-- Database-Level

DROP DATABASE *databaseName* -- Delete the database (irrecoverable!)

DROP DATABASE IF EXISTS *databaseName* -- Delete if it exists

CREATE DATABASE *databaseName* -- Create a new database

CREATE DATABASE IF NOT EXISTS *databaseName* -- Create only if it does not exists

SHOW DATABASES -- Show all the databases in this server

USE *databaseName* -- Set the default (current) database

SELECT DATABASE() -- Show the default database

SHOW CREATE DATABASE *databaseName* -- Show the CREATE DATABASE statement

-- Table-Level

DROP TABLE [IF EXISTS] *tableName*, ...

CREATE TABLE [IF NOT EXISTS] *tableName* (

*columnName columnType columnAttribute*, ...

PRIMARY KEY(*columnName*),

FOREIGN KEY (*columnNmae*) REFERENCES *tableName* (*columnNmae*)

)

SHOW TABLES -- Show all the tables in the default database

DESCRIBE|DESC *tableName* -- Describe the details for a table

ALTER TABLE *tableName* ... -- Modify a table, e.g., ADD COLUMN and DROP COLUMN

ALTER TABLE *tableName* ADD *columnDefinition*

ALTER TABLE *tableName* DROP *columnName*

ALTER TABLE *tableName* ADD FOREIGN KEY (*columnNmae*) REFERENCES *tableName* (*columnNmae*)

ALTER TABLE *tableName* DROP FOREIGN KEY *constraintName*

SHOW CREATE TABLE *tableName* -- Show the CREATE TABLE statement for this *tableName*

-- Row-Level

INSERT INTO *tableName*

VALUES (*column1Value*, *column2Value*,...) -- Insert on all Columns

INSERT INTO *tableName*

VALUES (*column1Value*, *column2Value*,...), ... -- Insert multiple rows

INSERT INTO *tableName* (*column1Name*, ..., *columnNName*)

VALUES (*column1Value*, ..., *columnNValue*) -- Insert on selected Columns

DELETE FROM *tableName* WHERE *criteria*

UPDATE *tableName* SET *columnName* = *expr*, ... WHERE *criteria*

SELECT \* | *column1Name* AS *alias1*, ..., *columnNName* AS *aliasN*

FROM *tableName*

WHERE *criteria*

GROUP BY *columnName*

ORDER BY *columnName* ASC|DESC, ...

HAVING *groupConstraints*

LIMIT *count* | *offset count*

-- Others

SHOW WARNINGS; -- Show the warnings of the previous statement

-- Insert a row with all the column values

mysql> **INSERT INTO products VALUES (1001, 'PEN', 'Pen Red', 5000, 1.23);**

Query OK, 1 row affected (0.04 sec)

-- Insert multiple rows in one command

-- Inserting NULL to the auto\_increment column results in max\_value + 1

mysql> **INSERT INTO products VALUES**

**(NULL, 'PEN', 'Pen Blue', 8000, 1.25),**

**(NULL, 'PEN', 'Pen Black', 2000, 1.25);**

Query OK, 2 rows affected (0.03 sec)

Records: 2 Duplicates: 0 Warnings: 0

-- Insert value to selected columns

-- Missing value for the auto\_increment column also results in max\_value + 1

mysql> **INSERT INTO products (productCode, name, quantity, price) VALUES**

**('PEC', 'Pencil 2B', 10000, 0.48),**

**('PEC', 'Pencil 2H', 8000, 0.49);**

Query OK, 2 row affected (0.03 sec)

-- Missing columns get their default values

mysql> **INSERT INTO products (productCode, name) VALUES ('PEC', 'Pencil HB');**

Query OK, 1 row affected (0.04 sec)

-- 2nd column (productCode) is defined to be NOT NULL

mysql> **INSERT INTO products values (NULL, NULL, NULL, NULL, NULL);**

ERROR 1048 (23000): Column 'productCode' cannot be null

-- Query the table

mysql> **SELECT \* FROM products;**

+-----------+-------------+-----------+----------+------------+

| productID | productCode | name | quantity | price |

+-----------+-------------+-----------+----------+------------+

| 1001 | PEN | Pen Red | 5000 | 1.23 |

| 1002 | PEN | Pen Blue | 8000 | 1.25 |

| 1003 | PEN | Pen Black | 2000 | 1.25 |

| 1004 | PEC | Pencil 2B | 10000 | 0.48 |

| 1005 | PEC | Pencil 2H | 8000 | 0.49 |

| 1006 | PEC | Pencil HB | 0 | 9999999.99 |

+-----------+-------------+-----------+----------+------------+

6 rows in set (0.02 sec)

-- Remove the last row

mysql> **DELETE FROM products WHERE productID = 1006;**

**INSERT INTO Syntax**

1

// Java program to implement

// a Simple Registration Form

// using Java Swing

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class MyFrame

extends JFrame

implements ActionListener {

// Components of the Form

private Container c;

private JLabel title;

private JLabel name;

private JTextField tname;

private JLabel mno;

private JTextField tmno;

private JLabel gender;

private JRadioButton male;

private JRadioButton female;

private ButtonGroup gengp;

private JLabel dob;

private JComboBox date;

private JComboBox month;

private JComboBox year;

private JLabel add;

private JTextArea tadd;

private JCheckBox term;

private JButton sub;

private JButton reset;

private JTextArea tout;

private JLabel res;

private JTextArea resadd;

private String dates[]

= { "1", "2", "3", "4", "5",

"6", "7", "8", "9", "10",

"11", "12", "13", "14", "15",

"16", "17", "18", "19", "20",

"21", "22", "23", "24", "25",

"26", "27", "28", "29", "30",

"31" };

private String months[]

= { "Jan", "feb", "Mar", "Apr",

"May", "Jun", "July", "Aug",

"Sup", "Oct", "Nov", "Dec" };

private String years[]

= { "1995", "1996", "1997", "1998",

"1999", "2000", "2001", "2002",

"2003", "2004", "2005", "2006",

"2007", "2008", "2009", "2010",

"2011", "2012", "2013", "2014",

"2015", "2016", "2017", "2018",

"2019" };

// constructor, to initialize the components

// with default values.

public MyFrame()

{

setTitle("Registration Form");

setBounds(300, 90, 900, 600);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setResizable(false);

c = getContentPane();

c.setLayout(null);

title = new JLabel("Registration Form");

title.setFont(new Font("Arial", Font.PLAIN, 30));

title.setSize(300, 30);

title.setLocation(300, 30);

c.add(title);

name = new JLabel("Name");

name.setFont(new Font("Arial", Font.PLAIN, 20));

name.setSize(100, 20);

name.setLocation(100, 100);

c.add(name);

tname = new JTextField();

tname.setFont(new Font("Arial", Font.PLAIN, 15));

tname.setSize(190, 20);

tname.setLocation(200, 100);

c.add(tname);

mno = new JLabel("Mobile");

mno.setFont(new Font("Arial", Font.PLAIN, 20));

mno.setSize(100, 20);

mno.setLocation(100, 150);

c.add(mno);

tmno = new JTextField();

tmno.setFont(new Font("Arial", Font.PLAIN, 15));

tmno.setSize(150, 20);

tmno.setLocation(200, 150);

c.add(tmno);

gender = new JLabel("Gender");

gender.setFont(new Font("Arial", Font.PLAIN, 20));

gender.setSize(100, 20);

gender.setLocation(100, 200);

c.add(gender);

male = new JRadioButton("Male");

male.setFont(new Font("Arial", Font.PLAIN, 15));

male.setSelected(true);

male.setSize(75, 20);

male.setLocation(200, 200);

c.add(male);

female = new JRadioButton("Female");

female.setFont(new Font("Arial", Font.PLAIN, 15));

female.setSelected(false);

female.setSize(80, 20);

female.setLocation(275, 200);

c.add(female);

gengp = new ButtonGroup();

gengp.add(male);

gengp.add(female);

dob = new JLabel("DOB");

dob.setFont(new Font("Arial", Font.PLAIN, 20));

dob.setSize(100, 20);

dob.setLocation(100, 250);

c.add(dob);

date = new JComboBox(dates);

date.setFont(new Font("Arial", Font.PLAIN, 15));

date.setSize(50, 20);

date.setLocation(200, 250);

c.add(date);

month = new JComboBox(months);

month.setFont(new Font("Arial", Font.PLAIN, 15));

month.setSize(60, 20);

month.setLocation(250, 250);

c.add(month);

year = new JComboBox(years);

year.setFont(new Font("Arial", Font.PLAIN, 15));

year.setSize(60, 20);

year.setLocation(320, 250);

c.add(year);

add = new JLabel("Address");

add.setFont(new Font("Arial", Font.PLAIN, 20));

add.setSize(100, 20);

add.setLocation(100, 300);

c.add(add);

tadd = new JTextArea();

tadd.setFont(new Font("Arial", Font.PLAIN, 15));

tadd.setSize(200, 75);

tadd.setLocation(200, 300);

tadd.setLineWrap(true);

c.add(tadd);

term = new JCheckBox("Accept Terms And Conditions.");

term.setFont(new Font("Arial", Font.PLAIN, 15));

term.setSize(250, 20);

term.setLocation(150, 400);

c.add(term);

sub = new JButton("Submit");

sub.setFont(new Font("Arial", Font.PLAIN, 15));

sub.setSize(100, 20);

sub.setLocation(150, 450);

sub.addActionListener(this);

c.add(sub);

reset = new JButton("Reset");

reset.setFont(new Font("Arial", Font.PLAIN, 15));

reset.setSize(100, 20);

reset.setLocation(270, 450);

reset.addActionListener(this);

c.add(reset);

tout = new JTextArea();

tout.setFont(new Font("Arial", Font.PLAIN, 15));

tout.setSize(300, 400);

tout.setLocation(500, 100);

tout.setLineWrap(true);

tout.setEditable(false);

c.add(tout);

res = new JLabel("");

res.setFont(new Font("Arial", Font.PLAIN, 20));

res.setSize(500, 25);

res.setLocation(100, 500);

c.add(res);

resadd = new JTextArea();

resadd.setFont(new Font("Arial", Font.PLAIN, 15));

resadd.setSize(200, 75);

resadd.setLocation(580, 175);

resadd.setLineWrap(true);

c.add(resadd);

setVisible(true);

}

// method actionPerformed()

// to get the action performed

// by the user and act accordingly

public void actionPerformed(ActionEvent e)

{

if (e.getSource() == sub) {

if (term.isSelected()) {

String data1;

String data

= "Name : "

+ tname.getText() + "\n"

+ "Mobile : "

+ tmno.getText() + "\n";

if (male.isSelected())

data1 = "Gender : Male"

+ "\n";

else

data1 = "Gender : Female"

+ "\n";

String data2

= "DOB : "

+ (String)date.getSelectedItem()

+ "/" + (String)month.getSelectedItem()

+ "/" + (String)year.getSelectedItem()

+ "\n";

String data3 = "Address : " + tadd.getText();

tout.setText(data + data1 + data2 + data3);

tout.setEditable(false);

res.setText("Registration Successfully..");

}

else {

tout.setText("");

resadd.setText("");

res.setText("Please accept the"

+ " terms & conditions..");

}

}

else if (e.getSource() == reset) {

String def = "";

tname.setText(def);

tadd.setText(def);

tmno.setText(def);

res.setText(def);

tout.setText(def);

term.setSelected(false);

date.setSelectedIndex(0);

month.setSelectedIndex(0);

year.setSelectedIndex(0);

resadd.setText(def);

}

}

}

// Driver Code

class Registration {

public static void main(String[] args) throws Exception

{

MyFrame f = new MyFrame();

}

}

2

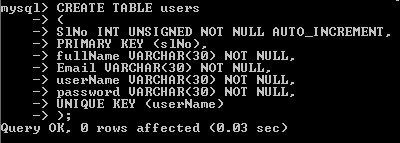
|  |
| --- |
| package com.mvc.controller;  import java.io.IOException;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;    import com.mvc.bean.RegisterBean;  import com.mvc.dao.RegisterDao;    public class RegisterServlet extends HttpServlet {         public RegisterServlet() {       }         protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {          //Copying all the input parameters in to local variables           String fullName = request.getParameter("fullname");           String email = request.getParameter("email");           String userName = request.getParameter("username");           String password = request.getParameter("password");             RegisterBean registerBean = new RegisterBean();          //Using Java Beans - An easiest way to play with group of related data           registerBean.setFullName(fullName);           registerBean.setEmail(email);           registerBean.setUserName(userName);           registerBean.setPassword(password);             RegisterDao registerDao = new RegisterDao();            //The core Logic of the Registration application is present here. We are going to insert user data in to the database.           String userRegistered = registerDao.registerUser(registerBean);             if(userRegistered.equals("SUCCESS"))   //On success, you can display a message to user on Home page           {              request.getRequestDispatcher("/Home.jsp").forward(request, response);           }           else   //On Failure, display a meaningful message to the User.           {              request.setAttribute("errMessage", userRegistered);              request.getRequestDispatcher("/Register.jsp").forward(request, response);           }       }  } |

***To summarize RegisterServlet.java flow :*** *1. Assign all the inputs(user details) to local variables.  
2. Call RegisterBean.java to set all the user details using java setters.  
3. Next, go to RegisterDao.java where you are just going to insert user details into the database.  
4. Once it is successful, you are displaying a successful message.*

**RegisterDao.java**

**Data Access Object** – It focuses on business logic with database connections and operations.  
Here the RegisterDao.java code makes a connection with the Database layer and inserts user details into the database.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | package com.mvc.dao;    import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.SQLException;  import com.mvc.bean.RegisterBean;  import com.mvc.util.DBConnection;    public class RegisterDao {       public String registerUser(RegisterBean registerBean)       {           String fullName = registerBean.getFullName();           String email = registerBean.getEmail();           String userName = registerBean.getUserName();           String password = registerBean.getPassword();             Connection con = null;           PreparedStatement preparedStatement = null;           try           {               con = DBConnection.createConnection();               String query = "insert into users(SlNo,fullName,Email,userName,password) values (NULL,?,?,?,?)"; //Insert user details into the table 'USERS'               preparedStatement = con.prepareStatement(query); //Making use of prepared statements here to insert bunch of data               preparedStatement.setString(1, fullName);               preparedStatement.setString(2, email);               preparedStatement.setString(3, userName);               preparedStatement.setString(4, password);                 int i= preparedStatement.executeUpdate();                 if (i!=0)  //Just to ensure data has been inserted into the database               return "SUCCESS";           }           catch(SQLException e)           {              e.printStackTrace();           }           return "Oops.. Something went wrong there..!";  // On failure, send a message from here.       }  } |

MySQL script to create users table

The structure of users table

**RegisterBean.java**  
JavaBeans are classes that encapsulate many objects into a single object (the bean). A JavaBean property is a named feature that can be accessed by the user of the object. Here the RegisterBean encapsulates registration properties fullName, email, userName, password. To set or access individual properties, set and get functions have been implemented.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | package com.mvc.bean;    public class RegisterBean {    private String fullName;  private String email;  private String userName;  private String password;    public String getUserName() {  return userName;  }  public void setUserName(String userName) {  this.userName = userName;  }  public String getPassword() {  return password;  }  public void setPassword(String password) {  this.password = password;  }  public void setFullName(String fullName) {  this.fullName = fullName;  }  public String getFullName() {  return fullName;  }  public void setEmail(String email) {  this.email = email;  }  public String getEmail() {  return email;  }  } |

**DBConnection.java**

We are making use of MySQL database server in this application. The URL format and driver name going to be different for different database servers.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | package com.mvc.util;    import java.sql.Connection;  import java.sql.DriverManager;    public class DBConnection {  public static Connection createConnection()  {       Connection con = null;       String url = "jdbc:mysql://localhost:3306/customers"; //MySQL URL followed by the database name       String username = "root"; //MySQL username       String password = "root123"; //MySQL password       System.out.println("In DBConnection.java class ");         try       {           try           {              Class.forName("com.mysql.jdbc.Driver"); //loading MySQL drivers. This differs for database servers           }           catch (ClassNotFoundException e)           {              e.printStackTrace();           }           con = DriverManager.getConnection(url, username, password); //attempting to connect to MySQL database           System.out.println("Printing connection object "+con);       }       catch (Exception e)       {          e.printStackTrace();       }       return con;  }  } |

Java code with database connectivity

import javax.swing.\*;

import java.awt.event.\*;

import java.awt.\*;

import java.sql.\*;

public class RegistrationForm implements ActionListener {

JFrame frame;

String[] gender={"Male","Female"};

JLabel nameLabel=new JLabel("NAME");

JLabel genderLabel=new JLabel("GENDER");

JLabel fatherNameLabel=new JLabel("FATHER NAME");

JLabel passwordLabel=new JLabel("PASSWORD");

JLabel confirmPasswordLabel=new JLabel("CONFIRM PASSWORD");

JLabel cityLabel=new JLabel("CITY");

JLabel emailLabel=new JLabel("EMAIL");

JTextField nameTextField=new JTextField();

JComboBox genderComboBox=new JComboBox(gender);

JTextField fatherTextField=new JTextField();

JPasswordField passwordField=new JPasswordField();

JPasswordField confirmPasswordField=new JPasswordField();

JTextField cityTextField=new JTextField();

JTextField emailTextField=new JTextField();

JButton registerButton=new JButton("REGISTER");

JButton resetButton=new JButton("RESET");

RegistrationForm()

{

createWindow();

setLocationAndSize();

addComponentsToFrame();

actionEvent();

}

public void createWindow()

{

frame=new JFrame();

frame.setTitle("Registration Form");

frame.setBounds(40,40,380,600);

frame.getContentPane().setBackground(Color.pink);

frame.getContentPane().setLayout(null);

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setResizable(false);

}

public void setLocationAndSize()

{

nameLabel.setBounds(20,20,40,70);

genderLabel.setBounds(20,70,80,70);

fatherNameLabel.setBounds(20,120,100,70);

passwordLabel.setBounds(20,170,100,70);

confirmPasswordLabel.setBounds(20,220,140,70);

cityLabel.setBounds(20,270,100,70);

emailLabel.setBounds(20,320,100,70);

nameTextField.setBounds(180,43,165,23);

genderComboBox.setBounds(180,93,165,23);

fatherTextField.setBounds(180,143,165,23);

passwordField.setBounds(180,193,165,23);

confirmPasswordField.setBounds(180,243,165,23);

cityTextField.setBounds(180,293,165,23);

emailTextField.setBounds(180,343,165,23);

registerButton.setBounds(70,400,100,35);

resetButton.setBounds(220,400,100,35);

}

public void addComponentsToFrame()

{

frame.add(nameLabel);

frame.add(genderLabel);

frame.add(fatherNameLabel);

frame.add(passwordLabel);

frame.add(confirmPasswordLabel);

frame.add(cityLabel);

frame.add(emailLabel);

frame.add(nameTextField);

frame.add(genderComboBox);

frame.add(fatherTextField);

frame.add(passwordField);

frame.add(confirmPasswordField);

frame.add(cityTextField);

frame.add(emailTextField);

frame.add(registerButton);

frame.add(resetButton);

}

public void actionEvent()

{

registerButton.addActionListener(this);

resetButton.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource()==registerButton)

{

try {

Connection connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/myDatabase","root","root");

PreparedStatement Pstatement=connection.prepareStatement("insert into student values(?,?,?,?,?,?,?)");

Pstatement.setString(1,nameTextField.getText());

Pstatement.setString(2,genderComboBox.getSelectedItem().toString());

Pstatement.setString(3,fatherTextField.getText());

Pstatement.setString(4,passwordField.getText());

Pstatement.setString(5,confirmPasswordField.getText());

Pstatement.setString(6,cityTextField.getText());

Pstatement.setString(7,emailTextField.getText());

if(passwordField.getText().equalsIgnoreCase(confirmPasswordField.getText()))

{

Pstatement.executeUpdate();

JOptionPane.showMessageDialog(null,"Data Registered Successfully");

}

else

{

JOptionPane.showMessageDialog(null,"password did not match");

}

} catch (SQLException e1) {

e1.printStackTrace();

}

}

if(e.getSource()==resetButton)

{

nameTextField.setText("");

genderComboBox.setSelectedItem("Male");

fatherTextField.setText("");

passwordField.setText("");

confirmPasswordField.setText("");

cityTextField.setText("");

emailTextField.setText("");

}

}

}