CNT 4714: Enterprise Computing
Fall 2015

Introduction to Servlet Technology– Part 3

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
Office Hours: M, T, W, & Th 3:00-4:30pm

Department of Computer Science
University of Central Florida

Servlets That Return Content Other Than Text/HTML

- The servlets that we have seen so far have all returned content which was text-based. Thus all of the servlets contained the following line of code:

```
response.setContentType("text/html");
```

- The Content-Type response header gives the MIME (Multipurpose Internet Mail Extension) type of the response document. Setting the value of this header is so common that the special method `setContentType` in `HttpServletResponse` was created.

- MIME types are of the format `maintype/subtype` for officially registered types. There are many officially registered types, some of which are shown in the table on the next page.

- The officially registered types can be found at http://www.iana.org/assignments/media-types/index.html

Some Common MIME Types

Type	Meaning
application/pdf	Acrobat (.pdf) file
application/jar	JAR file
application/vnd.ms-excel	Excel spreadsheet
application/vnd.ms-powerpoint	Powerpoint presentation
application/x-java-vm	Java bytecode (.class) file
application/zip	Zip archive
audio/midi	MIDI sound file
image/gif	GIF image
image/jpeg	JPEG image
text/html	HTML document
text/xml	XML document

Example Servlet That Returns An Excel Spreadsheet

- I’ve put an example on the code page for the class (you can run it directly, but I did not put a reference to it on the CNT4714 webapp index page) of a servlet that returns an Excel spreadsheet to the client.

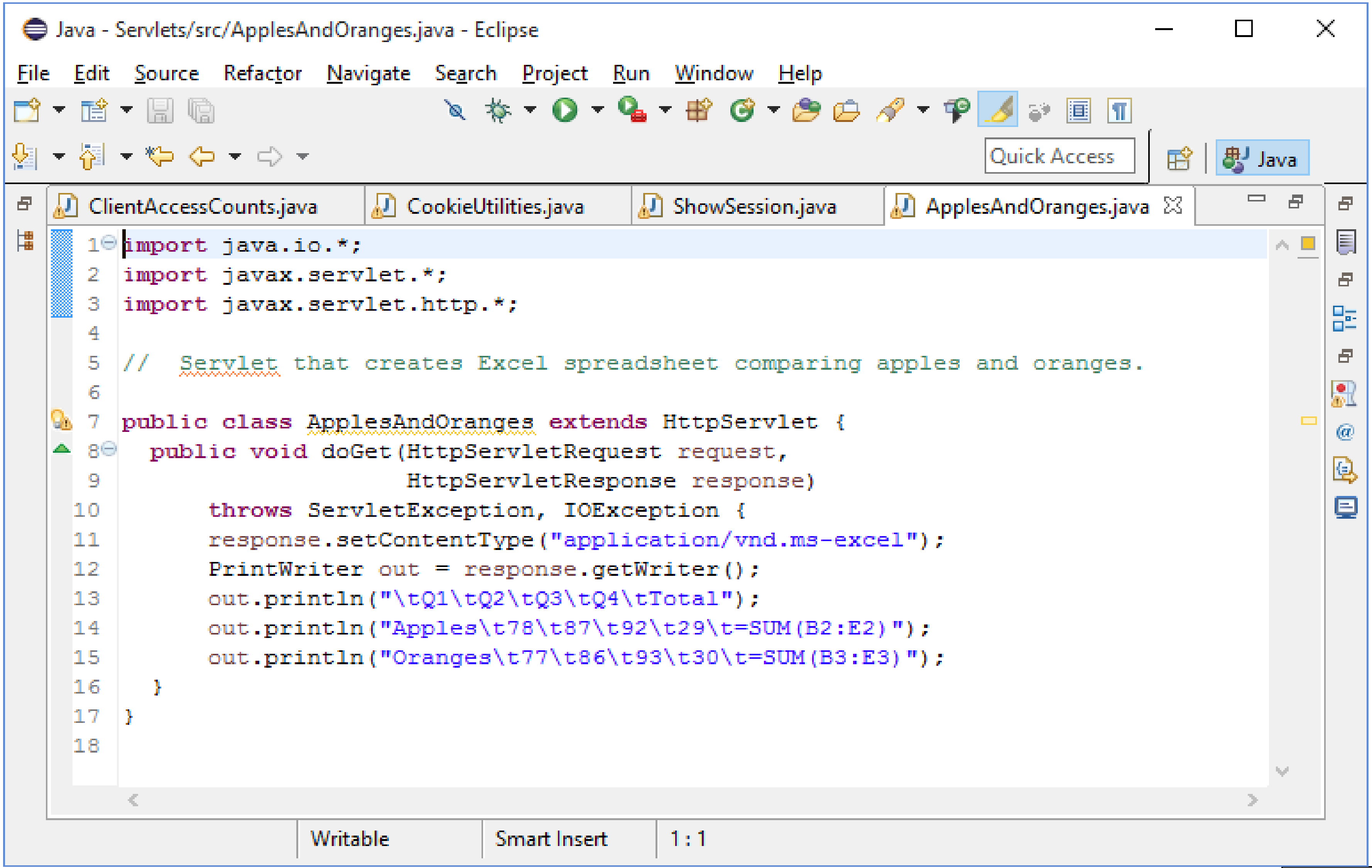
- I made this servlet very simple and it simply generates the Excel spreadsheet contents and returns it to the client. The servlet code is shown on the next page and the Excel spreadsheet that is returned is shown on the following page.

- Note that this servlet contains the following line of code:

```
response.setContentType("application/vnd.ms-excel ");
```

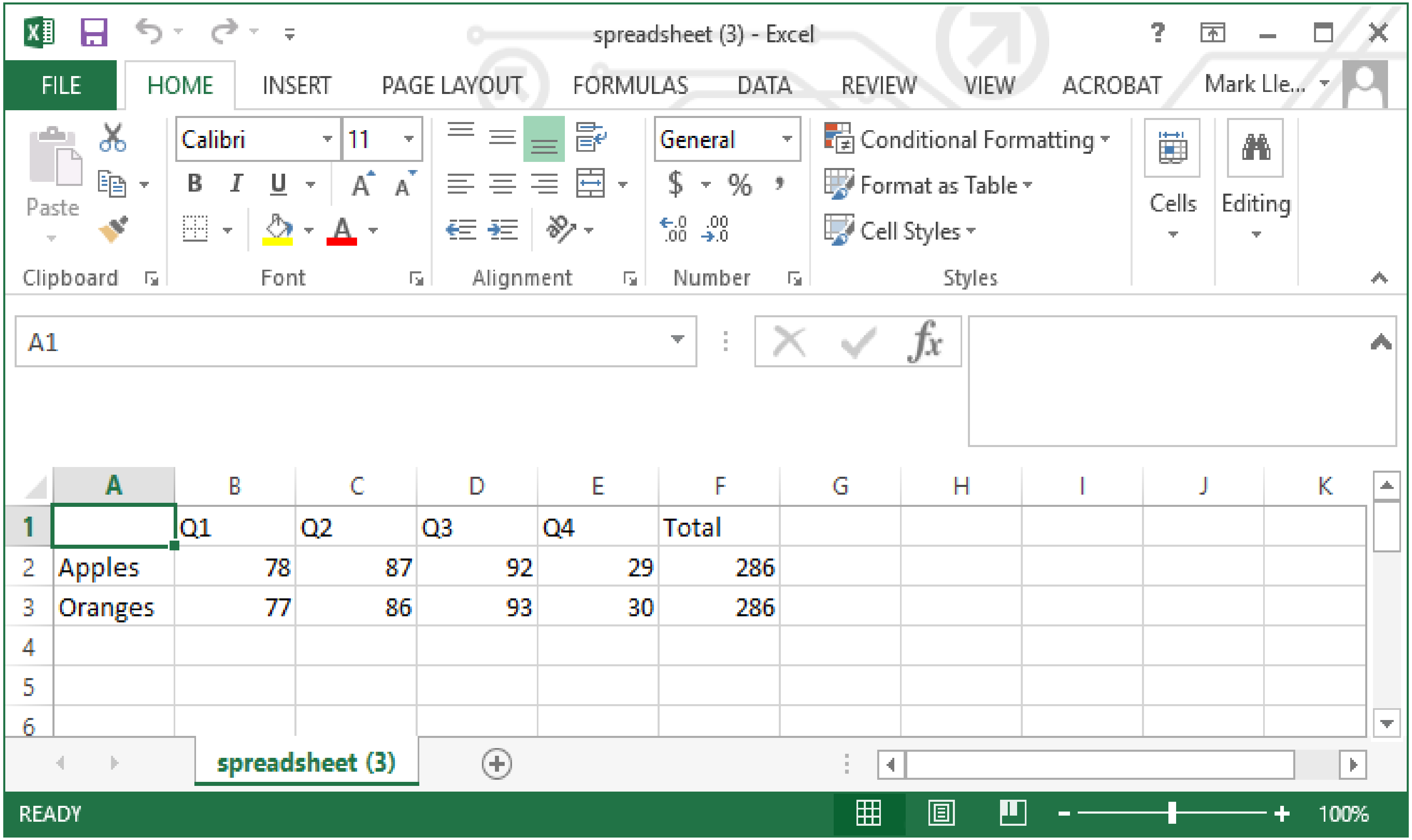
- To execute the servlet type: http://localhost:8080/CNT4714/spreadsheet

ApplesAndOranges Servlet



```java
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

//  Servlet that creates Excel spreadsheet comparing apples and oranges.

public class ApplesAndOranges extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
      response.setContentType("application/vnd.ms-excel");
      PrintWriter out = response.getWriter();
      out.println("\tQ1\tQ2\tQ3\tQ4\tTotal");
      out.println("Apples\t78\t87\t92\t29\t=SUM(B2:E2)");
      out.println("Oranges\t77\t86\t93\t30\t=SUM(B3:E3)");
  }
}
```

Response From ApplesAndOranges Servlet



	Q1	Q2	Q3	Q4	Total
Apples	78	87	92	29	286
Oranges	77	86	93	30	286

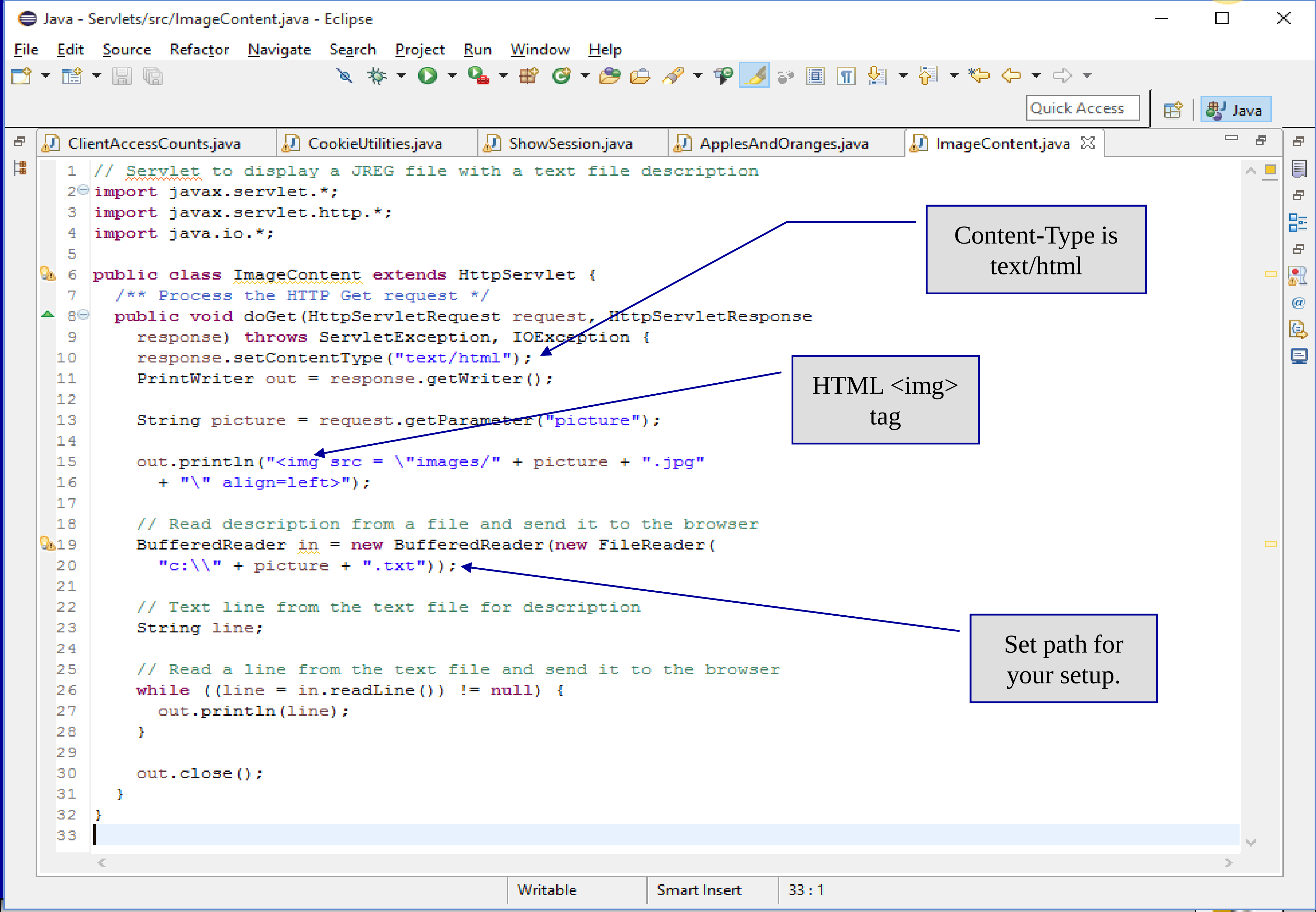
Example Servlet That Returns An Image File and Text

- You can return images from a servlet using the MIME type shown on page 6. However, if you also wish to return text along with the image a simple way to do this is to set the MIME type to text/html as before, but simply embed the image in the HTML document using the HTML `<img>` tag.

- The syntax for this tag is:

  <img src=URL alt=text align = [top | middle | bottom | texttop |…]>

- The following page illustrates a small servlet that displays such a document. I've modified the servlet index page to handle this servlet. The servlet is sent the name of the picture you wish to display. The servlet assumes that there is an accompanying description file (a `.txt` file) which provides a description of the picture being displayed. The text file is to be located in the root directory on the C: drive. I've only put two sets of files out there for you to use named: "Eddy Merckx" and "sprint kart". Feel free to add some of your own.

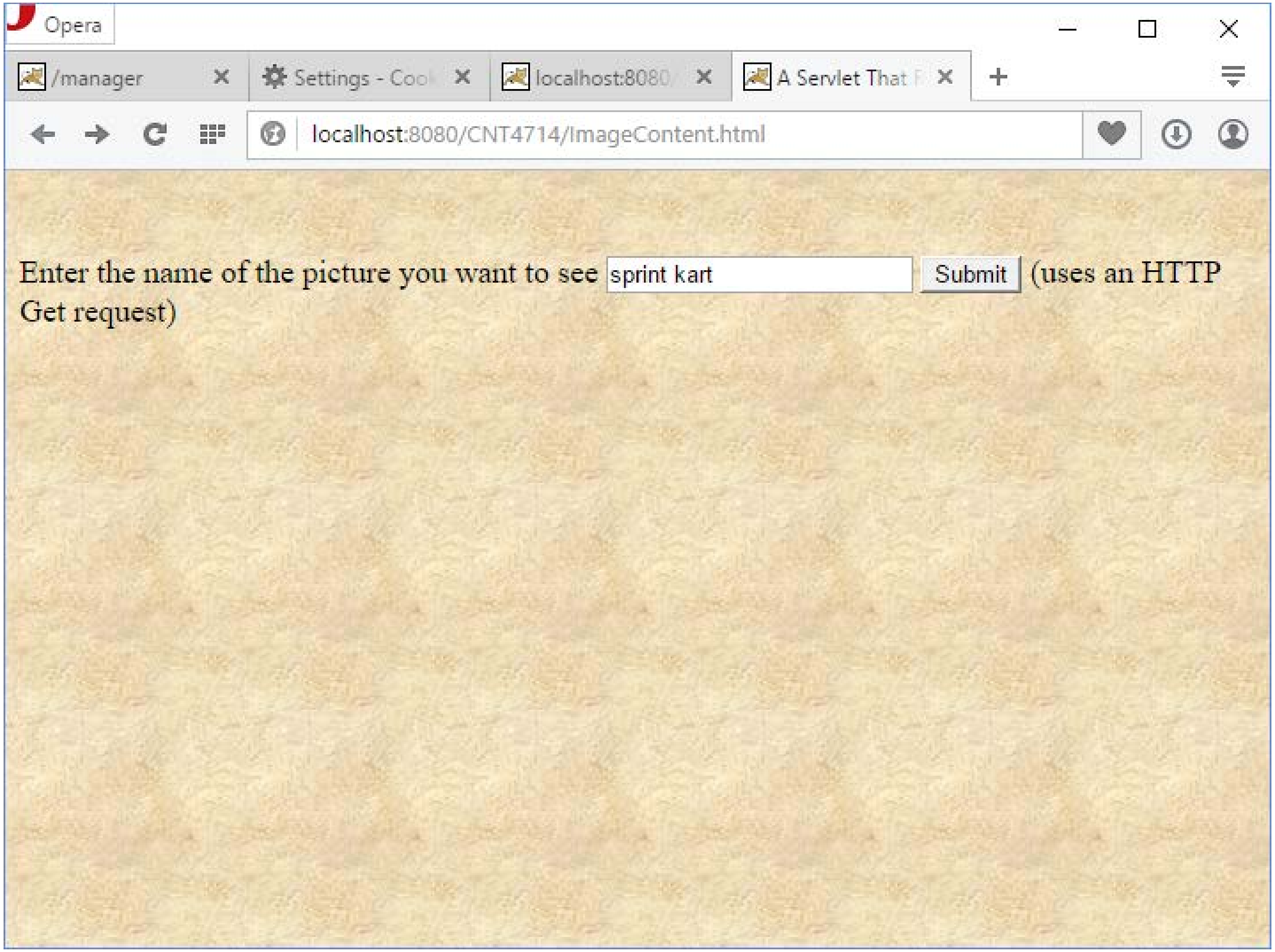


```java
// Servlet to display a JREG file with a text file description
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class ImageContent extends HttpServlet {
  /** Process the HTTP Get request */
  public void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    String picture = request.getParameter("picture");

    out.println("<img src = \"images/" + picture + ".jpg"
      + "\" align=left>");

    // Read description from a file and send it to the browser
    BufferedReader in = new BufferedReader(new FileReader(
      "c:\\" + picture + ".txt"));

    // Text line from the text file for description
    String line;

    // Read a line from the text file and send it to the browser
    while ((line = in.readLine()) != null) {
      out.println(line);
    }

    out.close();
  }
}
```

Content-Type is text/html

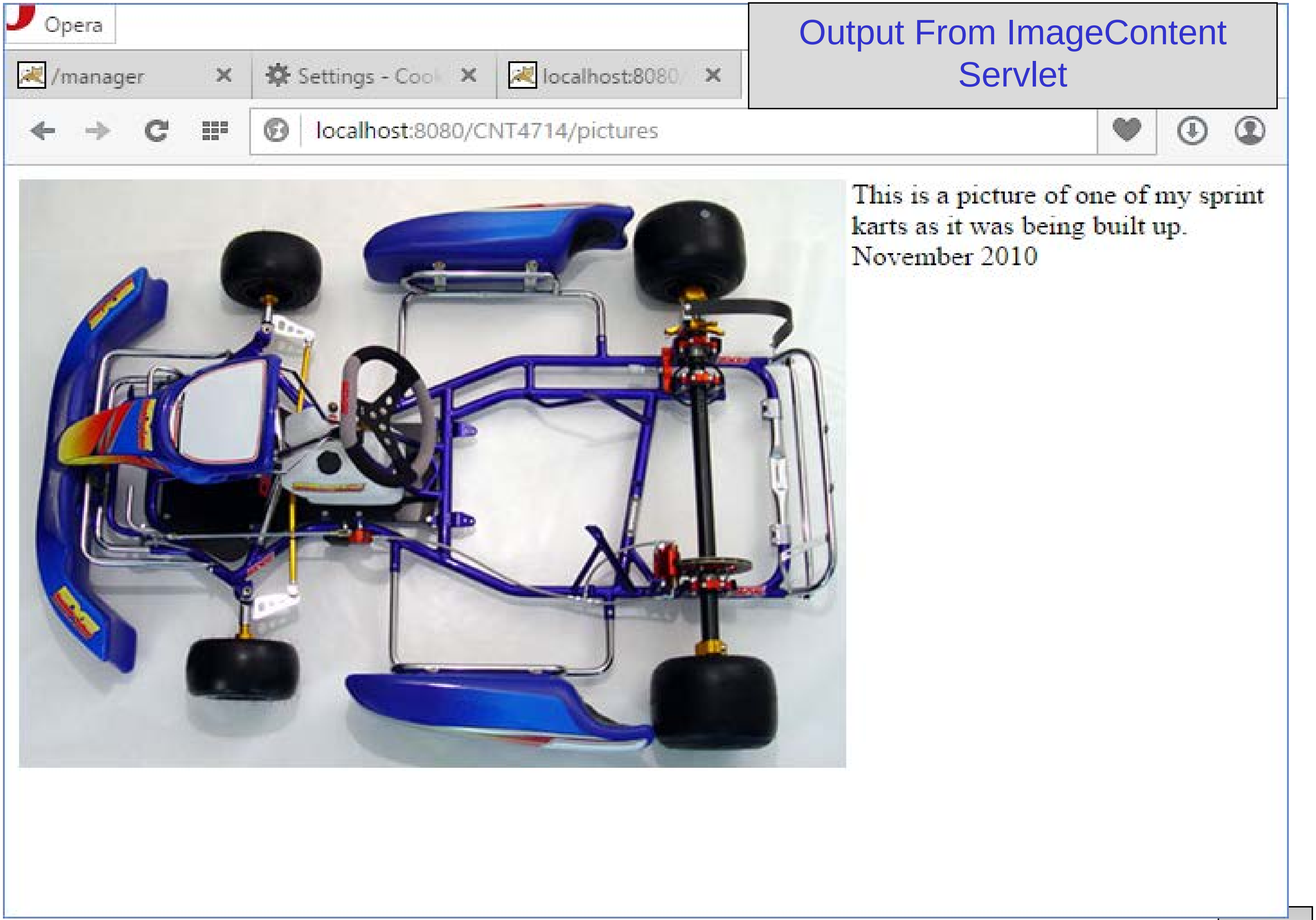
HTML <img> tag

Set path for your setup.



Enter the name of the picture you want to see sprint kart Submit (uses an HTTP Get request)

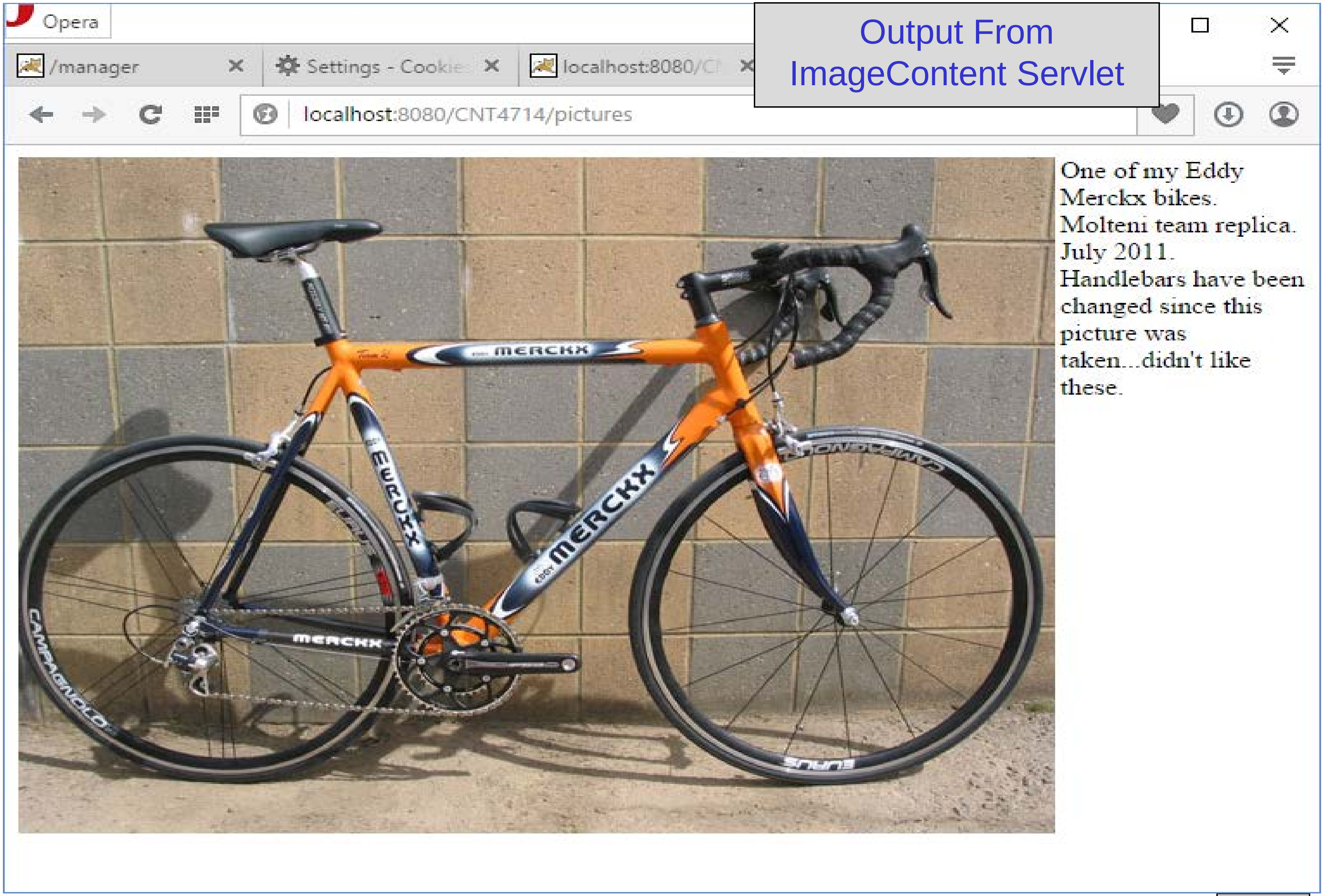
Output From ImageContent Servlet



localhost:8080/CNT4714/pictures

This is a picture of one of my sprint karts as it was being built up. November 2010

Output From ImageContent Servlet



One of my Eddy Merckx bikes. Molteni team replica. July 2011. Handlebars have been changed since this picture was taken...didn't like these.

Multi-tier Applications: Using JDBC From A Servlet

- Many of today's web applications are three-tier distributed applications, consisting of a user interface, business logic, and a database.
  - The first-tier or front-end is the user interface which is typically created using HTML.
  - Using the networking provided by the browser, the user interface communicates with the middle-tier business logic.
  - The middle-tier accesses the third-tier or backend database to manipulate the data.
- The three-tiers will often reside on separate computer systems which are connected through a network.

Multi-tier Applications: Using JDBC From A Servlet (cont.)

- In multi-tier architectures, web servers are often used in the middle-tier.

- Server-side components, such as servlets, execute in an application server alongside the web server. These components provide the business logic that manipulates the data from databases and communicates with client web browsers.

- Servlets, through JDBC, can interact with database systems.

- We’ll develop a small three-tier application that allows the user to interact with a database via a small on-line survey.

Multi-tier Applications: Using JDBC From A Servlet (cont.)

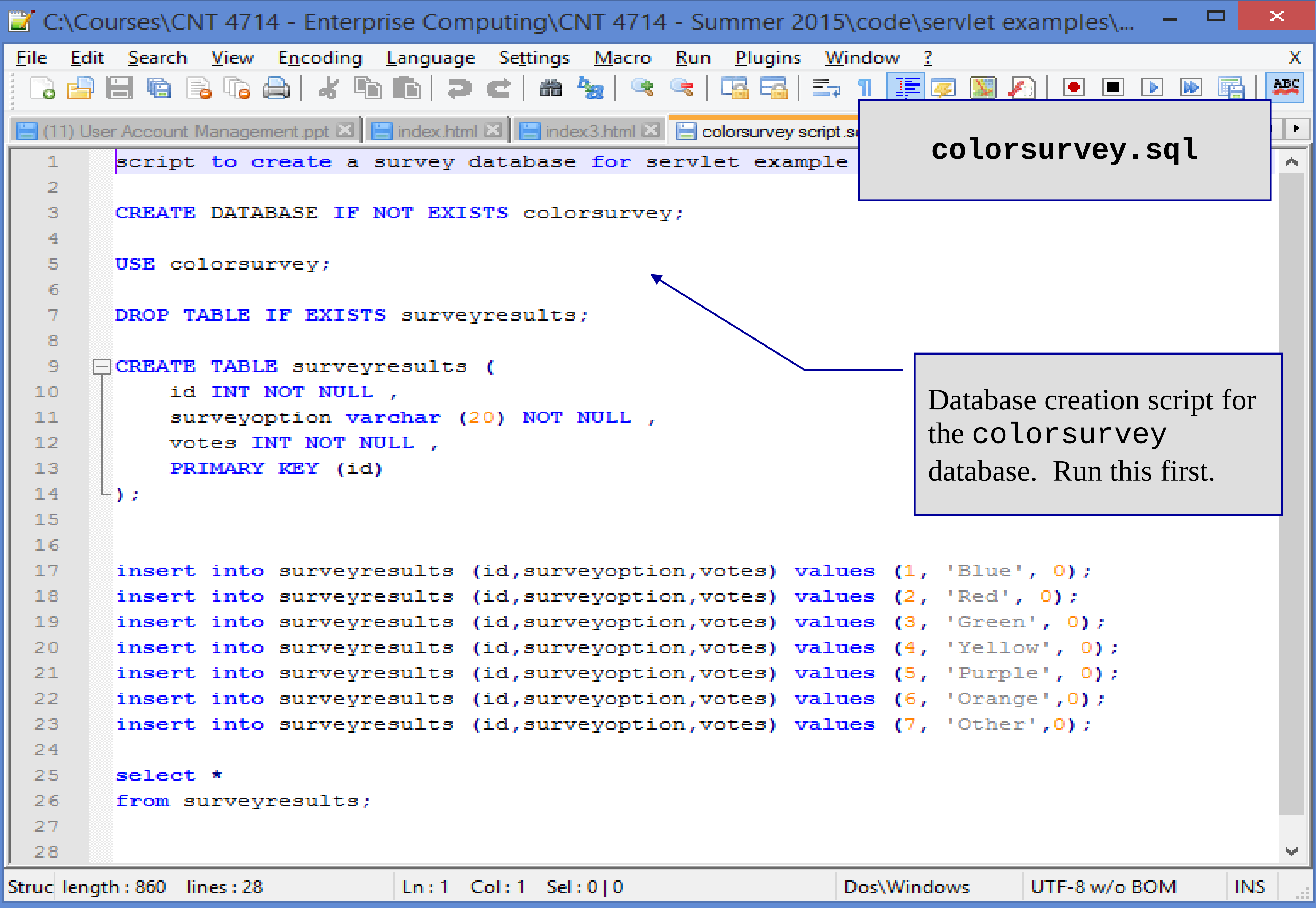
- SurveyServlet implements the middle-tier of our application which handles requests from the client browser (the front-end) and provides access to the third-tier – a MySQL database access via JDBC. Copy the `mysql-connector-java.5.1.36-bin.jar` file into the `WEB-INF/lib` folder.
- The servlet will allow the user to select their favorite color.
- When the servlet receives a post request from the web browser (the user has selected their favorite color), the servlet uses JDBC to update the total number of votes for that color choice in the database and returns a dynamically generated HTML document containing the survey results to the client.

Multi-tier Applications: Using JDBC From A Servlet (cont.)

- As before this web application is accessible from our index page using the `colorsurvey.html` file. The contents of this file are shown on page 16.

- The portion of the `web.xml` file that pertains to the color survey is shown on page 17.

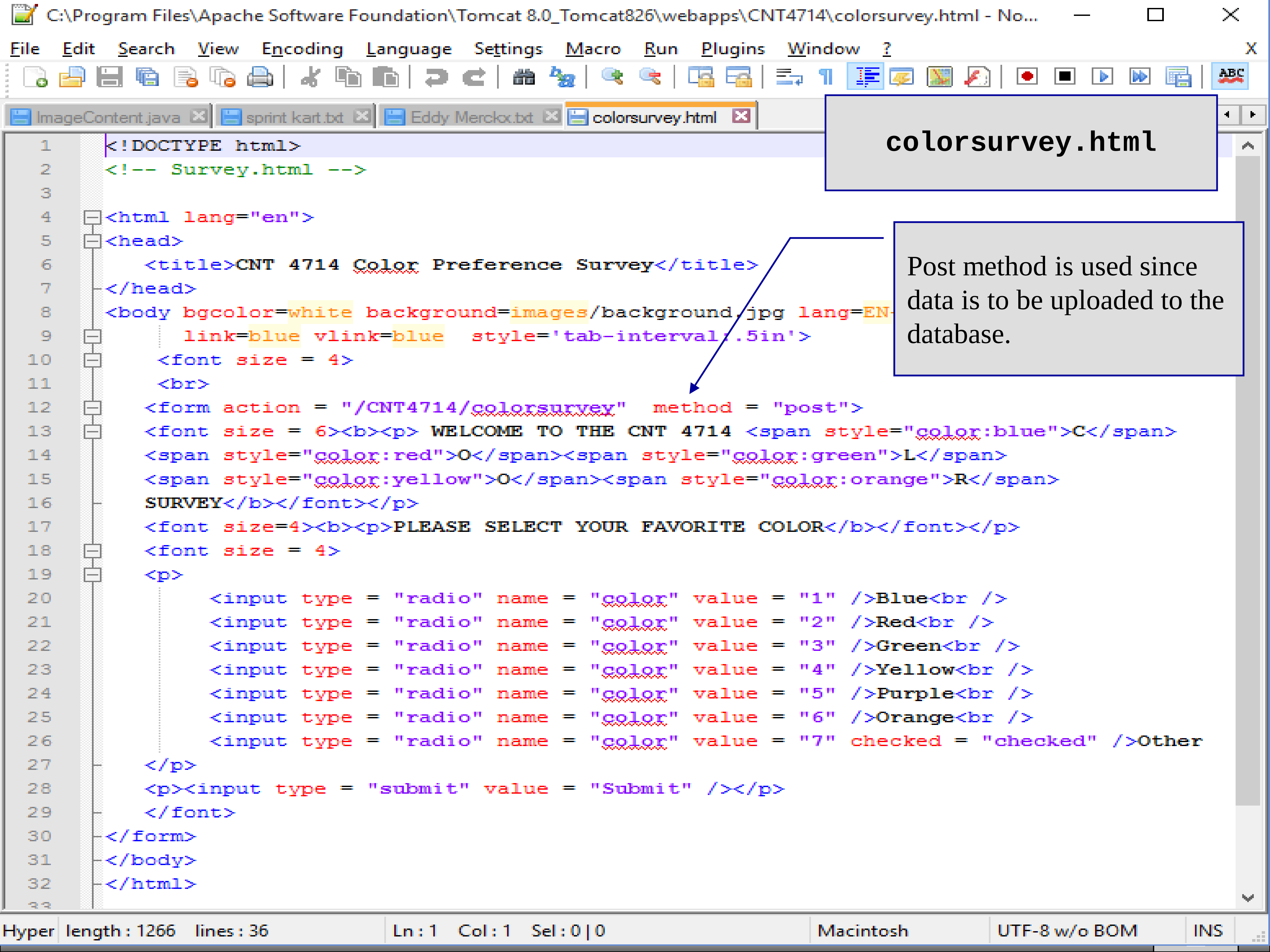
- Before this web application will run successfully, you will need to create the database it uses. I've provided a script file on the course website (code page) for creating the database. This script is also shown on the next page.


C:\Courses\CNT 4714 - Enterprise Computing\CNT 4714 - Summer 2015\code\servlet examples\...
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
(11) User Account Management.ppt
index.html
index3.html
colorsurvey script.sq
Struc length : 860 lines : 28 Ln : 1 Col : 1 Sel : 0 | 0 Dos\Windows UTF-8 w/o BOM INS

```
script to create a survey database for servlet example

CREATE DATABASE IF NOT EXISTS colorsurvey;

USE colorsurvey;

DROP TABLE IF EXISTS surveyresults;

CREATE TABLE surveyresults (
    id INT NOT NULL ,
    surveyoption varchar (20) NOT NULL ,
    votes INT NOT NULL ,
    PRIMARY KEY (id)
);


insert into surveyresults (id,surveyoption,votes) values (1, 'Blue', 0);
insert into surveyresults (id,surveyoption,votes) values (2, 'Red', 0);
insert into surveyresults (id,surveyoption,votes) values (3, 'Green', 0);
insert into surveyresults (id,surveyoption,votes) values (4, 'Yellow', 0);
insert into surveyresults (id,surveyoption,votes) values (5, 'Purple', 0);
insert into surveyresults (id,surveyoption,votes) values (6, 'Orange',0);
insert into surveyresults (id,surveyoption,votes) values (7, 'Other',0);

select *
from surveyresults;
```

colorsurvey.sql

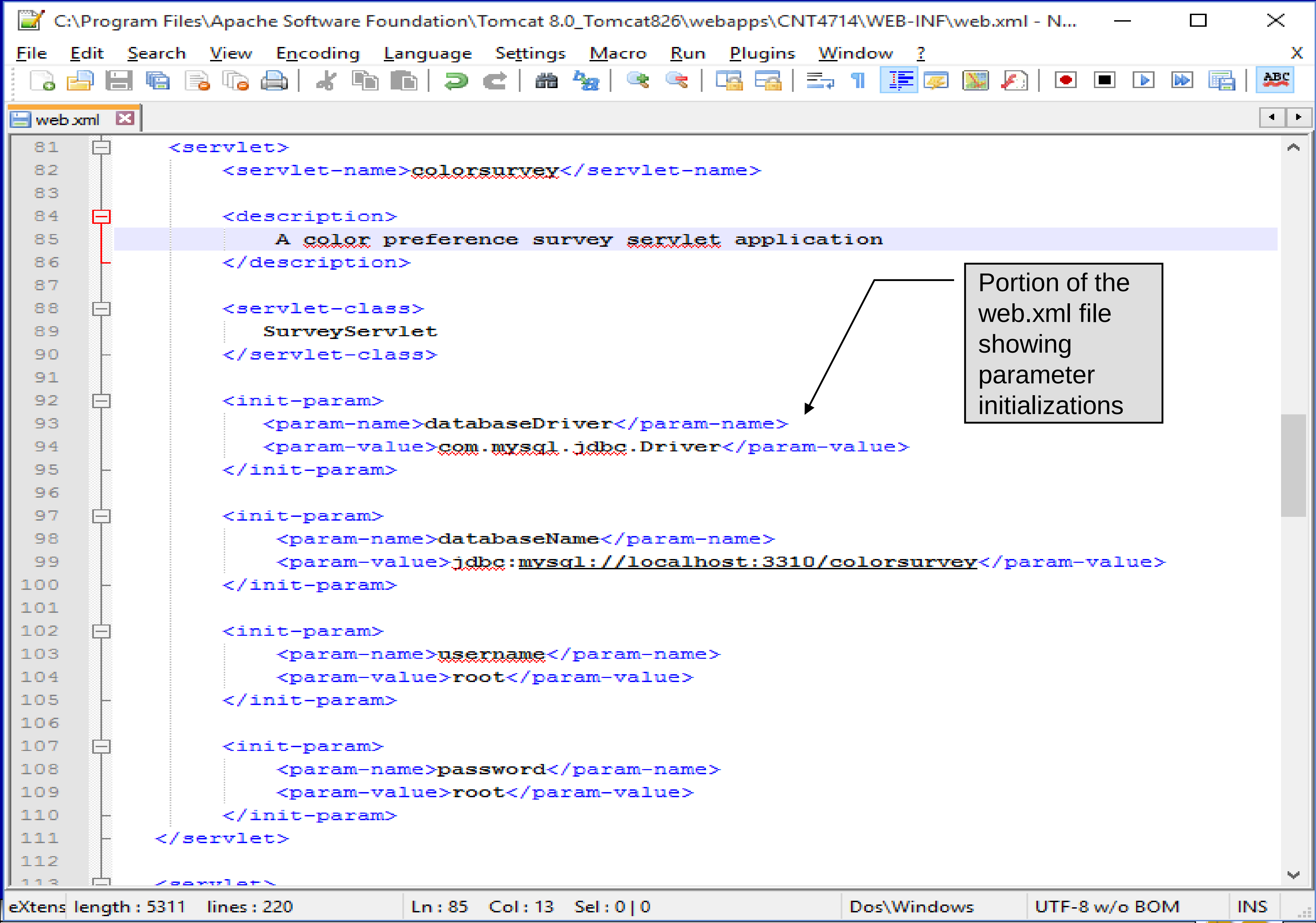
Database creation script for the `colorsurvey` database. Run this first.


C:\Program Files\Apache Software Foundation\Tomcat 8.0_Tomcat826\webapps\CNT4714\colorsurvey.html - No...
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
ImageContent.java
sprint kart.txt
Eddy Merckx.txt
colorsurvey.html
Hyper length : 1266 lines : 36 Ln : 1 Col : 1 Sel : 0 | 0 Macintosh UTF-8 w/o BOM INS

colorsurvey.html

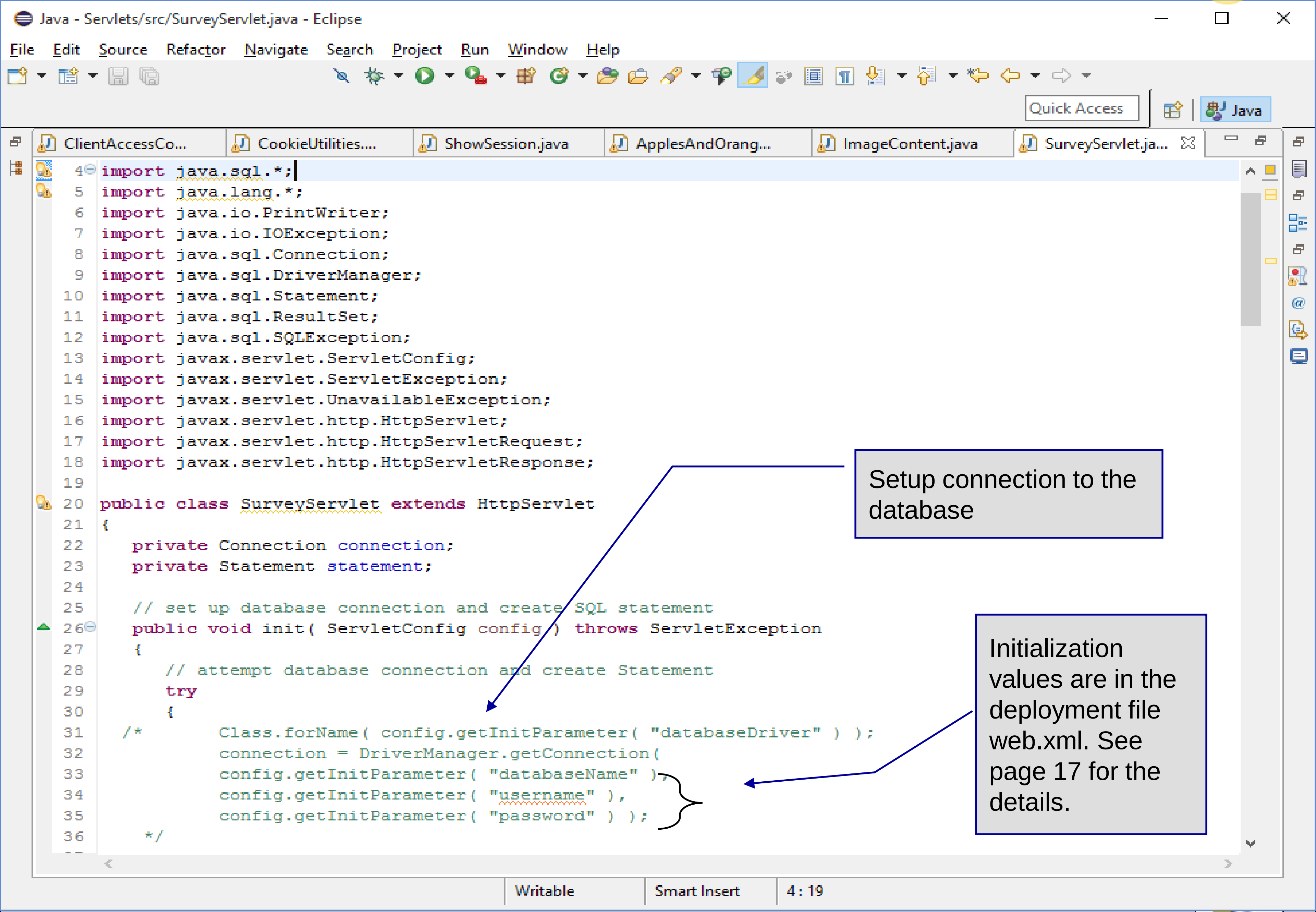
```
<!DOCTYPE html>
<!-- Survey.html -->

<html lang="en">
<head>
   <title>CNT 4714 Color Preference Survey</title>
</head>
<body bgcolor=white background=images/background.jpg lang=EN-
      link=blue vlink=blue  style='tab-interval:.5in'>
    <font size = 4>
    <br>
   <form action = "/CNT4714/colorsurvey"  method = "post">
   <font size = 6><b><p> WELCOME TO THE CNT 4714 <span style="color:blue">C</span>
   <span style="color:red">O</span><span style="color:green">L</span>
   <span style="color:yellow">O</span><span style="color:orange">R</span>
   SURVEY</b></font></p>
   <font size=4><b><p>PLEASE SELECT YOUR FAVORITE COLOR</b></font></p>
   <font size = 4>
   <p>
        <input type = "radio" name = "color" value = "1" />Blue<br />
        <input type = "radio" name = "color" value = "2" />Red<br />
        <input type = "radio" name = "color" value = "3" />Green<br />
        <input type = "radio" name = "color" value = "4" />Yellow<br />
        <input type = "radio" name = "color" value = "5" />Purple<br />
        <input type = "radio" name = "color" value = "6" />Orange<br />
        <input type = "radio" name = "color" value = "7" checked = "checked" />Other
   </p>
   <p><input type = "submit" value = "Submit" /></p>
   </font>
</form>
</body>
</html>
```

Post method is used since data is to be uploaded to the database.


C:\Program Files\Apache Software Foundation\Tomcat 8.0_Tomcat826\webapps\CNT4714\WEB-INF\web.xml - N...
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
web.xml
eXtens length : 5311 lines : 220
Ln : 85 Col : 13 Sel : 0 | 0
Dos\Windows
UTF-8 w/o BOM
INS

```
    <servlet>
        <servlet-name>colorsurvey</servlet-name>

        <description>
            A color preference survey servlet application
        </description>

        <servlet-class>
           SurveyServlet
        </servlet-class>

        <init-param>
           <param-name>databaseDriver</param-name>
           <param-value>com.mysql.jdbc.Driver</param-value>
        </init-param>

        <init-param>
            <param-name>databaseName</param-name>
            <param-value>jdbc:mysql://localhost:3310/colorsurvey</param-value>
        </init-param>

        <init-param>
            <param-name>username</param-name>
            <param-value>root</param-value>
        </init-param>

        <init-param>
            <param-name>password</param-name>
            <param-value>root</param-value>
        </init-param>
   </servlet>

   <servlet>
```

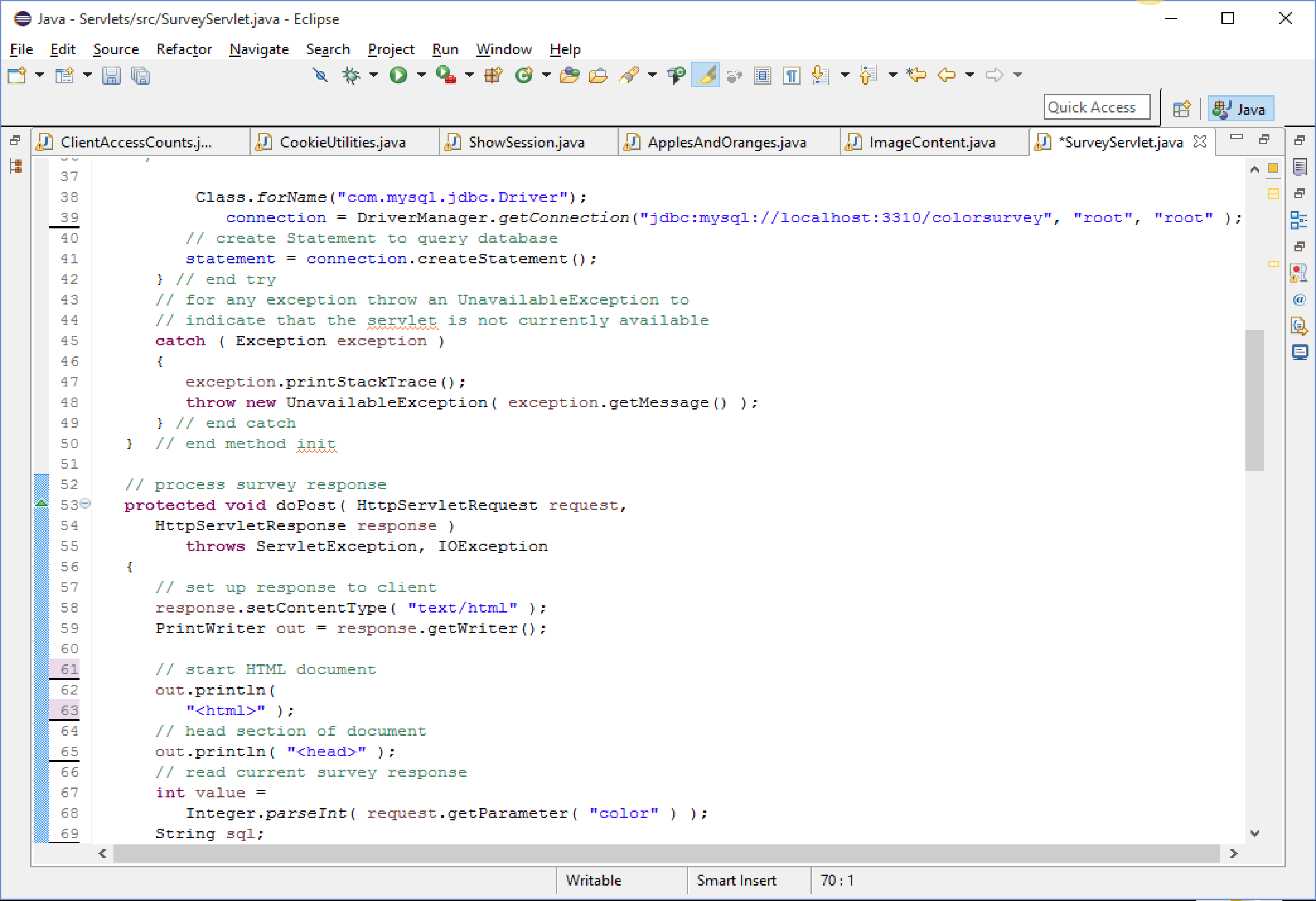
Portion of the web.xml file showing parameter initializations



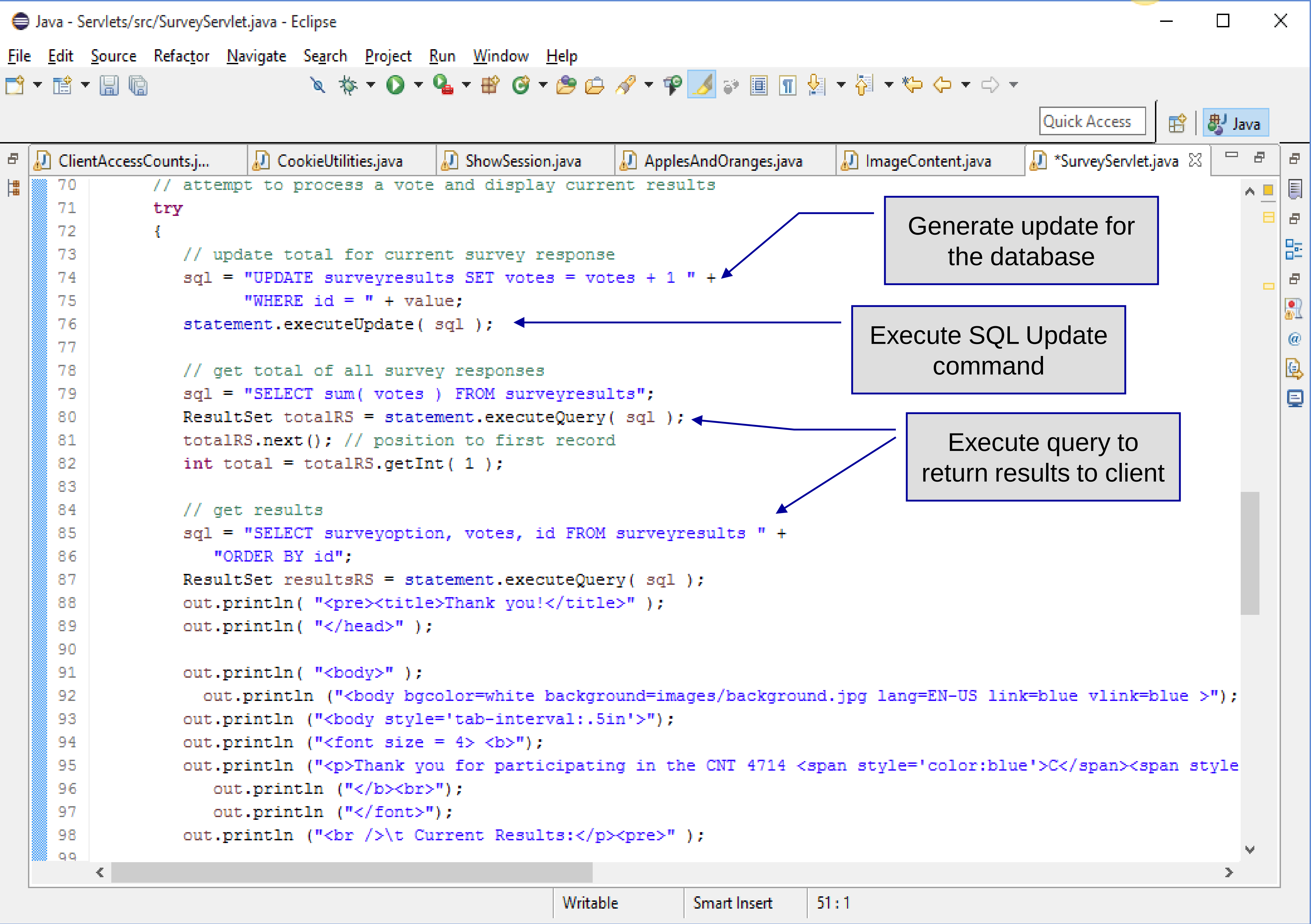
```java
import java.sql.*;
import java.lang.*;
import java.io.PrintWriter;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.UnavailableException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class SurveyServlet extends HttpServlet
{
   private Connection connection;
   private Statement statement;

   // set up database connection and create SQL statement
   public void init( ServletConfig config ) throws ServletException
   {
      // attempt database connection and create Statement
      try
      {
  /*         Class.forName( config.getInitParameter( "databaseDriver" ) );
           connection = DriverManager.getConnection(
           config.getInitParameter( "databaseName" ),
           config.getInitParameter( "username" ),
           config.getInitParameter( "password" ) );
    */
```

Setup connection to the database

Initialization values are in the deployment file web.xml. See page 17 for the details.



```

           Class.forName("com.mysql.jdbc.Driver");
              connection = DriverManager.getConnection("jdbc:mysql://localhost:3310/colorsurvey", "root", "root" );
          // create Statement to query database
          statement = connection.createStatement();
       } // end try
       // for any exception throw an UnavailableException to
       // indicate that the servlet is not currently available
       catch ( Exception exception )
       {
          exception.printStackTrace();
          throw new UnavailableException( exception.getMessage() );
       } // end catch
    }  // end method init

    // process survey response
    protected void doPost( HttpServletRequest request,
       HttpServletResponse response )
          throws ServletException, IOException
    {
       // set up response to client
       response.setContentType( "text/html" );
       PrintWriter out = response.getWriter();

       // start HTML document
       out.println(
          "<html>" );
       // head section of document
       out.println( "<head>" );
       // read current survey response
       int value =
          Integer.parseInt( request.getParameter( "color" ) );
       String sql;
```

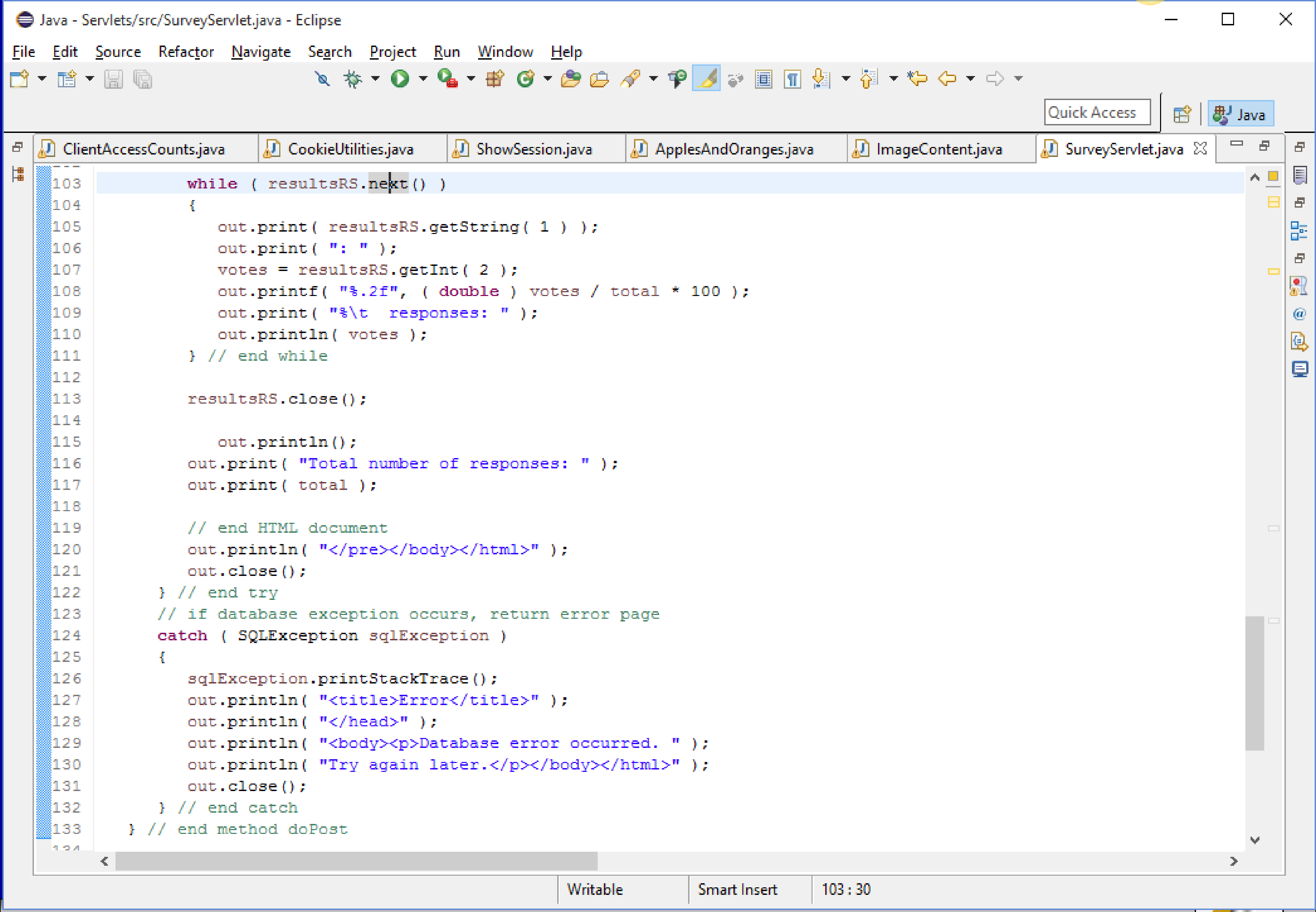


```java
      // attempt to process a vote and display current results
      try
      {
         // update total for current survey response
         sql = "UPDATE surveyresults SET votes = votes + 1 " +
               "WHERE id = " + value;
         statement.executeUpdate( sql );

         // get total of all survey responses
         sql = "SELECT sum( votes ) FROM surveyresults";
         ResultSet totalRS = statement.executeQuery( sql );
         totalRS.next(); // position to first record
         int total = totalRS.getInt( 1 );

         // get results
         sql = "SELECT surveyoption, votes, id FROM surveyresults " +
            "ORDER BY id";
         ResultSet resultsRS = statement.executeQuery( sql );
         out.println( "<pre><title>Thank you!</title>" );
         out.println( "</head>" );

         out.println( "<body>" );
           out.println ("<body bgcolor=white background=images/background.jpg lang=EN-US link=blue vlink=blue >");
         out.println ("<body style='tab-interval:.5in'>");
         out.println ("<font size = 4> <b>");
         out.println ("<p>Thank you for participating in the CNT 4714 <span style='color:blue'>C</span><span style
            out.println ("</b><br>");
            out.println ("</font>");
         out.println ("<br />\t Current Results:</p><pre>" );
```

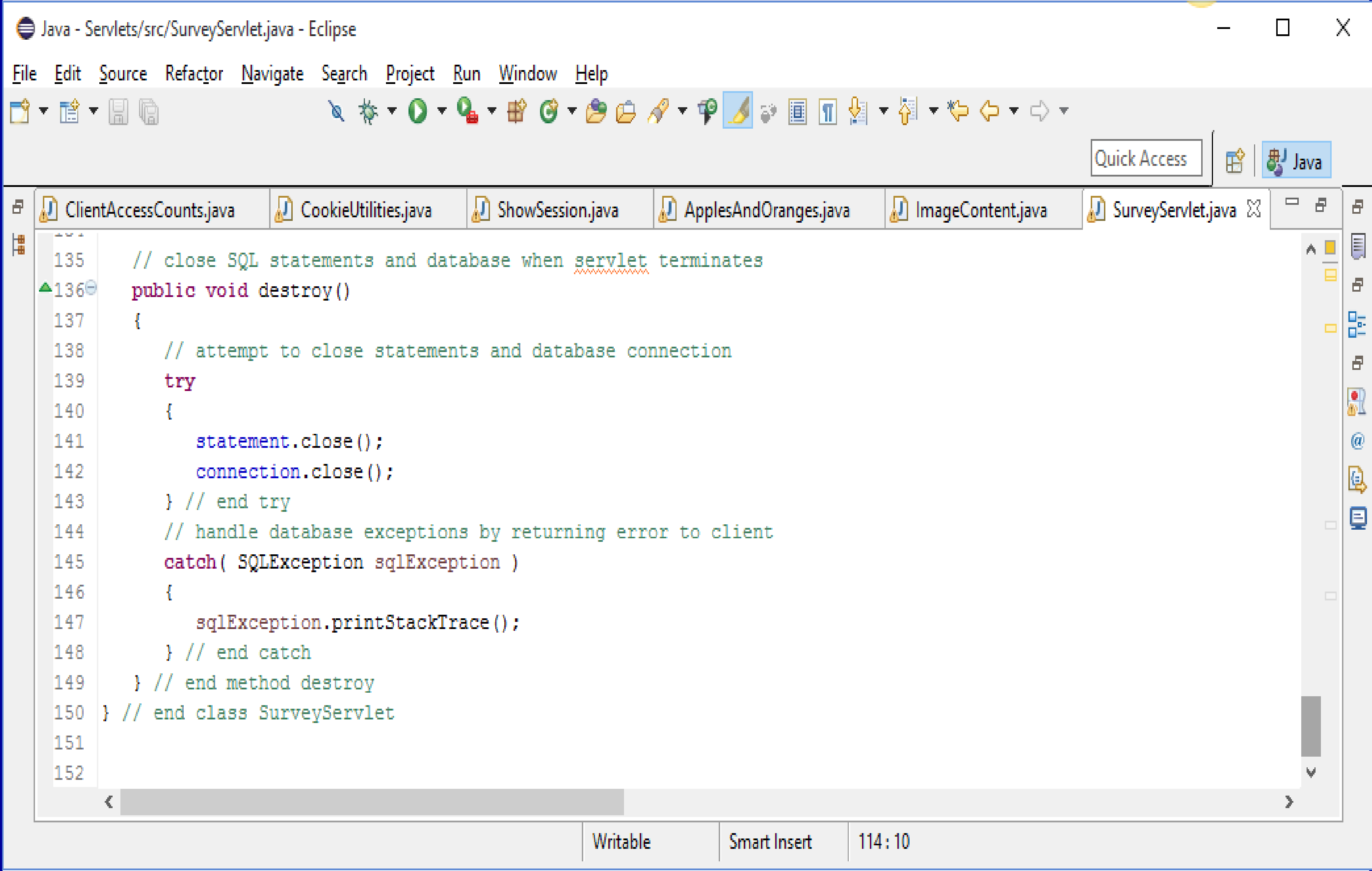
Generate update for the database

Execute SQL Update command

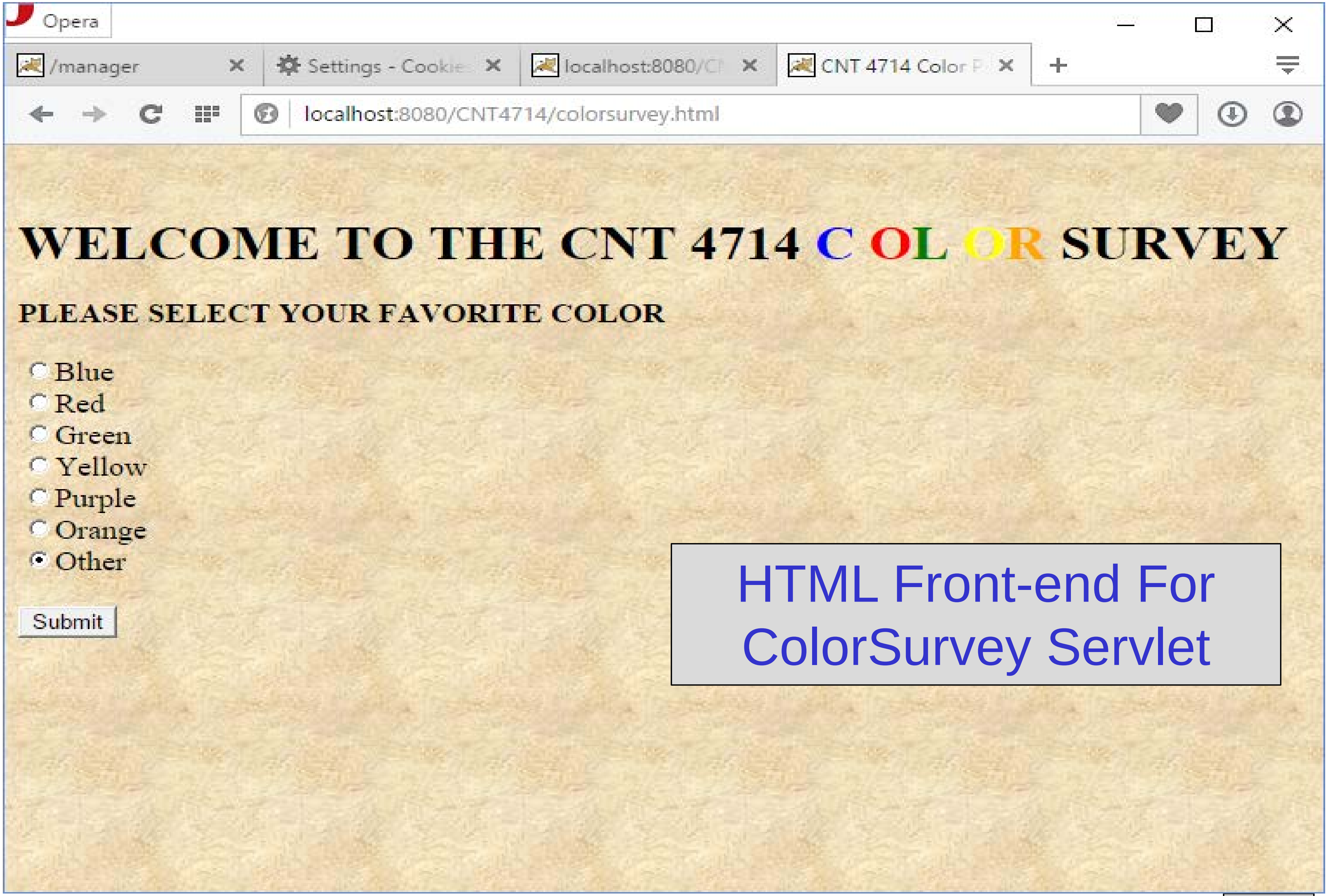
Execute query to return results to client



```
         while ( resultsRS.next() )
         {
            out.print( resultsRS.getString( 1 ) );
            out.print( ": " );
            votes = resultsRS.getInt( 2 );
            out.printf( "%.2f", ( double ) votes / total * 100 );
            out.print( "%\t  responses: " );
            out.println( votes );
         } // end while

         resultsRS.close();

            out.println();
         out.print( "Total number of responses: " );
         out.print( total );

         // end HTML document
         out.println( "</pre></body></html>" );
         out.close();
      } // end try
      // if database exception occurs, return error page
      catch ( SQLException sqlException )
      {
         sqlException.printStackTrace();
         out.println( "<title>Error</title>" );
         out.println( "</head>" );
         out.println( "<body><p>Database error occurred. " );
         out.println( "Try again later.</p></body></html>" );
         out.close();
      } // end catch
   } // end method doPost
```



```java
   // close SQL statements and database when servlet terminates
   public void destroy()
   {
      // attempt to close statements and database connection
      try
      {
         statement.close();
         connection.close();
      } // end try
      // handle database exceptions by returning error to client
      catch( SQLException sqlException )
      {
         sqlException.printStackTrace();
      } // end catch
   } // end method destroy
} // end class SurveyServlet
```



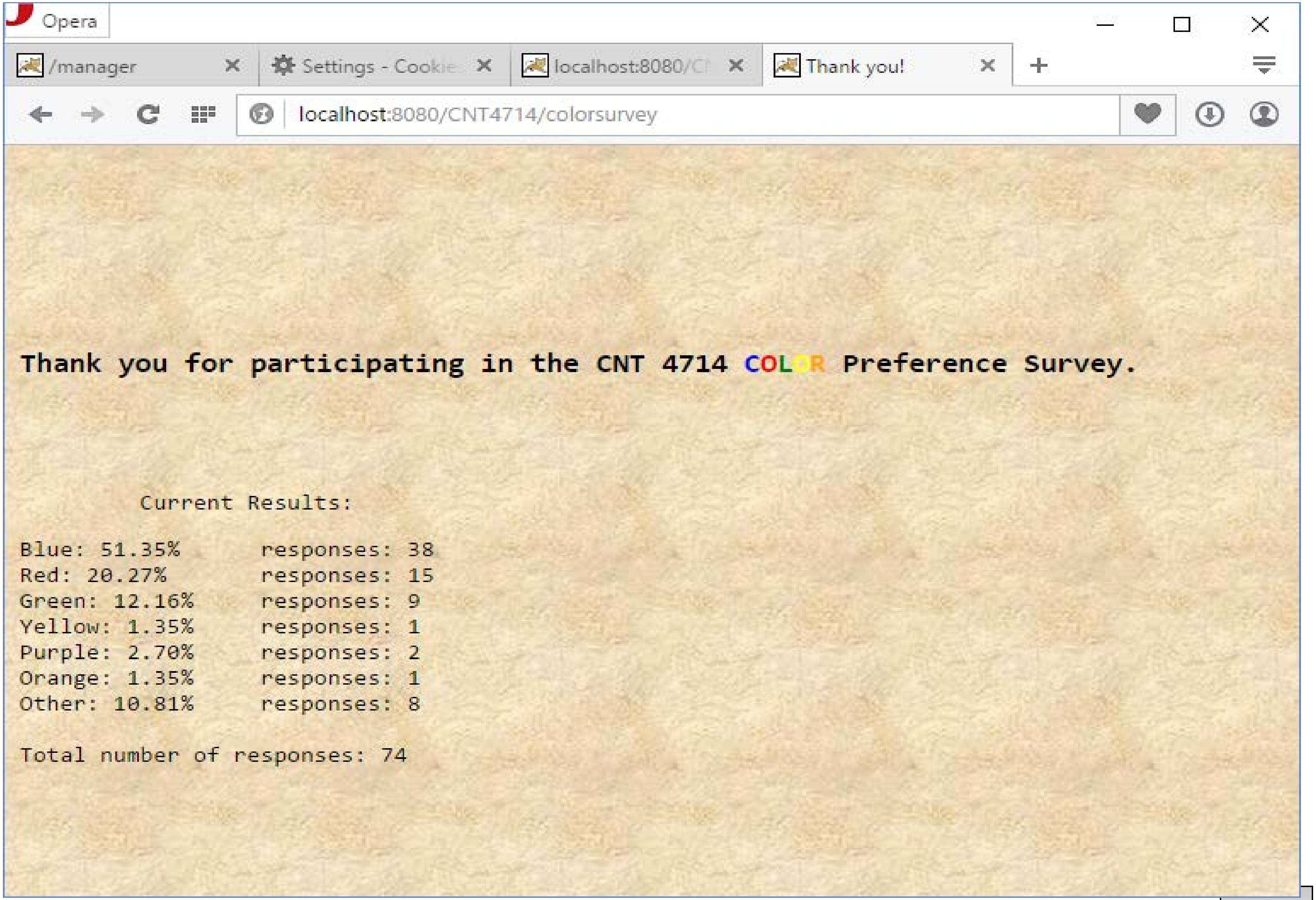
WELCOME TO THE CNT 4714 C OL OR SURVEY

PLEASE SELECT YOUR FAVORITE COLOR

- ○ Blue
- ○ Red
- ○ Green
- ○ Yellow
- ○ Purple
- ○ Orange
- ● Other

Submit

HTML Front-end For ColorSurvey Servlet



```
Thank you for participating in the CNT 4714 COLOR Preference Survey.



         Current Results:

Blue: 51.35%          responses: 38
Red: 20.27%           responses: 15
Green: 12.16%         responses: 9
Yellow: 1.35%         responses: 1
Purple: 2.70%         responses: 2
Orange: 1.35%         responses: 1
Other: 10.81%         responses: 8

Total number of responses: 74
```