Dear students,

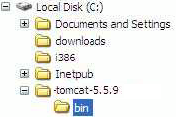
I hope all you are working on client side portion of your projects (HTML, CSS, Java Script). Next task, you’ll have to do is the server side portion (i.e. dynamic e.g. Database handling, passing data from one page to another etc. through java). Also, all the next lecture are on server side programming so, you must try and configure it.

For this, first of all you need a server (a software) running on your machines. We’ll use **apache tomcat**server.

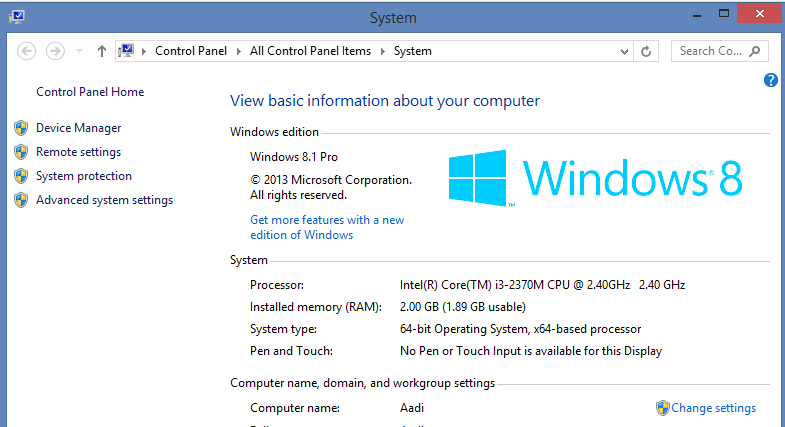
Here I’m guiding you step by step on how to install tomcat and create a basic dynamic application on your systems. Please do it before coming to class on Tuesday, as this is very important and must to complete your projects.

**Step 1 (Download and configure tomcat)**

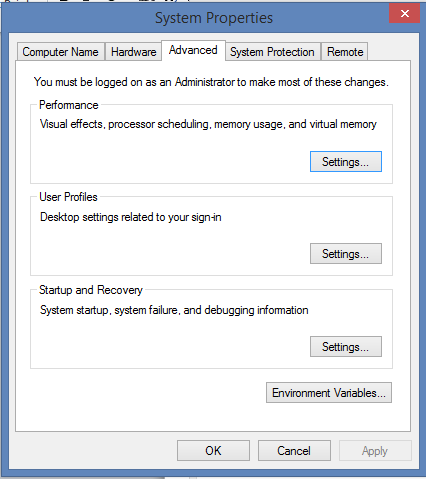
1. Download tomcat, you can use the one \\printsrv\Teacher Data\Aisha\IP\software or download it from internet.
2. Extract this folder and place it in C drive or any other drive.



1. Once it is placed in drive. Now, right click My Computer -> Properties, following window will appear.



1. Click on advanced system Settings.



1. Click Environment Variable, following dialog appears:



1. Click on New, Now you have to add three new environment variables JAVA\_HOME, CATALINA\_HOME, CLASSPATH.
   1. To add JAVA\_Home, click on new and write JAVA\_HOME in variable name and add your system’s jdk path in variable value.



Click OK to save it.

* 1. Similarly add second variable named CATALINA\_Home, here you have to give path of you tomcat folder that you pasted in step 2, and click ok.



* 1. Lastly add CLASSPATH variable, in it you have to add path or two files:
* C:\jakarta-tomcat-5.5.9\common\lib\**servlet-api.jar**
* C:\jakarta-tomcat-5.5.9\common\lib\**jsp-api.jar**
* Both these api’s are specified as values with semicolon between them. Remember to addsemicolon dot semicolon (;.;) at the end too. For example

Variable Name = Classpath

Variable Value = C:\tomcat-5.5.9\common\lib\servlet-api.jar**;**C:\tomcat5.5.9\common\lib\jsp-api.jar**;.;**



Save it and now you have configured your tomcat.

**Step 2 ( Test if tomcat is configured correctly)**

1. Run tomcat, to do it Open the C:\tomcat-5.5.9\bin folder and locate the startup.bat file.
2. Double clicking on this file will open up a DOS window, which will disappear, and another DOS window will appear, the second window will stay there(do not close it). If it does not your paths are not correctlyset.
3. Now to check whether your server is working or not, open up a browser window and type

http://localhost:8080. This should open the default page of Tomcat as shown in next diagram:

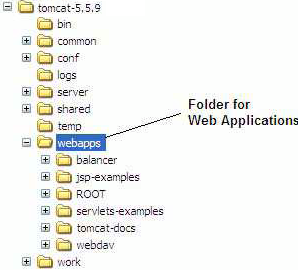


Now you have installed your tomcat and its running. Next step is to create an example, as we discussed in class.

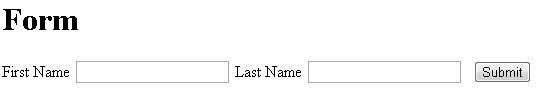
**Step 3 ( Create Directory structure for your project)**

For example you want to name your project MyfirstWebProject. First of all you have to create the directory structure as we discussed in class.

1. Go to c:\your tomcat folder\webapps folder and create a new folder named MyfirstWebProject.



1. Create a simple HTML pagee.g. **index.html**with two textboxes and a button on it like below



|  |
| --- |
| **Index.html**  <html>  <head>  </head>  <body>  <h1> Form </h1>  <form action="DataServlet" method="get">  First Name <input name="fn" type="text" ></input>  Last Name <input name="ln" type="text" ></input>  <input name="go" type="submit" ></input>  </form></body>  </html> |

1. Now, create WEB-INF folder in your project folder.



1. Goto WEB-INF folder and create “web.xml” file (open notepad and click save “with name web and extension xml”) and “classes”folder as below.



1. Now you have to create your servlet(java class). For this go to classes folder and create a file with java extension in notepad. Name this file as “DataServletExample.java”. paste the following code into it.

|  |
| --- |
| **DataServletExample.java**  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;  public class DataServletExample extends HttpServlet {    public DataServletExample() {  super();  }  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  {  PrintWriter pw=response.getWriter();  pw.println("Hello" + request.getParameter("fn") +" " + request.getParameter("ln"));  }  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  {    }  } |

1. Now you have to compile this java file, so go to commandline and write

javac your\_DataServletExample.java\_file\_path

It will create DataServletExample.class file.

Place this class file also in your project’s classes folder.

1. Last thing is to write in your web.xml file (that you created in 4th step). Open that file with notepad and paste following

|  |
| --- |
| <?xml version="1.0" encoding="ISO-8859-1"?>  <web-app>  <display-name>Servlet 2.4 Examples</display-name>  <description>  Servlet 2.4 Examples.  </description>  <servlet>  <servlet-name>DataServletExample</servlet-name>  <servlet-class>DataServletExample</servlet-class>  </servlet>  <servlet-mapping>  <servlet-name>DataServletExample</servlet-name>  <url-pattern>/DataServlet</url-pattern>  </servlet-mapping>  </web-app> |

1. You project is complete now. Time to test it. Open your browser and type following url:

http://localhost:8080/MyfirstWebProject/index.html

It should open your index page, enter some value in first name and last name and submit. Entered values should appear now with a hello.

**If you have successfully run it on your computer now you can easily add java (dynamic code) to your web sites along with html, css, java script.**

**Good luck ☺**

Once again, please do it before next class so that you can work on your projects, and all next lectures are based on it. If you have not tried it you cannot do the next lectures’ practice, also cannot complete your project. So, try not to delay.