Dt : 23/5/2022

Assignment-2:(Solution)

Construct Servlet Application using the following Layout:

input.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <form action="choice" method="post">
  <input type="submit" value="Book" name="s1">
  <input type="submit" value="Product" name="s1">

  </form>
</body>
</html>
```

book.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <form action="book" method="post">
  BookCode:<input type="text" name="bcode"><br>
  BookName:<input type="text" name="bname"><br>
  BookAuthor:<input type="text" name="bauthor"><br>
  BookPrice:<input type="text" name="bprice"><br>
  BookQty:<input type="text" name="bqty"><br>
  <input type="submit" value="DisplayBookDetails">
  </form>
</body>
</html>
```

product.html

```
<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <form action="product" method="post">
  ProdCode:<input type="text" name="pcode"><br>
  ProdName:<input type="text" name="pname"><br>
  ProdPrice:<input type="text" name="pprice"><br>
  ProdQty:<input type="text" name="pqty"><br>
  <input type="submit" value="DisplayProductDetails">
  </form>
</body>
</html>
```

ChoiceServlet.java

```
package test;

import java.io.*;

import javax.servlet.*;

import javax.servlet.annotation.*;

@SuppressWarnings("serial")

@WebServlet("/choice")

public class ChoiceServlet extends GenericServlet{

   public void init()throws ServletException{

        //NoCode

   }

   public void service(ServletRequest req,ServletResponse res)

   throws ServletException,IOException{

        String s1 = req.getParameter("s1");

        if(s1.equals("Book")) {

             RequestDispatcher rd = req.getRequestDispatcher("book.html");

             rd.forward(req, res);
```

```
        }else {
                RequestDispatcher rd = req.getRequestDispatcher("product.html");
                rd.forward(req, res);
        }
    }
    public void destroy() {
        //NoCode
    }
}
```

BookServlet.java

```
package test;
import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.*;
@SuppressWarnings("serial")
@WebServlet("/book")
public class BookServlet extends GenericServlet{
  public void init()throws ServletException{
        //NoCode
  }
  public void service(ServletRequest req,ServletResponse res)
  throws ServletException,IOException{
        PrintWriter pw = res.getWriter();
        res.setContentType("text/html");
```

```
            pw.println("====BookDetails====");
            pw.println("<br>BookCode:"+req.getParameter("bcode"));
            pw.println("<br>BookName:"+req.getParameter("bname"));
            pw.println("<br>BookAuthor:"+req.getParameter("bauthor"));
            pw.println("<br>BookPrice:"+req.getParameter("bprice"));
            pw.println("<br>BookQty:"+req.getParameter("bqty"));
            pw.println("<br>");
            RequestDispatcher rd=req.getRequestDispatcher("input.html");
            rd.include(req, res);
    }
    public void destroy() {
            //NoCode
    }
}
```

ProductServlet.java

```
package test;
import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.*;
@SuppressWarnings("serial")
@WebServlet("/product")
public class ProductServlet extends GenericServlet{
    public void init()throws ServletException{
            //NoCode
```

```
  }
  public void service(ServletRequest req,ServletResponse res)
  throws ServletException,IOException{
        PrintWriter pw = res.getWriter();
        res.setContentType("text/html");
        pw.println("====ProductDetails=====");
        pw.println("<br>ProdCode:"+req.getParameter("pcode"));
        pw.println("<br>ProdName:"+req.getParameter("pname"));
        pw.println("<br>ProdPrice:"+req.getParameter("pprice"));
        pw.println("<br>ProdQty:"+req.getParameter("pqty"));
        pw.println("<br>");
        RequestDispatcher rd = req.getRequestDispatcher("input.html");
        rd.include(req, res);
  }
  public void destroy() {
        //NoCode
  }
}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <welcome-file-list>
     <welcome-file>input.html</welcome-file>
  </welcome-file-list>
</web-app>
```

==========

*imp

define Java Bean Class?

=>The classes which are constructed with the following rules are known as Java Bean Classes.

Rule-1 : The class must be implemented from 'java.io.Serializable' interface.

Rule-2 : The variables declared in the class must be private variables.

Rule-3 : The class must be declared with 0-argument constructor or 0-parameter Constructor.

Rule-4 : The class must be declared with 'Setter' and 'Getter' methods.

define Setter methods?

=>The methods which are used to load the data to object are known as 'Setter' methods.

define Getter methods?

=>The methods which are used to get the data from the object are known as Getter methods.

Rule of Constructing Setter and Getter methods:

=>Every Variable in Class must have its own Setter method and Getter method.

Note:

=>These Java Bean Classes will generate bean objects and these bean objects will hold data going on to DataBase or bean Objects will hold data coming from

DataBase.

=>Based on serialization process,the objects are categorized into two types:

(i)Serializable objects

(ii)NonSerializable Objects

(i)Serializable objects:

=>The objects which support Serialization process are known as Serializable objects,which means we can convert Object state into binary Stream.

=>These Serializable objects can travel on N/W.

=>To generate Serializable objects the classes must be implemented from java.io.Serializable interface.

(ii)NonSerializable Objects:

=>The Objects which will not support Serialization process are known as NonSerializable objects.

Ex:

JDBC objects are NonSerializable objects.

**imp*

Summary of Objects generated from CoreJava:

1.User defined Class Object(Serializable object)

2.String Objects(Serializable objects)

3.WrapperClass Objects(Serializable Objects)

4.Array objects(Serializable Objects)

5.Collection<E> objects(Serializable Objects)

6.Map<K,V> objects(Serializable Objects)

7.Enum<E> object(Serializable Object)

==========