Start Your Project

Environment setup and template project

Notes on Web DB Programming

Outline

- Notes on Web DB Programming
- Create an Amazon RDS instance – MySQL DB instance (check AWS_RDS.pdf under resources)
- Setting up MySQL workbench and connecting to MySQL DB instance (check Setting up MySQL workbench.pdf)
- Create an Amazon EC2 instance (check AWS_EC2.pdf)
- Install a Web Server (check AWS_EC2.pdf)
- Set up Environment and Introduction of the template project

Notes on Web DB Programming

Based on M.Muscari and UCSD (anon)

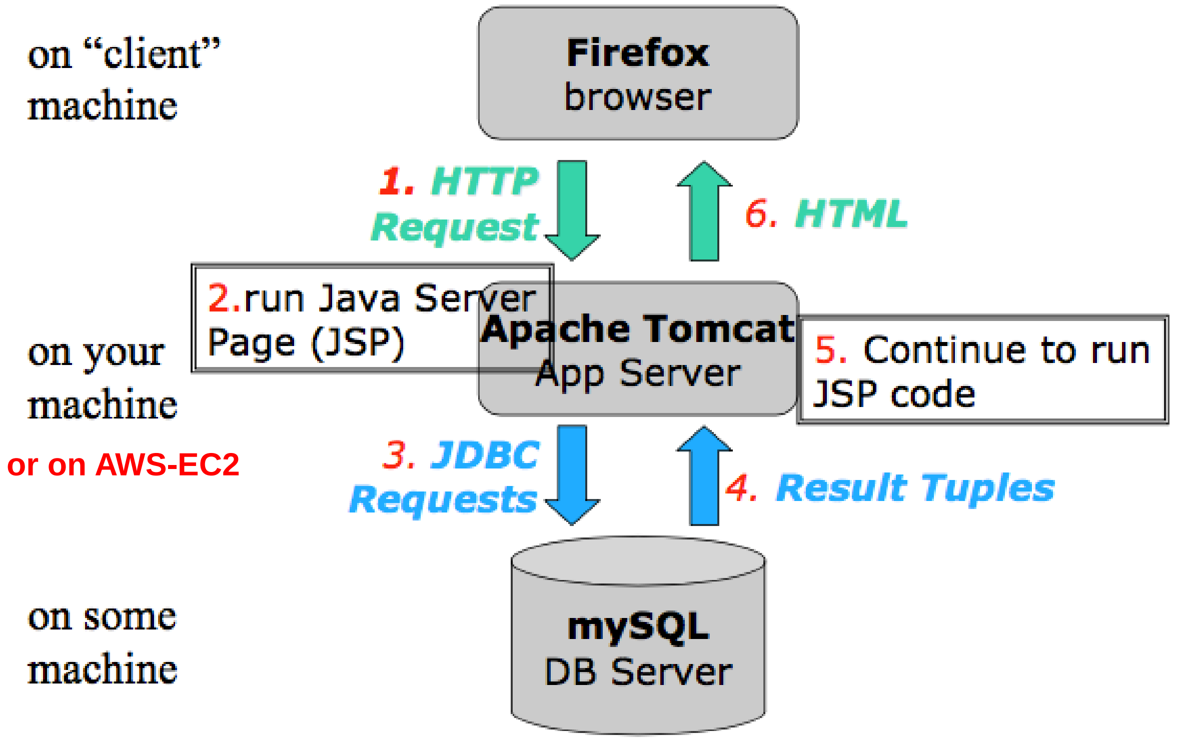
Needed tools and installation

- **JRE, IDE** (JAVA, Eclipse for EE developers)
- **MySQL** (it is your AWS RDS instance)
- **Apache Tomcat** (or any web server)
  - You will install it under the AWS EC2 instance AND locally in your computer for development purposes.
- **JDBC**

Three-Tier Application

- ➢**Presentation Tier** : user interface to make requests, provide input and see results
- ➢**Middle Tier**: application logic
- ➢**Data Management Tier:** database management

Three-Tier architecture

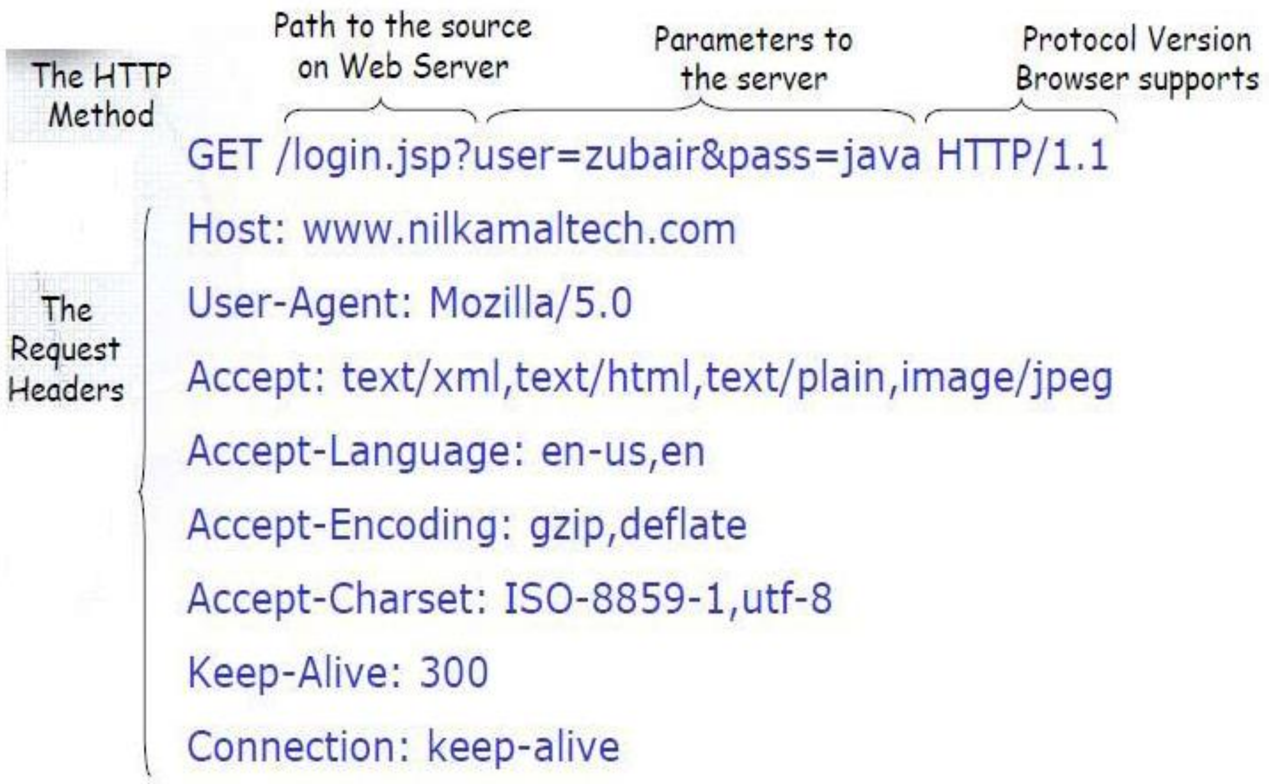


HTTP protocol

- ➢Protocol that allows web servers and clients to **exchange data** over the web.
- ➢It is a **request - response protocol**.
- ➢ Clients (web browsers) send requests to web servers
  - GET : ask for a resource
  - POST : send some data (e.g. HTML form)
- ➢Server sends response
  - Status code (200 OK, 404 Not Found!)
- ➢HTTP is a "***stateless***" protocol; each time a client retrieves a Webpage, the client opens a separate connection to the Web server and the server automatically does not keep any record of previous client request.

Difference between GET/POST requests

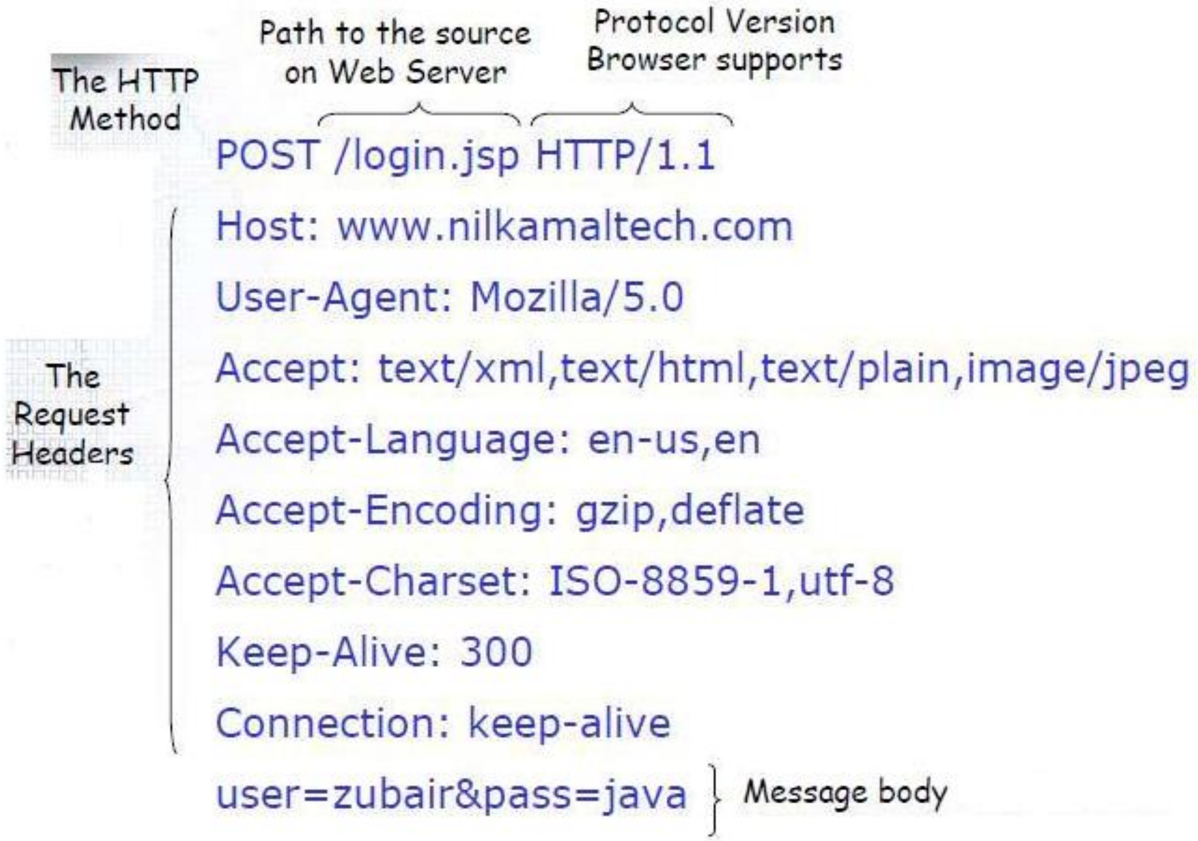
Anatomy of GET request



The HTTP Method
Path to the source on Web Server
Parameters to the server
Protocol Version Browser supports
The Request Headers

```
GET /login.jsp?user=zubair&pass=java HTTP/1.1
Host: www.nilkamaltech.com
User-Agent: Mozilla/5.0
Accept: text/xml,text/html,text/plain,image/jpeg
Accept-Language: en-us,en
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8
Keep-Alive: 300
Connection: keep-alive
```

Anatomy of POST request



The HTTP Method
Path to the source on Web Server
Protocol Version Browser supports
The Request Headers
Message body

```
POST /login.jsp HTTP/1.1
Host: www.nilkamaltech.com
User-Agent: Mozilla/5.0
Accept: text/xml,text/html,text/plain,image/jpeg
Accept-Language: en-us,en
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8
Keep-Alive: 300
Connection: keep-alive
user=zubair&pass=java
```

HTML (Hyper Text Markup Language)

- ➢Standard markup language for creating web pages
  - ➢Language for creating **structured documents**
    - ▪ It consists of elements which can be nested
  - ➢The HTML standard specifies a number of universally supported elements (“tags”)
- ➢Web browsers receive HTML documents from a webserver and render them into multimedia web pages.
- ➢HTML is commonly delivered as part of an HTTP response

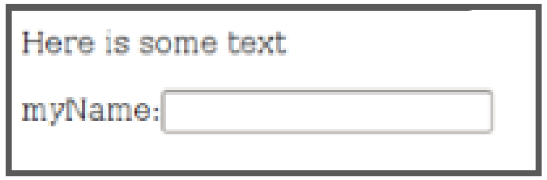
Tutorial: https://www.w3schools.com/html/

HTML Tags

- Common HTML tags include:
  - **<div></div>** - a logical division (section)
  - **<p> </p>** - a paragraph
  - **<table> </table>** -a table of values
    - **<tr> </tr>** - table row
    - **<td> </td>** -table column
  - **<form></form>** - a form enclosing input fields
    - **<input></input>** - an input field

HTML Example

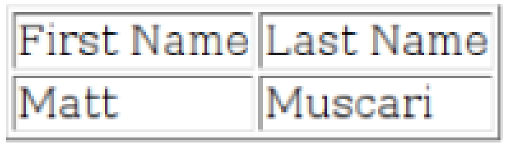
```
<html>
   <body>
       <div>
          <p>Here is some text</p>
          <form action=“submit.jsp” method=“post”>
             myName: <input name=“myInput” type=“text”/>
          </form>
       </div>
   </body>
</html>
```


Here is some text

myName:

HTML Table Example

```
<html>
   <body>
       <table border='1'>
          <tr>
              <td>First Name</td>
              <td>Last Name</td>
          </tr>
          <tr>
              <td>Matt</td>
              <td>Muscari</td>
          </tr>
       </table>
   </body>
</html>
```



First Name	Last Name
Matt	Muscari

JSP (Java Server Pages)

- A technology for building web applications that serve **dynamic content**
- A **JSP page** is a text document that contains two types of text:
  - **static data**, which can be expressed in any text-based format (e.g. HTML)
  - JSP elements, which construct **dynamic content**.
- The dynamic content in a **JSP page** is in specially marked Java code fragments (enclosed between <% and %>).
- To deploy and run JSPs, a compatible web server with a servlet container, such as Apache Tomcat is required.
- When executed, the Java code fragments usually generate additional HTML into the page (in our case either accessing the database or processing parameters passed to HTTP requests)
- At the end, the resulting HTML page is sent to the browser to be displayed.

JSP Syntax

- **Comment**

  <%-- Comment--%>

- **Expression**

  <%= Java expression %>

  Expression tag evaluates the **expression** placed in it, converts the result into String and send the result back to the client through response object.

  e.g. <p>Today is <%= new Date().toString(); %> </p>

- **Scriplet**

  <% java code fragment%>

  e.g. <% person.getFirstName();%>

- **Include**

  <jsp:include page="relativeURL"/>

JSP Implicit Objects

Object	Class
request	**HttpServletRequest**
response	**HttpServletResponse**
session	**HttpSession**
out	**Writer**

request

- <%@ page language="java" contentType="text/html"%>

```
<html>
        <head>
                <title>RequestExamplePage</title>
```

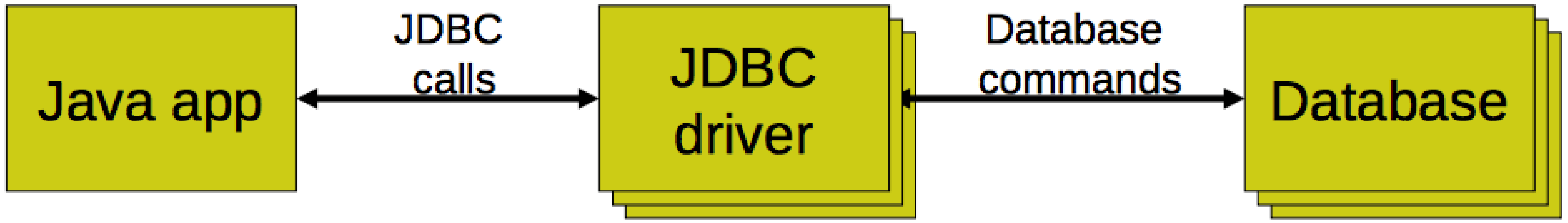
-
  ```
        </head>
         <body>
                <%
                // Get the User's Name from the request
                out.println("<b>Hello: " + request.getParameter("myInput") + "</b>");
                %>
        </body>
</html>
```

session

-
  ```
  <%@ page language="java" contentType="text/html"%>
  <html>
          <head>
                  <title>SessionExamplePage</title>

          </head>
           <body>
               <%
               HttpSession session = request.getSession(); //create a session object
               // Try and get the current count from the session
               Integer count = (Integer)session.getAttribute("COUNT");
               // If COUNT is not found, create it and add it to the session
               if ( count == null ) {
                   count = new Integer(1);
                   session.setAttribute("COUNT", count);
               } else {
                   count = new Integer(count.intValue() + 1);
                   session.setAttribute("COUNT", count);
               }
               // Print the number of times the user has visited the site
               out.println("<b>Hello you have visited this site: " + count + " times. </b>");
               %>
          </body> </html>
  ```

Java Database Connectivity (JDBC)

- An interface to communicate with a relational database
  - Allows database agnostic Java code
  - Treat database tables/rows/columns as Java objects
- JDBC driver
  - An implementation of the JDBC interface
  - Communicates with a particular database



JDBC steps

1. Connect to database
2. Query database (or insert/update/delete)
3. Process results
4. Close connection to database

1. Connect to database

- Load JDBC driver
  - Class.forName("com.mysql.jdbc.Driver").newInstance();
  - Make connection
    - Connection conn = DriverManager.getConnection(**url**);
- URL
  - Format: “jdbc:mysql//<*hostname*>:<port>/<*databaseName*>”
  - jdbc:mysql://cs336.ckksjtjg2jto.us-east-2.rds.amazonaws.com:3036/BarBeerDrinkerSample

2. Query database

- Create statement
  - Statement stmt = conn.createStatement();
    - stmt object sends SQL commands to database
  - Methods
    - executeQuery() for SELECT statements
    - executeUpdate() for INSERT, UPDATE, DELETE, statements
- Send SQL statements
  - stmt.executeQuery(“SELECT ...”);
  - stmt.executeUpdate(“INSERT ...”);

2. Query database

- ➢ Prepared Statements
  - If you want to execute dynamic or parameterized SQL queries, use a “PreparedStatement” object instead of a statement.

```
PreparedStatement updateStud=conn.prepareStatement("UPDATE Student SET
fname=? WHERE lastname LIKE?");

updateStud.setString(1,"John");
updateStud.setString(2,"Doe");
updateStud.executeUpdate();
```

3. Process results

- Result of a SELECT statement (rows/columns) returned as a ResultSet object
  - ResultSet rs = stmt.executeQuery("SELECT drinker,beer from LIKES");
- Step through each row in the result
  - rs.next()
- Get column values in a row
  - String userid = rs.getString(“drinker”);
  - int type = rs.getInt(“type”);

3. Process results

➢Add a row to the users table

- String str = "INSERT INTO LIKES VALUES('Bob', ‘Corona')”;

➢Returns number of rows in table

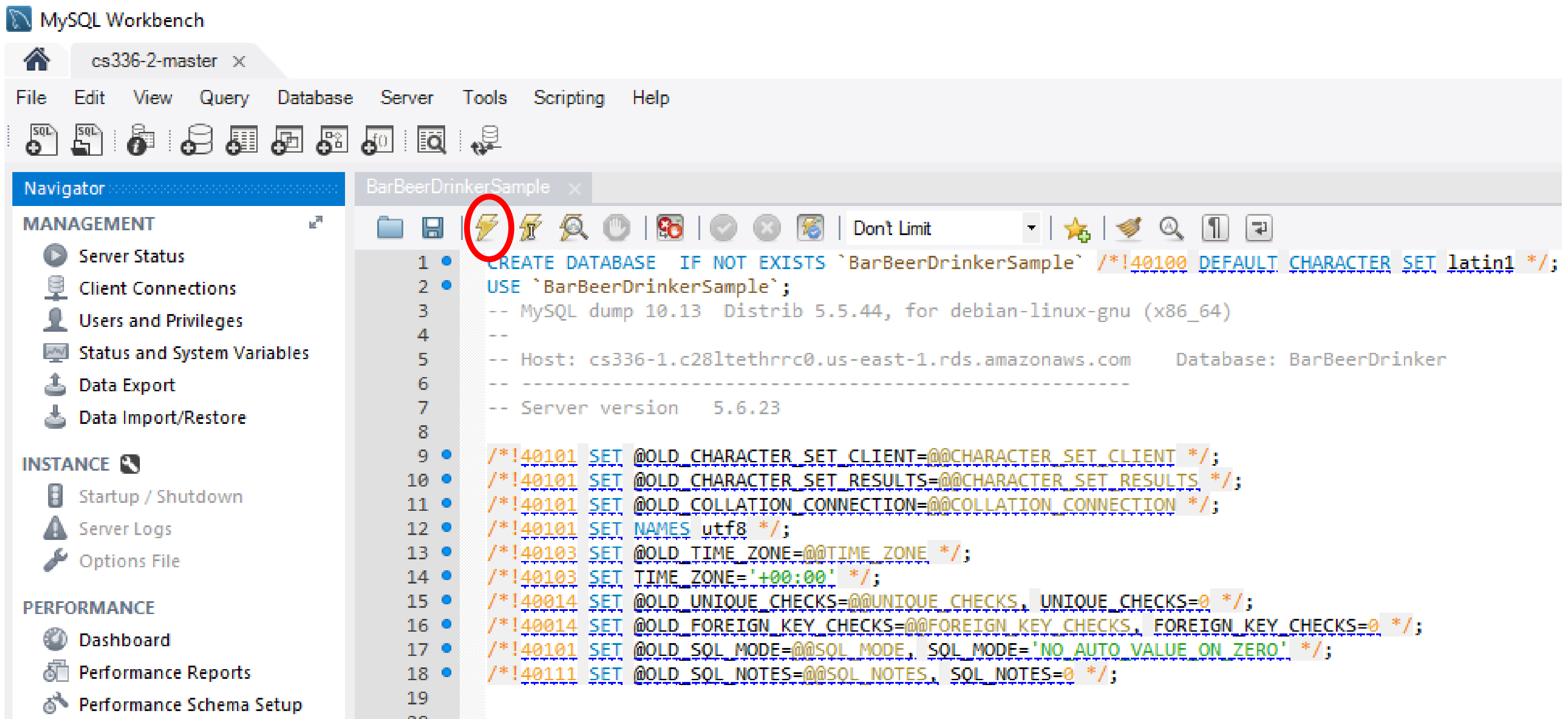
- int rows = stmt.executeUpdate(str);

4. Close connection to database

- Close the ResultSet object
  - rs.close();
- Close the Statement object
  - stmt.close();
- Close the connection
  - conn.close();

The Template Project

0. import schema **BarBeerDrinkerSample** in your created DB instance using the provided script ”BarBeerDrinkerSample.sql”. Open the script and run it in your MySqlWorkbench. (File->Open SQL script)

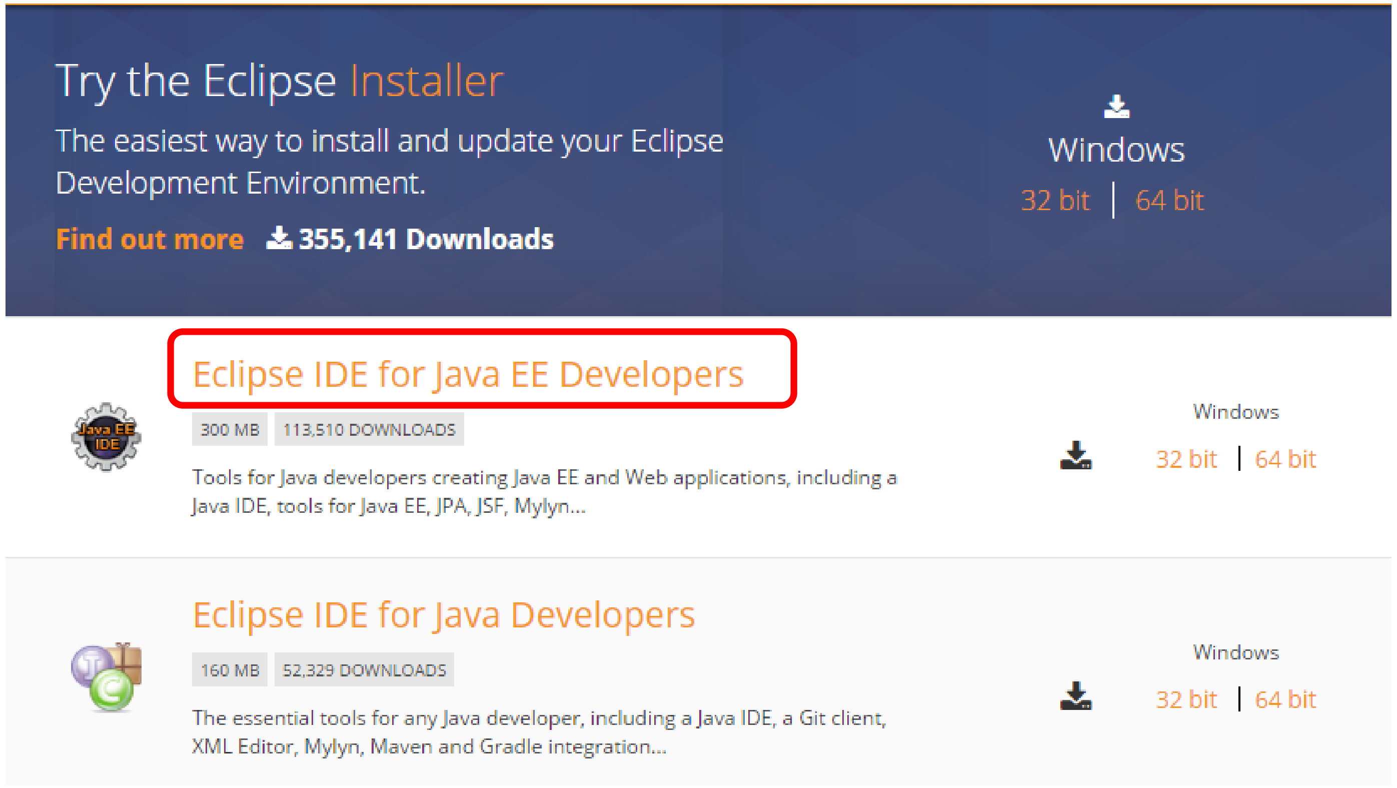


```
CREATE DATABASE  IF NOT EXISTS `BarBeerDrinkerSample` /*!40100 DEFAULT CHARACTER SET latin1 */;
USE `BarBeerDrinkerSample`;
-- MySQL dump 10.13  Distrib 5.5.44, for debian-linux-gnu (x86_64)
--
-- Host: cs336-1.c28ltethrrc0.us-east-1.rds.amazonaws.com    Database: BarBeerDrinker
-- ------------------------------------------------------
-- Server version	5.6.23

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

The Template Project

1. Download Eclipse IDE for **Java EE** Developers

https://eclipse.org/downloads/eclipse-packages/



Try the Eclipse Installer

The easiest way to install and update your Eclipse Development Environment.

Find out more 355,141 Downloads

Windows
32 bit | 64 bit

Eclipse IDE for Java EE Developers

300 MB 113,510 DOWNLOADS

Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn...

Windows
32 bit | 64 bit

Eclipse IDE for Java Developers

160 MB 52,329 DOWNLOADS

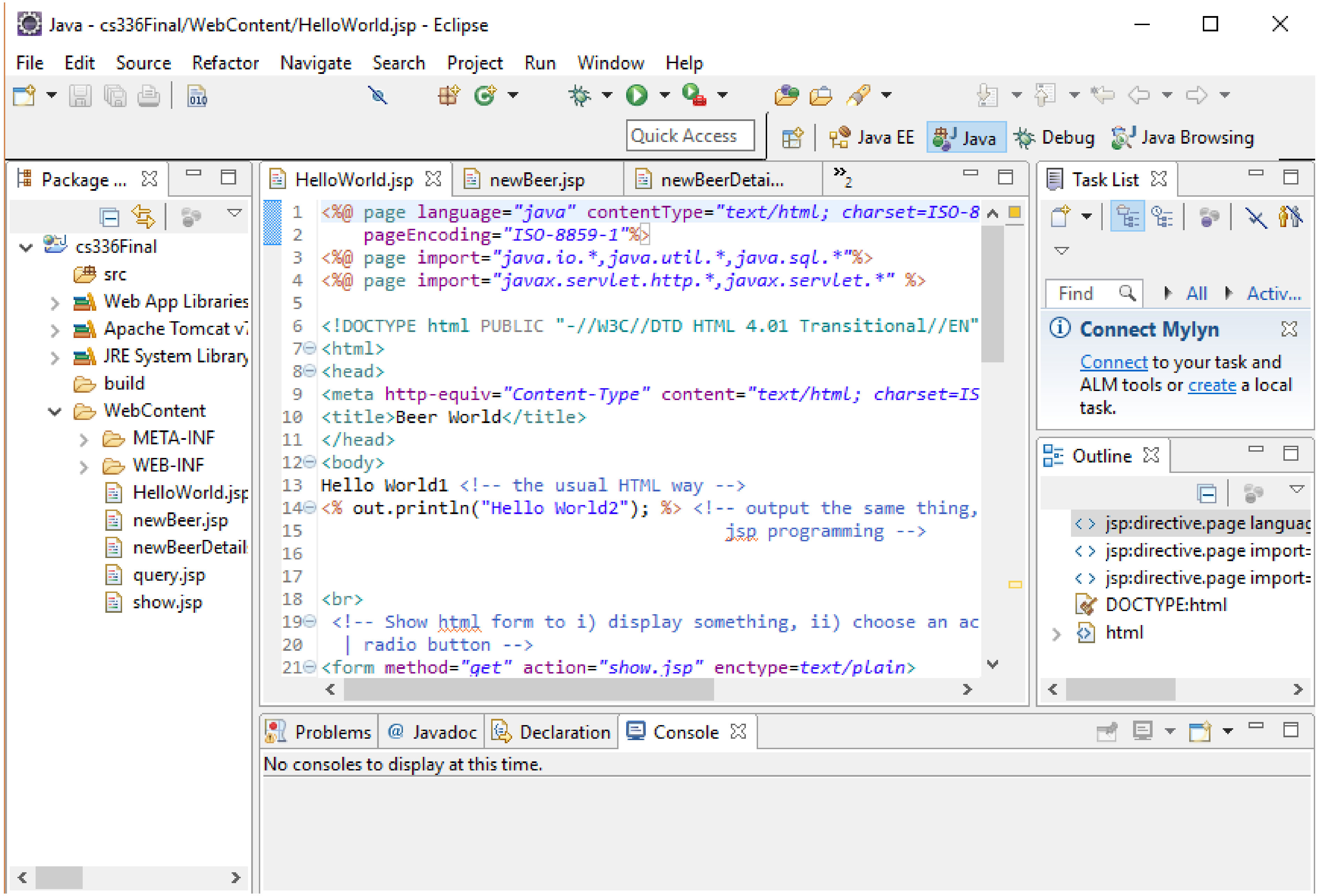
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration...

Windows
32 bit | 64 bit

The Template Project

2. Open eclipse and import the template project (cs336Final.war)

File – Import – Web – WAR file



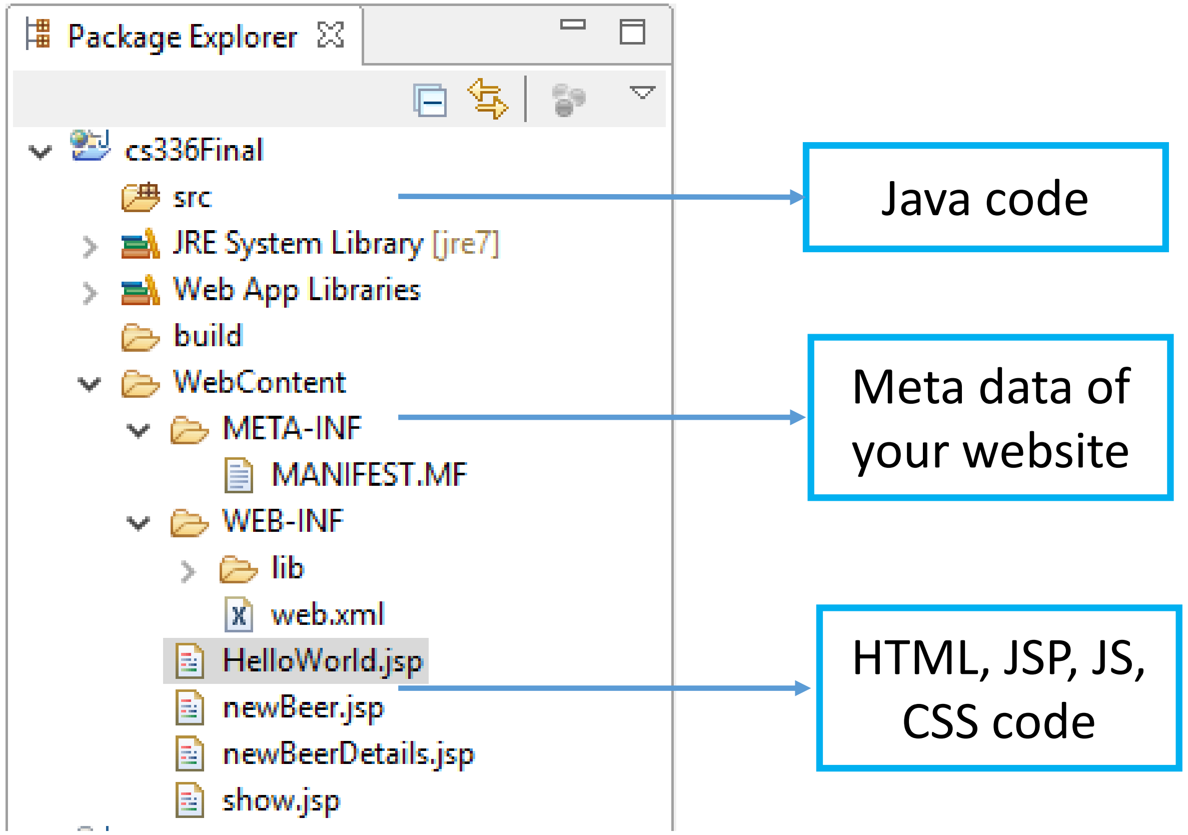
Java - cs336Final/WebContent/HelloWorld.jsp - Eclipse

```
<%@ page language="java" contentType="text/html; charset=ISO-8
    pageEncoding="ISO-8859-1"%>
<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%@ page import="javax.servlet.http.*,javax.servlet.*" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=IS
<title>Beer World</title>
</head>
<body>
Hello World1 <!-- the usual HTML way -->
<% out.println("Hello World2"); %> <!-- output the same thing,
                                    jsp programming -->


<br>
 <!-- Show html form to i) display something, ii) choose an ac
   | radio button -->
<form method="get" action="show.jsp" enctype=text/plain>
```

No consoles to display at this time.

The Template Project

3. Structure of the template



Package Explorer

cs336Final
- src
- JRE System Library [jre7]
- Web App Libraries
- build
- WebContent
  - META-INF
    - MANIFEST.MF
  - WEB-INF
    - lib
    - web.xml
  - HelloWorld.jsp
  - newBeer.jsp
  - newBeerDetails.jsp
  - show.jsp

The Template Project

4. Set your Tomcat server in eclipse

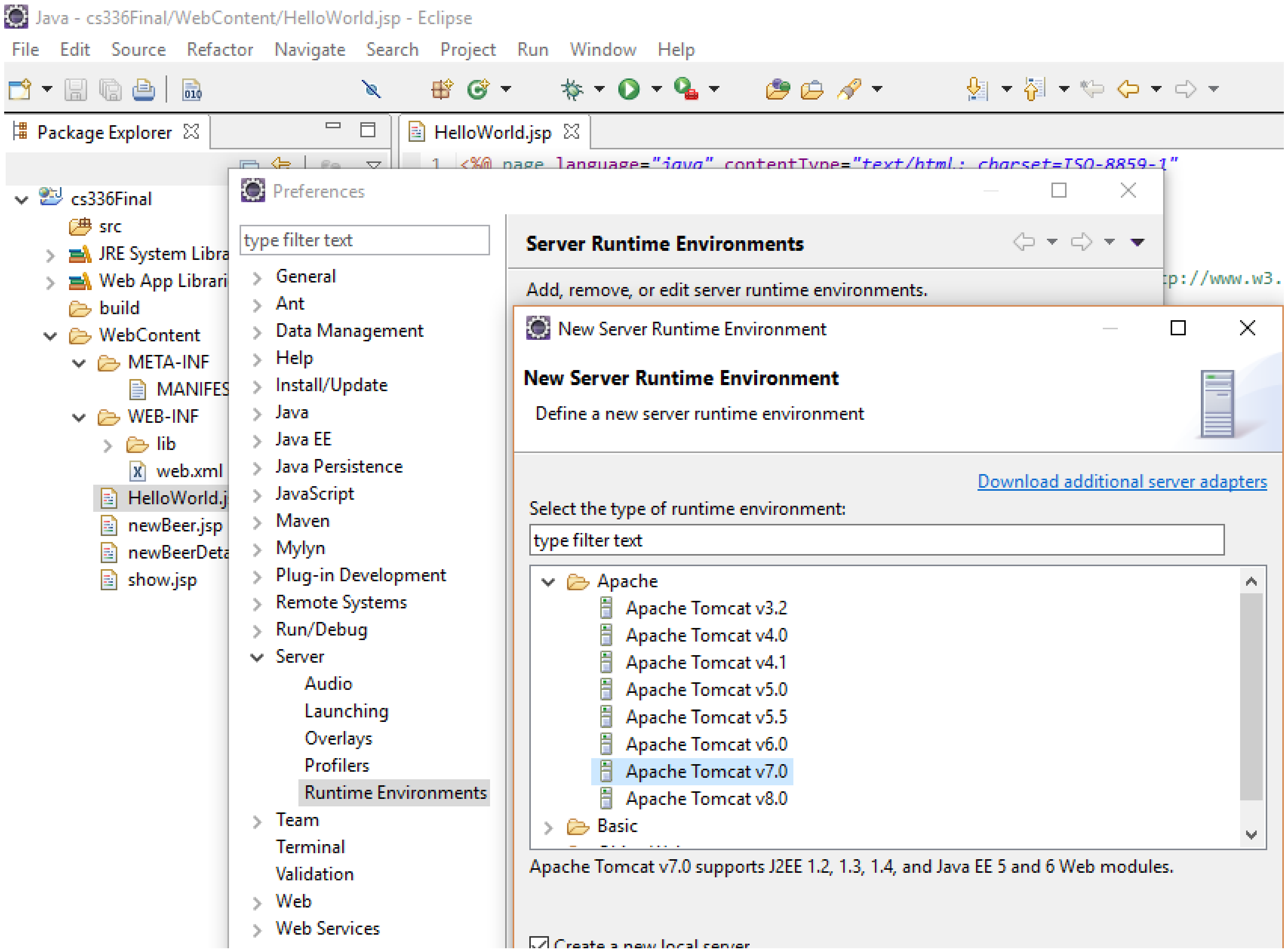
- If you don’t have tomcat yet go to: https://tomcat.apache.org/download-70.cgi

and download the binary distribution for your OS.

- After go back to eclipse:

Windows - Preference - Server - Runtime Environment - Add - Apache Tomcat v7.0 **or**

Eclipse- Preferences - Server - Runtime Environments - Add - Apache Tomcat v7.0



Java - cs336Final/WebContent/HelloWorld.jsp - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- cs336Final
  - src
  - JRE System Libra
  - Web App Librari
  - build
  - WebContent
    - META-INF
      - MANIFES
    - WEB-INF
      - lib
      - web.xml
    - HelloWorld.j
    - newBeer.jsp
    - newBeerDeta
    - show.jsp

HelloWorld.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

Preferences

type filter text

General, Ant, Data Management, Help, Install/Update, Java, Java EE, Java Persistence, JavaScript, Maven, Mylyn, Plug-in Development, Remote Systems, Run/Debug, Server (Audio, Launching, Overlays, Profilers, Runtime Environments), Team, Terminal, Validation, Web, Web Services

Server Runtime Environments

Add, remove, or edit server runtime environments.

New Server Runtime Environment

New Server Runtime Environment

Define a new server runtime environment

Download additional server adapters

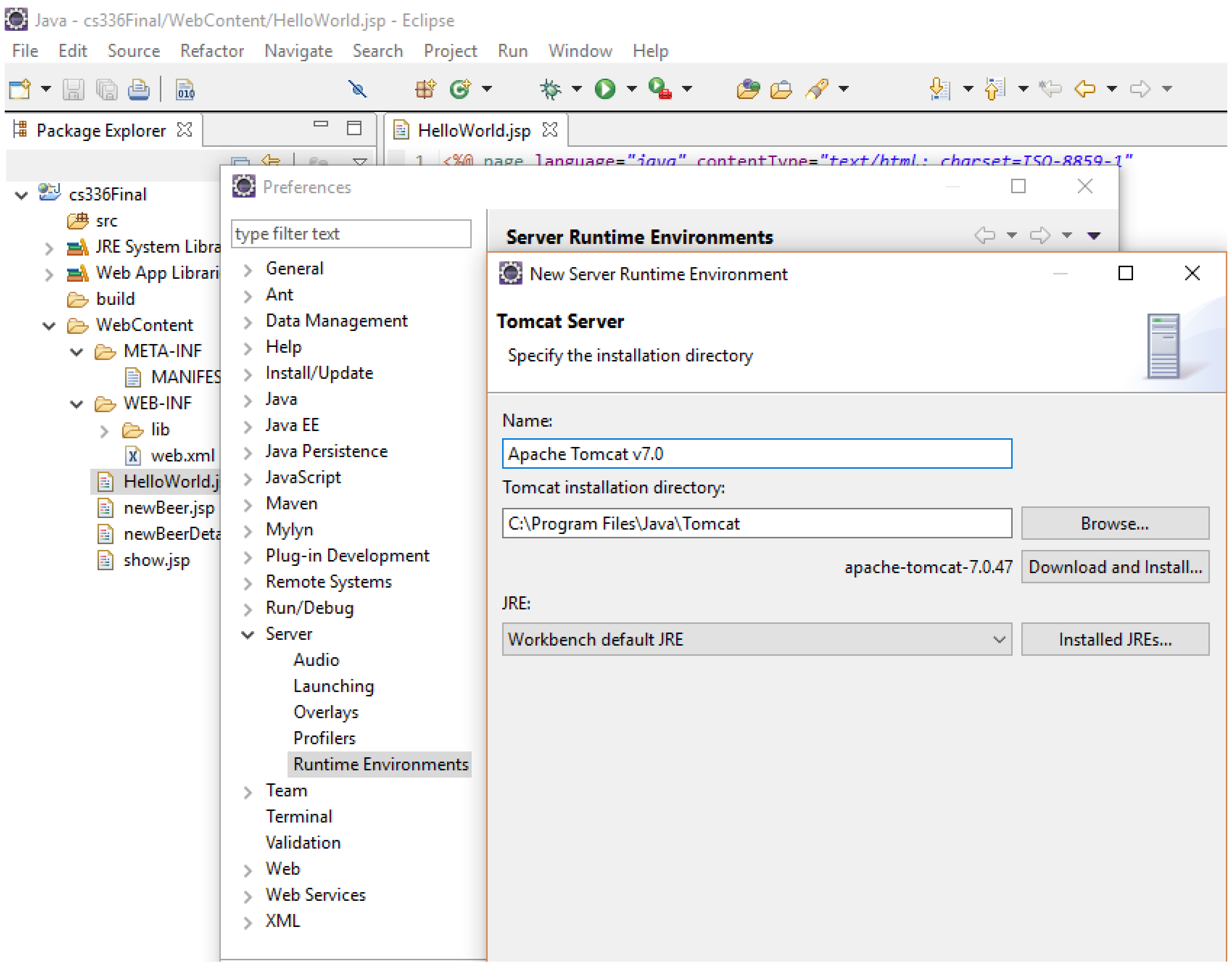
Select the type of runtime environment:

type filter text

- Apache
  - Apache Tomcat v3.2
  - Apache Tomcat v4.0
  - Apache Tomcat v4.1
  - Apache Tomcat v5.0
  - Apache Tomcat v5.5
  - Apache Tomcat v6.0
  - Apache Tomcat v7.0
  - Apache Tomcat v8.0
- Basic

Apache Tomcat v7.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5 and 6 Web modules.

☑ Create a new local server



Java - cs336Final/WebContent/HelloWorld.jsp - Eclipse

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

Tomcat Server

Specify the installation directory

Name: Apache Tomcat v7.0

Tomcat installation directory: C:\Program Files\Java\Tomcat

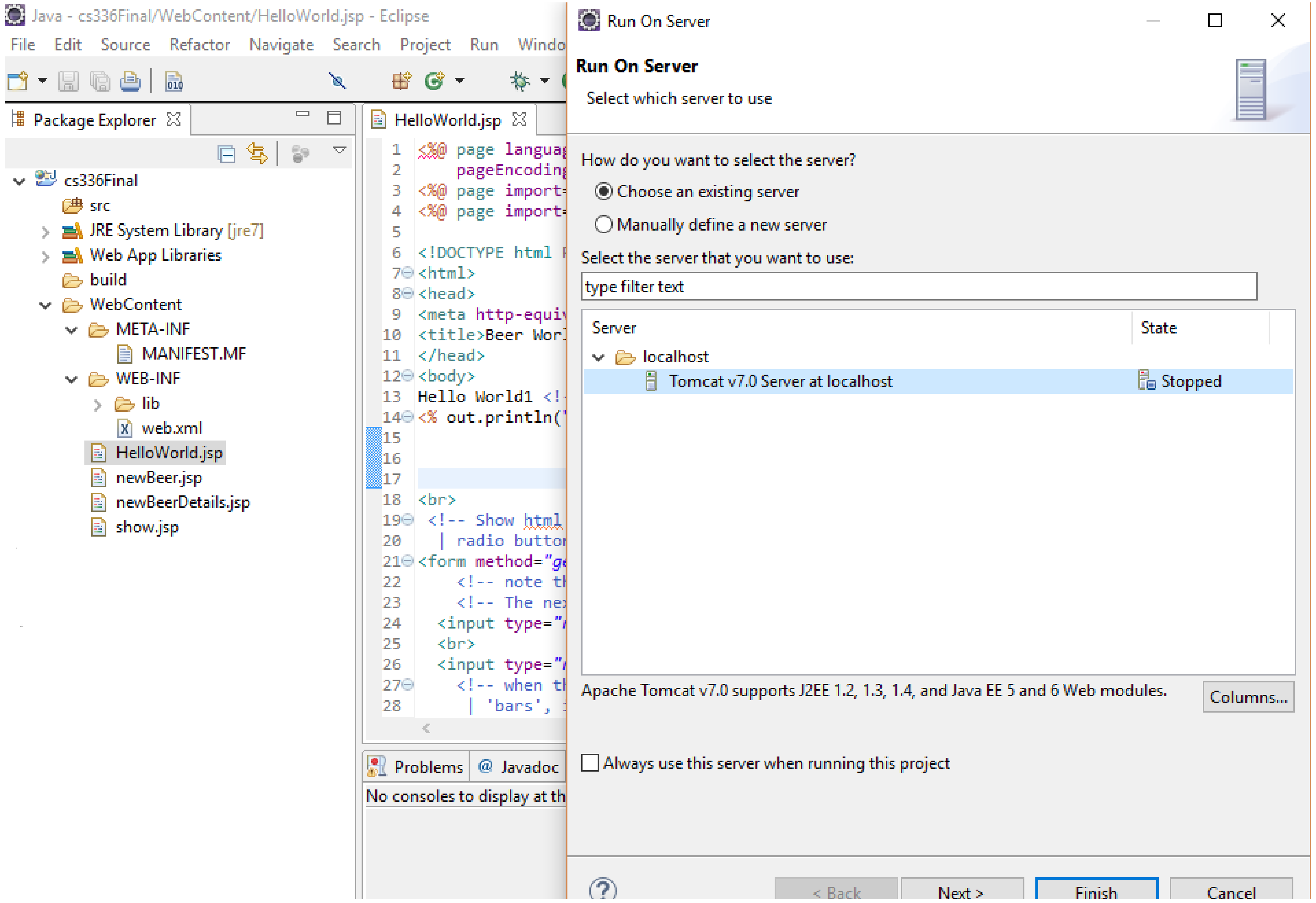
apache-tomcat-7.0.47

JRE: Workbench default JRE

The Template Project

5. Run the project based on Tomcat 7

Right click on the project - Run as - Run on Server - Apache - Tomcat7



```
cs336Final
  src
  JRE System Library [jre7]
  Web App Libraries
  build
  WebContent
    META-INF
      MANIFEST.MF
    WEB-INF
      lib
      web.xml
    HelloWorld.jsp
    newBeer.jsp
    newBeerDetails.jsp
    show.jsp
```

```
<%@ page languag
    pageEncoding
<%@ page import=
<%@ page import=

<!DOCTYPE html P
<html>
<head>
<meta http-equiv
<title>Beer Worl
</head>
<body>
Hello World1 <!-
<% out.println("



<br>
 <!-- Show html
  | radio button
<form method="ge
    <!-- note th
    <!-- The nex
  <input type="r
  <br>
  <input type="r
    <!-- when th
     | 'bars', i
```

No consoles to display at th

Run On Server

Select which server to use

How do you want to select the server?

(●) Choose an existing server

( ) Manually define a new server

Select the server that you want to use:

type filter text

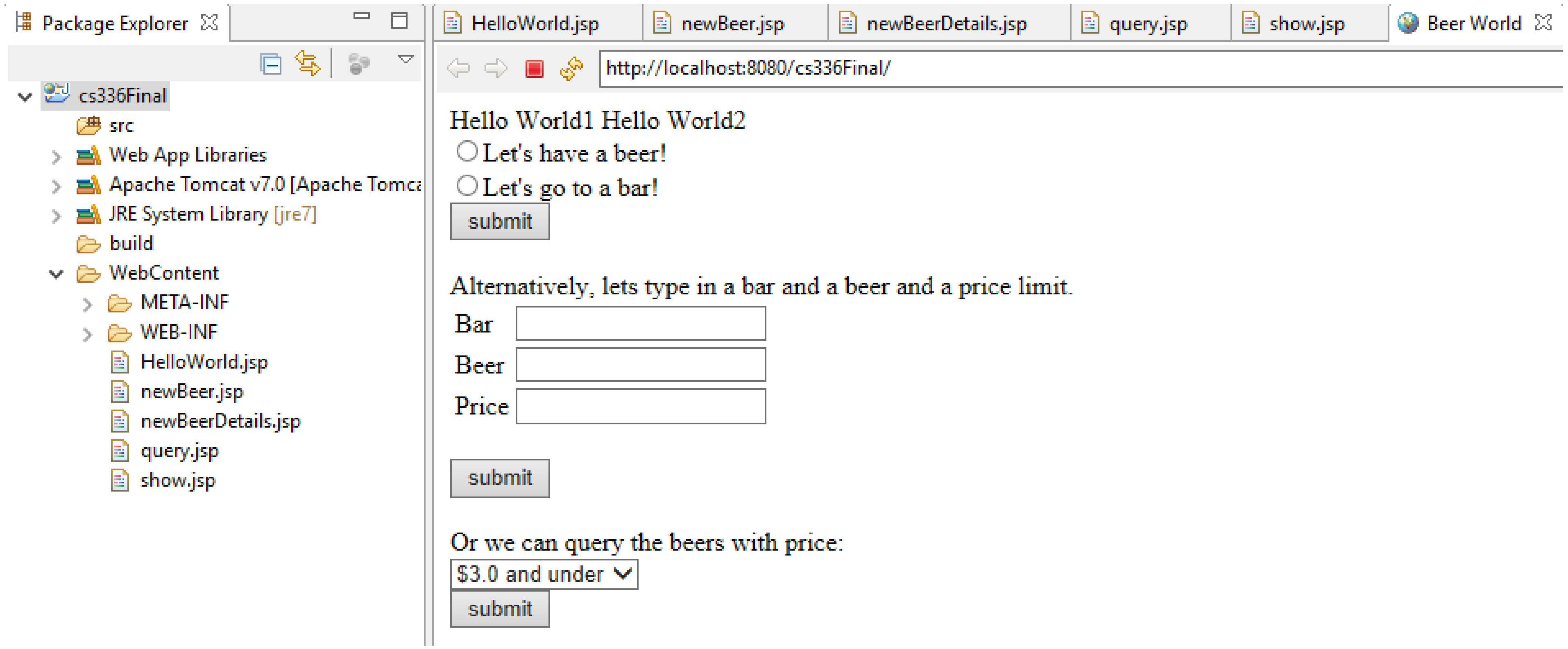
Server	State
localhost	
Tomcat v7.0 Server at localhost	Stopped

Apache Tomcat v7.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5 and 6 Web modules.

☐ Always use this server when running this project

The Template Project

- Now you can see your project home page, index.html page.



http://localhost:8080/cs336Final/

Hello World1 Hello World2

○ Let's have a beer!

○ Let's go to a bar!

submit

Alternatively, lets type in a bar and a beer and a price limit.

Bar

Beer

Price

submit

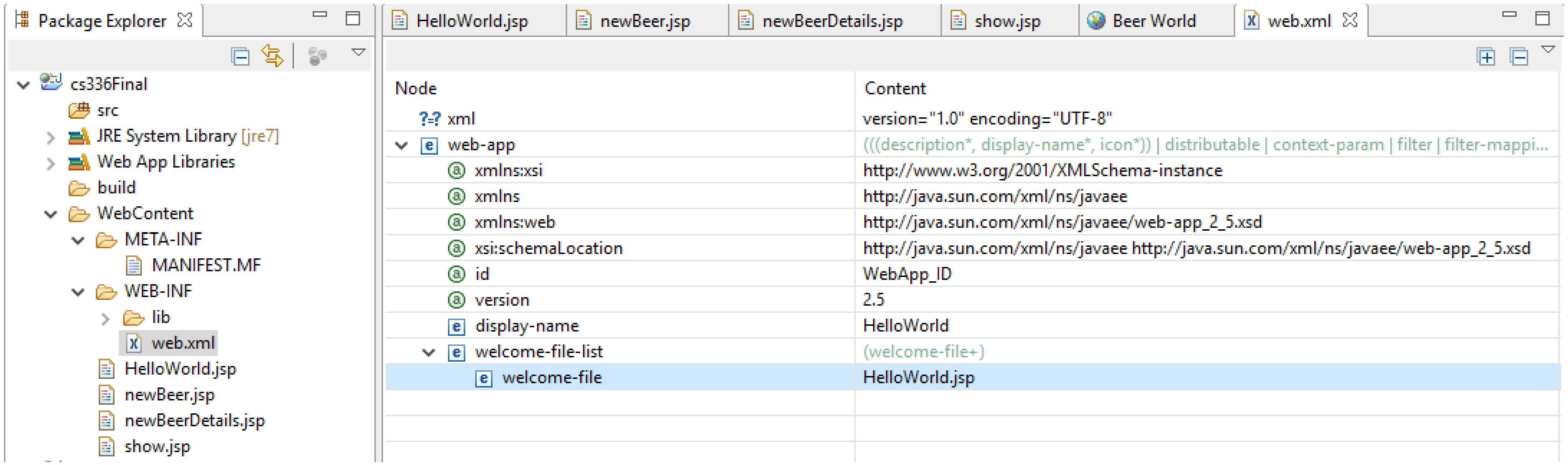
Or we can query the beers with price:

$3.0 and under

submit

The Template Project

6. The home page is set in web.xml, you can set your own page if you want.



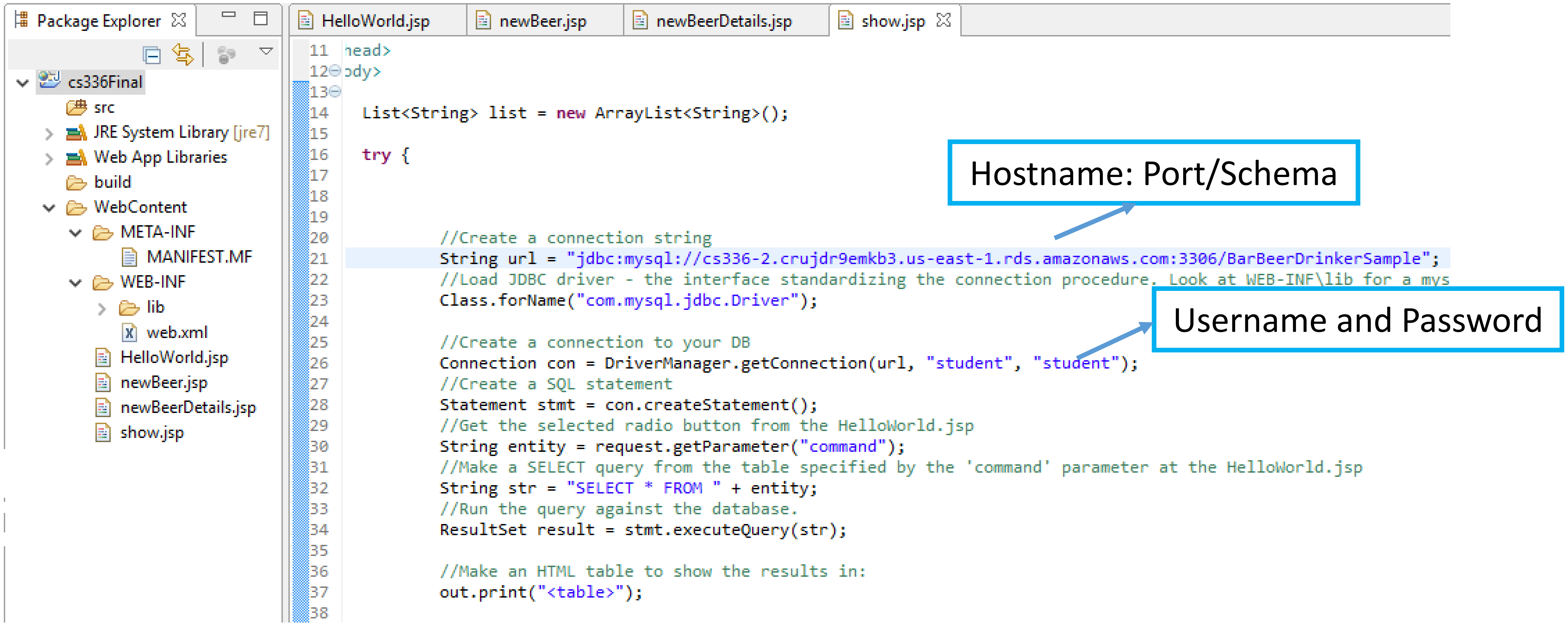
Node	Content
xml	version="1.0" encoding="UTF-8"
web-app	(((description*, display-name*, icon*)) | distributable | context-param | filter | filter-mappi...
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xmlns	http://java.sun.com/xml/ns/javaee
xmlns:web	http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd
xsi:schemaLocation	http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd
id	WebApp_ID
version	2.5
display-name	HelloWorld
welcome-file-list	(welcome-file+)
welcome-file	HelloWorld.jsp

The Template Project

7. Connect to your own db instance in Project

- In order to interact with db instance (add, delete, update, select), you need to set your own database address in the project.
- At the same time, the database username and password are both essential.
- Replace the database information with your own database information as follows.

The Template Project

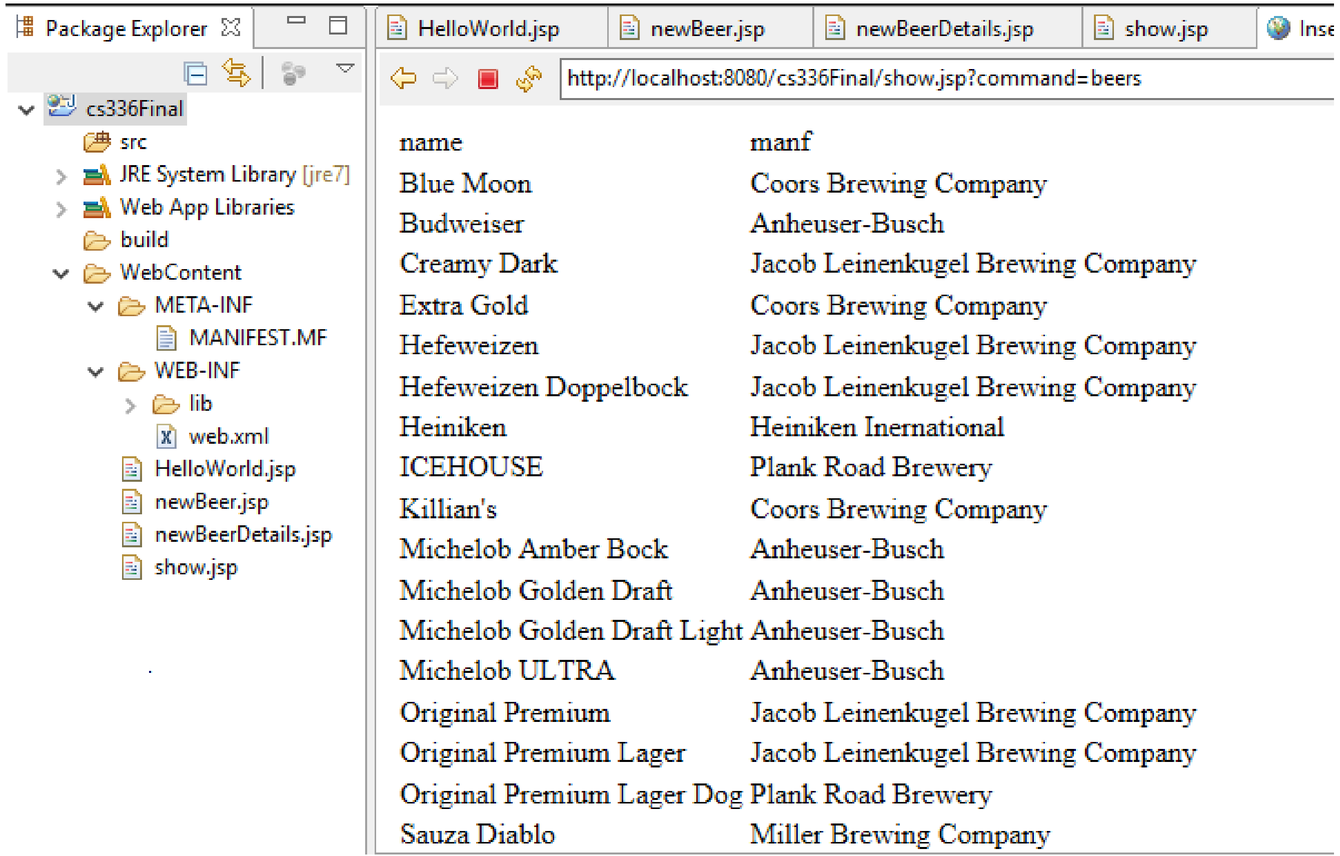


```
ıead>
ody>

  List<String> list = new ArrayList<String>();

  try {



        //Create a connection string
        String url = "jdbc:mysql://cs336-2.crujdr9emkb3.us-east-1.rds.amazonaws.com:3306/BarBeerDrinkerSample";
        //Load JDBC driver - the interface standardizing the connection procedure. Look at WEB-INF\lib for a mys
        Class.forName("com.mysql.jdbc.Driver");

        //Create a connection to your DB
        Connection con = DriverManager.getConnection(url, "student", "student");
        //Create a SQL statement
        Statement stmt = con.createStatement();
        //Get the selected radio button from the HelloWorld.jsp
        String entity = request.getParameter("command");
        //Make a SELECT query from the table specified by the 'command' parameter at the HelloWorld.jsp
        String str = "SELECT * FROM " + entity;
        //Run the query against the database.
        ResultSet result = stmt.executeQuery(str);

        //Make an HTML table to show the results in:
        out.print("<table>");
```

The Template Project

8. Let’s have a beer

Select the radio button and then click submit below it



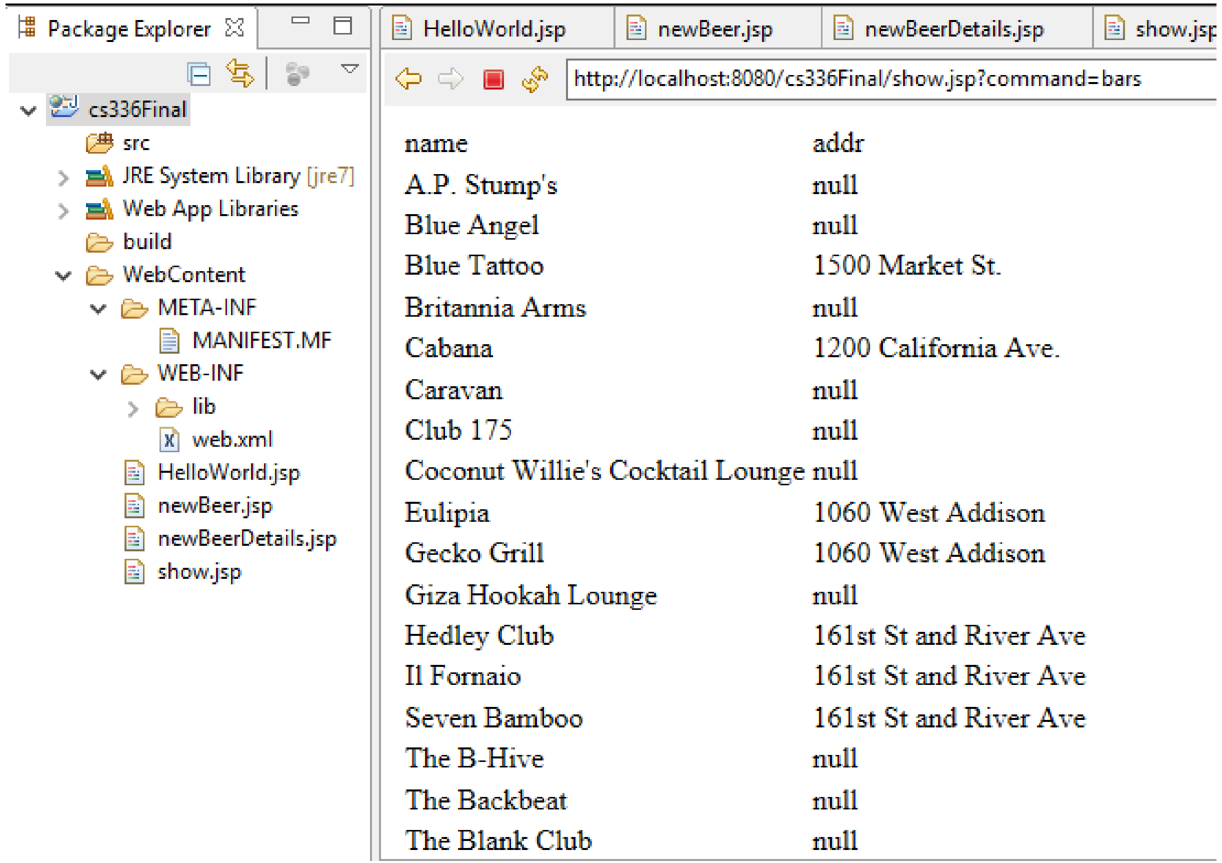
http://localhost:8080/cs336Final/show.jsp?command=beers

name	manf
Blue Moon	Coors Brewing Company
Budweiser	Anheuser-Busch
Creamy Dark	Jacob Leinenkugel Brewing Company
Extra Gold	Coors Brewing Company
Hefeweizen	Jacob Leinenkugel Brewing Company
Hefeweizen Doppelbock	Jacob Leinenkugel Brewing Company
Heiniken	Heiniken Inernational
ICEHOUSE	Plank Road Brewery
Killian's	Coors Brewing Company
Michelob Amber Bock	Anheuser-Busch
Michelob Golden Draft	Anheuser-Busch
Michelob Golden Draft Light	Anheuser-Busch
Michelob ULTRA	Anheuser-Busch
Original Premium	Jacob Leinenkugel Brewing Company
Original Premium Lager	Jacob Leinenkugel Brewing Company
Original Premium Lager Dog	Plank Road Brewery
Sauza Diablo	Miller Brewing Company

The Template Project

9. Let’s go to a pub

Select the radio button and then click submit below it



http://localhost:8080/cs336Final/show.jsp?command=bars

name	addr
A.P. Stump's	null
Blue Angel	null
Blue Tattoo	1500 Market St.
Britannia Arms	null
Cabana	1200 California Ave.
Caravan	null
Club 175	null
Coconut Willie's Cocktail Lounge	null
Eulipia	1060 West Addison
Gecko Grill	1060 West Addison
Giza Hookah Lounge	null
Hedley Club	161st St and River Ave
Il Fornaio	161st St and River Ave
Seven Bamboo	161st St and River Ave
The B-Hive	null
The Backbeat	null
The Blank Club	null

The Template Project

10. Insert a tuple into sells table

Input pub name, beer name and cost, then click submit.

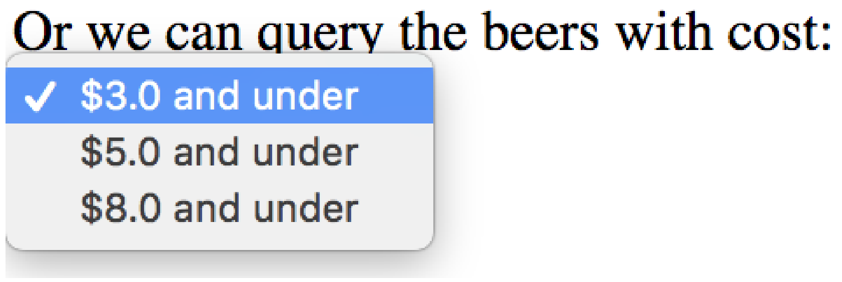
You can find a new record inserted into your database after submitting this form.

-NOTE: since you insert a tuple in sells table which has FKs in the bar and beer table, make sure the beer and bar you insert already exist in these two tables.

The Template Project

11. Query the beers with cost

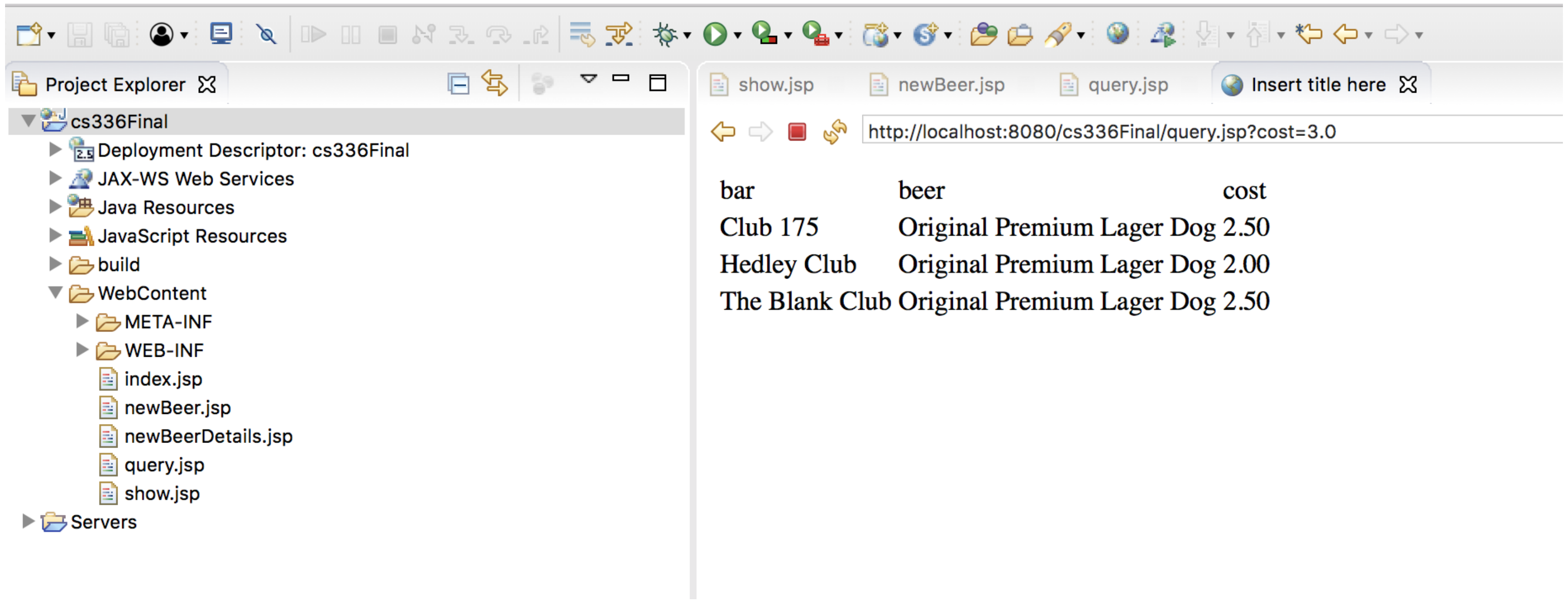
Choose one option from the dropdown menu, then click submit.


Or we can query the beers with cost:

✓ $3.0 and under
$5.0 and under
$8.0 and under

The Template Project

- Query the beers with cost <= 3



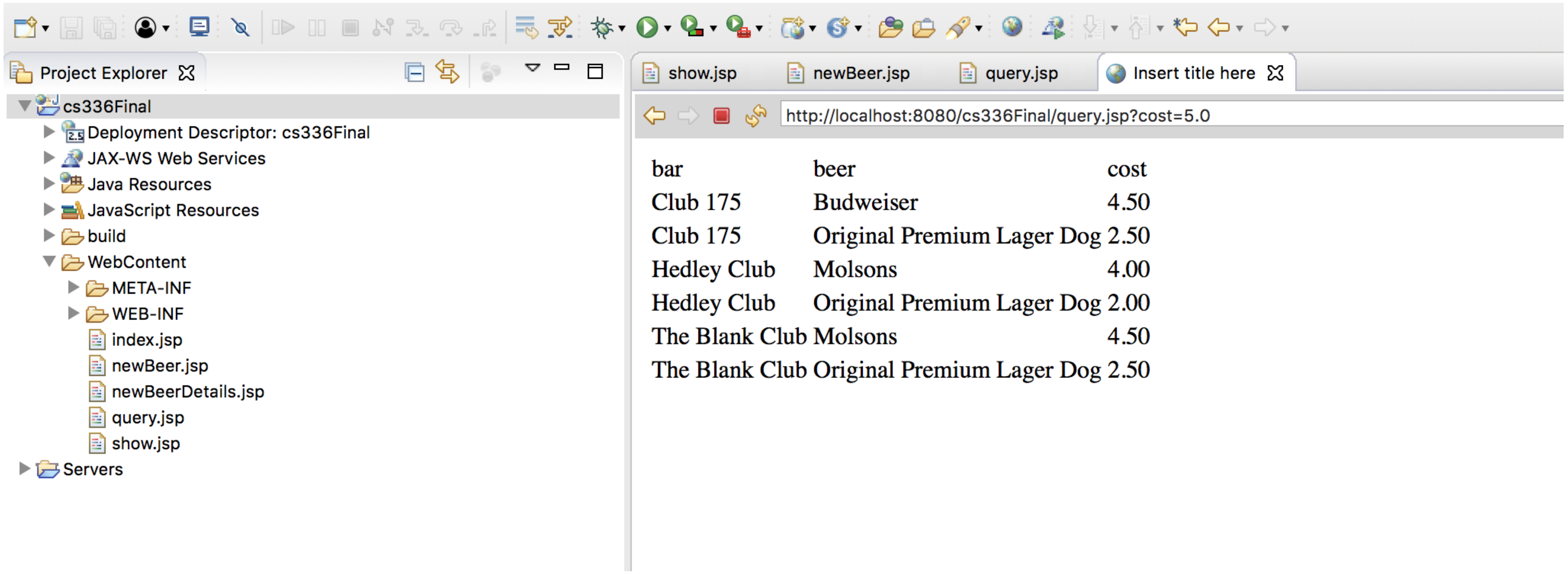
```
cs336Final
  Deployment Descriptor: cs336Final
  JAX-WS Web Services
  Java Resources
  JavaScript Resources
  build
  WebContent
    META-INF
    WEB-INF
    index.jsp
    newBeer.jsp
    newBeerDetails.jsp
    query.jsp
    show.jsp
Servers
```

http://localhost:8080/cs336Final/query.jsp?cost=3.0

bar	beer	cost
Club 175	Original Premium Lager Dog	2.50
Hedley Club	Original Premium Lager Dog	2.00
The Blank Club	Original Premium Lager Dog	2.50

The Template Project

- Query the beers with cost<= 5



http://localhost:8080/cs336Final/query.jsp?cost=5.0

```
bar            beer                        cost
Club 175       Budweiser                   4.50
Club 175       Original Premium Lager Dog  2.50
Hedley Club    Molsons                     4.00
Hedley Club    Original Premium Lager Dog  2.00
The Blank Club Molsons                     4.50
The Blank Club Original Premium Lager Dog  2.50
```