**ADVANCED JAVA PRACTICALS**

**PRACTICAL NO 1 :**

**Develop the presentation layer of Library Management software application with suitable menus.**

**Library.java**

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import java.awt.Color;

import java.awt.Font;

import javax.swing.JButton;

import javax.swing.LayoutStyle.ComponentPlacement;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class Library extends JFrame {

static Library frame;

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frame= new Library();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public Library() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblLibraryManagement = new JLabel("Library Management - JavaTpoint");

lblLibraryManagement.setFont(new Font("Tahoma", Font.PLAIN, 18));

lblLibraryManagement.setForeground(Color.GRAY);

JButton btnAdminLogin = new JButton("Admin Login");

btnAdminLogin.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

AdminLogin.main(new String[]{});

frame.dispose();

}

});

btnAdminLogin.setFont(new Font("Tahoma", Font.PLAIN, 15));

JButton btnLibrarianLogin = new JButton("Librarian Login");

btnLibrarianLogin.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

LibrarianLogin.main(new String[]{});

}

});

btnLibrarianLogin.setFont(new Font("Tahoma", Font.PLAIN, 15));

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(64)

.addComponent(lblLibraryManagement))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(140)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.TRAILING, false)

.addComponent(btnLibrarianLogin, Alignment.LEADING, GroupLayout.DEFAULT\_SIZE, GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(btnAdminLogin, Alignment.LEADING, GroupLayout.DEFAULT\_SIZE, 135, Short.MAX\_VALUE))))

.addContainerGap(95, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addContainerGap()

.addComponent(lblLibraryManagement)

.addGap(32)

.addComponent(btnAdminLogin, GroupLayout.PREFERRED\_SIZE, 52, GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(ComponentPlacement.UNRELATED)

.addComponent(btnLibrarianLogin, GroupLayout.PREFERRED\_SIZE, 53, GroupLayout.PREFERRED\_SIZE)

.addContainerGap(70, Short.MAX\_VALUE))

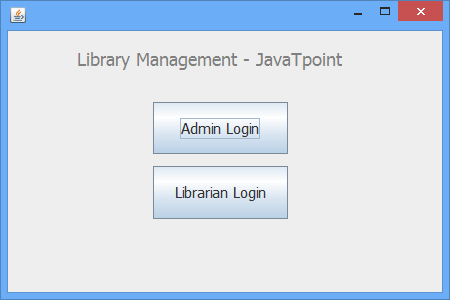
);

contentPane.setLayout(gl\_contentPane);

}

}

**Output:**



**AdminLogin.java**

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import java.awt.Color;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import javax.swing.JPasswordField;

public class AdminLogin extends JFrame {

static AdminLogin frame;

private JPanel contentPane;

private JTextField textField;

private JPasswordField passwordField;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frame = new AdminLogin();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public AdminLogin() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblAdminLoginForm = new JLabel("Admin Login Form");

lblAdminLoginForm.setForeground(Color.GRAY);

lblAdminLoginForm.setFont(new Font("Tahoma", Font.PLAIN, 18));

JLabel lblEnterName = new JLabel("Enter Name:");

JLabel lblEnterPassword = new JLabel("Enter Password:");

textField = new JTextField();

textField.setColumns(10);

JButton btnLogin = new JButton("Login");

btnLogin.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String name=textField.getText();

String password=String.valueOf(passwordField.getPassword());

if(name.equals("admin")&&password.equals("admin123")){

AdminSuccess.main(new String[]{});

frame.dispose();

}else{

JOptionPane.showMessageDialog(AdminLogin.this, "Sorry, Username or Password Error","Login Error!", JOptionPane.ERROR\_MESSAGE);

textField.setText("");

passwordField.setText("");

}

}

});

passwordField = new JPasswordField();

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.TRAILING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(124)

.addComponent(lblAdminLoginForm))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(19)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblEnterName)

.addComponent(lblEnterPassword))

.addGap(47)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING, false)

.addComponent(passwordField)

.addComponent(textField, GroupLayout.DEFAULT\_SIZE, 172, Short.MAX\_VALUE))))

.addContainerGap(107, Short.MAX\_VALUE))

.addGroup(gl\_contentPane.createSequentialGroup()

.addContainerGap(187, Short.MAX\_VALUE)

.addComponent(btnLogin, GroupLayout.PREFERRED\_SIZE, 86, GroupLayout.PREFERRED\_SIZE)

.addGap(151))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addComponent(lblAdminLoginForm)

.addGap(26)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblEnterName)

.addComponent(textField, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(28)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblEnterPassword)

.addComponent(passwordField, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addComponent(btnLogin, GroupLayout.PREFERRED\_SIZE, 37, GroupLayout.PREFERRED\_SIZE)

.addContainerGap(80, Short.MAX\_VALUE))

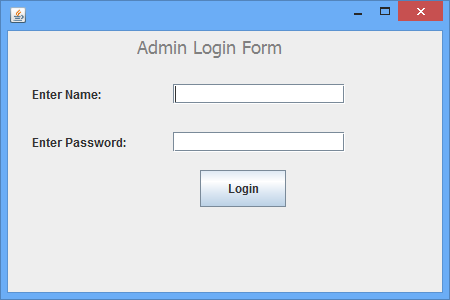
);

contentPane.setLayout(gl\_contentPane);

}

}

**Output:**



**AdminSuccess.java**

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import java.awt.Color;

import java.awt.Font;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import javax.swing.LayoutStyle.ComponentPlacement;

import java.sql.\*;

public class AdminSuccess extends JFrame {

static AdminSuccess frame;

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frame = new AdminSuccess();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public AdminSuccess() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 371);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblAdminSection = new JLabel("Admin Section");

lblAdminSection.setFont(new Font("Tahoma", Font.PLAIN, 22));

lblAdminSection.setForeground(Color.GRAY);

JButton btnNewButton = new JButton("Add Librarian");

btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 15));

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

LibrarianForm.main(new String[]{});

frame.dispose();

}

});

JButton btnViewLibrarian = new JButton("View Librarian");

btnViewLibrarian.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

ViewLibrarian.main(new String[]{});

}

});

btnViewLibrarian.setFont(new Font("Tahoma", Font.PLAIN, 15));

JButton btnDeleteLibrarian = new JButton("Delete Librarian");

btnDeleteLibrarian.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

DeleteLibrarian.main(new String[]{});

frame.dispose();

}

});

btnDeleteLibrarian.setFont(new Font("Tahoma", Font.PLAIN, 15));

JButton btnLogout = new JButton("Logout");

btnLogout.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

Library.main(new String[]{});

frame.dispose();

}

});

btnLogout.setFont(new Font("Tahoma", Font.PLAIN, 15));

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.TRAILING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addContainerGap(150, Short.MAX\_VALUE)

.addComponent(lblAdminSection, GroupLayout.PREFERRED\_SIZE, 151, GroupLayout.PREFERRED\_SIZE)

.addGap(123))

.addGroup(Alignment.LEADING, gl\_contentPane.createSequentialGroup()

.addGap(134)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(btnLogout, GroupLayout.PREFERRED\_SIZE, 181, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnDeleteLibrarian, GroupLayout.PREFERRED\_SIZE, 181, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnViewLibrarian, GroupLayout.PREFERRED\_SIZE, 181, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnNewButton, GroupLayout.PREFERRED\_SIZE, 181, GroupLayout.PREFERRED\_SIZE))

.addContainerGap(109, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addComponent(lblAdminSection, GroupLayout.PREFERRED\_SIZE, 40, GroupLayout.PREFERRED\_SIZE)

.addGap(11)

.addComponent(btnNewButton, GroupLayout.PREFERRED\_SIZE, 49, GroupLayout.PREFERRED\_SIZE)

.addGap(18)

.addComponent(btnViewLibrarian, GroupLayout.PREFERRED\_SIZE, 49, GroupLayout.PREFERRED\_SIZE)

.addGap(18)

.addComponent(btnDeleteLibrarian, GroupLayout.PREFERRED\_SIZE, 49, GroupLayout.PREFERRED\_SIZE)

.addGap(18)

.addComponent(btnLogout, GroupLayout.PREFERRED\_SIZE, 49, GroupLayout.PREFERRED\_SIZE)

.addContainerGap(21, Short.MAX\_VALUE))

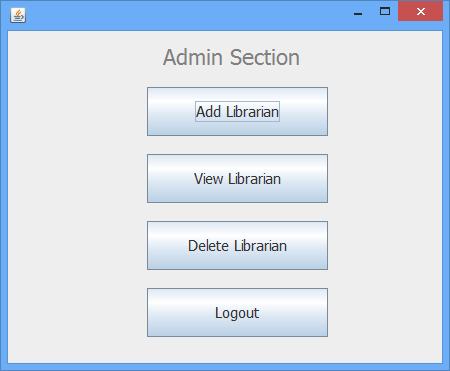
);

contentPane.setLayout(gl\_contentPane);

}

}

**Output:**



**LibrarianLogin.java**

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import java.awt.Color;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import javax.swing.JPasswordField;

public class LibrarianLogin extends JFrame {

static LibrarianLogin frame;

private JPanel contentPane;

private JTextField textField;

private JPasswordField passwordField;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frame = new LibrarianLogin();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public LibrarianLogin() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblAdminLoginForm = new JLabel("Librarian Login Form");

lblAdminLoginForm.setForeground(Color.GRAY);

lblAdminLoginForm.setFont(new Font("Tahoma", Font.PLAIN, 18));

JLabel lblEnterName = new JLabel("Enter Name:");

JLabel lblEnterPassword = new JLabel("Enter Password:");

textField = new JTextField();

textField.setColumns(10);

JButton btnLogin = new JButton("Login");

btnLogin.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String name=textField.getText();

String password=String.valueOf(passwordField.getPassword());

//System.out.println(name+" "+password);

if(LibrarianDao.validate(name, password)){

LibrarianSuccess.main(new String[]{});

frame.dispose();

}else{

JOptionPane.showMessageDialog(LibrarianLogin.this, "Sorry, Username or Password Error","Login Error!", JOptionPane.ERROR\_MESSAGE);

textField.setText("");

passwordField.setText("");

}

}

});

passwordField = new JPasswordField();

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.TRAILING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(124)

.addComponent(lblAdminLoginForm))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(19)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblEnterName)

.addComponent(lblEnterPassword))

.addGap(47)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING, false)

.addComponent(passwordField)

.addComponent(textField, GroupLayout.DEFAULT\_SIZE, 172, Short.MAX\_VALUE))))

.addContainerGap(107, Short.MAX\_VALUE))

.addGroup(gl\_contentPane.createSequentialGroup()

.addContainerGap(187, Short.MAX\_VALUE)

.addComponent(btnLogin, GroupLayout.PREFERRED\_SIZE, 86, GroupLayout.PREFERRED\_SIZE)

.addGap(151))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addComponent(lblAdminLoginForm)

.addGap(26)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblEnterName)

.addComponent(textField, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(28)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblEnterPassword)

.addComponent(passwordField, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addComponent(btnLogin, GroupLayout.PREFERRED\_SIZE, 37, GroupLayout.PREFERRED\_SIZE)

.addContainerGap(80, Short.MAX\_VALUE))

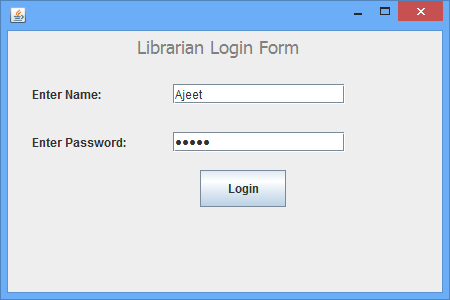
);

contentPane.setLayout(gl\_contentPane);

}

}

**Output:**



**LibrarySuccess.java**

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import java.awt.Font;

import java.awt.Color;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class LibrarianSuccess extends JFrame {

static LibrarianSuccess frame;

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frame = new LibrarianSuccess();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public LibrarianSuccess() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 433);

contentPane = new JPanel();

contentPane.setForeground(Color.GRAY);

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblLibrarianSection = new JLabel("Librarian Section - JavaTpoint");

lblLibrarianSection.setFont(new Font("Tahoma", Font.PLAIN, 22));

JButton btnNewButton = new JButton("Add Books");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

BooksForm.main(new String[]{});

frame.dispose();

}

});

btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 13));

JButton btnViewBooks = new JButton("View Books");

btnViewBooks.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

ViewBooks.main(new String[]{});

}

});

btnViewBooks.setFont(new Font("Tahoma", Font.PLAIN, 13));

JButton btnIssueBook = new JButton("Issue Book");

btnIssueBook.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

IssueBookForm.main(new String[]{});

frame.dispose();

}

});

btnIssueBook.setFont(new Font("Tahoma", Font.PLAIN, 13));

JButton btnViewIssuedBooks = new JButton("View Issued Books");

btnViewIssuedBooks.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

ViewIssuedBooks.main(new String[]{});

}

});

btnViewIssuedBooks.setFont(new Font("Tahoma", Font.PLAIN, 13));

JButton btnReturnBook = new JButton("Return Book");

btnReturnBook.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

ReturnBook.main(new String[]{});

frame.dispose();

}

});

btnReturnBook.setFont(new Font("Tahoma", Font.PLAIN, 13));

JButton btnLogout = new JButton("Logout");

btnLogout.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Library.main(new String[]{});

frame.dispose();

}

});

btnLogout.setFont(new Font("Tahoma", Font.PLAIN, 13));

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(Alignment.TRAILING, gl\_contentPane.createSequentialGroup()

.addContainerGap(81, Short.MAX\_VALUE)

.addComponent(lblLibrarianSection)

.addGap(54))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(132)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(btnLogout, GroupLayout.PREFERRED\_SIZE, 191, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnReturnBook, GroupLayout.PREFERRED\_SIZE, 191, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnViewIssuedBooks, GroupLayout.PREFERRED\_SIZE, 191, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnIssueBook, GroupLayout.PREFERRED\_SIZE, 191, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnViewBooks, GroupLayout.PREFERRED\_SIZE, 191, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnNewButton, GroupLayout.PREFERRED\_SIZE, 191, GroupLayout.PREFERRED\_SIZE))

.addContainerGap(101, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addContainerGap()

.addComponent(lblLibrarianSection)

.addGap(18)

.addComponent(btnNewButton, GroupLayout.PREFERRED\_SIZE, 37, GroupLayout.PREFERRED\_SIZE)

.addGap(18)

.addComponent(btnViewBooks, GroupLayout.PREFERRED\_SIZE, 37, GroupLayout.PREFERRED\_SIZE)

.addGap(18)

.addComponent(btnIssueBook, GroupLayout.PREFERRED\_SIZE, 37, GroupLayout.PREFERRED\_SIZE)

.addGap(18)

.addComponent(btnViewIssuedBooks, GroupLayout.PREFERRED\_SIZE, 37, GroupLayout.PREFERRED\_SIZE)

.addGap(18)

.addComponent(btnReturnBook, GroupLayout.PREFERRED\_SIZE, 37, GroupLayout.PREFERRED\_SIZE)

.addGap(18)

.addComponent(btnLogout, GroupLayout.PREFERRED\_SIZE, 37, GroupLayout.PREFERRED\_SIZE)

.addContainerGap(16, Short.MAX\_VALUE))

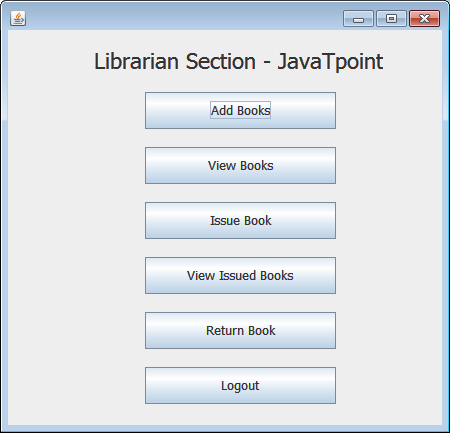
);

contentPane.setLayout(gl\_contentPane);

}

}

**Output:**



**PRACTICAL NO 2:**

**Design suitable database for Library Management System.**

**LibCreate.java**

import java.sql.\*;

public class LibCreate {

//JDBC driver name and database URL

static final String JDBC\_DRIVER="com.mysql.jdbc.Driver";

static final String DB\_URL="jdbc:mysql://LOCALHOST:3306/students";

//Database credentials

//static final String USER="root";

//static final String PASS="1234";

public static void main(String[] args){

Connection conn = null;

Statement stmt = null;

try{

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

System.out.println("Connecting to a selected database...");

conn = DriverManager.getConnection("jdbc:mysql://LOCALHOST:3306/students","root","12345");

System.out.println("Connected database successsfully...");

System.out.println("Creating table in given database...");

stmt = conn.createStatement();

String sql = "CREATE TABLE Librarian " +

"(name VARCHAR(255)," +

"password VARCHAR(255)," +

"email VARCHAR(255)," +

"address VARCHAR(255)," +

"city VARCHAR(255)," +

"contact VARCHAR(255))";

stmt.executeUpdate(sql);

System.out.println("Created table in given database...");

}catch(SQLException se){

se.printStackTrace();

}catch(Exception e){

e.printStackTrace();

}finally{

try{

if(stmt!=null)

conn.close();

}catch(SQLException se){

se.printStackTrace();

}//end finally try

}//end try

System.out.println("Goodbye!");

}//end main

}//end LibCreate

Steps to connect sql server

Goto project

Libraray

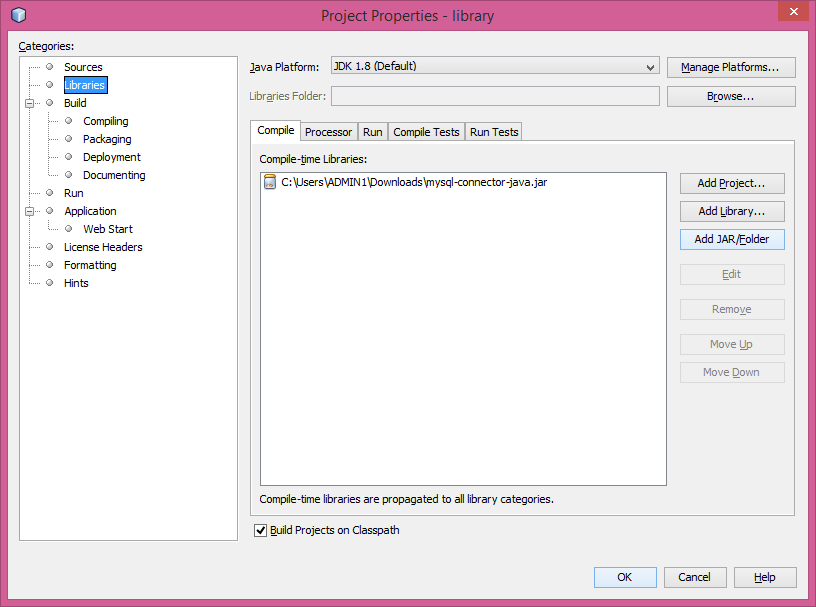
Goto project properties

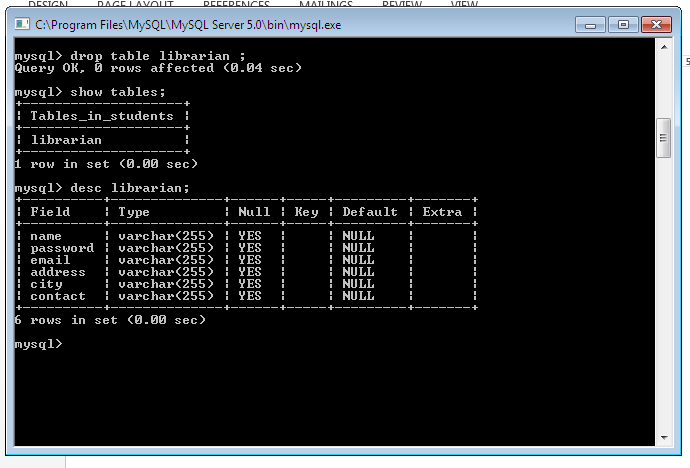
Add jar files

Select mysql java connector

Ok

Run shift +f6



****

**PRACTICA NO 3:**

**Develop business logic layer for Library Management System.**

**LibrarianDao.java**

import java.sql.\*;

public class LibrarianDao {

public static int save(String name,String password,String email,String address,String city,String contact){

int status=0;

try{

Connection con=DB.getConnection();

PreparedStatement ps=con.prepareStatement("insert into librarian(name,password,email,address,city,contact) values(?,?,?,?,?,?)");

ps.setString(1,name);

ps.setString(2,password);

ps.setString(3,email);

ps.setString(4,address);

ps.setString(5,city);

ps.setString(6,contact);

status=ps.executeUpdate();

con.close();

}catch(Exception e){System.out.println(e);}

return status;

}

public static int delete(int id){

int status=0;

try{

Connection con=DB.getConnection();

PreparedStatement ps=con.prepareStatement("delete from librarian where id=?");

ps.setInt(1,id);

status=ps.executeUpdate();

con.close();

}catch(Exception e){System.out.println(e);}

return status;

}

public static boolean validate(String name,String password){

boolean status=false;

try{

Connection con=DB.getConnection();

PreparedStatement ps=con.prepareStatement("select \* from librarian where name=? and password=?");

ps.setString(1,name);

ps.setString(2,password);

ResultSet rs=ps.executeQuery();

status=rs.next();

con.close();

}catch(Exception e){System.out.println(e);}

return status;

}

}

**LibrarianForm.java**

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import java.awt.Color;

import javax.swing.JTextField;

import javax.swing.JPasswordField;

import javax.swing.LayoutStyle.ComponentPlacement;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class LibrarianForm extends JFrame {

static LibrarianForm frame;

private JPanel contentPane;

private JTextField textField;

private JTextField textField\_1;

private JTextField textField\_2;

private JTextField textField\_3;

private JTextField textField\_4;

private JPasswordField passwordField;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frame = new LibrarianForm();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public LibrarianForm() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 450);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblAddLibrarian = new JLabel("Add Librarian");

lblAddLibrarian.setForeground(Color.DARK\_GRAY);

lblAddLibrarian.setFont(new Font("Tahoma", Font.PLAIN, 22));

JLabel lblName = new JLabel("Name:");

JLabel lblPassword = new JLabel("Password:");

JLabel lblEmail = new JLabel("Email:");

JLabel lblAddress = new JLabel("Address:");

JLabel lblCity = new JLabel("City:");

JLabel lblContactNo = new JLabel("Contact No:");

//JLabel lblid=new JLabel("ID");

textField = new JTextField();

textField.setColumns(10);

textField = new JTextField();

textField.setColumns(10);

textField\_1 = new JTextField();

textField\_1.setColumns(10);

textField\_2 = new JTextField();

textField\_2.setColumns(10);

textField\_3 = new JTextField();

textField\_3.setColumns(10);

textField\_4 = new JTextField();

textField\_4.setColumns(10);

passwordField = new JPasswordField();

JButton btnNewButton = new JButton("Add Librarian");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String id=textField.getText();

String name=textField.getText();

String password=String.valueOf(passwordField.getPassword());

String email=textField\_1.getText();

String address=textField\_2.getText();

String city=textField\_3.getText();

String contact=textField\_4.getText();

int i=LibrarianDao.save(name, password, email, address, city, contact);

if(i>0){

JOptionPane.showMessageDialog(LibrarianForm.this,"Librarian added successfully!");

AdminSuccess.main(new String[]{});

frame.dispose();

}else{

JOptionPane.showMessageDialog(LibrarianForm.this,"Sorry, unable to save!");

}

}

});

btnNewButton.setForeground(Color.DARK\_GRAY);

JButton btnBack = new JButton("Back");

btnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

AdminSuccess.main(new String[]{});

frame.dispose();

}

});

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.TRAILING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(20)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING, false)

.addComponent(lblPassword, GroupLayout.DEFAULT\_SIZE, 62, Short.MAX\_VALUE)

.addComponent(lblName)

.addComponent(lblEmail, GroupLayout.PREFERRED\_SIZE, 31, GroupLayout.PREFERRED\_SIZE)

.addComponent(lblAddress, GroupLayout.DEFAULT\_SIZE, GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblCity, GroupLayout.PREFERRED\_SIZE, 31, GroupLayout.PREFERRED\_SIZE)

.addComponent(lblContactNo, GroupLayout.DEFAULT\_SIZE, GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(58)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING, false)

.addComponent(textField\_4, GroupLayout.DEFAULT\_SIZE, 177, Short.MAX\_VALUE)

.addComponent(textField\_3, GroupLayout.DEFAULT\_SIZE, 177, Short.MAX\_VALUE)

.addComponent(textField\_2, GroupLayout.DEFAULT\_SIZE, 177, Short.MAX\_VALUE)

.addComponent(textField\_1, GroupLayout.DEFAULT\_SIZE, 177, Short.MAX\_VALUE)

.addComponent(textField, GroupLayout.DEFAULT\_SIZE, 177, Short.MAX\_VALUE)

.addComponent(passwordField))

.addContainerGap(107, Short.MAX\_VALUE))

.addGroup(gl\_contentPane.createSequentialGroup()

.addContainerGap(151, Short.MAX\_VALUE)

.addComponent(lblAddLibrarian)

.addGap(144))

.addGroup(gl\_contentPane.createSequentialGroup()

.addContainerGap(160, Short.MAX\_VALUE)

.addComponent(btnNewButton, GroupLayout.PREFERRED\_SIZE, 131, GroupLayout.PREFERRED\_SIZE)

.addGap(133))

.addGroup(gl\_contentPane.createSequentialGroup()

.addContainerGap(200, Short.MAX\_VALUE)

.addComponent(btnBack)

.addGap(169))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addComponent(lblAddLibrarian)

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addComponent(lblName)

.addGap(18)

.addComponent(lblPassword))

.addGroup(gl\_contentPane.createSequentialGroup()

.addComponent(textField, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(ComponentPlacement.UNRELATED)

.addComponent(passwordField, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblEmail)

.addComponent(textField\_1, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblAddress)

.addComponent(textField\_2, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblCity)

.addComponent(textField\_3, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblContactNo)

.addComponent(textField\_4, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addComponent(btnNewButton, GroupLayout.PREFERRED\_SIZE, 36, GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(ComponentPlacement.RELATED, 57, Short.MAX\_VALUE)

.addComponent(btnBack)

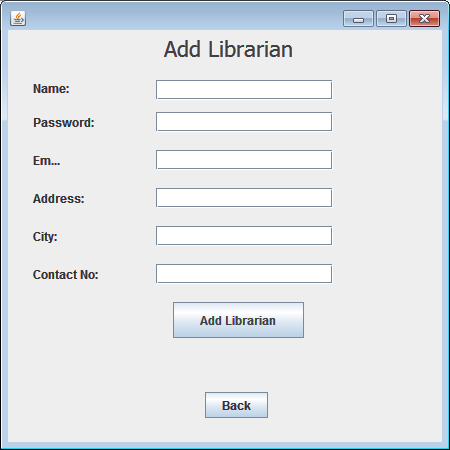
.addGap(19))

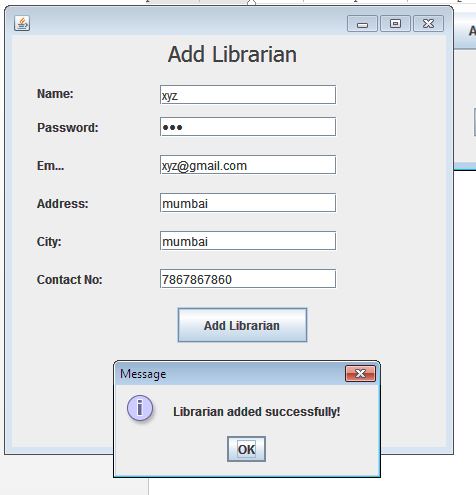
);

contentPane.setLayout(gl\_contentPane);

}

}





**ViewLibrarian.java**

import java.awt.BorderLayout;

import java.awt.EventQueue;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.border.EmptyBorder;

import javax.swing.JTable;

public class ViewLibrarian extends JFrame {

private JPanel contentPane;

private JTable table;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

ViewLibrarian frame = new ViewLibrarian();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public ViewLibrarian() {

setDefaultCloseOperation(JFrame.HIDE\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

contentPane.setLayout(new BorderLayout(0, 0));

setContentPane(contentPane);

String data[][]=null;

String column[]=null;

try{

Connection con=DB.getConnection();

PreparedStatement ps=con.prepareStatement("select \* from librarian",ResultSet.TYPE\_SCROLL\_SENSITIVE,ResultSet.CONCUR\_UPDATABLE);

ResultSet rs=ps.executeQuery();

ResultSetMetaData rsmd=rs.getMetaData();

int cols=rsmd.getColumnCount();

column=new String[cols];

for(int i=1;i<=cols;i++){

column[i-1]=rsmd.getColumnName(i);

}

rs.last();

int rows=rs.getRow();

rs.beforeFirst();

data=new String[rows][cols];

int count=0;

while(rs.next()){

for(int i=1;i<=cols;i++){

data[count][i-1]=rs.getString(i);

}

count++;

}

con.close();

}catch(Exception e){System.out.println(e);}

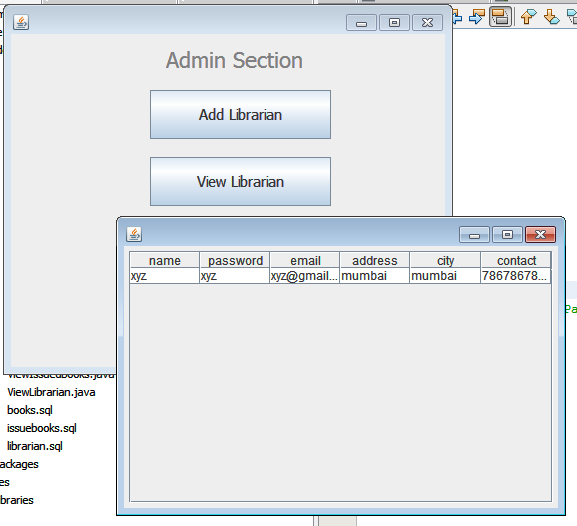
table = new JTable(data,column);

JScrollPane sp=new JScrollPane(table);

contentPane.add(sp, BorderLayout.CENTER);

}

}



**DeleteLibrarian.java**

**i**mport java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.Font;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class DeleteLibrarian extends JFrame {

static DeleteLibrarian frame;

private JPanel contentPane;

private JTextField textField;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frame = new DeleteLibrarian();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public DeleteLibrarian() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblEnterId = new JLabel("Enter Id:");

textField = new JTextField();

textField.setColumns(10);

JButton btnDelete = new JButton("Delete");

btnDelete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String sid=textField.getText();

if(sid==null||sid.trim().equals("")){

JOptionPane.showMessageDialog(DeleteLibrarian.this,"Id can't be blank");

}else{

int id=Integer.parseInt(sid);

int i=LibrarianDao.delete(id);

if(i>0){

JOptionPane.showMessageDialog(DeleteLibrarian.this,"Record deleted successfully!");

}else{

JOptionPane.showMessageDialog(DeleteLibrarian.this,"Unable to delete given id!");

}

}

}

});

btnDelete.setFont(new Font("Tahoma", Font.PLAIN, 13));

JButton btnNewButton = new JButton("Back");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

AdminSuccess.main(new String[]{});

frame.dispose();

}

});

btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 13));

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(39)

.addComponent(lblEnterId)

.addGap(57)

.addComponent(textField, GroupLayout.PREFERRED\_SIZE, 178, GroupLayout.PREFERRED\_SIZE)

.addContainerGap(107, Short.MAX\_VALUE))

.addGroup(Alignment.TRAILING, gl\_contentPane.createSequentialGroup()

.addContainerGap(175, Short.MAX\_VALUE)

.addComponent(btnDelete, GroupLayout.PREFERRED\_SIZE, 109, GroupLayout.PREFERRED\_SIZE)

.addGap(140))

.addGroup(Alignment.TRAILING, gl\_contentPane.createSequentialGroup()

.addContainerGap(322, Short.MAX\_VALUE)

.addComponent(btnNewButton, GroupLayout.PREFERRED\_SIZE, 92, GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(19)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(textField, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(lblEnterId))

.addGap(33)

.addComponent(btnDelete, GroupLayout.PREFERRED\_SIZE, 33, GroupLayout.PREFERRED\_SIZE)

.addGap(43)

.addComponent(btnNewButton)

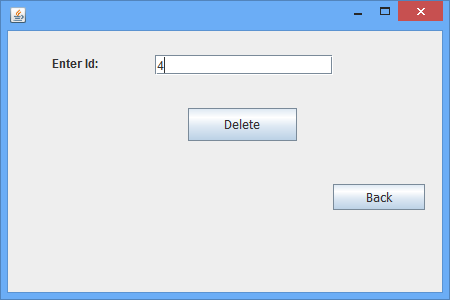
.addContainerGap(78, Short.MAX\_VALUE))

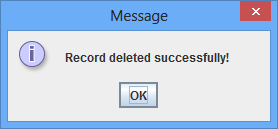
);

contentPane.setLayout(gl\_contentPane);

}

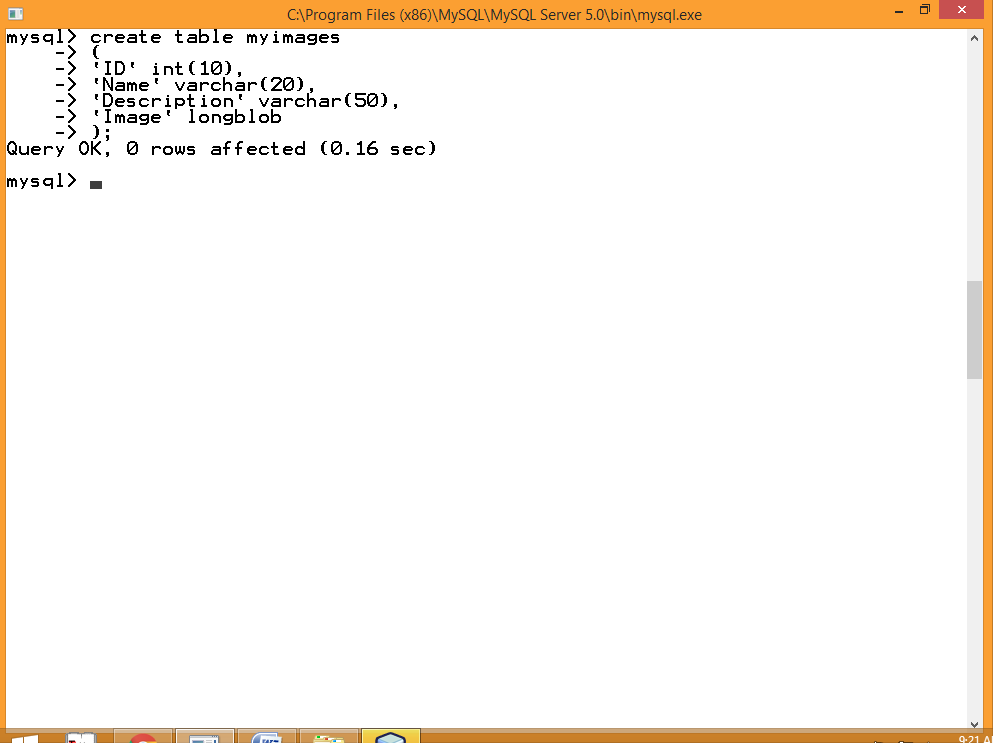
}





**PRACTICAL NO 4:**

**Develop Java application to store image in a database as well as retrieve image from database.**

****

**Work1.java**

import java.awt.Color;

import java.awt.Image;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.swing.\*;

public class Imagestore extends JFrame{

JButton button ;

JLabel label;

JTextField jtf;

public Imagestore(){

super("retrieve image from database in java");

button = new JButton("Retrieve");

button.setBounds(250,300,100,40);

jtf = new JTextField();

jtf.setBounds(360,310,100,20);

label = new JLabel();

label.setBounds(10,10,670,250);

add(button);

add(label);

add(jtf);

button.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try{

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/db\_images","root","1234");

Statement st = con.createStatement();

ResultSet rs = st.executeQuery("select \* from myimages where ID = '"+jtf.getText()+"'");

if(rs.next()){

byte[] img = rs.getBytes("Image");

//Resize The ImageIcon

ImageIcon image = new ImageIcon(img);

Image im = image.getImage();

Image myImg = im.getScaledInstance(label.getWidth(), label.getHeight(),Image.SCALE\_SMOOTH);

ImageIcon newImage = new ImageIcon(myImg);

label.setIcon(newImage);

}

else{

JOptionPane.showMessageDialog(null, "No Data");

}

}catch(Exception ex){

ex.printStackTrace();

}

}

});

setLayout(null);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

getContentPane().setBackground(Color.decode("#bdb76b"));

setLocationRelativeTo(null);

setSize(700,400);

setVisible(true);

}

public static void main(String[] args){

new Imagestore();

}

}

**PRACTICAL NO -5**

**Write a Java Application to demonstrate servlet life cycle.**

**ServletLifeCycle.java**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class ServletLifeCycle extends HttpServlet

{

public ServletLifeCycle()

{

System.out.println(“Am from default constructor”);

}

public void init(ServletConfig config)

{

System.out.println(“Am from Init method…!”);

}

public void doGet(HttpServletReuest req, HttpServletResponse res)throws ServletException, IOException

{

res.setContentType(“text/html”);

PrintWriter pw = res.getWriter();

pw.println(“I am from doGet method”);

pw.close();

}

public void destroy()

{

System.out.println(“Am from Destroy methods”);

}

}

**Web.xml**

<web-app>

<servlet>

<servlet-name>second</servlet-name>

<servlet-class>java4s.ServletLifeCycle</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>second</servlet-name>

<url-pattern>/lifecycle1</url-pattern>

</servlet-mapping>

</web-app>



**PRACTICAL NO 6**

**Design database for employee administration. Develop servlet(s) to perform CRUD operations.**

Index.html

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<h1>Add New Employee</h1>

<form action="SaveServlet" method="post">  <table>

<tr><td>Name:</td><td><input type="text" name="name"/></td></tr>

<tr><td>Password:</td><td><input type="password" name="password"/></td></tr>

<tr><td>Email:</td><td><input type="email" name="email"/></td></tr>

<tr><td>Country:</td><td>

<select name="country" style="width:150px">

<option>India</option>

<option>USA</option>

<option>UK</option>

<option>Other</option>

</select>

</td></tr>

<tr><td colspan="2"><input type="submit" value="Save Employee"/></td></tr>

</table>

</form>

<br/>

<a href="ViewServlet">view employees</a>

</body>

</html>

Emp.java

**public** **class** Emp {

**private** **int** id;

**private** String name,password,email,country;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getCountry() {

**return** country;

}

**public** **void** setCountry(String country) {

**this**.country = country;

}

}

EmpDao.java

**import** java.util.\*;

**import** java.sql.\*;

**public** **class** EmpDao {

**public** **static** Connection getConnection(){

        Connection con=**null**;

**try**{

            Class.forName("oracle.jdbc.driver.OracleDriver");

            con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","oracle");

        }**catch**(Exception e){System.out.println(e);}

**return** con;

    }

**public** **static** **int** save(Emp e){

**int** status=0;

**try**{

            Connection con=EmpDao.getConnection();

            PreparedStatement ps=con.prepareStatement(

                         "insert into user905(name,password,email,country) values (?,?,?,?)");

            ps.setString(1,e.getName());

            ps.setString(2,e.getPassword());

            ps.setString(3,e.getEmail());

            ps.setString(4,e.getCountry());

            status=ps.executeUpdate();

            con.close();

        }**catch**(Exception ex){ex.printStackTrace();}

**return** status;

    }

**public** **static** **int** update(Emp e){

**int** status=0;

**try**{

            Connection con=EmpDao.getConnection();

            PreparedStatement ps=con.prepareStatement(

                         "update user905 set name=?,password=?,email=?,country=? where id=?);

            ps.setString(1,e.getName());

       ps.setString(2,e.getPassword());

            ps.setString(3,e.getEmail());

            ps.setString(4,e.getCountry());

            ps.setInt(5,e.getId());

            status=ps.executeUpdate();

            con.close();

        }**catch**(Exception ex){ex.printStackTrace();}

**return** status;

    }

**public** **static** **int** delete(**int** id){

**int** status=0;

**try**{

            Connection con=EmpDao.getConnection();

  PreparedStatement ps=con.prepareStatement("delete from user905 where id=?");

            ps.setInt(1,id);

            status=ps.executeUpdate();

            con.close();

        }**catch**(Exception e){e.printStackTrace();}

**return** status;

    }

**public** **static** Emp getEmployeeById(**int** id){

        Emp e=**new** Emp();

**try**{

            Connection con=EmpDao.getConnection();

            PreparedStatement ps=con.prepareStatement("select \* from user905 where id=?";

           ps.setInt(1,id);

            ResultSet rs=ps.executeQuery();

**if**(rs.next()){

                e.setId(rs.getInt(1));

                e.setName(rs.getString(2));

                e.setPassword(rs.getString(3));

                e.setEmail(rs.getString(4));

                e.setCountry(rs.getString(5));

            }

            con.close();

        }**catch**(Exception ex){ex.printStackTrace();}

**return** e;

    }

**public** **static** List<Emp> getAllEmployees(){

        List<Emp> list=**new** ArrayList<Emp>();

**try**{

            Connection con=EmpDao.getConnection();

            PreparedStatement ps=con.prepareStatement("select \* from user905");

            ResultSet rs=ps.executeQuery();

**while**(rs.next()){

                Emp e=**new** Emp();

                e.setId(rs.getInt(1));

                e.setName(rs.getString(2));

                e.setPassword(rs.getString(3));

                e.setEmail(rs.getString(4));

                e.setCountry(rs.getString(5));

                list.add(e);

            }

            con.close();

        }**catch**(Exception e){e.printStackTrace();}

**return** list;

    }

}

SaveServlet.java

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

@WebServlet("/SaveServlet")

**public** **class** SaveServlet **extends** HttpServlet {

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out=response.getWriter();

        String name=request.getParameter("name");

        String password=request.getParameter("password");

        String email=request.getParameter("email");

        String country=request.getParameter("country");

        Emp e=**new** Emp();

        e.setName(name);

        e.setPassword(password);

        e.setEmail(email);

        e.setCountry(country);

**int** status=EmpDao.save(e);

**if**(status>0){

            out.print("<p>Record saved successfully!</p>");

            request.getRequestDispatcher("index.html").include(request, response);

        }**else**{

            out.println("Sorry! unable to save record");

        }

        out.close();

    }

}

EditServlet.java

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

@WebServlet("/EditServlet")

**public** **class** EditServlet **extends** HttpServlet {

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out=response.getWriter();

        out.println("<h1>Update Employee</h1>");

        String sid=request.getParameter("id");

**int** id=Integer.parseInt(sid);

        Emp e=EmpDao.getEmployeeById(id);

        out.print("<form action='EditServlet2' method='post'>");

        out.print("<table>");

        out.print("<tr><td></td><td><input type='hidden' name='id' value='"+e.getId()+"/></td></tr>");

        out.print("<tr><td>Name:</td><td><input type='text' name='name' value='"+e.getName()+"'/></td></tr>");

        out.print("<tr><td>Password:</td><td><input type='password' name='password' value='"+e.getPassword()+"'/>

                </td></tr>");

        out.print("<tr><td>Email:</td><td><input type='email' name='email' value='"+e.getEmail()+"'/></td></tr>");

        out.print("<tr><td>Country:</td><td>");

        out.print("<select name='country' style='width:150px'>");

        out.print("<option>India</option>");

        out.print("<option>USA</option>");

        out.print("<option>UK</option>");

        out.print("<option>Other</option>");

        out.print("</select>");

        out.print("</td></tr>");

        out.print("<tr><td colspan='2'><input type='submit' value='Edit & Save '/></td></tr>");

        out.print("</table>");

        out.print("</form>");

        out.close();

    }

}

EditServlet2.java

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

@WebServlet("/EditServlet2")

**public** **class** EditServlet2 **extends** HttpServlet {

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out=response.getWriter();

        String sid=request.getParameter("id");

**int** id=Integer.parseInt(sid);

        String name=request.getParameter("name");

        String password=request.getParameter("password");

        String email=request.getParameter("email");

        String country=request.getParameter("country");

        Emp e=**new** Emp();

        e.setId(id);

        e.setName(name);

        e.setPassword(password);

        e.setEmail(email);

        e.setCountry(country);

**int** status=EmpDao.update(e);

**if**(status>0){

            response.sendRedirect("ViewServlet");

        }**else**{

            out.println("Sorry! unable to update record");

        }

        out.close();

    }

}

  DeleteServlet.java

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

@WebServlet("/DeleteServlet")

**public** **class** DeleteServlet **extends** HttpServlet {

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {          String sid=request.getParameter("id");

**int** id=Integer.parseInt(sid);

        EmpDao.delete(id);

        response.sendRedirect("ViewServlet");

    }

}

ViewServlet.java

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.util.List;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

@WebServlet("/ViewServlet")

**public** **class** ViewServlet **extends** HttpServlet {

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out=response.getWriter();

        out.println("<a href='index.html'>Add New Employee</a>");

        out.println("<h1>Employees List</h1>");

        List<Emp> list=EmpDao.getAllEmployees();

        out.print("<table border='1' width='100%'");

        out.print("<tr><th>Id</th><th>Name</th><th>Password</th><th>Email</th><

h>Country</th>

                 <th>Edit</th><th>Delete</th></tr>");

**for**(Emp e:list){

         out.print("<tr><td>"+e.getId()+"</td><td>"+e.getName()+"</td><td>"+e.getPassword()+"</td>

                 <td>"+e.getEmail()+"</td><td>"+e.getCountry()+"</td><td><a href='EditServlet?id="+e.getId()+"'>edit</a></td>

                 <td><a href='DeleteServlet?id="+e.getId()+"'>delete</a></td></tr>");

        }

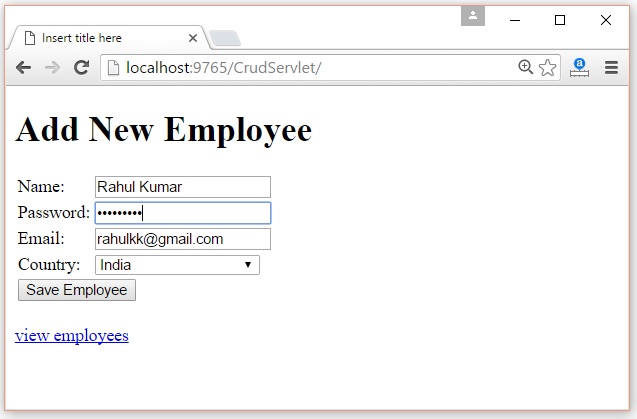
        out.print("</table>");

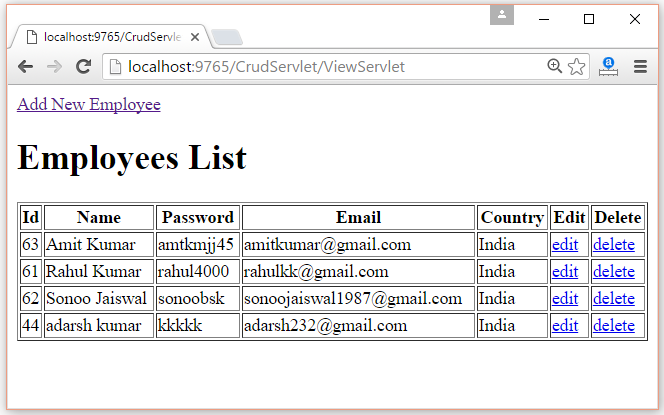
        out.close();

    }

}

Output:





**PRACTICAL NO 7:**

**Create Employees table in EMP database. Perform select, insert, update, delete operations on Employee table using JSP.**

**Index.jsp**

<!DOCTYPE html>

<html>

<head>

<meta http-euiv="Content-Type" content="text/html";charset=ISO-8859-1">

<title>JSP CRUD Example</title>

</head>

<body>

<h1>JSP CRUD Example</h1>

<a href="adduserform.jsp">Add User</a>

<a href="viewuser.jsp">View Users</a>

</body>

</html>

**Adduserform.jsp**

<!DOCTYPE html>

<html>

<head>

<meta http-euiv="Content-Type" content="text/html";charset=ISO-8859-1">

<title>JSP CRUD Example</title>

</head>

<body>

<jsp:include page="userform.html"></jsp:include>

</body>

</html>

Userform.html

0<a href="viewusers.jsp">View All Records</a><br/>

<h1>Add New User</h1>

<form action="adduser.jsp" method="post">

<table>

<tr><td>Name:</td><td><input type="text" name="name"/></td></tr>

<tr><td>Password:</td><td><input type="password" name="password"/></td></tr>

<tr><td>Email:</td><td><input type='"email" name="email"/></td></tr>

<tr><td>Sex:</td><td>

<input type="radio" name="sex" value"male"/>Male

<input type="radio" name="sex" value"female"/>Female</td></tr>

<tr><td>Country:</td><td>

<select name='country' style='width:150px'>

<option>India</option>

<option>Pakistan</option>

<option>Afghanistan</option>

<option>Berma</option>

<option>Other</option>

</select>

</td></tr>

<tr><td colspan='2'><input type="submit" value="Add User"/></td></tr>

</table>

</form>

**Adduser.jsp**

<%@page import="com.javapoint.dao.UserDao"%>

<jsp:useBean id="u" class="com.javapoint.bean.User"></jsp:useBean>

<jsp:setProperty property="\*" name="u"/>

<%

int i=UserDao.save(u);

if(i>0)

{

response.sendRedirect("adduser-success.jsp");

}

else

{

response.sendRedirect("adduser-error.jsp");

}

%>

**User.java**

package com.javapoint.bean;

public class User

{

private int id;

private String name,password,emai,sex,country;

}

**UserDao.java**

package com.javapoint.dao;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

import com.javapoint.bean.User;

public class UserDao

{

public static Connection getConnection()

{

Connection con=null;

try

{

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

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/test","","");

}

catch(Exception e)

{

System.out.println(e);

}

return con;

}

public static int save(User u)

{

int status=0;

try

{

Connection con = getConnection();

PreparedStatement ps = con.prepareStatement("insert into register(name,password,email,sex,country) values(?,?,?,?,?)");

ps.setString(1,u.getName());

ps.setString(2,u.getPassword());

ps.setString(3,u.getEmail());

ps.setString(4,u.getSex());

ps.setString(5,u.getCountry());

status=ps.executeUpdate();

}

catch(Exception e)

{

System.out.println(e);

}

return status;

}

public static int update(User u)

{

int status=0;

try

{

Connection con = getConnection();

PreparedStatement ps = con.prepareStatement("update register set name=?,password=?,email=?,sex=?,country=? where id =?");

ps.setString(1,u.getName());

ps.setString(2,u.getPassword());

ps.setString(3,u.getEmail());

ps.setString(4,u.getSex());

ps.setString(5,u.getCountry());

ps.setInt(6,u.getId());

status=ps.executeUpdate();

}

catch(Exception e)

{

System.out.println(e);

}

return status;

}

public static int delete(User u)

{

int status=0;

try

{

Connection con = getConnection();

PreparedStatement ps = con.prepareStatement("insert into register(name,password,email,sex,country) values(?,?,?,?,?)");

ps.setInt(6,u.getId());

status=ps.executeUpdate();

}

catch(Exception e)

{

System.out.println(e);

}

return status;

}

public static List<User> getAllRecords()

{

List<User> list = new ArayList<User>();

try

{

Connection con = getConnection();

PreparedStatement ps = con.prepareStatement("select \* from register");

ResultSet rs = ps.executeQuery();

while(rs.next())

{

User u = new User();

u.setId(rs.getInt("id"));

u.setName(rs.getString("name"));

u.setPassword(rs.getString("password"));

u.setEmail(rs.getString("email"));

u.setSex(rs.getString("sex"));

u.setCountry(rs.getString("country"));

list.add(u);

}

catch(Exception e)

{

System.out.println(e);

}

return list;

}

public static User getRecordById(int id)

{

User u = null;

try

{

Connection con = getConnection();

PreparedStatement ps = con.prepareStatement("select \* from register where id=?");

ps.setInt(1,id);

ResultSet rs = ps.executeQuery();

while(rs.next())

{

u = new User();

u.setId(rs.getInt("id"));

u.setName(rs.getString("name"));

u.setPassword(rs.getString("password"));

u.setEmail(rs.getString("email"));

u.setSex(rs.getString("sex"));

u.setCountry(rs.getString("country"));

}

catch(Exception e)

{

System.out.println(e);

}

return u;

}

}

**Adduser-success.jsp**

<!DOCTYPE html>

<html>

<head>

<meta http-euiv="Content-Type" content="text/html";charset=ISO-8859-1">

<title>Add User Success</title>

</head>

<body>

<p> Record successfully saved!</p>

<jsp:include page="userform.html"></jsp:include>

</body>

</html>

**Adduser-error.jsp**

<!DOCTYPE html>

<html>

<head>

<meta http-euiv="Content-Type" content="text/html";charset=ISO-8859-1">

<title>Add User Error</title>

</head>

<body>

<p>Sorry, an error occurred!</p>

<jsp:include page="userform.html"></jsp:include>

</body>

</html>

**Viewusers.jsp**

<!DOCTYPE html>

<html>

<head>

<meta http-euiv="Content-Type" content="text/html";charset=ISO-8859-1">

<title>View Users</title>

</head>

<body>

<%@page import="com.javapoint.dao.Userdao,com.javapoint.bean.\*.java.util.\*"%>

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

<h1>Users List</h1>

<%

List<User> list=UserDao.getAllRecords();

request.setAttribute("list",list);

%>

<table border="1" width="90%">

<tr><th>Id</th><th>Name</th><th>Password</th><th>Email</th><th>Sex</th><th>Country</th><th>Edit</th><th>Delete</th></tr>

<c:forEach items="${list}" var="u">

<tr><td>${u.getId()}</td><td>${u.getName()}</td><td>${u.getPassword()}</td><td>${u.getEmail()}</td><td>${u.getSex()}</td><td>${u.getCountry()}</td>

<td><a href="editform.jsp?id=$u.getId()}">Edit</a></td>

<td><a href="deleteuser.jsp?id=$u.getId()}">Delete</a></td></tr>

</c:forEach>

</table>

<br/><a href="adduserform.jsp">Add New User</a>

</body>

</html>

**Editform.jsp**

<!DOCTYPE html>

<html>

<head>

<meta http-euiv="Content-Type" content="text/html";charset=ISO-8859-1">

<title>Edit Form</title>

</head>

<body>

<%@page import="com.javapoint.dao.Userdao,com.javapoint.bean.\*.java.util.\*"%>

<%

String id=request.getParamater("id");

User u=UserDao.getRecordById(Intege.parseInt(id));

%>

<h1>Edit Form</h1>

<form action="edituser.jsp" method="post">

<input type="hidden" name="id" value="<%=u.getId()%>"/>

<table>

<tr><td>Name:</td><td><input type="text" name="name" value="<%=u.getName()%>/></td></tr>

<tr><td>Password:</td><td><input type="password" name="password"value="<%=u.getPassword()%>/></td></tr>

<tr><td>Email:</td><td><input type='"email" name="email" value="<%=u.getEmail()%>/></td></tr>

<tr><td>Sex:</td><td>

<input type="radio" name="sex" value"male"/>Male

<input type="radio" name="sex" value"female"/>Female</td></tr>

<tr><td>Country:</td><td>

<select name='country' style='width:150px'>

<option>India</option>

<option>Pakistan</option>

<option>Afghanistan</option>

<option>Berma</option>

<option>Other</option>

</select>

</td></tr>

<tr><td colspan='2'><input type="submit" value="Add User"/></td></tr>

</table>

</form>

</body>

</html>

**Edituser.jsp**

<%@page import="com.javapoint.dao.UserDao"%>

<jsp:useBean id="u" class="com.javapoint.bean.User"></jsp:useBean>

<jsp:setProperty property="\*" name="u"/>

<%

int i=UserDao.save(u);

response.sendRedirect("viewusrs.jsp");

%>

**Deleteuser.jsp**

<%@page import="com.javapoint.dao.UserDao"%>

<jsp:useBean id="u" class="com.javapoint.bean.User"></jsp:useBean>

<jsp:setProperty property="\*" name="u"/>

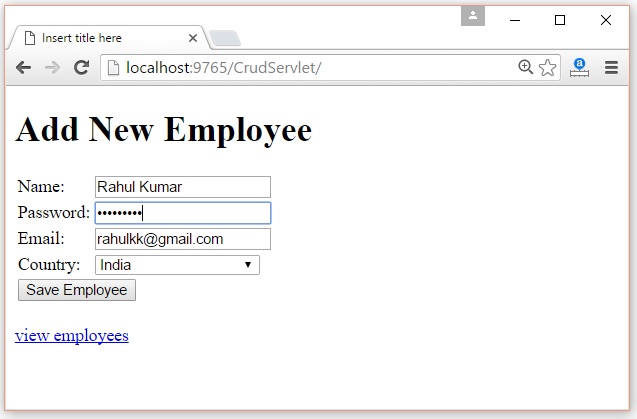
<%

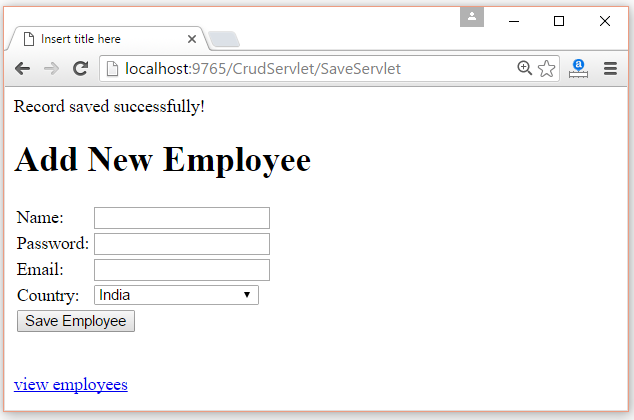
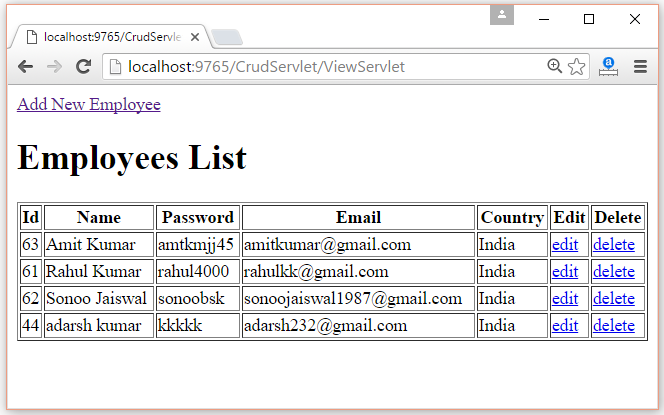
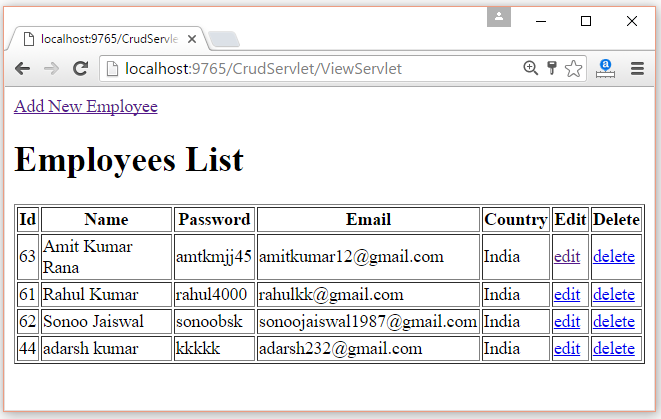
UserDao.delete(u);

response.sendRedirect("viewusers.jsp");

%>

Output





**PRACTICAL NO 8:**

**Write a Student class with three properties. The useBean action declares a JavaBean for use in a JSP. Write Java Application to access JavaBeans Properties.**

**Student.java**

import java .io.Serializable;

public class Student

{

private String sname;

private int rno, marks;

public Student()

{

sname="";

}

public void setSname(String name)

{

sname = name;

}

public String getSname()

{

return sname;

}

public void setRno(int r)

{

rno = r;

}

public intgetRno()

{

return rno;

}

public void setMarks(int m)

{

marks = m;

}

public intgetMarks()

{

return marks;

}

}

**Hello.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title> WELCOME TO STUDENT INFORMATION </title>

</head>

<jsp:useBean id="stud" class="Student" scope="reuest"/>

<jsp:setProperty name="stud" property="sname" value="Shreeji"/>

<jsp:setProperty name="stud" property="rno" value="52"/>

<jsp:setProperty name="stud" property="marks" value="95"/>

<body>

<h1>Name of Student is:<jsp:setProperty name="stud" property="sname"/></h1>

<h1>Roll No of Student is:<jsp:setProperty name="stud" property="rno"/></h1>

<h1>Marks of Student is:<jsp:setProperty name="stud" property="marks"/></h1>

</body>

</html>

Output

Let us make the **StudentsBean.class** available in CLASSPATH. Access the above JSP. the following result will be displayed −

Student First Name: Zara

Student Last Name: Ali

Student Age: 10

**PRACTICAL NO 9:**

**Design application using Struts2. Application must accept user name and greet user when command button is pressed.**

**Steps to create Struts Application –**

**Download and extract Struts2 binary distribution files. It is freely available online. Here it is recommend downloading the “Full Distribution” zip file containing all dependency jar files.**

1. **Create JSPs**
2. **Create action pages according to JSPs**
3. **Connect the JSPs ad action pages using configuration files (such as web.xml and struts.xml).**

**Web Application1**

* **Web pages**
* **Meta-INF Context.xml**
* **Web-INF Web.xml**
* **Source packages**
* **Default packages**
* **Libraries**
* **All struts libraries**
* **Configuration files**
* **Manifest.MF**
* **Context.xml**
* **Web-fragment.xml**
* **Web.xml**

1. **And create “index.jsp” by selecting new Java server page document.**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%taglib prefix = "s" uri="/struts-tags"%>

<html>

<body>

<s:form method ="post" action ="hello\_action">

<s:labelvalue ="Enter your name in text box and click the button" name="label1"/>

<s:textfield name ="name"/>

<s:submit value ="CLICK" name="submit"/>

</s:form>

</body>

</html>

1. **This will generate certain output.**
2. **Create another jsp page named as “result.jsp” inside jsp folder.**

//result.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%taglib prefix = "s" uri="/struts-tags"%>

<html>

<body>

<h1>Hello<s:property value="name"/></h1>

</body>

</html>

1. **The folder here is known as package. Create a folder named action\_jsp inside source packages.**
2. **Create action class “hello\_action” inside “action\_jsp” package.**
3. **So the hello\_action java file will be as follwos:**

import com.opensymphony.xwork2.ActionSupport;

//everything in here must be kept public to have package level access.

public class hello\_ation extends ActionSupport

{

private String name;

//getter and setter methods for the form element value

public String getName()

{

return name;

}

public void setName(String name)

{

this.name = name;

}

//method that returns success or failure on action

public String execute()

{

return "success";

}

}

1. **Struts.xml can be written as:**

<!DOCTYPE struts PUBLIC

"-//Apache Software Foundation//DTD Sturts Configuration 2.0//En"

"http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>

<package name ="default" extends ="struts-default">

<action name ="hello\_action" class ="action\_jsp.hello\_action" method ="execute">

<result name ="success">result.jsp</result>

<result name ="failure">index.jsp</result>

</action>

</package>

</struts>

1. **In web.xml, change the welcome file to our index.jsp**

<?xml version ="1.0" encoding ="UTF-8"?>

<web-app version ="3.1" xmlns = "http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation ="http://xmlns.jcp.org./xml/ns//javaee http://xmlns.jcp.org./xml/ns/javaee/web-app\_3\_1.xsd">

<filter>

<filter-name>struts2</filter-name>

<filter-class>org.apace.struts2.dispatcher.FilterDispatcer</filter-class>

</filter>

<filter-mapping>

<filter-name>struts<//filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<session-config>

<session-timeout>

30

</sesson-timeout>

</session-config>

<welcome-file-list>

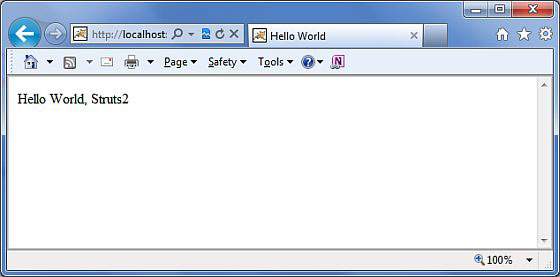
<welcome-file>jsp/index.jsp</welcome-file>

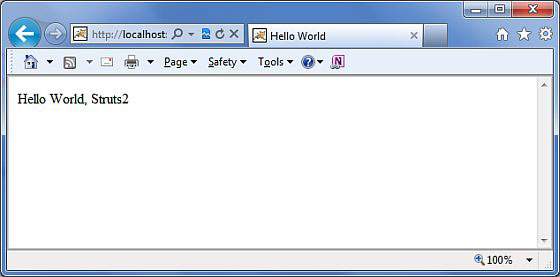
</welcome-file-list>

</web-app>

1. **Run the application. And see the welcome message.**

**Output**





**PRACTICAL NO 10:**

**Write Java Application to encoding and decoding JSON in Java.**

**Follow the steps:**

1. **Download the jar from here-**

[**https://code.google.com/archive/p/json-simple/downloads**](https://code.google.com/archive/p/json-simple/downloads)**.**

1. **Gotonetbeans- in your “Projects” window go for- “libraries” right click on it and “json-simple-1.1.1” folder to it.**
2. **Write encoding code:**

import java.util.HashMap;

import java.util.Map;

import org.json.simple.JSONValue;

public class JSON\_USING\_MAP

{

public static void main(String args[])

{

Map obj = new HashMap()

System.out.println("Encoding Using MAP....\n");

obj.put("COLLEGE NAME","ABC COLLEGE");

obj.put("YEAR OF ESTD",new Integer(1927));

obj.put("NO OF COURSES",new Double(60));

String txt = JSONValue.to.JSONString(obj);

System.out.print(txt);

}

}

1. **This will generate certain output.**

Encoding….

{“NO OF COURSES”:60.0,”YEAR OF ESTD”:1927,”COLLEGE NAME”:”XYZ COLLEGE”}

1. **We can also obtain same results using MAP.**

import org.json.simple.JSONObject;

public class JSON\_ENCODE\_DECODE

{

public static void main(String[] args)

{

JSONObject obj = new JSONObject();

System.out.println("Encoding.....\n");

obj.put("COLLEGE NAME","ABC COLLEGE");

obj.put("YEAR OF ESTD",new Integer(1927));

obj.put("NO OF COURSES",new Double(60));

System.out.print(obj);

}

}

1. **This will also generate certain output.**

Encoding Using MAP…..

{“NO OF COURSES”:60.0,”YEAR OF ESTD”:1927,”COLLEGE NAME”:”XYZ COLLEGE”}

1. **Let’s see how to decode…**

import org.json.simple.JSONObject;

import org.json.simple.JSONValue;

public class JSON\_DECODE

{

public static void main(String[] args)

{

String s="{\"COLLEGE NAME\":\"XYZ COLLEGE\",\"YEAR OF ESTD\":1987.0,\"NO OF COURSES\":56}";

Object obj = JSONValue.parse(s);

JSONObject Obj = (JSONObject) obj;

String col\_name = (String)jsObj.get("COLLEGE NAME");

double year = (Double)jsObj.get("YEAR OF ESTD");

long courses = (Long)jsObj.get("NO OF COURSES");

System.out.println(col\_name+"\n"+year+"\n"+courses);

}

}

Output

1. **This will generate following output:**

XYZ COLLEGE  
1987.0

56