JDBC

Student Application

Punith B

CONSOLE LOGIC

Index.java

```
package in.pentagon.studentapp.model;

import java.util.Scanner;

public class Index {
	public static void main(String[] args) {
		Scanner sc=new Scanner(System.in);
		int choice=0;
		System.out.println("Welcome to Student App:");
		do {
			System.out.println("1. SIGNUP");
			System.out.println("2. LOGIN");
			System.out.println("3. FORGOT PASSWORD?");
			System.out.println("4. EXIT");
				choice=sc.nextInt();
			switch(choice) {
			case 1: Signup.signup();
			break;
			case 2: Login.login();
			break;
			case 3: Password.forgot();
			break;
			case 4: System.out.println("Thank you");
			break;
```

```
                default :System.out.println("Invalid coice, please enter a valid value!");
                break;
                }

            }while(choice!=4);
      }
}
```

Login.java

```
package in.pentagon.studentapp.model;

import java.util.ArrayList;
import java.util.Scanner;

import in.pentagon.studentapp.dao.StudentDAO;
import in.pentagon.studentapp.dao.StudentDAOImpl;
import in.pentagon.studentapp.dto.Student;

public class Login {
      public static void login() {
            Scanner sc=new Scanner(System.in);
            StudentDAO sdao=new StudentDAOImpl();
            int choice=0;
            System.out.println("Enter the mail ID");
            String mail=sc.next();
```

```
System.out.println("Enter the password");
String pass=sc.next();
Student s=sdao.getStudent(mail, pass);
if(s!=null) {
          System.out.println("Logged in successfully, Welcome "+s.getName());
          do {
               System.out.println("1. View your account");
               System.out.println("2. Update the Account");
               System.out.println("3. Reset Password");
               System.out.println("4. Search user");
               System.out.println("5. Back to main menu");
               if(s.getId()==1) {
                    System.out.println("6. View All users");
                    System.out.println("7. Delete User");
               }
               choice=sc.nextInt();
               switch(choice) {
               case 1: System.out.println(s);
               break;
               case 2: Update.update(s);
               break;
               case 3: Password.forgot();
               break;
               case 4: System.out.println("Enter the user name:");
               ArrayList<Student> studentsList=sdao.getStudent(sc.next());
               for(Student s2:studentsList) {
                    System.out.println("=========================");
                    System.out.println("Id:"+s2.getId());
```

```
                        System.out.println("Name:"+s2.getName());
                        System.out.println("Branch"+s2.getBranch());
                        System.out.println("==========================");
                    }
                    break;
                    case 5: System.out.println("Going back to main menu...");
                    break;
                    case 6:
                        ArrayList<Student> students=sdao.getStudent();
                        for(Student s1:students) {
                            System.out.println(s1);
                        }
                    break;
                    case 7: System.out.println("Enter the Student ID to be deleted:");
                            boolean res=sdao.deleteStudent(sc.nextInt());
                            if(res) {
                                System.out.println("Data deleted successfully");
                            }
                            else {
                                System.out.println("Failed to delete the data");
                            }
                    default: System.out.println("Invalid choice!");
                    break;
                    }

                }while(choice!=5);
            }
            else {
```

```
                System.out.println("Failed to login!");
            }
        }
}
```

Signup.java

```
package in.pentagon.studentapp.model;

import java.util.Scanner;

import in.pentagon.studentapp.dao.StudentDAO;
import in.pentagon.studentapp.dao.StudentDAOImpl;
import in.pentagon.studentapp.dto.Student;

public class Signup {
      public static void signup() {
            Scanner sc=new Scanner(System.in);
            Student s=new Student();//creation of POJO class object
            StudentDAO sdao=new StudentDAOImpl();


            //collecting the data from the user
            System.out.println("<--Welcome to Signup Page-->");
            System.out.println("Enter the name:");
            //String name=sc.next();
            //s.setName(name);
            s.setName(sc.next());
            System.out.println("Enter the Phone number");
```

```
            s.setPhone(sc.nextLong());
            System.out.println("Enter the Mail ID");
            s.setMail(sc.next());
            System.out.println("Enter the Branch");
            s.setBranch(sc.next());
            System.out.println("Enter the Location");
            s.setLoc(sc.next());
            System.out.println("Set a new Password");
            String password=sc.next();
            System.out.println("Confirm the password");
            String confirmPassword=sc.next();

            if(password.equals(confirmPassword)) {
                  s.setPassword(confirmPassword);
                  boolean res=sdao.insertStudent(s);
                  if(res) {
                        System.out.println("Data added successfully");
                  }
                  else {
                        System.out.println("Failed to add the data");
                  }
            }
            else {
                  System.out.println("Password mismatch!");
            }
      }
}
```

Update.java

```
package in.pentagon.studentapp.model;

import java.util.Scanner;

import in.pentagon.studentapp.dao.StudentDAO;
import in.pentagon.studentapp.dao.StudentDAOImpl;
import in.pentagon.studentapp.dto.Student;

public class Update {
	public static void update(Student s) {
		Scanner sc=new Scanner(System.in);
		StudentDAO sdao=new StudentDAOImpl();
		int choice=0;
		System.out.println("Enter the field to be updated");
		do {
			System.out.println("1. NAME");
			System.out.println("2. PHONE");
			System.out.println("3. MAIL ID");
			System.out.println("4. BRANCH");
			System.out.println("5. LOCATION");
			System.out.println("6. BACK");
			choice=sc.nextInt();
			switch(choice) {
			case 1: System.out.println("Enter the name to be updated:");
				s.setName(sc.next());
```

```
            break;

case 2: System.out.println("Enter the new phone number");
            s.setPhone(sc.nextLong());
            break;

case 3: System.out.println("Enter the new mail");
            s.setMail(sc.next());
            break;

case 4: System.out.println("Enter the new Branch");
            s.setBranch(sc.next());
            break;

case 5: System.out.println("Enter the location");
          s.setLoc(sc.next());
          break;

case 6: System.out.println("Main menu");
            break;

default: System.out.println("Invalid choice!");
           break;
}
boolean res=sdao.updateStudent(s);
if(res) {
      System.out.println("Account updated!");
}
```

```
                else {
                      System.out.println("Failed to update");
                }
           }while(choice!=6);
      }
}
```

Password.java

```
package in.pentagon.studentapp.model;

import java.util.Scanner;

import in.pentagon.studentapp.dao.StudentDAO;
import in.pentagon.studentapp.dao.StudentDAOImpl;
import in.pentagon.studentapp.dto.Student;

public class Password {
      public static void forgot() {
            StudentDAO sdao=new StudentDAOImpl();
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter the Phone number:");
            long phone=sc.nextLong();
            System.out.println("Enter the mail ID");
            String mail=sc.next();
            Student s=sdao.getStudent(phone, mail);
            if(s!=null) {
                  System.out.println("Set a new Password");
```

```
                String password=sc.next();
                System.out.println("Confirm the new password");
                String confirm=sc.next();
                if(password.equals(confirm)) {
                      s.setPassword(password);
                      boolean res=sdao.updateStudent(s);
                      if(res) {
                            System.out.println("Password updated!");
                      }
                      else {
                            System.out.println("Failed to update the password!");
                      }
                }
                else {
                      System.out.println("Password mismatch");
                }
          }else {
                System.out.println("Student not found!");
          }
     }
}
```

DAO LOGIC

StudentDAO.java

```
package in.pentagon.studentapp.dao;

import java.util.ArrayList;

import in.pentagon.studentapp.dto.Student;

public interface StudentDAO {
	public boolean insertStudent(Student s);
	public boolean updateStudent(Student s);
	public boolean deleteStudent(int id);
	public Student getStudent(String mail,String password);
	public Student getStudent(long phone,String mail);
	public ArrayList<Student> getStudent();
	public ArrayList<Student> getStudent(String name);
}
```

StudentDAOImpl.java

```
package in.pentagon.studentapp.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
```

```
import java.util.ArrayList;

import in.pentagon.studentapp.connection.Connector;
import in.pentagon.studentapp.dto.Student;

public class StudentDAOImpl implements StudentDAO{
//All our JDBC logics will be written over here
	private Connection con;

	public StudentDAOImpl() {
		this.con=Connector.requestConnection();
	}

	@Override
	public boolean insertStudent(Student s) {
		PreparedStatement ps=null;
		String query="INSERT INTO STUDENT VALUES(0,?,?,?,?,?,?,SYSDATE())";
		int i=0;
		try {
			ps=con.prepareStatement(query);
			ps.setString(1, s.getName());
			ps.setLong(2, s.getPhone());
			ps.setString(3,s.getMail());
			ps.setString(4, s.getBranch());
			ps.setString(5, s.getLoc());
			ps.setString(6, s.getPassword());
			i=ps.executeUpdate();
		} catch (SQLException e) {
```

```
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            if(i>0) {
                return true;
            }
            else {
                return false;
            }
        }

        @Override
        public boolean updateStudent(Student s) {
            PreparedStatement ps=null;
            String query="UPDATE STUDENT SET NAME=?,PHONE=?,MAIL=?,BRANCH=?,LOCATION=?,PASSWORD=?
    WHERE ID=?";
            int i=0;
            try {
                ps=con.prepareStatement(query);
                ps.setString(1,s.getName());
                ps.setLong(2, s.getPhone());
                ps.setString(3,s.getMail());
                ps.setString(4,s.getBranch());
                ps.setString(5, s.getLoc());
                ps.setString(6, s.getPassword());
                ps.setInt(7, s.getId());
                i=ps.executeUpdate();
            } catch (SQLException e) {
```

```
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		if(i>0)
		{
			return true;
		}
		else {
			return false;
		}
	}

	@Override
	public boolean deleteStudent(int id) {
		// TODO Auto-generated method stub
		return false;
	}

	@Override
	public Student getStudent(String mail, String password) {
		PreparedStatement ps=null;
		String query="SELECT * FROM STUDENT WHERE MAIL=? AND PASSWORD=?";
		Student s=null;
		try {
			ps=con.prepareStatement(query);
			ps.setString(1, mail);
			ps.setString(2, password);
			ResultSet rs=ps.executeQuery();
```

```
		while(rs.next()) {
			s=new Student();
		//	int id=rs.getInt("id");
		//	s.setId(id);
			s.setId(rs.getInt("id"));
			s.setName(rs.getString("name"));
			s.setPhone(rs.getLong("phone"));
			s.setMail(rs.getString("mail"));
			s.setBranch(rs.getString("branch"));
			s.setLoc(rs.getString("location"));
			s.setPassword(rs.getString("password"));
			s.setDate(rs.getString("date"));
		}
	} catch (SQLException e) {
		// TODO Auto-generated catch block
		e.printStackTrace();
	}

	return s;
}

@Override
public Student getStudent(long phone, String mail) {
	PreparedStatement ps=null;
	String query="SELECT * FROM STUDENT WHERE PHONE=? AND MAIL=?";
	Student s=null;
	try {
		ps=con.prepareStatement(query);
```

```
			ps.setLong(1, phone);
			ps.setString(2, mail);
			ResultSet rs=ps.executeQuery();
			while(rs.next()) {
				s=new Student();
				s.setId(rs.getInt("id"));
				s.setName(rs.getString("name"));
				s.setPhone(rs.getLong("phone"));
				s.setMail(rs.getString("mail"));
				s.setBranch(rs.getString("branch"));
				s.setLoc(rs.getString("location"));
				s.setPassword(rs.getString("password"));
				s.setDate(rs.getString("date"));
			}
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		return s;
	}

	@Override
	public ArrayList<Student> getStudent() {
		PreparedStatement ps=null;
		ArrayList<Student> studentsList=new ArrayList<Student>();
		Student s=null;
		String query="SELECT * FROM STUDENT WHERE ID!=1";
		try {
```

```
			ps=con.prepareStatement(query);
			ResultSet rs=ps.executeQuery();
			while(rs.next()) {
				s=new Student();
				s.setId(rs.getInt("id"));
				s.setName(rs.getString("name"));
				s.setPhone(rs.getLong("phone"));
				s.setMail(rs.getString("mail"));
				s.setBranch(rs.getString("branch"));
				s.setLoc(rs.getString("location"));
				s.setPassword(rs.getString("password"));
				s.setDate(rs.getString("date"));
				studentsList.add(s);
			}

		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		return studentsList;
	}

	@Override
	public ArrayList<Student> getStudent(String name) {
		ArrayList<Student> students=new ArrayList<>();
		Student s=null;
		PreparedStatement ps=null;
		String query="SELECT * FROM STUDENT WHERE NAME=?";
```

```
		try {
			ps=con.prepareStatement(query);
			ps.setString(1,name);
			ResultSet rs=ps.executeQuery();
			while(rs.next()) {
				s=new Student();
				s.setId(rs.getInt("id"));
				s.setName(rs.getString("name"));
				s.setPhone(rs.getLong("phone"));
				s.setMail(rs.getString("mail"));
				s.setBranch(rs.getString("branch"));
				s.setLoc(rs.getString("location"));
				s.setPassword(rs.getString("password"));
				s.setDate(rs.getString("date"));
				students.add(s);

			}
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		return students;
	}

}
```

CONNECTION LOGIC

Connector.java

```
package in.ps.studentapp.connection;

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

//connector factory class
public class Connector {
	public static Connection requestConnection() {
		Connection con=null;
		String url="jdbc:mysql://localhost:3306/students";
		String user="root";
		String password="tiger";
		try {
			Class.forName("com.mysql.cj.jdbc.Driver");
			con=DriverManager.getConnection(url, user, password);

		} catch (ClassNotFoundException | SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		return con;
	}
```

```
}
```

DTO LOGIC

Student.java

```
package in.ps.studentapp.dto;

//pojo class
public class Student {
	//instance variables
	private int id;
	private String name;
	private long phone;
	private String mail;
	private String branch;
	private String loc;
	private String password;
	private String date;

	//getters and setters
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getName() {
		return name;
```

```
}
public void setName(String name) {
	this.name = name;
}
public long getPhone() {
	return phone;
}
public void setPhone(long phone) {
	this.phone = phone;
}
public String getMail() {
	return mail;
}
public void setMail(String mail) {
	this.mail = mail;
}
public String getBranch() {
	return branch;
}
public void setBranch(String branch) {
	this.branch = branch;
}
public String getLoc() {
	return loc;
}
public void setLoc(String loc) {
	this.loc = loc;
}
```

```
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}
	public String getDate() {
		return date;
	}
	public void setDate(String date) {
		this.date = date;
	}

	//toString() to print the content of Student
	@Override
	public String toString() {
		return "Student [id=" + id + ", name=" + name + ", phone=" + phone + ", mail=" + mail +
", branch=" + branch
				+ ", loc=" + loc + "]";
	}
}
```