CNT 4714: Enterprise Computing
Fall 2011

Introduction to Servlet Technology– Part 5

Instructor :   Dr. Mark Llewellyn
         markl@cs.ucf.edu
         HEC 236, 407-823-2790
         http://www.cs.ucf.edu/courses/cnt4714/fall2011

Department of Electrical Engineering and Computer Science
Computer Science Division
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 servlet 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
//Servlet that returns an Excel spreadsheet
//Spreadsheet compares apples and oranges!!

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

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)");
  }
}
```
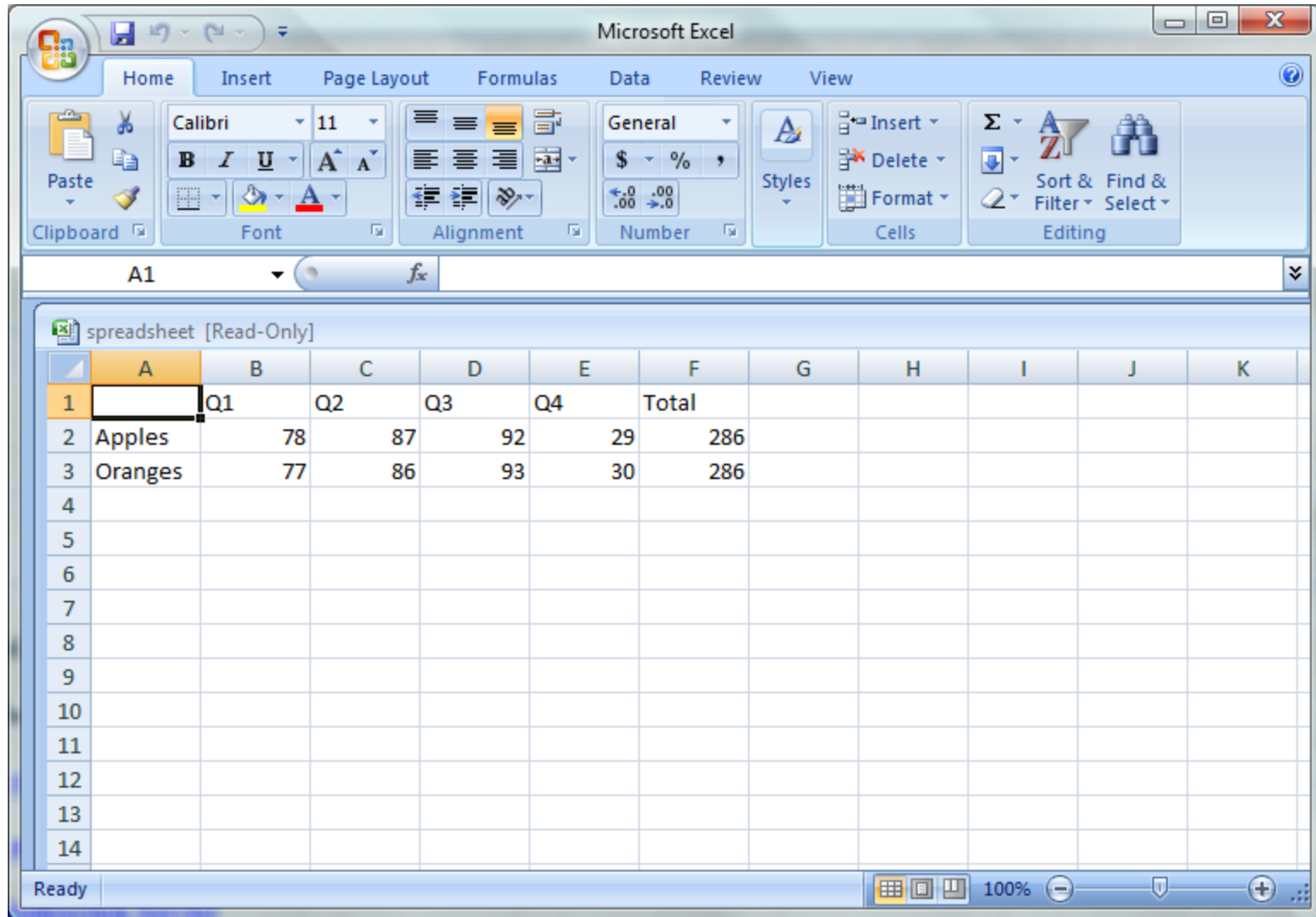
Response From ApplesAndOranges Servlet

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.

ImageContent Servlet

```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 the 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();
            }
        }
```
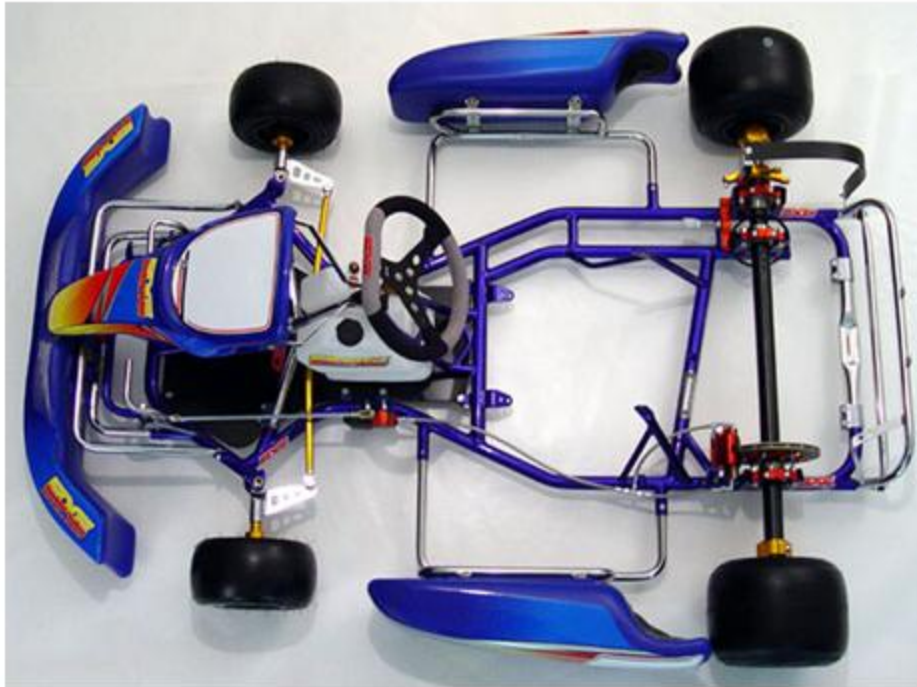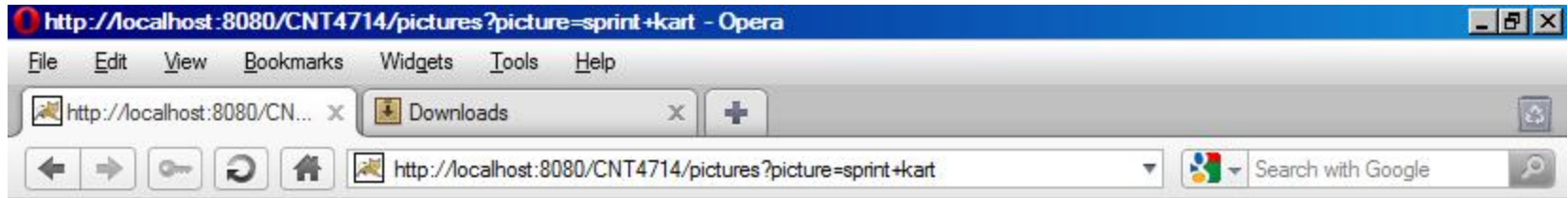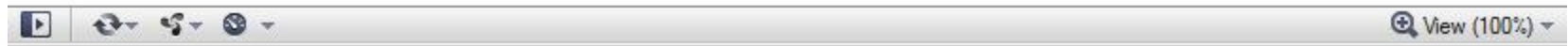
Content-Type is text/html
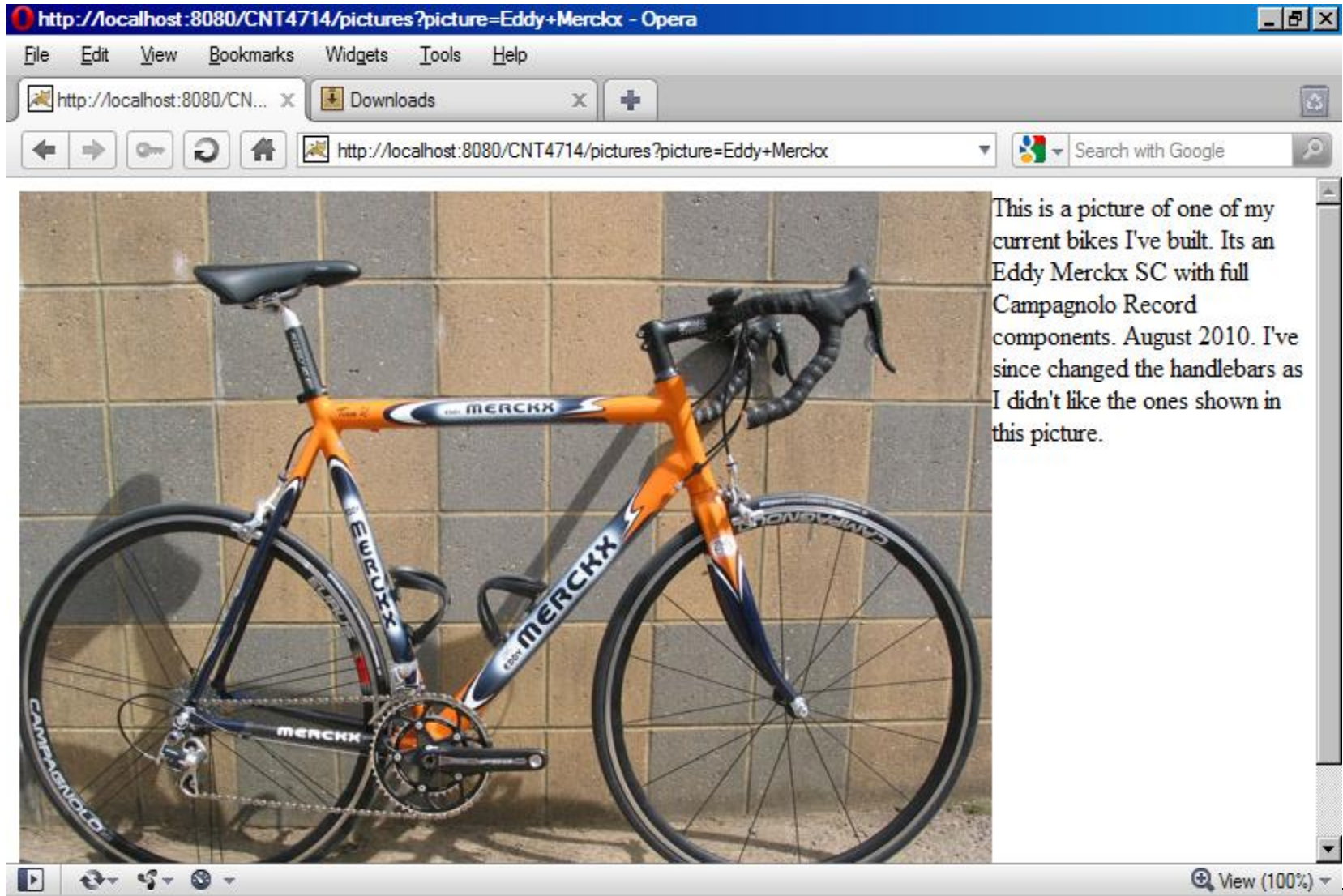
HTML <img> tag

Set path for your setup.

Output From ImageContent Servlet

Output From ImageContent Servlet

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 or XHTML.

  - 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.5.1.17-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 XHTML 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 the next page.

• The portion of the `web.xml` file that pertains to the color survey is shown on page 15.

• 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.

File   Edit   Format   View   Help

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Survey.html -->

<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
    <title>CNT 4714 Color Preference Survey</title>
</head>

<body>
    <font size = 4>
    <body bgcolor=white background=images/background.jpg lang=EN-US
        link=blue vlink=blue  style='tab-interval:.5in'>
     <br>
    <form action = "/CNT4714/colorsurvey"  method = "post">
    <font size = 6><b><p> WELCOME TO THE CNT 4714 COLOR 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>
```

colorsurvey.html

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

Modified `web.xml` Deployment File

```xml
      <servlet-class>WelcomeServlet3</servlet-class>
  </servlet>
- <servlet>
    <servlet-name>redirect</servlet-name>
    <description>A redirection servlet</description>
    <servlet-class>ReDirectionServlet</servlet-class>
  </servlet>
- <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/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>
    <servlet-name>searcheng</servlet-name>
    <servlet-class>searchEngine</servlet-class>
  </servlet>
- <servlet>
    <servlet-name>cookies</servlet-name>
```

> Portion of the web.xml file showing parameter initializations

SurveyServlet.java

```java
// SurveyServlet.java
// A Web-based survey that uses JDBC from a servlet.

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;
```
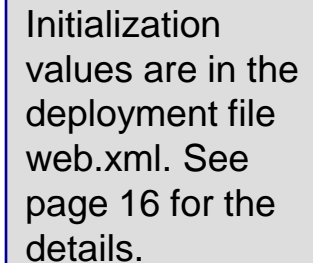
Setup connection to the database

Statement used for updating the vote count for a color after the user makes their choice, totaling the votes and returning the results of the vote.

```
// 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" ) );

        // 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
```

Initialization values are in the deployment file web.xml. See page 16 for the details.

```java
// 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 XHTML document
    out.println( "<?xml version = \"1.0\"?>" );
    out.printf( "%s%s%s", "<!DOCTYPE html PUBLIC",
        " \"-//W3C//DTD XHTML 1.0 Strict//EN\"",
        " \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd\">\n" );
    out.println(
        "<html xmlns = \"http://www.w3.org/1999/xhtml\">" );

    // head section of document
    out.println( "<head>" );

        // read current survey response
    int value =
        Integer.parseInt( request.getParameter( "color" ) );
    String sql;
```

```
// 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( "<title>Thank you!</title>" );
  out.println( "</head>" );

  out.println( "<body>" );
  out.println( "<p>Thank you for participating." );
  out.println( "<br />Results:</p><pre>" );
```

Generate update for the database

Execute SQL Update command

Execute query to return results to client

```
  // process results
   int votes;

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

   resultsRS.close();

   out.print( "Total responses: " );
   out.print( total );

   // end XHTML document
   out.println( "</pre></body></html>" );
   out.close();
} // end try
```

```java
        // 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


    // 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
```
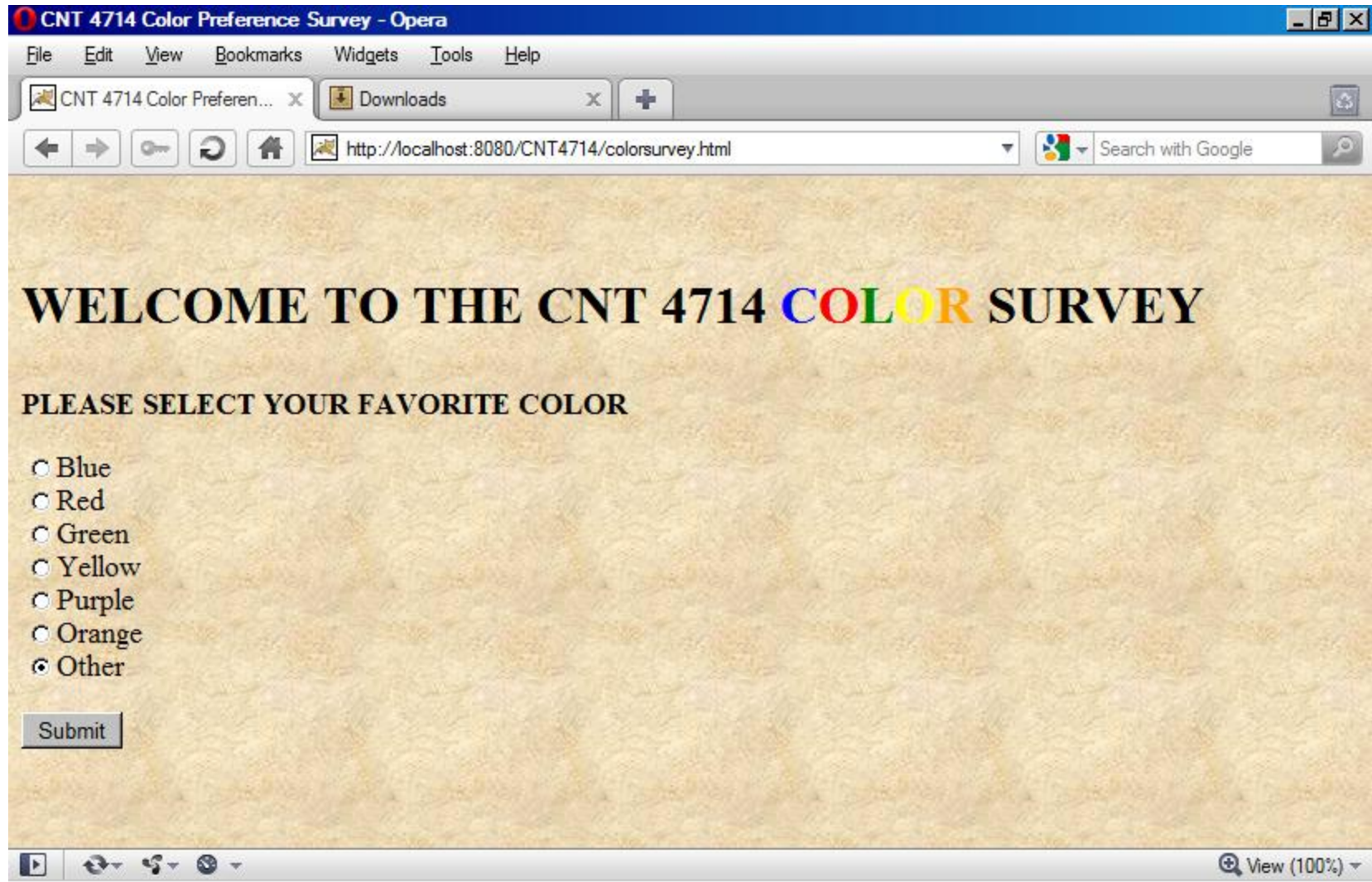
HTML Front-end For ColorSurvey Servlet

Response From ColorSurvey Servlet