Steps to create a simple web application using Struts 2.0

Step 1: *create a JSP page containing a **struts <form>** element.*

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
     pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ taglib uri="/struts-tags" prefix="s"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>Numbers</title>
</head>
<body>
      <s:form action="firstaction">
           <s:textfield label="Enter any number from 1 to 10"
                                              name="number"></s:textfield>
           <s:submit label="Submit"></s:submit>
      </s:form>

</body>
</html>
```

This name should be same as **struts.xml** file **action**

Step 2: *create a two JSP pages **Success.jsp** and **Failure.jsp**.*

Success.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>Success Page</title>
</head>
<body>
<h1>You have entered <s:property value="number"/> </h1>
</body>
</html>
```

This value should be same as **property** in Action class

Failure.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>Failure Page</title>
</head>
<body>
<h1>Sorry! Invalid Number</h1>
</body>
</html>
```

Step 3: *Create an **Action** class with an Action Method that returns either "**success**" or "**failure**" based on the action.*

```
package com.wipro.action;

public class NumberAction {

	private int number;

	public int getNumber() {
		return number;
	}

	public void setNumber(int number) {
		this.number = number;
	}

	public String run() {
		if ((number >= 1) && (number <= 10))
			return "success";
		else
			return "failure";
	}

}
```

Step 4: *Create a struts configuration file **struts.xml** inside the **src folder**.*

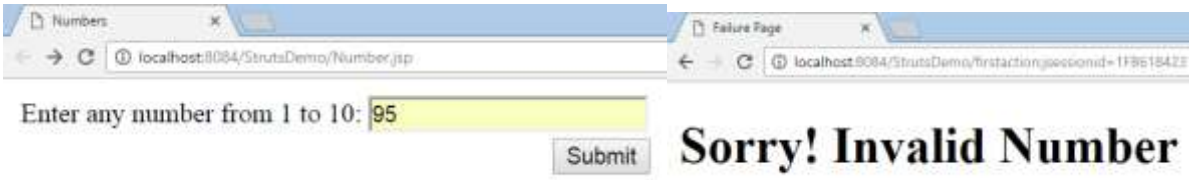
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
   "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
   "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
```

```
        <package name="default" namespace="/" extends="struts-default">
            <action name="firstaction"
class="com.wipro.action.NumberAction" method="run">
                <result name="success">Success.jsp</result>
                <result name="failure">Failure.jsp</result>
            </action>
        </package>
</struts>
```

Step 5: *Create a **web.xml** inside the **Web Content/WEB-INF** folder to configure Struts.*

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns="http://xmlns.jcp.org/xml/ns/javaee"
        xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
        id="WebApp_ID" version="3.1">
        <filter>
            <filter-name>Struts2Filter</filter-name>
            <filter-
class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteF
ilter</filter-class>
        </filter>
        <filter-mapping>
            <filter-name>Struts2Filter</filter-name>
            <url-pattern>/*</url-pattern>
        </filter-mapping>
</web-app>
```

Step 6: *Add the required jar files in **WEB-INF/lib folder** and Run the JSP file.*



Step 7: *create a JSP page containing a* ***struts <form>*** *element.*

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
     pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ taglib uri="/struts-tags" prefix="s"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=ISO-8859-
1">
<title>Person Details</title>
</head>
<body>
     <h1>Person Details</h1>
     <s:form action="insertDB">
     <s:textfield label="First Name" name="firstName"></s:textfield>
     <s:textfield label="Last Name" name="lastName"></s:textfield>
     <s:textfield label="Age" name="age"></s:textfield>
     <s:textfield label="Phone No." name="phone"></s:textfield>
     <s:submit label="Submit"></s:submit>
     </s:form>
</body>
</html>
```

Step 8: *create a two JSP pages* ***Success.jsp*** *and* ***Failure.jsp****.*

Success.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>Success Page</title>
</head>
<body>
<h1>The Details </h1>
<p> <s:property value="firstName"/><br>
<s:property value="lastName"/><br>
<s:property value="age"/><br>
```

These values should be same as **property** in Action class

```
<s:property value="phone"/>
</p>
<h1>Registered Successfully!</h1>
</body>
</html>
```

Failure.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>Failure Page</title>
</head>
<body>
<h1>Sorry! Error Occured</h1>
</body>
</html>
```

Step 9: *Create the following table.*

```
create table persondetails(
firstname varchar2(15),
lastname varchar2(15),
age number(2),phone number(10));
```

Step 10: *Create a* ***DBUtil*** *class that returns a sql Connection.*

```
package com.wipro.util;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBUtil {

     private static Connection con;

     public static Connection getConnection() {
          try {
               if (con == null) {

          Class.forName("oracle.jdbc.driver.OracleDriver");
con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"admin", "admin");
               }

          } catch (Exception e) {
```

```
                System.out.println(e);
            }

            return con;
        }
    }
```

Step 11: *Create an **Action** class with an Action Method that returns either "**success**" or "**failure**" based on the data being inserted into DB.*

```
package com.wipro.action;

import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Statement;

import com.wipro.util.DBUtil;

public class PersonAction {

    private String firstName;
    private String lastName;
    private int age;
    private long phone;

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public long getPhone() {
        return phone;
    }
```

```
	public void setPhone(long phone) {
		this.phone = phone;
	}


public String insert() {
		Connection con = null;
		Statement st = null;
		try {
			con = DBUtil.getConnection();
			st = con.createStatement();

st.executeUpdate("insert into PersonDetails values('" + firstName +
"','" + lastName+ "'," + age + "," + phone + ")");
			return "success";
		} catch (SQLException e) {
			return "failure";
		}
	}
}
```

Step 12: *Create a struts configuration file* ***struts.xml*** *inside the* ***src folder****.*

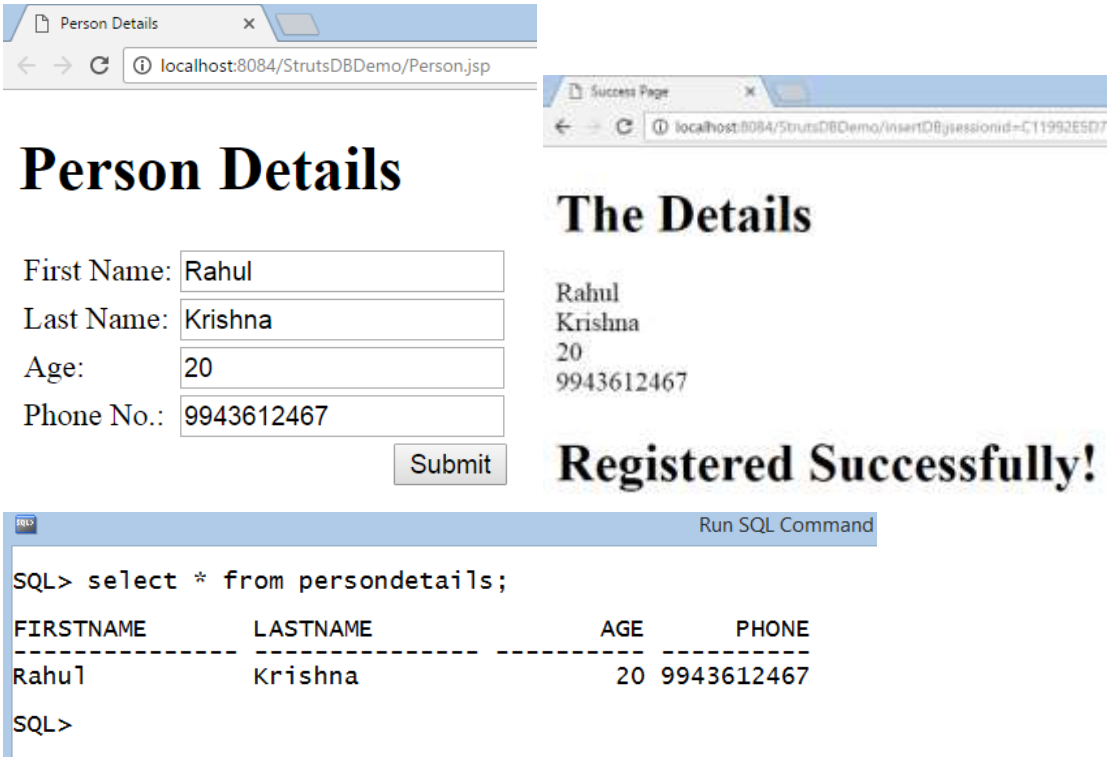
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
   "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
   "http://struts.apache.org/dtds/struts-2.0.dtd">
<struts>
	<package name="default" namespace="/" extends="struts-default">
	<action name="insertDB" class="com.wipro.action.PersonAction"
			method="insert">
			<result name="success">Success.jsp</result>
			<result name="failure">Failure.jsp</result>
		</action>
	</package>
</struts>
```

Step 13: *Create a* ***web.xml*** *inside the* ***Web Content/WEB-INF*** *folder to configure Struts.*

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xmlns="http://xmlns.jcp.org/xml/ns/javaee"
	xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
	id="WebApp_ID" version="3.1">
	<filter>
		<filter-name>Struts2Filter</filter-name>
<filter-
class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteF
ilter</filter-class>
	</filter>
```

```
        <filter-mapping>
            <filter-name>Struts2Filter</filter-name>
            <url-pattern>/*</url-pattern>
        </filter-mapping>
</web-app>
```

Step 14: *Add the required jar files in **WEB-INF/lib folder** and Run the JSP file.*



Steps to create a web application using Hibernate 3

Step 1: *create a HTML page containing a **<form>** element.*

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Item Details</title>
</head>
<body>
    <form name="itemForm" action="ItemServlet" method="post">
        <table>
            <tr>
                <td>Item No</td>
                <td><input type="text" name="itemNo" /></td>
            </tr>
            <tr>
                <td>Item Name</td>
```

This is the name of the **servlet** file given in **@WebServlet**

```
                    <td><input type="text" name="itemName" /></td>
               </tr>
               <tr>
                    <td>Quantity</td>
                    <td><input type="text" name="quantity" /></td>
               </tr>
               <tr>
                    <td>Unit Price</td>
                    <td><input type="text" name="unitPrice" /></td>
               </tr>
               <tr>
          <td><input type="submit" name="option" value="Add" />
<input type="submit" name="option" value="Update" /></td>
<td><input type="submit" name="option" value="Delete" />
<input type="submit" name="option" value="Show" /></td>
               </tr>
          </table>
     </form>
</body>
</html>
```

Step 2: *create a model / bean / POJO class.*

```
package com.wipro.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "ItemDetails")
public class Item {

     @Id
     private int itemNo;

     @Column(length = 15)
     private String itemName;

     private int quantity;

     private float unitPrice;

     public int getItemNo() {
          return itemNo;
     }

     public void setItemNo(int itemNo) {
          this.itemNo = itemNo;
```

These annotations are used to map the Java class to DB Table

This annotation maps the **itemNo** as **PRIMARY KEY**

This annotation is used to specify the column length, name etc.

```
	}

	public String getItemName() {
		return itemName;
	}

	public void setItemName(String itemName) {
		this.itemName = itemName;
	}

	public int getQuantity() {
		return quantity;
	}

	public void setQuantity(int quantity) {
		this.quantity = quantity;
	}

	public float getUnitPrice() {
		return unitPrice;
	}

	public void setUnitPrice(float unitPrice) {
		this.unitPrice = unitPrice;
	}

}
```

Step 3: *create a **HibernateUtil** class that returns a **SessionFactory** object.*

```
package com.wipro.util;

import org.hibernate.cfg.Configuration;
import org.hibernate.SessionFactory;

public class HibernateUtil {
	private static SessionFactory sessionFactory;

	public static SessionFactory getSessionFactory() {
		try {
			sessionFactory = new
Configuration().configure("hibernate.cfg.xml").buildSessionFactory();
		} catch (Exception e) {
			System.out.println(e);

		}
		return sessionFactory;
```

```
        }
}
```

Step 4: *create a **hibernate.cfg.xml** to configure hibernate parameters.*

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate
Configuration
     DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-
configuration-3.0.dtd" -->
<!DOCTYPE hibernate-configuration SYSTEM
"classpath://org/hibernate/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
     <session-factory>
          <!-- Oracle dialect -->
          <property
name="hibernate.dialect">org.hibernate.dialect.OracleDialect</property
>
          <!-- Database connection settings -->
          <property
name="hibernate.connection.driver_class">oracle.jdbc.driver.OracleDriv
er</property>
          <property
name="hibernate.connection.url">jdbc:oracle:thin:@localhost:1521:XE</p
roperty>
          <property
name="hibernate.connection.username">system</property>
          <property
name="hibernate.connection.password">oracle</property>
          <!-- Echo all executed SQL to stdout -->
          <property name="hibernate.show_sql">true</property>
          <property name="hibernate.hbm2ddl.auto">update</property>
     <!-- Enable Hibernate's automatic session context management -->
          <property
name="hibernate.current_session_context_class">thread</property>

<mapping class="com.wipro.model.Item" />
     </session-factory>
</hibernate-configuration>
```

This name should be the bean class name

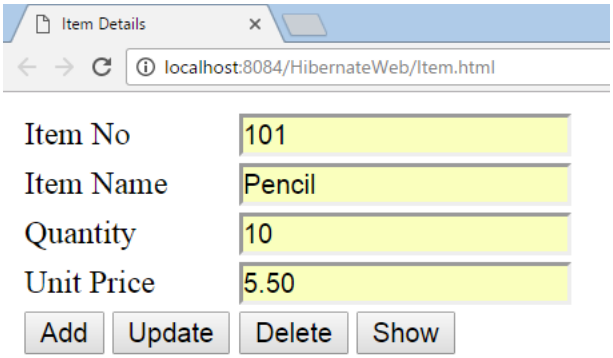
Step 5: *create a **servlet** to process the request.*

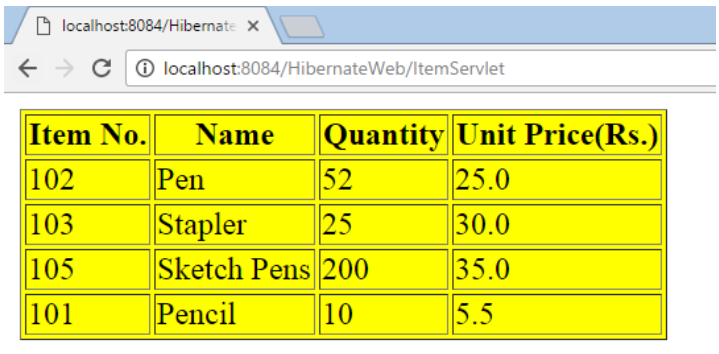
```
package com.wipro.servlet;

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;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import com.wipro.model.Item;
import com.wipro.util.HibernateUtil;

@WebServlet("/ItemServlet")
public class ItemServlet extends HttpServlet {

	private SessionFactory sessionFactory;
	private Session session;

	protected void doPost(HttpServletRequest request,
HttpServletResponse response)
			throws ServletException, IOException {

//Retrieve all the parameters
int itemNo = Integer.parseInt(request.getParameter("itemNo"));
String itemName = request.getParameter("itemName");
int quantity = Integer.parseInt(request.getParameter("quantity"));
float unitPrice = Float.parseFloat(request.getParameter("unitPrice"));
String option = request.getParameter("option");

		//Set the content type and get the writer
		response.setContentType("text/html");
		PrintWriter out = response.getWriter();

		//Populate the object
		Item myItem = new Item();
		myItem.setItemNo(itemNo);
		myItem.setItemName(itemName);
		myItem.setQuantity(quantity);
		myItem.setUnitPrice(unitPrice);

		//Get the hibernate session from the factory
		sessionFactory = HibernateUtil.getSessionFactory();

		//Open the hibernate Session
		session = sessionFactory.openSession();

		//Begin the transaction
		session.getTransaction().begin();

		if (option.equals("Add")) {
```

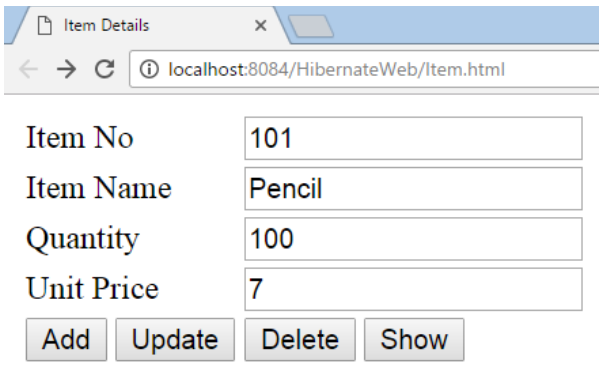
```
                session.save(myItem);
                out.println("<h1>Data Saved Successfully!</h1>");
            }
            if (option.equals("Update")) {
                session.update(myItem);
                out.println("<h1>Data updated Successfully!</h1>");
            }
            if (option.equals("Delete")) {
                session.delete(myItem);
                out.println("<h1>Data deleted Successfully!</h1>");
            }
            if (option.equals("Show")) {

        List<Item> list = session.createCriteria(Item.class).list();

out.print("<table border='1' bgcolor='yellow'>");
out.print("<th>Item No.</th><th>Name</th><th>Quantity</th><th>Unit
Price(Rs.)</th>");

for (Item item : list) {
out.print("<tr><td>" + item.getItemNo() + "</td><td>" +
item.getItemName() + "</td><td>"+ item.getQuantity() + "</td><td>" +
item.getUnitPrice() + "</td></tr>");
                }

out.print("</table>");
            }

            //Commit the transaction
            session.getTransaction().commit();
        }

}
```

Step 6: *Add the required jar files in* ***WEB-INF/lib folder*** *and Run the HTML file.*

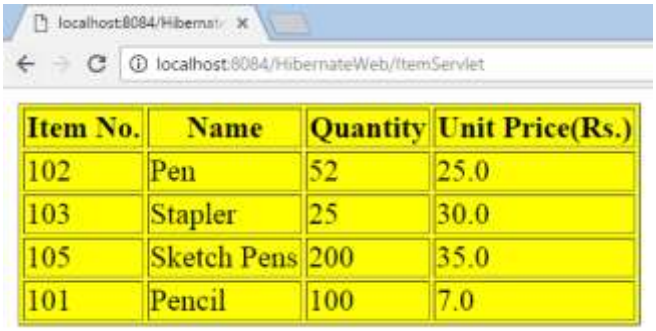




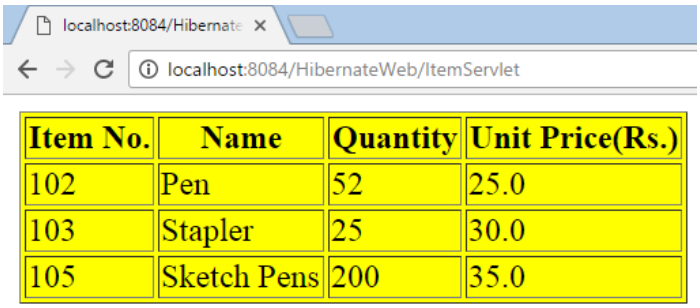
Item No.	Name	Quantity	Unit Price(Rs.)
102	Pen	52	25.0
103	Stapler	25	30.0
105	Sketch Pens	200	35.0
101	Pencil	10	5.5


Item Details
localhost:8084/HibernateWeb/Item.html
Item No
101
Item Name
Pencil
Quantity
100
Unit Price
7
Add
Update
Delete
Show


Data updated Successfully!


localhost:8084/HibernateWeb/ItemServlet
Item No. Name Quantity Unit Price(Rs.)
102 Pen 52 25.0
103 Stapler 25 30.0
105 Sketch Pens 200 35.0
101 Pencil 100 7.0


Data deleted Successfully!


localhost:8084/HibernateWeb/ItemServlet
Item No. Name Quantity Unit Price(Rs.)
102 Pen 52 25.0
103 Stapler 25 30.0
105 Sketch Pens 200 35.0