**LIBRARY MANAGEMENT SYSTEM**

**Aim :**

To Implement Library Management System with database using front end tool(Java fx)

**Schema :**

Student(Stu-id , Stu\_name , Dept , Year , Phn\_no)

Book(B-id , B\_name , Author , Section , Copies)

Lb(Stud\_id , B\_id , out\_date , R\_date)

All these three tables are in Normalized form.

**ER-Diagram :**

BOOK

LB

STUDENT

**JAVAFX :**

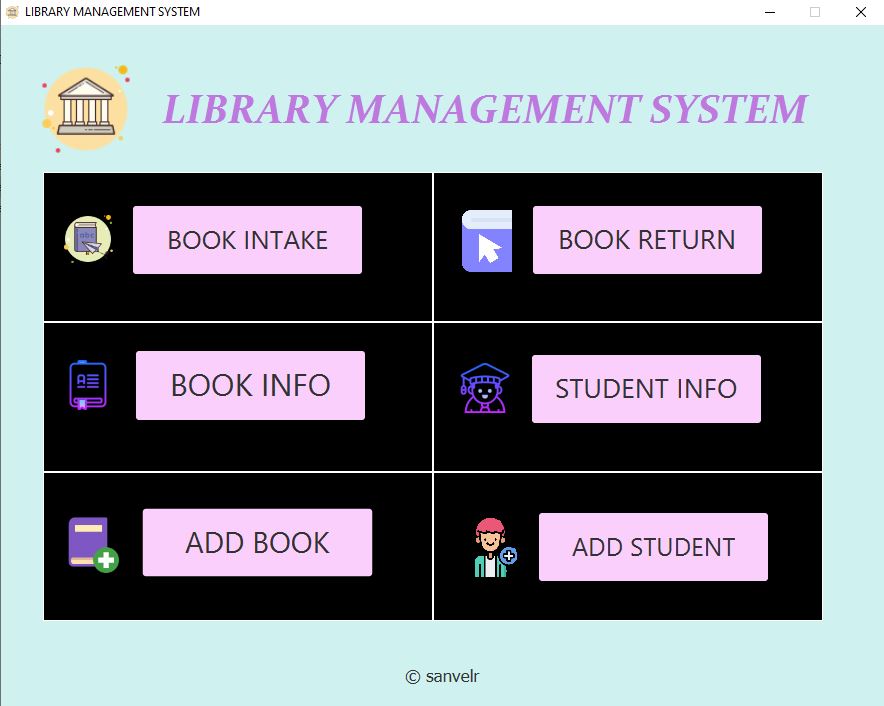
* + JavaFX is a java library that is used to develop Desktop application as well as Rich Internet Application(RIA).
  + Usually it requires three files namely main java file , fxml file , fxml controller java file

**JDBC:**

* + JDBC is a Java API to connect and execute the query with he database
  + In order to work with JDBC we need to download ojdbc6 and load it in project
  + Usually it requires seven steps

1. Import statement
2. Load and Register the driver
3. Create a connection
4. Create a statement
5. Execute the query
6. Analyze the result
7. Close all the connections

**MAIN MENU :**



**Main java Source :**

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.image.Image;

import javafx.stage.Stage;

public class LbManagent extends Application

{

@Override

public void start(Stage stage) throws Exception

{

Image img = new Image("/IMG/1.png");

FXMLLoader root =new FXMLLoader();

root.setLocation(getClass().getResource("FXMLDocument.fxml"));

Parent root1 =root.load();

FXMLDocumentController c = root.getController();

Scene scene = new Scene(root1);

stage.setTitle("LIBRARY MANAGEMENT SYSTEM");

stage.setScene(scene);

stage.getIcons().add(img);

c.sets(stage);

stage.setResizable(false);

stage.show();

}

public static void main(String[] args) {

launch(args);

}

}

**FXML Controller :**

import java.net.URL;

import java.sql.\*;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.image.Image;

import javafx.stage.Modality;

import javafx.stage.Stage;

public class FXMLDocumentController implements Initializable

{

Stage C\_s;

Connection con;

@FXML

void addS(ActionEvent event) throws Exception

{

try{

FXMLLoader root=new FXMLLoader();

root.setLocation(getClass().getResource("AddStudent.fxml"));

Parent root1 =root.load();

AddStudentController c = root.getController();

Image img = new Image("/IMG/1.png");

Stage stage = new Stage();

stage.initModality(Modality.APPLICATION\_MODAL);

Scene scene = new Scene(root1);

stage.setTitle("ADD STUDENT");

stage.setScene(scene);

stage.getIcons().add(img);

stage.setResizable(false);

c.sets(stage);

stage.show();

}

catch(Exception e)

{

Alertmsg.error(C\_s,e.toString());

}

}

@FXML

void addB(ActionEvent event) throws Exception

{

try{

FXMLLoader root=new FXMLLoader();

root.setLocation(getClass().getResource("AddBook.fxml"));

Parent root1 =root.load();

AddBookController c = root.getController();

Image img = new Image("/IMG/1.png");

Stage stage = new Stage();

stage.initModality(Modality.APPLICATION\_MODAL);

Scene scene = new Scene(root1);

stage.setTitle("ADD BOOK");

stage.setScene(scene);

stage.getIcons().add(img);

stage.setResizable(false);

c.sets(stage);

stage.show();

}

catch(Exception e)

{

Alertmsg.error(C\_s,e.toString());

}

}

@FXML

void bookInfo(ActionEvent event)

{

Stage stage= new Stage();

stage.initModality(Modality.APPLICATION\_MODAL);

Image img = new Image("/IMG/1.png");

try

{

FXMLLoader root=new FXMLLoader();

root.setLocation(getClass().getResource("Bookinfo.fxml"));

Parent root1 =root.load();

BookinfoController c = root.getController();

c.sets(stage);

Scene scene = new Scene(root1);

stage.setTitle("Book Info");

stage.setScene(scene);

stage.getIcons().add(img);

stage.setResizable(false);

stage.show();

}

catch(Exception e)

{

Alertmsg.error(C\_s,e.toString());

}

}

@FXML

void studentInfo(ActionEvent event)

{

Stage stage= new Stage();

stage.initModality(Modality.APPLICATION\_MODAL);

Image img = new Image("/IMG/1.png");

try

{

FXMLLoader root=new FXMLLoader();

root.setLocation(getClass().getResource("Studentinfo.fxml"));

Parent root1 =root.load();

StudentinfoController c = root.getController();

Scene scene = new Scene(root1);

c.sets(stage);

stage.setTitle("Student Info");

stage.setScene(scene);

stage.getIcons().add(img);

stage.setResizable(false);

stage.show();

} catch(Exception e)

{

Alertmsg.error(C\_s,e.toString());

}}

@FXML

void intake(ActionEvent event)

{ try

{

Stage stage = new Stage();

FXMLLoader root =new FXMLLoader();

root.setLocation(getClass().getResource("BookIntake.fxml"));

Parent root1 =root.load();

BookIntakeController c = root.getController();

Image img = new Image("/IMG/1.png");

Scene scene = new Scene(root1);

stage.setTitle("Book Intake");

stage.setScene(scene);

stage.getIcons().add(img);

stage.setResizable(false);

c.sets(stage);

stage.show();

}

catch(Exception e){

Alertmsg.error(C\_s,e.toString());

}

}

@FXML

void b\_return(ActionEvent event)

{

try{

Stage stage = new Stage();

FXMLLoader root =new FXMLLoader();

root.setLocation(getClass().getResource("Return.fxml"));

Parent root1 =root.load();

ReturnController c = root.getController();

Image img = new Image("/IMG/1.png");

Scene scene = new Scene(root1);

stage.setTitle("Book Return");

stage.setScene(scene);

stage.getIcons().add(img);

stage.setResizable(false);

c.sets(stage);

stage.show();

}

catch(Exception e)

{

Alertmsg.error(C\_s,e.toString());

}

}

public void sets(Stage s)

{

C\_s=s;

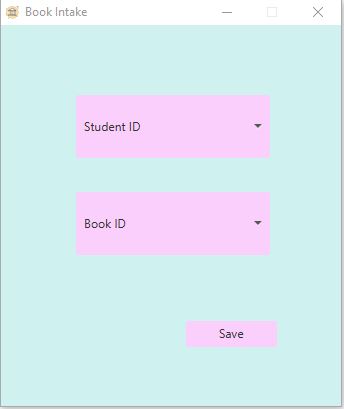
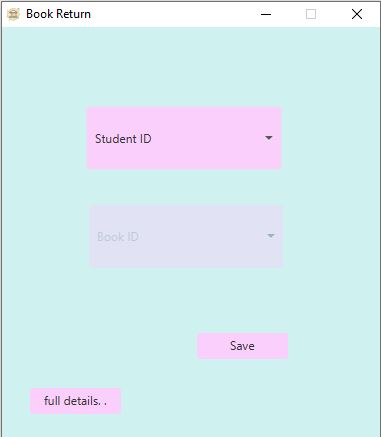
}

@Override

public void initialize(URL url, ResourceBundle rb) {}

}

**Book Intake and Return:**

** **

**Source:**

import java.sql.\*;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.ComboBox;

import javafx.stage.Stage;

public class BookIntakeController implements Initializable {

Stage C\_s;

@FXML

private ComboBox<Integer> sid;

@FXML

private ComboBox<Integer> bid;

@FXML

void save(ActionEvent event)

{

if(sid.getSelectionModel().isEmpty() || bid.getSelectionModel().isEmpty())

{

Alertmsg.error(C\_s, "Select the fields...! ");

return;

}

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");

PreparedStatement st = con.prepareStatement("insert into lb(stud\_id,book\_id,out\_d) values(?,?,sysdate)");

st.setInt(1, sid.getValue());

st.setInt(2, bid.getValue());

st.execute();

PreparedStatement st2 = con.prepareStatement("update book set copies\_a= copies\_a-1 where b\_id = ?");

st2.setInt(1, bid.getValue());

st2.execute();

st.close();

con.close();

Alertmsg.message(C\_s, "Saved Successfully...! ");

C\_s.close();

}

catch(SQLIntegrityConstraintViolationException e)

{

Alertmsg.error(C\_s, "Book has been already taken.. ");

}

catch(Exception e)

{

Alertmsg.error(C\_s,e.toString());

}

}

void sets(Stage s)

{

C\_s=s;

}

@Override

public void initialize(URL url, ResourceBundle rb)

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");

Statement st = con.createStatement();

ResultSet rs = st.executeQuery("select stu\_id from student order by stu\_id");

while(rs.next()){

sid.getItems().add(rs.getInt(1));

}

ResultSet rs2 = st.executeQuery("select b\_id from book where copies\_a > 0 order by b\_id");

while(rs2.next()){

bid.getItems().add(rs2.getInt(1));

}

rs.close();

rs2.close();

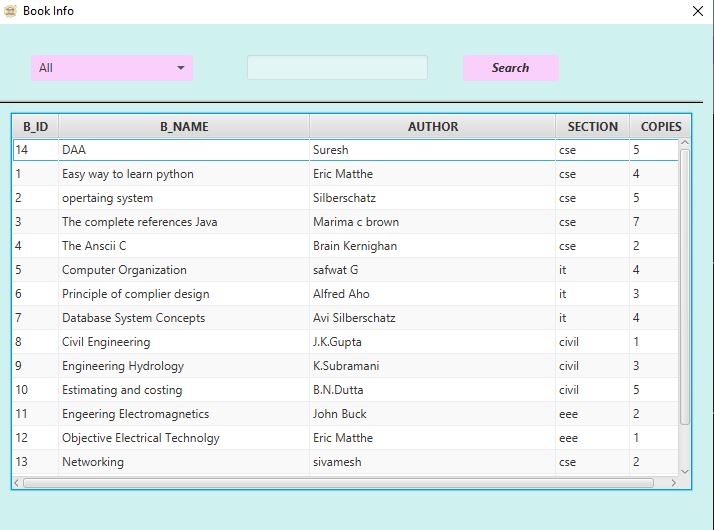
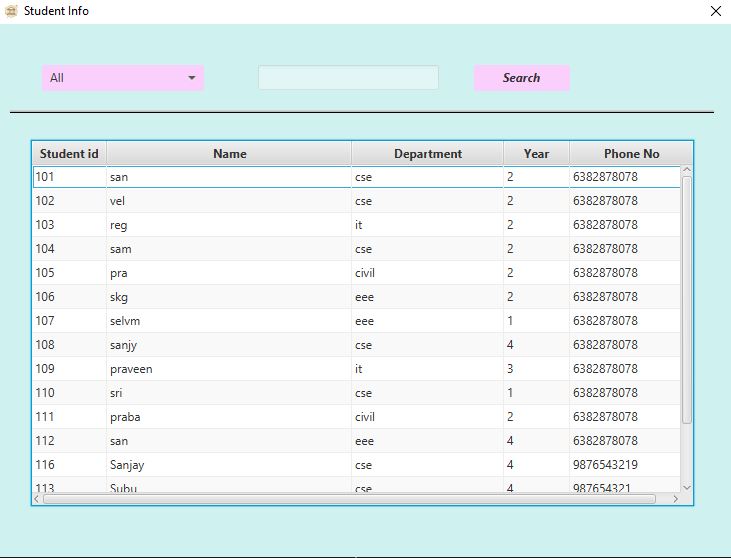
st.close();

}

catch(Exception e)

{ Alertmsg.error(C\_s,e.toString()); } } }

**STUDENT AND BOOK INFO :**

** **

**Source:**

import java.net.URL;

import java.sql.\*;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

public class BookinfoController implements Initializable {

Stage C\_s;

String new\_i;

@FXML

private TableView<Book> table;

@FXML

private TableColumn<Book, Integer> b\_id;

@FXML

private TableColumn<Book, String> b\_name;

@FXML

private TableColumn<Book, String> author;

@FXML

private TableColumn<Book, String> section;

@FXML

private TableColumn<Book, Integer> copies;

@FXML

private ComboBox<String> combo;

@FXML

private TextField text;

@FXML

private Button bt;

@FXML

void search(ActionEvent event)

{

if(text.getText().equals("") && !(new\_i.equals("All")))

{

Alertmsg.error(C\_s, "Please fill the Search column");

return;

}

table.getItems().clear();

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");

if(new\_i.equals("All"))

{

table.setItems(getProduct());

}

else if(new\_i.equals("B\_ID"))

{

ObservableList<Book> products = FXCollections.observableArrayList();

PreparedStatement st = con.prepareStatement("select \* from book where b\_id = ?");

st.setInt(1, Integer.valueOf(text.getText()));

ResultSet rs = st.executeQuery();

while(rs.next()){

products.add(new Book(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getInt(5)));

}

st.close();

rs.close();

table.getItems().addAll(products);

}

else if(new\_i.equals("B\_Name"))

{

ObservableList<Book> products = FXCollections.observableArrayList();

PreparedStatement st = con.prepareStatement("select \* from book where b\_name like ?");

st.setString(1, "%"+text.getText()+"%");

ResultSet rs = st.executeQuery();

while(rs.next()){

products.add(new Book(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getInt(5)));

}

st.close();

rs.close();

table.getItems().addAll(products);

}

else if(new\_i.equals("Author"))

{

ObservableList<Book> products = FXCollections.observableArrayList();

PreparedStatement st = con.prepareStatement("select \* from book where author like ?");

st.setString(1, "%"+text.getText()+"%");

ResultSet rs = st.executeQuery();

while(rs.next()){

products.add(new Book(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getInt(5)));

}

st.close();

rs.close();

table.getItems().addAll(products);

}

else if(new\_i.equals("Section"))

{

ObservableList<Book> products = FXCollections.observableArrayList();

PreparedStatement st = con.prepareStatement("select \* from book where sec = ?");

st.setString(1, text.getText());

ResultSet rs = st.executeQuery();

while(rs.next()){

products.add(new Book(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getInt(5)));

}

st.close();

rs.close();

table.getItems().addAll(products);

}

} catch (Exception ex)

{

Alertmsg.error(C\_s, ex.toString());

}

}

@Override

public void initialize(URL url, ResourceBundle rb)

{

b\_id.setCellValueFactory(new PropertyValueFactory<>("b\_id"));

b\_name.setCellValueFactory(new PropertyValueFactory<>("b\_name"));

author.setCellValueFactory(new PropertyValueFactory<>("author"));

section.setCellValueFactory(new PropertyValueFactory<>("section"));

copies.setCellValueFactory(new PropertyValueFactory<>("copies"));

table.setItems(getProduct());

combo.getItems().addAll("All","B\_ID","B\_Name","Author","Section");

combo.getSelectionModel().selectedItemProperty().addListener((v,old,new\_i)->

{

text.setText("");

text.setDisable(false);

bt.setDisable(false);

if(new\_i.equals("All"))

{

this.new\_i = new\_i;

text.setDisable(true);

}

else if(new\_i.equals("B\_ID"))

this.new\_i = new\_i;

else if(new\_i.equals("B\_Name"))

this.new\_i = new\_i;

else if(new\_i.equals("Author"))

this.new\_i = new\_i;

else if(new\_i.equals("Section"))

this.new\_i = new\_i;

});

combo.getSelectionModel().selectFirst();

}

public void sets(Stage s)

{ C\_s = s; }

public ObservableList<Book> getProduct()

{

bt.setDisable(true);

ObservableList<Book> products = FXCollections.observableArrayList();

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");

Statement st = con.createStatement();

ResultSet rs = st.executeQuery("select \* from book");

while(rs.next()){

products.add(new Book(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getInt(5)));

}

rs.close();

st.close();

} catch (Exception ex)

{

Alertmsg.error(C\_s, ex.toString());

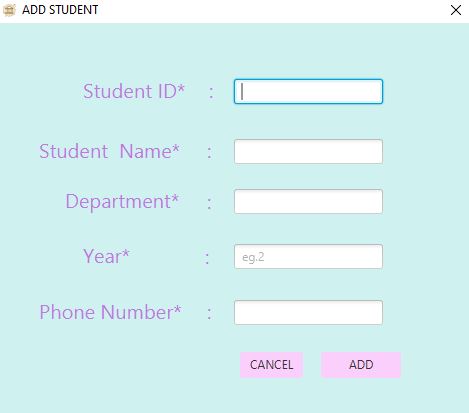
}

return products;

}

}

**ADD STUDENT AND BOOK :**

** **

**Source :**

import java.net.URL;

import java.sql.\*;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.Initializable;

import javafx.fxml.FXML;

import javafx.scene.control.\*;

import javafx.stage.Stage;

public class AddStudentController implements Initializable

{

Connection c;

PreparedStatement ps;

Stage C\_s;

@FXML

private TextField t1;

@FXML

private TextField t2;

@FXML

private TextField t3;

@FXML

private TextField t4;

@FXML

private TextField t5;

@FXML

private Button add;

@FXML

private Button cancel;

@FXML

void add(ActionEvent event)

{

if(t1.getText().equals("") || t2.getText().equals("") || t3.getText().equals("") || t4.getText().equals("") || t5.getText().equals(""))

{

Alertmsg.message(C\_s, "Fill all the fields.....");

return;

}

Connection con;

PreparedStatement ps;

try{

Class.forName("oracle.jdbc.driver.OracleDriver");

con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");

ps = con.prepareStatement("insert into student values(?,?,?,?,?)");

ps.setInt(1,Integer.valueOf(t1.getText()));

ps.setString(2,t2.getText());

ps.setString(3, t3.getText());

ps.setInt(4, Integer.valueOf(t4.getText()));

ps.setLong(5, Long.valueOf(t5.getText()));

ps.execute();

ps.close();

con.close();

Alertmsg.message(C\_s, "STUDENT ADDED SUCCESSFULLY.....");

C\_s.close();

}

catch(SQLIntegrityConstraintViolationException e)

{

Alertmsg.error(C\_s," Student Id already Exist..");

}

catch(Exception E)

{

Alertmsg.error(C\_s,E.toString());

}

}

@FXML

void cancel(ActionEvent event)

{

C\_s.close();

}

void sets(Stage S)

{

C\_s = S;

}

@Override

public void initialize(URL url, ResourceBundle rb)

{

}

}

**Result :**

Thus the Library Management System using database is implemented Successfully.