Week-7: **Date: 11-08-08**

AIM: Install TOMCAT web server and APACHE.

While installation assign port number 8080 to APACHE. Make sure that these ports are available i.e., no other process is using this port.

DESCRIPTION:

- ***Set the JAVA_HOME Variable***

You must set the `JAVA_HOME` environment variable to tell Tomcat where to find Java. Failing to properly set this variable prevents Tomcat from handling JSP pages. This variable should list the base JDK installation directory, not the bin subdirectory.

On Windows XP, you could also go to the Start menu, select Control Panel, choose System, click on the Advanced tab, press the Environment Variables button at the bottom, and enter the `JAVA_HOME` variable and value directly as:

Name: JAVA_HOME

Value: C:\jdk

- ***Set the CLASSPATH***

Since servlets and JSP are not part of the Java 2 platform, standard edition, you have to identify the servlet classes to the compiler. The server already knows about the servlet classes, but the compiler (i.e., `javac`) you use for development probably doesn't. So, if you don't set your `CLASSPATH`, attempts to compile servlets, tag libraries, or other classes that use the servlet and JSP APIs will fail with error messages about unknown classes.

Name: JAVA_HOME

Value: *install_dir/common/lib/servlet-api.jar*

- ***Turn on Servlet Reloading***

The next step is to tell Tomcat to check the modification dates of the class files of requested servlets and reload ones that have changed since they were loaded into the server's memory. This slightly degrades performance in deployment situations, so is turned off by default. However, if you fail to turn it on for your development server, you'll have to restart the server every time you recompile a servlet that has already been loaded into the server's memory.

Technical Symposium

To turn on servlet reloading, edit *install_dir/conf/server.xml* and add a `DefaultContext` subelement to the main `Host` element and supply `true` for the `reloadable` attribute. For example, in Tomcat 5.0.27, search for this entry:

```
<Host name="localhost" debug="0" appBase="webapps"  ...>
```

and then insert the following immediately below it:

```
<DefaultContext reloadable="true"/>
```

Be sure to make a backup copy of *server.xml* before making the above change.

- ***Enable the Invoker Servlet***

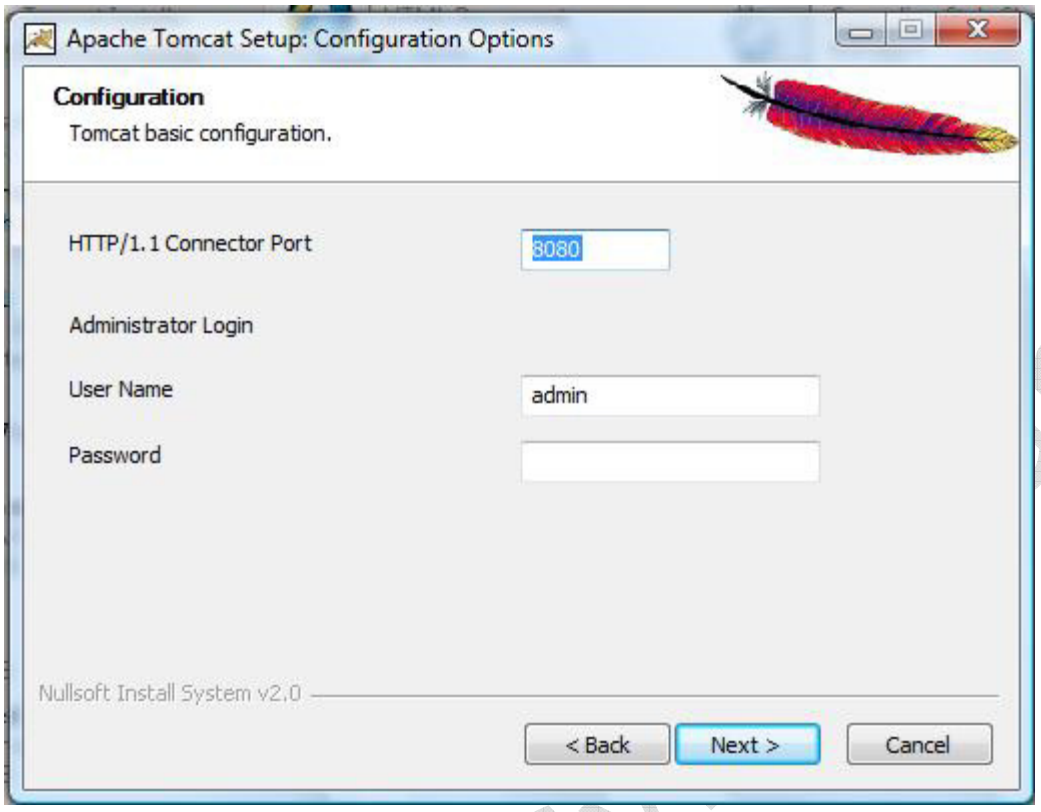
The invoker servlet lets you run servlets without first making changes to your Web application's deployment descriptor. Instead, you just drop your servlet into *WEB-INF/classes* and use the URL *http://host/servlet/ServletName*. The invoker servlet is extremely convenient when you are learning and even when you are doing your initial development.

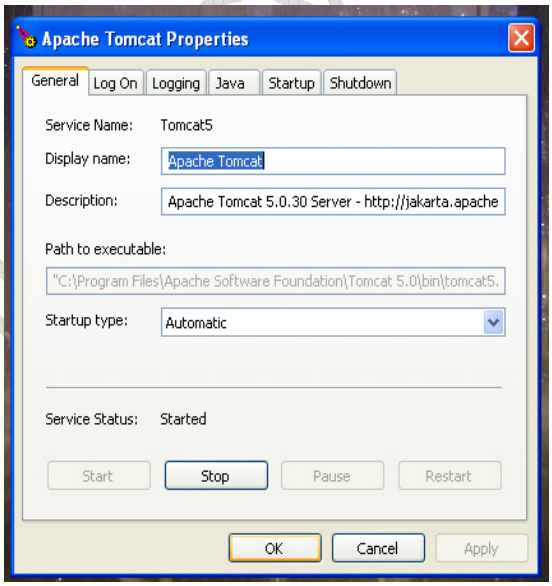
To enable the invoker servlet, uncomment the following `servlet` and `servlet-mapping` elements in *install_dir/conf/web.xml*. Finally, remember to make a backup copy of the original version of this file before you make the changes.

```
<servlet>
  <servlet-name>invoker</servlet-name>
  <servlet-class>
   org.apache.catalina.servlets.InvokerServlet
  </servlet-class>
  ...
</servlet>
...
<servlet-mapping>
  <servlet-name>invoker</servlet-name>
  <url-pattern>/servlet/*</url-pattern>
</servlet-mapping>
```

OUTPUT :

Technical Symposium



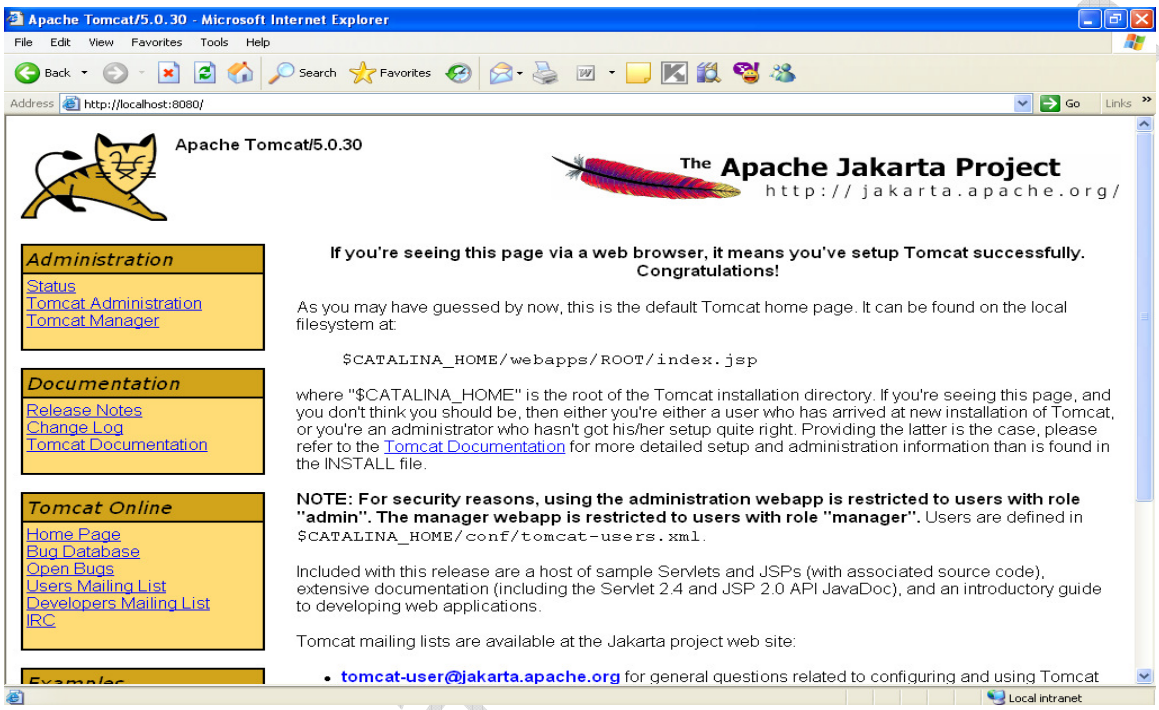


RESULT: Thus TOMCAT web server was installed successfully.

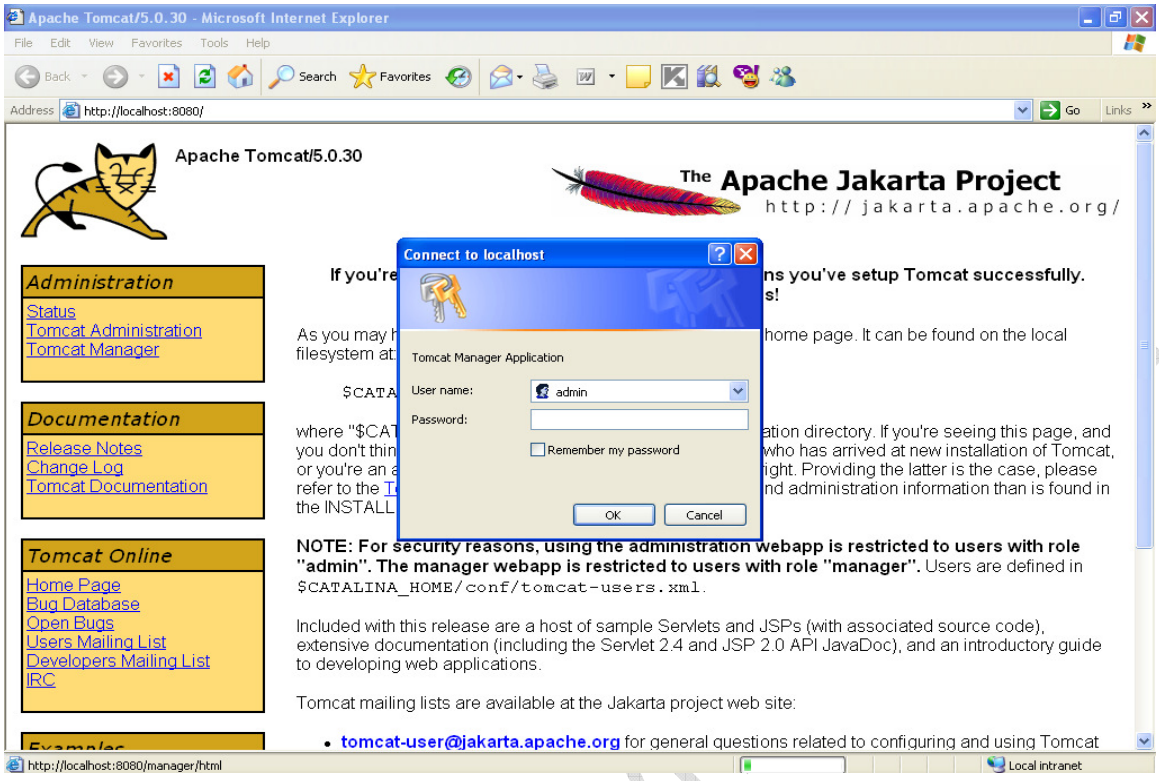
Week-7: **Date: 11-08-08**

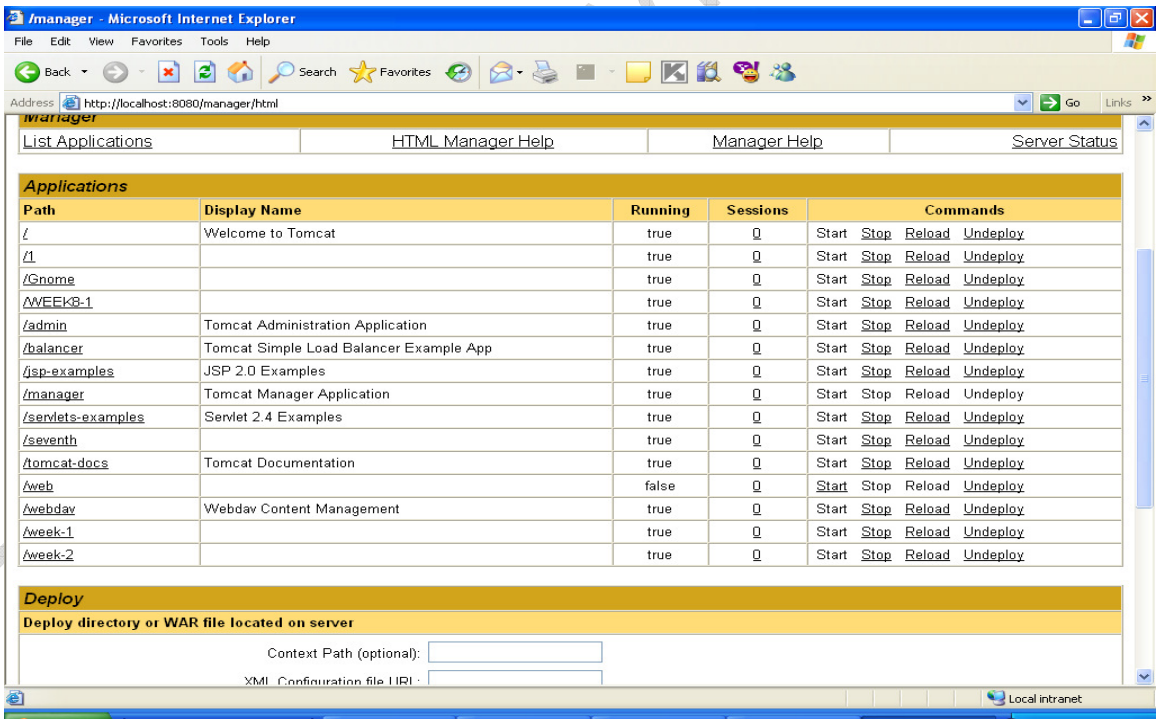
AIM: Access the developed static web pages for books web site, using these servers by putting the web pages developed in week-1 and week-2 in the document root.

OUTPUT



Technical





RESULT:

Thus week-1 and week-2 pages are accessed using the TOMCAT web server successfully

Technical Symposium