

## Lab 1.1 Web Environment Setup-Java/Tomcat/Eclipse

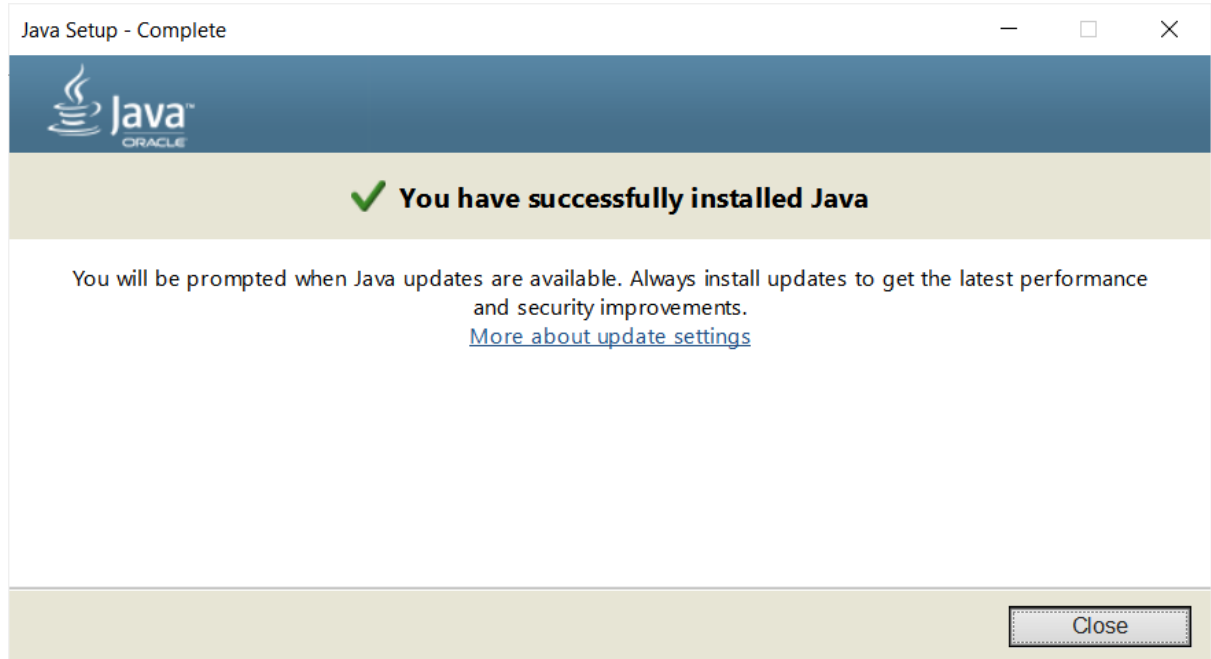
To Do List : Learn how to setup the web environment, include Java, Tomcat and Eclipse. It is the very base of all our further work about Web.

Steps for Windows 10 operating system:

1. Setup the Java.

Download the java from this link [http://www.java.com/zh\\_CN/](http://www.java.com/zh_CN/).

Run the Java installer and check the version.



Checked the version by using command java -version.

```
C:\Users\ASUS>java -version
Picked up _JAVA_OPTIONS: -Xmx512M
java version "1.8.0_291"
Java(TM) SE Runtime Environment (build 1.8.0_291-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.291-b10, mixed mode)
```

2. Setup the Tomcat

Download the Tomcat Installer from this link

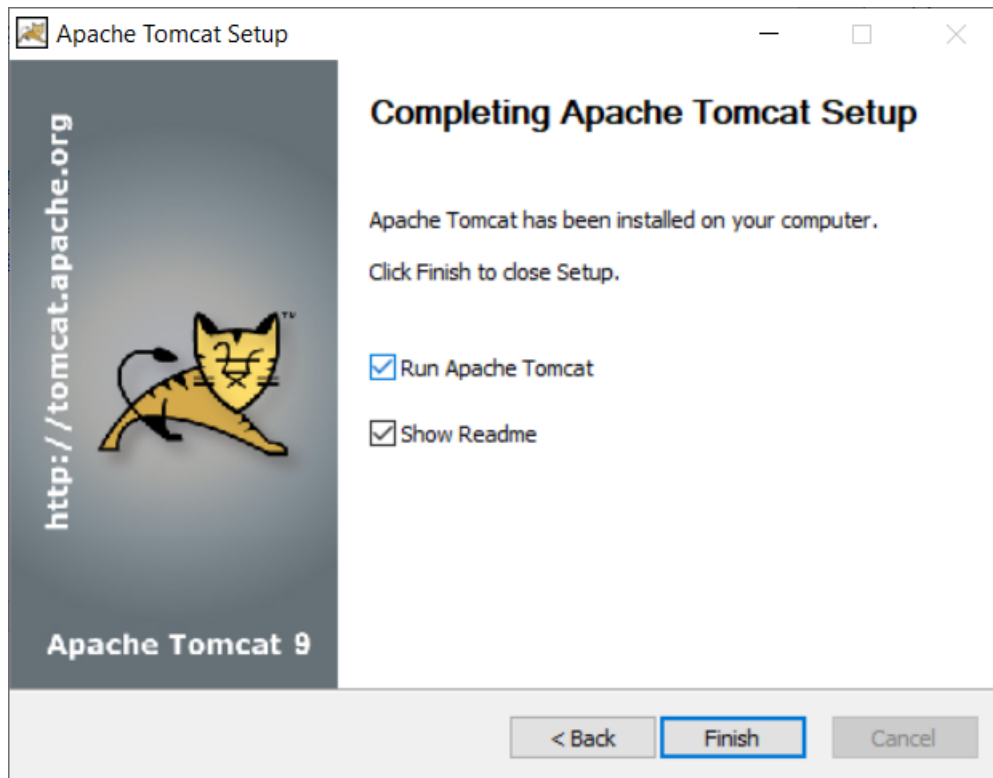
<https://downloads.apache.org/tomcat/tomcat-9/v9.0.46/bin/apache-tomcat-9.0.46.exe>

Or link from this website <https://tomcat.apache.org/download-90.cgi>

- Core:

- [zip](#) ([pgp](#), [sha512](#))
- [tar.gz](#) ([pgp](#), [sha512](#))
- [32-bit Windows zip](#) ([pgp](#), [sha512](#))
- [64-bit Windows zip](#) ([pgp](#), [sha512](#))
- [32-bit/64-bit Windows Service Installer](#) ([pgp](#), [sha512](#))

click the installer



Tomcat v9.0 Install Complete

### 3. Setup the Eclipse

Download Eclipse installer from this link <https://www.eclipse.org/downloads/>



## Develop a “Hello World!”


```
helloworld.java x
```

```
1  
2 public class helloworld {  
3     public static void main(String[] args) {  
4         System.out.println("Hello World!");  
5     }  
6 }
```

Problems @ Javadoc Declaration Console Debug Coverage

<terminated> helloworld (1) [Java Application] C:\Program Files\Java\jre1.8.0\_291\t  
Picked up \_JAVA\_OPTIONS: -Xmx512M  
Hello World!

Develop “Hello World!” on web page. (.jsp file)


  
 Hello World!

## Lab 1.2 Implementation of Web Application

## Overview (List to do)

In lab 1.1, you have learned to setup the Web Programming Environment. It is easy, but very important, which is the base of lab 1.2.

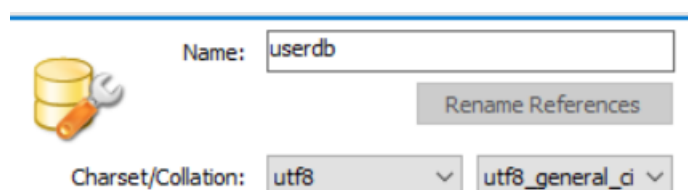
In this lab, you will learn how to implement a web application. Usually, a web application is related with database, web server, data access, service, page design and so on. So there are really much you have to learn before you start the Implementation.

Back to the lab, what's our objective:

1. Design and access the database (MySQL recommended).
2. Learn to use web server and deploy your web application (Tomcat recommended).
3. Implement the application (Html, JSP recommended).
4. CSS is required to be used for controlling the page display in the “External Style Sheet” way.

**Environment (IDE) :** Eclipse (2021-03), MySQL, Tomcat v9.0, bootstrap v5.0.1

**Database** : userdb , **Table** : records, login



```
CREATE SCHEMA `userdb` DEFAULT CHARACTER SET utf8mb4 ; (SQL query)
```

## SP2021-LAB1.1&1.2-姚熙源-3190300677

Table Name: records		Schema: userdb								
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	
id	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(60)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
course	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
fee	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

records table

CREATE TABLE `userdb`.`records` (`id` INT(10) NOT NULL, name` VARCHAR(60) NOT NULL, course` VARCHAR(100) NOT NULL, `fee` INT(10) NOT NULL, PRIMARY KEY (`id`)) ENGINE = InnoDB DEFAULT CHARACTER SET = utf8; (SQL query)

Table Name: login		Schema: userdb								
Charset/Collation: utf8 utf8_bin		Engine: InnoDB								
Comments:										
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	
userid	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
password	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

login table

CREATE TABLE `userdb`.`login` (`userid` VARCHAR(45) NOT NULL, `password` VARCHAR(45) NOT NULL, UNIQUE INDEX `userid\_UNIQUE` (`userid` ASC)) ENGINE = InnoDB DEFAULT CHARACTER SET = utf8; (SQL query)

JSP file : delete.jsp, index.jsp, login.jsp, update.jsp

- ▼ CourseRegistration
  - > Deployment Descriptor: CourseRegistration
  - > JAX-WS Web Services
  - > JRE System Library [JavaSE-1.8]
  - > src/main/java
  - > Apache Tomcat v9.0 [Apache Tomcat v9.0]
  - > Web App Libraries
  - ▼ bootstrap
    - > css
    - > js
    - build
  - ▼ src
    - ▼ main
      - java
      - ▼ webapp
        - > META-INF
        - ▼ WEB-INF
          - ▼ lib
            - mysql-connector-java-8.0.22.jar
            - web.xml
            - delete.jsp
            - index.jsp
            - login.jsp
            - update.jsp

## JSP file Introduction :

1. login.jsp : The main function is to let admin login to the system and also a new registration for admin to have a new account to login system.
2. index.jsp : A homepage for course registration system, can done some function such as register a course for someone(student), edit the information that inserted wrongly and delete the person's information who cancelled the course registration.
3. update.jsp : An update form that can edit someone's information and update new information to database.
4. delete.jsp : Delete someone's information in the database.

## Deploy Web Application :

### 1. Create Dynamic Web Project

**New Dynamic Web Project**

**Dynamic Web Project**  
Create a standalone Java-based Web Application or add it to a new or existing Enterprise Application.

Project name:

Project location  
☒ Use default location  
Location:

Target runtime

Dynamic web module version

Configuration  
   
A good starting point for working with Apache Tomcat v9.0 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership  
☐ Add project to an EAR  
EAR project name:

Working sets  
☐ Add project to working sets   
Working sets:

### 2. Open the Tomcat v9.0 Server properties and setting them as below picture

**Overview**

**General Information**  
Specify the host name and other common settings.

Server name:

Host name:

Runtime Environment:

Configuration path:

[Open launch configuration](#)

**Server Locations**  
Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.

☒ Use workspace metadata (does not modify Tomcat installation)  
☐ Use Tomcat installation (takes control of Tomcat installation)  
☐ Use custom location (does not modify Tomcat installation)

Server path:

[Set deploy path to the default value \(currently set\)](#)

Deploy path:

**Publishing**

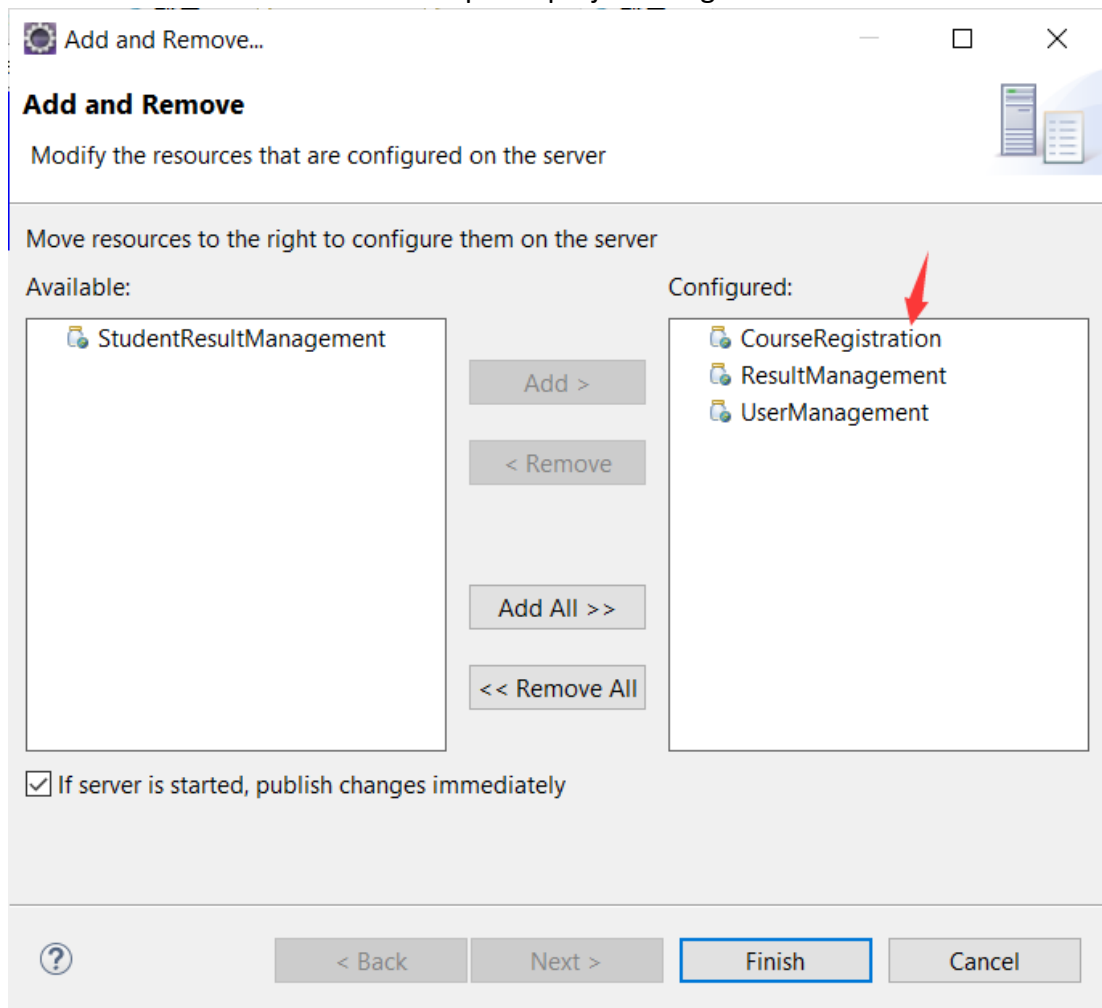
**Timeouts**

**Ports**  
Modify the server ports.

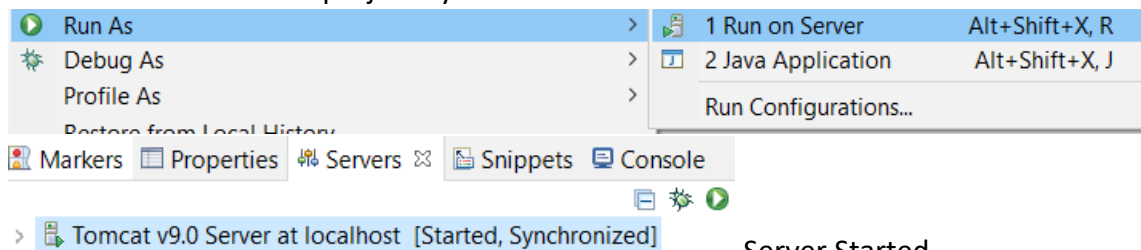
Port Name	Port Number
Tomcat admin port	8009
HTTP/1.1	8080

**MIME Mappings**

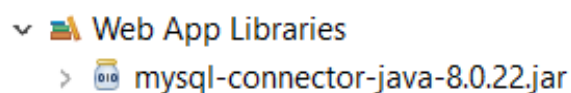
3. Add the Course Registration project with runtime environment(Tomcat v9.0 Server). Click Run on Server and add the required project to right hand side.



4. Run on Server : Run the project by Tomcat v9.0 Server at localhost



5. Import the mysql-connector.jar into library.



## Implementation of Web Application (Steps)

1. After running the server, it will go to this website (default website : login.jsp)  
<http://localhost:8080/CourseRegistration/> which is the login.jsp page.

课程报名系统

管理员登录

用户名:

密码:

登录

管理员注册

用户名:

密码:

注册

2. Register a new admin account.

管理员注册

用户名:

密码:

注册 ←

Registration complete will show an alert picture that says registration successful.

localhost:8080 显示

注册成功!!!

确定

3. Login admin account with userid : admin and password : admin.

管理员登录

用户名:

密码:

登录 ←

Login successful will also show an alert message

localhost:8080 显示

登录成功!!!

确定

## SP2021-LAB1.1&1.2-姚熙源-3190300677

If there is userid or password that not match to the data in database, then it will show an alert message. You need to relogin the system with correct userid and password.

localhost:8080 显示

用户名或密码输入错误，请重新输入!!!

确定

### 4. Homepage of Course Registration System

课程报名系统

登出

管理员 : admin

查看课程选择列表: 请选择课程 ▾

填写报名资料

姓名
<input type="text" value="请输入姓名"/>
课程
<input type="text" value="请输入课程名称"/>
报名费用 ¥
<input type="text" value="请输入报名费用"/>
<input type="button" value="报名"/> <input type="button" value="重置"/>

查看报名信息

姓名	课程	报名费用¥	操作
peter	Java	200	<a href="#">修改</a> <a href="#">删除</a>
tony	C++	500	<a href="#">修改</a> <a href="#">删除</a>

### 5. Check which course is opened for registration and fill it in the course textbox.

查看课程选择列表: 请选择课程 ▾

请选择课程

Java

C++

C

Python

Html

PHP

填写报名资料

姓名



6. Admin register course for someone(Gary)

## 填写报名资料

姓名

Gary

课程

Java

报名费用 ¥

300

报名

重置

Registration complete will also show an alert message.

localhost:8080 显示

数据已记录!!!

确定

7. Check the list whether the new course registration is successful or not.

## 查看报名信息

姓名	课程	报名费用¥	操作
peter	Java	200	<a href="#">修改</a> <a href="#">删除</a>
tony	C++	500	<a href="#">修改</a> <a href="#">删除</a>
Gary 	Java	300	<a href="#">修改</a> <a href="#">删除</a>

The registration is successful.

8. Edit the registration of Gary. Eg: edit his course to Python.

## 课程报名系统

[主页](#)

管理员 : admin

### 更新信息

姓名	<input type="text" value="Gary"/>
课程	<input type="text" value="Python"/>
报名费用	<input type="text" value="300"/>
<input type="button" value="更新"/> <input type="button" value="重置"/>	

Edit successful will also show an alert message.

localhost:8080 显示

数据已更新, 请返回主页!!!

确定

See the list of registration information.

Gary	Python	300	<a href="#">修改</a>	<a href="#">删除</a>
------	--------	-----	--------------------	--------------------

9. Delete someone's registration information. Eg: delete Gary's registration information. Click Delete.

Gary	Python	300	<a href="#">修改</a>	<a href="#">删除</a>
------	--------	-----	--------------------	--------------------

Delete successful will show an alert message.

localhost:8080 显示

信息已删除!!!

确定

Then goes to this webpage.

<http://localhost:8080/CourseRegistration/delete.jsp?id=20&userid=admin>

信息已删除，请返回主页！！

[主页](#)

Check the list of registration information.

查看报名信息

姓名	课程	报名费用¥	操作
peter	Java	200	<a href="#">修改</a> <a href="#">删除</a>
tony	C++	500	<a href="#">修改</a> <a href="#">删除</a>

The information's of Gary has been deleted successfully.

## 10. Logout function

课程报名系统

管理员 : admin

查看课程选择列表: [请选择课程](#)

填写报名资料

姓名

请输入姓名

课程

请输入课程名称

报名费用 ¥

请输入报名费用

报名

重置

查看报名信息

姓名	课程	报名费用¥	操作
peter	Java	200	<a href="#">修改</a> <a href="#">删除</a>
tony	C++	500	<a href="#">修改</a> <a href="#">删除</a>

[登出](#)

Logout successful will show an alert message.

localhost:8080 显示

登出成功!!!

确定

All of the above is the step to implement this web application and some function introduction.

### CSS requirement :

Get bootstrap in this website : <https://getbootstrap.com/> .You can download the bootstrap and put the css & js file in your project or just copy the css link from the webpage(as picture below)

```
<!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" |
```

and paste it in your jsp file such as picture below :

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOO0L7+AMvyTG2x"
crossorigin="anonymous">
```

CSS that used in this project :

Some example :

- class = "navbar navbar-expand-md navbar-dark"
- class="row"
- class = "col-sm-4"
- class = "card"
- class="card-body"
- class="form-label"
- class="form-control"
- class="btn btn-success"
- class="btn btn-warning"
- class = "col-sm-7"
- class="panel-body"
- class="table table-responsive table-bordered"
- class="fa fa-rmb"