LAB 4: USER INPUT HANDLING WITH SERVLET

Lab Activities

You are required to develop a Music Library web application using web technologies. The web application allows users to list the songs details such as title, year and genre. This lab stode the music data using an ArrayList.

1. Developing the View Servlet

In this exercise, you will develop the List library view servlet. This view generates a dynamic response by displaying the collection of musics to the user.

Task 1: Create the MusicItem Model file

The MusicItem model component is a class based on the JavaBeans architecture, which represents a single music's information.

a) Create the MusicLibrary Web application project with the following characteristic: Project: MusicLibrary
Location: C:\Users\User\Documents\NetBeansProjects
Folder: C:\Users\User\Documents\NetBeansProjects\ MusicLibrary
Server : GlassFish
Java EE version: Java EE
Contextpath: / MusicLibrary

b) Create a java class with the following characteristic: Class name: MusicItem
Project : MusicLibrary
Location:Source Package
Package : com.music.model

c) Modify the MusicItem class to implement the *java.io.Serializable* interface

d) Add the following private properties to the MusicItem class and provide a getter method for each property:

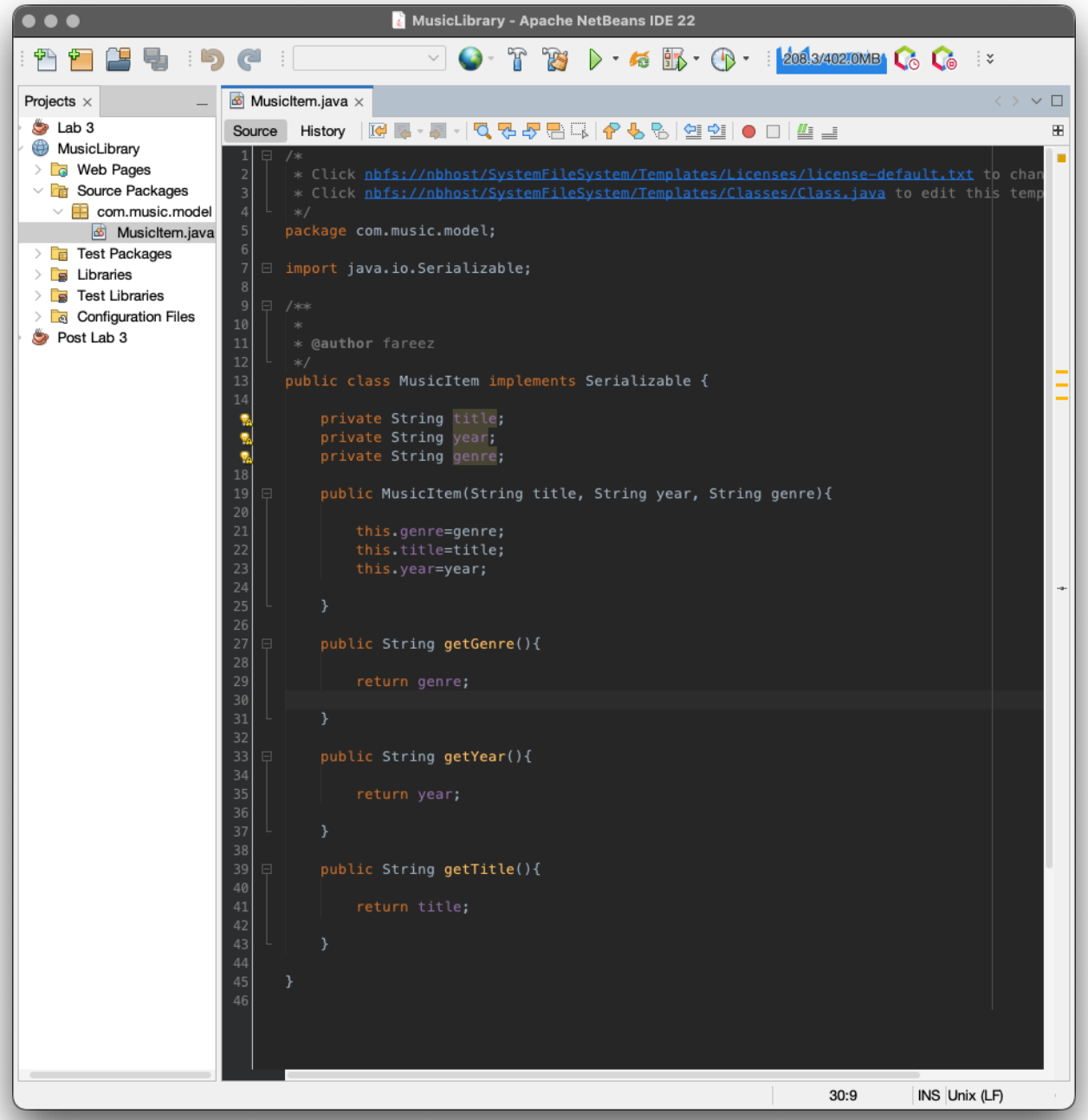
```
String title String
year
String genre
```

e) Add a constructor to the MusicItem class with the following parameters

```
String title String
year
String genre
```

f) Code the constructor to initialize the fields with the parameter values

g) Compile the MusicItem Class.


MusicLibrary - Apache NetBeans IDE 22
208.3/402.0MB
Projects
Lab 3
MusicLibrary
Web Pages
Source Packages
com.music.model
MusicItem.java
Test Packages
Libraries
Test Libraries
Configuration Files
Post Lab 3
MusicItem.java
Source
History
/*
* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to chan
* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this temp
*/
package com.music.model;
import java.io.Serializable;
/**
*
* @author fareez
*/
public class MusicItem implements Serializable {
private String title;
private String year;
private String genre;
public MusicItem(String title, String year, String genre){
this.genre=genre;
this.title=title;
this.year=year;
}
public String getGenre(){
return genre;
}
public String getYear(){
return year;
}
public String getTitle(){
return title;
}
}
30:9
INS
Unix (LF)

Task 2: Write the ListLibraryServlet View Servlet

In this task, you create a ListLibraryServlet to generate an HTML response that display the number of Musics in the user collection and their titles.

a) Create a Java Servlet with the following characteristic:
   Class name: ListLibraryServlet
   Project : MusicLibrary
   Location:Source Package
   Package : com.music.view
   Servlet name: ListLibraryServlet URL
   Pattern : /list_library.view
   Initialization Parameters : None

b) Tick on the checkbox Add information to deployment descriptor (web.xml)

b) Import the following classes in the ListLibraryServlet class: `java.util.List java.util.Iterator java.util.ArrayList com.music.model.MusicItem`

c) Within the processrequest method, create a collection of MusicItem objects, each one representing a Music in the user's Music collection.

```
List musics = new ArrayList();
musics.add(new MusicItem("Cardigan", "2020", "Indie Folk")); musics.add(new
MusicItem("Someone like you", "2011", "Pop/Soul")); musics.add(new
MusicItem("Fight Song", "2015", "Pop"));
```

d) Use print statements to generate the HTML response described at the beginning of this task to list the number of musics in your collection and display the title, year and genre in a table structure as below.

```
PrintWriter out = response.getWriter();

out.println("<!DOCTYPE html>");
out.println("<html>");
out.println("<head>");
out.println("<title>ListLibraryServlet</title>");
out.println("</head>");
out.println("<body>");
out.println("You currently have <b>" + musics.size()
        + "</b> Musics in your collection: <br>");
out.println("<table border='0' cellspacing='0' cellpadding='5'>");
out.println("<tr>");
out.println("<th>TITLE</th>");
out.println("<th>GENRE</th>");
out.println("<th>YEAR</th>");
out.println("</tr>");

Iterator it = musics.iterator();
while (it.hasNext()) {

 MusicItem item = (MusicItem) it.next();
 out.println("<tr>");
 out.println("<td>" + item.getTitle() + "</td>");
 out.println("<td>" + item.getGenre() + "</td>");
 out.println("<td>" + item.getYear() + "</td>");
 out.println("</tr>");
}

out.println("</table>");
out.println("</body>");
out.println("</html>");
out.close();
}
```

e) Compile the ListLibraryServlet servlet class.

2. Configuring the Web application

Task 1 : Verify the view servlet configuration

a) Open the web.xml deployment descriptor
b) View the definition and the mapping for the ListlibraryServlet servlet.

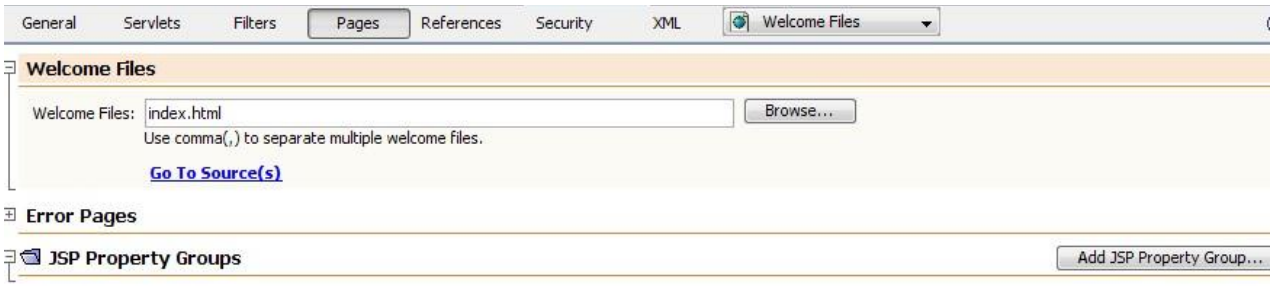
Task 2: Create the homepage

a) Create an HTML file with the following characteristics:
   Project : MusicLibrary
   HTML file name: index
   Location:Web Pages

b) Edit the index.html file so that it displays the following text.
   Display my Music Library
c) Edit the index.html file to add a link to the List library page.

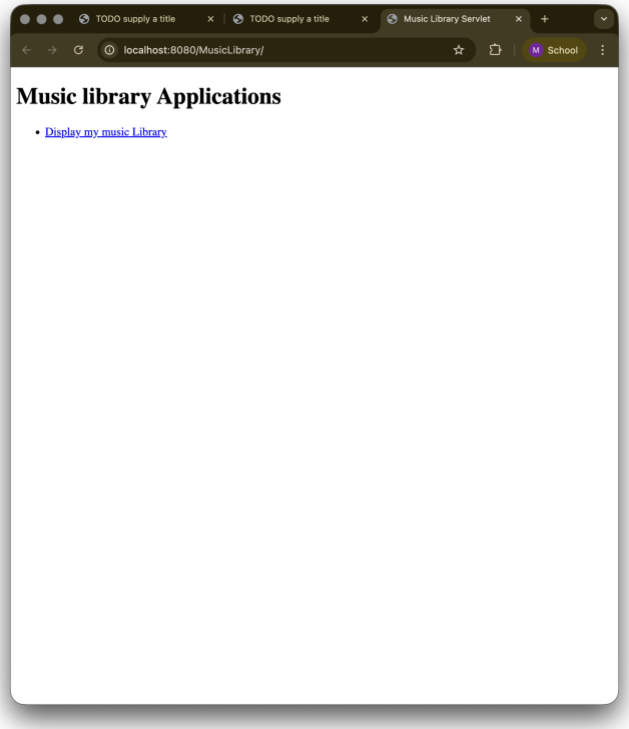
```
<html>
 <head>
  <title>Music Library Servlet</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
 </head>
 <body>
  <h1>Music Library Application</h1>
  <ul>
   <li>
    <a href="list_library.view">Display my music Library</a></li>
  </ul>
 </body>
</html>
```

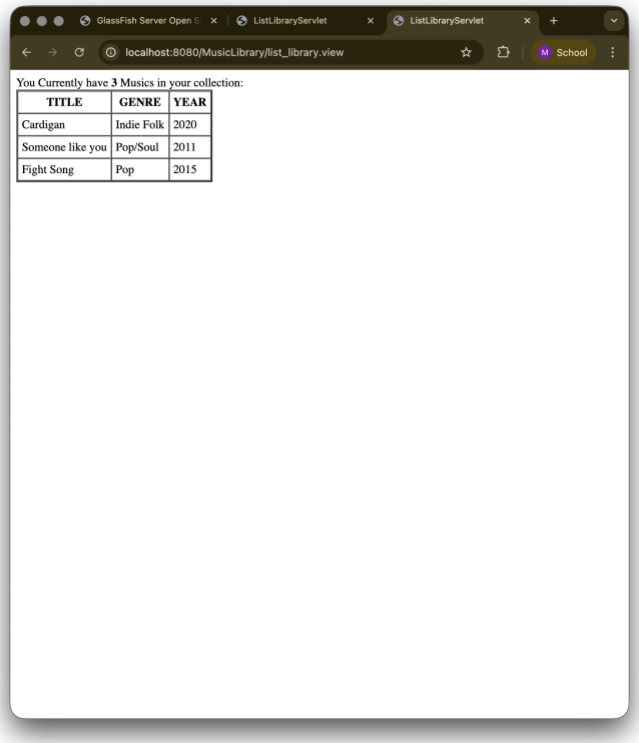
d) Open the web.xml deployment descriptor and set index.html as the welcome file.



Task 3: Deploy the web application

a) Build the MusicLibrary web application. Correct any errors you encounter
b) Deploy the MusicLibrary web Application
c) Run the web application project
d) Click the display my MusicLibrary link to display the list of Music.


TODO supply a title
TODO supply a title
Music Library Servlet
localhost:8080/MusicLibrary/
School
Music library Applications
Display my music Library


GlassFish Server Open S
ListLibraryServlet
ListLibraryServlet
localhost:8080/MusicLibrary/list_library.view
School
You Currently have 3 Musics in your collection:
TITLE
GENRE
YEAR
Cardigan
Indie Folk
2020
Someone like you
Pop/Soul
2011
Fight Song
Pop
2015

Postlab Exercise

You need to extend the functionality of the `ListLibraryServlet` and `Mu` Sort by Artist`sicItem` class by adding new property. Perform the following tasks:

1. Modify the `MusicItem` class

    a. Add a new property to the `MusicItem` class: String artist (Store the name of the music artist)

    b. Update the constructor, assesor and mutator methods.

```
package com.music.model;

import java.io.Serializable;

public class MusicItem implements Serializable {

   private String title;
   private String year;
   private String genre;
   private String artist;

   public MusicItem(String title, String year, String genre, String artist){

      this.genre=genre;
      this.title=title;
      this.year=year;
      this.artist=artist;

   }

   public String getGenre(){

      return genre;

   }

   public String getYear(){

      return year;

   }

   public String getTitle(){
```

```
        return title;

    }
    public String getArtist(){

        return artist;

    }
    public void setTitle(String title) { this.title = title; }
    public void setYear(String year) { this.year = year; }
    public void setGenre(String genre) { this.genre = genre; }
    public void setArtist(String artist) { this.artist = artist; }
}
```

2. Modify the `ListLibraryServlet`

    a. Update the `ListLibraryServlet` to include the new attribute in the music list display.

    b. Modify the servlet to allow sorting of the music collection by year or artist.

```
package com.music.view;
import com.music.model.MusicItem;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.util.*;

public class ListLibraryServlet extends HttpServlet {


  protected void processRequest(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");


    List musics = new ArrayList();
    musics.add(new MusicItem("Cardigan", "2020", "Indie Folk", "Jamal Abdillah"));
    musics.add(new MusicItem("Someone like you", "2011", "Pop/Soul", "Adele"));
    musics.add(new MusicItem("Fight Song", "2015", "Pop", "Rachel Platten"));

    String sortBy = request.getParameter("sort");
```

```
if (sortBy != null) {
   if (sortBy.equals("year")) {
      musics.sort(Comparator.comparing(MusicItem::getYear));
   } else if (sortBy.equals("artist")) {
      musics.sort(Comparator.comparing(MusicItem::getArtist));
   }
}

   try (PrintWriter out = response.getWriter()) {
      /* TODO output your page here. You may use following sample code. */
      out.println("<!DOCTYPE html>");

      out.println("<html>");

         out.println("<head>");
            out.println("<title>ListLibraryServlet</title>");
         out.println("</head>");

         out.println("<body>");

            out.println("<a href='list_library.view?sort=year'>Sort by Year</a> | ");
            out.println("<a href='list_library.view?sort=artist'>Sort by Artist</a><br><br>");

            out.println("You Currently have <b>" + musics.size() + "</b> Musics in your collection: <br>");
            out.println("<table border='2', cellspacing='0', cellpadding='5'");

               out.println("<tr>");
               out.println("<th>TITLE</th>");
               out.println("<th>GENRE</th>");
               out.println("<th>YEAR</th>");
               out.println("<th>ARTIST</th>");
               out.println("</tr>");

               Iterator it = musics.iterator();
               while(it.hasNext()){

                  MusicItem item = (MusicItem) it.next();
                  out.println("<tr>");
                  out.println("<td>" + item.getTitle() + "</td>");
                  out.println("<td>" + item.getGenre() + "</td>");
                  out.println("<td>" + item.getYear() + "</td>");
                  out.println("<td>" + item.getArtist() + "</td>");
                  out.println("</td>");
               }

            out.println("</table>");

         out.println("</body>");
```

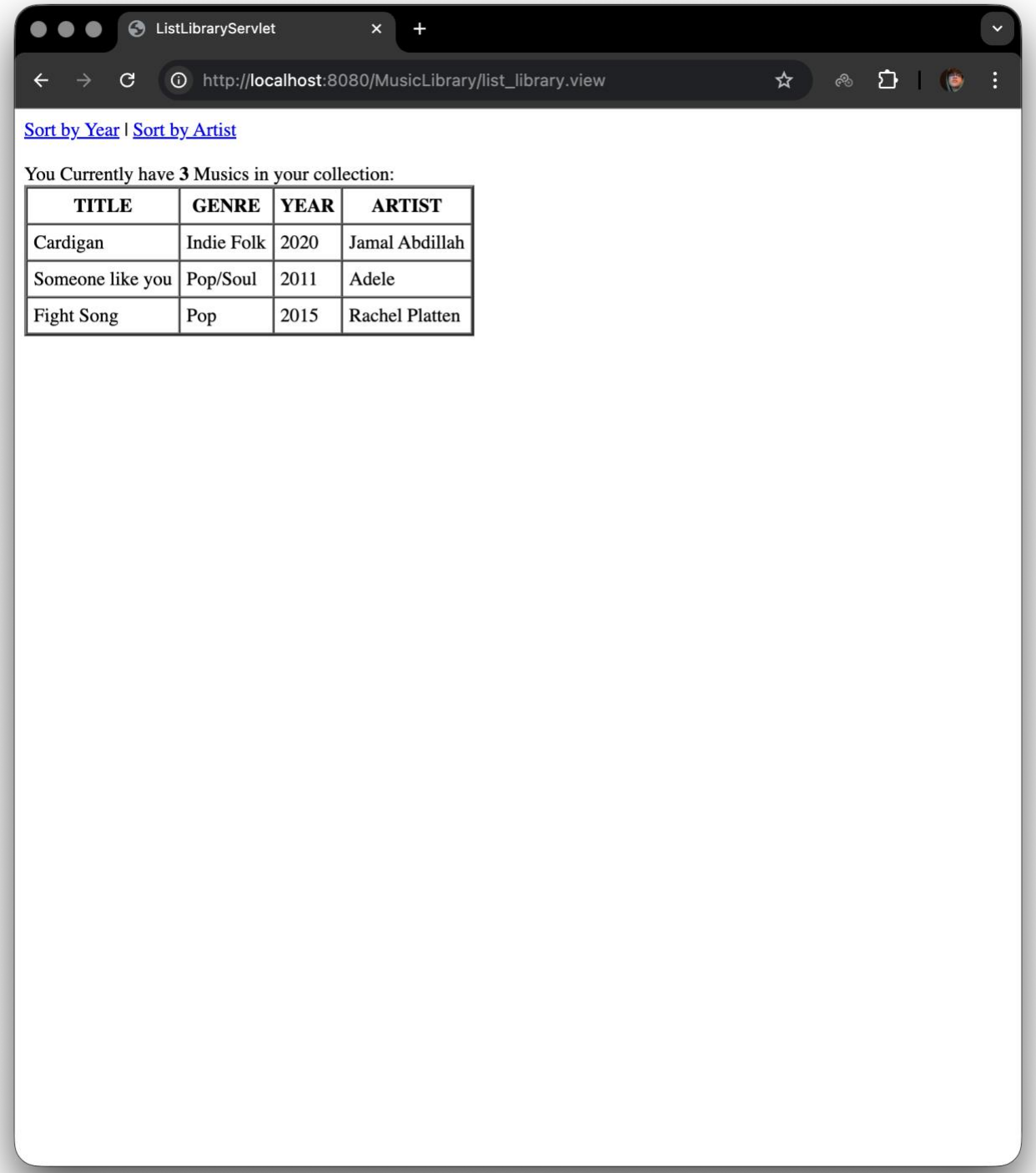
```
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to
edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```


ListLibraryServlet
http://localhost:8080/MusicLibrary/list_library.view
Sort by Year | Sort by Artist
You Currently have 3 Musics in your collection:
TITLE
GENRE
YEAR
ARTIST
Cardigan
Indie Folk
2020
Jamal Abdillah
Someone like you
Pop/Soul
2011
Adele
Fight Song
Pop
2015
Rachel Platten