Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**ЗВІТ**

**Про виконання практичної роботи № 3**

**з дисципліни «Програмне забезпечення мережевих технологій»**

**на тему «Cтворення та робота з базою даних за допомогою стандартних бібліотек Java»**

**Лектор:**

проф. кафедри ПЗ

Мельник Р.А.

**Виконав:**

студ. групи ПІ-31

Павленчик М. М.

**Прийняв:**

проф. кафедри ПЗ

Мельник Р.А.

«\_\_\_» \_\_\_\_\_\_ 2017 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2017

**ТЕМА РОБОТИ**: Cтворення та робота з базою даних за допомогою стандартних бібліотек Java.

**Завдання роботи**

Ознайомитись із концепцією роботи із базою даних в JAVA. Засвоїти основні стандартні класи і методи бібліотек sql та swing, та бібліотек, що дозволяють опрацьовувати дані СУБД.

**Результати виконання**

На початку роботи застосування, після запуску сервера, клієнт отримує у відповідь на запит стартову сторінку (рис. 1).

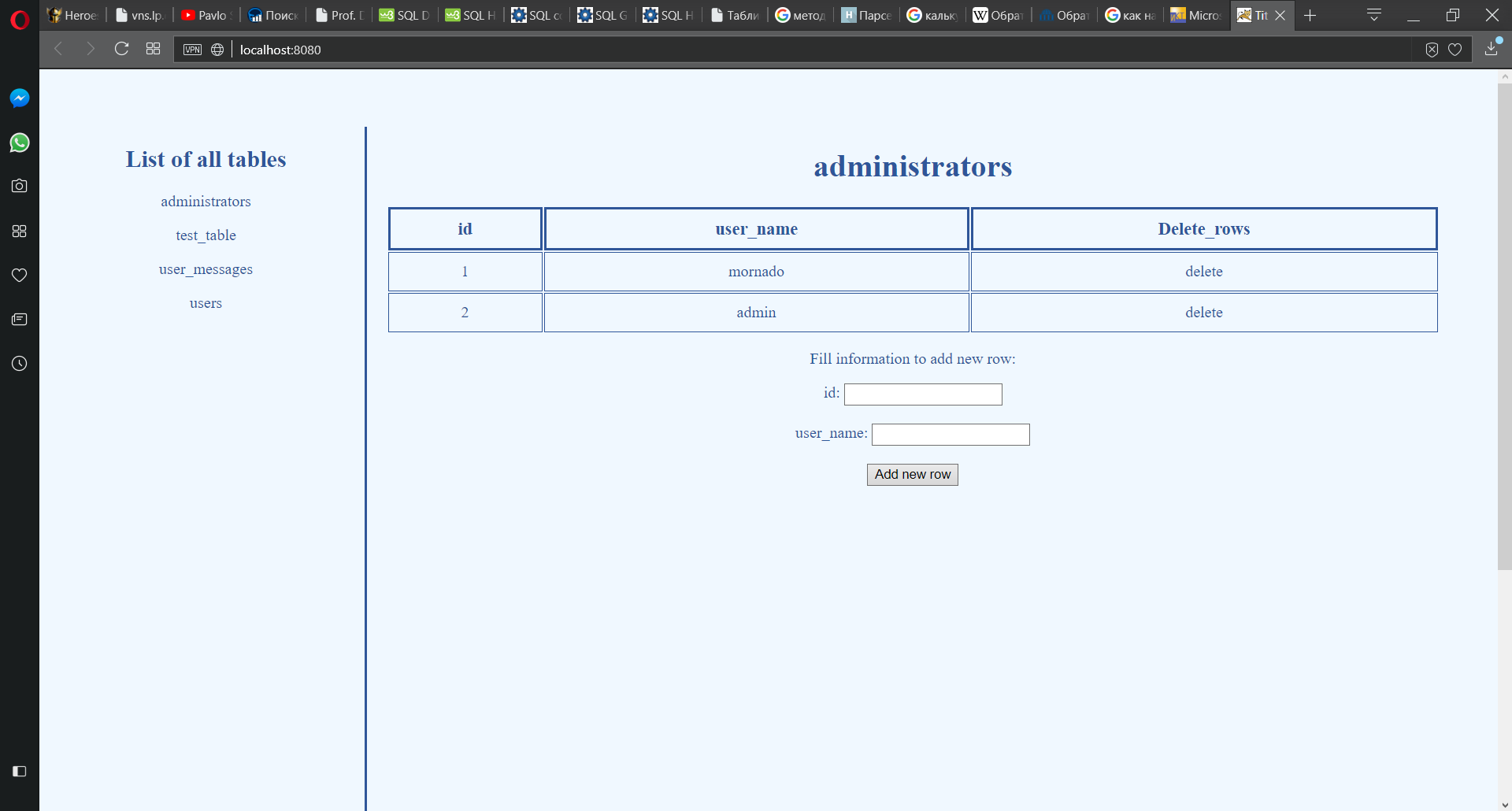


Рис. 1. Початок роботи

Для того, щоб отримати дані з іншої таблиці, потрібно обрати відповідну таблицю у списку.

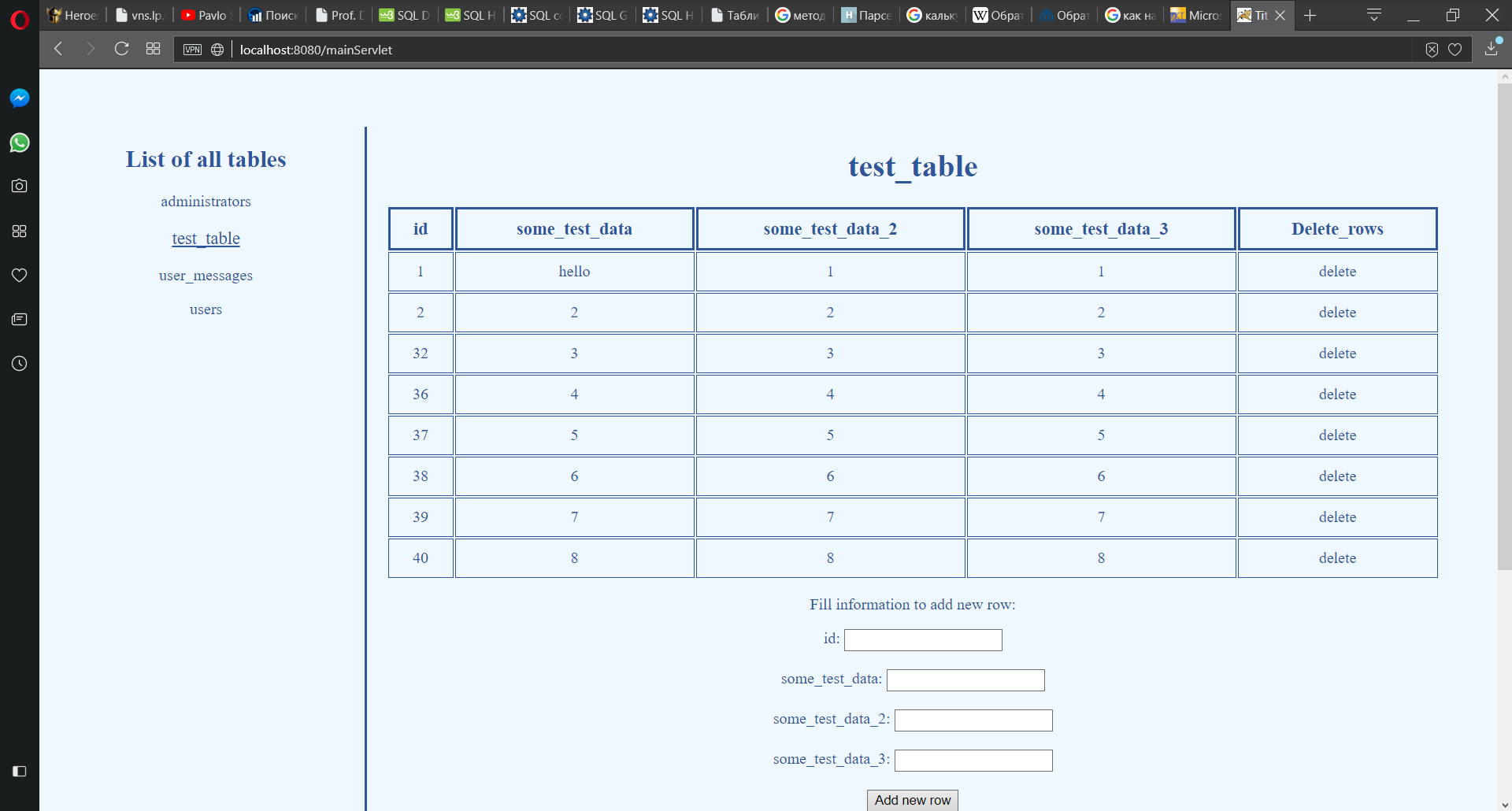


Рис. 2. Активна таблиця test\_table

Для того, щоб додати новий запис користувачу потрібно заповнити всі поля у формі додавання та натиснути кнопку “Add new row”.

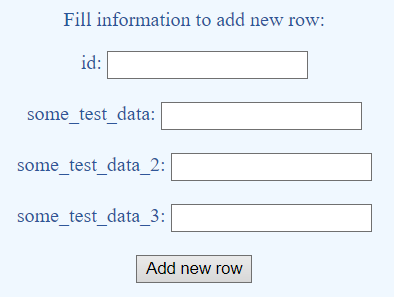


Рис. 3. Форма для додавання нового запису в таблицю

Для видалення запису з таблиці достатньо натиснути на клітинку “delete” навпроти відповідного рядка (рис. 4), а для видалення всіх записів потрібно натиснути на клітинку “Delete\_rows”.

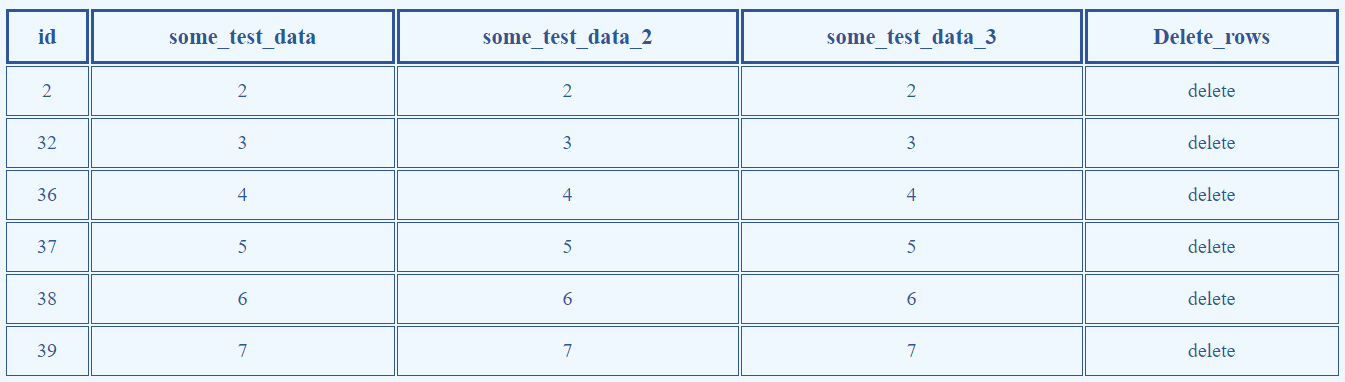


Рис. 4. Таблиця test\_table після видалення першого та останнього рядків

Вихідний код

//Код сервлету

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.sql.\*;

import java.util.Vector;

@WebServlet(name = "mainServlet")

public class MainServlet extends HttpServlet {

private String URL = "jdbc:mysql://localhost:3306/lab\_2\_db";

private String USERNAME = "mornado";

private String PASSWORD = "1224";

private ResultSet rs;

private Connection connection;

private int ActiveTableIdx = 0;

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

String TableName = req.getParameter("LoadName");

String Query = "INSERT INTO ";

Vector<String> Tables = new Vector<String>();

Vector<String> TableColumns = new Vector<String>();

Vector<Vector<String>> TableRows = new Vector<Vector<String>>();

try {

Class.forName("com.mysql.jdbc.Driver");

connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);

Statement statement = connection.createStatement();

rs = statement.executeQuery(

"SELECT table\_name FROM information\_schema.tables WHERE table\_schema = 'lab\_2\_db'");

while(rs.next()){

Tables.add(rs.getString(1));

}

if(TableName == null) ActiveTableIdx = 0;

else ActiveTableIdx = Tables.indexOf(TableName);

Query += "" + Tables.get(ActiveTableIdx) + " (";

rs = statement.executeQuery(

"SELECT COLUMN\_NAME FROM information\_schema.columns " +

"WHERE TABLE\_SCHEMA='lab\_2\_db' AND TABLE\_NAME ='"+ Tables.get(ActiveTableIdx) +"'");

while(rs.next()){

TableColumns.add(rs.getString(1));

Query += rs.getString(1) + ",";

}

Query = Query.substring(0, Query.length() - 1);

Query += ") VALUES ";

String del = req.getParameter("DeleteAll");

String rowDel = req.getParameter("DeleteIdx");

if (rowDel != null && rowDel != ""){

int deleteRow = Integer.parseInt(rowDel);

String rowdata = "";

rs = statement.executeQuery("SELECT \* FROM " + Tables.get(ActiveTableIdx));

for (int i = 0; rs.next(); i++){

if (i == deleteRow)

rowdata = rs.getString(1);

}

statement.executeUpdate("DELETE FROM " + Tables.get(ActiveTableIdx) +

" WHERE " + TableColumns.get(0) + "=" + rowdata);

}

if (del != null && del != ""){

int deleteAll = Integer.parseInt(del);

if (deleteAll == 1)

statement.executeUpdate("DELETE FROM " + Tables.get(ActiveTableIdx));

}

String params = "(";

boolean HaveAllParams = true;

for (int i = 0; i < TableColumns.size(); ++i){

String str = req.getParameter("" + i);

if (str == null || str == ""){

HaveAllParams = false;

break;

}

if (i == 0) params += "'" + str + "'";

else params += ", '" + str + "'";

}

params += ")";

Query += params;

if (HaveAllParams){

statement.executeUpdate(Query);

}

rs = statement.executeQuery("SELECT \* FROM " + Tables.get(ActiveTableIdx));

for (int i = 0; rs.next(); ++i){

TableRows.add(new Vector<String>());

for (int j = 1; j <= TableColumns.size(); ++j){

TableRows.get(i).add(rs.getString(j));

}

}

}catch (SQLException e){

e.printStackTrace();

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

req.setAttribute("Tables", Tables);

req.setAttribute("TableColumns", TableColumns);

req.setAttribute("TableRows", TableRows);

req.setAttribute("ActiveTableIdx", ActiveTableIdx);

req.getRequestDispatcher("WEB-INF/servletPage.jsp").forward(req, resp);

}

}

//Код jsp сторінки

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<html>

<head>

<title>Title</title>

<style>

<%@include file="/WEB-INF/styles.css"%>

</style>

<script>

<%@include file="method.js"%>

</script>

</head>

<body>

<div id="left">

<h2>List of all tables</h2>

<c:forEach var="TableName" items="${Tables}">

<p class="link" onclick="SetTable(id)" id="${TableName}">${TableName}</p>

</c:forEach>

</div>

<div id="center">

<h1 id="table\_name">${Tables.get(ActiveTableIdx)}</h1>

<div id="table">

<table class="table">

<tr>

<c:forEach var="Column" items="${TableColumns}">

<th>${Column}</th>

</c:forEach>

<th onclick="DeleteAllRows()">Delete\_rows</th>

</tr>

<c:forEach var="Row" items="${TableRows}">

<tr>

<c:forEach var="Cell" items="${Row}">

<td>${Cell}</td>

</c:forEach>

<td class="delete" onclick="DeleteRow(${TableRows.indexOf(Row)})">delete</td>

</tr>

</c:forEach>

</table>

</div>

<form action="mainServlet" name="hiddenForm" method="get" hidden>

<input name="action" type="text">

<input name="LoadName" type="text">

<input name="DeleteIdx" type="text">

<input name="DeleteAll" type="number">

</form>

<div id="INSERT">

<p>Fill information to add new row:</p>

<form action="mainServlet" name="InsertForm" method="get">

<c:forEach var="Column" items="${TableColumns}">

<div>${TableColumns.get(TableColumns.indexOf(Column))}:

<input name="${TableColumns.indexOf(Column)}" type="text">

</div><br>

</c:forEach>

<button type="submit">Add new row</button>

<input name="LoadName" type="text" value="${Tables.get(ActiveTableIdx)}" hidden>

</form>

</div>

</div>

</body>

</html>

//Методи JavaScript, якими опрацьовуються дії користувача

function SetTable(Text) {

document.getElementsByTagName("input")[1].value = Text;

document.getElementsByTagName("input")[3].value = 0;

document.forms[0].submit();

}

function DeleteRow(idx) {

if (!confirm("Delete this row?")) return;

document.getElementsByTagName("input")[1].value =

document.getElementById("table\_name").innerText;

document.getElementsByTagName("input")[2].value = idx;

document.getElementsByTagName("input")[3].value = 0;

document.forms[0].submit();

}

function DeleteAllRows() {

if (!confirm("Delete all rows?")) return;

document.getElementsByTagName("input")[1].value =

document.getElementById("table\_name").innerText;

document.getElementsByTagName("input")[3].value = 1;

document.forms[0].submit();

}