

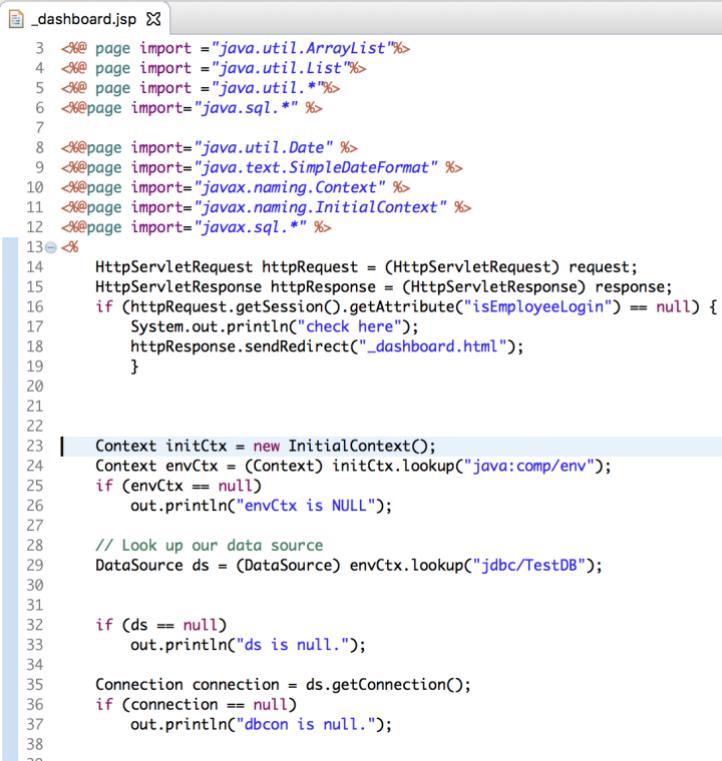
CS 122B Project 5 Report

Team-88

Ye Yuan 49889946 | Lingzhe Kong 93985614

Task 1:

1. Connection Polling:

File Name	Line Number	Snapshot
_dashboard.jsp	23	 <pre>3 <%@ page import="java.util.ArrayList" %> 4 <%@ page import="java.util.List" %> 5 <%@ page import="java.util.*" %> 6 <%@page import="java.sql.*" %> 7 8 <%@page import="java.util.Date" %> 9 <%@page import="java.text.SimpleDateFormat" %> 10 <%@page import="javax.naming.Context" %> 11 <%@page import="javax.naming.InitialContext" %> 12 <%@page import="javax.sql.*" %> 13<%> 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.");</pre>

BrowsePage.jsp	16	<pre> BrowsePage.jsp ✘ 1 <%@page import="java.sql.*" %> 2 <%@ page import = "java.util.ArrayList"%> 3 <%@ page import = "java.util.List"%> 4 <%@page import="javax.naming.Context" %> 5 <%@page import="javax.naming.InitialContext" %> 6 <%@page import="javax.sql.*" %> 7 <%@ page language="java" contentType="text/html; charset=UTF-8" 8 pageEncoding="UTF-8"%> 9 10<%> 11 HttpServletRequest httpRequest = (HttpServletRequest) request; 12 HttpServletResponse httpResponse = (HttpServletResponse) response; 13 14 15 16 Context initCtx = new InitialContext(); 17 Context envCtx = (Context) initCtx.lookup("java:comp/env"); 18 if (envCtx == null) 19 out.println("envCtx is NULL"); 20 21 // Look up our data source 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<String> genres = new ArrayList(); </pre>
checkout.jsp	14	<pre> checkout.jsp ✘ 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="java.util.Date" %> 8 <%@page import="java.text.SimpleDateFormat" %> 9 <%@page import="javax.naming.Context" %> 10 <%@page import="javax.naming.InitialContext" %> 11 <%@page import="javax.sql.*" %> 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/InsertDB"); 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 // declare statement 30 Statement statement = connection.createStatement(); 31 32 String user_id=request.getParameter("id"); </pre>

confirmation.jsp	13	<pre> <code> 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="java.util.Date" %> 8 <%@page import="java.text.SimpleDateFormat" %> 9 <%@page import="javax.naming.Context" %> 10 <%@page import="javax.naming.InitialContext" %> 11 <%@page import="javax.sql.*" %> 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(); </code> </pre>
insertMovie.jsp	17	<pre> <code> 1 <%@ page language="java" contentType="text/html; charset=UTF-8" 2 pageEncoding="UTF-8"%> 3 <%@page import="java.sql.*" %> 4 <%@ page import ="java.util.ArrayList"%> 5 <%@ page import ="java.util.List"%> 6 <%@ page import ="java.util.*"%> 7 <%@page import="javax.naming.Context" %> 8 <%@page import="javax.naming.InitialContext" %> 9 <%@page import="javax.sql.*" %> 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/InsertDB"); 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"); </code> </pre>

<pre>insertStar.jsp</pre>	12	<pre>insertStar.jsp 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<% 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/InsertDB"); 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");</pre>
<pre>MoviePage.jsp</pre>	14	<pre>MoviePage.jsp 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<% 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"; 35 PreparedStatement moviesStatement=connection.prepareStatement(movieQuery); 36 moviesStatement.setString(1,movieId); 37 38 ResultSet movieResult = moviesStatement.executeQuery(); 39 String movieTitle="";</pre>

		<pre> search.jsp <% 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){</pre>
		<pre> shoppingcart.jsp <% 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("previousItems"); 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 } else {</pre>

StarPage.jsp	13	<pre> 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<%> 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(starQue 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 response) 12 */ 13 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException { 14 PrintWriter out = response.getWriter(); 15 16 response.setContentType("text/html"); 17 18 try { 19 Context initCtx = new InitialContext(); 20 Context envCtx = (Context) initCtx.lookup("java:comp/env"); 21 if (envCtx == null) 22 out.println("envCtx is NULL"); 23 24 // Look up our data source 25 DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 26 27 28 if (ds == null) 29 out.println("ds is null."); 30 31 Connection connection = ds.getConnection(); 32 if (connection == null) 33 out.println("dbcon is null."); 34 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>

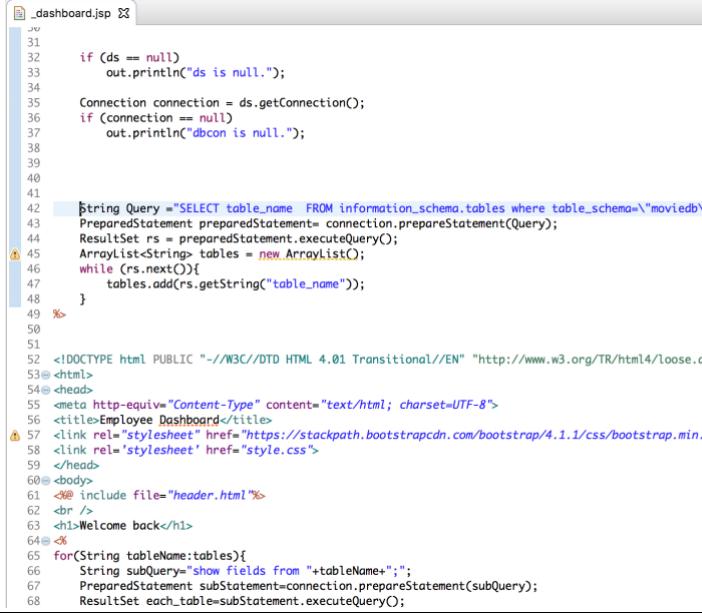
AndroidSearch.java	45	<pre> 37 38 // Output stream to STDOUT 39 PrintWriter out = response.getWriter(); 40 41 42 43 try 44 { 45 Context initCtx = new InitialContext(); 46 Context envCtx = (Context) initCtx.lookup("java:comp/env"); 47 if (envCtx == null) 48 out.println("envCtx is NULL"); 49 50 // Look up our data source 51 DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 52 53 54 if (ds == null) 55 out.println("ds is null."); 56 57 Connection connection = ds.getConnection(); 58 if (connection == null) 59 out.println("dbcon is null."); 60 // Declare our statement 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 wh 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 do 51 * 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("<html>"); 41 out.println("<head><title>Error</title></head>"); 42 out.println("<body>"); 43 out.println("<p>recaptcha verification error</p>"); 44 out.println("<p>" + e.getMessage() + "</p>"); 45 out.println("</body>"); 46 out.println("</html>"); 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("<p>" + e.getMessage() + "</p>"); 45 out.println("</body>"); 46 out.println("</html>"); 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 Numb er	Snapshot
_dashboard.jsp	42	 <pre>31 if (ds == null) 32 out.println("ds is null."); 33 34 Connection connection = ds.getConnection(); 35 if (connection == null) 36 out.println("dbcon is null."); 37 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<String> tables = new ArrayList(); 46 while (rs.next()){ 47 tables.add(rs.getString("table_name")); 48 } 49 50 51 52 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"> 53<html> 54<head> 55 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> 56 <title>Employee Dashboard</title> 57 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"> 58 <link rel="stylesheet" href="style.css"> 59 </head> 60<body> 61 <%@ include file="header.html"%> 62
 63 <h1>Welcome back</h1> 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>

BrowsePage.jsp	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<String> genres = new ArrayList(); 40 while(rs.next()){ 41 genres.add(rs.getString("name")); 42 } 43 44 45 %> 46 47 48 <!DOCTYPE html> 49 @<html> 50 @<head> 51 @<meta charset="UTF-8"> 52 @<title>Browse the Movies</title> 53 @<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"> 54 @<link rel='stylesheet' href='style.css'> 55 @</head> 56 @<body> 57 @@ include file="header.html"@@ 58 @<div> 59 <h1>Browse by movie title first alphanumeric letter</h1> 60 @</pre>
checkout.jsp	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<String,Integer> previousItems = (Map<String,Integer>)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>

		<pre> confirmation.jsp ✘ 10 <%@page import="javax.naming.InitialContext" %> 11 <%@page import="javax.sql.*" %> 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 + "where s.saleDate=? and s.customerId=? and s.movieId=m.id"; 34 PreparedStatement preparedStatement =connection.prepareStatement(query); 35 preparedStatement.setString(1,dateFormat.format(date)); 36 preparedStatement.setString(2,user_id); 37 38 ResultSet salesResult = preparedStatement.executeQuery(); 39 40 %> 41 42 43 44 45 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/strict.dtd"> 46 @<html> 47 @<head> 48 @<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> 49 @<title>Insert New Movie</title> 50 @<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"> 51 @<link rel="stylesheet" href="style.css"> 52 @</head> </pre>
insertMovie.jsp	39	<pre> insertMovie.jsp ✘ 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 %> 48 49 50 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/strict.dtd"> 51 @<html> 52 @<head> 53 @<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> 54 @<title>Insert New Movie</title> 55 @<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"> 56 @<link rel="stylesheet" href="style.css"> 57 @</head> </pre>

insertStar.jsp	43	<pre> 21 out.println("ds is null."); 22 23 Connection connection = ds.getConnection(); 24 if (connection == null) 25 out.println("dbcon is null."); 26 // declare statement 27 String star_name=request.getParameter("name"); 28 String star_year=request.getParameter("year"); 29 if(star_year.equals("")){ 30 star_year=null; 31 } 32 String maxIDquery="select max(id) as id from stars"; 33 System.out.println(star_year); 34 PreparedStatement preparedStatement =connection.prepareStatement(maxIDquery); 35 ResultSet rs=preparedStatement.executeQuery(); 36 String maxid=""; 37 while(rs.next()){ 38 maxid = rs.getString("id"); 39 maxid = "nm" + (Integer.parseInt(maxid.substring(2,maxid.length()))); 40 } 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 % 56 57 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"> 58<html> 59<head></pre>
MoviePage.jsp	34	<pre> 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</pre>

search.jsp

67

```
search.jsp ✘
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+"%\'
73 |                 + and sm.starId=s.id) as nm, ratings 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 |             preparedStatement=connection.prepareStatement(query);
93 |             preparedStatement.setString(1,"%"+star_name+"%");
94 |             preparedStatement.setString(2,"%"+title+"%");
95 |             preparedStatement.setString(3,"%"+year+"%");
96 |             preparedStatement.setString(4,"%"+director+"% ");
97 |             //preparedStatement.setString(5,sorted_by);
98 |             preparedStatement.setInt(5,number_per_page);
99 |             preparedStatement.setInt(6,(start_from-1)*number_per_page);
100 |
101 }else if(browse_type.equals("a")){
102 |
103 }
```

shoppingcart.jsp

81

```
shoppingcart.jsp ✘
63 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/css/bootstrap.min.css">
64 <link rel='stylesheet' href='style.css'>
65 </head>
66 @body>
67 <%@ include file="header.html"%>
68 <br />
69 <div>
70 @table border>
71 <tr><th colspan="3">Shopping Cart</th></tr>
72 <td>Movie Name</td><td>Quantity</td><td>Modify the Quantity</td></tr>
73 <%
74 int exist_item=0;
75 for(String m_id:previousItems.keySet()){
76     int quantity=previousItems.get(m_id);
77
78     if(quantity>0){
79         exist_item=1;
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     }
88 <tr>
89 <td>@movie_title @</td><td>@quantity@</td>
90 @td>
91 <form id="quantity_form" method="post" action="shoppingcart.jsp">
92 <input type="hidden" name="movieId" value="@m_id@"/>
93 <input type="text" name="q" placeholder="@quantity@" value="@quantity@" />
94 <input type="submit" value="Apply">
95 </form>
96 @a href="shoppingcart.jsp?movieId=@m_id@&q=0">
97 <button>Delete All</button></a>
98 </td>
99 </tr>
100
101 </div>
```

StarPage.jsp

33

```
StarPage.jsp [3]
1  out.println("ds is null.");
2
3  Connection connection = ds.getConnection();
4  if (connection == null)
5      out.println("dbcon is null.");
6
7  String star_name=request.getParameter("star_name");
8
9
10 String starQuery="select name,birthYear from stars as s where s.name=?";
11 PreparedStatement starStatement= connection.prepareStatement(starQuery);
12 starStatement.setString1(star_name);
13 ResultSet starResult = starStatement.executeQuery();
14
15 String star_year="";
16
17 while(starResult.next()){
18     star_year=starResult.getString("birthYear");
19 }
20
21
22 String moviesQuery="select m.title, m.id from stars as s, movies as m, stars_in_movies as sm where s.name=? and sm.movie_id=m.id";
23 PreparedStatement moviesStatement= connection.prepareStatement(moviesQuery);
24 moviesStatement.setString1(star_name);
25 ResultSet moviesResult = moviesStatement.executeQuery();
26 ArrayList<String> movies_name = new ArrayList();
27 ArrayList<String> movies_id = new ArrayList();
28
29 while(moviesResult.next()){
30     movies_name.add(moviesResult.getString("title"));
31     movies_id.add(moviesResult.getString("id"));
32 }
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58 %>
```

AndroidLogin.java

61

```
AndroidLogin.java [3]
1  DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB");
2
3
4  if (ds == null)
5      out.println("ds is null.");
6
7  Connection connection = ds.getConnection();
8  if (connection == null)
9      out.println("dbcon is null.");
10
11
12 String password=request.getParameter("password");
13 String user=request.getParameter("user_email");
14
15 PasswordEncryptor passwordEncryptor = new StrongPasswordEncryptor();
16 String query = "SELECT * from customers where email= ?;";
17 PreparedStatement preparedStatement =connection.prepareStatement(query);
18
19 preparedStatement.setString1(user);
20 System.out.println("the query is: "+preparedStatement);
21 ResultSet rs = preparedStatement.executeQuery();
22 int user_id=1;
23 boolean success = false;
24 if (rs.next()) {
25     // get the encrypted password from the database
26     String encryptedPassword = rs.getString("password");
27     user_id=rs.getInt("id");
28     // use the same encryptor to compare the user input password with encrypted pass
29     success = new StrongPasswordEncryptor().checkPassword(password, encryptedPassw
30 }
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
```

AndroidSearch.java	67	<pre> 46 Context envCtx = (Context) initCtx.lookup("java:comp/env"); 47 if (envCtx == null) 48 out.println("envCtx is NULL"); 49 50 // Look up our data source 51 DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB"); 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 String input = request.getParameter("search"); 63 System.out.println("user input " + input); 64 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 JSONArray jsonArray = new JSONArray(); 79 while (rs.next()) 80 { 81 82 String movie_id = rs.getString(1); 83 String movie_title = rs.getString(2); 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 </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+"%"); 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 </pre>

EmployeeLoginServlet
.java

76

```
EmployeeLoginServlet.java ✘
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, encr
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     }
104 }
```

LoginServlet.java

77

```
LoginServlet.java ✘
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();
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 er
89         success = new StrongPasswordEncryptor().checkPassword(password, encr
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 incorre
100         response.getWriter().write(responseJsonObject.toString());
101     }
102 }
```

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.

Task 2

- Address of AWS and Google instances

AWS instance 1: 52.14.246.47

Instance 2 (master): 18.221.120.129

Instance 3 (slave): 18.219.152.237

Google cloud platform: 35.185.89.240

- Have you verified that they are accessible? Does Fablix site get opened both on Google's 80 port and AWS' 8080 port?

YES

- Explain how connection pooling works with two backend SQL (in your code)?

- File name, line numbers as in Github

In /project5/WebContent/META-INF/context.xml

Line 15- line 23

In /project5/WebContent/WEB-INF/web.xml

Line 8 – line 23

We make two datasource one is TestDB and one is InsertDB and store the user name, password, url of mysql in it.

- Snapshots

```

13
14
15 <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource"
16     maxTotal="100" maxIdle="30" maxWaitMillis="10000" username="mytestuser"
17     password="mypassword" driverClassName="com.mysql.jdbc.Driver"
18     url="jdbc:mysql://localhost:3306/moviedb?autoReconnect=true&useSSL=false"/>
19
20 <Resource name="jdbc/InsertDB" auth="Container" type="javax.sql.DataSource"
21     maxTotal="100" maxIdle="30" maxWaitMillis="10000" username="mytestuser"
22     password="mypassword" driverClassName="com.mysql.jdbc.Driver"
23     url="jdbc:mysql://172.31.27.61:3306/moviedb?autoReconnect=true&useSSL=false"/>
24 </Context>
```

```

8 <resource-ref>
9     <description>
10        Read
11    </description>
12    <res-ref-name>jdbc/testDB</res-ref-name>
13    <res-type>javax.sql.DataSource</res-type>
14    <res-auth>Container</res-auth>
15 </resource-ref>
16 <resource-ref>
17     <description>
18        Insert
19    </description>
20    <res-ref-name>jdbc/InsertDB</res-ref-name>
21    <res-type>javax.sql.DataSource</res-type>
22    <res-auth>Container</res-auth>
23 </resource-ref>
24 </web-app>
```

- How read/write requests were routed?
 - File name, line numbers as in Github
- In /project5/WebContent/META-INF/context.xml

We let all the read request get connection with datasource jdbc/TestDB which has url “localhost”

And then , we let all the insert request get connection with datasource jdbc/InsertDB which has the internal IP address of the master instance, so in this case ,all the write requests will go to master.

- Snapshots

```
13
14
15    <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource"
16        maxTotal="100" maxIdle="30" maxWaitMillis="10000" username="mytestuser"
17        password="mypassword" driverClassName="com.mysql.jdbc.Driver"
18        url="jdbc:mysql://localhost:3306/moviedb?autoReconnect=true&useSSL=false"/>
19
20    <Resource name="jdbc/InsertDB" auth="Container" type="javax.sql.DataSource"
21        maxTotal="100" maxIdle="30" maxWaitMillis="10000" username="mytestuser"
22        password="mypassword" driverClassName="com.mysql.jdbc.Driver"
23        url="jdbc:mysql://172.31.27.61:3306/moviedb?autoReconnect=true&useSSL=false"/>
24  </Context>
```

Notice: we let all the insert servlets use jdbc/InsertDB to get connection, you can see the screenshots from above(Task 1 screenshots)

Task 3

- Have you uploaded the log files to Github? Where is it located?

It is in the folder called “report”. There are 9 cases logs in this folder.

- Have you uploaded the HTML file (with all sections including analysis, written up) to Github? Where is it located?

Yes, it is in the “report” folder.

- Have you uploaded the script to Github? Where is it located?

Yes, it is in the “report” folder and each subfolder as well.

- Have you uploaded the WAR file and README to Github? Where is it located?

Yes, They are in the root folder.