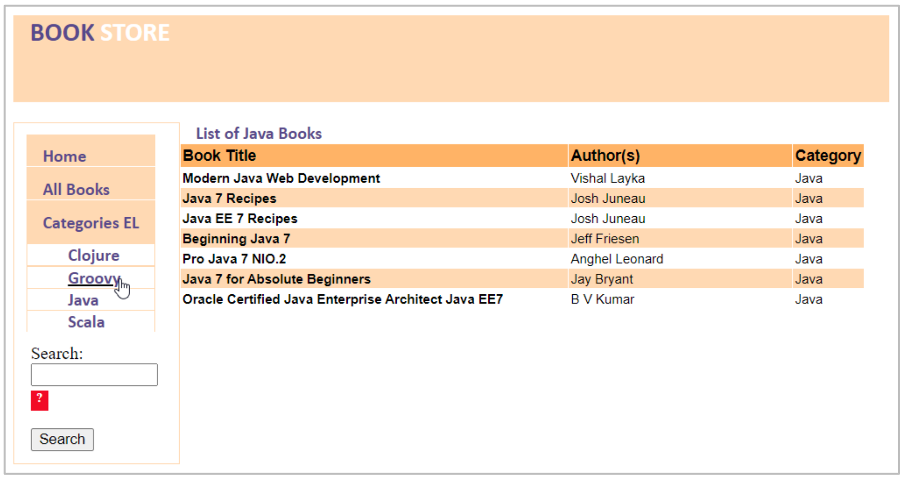
Search:
Search

List of All Books

Book Title	Author(s)	Category
Practical Clojure	Luke VanderHart	Clojure
Beginning Groovy, Grails and Griffon	Vishal Layka	Groovy
Definitive Guide to Grails 2	Jeff Brown	Groovy
Groovy and Grails Recipes	Bashar Jawad	Groovy
Modern Java Web Development	Vishal Layka	Java
Java 7 Recipes	Josh Juneau	Java
Java EE 7 Recipes	Josh Juneau	Java
Beginning Java 7	Jeff Friesen	Java
Pro Java 7 NIO.2	Anghel Leonard	Java
Java 7 for Absolute Beginners	Jay Bryant	Java
Oracle Certified Java Enterprise Architect Java EE7	B V Kumar	Java
Beginning Scala	David Pollak	Scala
Scala basics	Sue Armstrong	Scala

Java category search result.

Click Groovy Category


BOOK STORE

Home
All Books
Categories EL
Clojure
Groovy
Java
Scala

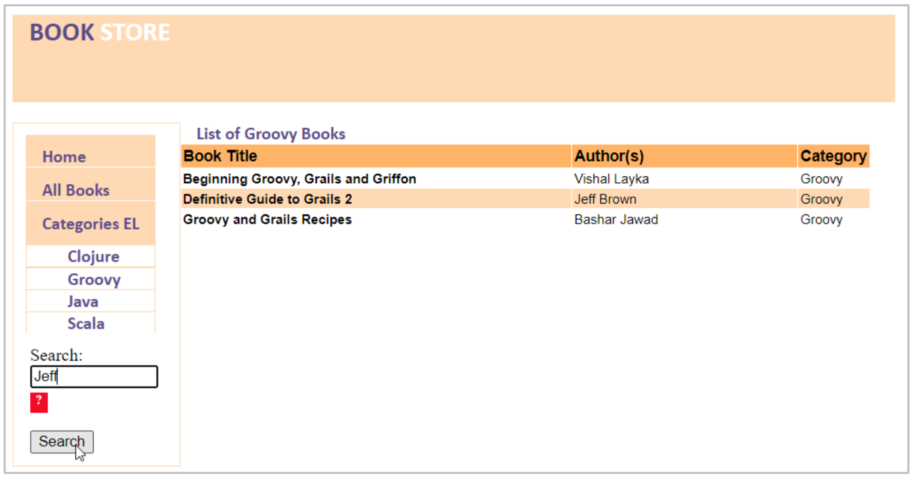
Search:
Search

List of Java Books

Book Title	Author(s)	Category
Modern Java Web Development	Vishal Layka	Java
Java 7 Recipes	Josh Juneau	Java
Java EE 7 Recipes	Josh Juneau	Java
Beginning Java 7	Jeff Friesen	Java
Pro Java 7 NIO.2	Anghel Leonard	Java
Java 7 for Absolute Beginners	Jay Bryant	Java
Oracle Certified Java Enterprise Architect Java EE7	B V Kumar	Java

Groovy category search result.

Then, search by keyword 'Jeff'


BOOK STORE

Home
All Books
Categories EL
Clojure
Groovy
Java
Scala

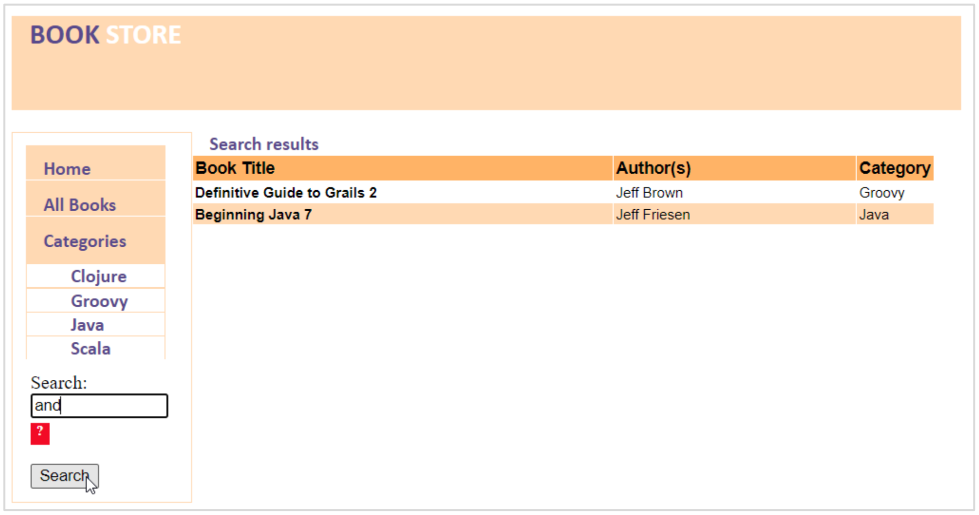
Search:
Jeff
Search

List of Groovy Books

Book Title	Author(s)	Category
Beginning Groovy, Grails and Griffon	Vishal Layka	Groovy
Definitive Guide to Grails 2	Jeff Brown	Groovy
Groovy and Grails Recipes	Bashar Jawad	Groovy

Keyword ‘Jeff‘ search result

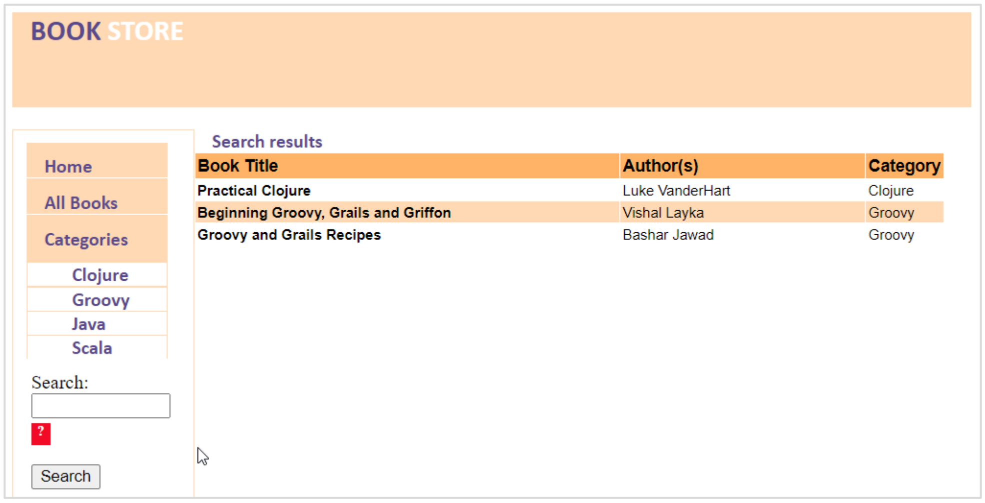
Then, search by keyword ‘and’



Search results

Book Title	Author(s)	Category
Definitive Guide to Grails 2	Jeff Brown	Groovy
Beginning Java 7	Jeff Friesen	Java

Keyword ‘and‘ search result



Search results

Book Title	Author(s)	Category
Practical Clojure	Luke VanderHart	Clojure
Beginning Groovy, Grails and Griffon	Vishal Layka	Groovy
Groovy and Grails Recipes	Bashar Jawad	Groovy

See video for more information.

- Submission: export as was file with default name **lab6servletQ2.war**. Make sure to check ‘explort source files’

Hope this program help you with the product search part of the project.

Submissions.

You should have exported **lab6servletQ1.war lab6servletQ2.war**. Make sure you have checked “Export source files” when exporting.

Submit the 2 war files separately on eClass.

Late submissions or submissions by email will NOT be accepted. Plan ahead and submit early.