Date

deposit_Scheme.servlet

```
package com.green.bank;

import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;

import com.green.bank.database
{

String account_no, deposit_amount, value;
int year, interest_rate, amount;
```

```
Connection conn;
statement stmt;
Boolean pass_wrong = false;
protected void dopost (HttpServletRequest
request, HttpSerletResponse response)

Throws ServletException, IOException
    {
Account_no = request.getparameter ("acco-
        unt_no");
year = Integer.parseInt (request.get-
        parameter ("year"));
Interest_rate = Integer.parseInt (reque-
st.getparameter) "interest_rate"));
Deposit_amount = request.getParameter
    ("deposit_amount");
value = request.getparameter ("value");

if (deposit_amount.equals ("1,00,000 &#
        2547;"))
    {
    Amount = 100000;
}
else if (deposit_amount.equals ("3,00,000
    &#2547;"))
    {
```

```
        Amount = 300000;
    }
    else if (deposit_amount.equals
("5,00,000 &#2547;"))
    {
        Amount = 500000;
    }

DepositSchemeModel dpmodel = new
DepositSchemeModel();
    dpmodel.setAccount_no(account_no);
    dpmodel.setyear(year);
dpmodel.setInterest_rate(interest_
        rate);
    dpmodel.setAmount(amount);
    dpmodel.setvalue(value);

    try
    {
    JDBC_connect connect = new JDBC_
                Connect();
    Connection conn = connect.getConnection();
DatabaseOperations operations = new
    DatabaseOperations();

Account model am = operations.getAcco
```

```
unt Details (conn, account_no);

if(am.getAmount() >= amount)
{
int main_amount = am.getAmount() -
                  amount;
preparestatement ps = conn.prepareState-
ment ("update amount set amount = ?
      where id = ?");
   ps.setint (1, main_amount);
   ps.setString (2, account_no);
    ps.executeUpdate();
Boolean      allRight = operations.insert
Depositscheme (dpmodel);
Request.setAttribute("Deposistscheme",
  dpmodel);

 Request.setAttribute ("allRight", allRight);
RequestDispatcher rd = request.getRequest-
Dispatcher ("deposit-scheme-progres.jsp");
  Rd.Forward(request, response);
  }
  else
  {
Request.setAttribute("Not.Enough", "yes");
```

```
RequestDispatcher rd = request.get-
RequestDispatcher("single-deposit_sche-
me.jsp?value=" + value);

Rd.Forward(request,response);
  }
} Catch(Exception e)
  {
  e.printStackTrace();
  }
 }
}
```

```
package m6.table_model;
import m6.ConnectionManager;
import javax.swing.*;
import javax.swing.table.AbstractTable
    model;
import java.sql.Connection;
import java.sql.preparedstatement;
import java.sql.Resultset;
import java.sql.SQLException;
import java.util.Vector;

public class AccountTableModel extends
AbstractTableModel
  {
private vector<String> columnNames =
    new vector<>();
  private vector<object[]> data;
  private JFrame ui;

public AccountTableModel(JFrame ui)
  {
columnNames.add("Username");
columnNames.add("Account Holder");
columnNames.add("Account Number");
```

Date

```
columnNames.add("Balance");
data = readFromDb();
this.ui = ui;
}
private vector<object[]> readFromDb()
{
Connection conn = ConnectionManager.
getInstance().getConnection();
vector<object[]> v = new vector<>[];
try
{
PreparedStatement ps = conn.prepareState
ment
("SELECT login.name, login.userna-
me, account.accountNumber, account.
balance FROM account, customer, login"
+ "WHERE account.accountNumber =
customer.accountNumber AND login.user-
name = customer.username" +
"ORDER By login.name Asc"
);
ResultSet rs = ps.executeQuery();
while (rs.next())
{
```

```
String name = rs.getString("login.name");
String username = rs.getString("login.username");
String accountNumber = rs.getString("account.accountNumber");
double balance = rs.getDouble("account.balance");
v.add(new Object[] {username, name, accountNumber, balance});
}
} catch (SQLException e)
{
e.printStackTrace();
JOptionPane.showMessageDialog(ui, "Error! Failed to fetch data");
}
return v;
}
public int getColumnCount()
{
return columnNames.size();
}
public int getRowCount()
{
return data.size();
}
```

```
public String getColumnName(int col)
    {
      return columnNames.get(col);
    }
  public object getValue(int row,
                         int col)
    {
      return data.get(row)[col];
    }
  /*
* JTable uses this method to determine
  the default renderer /
* editor for each cell.
*/
public class getColumnClass(int c)
    {
      return getValueAt(0, c).getClass();
    }

  /*
* Don't need to implement this method
  unless your table's
* editable.
*/
public boolean isCellEditable(int row,
                              int col)
  {
```

```
//Note that the data / cell address is
        constant
//no matter where the cell appears on
        screen
// if (col < 2)
//     {
// return false;
// } else {
// return true;
// }
    return false;
    }
    /*
* Don't need to implement this method
unless your table's
* data can change.
*/
public void setValueAt(Object value, int
        row, int col)
    {
data.get(row)[col] = value;
// FireTableCellUpdated(row, col);
    }

private void printDebugData()
    {
```

```
int numRows = getRowCount();
int numCols = getColumnCount();

For(int i=0 ; i<numRows ; i++)
    {
    int numRows = getRowCount();
    int numCols = getColumnCount();

For(int i=0 ; i<numRows ; i++)
        {
System.out.print("row "+i+" :");
For(int j=0 ; j<numCols ; j++)
    {
System.out.print(" " + data.get(i)[j]
    }
System.out.println();
}
System.out.println("-----");
}
}
```

Create Account servlet

```
import java.io.IOException;
import java.io.printwriter;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Random;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.webser-
    vlet;
import javax.servlet.http.Httpser-
        vlet;
import javax.servlet.http.HttpservletRequest;
import javax.servlet.http.Httpservlet
        Response;
import com.green.bank.model.Account-
        model;
public class CreateAccountServet extends
    Httpservlet
    {
String account_no, first_name, last_name,
address, city, branch, zip, username,
```

```
password ,re-password , phone-number
email , account-type;
int amount;

protected void doPost (HttpServletRequest
request , HttpServletResponse response)
throws ServletException , IOException
{
PrintWriter out = response.getWriter();

First-name = request.getParameter("First-
                name");
last-name = request.getParameter("last-
                name");
address = request.getParameter("address");
city = request.getParameter("city");
branch = request.getParameter("branch");
zip = request.getParameter("zip");
username = request.getParameter("username");
password = request.getParameter("password");
re-password = request.getParameter("
                re-password");
phone-number = request.getParameter
                ("phone");
email = request.getParameter("email");
account-type = request.getParameter
                ("account-type");
```

```
amount = Integer.parseInt(request.
        getParameter("amount"));

// Generating account number
Random rand = new Random();
int random_num = 100000 + rand.
        nextInt(999999);
account_no = first_name.substring(0,2)
last_name.substring(0,2) + random_
        num;
System.out.println(account_no);

// Getting current date
DateFormat df = new SimpleDateFormat
        ("dd/MM/yyyy");
String reg_date = df.format(new Date
        ());

// setting all variables to model class
AccountModel am = new AccountModel();
am.setAccount_no(account_no);
am.setfirst_name(first_name);
am.setLast_name(last_name);
am.setAddress(address);
am.setCity(city);
```

```
am.setBranch(branch);
am.setzip(zip);
am.setusername(username);
am.setpassword(password);
am.setphone_number(phone_number);
am.setEmail(email);
am.setAccount_type(account_type);
am.setAmount(amount);
am.setReg_date(reg_date);

if(password.equals(re_password))
    {
request.setAttribute("Account_details",
        am);
    RequestDispatcher rd = request.
getRequestDispatcher("create_account_
progress.jsp");
    rd.Forward(request,response);
    }
    else
    {
request.setAttribute("not_match","yes");
RequestDispatcher rd = request.getRe-
questDispatcher("create_account.jsp");
    rd.Forward(request,response);}
    }
}
```