/*_____
Assignment no :-10

Title:- Design and implement a business interface with necessary business logic for any web application using EJB.
e.g., Design and implement the web application logic for deposit and withdraw amount transactions using EJB.

Name:-Pravin Jain Roll No:-74 Batch:-T4 Subject:-WTL

_____*/

EJB

Index.html

```
<html>
   <head>
      <title>Implement Bank Application using EJB</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>

      <form method="post" action="transact">
         <h1>Bank Application</h1>
         Enter Amount :<input type="text" name="t1"/><br/><br/>

         <h1>Select Option</h1>
         <input type="radio" name="transcation" value="deposit"/>Deposit<br/><br/>
         <input type="radio" name="transcation" value="withdraw"/>Withdraw<br/><br/>
         <input type="submit" value="submit"/>
      </form>>
      <div>TODO write content</div>
   </body>
</html>
```

Transact.java

```
import bankexam.BankTransactLocal;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


public class transact extends HttpServlet {

    BankTransactLocal bankTransact = lookupBankTransactLocal();

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
```

```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */



            String selectedType = request.getParameter("transcation");
            int amount = Integer.parseInt(request.getParameter("t1"));


            if(selectedType.equals("deposit")){
             bankTransact.deposit(amount);
            }
            if(selectedType.equals("withdraw")){
            int amt = bankTransact.withdraw(amount);
            out.println(amount +" successfully Withdrawn . Your Balance is : Rs."+amt);
            }
        }
    }


    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on
the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
```

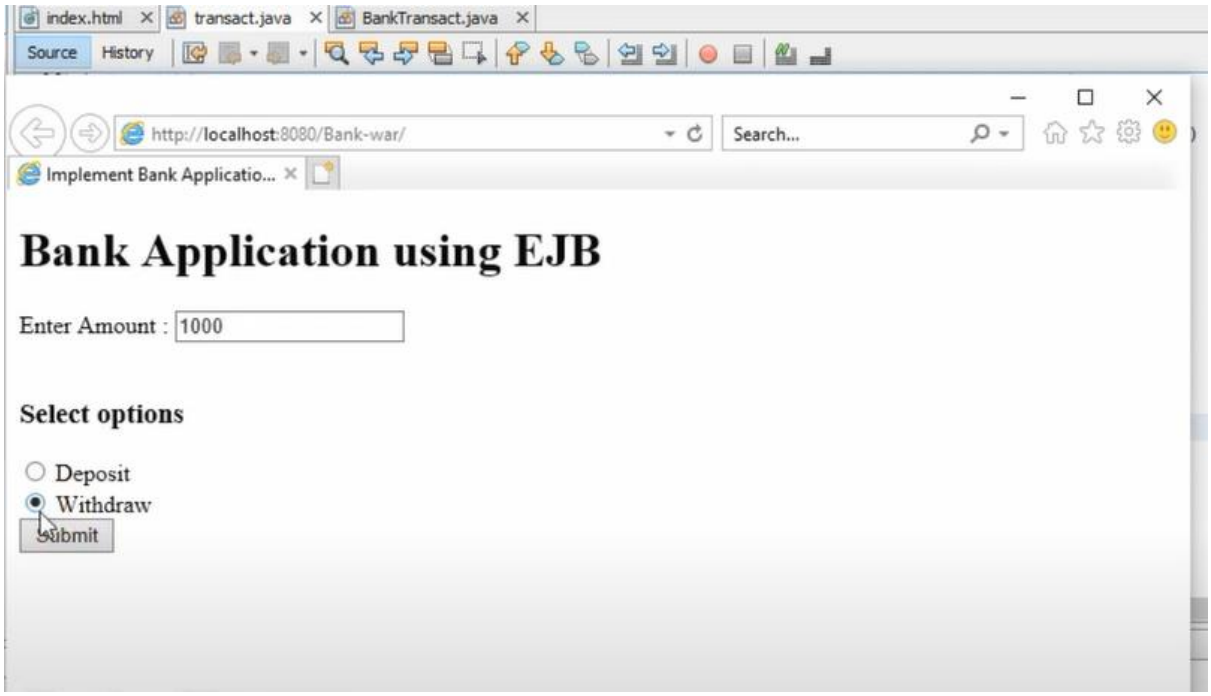
```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
}// </editor-fold>

private BankTransactLocal lookupBankTransactLocal() {
```

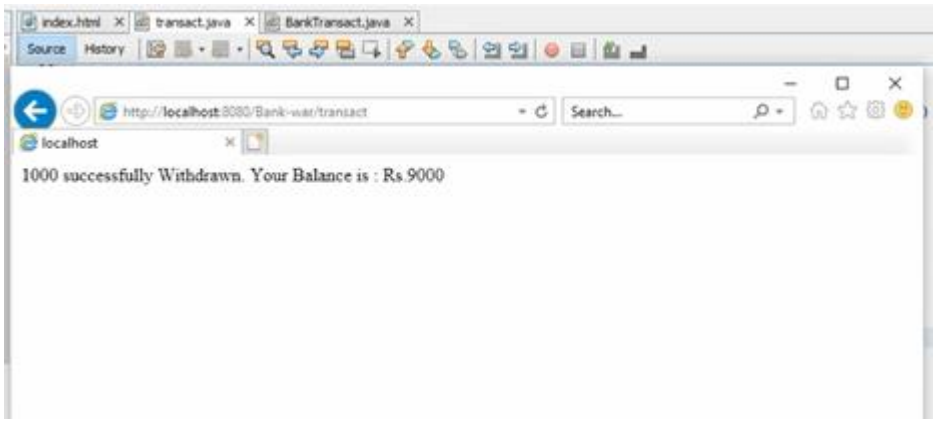
```
        try {
            Context c = new InitialContext();
            return (BankTransactLocal) c.lookup("java:global/Bank/Bank-
ejb/BankTransact!bankexam.BankTransactLocal");
        } catch (NamingException ne) {
            Logger.getLogger(getClass().getName()).log(Level.SEVERE, "exception caught",
ne);
            throw new RuntimeException(ne);
        }
    }
}
```

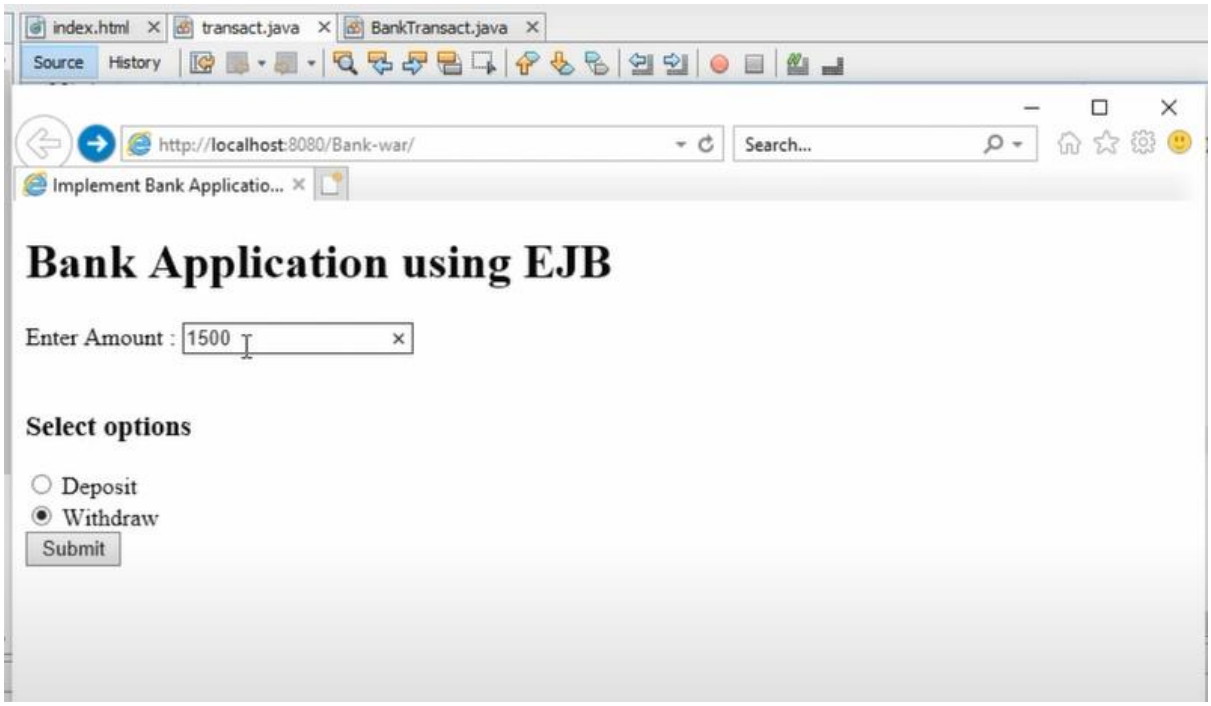
Bank Transact.java

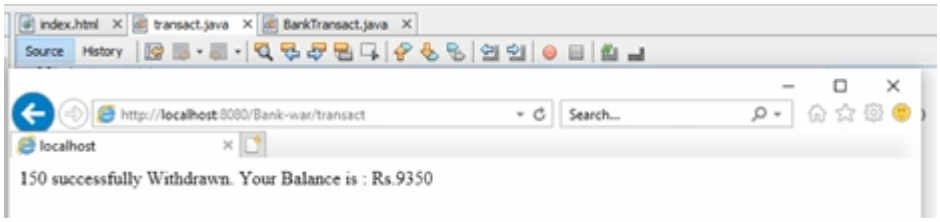
```
package bankexam;

import javax.ejb.Stateful;
@Stateful
public class BankTransact implements BankTransactLocal {

    int balance=10000;
    @Override
    public void deposit(int amount) {
        balance = balance + amount;
    }

    // Add business logic below. (Right-click in editor and choose
    // "Insert Code > Add Business Method")

    @Override
    public int withdraw(int amount) {
        balance = balance - amount;
        return balance ;
    }
}
```

Output


index.html
transact.java
BankTransact.java
Source
History
http://localhost:8080/Bank-war/
Search...
Implement Bank Applicatio...
Bank Application using EJB
Enter Amount : 1000
Select options
Deposit
Withdraw
Submit


index.html
transact.java
BankTransact.java
Source
History
http://localhost:8080/Bank-war/transact
Search...
localhost
1000 successfully Withdrawn. Your Balance is : Rs 9000


index.html
transact.java
BankTransact.java
Source
History
http://localhost:8080/Bank-war/
Search...
Implement Bank Applicatio...
Bank Application using EJB
Enter Amount : 1500
Select options
Deposit
Withdraw
Submit


index.html
transact.java
BankTransact.java
Source
History
http://localhost:8080/Bank-war/transact
Search...
localhost
150 successfully Withdrawn. Your Balance is : Rs.9350