Createaccount

```
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.WebServlet
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest
import javax.servlet.http.HttpServletResponse
import com.green.bank.model.AccountModel;

Public class CreateAccountServlet extends
 HttpServlet{

String account-no, first-name, last-name,
address, city, branch, zip, username,
 Password, re-password, phone-numbe
email, account-type;

int amount
```

MOHIT
Page No.:
Date:

```
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_Passwo
phone_number = request.getparameter("phone_numb
email = request.getparameter("email")
account_type = request.getparameter("account_t
amount = Integer.ParseInt(request.getparamete
("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);
Dateformat df = new simpleDate.Format("dd/M
String reg_date = df.format(new Date());

Accountmodel am = new Accountmodel();
am.setAccount_no(account_no);
am.setfirst_name(first_name);
am.setfirst_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.getRequestDispat-
cher("create_account.jsp");
  rd.forward(request, response);
}
}
}
```

MOHIT
Page No.:
Date:

Account Table

```
Package m6.table_model;
import m6.connectionManager;
import javax.swing.*;
import javax.swing.table.AbstractTableModel;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Vector;
Public class AccountTableModel extends Abstract
TableModel {
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");
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.prepareStatement(
"SELECT login.name, login.username, account.
accountNumber, account.balance FROM account,
customer, login" +

        "WHERE account.accountNumber =
customer.accountNumber AND login.
username = 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 vi
}
public int getColumnCount () {
return columnNames.size();
}
Public int getRowCount() {
return data.size();
}
Public String getColumnName (int col) {
return columnNames.get(col);
}
public object getValueAt (int row, int col) {
    return data.get(row)[col];
}
Public Class getColumnClass (int c) {
return getValueAt(0,c).getClass();
}
```

```
public boolean isCellEditable (int row, int col)
{
 if (col<2) {
return false;
} else
{
 return true;
 }
return false;
}
public void setValueAt (Object value, int row,
 int col){
 data.get (row)[col] = value;
 }
 private void printDebugData (){
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 ();
}
```

MOHIT
Page No.:
Date:

System.out.println("*******");
}
}

```
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.webserve
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpservletReques
import javax.servlet.http.HttpservletRespon
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 (HttpServletRequ
 request, HttpservletResponse response)
```

```
Throws ServletException , IoException {
Account_no = request.getParameter ("account_no")
year = Integer.Parseint (request.getParameter
   ("year"));
interest_rate = integer.Parseint (request.getParamel.
er ("interest_rate"));
Deposit_amount = request.getParameter ("deposit_
   amount");
value : request.getParameter ("value");

if (deposit_amount.equals ("1,00,000 &#7547 "))
{
 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 DepositSche.
memodel ();
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 databuse
operations();
AccountModel am = operations.getAccountdetails
(conn, account_no);
if(am.getAmount() >= amount){
int main_amount = am.getAmound() - amount;
PreparedStatement PS = conn.preparestatement
("update amount set amount = ?, where id = ?");
PS.setint(1, main_amount);
PS.setstring(2, account_no);
PS.executeupdate();
Boolean allRight = operations.insertdeposit
scheme(dpmodel);
Request.setAttribute("depositscheme", dpmodel);
Request.setAttribute("allRight", allRight);

RequestDispatcher rd = request.getRequest
Dispatcher("deposit-scheme-Progress.jsp");
Rd.forward(request, response);

MOHIT
Page No.:
Date :

```
}else {
Request.setAttributes("Not-Enough","yes");
RequestDispatcher rd = request.getRequest
dispatcher("single-deposit-scheme.jsp?valu
 +value);
Rd.forward(request, response);
}
}
catch(Exception e){
 e.printStackTrace();
}
}
}
```