

# CS 122B Project 5 Report

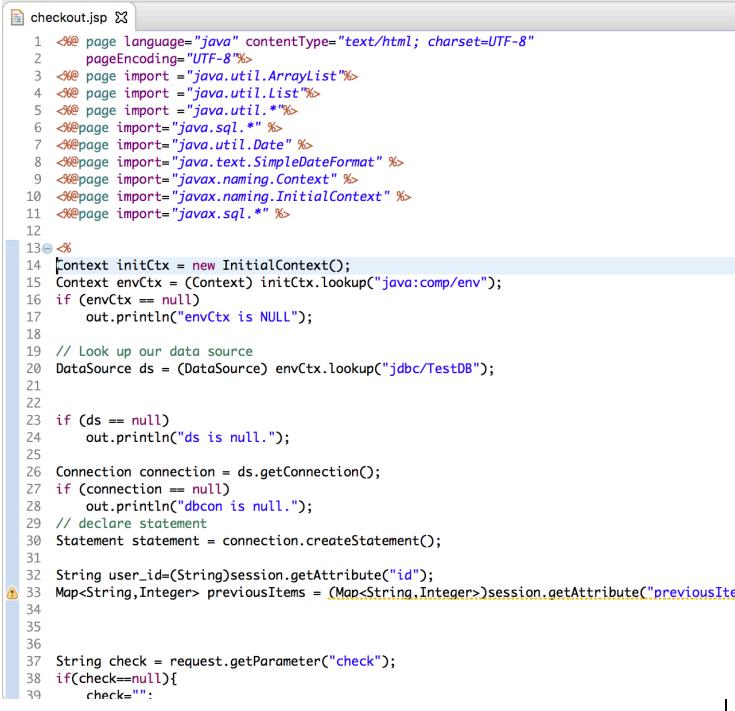
Team-88

Ye Yuan 49889946 | Lingzhe Kong 93985614

## Task 1:

### 1. Connection Polling:

File Name	Line Number	Snapshot
_dashboard.jsp	23	<pre>_dashboard.jsp &lt;input type="checkbox" checked="" checked="checked" value="1" /&gt; 3 &lt;%@ page import = "java.util.ArrayList" %&gt; 4 &lt;%@ page import = "java.util.List" %&gt; 5 &lt;%@ page import = "java.util.*" %&gt; 6 &lt;%@page import="java.sql.*" %&gt; 7 8 &lt;%@page import="java.util.Date" %&gt; 9 &lt;%@page import="java.text.SimpleDateFormat" %&gt; 10 &lt;%@page import="javax.naming.Context" %&gt; 11 &lt;%@page import="javax.naming.InitialContext" %&gt; 12 &lt;%@page import="javax.sql.*" %&gt; 13&lt;%&gt; 14     HttpServletRequest httpRequest = (HttpServletRequest) request; 15     HttpServletResponse httpResponse = (HttpServletResponse) response; 16     if (httpRequest.getSession().getAttribute("isEmployeeLogin") == null) { 17         System.out.println("check here"); 18         httpResponse.sendRedirect("_dashboard.html"); 19     } 20 21 22 23       Context initCtx = new InitialContext(); 24       Context envCtx = (Context) initCtx.lookup("java:comp/env"); 25       if (envCtx == null) 26           out.println("envCtx is NULL"); 27 28 // Look up our data source 29 DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 30 31 32 if (ds == null) 33     out.println("ds is null."); 34 35 Connection connection = ds.getConnection(); 36 if (connection == null) 37     out.println("dbcon is null."); 38 39</pre>

		 A screenshot of a Java Server Page (JSP) code editor showing the file 'BrowsePage.jsp'. The code uses JSTL and Java imports to interact with a database. It includes imports for java.sql, java.util.ArrayList, java.util.List, javax.naming, and javax.sql. The page sets its language to Java, content type to text/html, and character encoding to UTF-8. It retrieves an HttpServletRequest and HttpServletResponse, initializes contexts, looks up a data source, and executes a query to select all genres from the genres table. A warning icon is visible near the bottom of the code.
BrowsePage.jsp	16	 A screenshot of a Java Server Page (JSP) code editor showing the file 'checkout.jsp'. The code uses JSTL and Java imports to interact with a database. It includes imports for java.util.ArrayList, java.util.List, java.util, java.sql, java.util.Date, java.text.SimpleDateFormat, javax.naming, and javax.sql. The page sets its language to Java, content type to text/html, and character encoding to UTF-8. It retrieves an HttpServletRequest and HttpServletResponse, initializes contexts, looks up a data source, and executes a query to select all genres from the genres table. A warning icon is visible near the bottom of the code.

confirmation.jsp	13	<pre> confirmation.jsp %% 1 &lt;%@ page language="java" contentType="text/html; charset=UTF-8" 2   pageEncoding="UTF-8"%&gt; 3 &lt;%@page import ="java.util.ArrayList"%&gt; 4 &lt;%@page import ="java.util.List"%&gt; 5 &lt;%@page import ="java.util.*"%&gt; 6 &lt;%@page import="java.sql.*" %&gt; 7 &lt;%@page import="java.util.Date" %&gt; 8 &lt;%@page import="java.text.SimpleDateFormat" %&gt; 9 &lt;%@page import="javax.naming.Context" %&gt; 10 &lt;%@page import="javax.naming.InitialContext" %&gt; 11 &lt;%@page import="javax.sql.*" %&gt; 12 @@ 13 Context initCtx = new InitialContext(); 14 Context envCtx = (Context) initCtx.lookup("java:comp/env"); 15 if (envCtx == null) 16   out.println("envCtx is NULL"); 17 18 // Look up our data source 19 DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 20 21 22 if (ds == null) 23   out.println("ds is null."); 24 25 Connection connection = ds.getConnection(); 26 if (connection == null) 27   out.println("dbcon is null."); 28 29 String user_id=(String)session.getAttribute("id"); 30 Date date = new Date(); 31 SimpleDateFormat dateFormat= new SimpleDateFormat("yyyy/MM/dd"); 32 33 String query="select s.id ,m.title from sales as s , movies as m " 34   + "where s.saleDate=? and s.customerId=? and s.movieId=m.id"; 35 PreparedStatement preparedStatement =connection.prepareStatement(query); 36 preparedStatement.setString(1,dateFormat.format(date)); 37 preparedStatement.setString(2,user_id); 38 39 ResultSet salesResult = preparedStatement.executeQuery(); </pre>
insertMovie.jsp	17	<pre> insertMovie.jsp %% 1 &lt;%@ page language="java" contentType="text/html; charset=UTF-8" 2   pageEncoding="UTF-8"%&gt; 3 &lt;%@page import="java.sql.*" %&gt; 4 &lt;%@ page import ="java.util.ArrayList"%&gt; 5 &lt;%@page import ="java.util.List"%&gt; 6 &lt;%@ page import ="java.util.*"%&gt; 7 &lt;%@page import="javax.naming.Context" %&gt; 8 &lt;%@page import="javax.naming.InitialContext" %&gt; 9 &lt;%@page import="javax.sql.*" %&gt; 10 @@ 11 HttpServletRequest httpRequest = (HttpServletRequest) request; 12 HttpServletResponse httpResponse = (HttpServletResponse) response; 13 if (httpRequest.getSession().getAttribute("isEmployeeLogin") == null) { 14   System.out.println("check here"); 15   httpResponse.sendRedirect("_dashboard.html"); 16 } 17 Context initCtx = new InitialContext(); 18 Context envCtx = (Context) initCtx.lookup("java:comp/env"); 19 if (envCtx == null) 20   out.println("envCtx is NULL"); 21 22 // Look up our data source 23 DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 24 25 26 if (ds == null) 27   out.println("ds is null."); 28 29 Connection connection = ds.getConnection(); 30 if (connection == null) 31   out.println("dbcon is null."); 32 33 String movie_name=request.getParameter("name"); 34 String movie_director=request.getParameter("director"); 35 String movie_year=request.getParameter("year"); 36 String movie_genre=request.getParameter("genre"); 37 String movie_star=request.getParameter("star"); 38 39 String checkExistQuery="select * from movies where title=? and directo </pre>

insertStar.jsp	12	<pre> 1 &lt;%@ page language="java" contentType="text/html; charset=UTF-8" 2 pageEncoding="UTF-8"%&gt; 3 &lt;%@ page import ="java.util.ArrayList"%&gt; 4 &lt;%@ page import ="java.util.List"%&gt; 5 &lt;%@ page import ="java.util.*"%&gt; 6 &lt;%@page import="java.sql.*" %&gt; 7 &lt;%@page import="javax.naming.Context" %&gt; 8 &lt;%@page import="javax.naming.InitialContext" %&gt; 9 &lt;%@page import="javax.sql.*" %&gt; 10 11&lt; 12 Context initCtx = new InitialContext(); 13 Context envCtx = (Context) initCtx.lookup("java:comp/env"); 14 if (envCtx == null) 15     out.println("envCtx is NULL"); 16 17 // Look up our data source 18 DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 19 20 21 if (ds == null) 22     out.println("ds is null."); 23 24 Connection connection = ds.getConnection(); 25 if (connection == null) 26     out.println("dbcon is null."); 27 // declare statement 28 String star_name=request.getParameter("name"); 29 String star_year=request.getParameter("year"); 30 if(star_year.equals("")){ 31     star_year=null; 32 } 33 String maxIDquery="select max(id) as id from stars"; 34 System.out.println(star_year); 35 PreparedStatement preparedStatement =connection.prepareStatement(maxIDquery); 36 ResultSet rs=preparedStatement.executeQuery(); 37 String maxid=""; 38 while(rs.next()){ 39     maxid = rs.getString("id");         </pre>
MoviePage.jsp	14	<pre> 1 &lt;%@page import="java.sql.*" %&gt; 2 &lt;%@ page import ="java.util.ArrayList"%&gt; 3 &lt;%@ page import ="java.util.List"%&gt; 4 &lt;%@ page language="java" contentType="text/html; charset=UTF-8" 5 pageEncoding="UTF-8"%&gt; 6 &lt;%@page import="javax.naming.Context" %&gt; 7 &lt;%@page import="javax.naming.InitialContext" %&gt; 8 &lt;%@page import="javax.sql.*" %&gt; 9&lt; 10     HttpServletRequest httpRequest = (HttpServletRequest) request; 11     HttpServletResponse httpResponse = (HttpServletResponse) response; 12 13 14     Context initCtx = new InitialContext(); 15     Context envCtx = (Context) initCtx.lookup("java:comp/env"); 16     if (envCtx == null) 17         out.println("envCtx is NULL"); 18 19 // Look up our data source 20 DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 21 22 23 if (ds == null) 24     out.println("ds is null."); 25 26 Connection connection = ds.getConnection(); 27 if (connection == null) 28     out.println("dbcon is null."); 29 30 String movieId=request.getParameter("movieId"); 31 32 System.out.println("Now movieId is:"+movieId); 33 34 String movieQuery="select m.title,m.year,m.director from movies as m where m.i 35 PreparedStatement moviesStatement=connection.prepareStatement(movieQuery); 36 moviesStatement.setString(1,movieId); 37 38 ResultSet movieResult = moviesStatement.executeQuery(); 39 String movie title="":         </pre>

search.jsp

11

```
1 <%@page import="java.sql.*" %>
2 <%@ page import = "java.util.ArrayList"%>
3 <%@ page import = "java.util.List"%>
4 <%@ page language="java" contentType="text/html; charset=UTF-8"
5    pageEncoding="UTF-8"%>
6 <%@page import="javax.naming.Context" %>
7 <%@page import="javax.naming.InitialContext" %>
8 <%@page import="javax.sql.*" %>
9 <%-- these statements are just normal Java code, they need to be inside the <% %> bracket
10<%>
11 Context initCtx = new InitialContext();
12 Context envCtx = (Context) initCtx.lookup("java:comp/env");
13 if (envCtx == null)
14     out.println("envCtx is NULL");
15
16 // Look up our data source
17 DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB");
18
19
20 if (ds == null)
21     out.println("ds is null.");
22
23 Connection connection = ds.getConnection();
24 if (connection == null)
25     out.println("dbcon is null.");
26
27
28 String title=request.getParameter("title");
29 String year=request.getParameter("year");
30 String director=request.getParameter("director");
31 String star_name=request.getParameter("star_name");
32 String browse_type=request.getParameter("browse_type");
33 String browse_genre=request.getParameter("browse_genre");
34 int number_per_page=Integer.parseInt(request.getParameter("number_per_page"));
35 int start_from=Integer.parseInt(request.getParameter("start_from"));
36 String sorted_by=request.getParameter("sorted_by");
37
38 System.out.println("Now sorted_by is:"+sorted_by);
39 if(sorted_hv==null)f
```

shoppingcart.jsp

11

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2    pageEncoding="UTF-8"%>
3 <%@ page import = "java.util.ArrayList"%>
4 <%@ page import = "java.util.List"%>
5 <%@ page import = "java.util.*"%>
6 <%@page import="java.sql.*" %>
7 <%@page import="javax.naming.Context" %>
8 <%@page import="javax.naming.InitialContext" %>
9 <%@page import="javax.sql.*" %>
10<%>
11 Context initCtx = new InitialContext();
12 Context envCtx = (Context) initCtx.lookup("java:comp/env");
13 if (envCtx == null)
14     out.println("envCtx is NULL");
15
16 // Look up our data source
17 DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB");
18
19
20 if (ds == null)
21     out.println("ds is null.");
22
23 Connection connection = ds.getConnection();
24 if (connection == null)
25     out.println("dbcon is null.");
26
27 String user_id=(String)session.getAttribute("id");
28 Map<String, Integer> previousItems = (Map<String, Integer>)session.getAttribute("previousI");
29 if (previousItems == null) {
30     previousItems = new HashMap();
31     session.setAttribute("previousItems", previousItems);
32 }
33
34 String movieId = request.getParameter("movieId");
35 if(movieId!=null){
36     String q=request.getParameter("q");
37     if(q.equals("233")){
38         q="1";
39     }
```

StarPage.jsp	13	<pre> 1 &lt;%@page import="java.sql.*" %&gt; 2 &lt;%@ page import ="java.util.ArrayList"%&gt; 3 &lt;%@ page import ="java.util.List"%&gt; 4 &lt;%@ page language="java" contentType="text/html; charset=UTF-8" 5 pageEncoding="UTF-8"%&gt; 6 &lt;%@page import="javax.naming.Context" %&gt; 7 &lt;%@page import="javax.naming.InitialContext" %&gt; 8 &lt;%@page import="javax.sql.*" %&gt; 9 &lt;%&gt; 10 HttpServletRequest httpRequest = (HttpServletRequest) request; 11 HttpServletResponse httpResponse = (HttpServletResponse) response; 12 13 Context initCtx = new InitialContext(); 14 Context envCtx = (Context) initCtx.lookup("java:comp/env"); 15 if (envCtx == null) 16     out.println("envCtx is NULL"); 17 18 // Look up our data source 19 DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 20 21 22 if (ds == null) 23     out.println("ds is null."); 24 25 Connection connection = ds.getConnection(); 26 if (connection == null) 27     out.println("dbcon is null."); 28 29 String star_name=request.getParameter("star_name"); 30 31 32 33 String starQuery="select name,birthYear from stars as s where s.name= 34 PreparedStatement starStatement= connection.prepareStatement(starQuer 35 starStatement.setString(1,star_name); 36 ResultSet starResult = starStatement.executeQuery(); 37 38 String star_year=""; </pre>
AndroidLogin.java	39	<pre> 1 *AndroidLogin.java * 2 3 import com.google.gson.JsonObject; 4 /** 5  * @WebServlet(name = "AndroidLogin", urlPatterns = "/api/android_login") 6  public class AndroidLogin extends HttpServlet { 7     private static final long serialVersionUID = 1L; 8 9 10 /** 11  * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse res 12 */ 13 protected void doPost(HttpServletRequest request, HttpServletResponse respons 14     PrintWriter out = response.getWriter(); 15 16     response.setContentType("text/html"); 17 18 19     try { 20         Context initCtx = new InitialContext(); 21         Context envCtx = (Context) initCtx.lookup("java:comp/env"); 22         if (envCtx == null) 23             out.println("envCtx is NULL"); 24 25         // Look up our data source 26         DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 27 28 29         if (ds == null) 30             out.println("ds is null."); 31 32         Connection connection = ds.getConnection(); 33         if (connection == null) 34             out.println("dbcon is null."); 35 36 37         String password=request.getParameter("password"); 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 </pre>

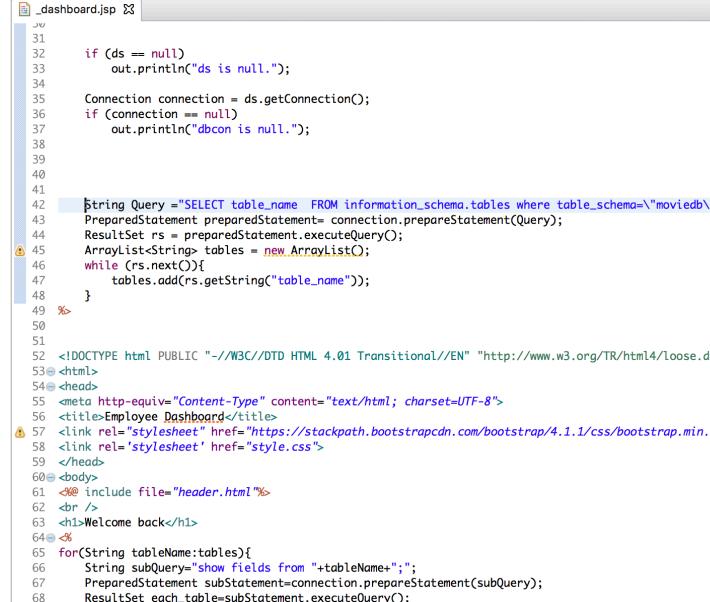
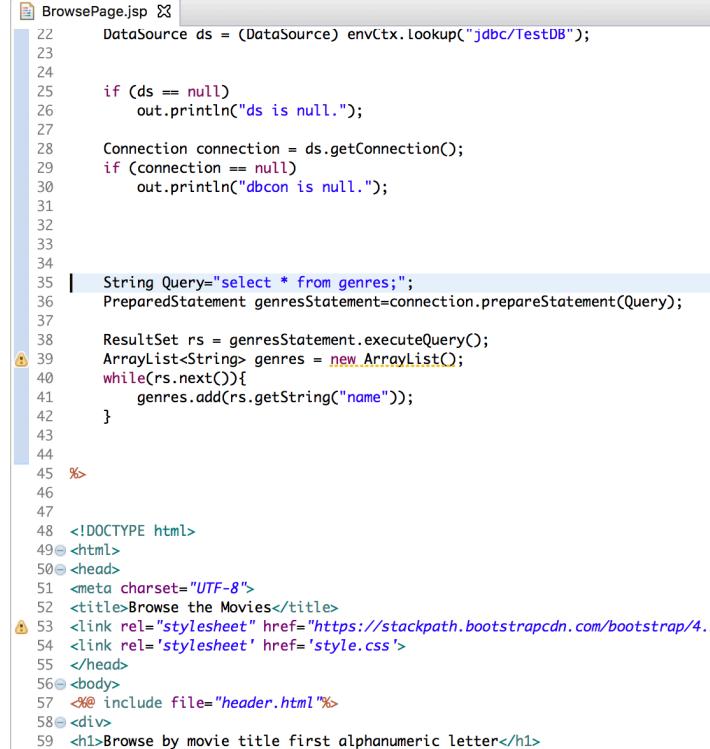
AndroidSearch.java	45	<pre> 3/ 38     // Output stream to STDOUT 39     PrintWriter out = response.getWriter(); 40 41 42     try 43     { 44         Context initCtx = new InitialContext(); 45         Context envCtx = (Context) initCtx.lookup("java:comp/env"); 46         if (envCtx == null) 47             out.println("envCtx is NULL"); 48 49         // Look up our data source 50         DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 51 52 53         if (ds == null) 54             out.println("ds is null."); 55 56         Connection connection = ds.getConnection(); 57         if (connection == null) 58             out.println("dbcon is null."); 59         // Declare our statement 60 61 62 63         String input = request.getParameter("search"); 64         System.out.println("user input " + input); 65 66 67         String query="select id, title, year,director from movies where 68 69         PreparedStatement preparedStatement=connection.prepareStatement(); 70         preparedStatement.setString(1,"%"+input+"%"); 71 72         // Perform the query 73         ResultSet rs = preparedStatement.executeQuery(); 74         //ResultSet rs = statement.executeQuery(query);     </pre>
AutoComplete.java	64	<pre> 49     * The format is like this because it can be directly used by the 50     * JSON auto complete library this example is using. So that you don't 51     * have to parse the JSON response. 52     * The response contains a list of suggestions. 53     * In each suggestion object, the "value" is the item string shown in 54     * the "data" object can contain any additional information. 55     * 56     */ 57 58     protected void doGet(HttpServletRequest request, HttpServletResponse response) 59     { 60         PrintWriter out = response.getWriter(); 61 62         try { 63 64             Context initCtx = new InitialContext(); 65             Context envCtx = (Context) initCtx.lookup("java:comp/env"); 66             if (envCtx == null) 67                 out.println("envCtx is NULL"); 68 69             // Look up our data source 70             DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 71 72 73             if (ds == null) 74                 out.println("ds is null."); 75 76             Connection connection = ds.getConnection(); 77             if (connection == null) 78                 out.println("dbcon is null."); 79 80 81             // setup the response json array 82             JSONArray jsonArray = new JSONArray(); 83 84             // get the query string from parameter 85             String query = request.getParameter("query"); 86 87             // return the empty json array if query is null or empty     </pre>

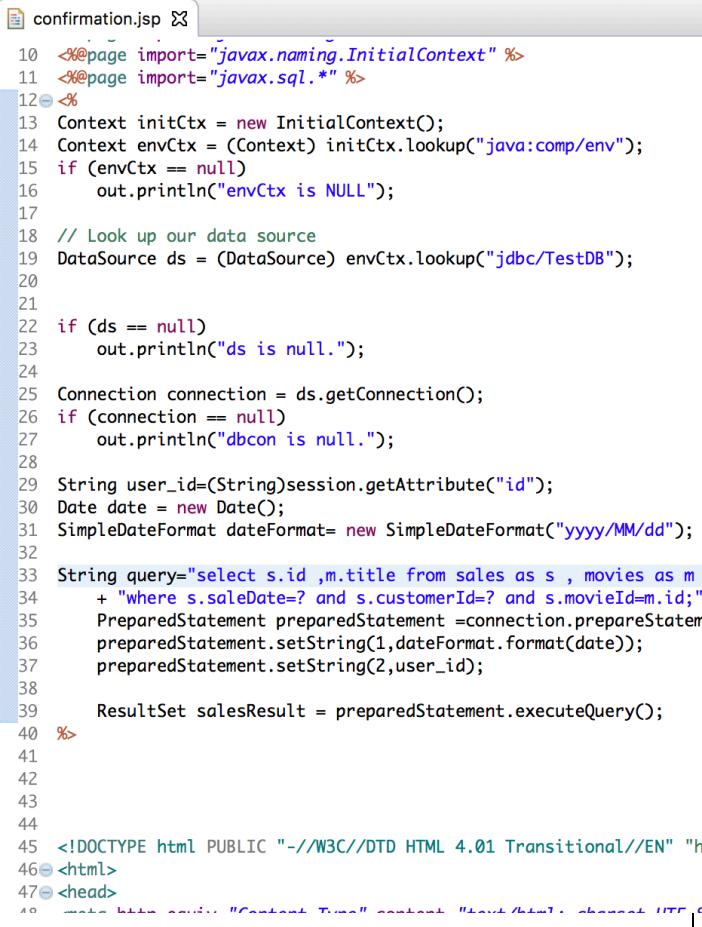
EmployeeLoginServlet.java	55	<pre> 37     try { 38         RecaptchaVerifyUtils.verify(gRecaptchaResponse); 39     } catch (Exception e) { 40         out.println("&lt;html&gt;"); 41         out.println("&lt;head&gt;&lt;title&gt;Error&lt;/title&gt;&lt;/head&gt;"); 42         out.println("&lt;body&gt;"); 43         out.println("&lt;p&gt;recaptcha verification error&lt;/p&gt;"); 44         out.println("&lt;p&gt;" + e.getMessage() + "&lt;/p&gt;"); 45         out.println("&lt;/body&gt;"); 46         out.println("&lt;/html&gt;"); 47 48         out.close(); 49         return; 50     } 51 52 53 54     try { 55         Context initCtx = new InitialContext(); 56         Context envCtx = (Context) initCtx.lookup("java:comp/env"); 57         if (envCtx == null) 58             out.println("envCtx is NULL"); 59 60         // Look up our data source 61         DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 62 63 64         if (ds == null) 65             out.println("ds is null."); 66 67         Connection connection = ds.getConnection(); 68         if (connection == null) 69             out.println("dbcon is null."); 70 71 72         String password=request.getParameter("password"); 73         String user=request.getParameter("user_email"); 74 75         PasswordEncryptor passwordEncryptor = new StrongPasswordEncryptor();     </pre>
LoginServlet.java	55	<pre> 44         out.println("&lt;p&gt;" + e.getMessage() + "&lt;/p&gt;"); 45         out.println("&lt;/body&gt;"); 46         out.println("&lt;/html&gt;"); 47 48         out.close(); 49         return; 50     } 51 52 53 54     try { 55         Context initCtx = new InitialContext(); 56         Context envCtx = (Context) initCtx.lookup("java:comp/env"); 57         if (envCtx == null) 58             out.println("envCtx is NULL"); 59 60         // Look up our data source 61         DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 62 63 64         if (ds == null) 65             out.println("ds is null."); 66 67         Connection connection = ds.getConnection(); 68         if (connection == null) 69             out.println("dbcon is null."); 70 71 72 73         String password=request.getParameter("password"); 74         String user=request.getParameter("user_email"); 75 76         PasswordEncryptor passwordEncryptor = new StrongPasswordEncryptor(); 77         String query = "SELECT * from customers where email= ?;"; 78         PreparedStatement preparedStatement =connection.prepareStatement(query); 79 80         preparedStatement.setString(1, user); 81         System.out.println("the query is: "+preparedStatement); 82         ResultSet rs = preparedStatement.executeQuery();     </pre>

### **How connection polling was implemented:**

We removed the codes of username, password and URL, as well as the code that we used to make JDBC connection with. Then we implemented the codes from the given example, which will get information about JDBC from “META-INF/content.xml” .

2. Prepared statements:

File Name	Line Number	Snapshot
<code>_dashboard.jsp</code>	42	 <pre> 31 32     if (ds == null) 33         out.println("ds is null."); 34 35     Connection connection = ds.getConnection(); 36     if (connection == null) 37         out.println("dbcon is null."); 38 39 40 41 42     String Query ="SELECT table_name FROM information_schema.tables where table_schema=\\"moviedb\\"; 43     PreparedStatement preparedStatement= connection.prepareStatement(Query); 44     ResultSet rs = preparedStatement.executeQuery(); 45     ArrayList&lt;String&gt; tables = new ArrayList(); 46     while (rs.next()){ 47         tables.add(rs.getString("table_name")); 48     } 49 % 50 51 52     &lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"&gt; 53     &lt;html&gt; 54         &lt;head&gt; 55             &lt;meta http-equiv="Content-Type" content="text/html; charset=UTF-8"&gt; 56             &lt;title&gt;Employee Dashboard&lt;/title&gt; 57             &lt;link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"&gt; 58             &lt;link rel='stylesheet' href="style.css"&gt; 59         &lt;/head&gt; 60         &lt;body&gt; 61             &lt;%@ include file="header.html"%&gt; 62             &lt;br /&gt; 63             &lt;h1&gt;Welcome back&lt;/h1&gt; 64 % 65             for(String tableName:tables){ 66                 String subQuery="show fields from "+tableName+";"; 67                 PreparedStatement subStatement=connection.prepareStatement(subQuery); 68                 ResultSet each_table=subStatement.executeQuery(); </pre>
<code>BrowsePage.jsp</code>	35	 <pre> 22     DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 23 24 25     if (ds == null) 26         out.println("ds is null."); 27 28     Connection connection = ds.getConnection(); 29     if (connection == null) 30         out.println("dbcon is null."); 31 32 33 34 35     String Query="select * from genres;" ; 36     PreparedStatement genresStatement=connection.prepareStatement(Query); 37 38     ResultSet rs = genresStatement.executeQuery(); 39     ArrayList&lt;String&gt; genres = new ArrayList(); 40     while(rs.next()){ 41         genres.add(rs.getString("name")); 42     } 43 44 45 % 46 47 48     &lt;!DOCTYPE html&gt; 49     &lt;html&gt; 50         &lt;head&gt; 51             &lt;meta charset="UTF-8"&gt; 52             &lt;title&gt;Browse the Movies&lt;/title&gt; 53             &lt;link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"&gt; 54             &lt;link rel='stylesheet' href='style.css'&gt; 55         &lt;/head&gt; 56         &lt;body&gt; 57             &lt;%@ include file="header.html"%&gt; 58             &lt;div&gt; 59                 &lt;h1&gt;Browse by movie title first alphanumeric letter&lt;/h1&gt; 60                 &lt;a href="search.jsp?title=a&amp;browse_type=a&amp;start_from=1&amp;number_per_page=20"&gt;</pre>

<b>checkout.jsp</b>  47	 <pre> 25 26 Connection connection = ds.getConnection(); 27 if (connection == null) 28   out.println("dbcon is null."); 29 // declare statement 30 Statement statement = connection.createStatement(); 31 32 String user_id=(String)session.getAttribute("id"); 33 Map&lt;String,Integer&gt; previousItems = (Map&lt;String,Integer&gt;)session.getAttribute("previousItems"); 34 35 36 37 String check = request.getParameter("check"); 38 if(check==null){ 39   check=""; 40 }else{ 41   String firstname = request.getParameter("firstname"); 42   String lastname = request.getParameter("lastname"); 43   String cardnumber = request.getParameter("cardnumber"); 44   String expiredate = request.getParameter("expiredate"); 45 46   String query="select id from creditcards where firstName=? and lastName=? and id=? and expiration=?"; 47   PreparedStatement preparedStatement =connection.prepareStatement(query); 48 49   preparedStatement.setString(1,firstname ); 50   preparedStatement.setString(2,lastname ); 51   preparedStatement.setString(3,cardnumber); 52   preparedStatement.setString(4,expiredate ); 53 54   //System.out.println("The user email is: "); 55   ResultSet resultSet = preparedStatement.executeQuery(); 56 57 58 59 60   String card_number="-1"; 61   if(resultSet.next()) { 62     card_number= resultSet.getString("id"); 63   } </pre>
<b>confirmation.jsp</b>  33	 <pre> 10 &lt;%@page import="javax.naming.InitialContext" %&gt; 11 &lt;%@page import="javax.sql.*" %&gt; 12 &lt;% 13 Context initCtx = new InitialContext(); 14 Context envCtx = (Context) initCtx.lookup("java:comp/env"); 15 if (envCtx == null) 16   out.println("envCtx is NULL"); 17 18 // Look up our data source 19 DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 20 21 22 if (ds == null) 23   out.println("ds is null."); 24 25 Connection connection = ds.getConnection(); 26 if (connection == null) 27   out.println("dbcon is null."); 28 29 String user_id=(String)session.getAttribute("id"); 30 Date date = new Date(); 31 SimpleDateFormat dateFormat= new SimpleDateFormat("yyyy/MM/dd"); 32 33 String query="select s.id ,m.title from sales as s , movies as m 34   + "where s.saleDate=? and s.customerId=? and s.movieId=m.id;"" 35   PreparedStatement preparedStatement =connection.prepareStatement(query); 36   preparedStatement.setString(1,dateFormat.format(date)); 37   preparedStatement.setString(2,user_id); 38 39   ResultSet salesResult = preparedStatement.executeQuery(); 40 %&gt; 41 42 43 44 45 &lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "h 46 &lt;html&gt; 47 &lt;head&gt; 48   meta http-equiv="Content-Type" content="text/html; charset=UTF-8" </pre>

insertMovie.jsp	39	<pre> 19  if (envCtx == null) 20      out.println("envCtx is NULL"); 21 22 // Look up our data source 23 DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 24 25 26 if (ds == null) 27     out.println("ds is null."); 28 29 Connection connection = ds.getConnection(); 30 if (connection == null) 31     out.println("dbcon is null."); 32 33 String movie_name=request.getParameter("name"); 34 String movie_director=request.getParameter("director"); 35 String movie_year=request.getParameter("year"); 36 String movie_genre=request.getParameter("genre"); 37 String movie_star=request.getParameter("star"); 38 39 String check_exist_query="select * from movies where title=? and director=? and year=?"; 40 PreparedStatement preparedStatement=connection.prepareStatement(check_exist_query); 41 preparedStatement.setString(1,movie_name); 42 preparedStatement.setString(2,movie_director); 43 preparedStatement.setString(3,movie_year); 44 45 ResultSet rs=preparedStatement.executeQuery(); 46 47 %&gt; 48 49 50 &lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/1 51@&lt;html&gt; 52@&lt;head&gt; 53 &lt;meta http-equiv="Content-Type" content="text/html; charset=UTF-8"&gt; 54 &lt;title&gt;Insert New Movie&lt;/title&gt; 55 &lt;link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap 56 &lt;link rel='stylesheet' href="style.css"&gt; 57 &lt;/head&gt;</pre>
insertStar.jsp	43	<pre> 22     out.println("ds is null."); 23 24 Connection connection = ds.getConnection(); 25 if (connection == null) 26     out.println("dbcon is null."); 27 // declare statement 28 String star_name=request.getParameter("name"); 29 String star_year=request.getParameter("year"); 30 if(star_year.equals("")){ 31     star_year=null; 32 } 33 String maxIDquery="select max(id) as id from stars;"; 34 System.out.println(star_year); 35 PreparedStatement preparedStatement =connection.prepareStatement(maxIDquery); 36 ResultSet rs=preparedStatement.executeQuery(); 37 String maxid=""; 38 while(rs.next()){ 39     maxid = rs.getString("id"); 40     maxid = "nm" + (Integer.parseInt(maxid.substring(2,maxid.length()))+1); 41 } 42 43 String insertQuery="INSERT INTO stars(id,name,birthYear) VALUES(?, ?, ?);" 44 PreparedStatement insertStatement =connection.prepareStatement(insertQuery); 45 insertStatement.setString(1,maxid); 46 insertStatement.setString(2,star_name); 47 if(star_year==null){ 48     insertStatement.setString(3,star_year); 49 }else{ 50     insertStatement.setInt(3,Integer.parseInt(star_year)); 51 } 52 int result=insertStatement.executeUpdate(); 53 54 55 %&gt; 56 57 &lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://w 58@ &lt;html&gt; 59@ &lt;head&gt;</pre>

MoviePage.jsp

34

```
21
22
23     if (ds == null)
24         out.println("ds is null.");
25
26     Connection connection = ds.getConnection();
27     if (connection == null)
28         out.println("dbcon is null.");
29
30     String movieId=request.getParameter("movieId");
31
32     System.out.println("Now movieId is:"+movieId);
33
34     String movieQuery="select m.title,m.year,m.director from movies as m where m.id=?";
35     PreparedStatement moviesStatement=connection.prepareStatement(movieQuery);
36     moviesStatement.setString(1,movieId);
37
38     ResultSet movieResult = moviesStatement.executeQuery();
39     String movie_title="";
40     String movie_year="";
41     String movie_director="";
42     String movie_rating="";
43     while(movieResult.next()){
44         movie_title=movieResult.getString("title");
45         movie_year=movieResult.getString("year");
46         movie_director=movieResult.getString("director");
47     }
48
49     String ratingQuery="select r.rating from movies as m, ratings as r where m.id=r.movieId and m.id=?";
50     PreparedStatement ratingStatement=connection.prepareStatement(ratingQuery);
51     ratingStatement.setString(1,movieId);
52     ResultSet ratingResult = ratingStatement.executeQuery();
53     while(ratingResult.next()){
54         movie_rating=ratingResult.getString("rating");
55     }
56     if(movie_rating.equals ""){
57         movie_rating="null";
58     }
59
```

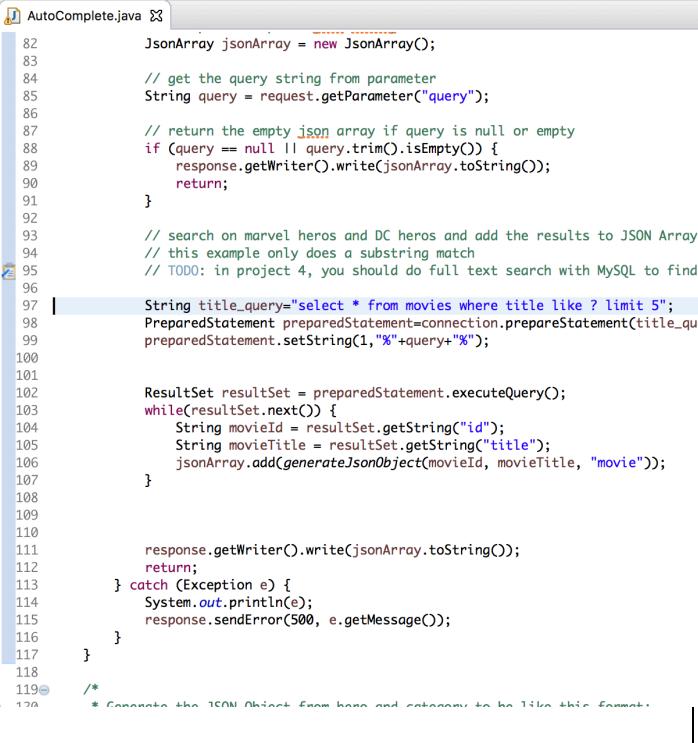
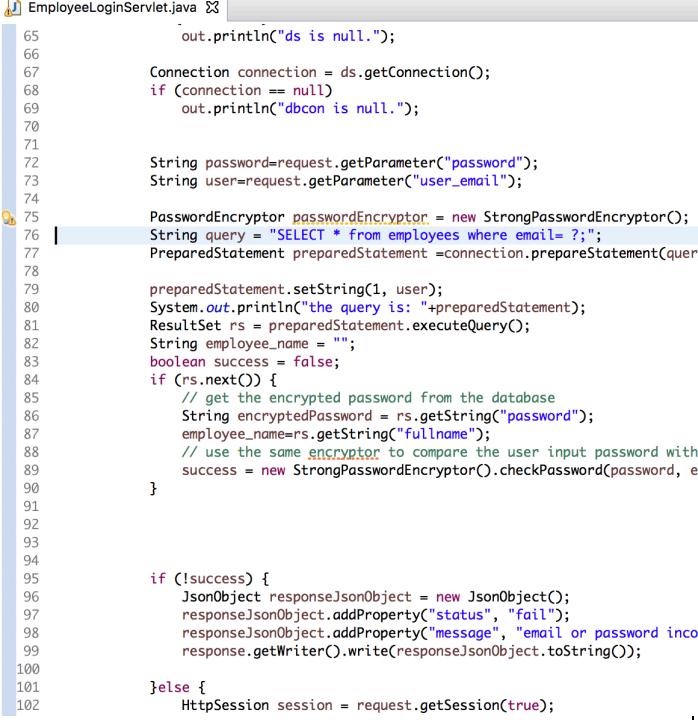
search.jsp

67

```
66     String query="empty";
67     | PreparedStatement preparedStatement=connection.prepareStatement(query);
68     | if(browse_type==null || browse_type.equals("")){
69     | /**
70     | query="select m.id from movies as m,
71     | + "(select distinct sm.movieId from stars_in_movies as sm, stars as s "
72     | + "where s.name like \'%" +star_name+ "%\' and s.id=sm.starId) as nm, rat-
73     | + "ing as r
74     | + "where m.title LIKE \'%" +title+ "%\' "
75     | + "and m.year Like \'%" +year+ "%\' "
76     | + "and m.director Like \'%" +director+ "%\' "
77     | + "and nm.movieid=m.id and r.movieId= m.id "
78     | + "order by "+sorted_by+
79     | + "limit "+number_per_page+
80     | + "offset "+(start_from-1)*number_per_page+";"
81     |
82     | query="select m.id from (select m.id,m.title from movies as m,
83     | + "(select distinct sm.movieId from stars_in_movies as sm, stars as s "
84     | + "where s.name like ? and s.id=sm.starId) as nm "
85     | + "where m.title LIKE ? "
86     | + "and m.year Like ? "
87     | + "and m.director Like ? "
88     | + "and nm.movieid=m.id) as m left join ratings as r on r.movieId=m.id "
89     | + "order by "+sorted_by+
90     | + "limit ? "
91     | + "offset ?;";
92     |
93     | preparedStatement=connection.prepareStatement(query);
94     | preparedStatement.setString(1,"%"+star_name+"%");
95     | preparedStatement.setString(2,"%"+title+"%");
96     | preparedStatement.setString(3,"%"+year+"%");
97     | preparedStatement.setString(4,"%"+director+"% ");
98     //preparedStatement.setString(5,sorted_by);
99     preparedStatement.setInt(5,number_per_page);
100    preparedStatement.setInt(6,(start_from-1)*number_per_page);
101
102    }else if(browse_type.equals("a")){
103
```

shoppingcart.jsp	81	<pre> 63 &lt;link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/ 64 &lt;link rel='stylesheet' href='style.css'&gt; 65 &lt;/head&gt; 66&lt;body&gt; 67 &lt;%@ include file="header.html"%&gt; 68 &lt;br /&gt; 69 &lt;div&gt; 70&lt;table border&gt; 71 &lt;tr&gt;&lt;th colspan="3"&gt;Shopping Cart&lt;/th&gt;&lt;/tr&gt; 72 &lt;td&gt;Movie Name&lt;/td&gt;&lt;td&gt;Quantity&lt;/td&gt;&lt;td&gt;Modify the Quantity&lt;/td&gt;&lt;/tr&gt; 73&lt;br&gt; 74 int exist_item=0; 75 for(String m_id:previousItems.keySet()){ 76     int quantity=previousItems.get(m_id); 77 78     if(quantity&gt;0){ 79         exist_item++; 80         String movie_title=""; 81         String movieQuery="select title from movies where id=?"; 82         PreparedStatement preparedStatement = connection.prepareStatement(movieQuery); 83         preparedStatement.setString(1, m_id); 84         ResultSet movieResult = preparedStatement.executeQuery(); 85         while(movieResult.next()){ 86             movie_title=movieResult.getString("title"); 87         }%&gt; 88&lt;tr&gt; 89 &lt;td&gt;&lt;%=movie_title %&gt;&lt;/td&gt;&lt;td&gt;&lt;%=quantity%&gt;&lt;/td&gt; 90&lt;td&gt; 91&lt;form id="quantity_form" method="post" action="shoppingcart.jsp"&gt; 92 &lt;input type="hidden" name="movieId" value=&lt;%=m_id%&gt;*&gt; 93 &lt;input type="text" name="q" placeholder=&lt;%=quantity%&gt; value=&lt;%=quantity%&gt;*&gt; 94 &lt;input type="submit" value="Apply"&gt; 95 &lt;/form&gt; 96&lt;a href="shoppingcart.jsp?movieId=&lt;%=m_id %&gt;&amp;q=0"&gt; 97 &lt;button&gt;Delete All&lt;/button&gt;&lt;/a&gt; 98 &lt;/td&gt; 99 &lt;/tr&gt; 100&lt;br&gt; </pre>
StarPage.jsp	33	<pre> 23 .. ... 24     out.println("ds is null."); 25 26     Connection connection = ds.getConnection(); 27     if (connection == null) 28         out.println("dbcon is null."); 29 30     String star_name=request.getParameter("star_name"); 31 32 33 String starQuery="select name,birthYear From stars as s where s.name=?"; 34 PreparedStatement starStatement= connection.prepareStatement(starQuery); 35 starStatement.setString(1,star_name); 36 ResultSet starResult = starStatement.executeQuery(); 37 38 String star_year=""; 39 40 while(starResult.next()){ 41     star_year=starResult.getString("birthYear"); 42 } 43 44 45 46 String moviesQuery="select m.title, m.id from stars as s, movies as m, stars_in_movies as sm where s.name=? and 47 PreparedStatement moviesStatement= connection.prepareStatement(moviesQuery); 48 moviesStatement.setString(1,star_name); 49 ResultSet moviesResult = moviesStatement.executeQuery(); 50 ArrayList&lt;String&gt; movies_name = new ArrayList(); 51 ArrayList&lt;String&gt; movies_id = new ArrayList(); 52 53 while(moviesResult.next()){ 54     movies_name.add(moviesResult.getString("title")); 55     movies_id.add(moviesResult.getString("id")); 56 } 57 58 %&gt; </pre>

AndroidLogin.java	61	<pre> AndroidLogin.java ✘ 45   DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 46 47 48   if (ds == null) 49     out.println("ds is null."); 50 51   Connection connection = ds.getConnection(); 52   if (connection == null) 53     out.println("dbcon is null."); 54 55 56 57   String password=request.getParameter("password"); 58   String user=request.getParameter("user_email"); 59 60   PasswordEncryptor passwordEncryptor = new StrongPasswordEncryptor(); 61   String query = "SELECT * from customers where email=?;"; 62   PreparedStatement preparedstatement =connection.prepareStatement(query); 63 64   preparedStatement.setString(1, user); 65   System.out.println("the query is :"+preparedStatement); 66   ResultSet rs = preparedStatement.executeQuery(); 67   int user_id=1; 68   boolean success = false; 69   if (rs.next()) { 70     // get the encrypted password from the database 71     String encryptedPassword = rs.getString("password"); 72     user_id=rs.getInt("id"); 73     // use the same encryptor to compare the user input password with encrypted passw 74     success = new StrongPasswordEncryptor().checkPassword(password, encryptedPassword 75   } 76 77 78 79 80   if (!success) { 81     JsonObject responseJsonObject = new JsonObject(); 82     responseJsonObject.addProperty("status", "fail"); 83     responseJsonObject.addProperty("message", "email or password incorrect!"); } </pre>
AndroidSearch.java	67	<pre> AndroidSearch.java ✘ 46   Context envCtx = (Context) initCtx.lookup("java:comp/env"); 47   if (envCtx == null) 48     out.println("envCtx is NULL"); 49 50 51   // Look up our data source 52   DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 53 54 55   if (ds == null) 56     out.println("ds is null."); 57 58   Connection connection = ds.getConnection(); 59   if (connection == null) 60     out.println("dbcon is null."); 61   // Declare our statement 62 63   String input = request.getParameter("search"); 64   System.out.println("user input " + input); 65 66 67   String query="select id, title, year,director from movies where title like ?"; 68 69   PreparedStatement preparedStatement=connection.prepareStatement(query); 70   preparedStatement.setString(1,"%"+input+"%"); 71 72   // Perform the query 73   ResultSet rs = preparedStatement.executeQuery(); 74   //ResultSet rs = statement.executeQuery(query); 75 76 77 78 79   JSONArray jsonArray = new JSONArray(); 80   while (rs.next()) 81   { 82 83     String movie_id = rs.getString(1); </pre>

AutoComplete.java	97	 <pre> 82         JSONArray jsonArray = new JSONArray(); 83 84         // get the query string from parameter 85         String query = request.getParameter("query"); 86 87         // return the empty json array if query is null or empty 88         if (query == null    query.trim().isEmpty()) { 89             response.getWriter().write(jsonArray.toString()); 90             return; 91         } 92 93         // search on marvel heros and DC heros and add the results to JSON Array 94         // this example only does a substring match 95         // TODO: in project 4, you should do full text search with MySQL to find 96 97         String title_query="select * from movies where title like ? limit 5"; 98         PreparedStatement preparedStatement=connection.prepareStatement(title_qu 99         preparedStatement.setString(1,"%"+query+"%"); 100 101 102         ResultSet resultSet = preparedStatement.executeQuery(); 103         while(resultSet.next()) { 104             String movieId = resultSet.getString("id"); 105             String movieTitle = resultSet.getString("title"); 106             jsonArray.add(generateJsonObject(movieId, movieTitle, "movie")); 107         } 108 109 110 111         response.getWriter().write(jsonArray.toString()); 112         return; 113     } catch (Exception e) { 114         System.out.println(e); 115         response.sendError(500, e.getMessage()); 116     } 117 } 118 */ 119 /** 120  * Generate the JSON object from hero and category to be like this format: 121 */ </pre>
EmployeeLoginServlet.java	76	 <pre> 65         out.println("ds is null."); 66 67         Connection connection = ds.getConnection(); 68         if (connection == null) 69             out.println("dbcon is null."); 70 71 72         String password=request.getParameter("password"); 73         String user=request.getParameter("user_email"); 74 75         PasswordEncryptor passwordEncryptor = new StrongPasswordEncryptor(); 76         String query = "SELECT * from employees where email= ?;"; 77         PreparedStatement preparedStatement =connection.prepareStatement(query 78 79         preparedStatement.setString(1, user); 80         System.out.println("the query is: "+preparedStatement); 81         ResultSet rs = preparedStatement.executeQuery(); 82         String employee_name = ""; 83         boolean success = false; 84         if (rs.next()) { 85             // get the encrypted password from the database 86             String encryptedPassword = rs.getString("password"); 87             employee_name=rs.getString("fullname"); 88             // use the same encryptor to compare the user input password with 89             success = new StrongPasswordEncryptor().checkPassword(password, en 90         } 91 92 93 94 95         if (!success) { 96             JSONObject responseJsonObject = new JSONObject(); 97             responseJsonObject.addProperty("status", "fail"); 98             responseJsonObject.addProperty("message", "email or password incor 99             response.getWriter().write(responseJsonObject.toString()); 100 101     } else { 102         HttpSession session = request.getSession(true); 103     } </pre>

## LoginServlet.java

77

```
63     if (ds == null)
64         out.println("ds is null.");
65
66     Connection connection = ds.getConnection();
67     if (connection == null)
68         out.println("dbcon is null.");
69
70
71
72     String password=request.getParameter("password");
73     String user=request.getParameter("user_email");
74
75     PasswordEncryptor passwordEncryptor = new StrongPasswordEncryptor();
76     String query = "SELECT * from customers where email= ?;";
77     PreparedStatement preparedStatement =connection.prepareStatement(query);
78
79     preparedStatement.setString(1, user);
80     System.out.println("the query is: "+preparedStatement);
81     ResultSet rs = preparedStatement.executeQuery();
82     int user_id=1;
83     boolean success = false;
84     if (rs.next()) {
85         // get the encrypted password from the database
86         String encryptedPassword = rs.getString("password");
87         user_id=rs.getInt("id");
88         // use the same encryptor to compare the user input password with enc
89         success = new StrongPasswordEncryptor().checkPassword(password, encry
90
91
92
93
94
95
96     if (!success) {
97         JSONObject responseJsonObject = new JSONObject();
98         responseJsonObject.addProperty("status", "fail");
99         responseJsonObject.addProperty("message", "email or password incorrec
100
101
```

## How prepared statement was implemented?

We use prepared statements to replace the statements that created by connection.createStatement(), and use setString()/setInt() to fill in the "?" in the prepared statement to make it be a full search query.