Objective: To learn about JSP and implement a CRUD application to create, read, update, and delete data using MySQL database as well as integrating HTML, CSS, and JavaScript in JSP application web.

Resources:
- **JSP**
  - http://www.tutorialspoint.com/jsp/
  - http://www.jsptut.com/
- **MySQL:**
  - https://www.mysql.com/products/workbench/
  - **Basic SQL: https://www.digitalocean.com/community/tutorials/a-basic-mysql-tutorial**

Database: Open MySQL Workbench and run the following command to create a database and a table
1) Create a new database called "moviesdb"

2) Make sure to select database and run the command to create a table called "movies"

```
CREATE TABLE IF NOT EXISTS `movies` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `title` varchar(32) NOT NULL,
 `year` int(4) NOT NULL,
 `poster` varchar(255) NOT NULL,
 `modified` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=52 ;
```

3) Add a few record using the given command below

```
INSERT INTO `movies` (`id`, `title`, `year`, `poster`, `modified`) VALUES
(1, 'John Carter', '2012',
'http://image.tmdb.org/t/p/w185/p4v7ke01C3TVkyIVsQLb7u5jbYV.jpg', '2012-01-11 11:11:11'),
(2, 'Avatar', '2009', 'http://image.tmdb.org/t/p/w185/tcqb9NHdw9SWs2a88KCDD4V8sVR.jpg',
'2009-01-01 18:44:19');
```

Now you have a database with 2 rows of data.

New WebApplication – Open Eclipse
1) Create a new Dynamic Web Project called "moviesApp"
2) Update application web.xml, adding context parameter and values – make sure that the value matches your setup.
…
```xml
  <display-name>moviesApp</display-name>

  <!-- Context initialization parameters -->
  <!-- Provide the database related parameters -->
  <context-param>
     <param-name>jdbcDriver</param-name>
     <param-value>com.mysql.jdbc.Driver</param-value>
  </context-param>
  <context-param>
     <param-name>databaseURL</param-name>
     <param-value>jdbc:mysql://local:3306/moviesdb</param-value>
  </context-param>
```

```
<context-param>
    <param-name>username</param-name>
    <param-value>yourusername</param-value>
</context-param>
<context-param>
    <param-name>password</param-name>
    <param-value>yourpassword</param-value>
</context-param>
```
…

3) Create a new jsp page call, **index.jsp**, under WebContent -- this file will have a basic command to see how can be integrate java code in HTML.

3.a) **Show Current Date and Time**.   In between body tags, but the following code

```
<%
    // this is to start Java code
    java.util.Date date = new java.util.Date();
%>
```

To print the value on the screen, there are 2 ways (make sure to try both)

Way 1: As a part of Java

```
out.println("Current Date and Time is  " + date);
```

Way 2:   Integrate with HTML

```
<p>Current Date and Time is   <%= date %>
```

3.2) **Accessing the Application Parameters** and show on the screen.

To access, you can use `application.getInitParameter( )`   to access the parameter and value that we define in the web.xml file

```
<%
    String username = application.getInitParameter("username");
%>
```

Use the same approach to print out the value. Make sure to have appropriate heading

4) Create a new jsp page call, **read.jsp**, under WebContent -- this file will have a basic command to retrieve movie data from the database and show it on the page

4.a) add JSP import statement at the beginning

```
<%@ page import="java.text.*, java.util.*, java.sql.*, javax.servlet.*,
javax.servlet.http.*"%>
```

4.b) Make sure to get information for your database – url, username, and password

```
<%
    String databaseURL = application.getInitParameter("databaseURL");
    String username = application.getInitParameter("username");
    String password = application.getInitParameter("password");

    // more code will be added here
%>
```

4.c) define additional variable for database connection

```
    Connection conn = null;
    Statement stmt = null;
    com.mysql.jdbc.Driver d = null;
```

4.d) put a try-catch to test the connection to the database and if it works. For now, copy and paste this code. We will go over each section

```
    try {
        d = new com.mysql.jdbc.Driver();
        conn = DriverManager.getConnection(databaseURL, username, password);
        stmt = conn.createStatement();
        String sqlStr = "select id, title, year, poster from movies";
        // System.out.println(sqlStr); // for debugging
        ResultSet rset = stmt.executeQuery(sqlStr);

        out.println("<table border='1'>");

        out.println("<tr><th>Id</th><th>Title</th><th>Year</th><th>Poster</th></tr>");

        while (rset.next()) { // list all the authors
            String sId = rset.getString("id");
            String sTi = rset.getString("title");
            String sYe = rset.getString("year");
            String sPo = rset.getString("poster");
            out.println("<tr><td>"
                        + sId
                        + "</td><td>"
                        + sTi
                        + "</td><td>"
                        + sYe
                        + "</td><td><a href=http://image.tmdb.org/t/p/original/"
                        + sPo + ">poster</a></td></tr>");
        }

        out.println("</table>");

    } catch (SQLException ex) {
        out.println("<h3>Service not available. Please try again later!</h3>");
        out.println("SQLException: " + ex.getMessage());
        out.println("SQLState: " + ex.getSQLState());
        out.println("VendorError: " + ex.getErrorCode());

    } finally {
        out.close();
        try {
            if (stmt != null)
                stmt.close();
            if (conn != null)
                conn.close();
```

```
        } catch (SQLException ex) {
                out.println("SQLException: " + ex.getMessage());
                out.println("SQLState: " + ex.getSQLState());
                out.println("VendorError: " + ex.getErrorCode());

        }
    }
```

4.e) Test your application (show see table like this) and click on each poster link to see what will show on the screen. Make sure to take a few screenshots.

Id	Title	Year	Poster
1	John Carter	2012	poster
2	Avatar	2009	poster

4.f) **Link this index.jsp and read.jsp**.   Go back to index.jsp and create an <a> tag "Read Movie Database" and it should show the list of movies

4.i) Show Image -- **read.jsp**
- Instead of having a poster link and show picture, include the picture using <img> tag, so the page should like something like this

4.j) Format Movie Table Page --   read.jsp
- Add header for this page
- Use CSS file (my.css) to format – color, font, style – so, the information are evenly distributed on the page and not just left most
- Take screenshots before after

4.k) Manually add more data into the database using MySQL Workbench. – add **10 additional movies**. Make sure that the poster URL will work -- tmdb.org

Extra Credit (this will be for next week, but you will get extra credit if you are able to complete it by Friday): Create a new jsp page call, **create.jsp**, under WebContent -- this file will have textbox where you wish to add new movie into the database.

(a) create text field for title, year, and poster with a form post.
(b) create a button for Create, which then save the information into database

SQL command: insert into movies (title, year, poster) values ("movies", "year", "poster");

Hint: You can use String concatenation to create this statement.

(c) to verify that this work, when you go to read.jsp, you should a new movie that you added

Submission
Submission for a temporary and complete assignment: (prefix each file with your lastname_M8) – First version by the end of class, and complete version **Due by Friday, April 3, 11:59 PM**
   a. Zip file of the source code
   b. Word Document for Screenshot