JDBC

Student Application

Punith B

CONSOLE LOGIC

Index.java

```
package in.ps.studentapp.app;

import java.util.Scanner;

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

```
                default:System.out.println("Invalid choice!Try again");
                break;
                }
            }while(in!=3);
      }
}
```

Login.java

```
package in.ps.studentapp.app;

import java.util.Scanner;

import in.ps.studentapp.dao.StudentDAO;
import in.ps.studentapp.dao.StudentDAOImpl;
import in.ps.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 password=sc.next();
            Student s=sdao.getStudent(mail, password);
```

```
if(s!=null) {
	System.out.println("login successful, Welcome "+s.getName());
	do {
		System.out.println("1. View Account");
		System.out.println("2. Update Account");
		System.out.println("3. Search User");
		System.out.println("4. Main menu");
		if(s.getId()==1) {//for admin purpose
			System.out.println("5. Delete User");
			System.out.println("6. View all Students");

		}
		choice=sc.nextInt();
		switch(choice) {
		case 1: System.out.println(s);
		break;
		case 2: Update.update(s);
		break;
		case 3: //logic for Search user
		break;
		case 4: System.out.println("Going back to main menu..");
		break;
		case 5: 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");
                             }
                        break;
                        case 6: ArrayList<Student> studentList=sdao.getStudent();
                        for(Student s1:studentList) {
                             System.out.println(s1);
                        }
                        default:System.out.println("Invalid choice, choose the right one");
                        break;
                        }
                   }while(choice!=4);

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

Signup.java

```
package in.ps.studentapp.app;

import java.util.Scanner;

import in.ps.studentapp.dao.StudentDAO;
import in.ps.studentapp.dao.StudentDAOImpl;
import in.ps.studentapp.dto.Student;
```

```
public class Signup {
	public static void signup() {
		StudentDAO sdao=new StudentDAOImpl();//creating ref of StudentDAO interface
		Scanner sc=new Scanner(System.in);
		//creating an object of pojo class
		Student s=new Student();
		//collecting the data from the user
		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 the new password");
		String password=sc.next();
		System.out.println("Confirm the password");
		String confirmPassword=sc.next();
		if(password.equals(confirmPassword)) {
			s.setPassword(confirmPassword);
			boolean status=sdao.insertStudent(s);
			if(status) {
```

```
                    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.ps.studentapp.app;

import java.util.Scanner;

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

public class Update {
    public static void update(Student s) {
        Scanner sc=new Scanner(System.in);
        StudentDAO sdao=new StudentDAOImpl();
        int in=0;
        boolean status=false;
```

```
do {
    System.out.println("Enter the field, you would like to update:");
    System.out.println("1. NAME");
    System.out.println("2. PHONE");
    System.out.println("3. MAIL");
    System.out.println("4. BRANCH");
    System.out.println("5. LOCATION");
    System.out.println("6. BACK");
    in=sc.nextInt();
    switch(in) {
    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 mail");
    s.setMail(sc.next());
    break;
    case 4: System.out.println("Enter the Branch");
    s.setBranch(sc.next());
    break;
    case 5: System.out.println("Enter the location:");
    s.setLoc(sc.next());
    break;
    case 6: System.out.println("loading...");
    break;
    default:System.out.println("Enter the valid option:");
```

```
                    break;
                    }
                    status=sdao.updateStudent(s);
                    if(status) {
                          System.out.println("Data updated successfully");
                    }else {
                          System.out.println("Failed to update the data");
                    }
              }
              while(in!=6);
        }
  }
```

DAO LOGIC

StudentDAO.java

```
package in.ps.studentapp.dao;

import java.util.ArrayList;

import in.ps.studentapp.dto.Student;

public interface StudentDAO {
	public boolean insertStudent(Student s);//create
	public boolean updateStudent(Student s);//update
	public boolean deleteStudent(Student s);//delete
	public Student getStudent(String mail, String password);//retrieve
	public ArrayList<Student> getStudent();//retrieve
	public ArrayList<Student> getStudent(String name);//retrieve
	public Student getStudent(long phone,String mail);//retrieve
}
```

StudentDAOImpl.java

```
package in.ps.studentapp.dao;
```

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

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

public class StudentDAOImpl implements StudentDAO{
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 res=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());
                res=ps.executeUpdate();
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            if(res>0) {
                return true;
            }
            else {
                return false;
            }
        }

        @Override
        public boolean updateStudent(Student s) {
            PreparedStatement ps=null;
            String query="UPDATE STUDENT SET
NAME=?,PHONE=?,MAIL=?,BRANCH=?,LOC=?,PASSWORD=?,DATE=SYSDATE() WHERE ID=?";
            int res=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());
			res=ps.executeUpdate();
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		if(res>0) {
			return true;
		}
		else {
			return false;
		}
	}

	@Override
	public boolean deleteStudent(Student s) {
		PreparedStatement ps=null;
		String query="DELETE FROM STUDENT WHERE ID=? AND ID!=1";
		int res=0;
		try {
			ps=con.prepareStatement(query);
			ps.setInt(1,id);
			res=ps.executeUpdate();
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		if(res>0) {
```

```
			return true;
		}else {
			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("loc"));
				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;
		String query="SELECT * FROM STUDENT";
		ArrayList<Student> list=new ArrayList<>();
		Student s=null;
		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("loc"));
				s.setPassword(rs.getString("password"));
				s.setDate(rs.getString("date"));
```

```
                list.add(s);
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        // TODO Auto-generated method stub
        return list;
    }

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

```
                s.setPassword(rs.getString("password"));
                s.setDate(rs.getString("date"));
                list.add(s);
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        // TODO Auto-generated method stub
        return list;
    }

    @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();
                //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("loc"));
                    s.setPassword(rs.getString("password"));
                    s.setDate(rs.getString("date"));
               }

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

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 + "]";
	}
}
```