Course: Web Application Development
Lab 4 - Database Connection

Content:

- How to connect to Database Server (VSCode)
- Practices and Exercises

Part 1: Database Management System (MySQL)

To install MySQL and MySQL Workbench, please refer to Lab 01
(file: WAD_Lab01_Basic_HTML.pdf)

You can also find the user manual of MySQL Workbench at
https://dev.mysql.com/doc/workbench/en/

To review SQL you can visit https://www.w3schools.com/sql/

**Note: MySQL Workbench is optional, you can use your system's command prompt or terminal to create a database for your website.

Example:
- Create a database for courses registration
- Create the following tables in your new database

```
-- --------------------------------------------------------

--
-- Table structure for table 'course'
--

CREATE TABLE IF NOT EXISTS course (
  _ID int(10) NOT NULL auto_increment,
  CourseName varchar(255) NOT NULL,
  Credits int(1) NOT NULL,
  PRIMARY KEY  (_ID),
  KEY Course_CourseID_INDEX (CourseName)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;

-- --------------------------------------------------------

--
-- Table structure for table 'student'
--
CREATE TABLE IF NOT EXISTS student (
  _ID int(10) NOT NULL auto_increment,
  StudentID varchar(10) NOT NULL,
  StudentName varchar(255) NOT NULL,
  PRIMARY KEY  (_ID)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 AUTO_INCREMENT=6 ;
-- --------------------------------------------------------
```

```
--
-- Table structure for table 'studentcourse'
--

CREATE TABLE IF NOT EXISTS studentcourse (
  Student_ID int(10) NOT NULL,
  Course_ID int(10) NOT NULL,
  PRIMARY KEY  (Student_ID,Course_ID),
  KEY StudentCourse_StudentID_INDEX (Student_ID),
  KEY StudentCourse_CourseID_INDEX (Course_ID)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

- Use file <course.csv> to import records to your database (you can either copy/paste, write java code, or SQL query).
- Create one or more records for student table.

Part 2: Connect to MySQL Server from a JSP file

- **Step 1:** Follow the steps in the previous lab to create a new Java Web project
- **Step 2:** Add Server Tomcat to your project
- **Step 3:** Add JDBC driver dependency to pom.xml

```
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.49</version>
</dependency>
```

Note: The version will be different depending on the version of your MySQL Server. Please check it carefully.

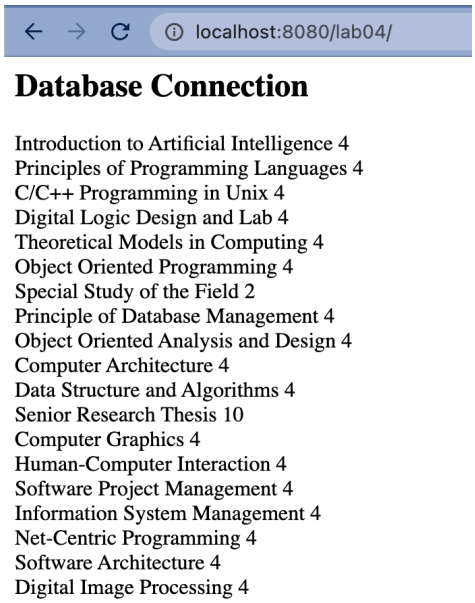
- **Step 4:** Update the content of index.jsp as follow:

Note: for a better experience with JSP, you may want to install Java Server Pages (JSP) from the extension market.

```jsp
<html>
<%@ page import="java.sql.*"%>
<body>
    <h2>Database Connection</h2>
</body>
<%
    String url = "jdbc:mysql://localhost:3306/course_registration_demo"; // Define connection URL
    String user = "demo_java"; //username of your database
    String password = "1234"; //password
    Connection dbConnection = null;
    Statement stmt = null;
    ResultSet rs = null;

    Class.forName("com.mysql.jdbc.Driver");// Load JDBC driver
    dbConnection = DriverManager.getConnection(url, user, password); //Establish a connection
    stmt = dbConnection.createStatement(); //Create a query object
    rs = stmt.executeQuery("SELECT * FROM course"); //Execute a query
    //Process the query result
    while (rs.next()) {
        out.println(rs.getString("CourseName"));
        out.println(rs.getString("Credits")+"<br>");
    }
    rs.close(); //Close the connection
%>
</html>
```

- **Step 5:** use Maven to verify the project, and run it on Tomcat



Part 3: Exercises

Use what you have learned to create several JSP files:

- **login.jsp**: provide login form and check username and password for authentication. If the user inputs a valid username and password, he/she will go to homepage.jsp.

** Note: Use StudentID column in student table to validate username and password.

- **homepage.jsp:** display your username and registered courses. There should be a button or link to add/update registered course;
- **register.jsp:** allow user to register courses by providing a list of available courses and check-box for selection.