
Date 23/5/2023

CSM3023

WEB BASED APPLICATION DEVELOPMENT

DR FAIZAH BINTI APLOP

OMAR ISMAIL ABDJALEEL ALOMORY

LAB 6(MP2)

Task 1: Using JSP Page to Access a Simple MySQL Database:

1.1 Create a table namely FirstTable using phpMyAdmin.



The screenshot shows the phpMyAdmin interface for a database named 'csflab6'. The 'Table: firsttable' tab is selected. The 'Table structure' view is active, displaying a table with one column named 'NAME' of type 'varchar(45)'. The table is using the 'utf8mb4_general_ci' collation. The 'NAME' column is highlighted with a red arrow. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	NAME	varchar(45)	utf8mb4_general_ci		Yes	NULL			Change Drop More

1.2 Create SampleInsertionRecord.jsp to insert data in FirstTable table.

Compiled successfully.

```
ant -f "C:\Users\komar\OneDrive - umt.edu.my\Semester 4\CSM3023 WEB BASED APPLICATION DEVELOPMENT by FAIZAH BINTI APLOP APLOP\Lab solution\Lab6" -Dnb.internal.action.name=compile.single -Dfor.compile.single-jsp:
init:
deps-module-jar:
deps-ear-jar:
deps-jar:
library-inclusion-in-archive:
library-inclusion-in-manifest:
compile:
Created dir: C:\Users\komar\OneDrive - umt.edu.my\Semester 4\CSM3023 WEB BASED APPLICATION DEVELOPMENT by FAIZAH BINTI APLOP APLOP\Lab solution\Lab6\build\generated\src
Created dir: C:\Users\komar\OneDrive - umt.edu.my\Semester 4\CSM3023 WEB BASED APPLICATION DEVELOPMENT by FAIZAH BINTI APLOP APLOP\Lab solution\Lab6\build\generated\classes
Compiling 1 source file to C:\Users\komar\OneDrive - umt.edu.my\Semester 4\CSM3023 WEB BASED APPLICATION DEVELOPMENT by FAIZAH BINTI APLOP APLOP\Lab solution\Lab6\build\generated\classes
BUILD SUCCESSFUL (total time: 1 second)
```

Output (data saved into the database then retrieved)

Lab 6 - Task 1 - Sample Insertion records into MySQL through JSP's page

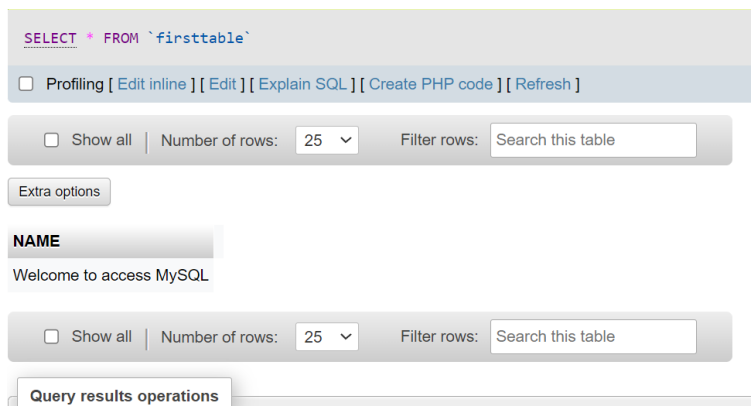
Step 1: MySql Driver is loaded ...!

Step 2: Database is Connected

Step 3: Prepared statement created

Step 4: Perform insertion to the record. com.mysql.jdbc.exceptions.MySQLException: Data truncation: Data too long for column 'NAME' at row 1

1.3 – go to the database

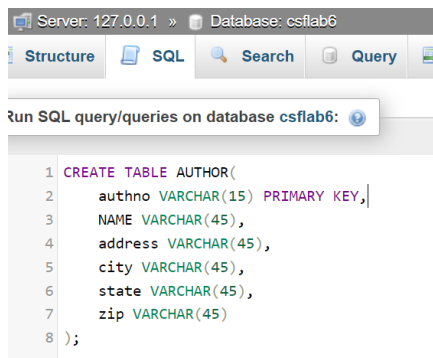


The screenshot shows the phpMyAdmin interface for the 'firsttable' table. The 'Query results operations' tab is active, displaying the results of a 'SELECT * FROM `firsttable`' query. The results show a single row with the value 'NAME' in the 'NAME' column. The table structure is as follows:

NAME
NAME

Task 2: Create Records via JSP Page

Create database:



Server: 127.0.0.1 » Database: csflab6

Structure SQL Search Query

Run SQL query/queries on database csflab6:

```
1 CREATE TABLE AUTHOR(  
2     authno VARCHAR(15) PRIMARY KEY,  
3     NAME VARCHAR(45),  
4     address VARCHAR(45),  
5     city VARCHAR(45),  
6     state VARCHAR(45),  
7     zip VARCHAR(45)  
8 );
```

Creating Author.java class with setters and getters

```
package lab6.com.java;  
  
/**  
 *  
 * @author Omar Alomory(S63955)  
 */  
public class Author {  
    private String authno;  
    private String name;  
    private String address;  
    private String city;  
    private String state;  
    private String zip;  
  
    public String getAuthno() {  
        return authno;  
    }  
  
    public void setAuthno(String authno) {  
        this.authno = authno;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

Compiling processAuthor.jsp



processAuthor.jsp

```
37 myPS.setString(6, myAuthor.getZip());  
38  
39 int result = myPS.executeUpdate();
```

Output - Lab6 (compile-single-jsp)

```
ant -f "C:\Users\komar\OneDrive - umt.edu.my\Semester 4\CSM3023 WEB BASED APPLICATION DEVELOPMENT by FAIZAH BINTI APLOP APLOP\Lab solution\Lab6" -Dmb.internal.action.name=compile.single -Dfor  
compile-single-jsp:  
init:  
deps-module-jar:  
deps-ear-jar:  
deps-jar:  
library-inclusion-in-archive:  
library-inclusion-in-manifest:  
compile:  
Compiling 1 source file to C:\Users\komar\OneDrive - umt.edu.my\Semester 4\CSM3023 WEB BASED APPLICATION DEVELOPMENT by FAIZAH BINTI APLOP APLOP\Lab solution\Lab6\build\generated\classes  
BUILD SUCCESSFUL (total time: 7 seconds)
```

Output

Lab 6 - Task 2 - Perform creating and retrieving record via JSP page

Record Successfully added into Author table...!

Record with Author No gsk23322 successfully created ..!

Details of record are:

Address : Malaysia

City : KT

State : UMT

Zip : 23

Reflection

1. What have you learnt from this exercise?

in this exercise, I have learnt how to use JSP page to insert and retrieve information from database, using JavaBeans to store the coming parameters from the request, then saving it to the database, then retrieve (from the JavaBeans) it and display it to the client.

2. Define step by step before you successfully perform the transaction in a database.

1. First, we have to call the driver jar file.
 2. Second, we have to establish the connection with database with some parameters such as database URL, username, and password of the database.
 3. Third, we have to create the preparedStatement Object which will hold the query we want to do in the database. This can hold the operations we want to make such as INSERT, UPDATE, SELECT, DELETE, etc.
 4. The preparedStatement will also hold the parameters included in the database and its values such as column names and their corresponding values, by setting it individually: e.g., preparedStatement.setDataType(column no, valueToBeSaved);
 5. Then we going to execute the query by calling executeUpdate() method from the preparedStatement object, this method will return integer value represent the columns effected.
 - If the columns effected are more than zero, its inserted successfully.
 6. Display the result, if required, and close the connection.
-
-

Task 3: Create Records Constrained by Regular Expression In JSP

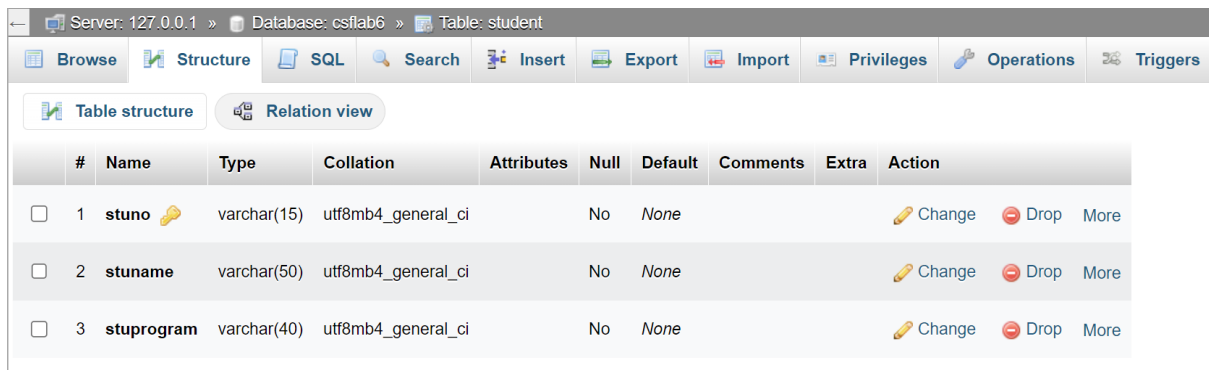
Step 1 - Create a table book using phpMyAdmin.

SQL:

Run SQL query/queries on table csflab6.student: 

```
1 CREATE TABLE student (  
2     stuno VARCHAR(15) PRIMARY KEY,  
3     stuname VARCHAR(50),  
4     stuprogram VARCHAR(40)  
5 );
```

Created successfully.



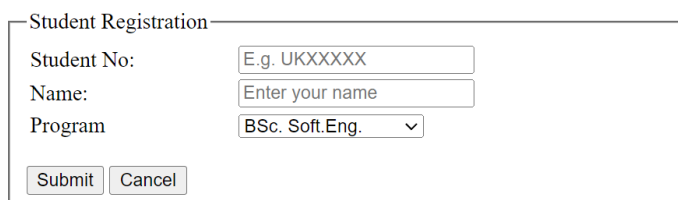
The screenshot shows the phpMyAdmin interface for the 'student' table in the 'csflab6' database. The 'Table structure' tab is selected, displaying a table with 3 columns: 'stuno' (VARCHAR(15), PRIMARY KEY), 'stuname' (VARCHAR(50)), and 'stuprogram' (VARCHAR(40)). All columns have a 'utf8mb4_general_ci' collation and are 'Not Null' with a 'None' default value. The 'Action' column provides links to 'Change', 'Drop', and 'More' for each column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	stuno	varchar(15)	utf8mb4_general_ci		No	None			Change Drop More
2	stuname	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	stuprogram	varchar(40)	utf8mb4_general_ci		No	None			Change Drop More

Step 2 - Create insertStudent.jsp as a main interface to register a new student.

Output

Lab6 - Task 3 -



The screenshot shows a web form titled 'Student Registration'. It contains three input fields: 'Student No:' with a text input containing 'E.g. UKXXXXX', 'Name:' with a text input containing 'Enter your name', and 'Program' with a dropdown menu showing 'BSc. Soft.Eng.'. Below the inputs are 'Submit' and 'Cancel' buttons.

Student Registration	
Student No:	<input type="text" value="E.g. UKXXXXX"/>
Name:	<input type="text" value="Enter your name"/>
Program	<input type="text" value="BSc. Soft.Eng."/>
<input type="button" value="Submit"/> <input type="button" value="Cancel"/>	

Step 3 - Create Student JavaBeans

```
public class Student {
    private String stuno;
    private String stuname;
    private String stuprogram;

    public String getStuno() {
        Pattern pt = Pattern.compile(regex: "[A-Z0-9]*");
        Matcher mt = pt.matcher(input: stuno);
        boolean bl = mt.matches();

        if(bl == true){
            return this.stuno;
        }else{
            throw new NumberFormatException("<p style='color:red'>Error occur: Format is invaild for <b>"+this.stuno+" </b><p>");
        }
    }

    public void setStuno(String stuno) {
        this.stuno = stuno;
    }

    public String getStuname() {
        return stuname;
    }

    public void setStuname(String stuname) {
        this.stuname = stuname;
    }

    public String getStuprogram() {
        return stuprogram;
    }

    public void setStuprogram(String stuprogram) {
```

If user enter mismatched format error will occur in errorStudent.jsp

Step 4 - Create processStudent.jsp to insert a record into the database.

```
int result;
Connection myConnection = null;

Class.forName("com.mysql.jdbc.Driver");
System.out.println("Step 1: MySQL driver loaded...!");

String myURL = "jdbc:mysql://localhost/csflab6";
myConnection = DriverManager.getConnection(myURL, "root", "admin");
System.out.println("Step 2: Database is connected...!");

String sInsertQry = "INSERT INTO students (stuno, stuname, stuprogram) VALUES(?, ?, ?)";
PreparedStatement myPS = myConnection.prepareStatement(sInsertQry);
System.out.println("Step 3: Prepared Statement created!");
System.out.println("Step 4: Perform insertion of record!");
myPS.setString(1, myStudent.getStuno());
myPS.setString(2, myStudent.getStuname());
myPS.setString(3, myStudent.getStuprogram());

result = myPS.executeUpdate();
if (result > 0) {
    System.out.println("\tRecord Successfully added into Student's table...!");
    out.print("<div class= 'theStyle'><p> " + "Record with Student No " + myStudent.getStuno() + " successfully created ..!" + "</p>");
    out.print("<p>" + "Details of record are: " + "</p>");
    out.print("<p>Student No : " + myStudent.getStuno() + "</p>");
    out.print("<p>Name : " + myStudent.getStuname() + "</p>");
    out.print("<p>Program : " + myStudent.getStuprogram() + "</p></div>");
}

System.out.println("Step 5: Close database connection ...!");
myConnection.close();
System.out.println("Database Connection is closed ...!");
```

Step 5 - Create errorBook.jsp to display any error message.

```
<!--
Document   : errorStudent
Created on : May 28, 2023, 8:56:38 PM
Author    : Omar Alomory (S63955)
-->
<%@ page isErrorPage="true" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<form id="errorFrm" action="insertStudent.jsp" method="post">
<h2>Lab6 - Task 3 - Perform creating and retrieving record via JSP page</h2>
<p>When inserting record ...!</p>
<p><jsp:expression> exception.getMessage() </jsp:expression></p>
<br>
</form>
</body>
</html>
```

Step 6 - Running the program and create a new database.

Lab6 - Task 3 - Perform creating and retrieving record via JSP page

Student Registration	
Student No:	<input type="text" value="UK88888"/>
Name:	<input type="text" value="Mohamad Nor Hassan"/>
Program	<input type="text" value="BSc. in Networking"/>
<input type="button" value="Submit"/> <input type="button" value="Cancel"/>	

©2023-Omar Alomory

Process Student Page

Record with Student No UK88888 successfully created ..!

Details of record are:

Student No : UK88888

Name : Mohamad Nor Hassan

Program : BSc. in Networking

In case there is mismatch format for student no:

Lab6 - Task 3 - Perform creating and retrieving record via JSP page

Student Registration

Student No:

sdfgsd

Name:

Mohamad Nor Hassan

Program

BSc. in Networking ▾

Submit

Cancel

©2023-Omar Alomory

Process Student Page

Error: java.lang.NumberFormatException:

Error occur: Format is invaild for sdfgsd

Reflection

1. What have you learnt from this exercise?

Mostly the same like the previous task, additionally I have learnt how to handle error and create a separated page that can recognize the error and print it out in individual, such as mismatching error or 404 error , etc.

2. Define step by step before you successfully perform the transaction in a database.

1. First, we have to call the driver jar file.
2. Second, we have to establish the connection with database with some parameters such as database URL, stuno, and stuname of the database.
3. Third, we have to create the preparedStatement Object which will hold the query we want to do in the database. This can hold the operations we want to make such as INSERT, UPDATE, SELECT, DELETE, etc.
4. The preparedStatement will also hold the parameters included in the database and its values such as column names and their corresponding values, by setting it individually: e.g., preparedStatement.setDataType(column no, valueToBeSaved);
5. Then we going to execute the query by calling executeUpdate() method from the preparedStatement object, this method will return integer value represent the columns effected.
 - If the columns effected are more than zero, its inserted successfully.
6. Display the result, if required, and close the connection.

Task 4: Perform Retrieving Records Via JSP Page.

Run program insertStudent.jsp from Task 3.

Insert the following records:

Process Student Page

Record with Student No UK12489 successfully created ..!
Details of record are:
Student No : UK12489
Name : Ahmad Salam
Program : BSc. with IM

Process Student Page

Record with Student No UK56789 successfully created ..!
Details of record are:
Student No : UK56789
Name : Rosnah Azman
Program : BSc. Soft.Eng.

Process Student Page

Record with Student No UK67342 successfully created ..!
Details of record are:
Student No : UK67342
Name : Liew Cheng Huat
Program : BSc. in Robotics

Compiled successfully.

```
ant -f "C:\Users\komar\OneDrive - umt.edu.my\Semester 4\CSM3023 WEB BASED APPLICATION DEVELOPMENT by FAIZAH BINTI APLOP APLOP\Lab solution\Lab6" -Ddb.internal.action.name=compile.single -Dfor.compile-single-jsp:  
init:  
deps-module-jar:  
deps-ear-jar:  
deps-jar:  
library-inclusion-in-archive:  
library-inclusion-in-manifest:  
compile:  
Compiling 1 source file to C:\Users\komar\OneDrive - umt.edu.my\Semester 4\CSM3023 WEB BASED APPLICATION DEVELOPMENT by FAIZAH BINTI APLOP APLOP\Lab solution\Lab6\build\generated\classes  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Output

Lab 6 - Task 4 : Retrieving record vis JSP page

ISBNNo	Author	Title
UK12489	Ahmad Salam	BSc. with IM
UK56789	Rosnah Azman	BSc. Soft.Eng.
UK67342	Liew Cheng Huat	BSc. in Robotics
UK88888	Mohamad Nor Hassan	BSc. in Networking

Reflection

1. What have you learnt from this exercise?

I have learnt how to retrieve stored data from database and then display it in JSP page in organized order.

2. Explain the differences when using Statement() and PreparedStatement().

Statement (): It is used for accessing your database. **Statement interface** cannot accept parameters and useful when you are using static **SQL** statements at runtime. If you want to run **SQL** query only once, then this interface is preferred over **PreparedStatement**.

PreparedStatement (): It is used when you want to use **SQL** statements many times. The **PreparedStatement** interface accepts input parameters at runtime. Its extended from **Statement** interface.

Task 5: Create A Record Using JSP Model 1

Compiled successfully.

```
ant -f "C:\Users\komar\OneDrive - umt.edu.my\Semester 4\CSM3023 WEB BASED APPLICATION DEVELOPMENT by FAIZAH BINTI APLOP AFLOP\Lab solution\Lab6" -Dmb.internal.action.name=compile.single -Dfor
compile-single-jsp:
init:
deps-module-jar:
deps-war-jar:
deps-jar:
library-inclusion-in-archive:
library-inclusion-in-manifest:
compile:
Compiling 1 source file to C:\Users\komar\OneDrive - umt.edu.my\Semester 4\CSM3023 WEB BASED APPLICATION DEVELOPMENT by FAIZAH BINTI APLOP AFLOP\Lab solution\Lab6\build\generated\classes
BUILD SUCCESSFUL (total time: 0 seconds)
```

Output

Lab6 - Task 3 - Perform creating and retrieving record via JSP page

Student Registration

IC No:

890710-11-2369

Name:

Mohamad Nor Hassan

Program

7 KM

Submit

Cancel

©2023-Omar Alomory

Record with IC No 890710-11-2369 successfully created ..!

Details of record are:

IC No : 890710-11-2369

Name : Mohamad Nor Hassan

Category : 7 KM

©2023-Omar Alomory

Reflection

1. What have you learnt from this exercise?

I have learnt how to separation of business logic and display. We had the database connectivity and the closing of it in separated class. The user input is handled by JavaBeans, the insertion is made by DAO class, and lastly we display the information using JSP.

2. Describe the benefits of using JavaBeans.

It promote code reusability and modularity, allowing components to be easily integrated into different applications. JavaBeans enforce encapsulation, hiding implementation details and providing well-defined interfaces. They support event-driven programming, enabling components to generate and respond to events. JavaBeans are well-supported by IDEs, making it convenient to design and connect components visually. They are serializable, facilitating persistence and communication. The standardized conventions and naming patterns of JavaBeans promote interoperability and ease of use. Overall, JavaBeans enhance developer productivity, maintainability, and scalability in Java applications.

EXERCISE:

Creating database:

```
Run SQL query/queries on table csflab6.userprofile: ?

1 CREATE TABLE userporfile(
2     username VARCHAR(15) PRIMARY KEY,
3     password VARCHAR(10),
4     firstname VARCHAR(50),
5     lastname VARCHAR(50)
6 );
```

insertUser.html

Register

Username:

Password:

First Name:

Last Name:

[Already Memeber](#)

In between is processUser.jsp to save user information to the database

User information in the database:

	username	password	firstname	lastname
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	Nightmare	Dream	Omar	Alomory

The process will be redirected to login.jsp

Register

Username:

Password:

[Register Now](#)

In between there is dologin.jsp to process the login and retrieve the required information from the database.

main.jsp

Welcome Omar Alomory

[Logout](#)

When login pressed the session attributes will be invalidated and totally removed.

If user entered invalid username or password, a message will be displayed indicating there is misspelled in one of the fields.

Register

invalid username or password

Username:

Password:

[Register Now](#)

That's all thank you.
