INFS3202/7202 Practical 5
Server Side Scripting

You must present this practical to your lab tutor during your scheduled lab sessions in week 8 that starts 23/04/2012. The prac could be done either in lab, or at home.

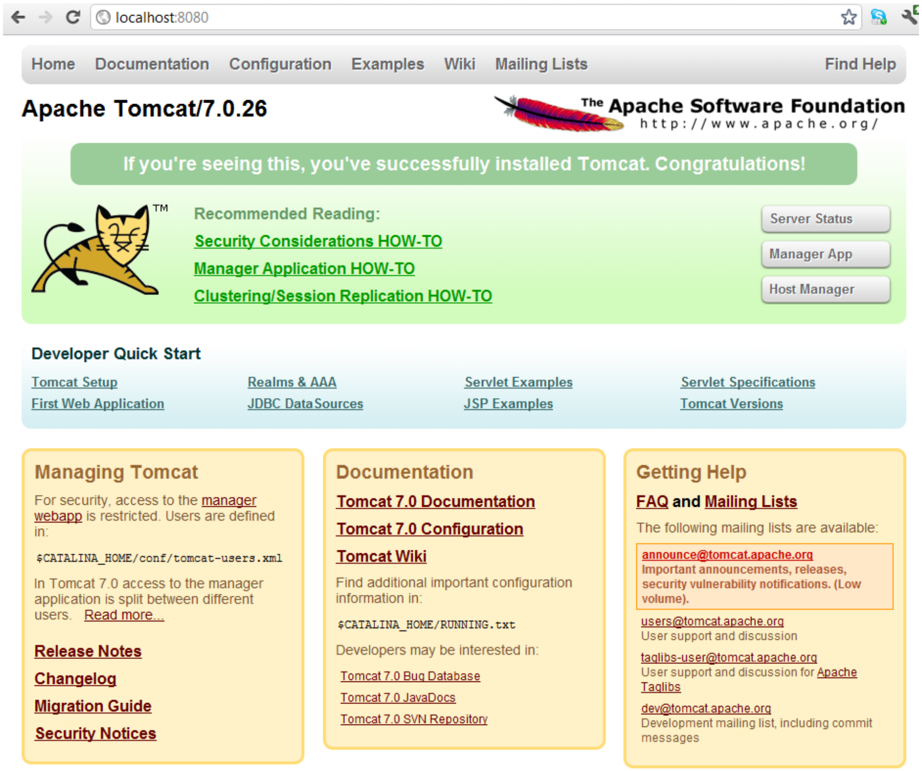
Preparation

Before attempting this practical you should have a good working knowledge of HTML/XHTML (Client Side) and Java Servlets & JSP (Server Side) you should understand how client server communication works and the Java Servlet architecture for building web applications, and please ensure:

○ You have covered the material in Lectures 1- 7.
○ You should have installed(if not using lab machines)Apache Tomcat
○ Understand how to build Java Web Applications using Java Servlets and JSP
○ investigate file manipulation using Java/JSP

Using Tomcat in the Lab

○ Follow the lecture notes to download, install and start Tomcat in your H: drive.
○ If you can successfully install and start Tomcat, then a page similar to this one should be displayed when you type http://localhost:8080/



Apache Tomcat/7.0.26

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Servlet Examples
JSP Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 7.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation

Tomcat 7.0 Documentation
Tomcat 7.0 Configuration
Tomcat Wiki

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

Tomcat 7.0 Bug Database
Tomcat 7.0 JavaDocs
Tomcat 7.0 SVN Repository

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

announce@tomcat.apache.org
Important announcements, releases, security vulnerability notifications. (Low volume).

users@tomcat.apache.org
User support and discussion

taglibs-user@tomcat.apache.org
User support and discussion for Apache Taglibs

dev@tomcat.apache.org
Development mailing list, including commit messages

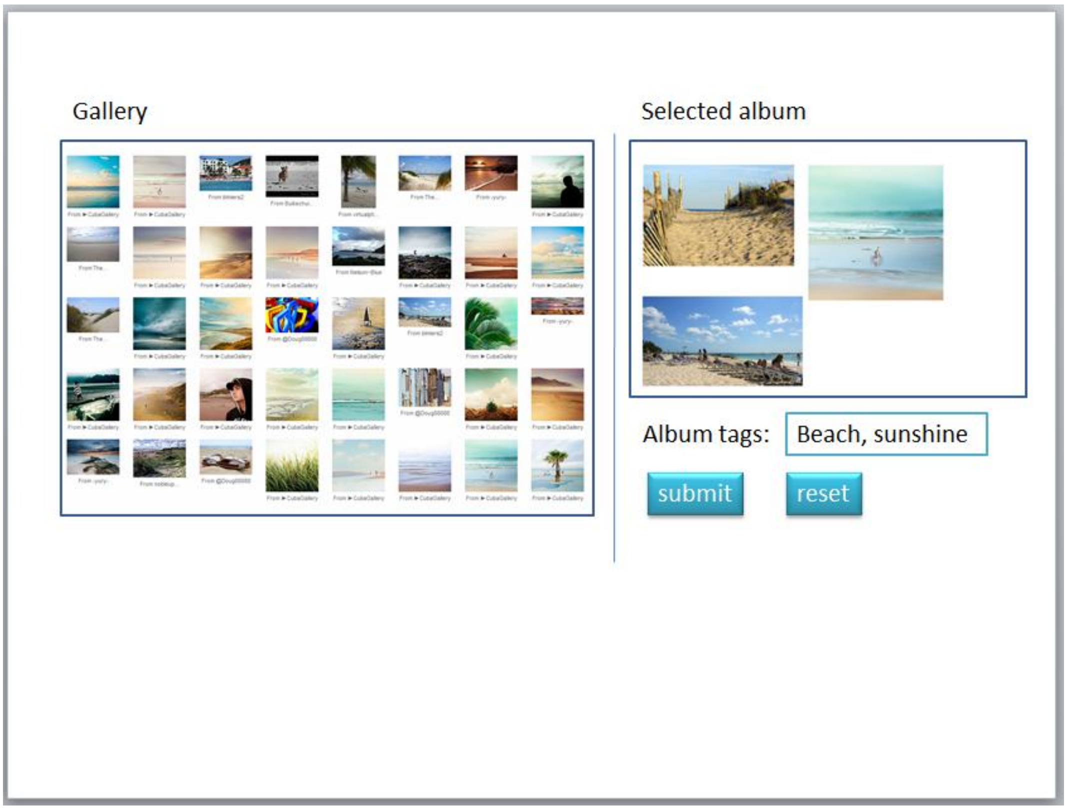
This prac should be done using JSP or Java Servlets.

Task1: Redesign gallery page by using JSP (1 mark)

In Prac 1, you have designed a gallery page at client side. Now you need to use JSP to redesign it and count the total number of accesses to this page since server reboot. This number should be displayed on the gallery page.

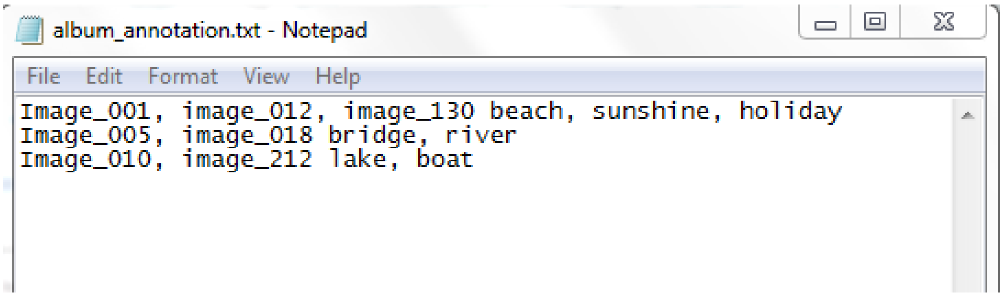
Task2: Make album with tags (4 marks)

In the task2 of Prac 2, a page is designed to implement drag-n-drop. Now, you are required to design a form on the page to input the tags for the selected album. Here is an example:



Beach, sunshine

When the tags are submitted, a new entry will be written to an annotation file named “album_annotation.txt” which is used to store all album annotations. All of the image Ids (or image file names) in the selected album div and the tags provided by the user should be stored. The file looks like this:



```
Image_001, image_012, image_130 beach, sunshine, holiday
Image_005, image_018 bridge, river
Image_010, image_212 lake, boat
```

“Image_001”, “Image_012” and “Image_130” are the Ids of the images in the first album. “Beach”, “sunshine” and “holiday” are the associated album tags.

JavaBeans should be used to handle the form. When the form is submitted, servlet page should call the functions from bean to write ids and tags to target txt file. For example, assume my JavaBean class name is Filewriter. I can simply call Filewiter.write(ids, tags); to write ids and tags to file. The reason to implement JavaBean class is to make your code reusable.

Hint:

- To retrieve the image ids from the selected album div, you may store the Id on the page by using <input> tags when every time ondrop event triggers.

  Here is an example:

  In the JSP page, these tags can be generated when ondrag event triggers.

  ```
  <input type="hidden" name="id" value="img001" />
  <input type="hidden" name="id" value="img002" />
  <input type="hidden" name="id" value="img003" />
  ```

  Then you can use getParameterValues() to retrieve the ids in servlet page.

  ```
  String [] ids = request.getParameterValues(“id”);
  ```

- To submit image ids with the inputted tags together, the <input> tags for storing image Ids may put inside the form. Then you can get the ids and image tags by their name attribute after form submitted.

- To implement JavaBean, you may need to compile your java code in Eclipse, NetBeans or terminal. Then put the .class file under “tomcat/webapps/your project/WEB-INF/classes" folder.

- Here is a reference for JSP file writing.
  http://docs.oracle.com/javase/1.4.2/docs/api/java/io/PrintWriter.html