**Comsats University Islamabad, Lahore campus**

**Assignment #4**

**Assigned Date: 6 Dec 2024**

**Due Date: 20 Dec 2024**

**Submitted To:**

**Sir Shahid Bhatti**

**Submitted by:**

**Zain Yaqoob**

**Sp24-BSE-125**

**Samiullah Ashfaq**

**Sp24-BSE-107**

**Muhammad Asad Ali  
Sp24-BSE-068**

**Section: A**

a\/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers.Employee;

import DBConnection.DBHandler;

import Model.EmployeeDetails;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ResourceBundle;

import javafx.collections.ObservableList;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

import DBConnection.DBHandler;

import Model.LeavedEmployeeDetails;

import java.sql.ResultSet;

import java.sql.SQLException;

import javafx.collections.FXCollections;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.StageStyle;

import javax.swing.JOptionPane;

public class All\_Employee\_LeavedController implements Initializable {

@FXML

private TableView<LeavedEmployeeDetails> tableEmployee;

@FXML

private TableColumn<LeavedEmployeeDetails, String> col\_id;

@FXML

private TableColumn<LeavedEmployeeDetails, String> col\_name;

@FXML

private TableColumn<LeavedEmployeeDetails, String> col\_nic;

@FXML

private TableColumn<LeavedEmployeeDetails, String> col\_phonenumber;

@FXML

private TableColumn<LeavedEmployeeDetails, String> col\_emgtel;

@FXML

private TableColumn<LeavedEmployeeDetails, String> col\_leav\_date;

@FXML

private Button btn\_back;

private ObservableList<LeavedEmployeeDetails> data;

private Connection connection;

private DBHandler handler;

private PreparedStatement pst;

@Override

public void initialize(URL url, ResourceBundle rb) {

connection = handler.connectDB();

data = FXCollections.observableArrayList();

try {

ResultSet rs = connection.createStatement().executeQuery("SELECT \* FROM leaved\_employee");

while (rs.next()) {

// get string from db

data.add(new LeavedEmployeeDetails(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(6)));

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

}

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

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

col\_nic.setCellValueFactory(new PropertyValueFactory<>("nic"));

col\_phonenumber.setCellValueFactory(new PropertyValueFactory<>("tel"));

col\_emgtel.setCellValueFactory(new PropertyValueFactory<>("emgTel"));

col\_leav\_date.setCellValueFactory(new PropertyValueFactory<>("date"));

tableEmployee.setItems(null);

tableEmployee.setItems(data);

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Employee/Employee\_Menu.fxml"));

Scene scene = new Scene(root);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.setScene(scene);

stu\_Menu.show();

stu\_Menu.setResizable(false);

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers.Employee;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Button;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.input.MouseEvent;

import DBConnection.DBHandler;

import Model.EmployeeDetails;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

import javax.swing.JOptionPane;

public class All\_Employee\_LivingController implements Initializable {

@FXML

private TableView<EmployeeDetails> tableEmployee;

@FXML

private TableColumn<EmployeeDetails, String> col\_id;

@FXML

private TableColumn<EmployeeDetails, String> col\_name;

@FXML

private TableColumn<EmployeeDetails, String> col\_nic;

@FXML

private TableColumn<EmployeeDetails, String> col\_phonenumber;

@FXML

private TableColumn<EmployeeDetails, String> col\_emgtel;

@FXML

private Button btn\_back;

private ObservableList<EmployeeDetails> data;

private Connection connection;

private DBHandler handler;

private PreparedStatement pst;

@Override

public void initialize(URL url, ResourceBundle rb) {

connection = handler.connectDB();

data = FXCollections.observableArrayList();

try {

ResultSet rs = connection.createStatement().executeQuery("SELECT \* FROM register\_employee");

while (rs.next()) {

data.add(new EmployeeDetails(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5)));

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

}

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

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

col\_nic.setCellValueFactory(new PropertyValueFactory<>("nic"));

col\_phonenumber.setCellValueFactory(new PropertyValueFactory<>("tel"));

col\_emgtel.setCellValueFactory(new PropertyValueFactory<>("emgTel"));

tableEmployee.setItems(null);

tableEmployee.setItems(data);

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Employee/Employee\_Menu.fxml"));

Scene scene = new Scene(root);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.setScene(scene);

stu\_Menu.show();

stu\_Menu.setResizable(false);

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers.Student;

import java.net.URL;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Alert;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javax.swing.JOptionPane;

import DBConnection.DBHandler;

import Model.LeavedStudentDetails;

import java.io.IOException;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

public class All\_Student\_LeavedController implements Initializable {

private ObservableList<LeavedStudentDetails> data;

@FXML

private TableColumn<LeavedStudentDetails, String> col\_id;

@FXML

private TableView<LeavedStudentDetails> tableStudent;

@FXML

private TableColumn<LeavedStudentDetails, String> col\_name;

@FXML

private TableColumn<LeavedStudentDetails, String> col\_nsbmid;

@FXML

private TableColumn<LeavedStudentDetails, String> col\_email;

@FXML

private TableColumn<LeavedStudentDetails, String> col\_phonenumber;

@FXML

private TableColumn<LeavedStudentDetails, String> col\_nic;

@FXML

private TableColumn<LeavedStudentDetails, String> col\_address;

@FXML

private TableColumn<LeavedStudentDetails, String> col\_g\_name;

@FXML

private TableColumn<LeavedStudentDetails, String> col\_g\_tel;

@FXML

private TableColumn<LeavedStudentDetails, String> col\_l\_date;

private Connection connection;

private DBHandler handler;

private PreparedStatement pst;

@FXML

private Button btn\_back;

@Override

public void initialize(URL url, ResourceBundle rb) {

connection = handler.connectDB();

data = FXCollections.observableArrayList();

try {

ResultSet rs = connection.createStatement().executeQuery("SELECT \* FROM leaved\_students");

while (rs.next()) {

data.add(new LeavedStudentDetails(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(6), rs.getString(7), rs.getString(8), rs.getString(9),rs.getString(10)));

}

} catch (SQLException ex) {

System.out.println(ex.getMessage());

showAlert("Database Error", "An error occurred while connecting to the database", Alert.AlertType.ERROR);

}

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

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

col\_nsbmid.setCellValueFactory(new PropertyValueFactory<>("nsbmId"));

col\_email.setCellValueFactory(new PropertyValueFactory<>("email"));

col\_phonenumber.setCellValueFactory(new PropertyValueFactory<>("phoneNumber"));

col\_nic.setCellValueFactory(new PropertyValueFactory<>("nic"));

col\_address.setCellValueFactory(new PropertyValueFactory<>("address"));

col\_g\_name.setCellValueFactory(new PropertyValueFactory<>("guardName"));

col\_g\_tel.setCellValueFactory(new PropertyValueFactory<>("guardTel"));

col\_l\_date.setCellValueFactory(new PropertyValueFactory<>("date"));

tableStudent.setItems(null);

tableStudent.setItems(data);

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Student/Student\_Menu.fxml"));

Scene scene = new Scene(root);

stu\_Menu.setScene(scene);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.show();

stu\_Menu.setResizable(false);

}

private void showAlert(String title, String content, Alert.AlertType alertType) {

Alert alert = new Alert(alertType);

alert.setTitle(title);

alert.setHeaderText(null);

alert.setContentText(content);

alert.showAndWait();

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers.Student;

import Model.StudentDetails;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.collections.ObservableList;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Alert;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.input.MouseEvent;

import DBConnection.DBHandler;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javafx.collections.FXCollections;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

import javax.swing.JOptionPane;

public class All\_Student\_LivingController implements Initializable {

private ObservableList<StudentDetails> data;

@FXML

private TableView<StudentDetails> tableStudent;

@FXML

private TableColumn<StudentDetails, String> col\_id1;

@FXML

private TableColumn<StudentDetails, String> col\_name1;

@FXML

private TableColumn<StudentDetails, String> col\_nsbmid1;

@FXML

private TableColumn<StudentDetails, String> col\_email1;

@FXML

private TableColumn<StudentDetails, String> col\_phonenumber1;

@FXML

private TableColumn<StudentDetails, String> col\_nic1;

@FXML

private TableColumn<StudentDetails, String> col\_address1;

@FXML

private TableColumn<StudentDetails, String> col\_g\_name1;

@FXML

private TableColumn<StudentDetails, String> col\_g\_tel1;

private Connection connection;

private DBHandler handler;

private PreparedStatement pst;

@FXML

private Button btn\_back;

@Override

public void initialize(URL url, ResourceBundle rb) {

connection = handler.connectDB();

data = FXCollections.observableArrayList();

try {

ResultSet rs = connection.createStatement().executeQuery("SELECT \* FROM register\_students");

while (rs.next()) {

data.add(new StudentDetails(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(6), rs.getString(7), rs.getString(8), rs.getString(9)));

}

} catch (SQLException ex) {

System.out.println(ex.getMessage());

showAlert("Database Error", "An error occurred while connecting to the database", Alert.AlertType.ERROR);

}

col\_id1.setCellValueFactory(new PropertyValueFactory<>("id"));

col\_name1.setCellValueFactory(new PropertyValueFactory<>("name"));

col\_nsbmid1.setCellValueFactory(new PropertyValueFactory<>("nsbmId"));

col\_email1.setCellValueFactory(new PropertyValueFactory<>("email"));

col\_phonenumber1.setCellValueFactory(new PropertyValueFactory<>("phoneNumber"));

col\_nic1.setCellValueFactory(new PropertyValueFactory<>("nic"));

col\_address1.setCellValueFactory(new PropertyValueFactory<>("address"));

col\_g\_name1.setCellValueFactory(new PropertyValueFactory<>("guardName"));

col\_g\_tel1.setCellValueFactory(new PropertyValueFactory<>("guardTel"));

tableStudent.setItems(null);

tableStudent.setItems(data);

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Student/Student\_Menu.fxml"));

Scene scene = new Scene(root);

stu\_Menu.setScene(scene);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.show();

stu\_Menu.setResizable(false);

}

private void showAlert(String title, String content, Alert.AlertType alertType) {

Alert alert = new Alert(alertType);

alert.setTitle(title);

alert.setHeaderText(null);

alert.setContentText(content);

alert.showAndWait();

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package DBConnection;

import java.sql.Connection;

import java.sql.DriverManager;

import javax.swing.JOptionPane;

public class DBHandler {

Connection con = null;

public static Connection connectDB() {

try {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/hostel\_management", "root", "zainyaqoob");

return con;

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, JOptionPane.ERROR\_MESSAGE);

System.out.println(ex.toString());

return null;

}

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers.Employee;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

import java.time.format.DateTimeFormatter;

import DBConnection.DBHandler;

import Model.EmployeeDetails;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.StageStyle;

import javax.swing.\*;

public class Delete\_EmployeeController implements Initializable {

private String leavedID;

@FXML

private Button btn\_update\_employee;

@FXML

private Button btn\_refersh;

@FXML

private TextField emp\_id;

@FXML

private TextField reg\_txt\_emp\_emgtel;

@FXML

private TextField reg\_txt\_emp\_username;

@FXML

private TextField reg\_txt\_emp\_nic;

@FXML

private TextField reg\_txt\_emp\_phnmb;

@FXML

private TableView<EmployeeDetails> tableEmployee;

@FXML

private TableColumn<EmployeeDetails, String> col\_id;

@FXML

private TableColumn<EmployeeDetails, String> col\_name;

@FXML

private TableColumn<EmployeeDetails, String> col\_nic;

@FXML

private TableColumn<EmployeeDetails, String> col\_phonenumber;

@FXML

private TableColumn<EmployeeDetails, String> col\_emgtel;

@FXML

private Button btn\_back;

private ObservableList<EmployeeDetails> data;

private Connection connection;

private DBHandler handler;

private PreparedStatement pst;

@FXML

private DatePicker dateLeaved;

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

handler = new DBHandler();

emp\_id.setDisable(true);

reg\_txt\_emp\_username.setDisable(true);

reg\_txt\_emp\_nic.setDisable(true);

reg\_txt\_emp\_phnmb.setDisable(true);

reg\_txt\_emp\_emgtel.setDisable(true);

}

@FXML

private void deleteEmployeeButtonAction(MouseEvent event) {

if (dateLeaved.getValue() == null) {

showAlert("Warning", "Please Select a date before deleting", Alert.AlertType.WARNING);

return;

}

updateLeavedEmployeeDB();

String delete = "DELETE from register\_employee where id = ?";

connection = handler.connectDB();

try {

pst = connection.prepareStatement(delete);

pst.setString(1, emp\_id.getText());

leavedID = emp\_id.getText();

pst.executeUpdate();

showAlert("Message", "Deleted Selected Employee", Alert.AlertType.INFORMATION);

clearFields();

autoRefresh();

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, ex);

}

}

private void clearFields() {

emp\_id.setText("");

reg\_txt\_emp\_username.setText("");

reg\_txt\_emp\_nic.setText("");

reg\_txt\_emp\_phnmb.setText("");

reg\_txt\_emp\_emgtel.setText("");

dateLeaved.setValue(null);

}

private void autoRefresh() {

connection = handler.connectDB();

data = FXCollections.observableArrayList();

try {

ResultSet rs = connection.createStatement().executeQuery("SELECT \* FROM register\_employee");

while (rs.next()) {

data.add(new EmployeeDetails(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5)));

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

}

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

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

col\_nic.setCellValueFactory(new PropertyValueFactory<>("nic"));

col\_phonenumber.setCellValueFactory(new PropertyValueFactory<>("tel"));

col\_emgtel.setCellValueFactory(new PropertyValueFactory<>("emgTel"));

tableEmployee.setItems(null);

tableEmployee.setItems(data);

}

private void updateLeavedEmployeeDB() {

String query = "INSERT INTO leaved\_employee (id,name,nic,tel,emg\_tel,leave\_date) VALUES(?,?,?,?,?,?)";

connection = handler.connectDB();

try {

pst = connection.prepareStatement(query);

pst.setString(1, emp\_id.getText());

pst.setString(2, reg\_txt\_emp\_username.getText());

pst.setString(3, reg\_txt\_emp\_nic.getText());

pst.setString(4, reg\_txt\_emp\_phnmb.getText());

pst.setString(5, reg\_txt\_emp\_emgtel.getText());

if (dateLeaved.getValue() == null) {

showAlert("Warning", "Please select leave date", Alert.AlertType.WARNING);

return;

}

if (dateLeaved.getValue() != null) {

String formattedDate = dateLeaved.getValue().format(DateTimeFormatter.ofPattern("yyyy-MM-dd"));

pst.setString(6, formattedDate);

} else {

pst.setNull(6, java.sql.Types.DATE);

}

pst.executeUpdate();

autoRefresh();

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, ex);

}

}

@FXML

private void refreshButtionClickAction(MouseEvent event) {

connection = handler.connectDB();

data = FXCollections.observableArrayList();

try {

ResultSet rs = connection.createStatement().executeQuery("SELECT \* FROM register\_employee");

while (rs.next()) {

data.add(new EmployeeDetails(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5)));

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

}

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

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

col\_nic.setCellValueFactory(new PropertyValueFactory<>("nic"));

col\_phonenumber.setCellValueFactory(new PropertyValueFactory<>("tel"));

col\_emgtel.setCellValueFactory(new PropertyValueFactory<>("emgTel"));

tableEmployee.setItems(null);

tableEmployee.setItems(data);

}

@FXML

private void displaySelectedAction(MouseEvent event) {

EmployeeDetails Employee = tableEmployee.getSelectionModel().getSelectedItem();

if (Employee == null) {

showAlert("Warning", "Nothing Selected", Alert.AlertType.WARNING);

} else {

String id = Employee.getId();

String name = Employee.getName();

String nic = Employee.getNic();

String tel = Employee.getTel();

String emg\_tel = Employee.getEmgTel();

emp\_id.setText(id);

reg\_txt\_emp\_username.setText(name);

reg\_txt\_emp\_nic.setText(nic);

reg\_txt\_emp\_phnmb.setText(tel);

reg\_txt\_emp\_emgtel.setText(emg\_tel);

}

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Employee/Employee\_Menu.fxml"));

Scene scene = new Scene(root);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.setScene(scene);

stu\_Menu.show();

stu\_Menu.setResizable(false);

}

private void showAlert(String title, String content, Alert.AlertType alertType) {

Alert alert = new Alert(alertType);

alert.setTitle(title);

alert.setHeaderText(null);

alert.setContentText(content);

alert.showAndWait();

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers.Student;

import DBConnection.DBHandler;

import Model.StudentDetails;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

import javax.swing.JOptionPane;

import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

public class Delete\_StudentController implements Initializable {

@FXML

private TableView<StudentDetails> tableStudent;

@FXML

private TableColumn<StudentDetails, String> col\_id;

@FXML

private TableColumn<StudentDetails, String> col\_name;

@FXML

private TableColumn<StudentDetails, String> col\_nsbmid;

@FXML

private TableColumn<StudentDetails, String> col\_nic;

@FXML

private TextField reg\_txt\_nic;

@FXML

private TextField reg\_txt\_id;

@FXML

private TextField reg\_txt\_username;

@FXML

private TextField reg\_txt\_nsbmid;

@FXML

private Button btn\_delete\_student;

@FXML

private Button btn\_refersh;

private ObservableList<StudentDetails> data;

private Connection connection;

private DBHandler handler;

private PreparedStatement pst;

private String leavedID;

@FXML

private TextField reg\_txt\_guardname;

@FXML

private TextField reg\_txt\_guardtel;

@FXML

private TextField reg\_txt\_email;

@FXML

private TextField reg\_txt\_phnmb;

@FXML

private TextField reg\_txt\_address;

@FXML

private TableColumn<StudentDetails, String> col\_email;

@FXML

private TableColumn<StudentDetails, String> col\_phonenumber;

@FXML

private TableColumn<StudentDetails, String> col\_address;

@FXML

private TableColumn<StudentDetails, String> col\_g\_name;

@FXML

private TableColumn<StudentDetails, String> col\_g\_tel;

@FXML

private DatePicker dateLeaved;

@FXML

private Button btn\_back;

@Override

public void initialize(URL url, ResourceBundle rb) {

handler = new DBHandler();

reg\_txt\_nic.setDisable(true);

reg\_txt\_id.setDisable(true);

reg\_txt\_guardname.setDisable(true);

reg\_txt\_guardtel.setDisable(true);

reg\_txt\_email.setDisable(true);

reg\_txt\_phnmb.setDisable(true);

reg\_txt\_address.setDisable(true);

reg\_txt\_username.setDisable(true);

reg\_txt\_nsbmid.setDisable(true);

}

@FXML

private void deleteStudentButtonAction(MouseEvent event) {

if (dateLeaved.getValue() == null) {

showAlert("Warning", "Please select a date before deleting", Alert.AlertType.WARNING);

return;

}

updateLeavedStudentDB();

String delete = "DELETE from register\_students where id = ?";

connection = handler.connectDB();

try {

pst = connection.prepareStatement(delete);

pst.setString(1, reg\_txt\_id.getText());

leavedID = reg\_txt\_id.getText();

pst.executeUpdate();

showAlert("Message", "Deleted Selected Student", Alert.AlertType.INFORMATION);

clearFields();

autoRefresh();

} catch (SQLException ex) {

System.out.println(ex.getMessage());

showAlert("Database Error", "An error occurred while connecting to the database", Alert.AlertType.ERROR);

}

}

private void updateLeavedStudentDB() {

String query = "INSERT INTO leaved\_students (id, name, nsbmid, email, phonenumber, nic, address, guardname, guardtel, leave\_date) VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

connection = handler.connectDB();

try {

pst = connection.prepareStatement(query);

pst.setString(1, reg\_txt\_id.getText());

pst.setString(2, reg\_txt\_username.getText());

pst.setString(3, reg\_txt\_nsbmid.getText());

pst.setString(4, reg\_txt\_email.getText());

pst.setString(5, reg\_txt\_phnmb.getText());

pst.setString(6, reg\_txt\_nic.getText());

pst.setString(7, reg\_txt\_address.getText());

pst.setString(8, reg\_txt\_guardname.getText());

pst.setString(9, reg\_txt\_guardtel.getText());

LocalDate leaveDate = dateLeaved.getValue(); // Get LocalDate from DatePicker

if (leaveDate != null) {

String formattedDate = leaveDate.format(DateTimeFormatter.ofPattern("yyyy-MM-dd"));

pst.setString(10, formattedDate);

} else {

pst.setString(10, null); // Handle null date

}

pst.executeUpdate();

autoRefresh();

} catch (SQLException ex) {

System.out.println(ex.getMessage());

showAlert("Database Error", "An error occurred while connecting to the database", Alert.AlertType.ERROR);

}

}

private void autoRefresh() {

connection = handler.connectDB();

data = FXCollections.observableArrayList();

try {

ResultSet rs = connection.createStatement().executeQuery("SELECT \* FROM register\_students");

while (rs.next()) {

data.add(new StudentDetails(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(6), rs.getString(7), rs.getString(8), rs.getString(9)));

}

} catch (SQLException ex) {

System.out.println(ex.getMessage());

showAlert("Database Error", "An error occurred while connecting to the database", Alert.AlertType.ERROR);

}

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

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

col\_nsbmid.setCellValueFactory(new PropertyValueFactory<>("nsbmId"));

col\_email.setCellValueFactory(new PropertyValueFactory<>("email"));

col\_phonenumber.setCellValueFactory(new PropertyValueFactory<>("phoneNumber"));

col\_nic.setCellValueFactory(new PropertyValueFactory<>("nic"));

col\_address.setCellValueFactory(new PropertyValueFactory<>("address"));

col\_g\_name.setCellValueFactory(new PropertyValueFactory<>("guardName"));

col\_g\_tel.setCellValueFactory(new PropertyValueFactory<>("guardTel"));

tableStudent.setItems(null);

tableStudent.setItems(data);

}

@FXML

private void refreshButtionClickAction(MouseEvent event) {

connection = handler.connectDB();

data = FXCollections.observableArrayList();

try {

ResultSet rs = connection.createStatement().executeQuery("SELECT \* FROM register\_students");

while (rs.next()) {

data.add(new StudentDetails(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(6), rs.getString(7), rs.getString(8), rs.getString(9)));

}

} catch (SQLException ex) {

System.out.println(ex.getMessage());

showAlert("Database Error", "An error occurred while connecting to the database", Alert.AlertType.ERROR);

}

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

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

col\_nsbmid.setCellValueFactory(new PropertyValueFactory<>("nsbmId"));

col\_email.setCellValueFactory(new PropertyValueFactory<>("email"));

col\_phonenumber.setCellValueFactory(new PropertyValueFactory<>("phoneNumber"));

col\_nic.setCellValueFactory(new PropertyValueFactory<>("nic"));

col\_address.setCellValueFactory(new PropertyValueFactory<>("address"));

col\_g\_name.setCellValueFactory(new PropertyValueFactory<>("guardName"));

col\_g\_tel.setCellValueFactory(new PropertyValueFactory<>("guardTel"));

tableStudent.setItems(null);

tableStudent.setItems(data);

}

@FXML

private void displaySelectedAction(MouseEvent event) {

StudentDetails student = tableStudent.getSelectionModel().getSelectedItem();

if (student == null) {

showAlert("Warning", "Nothing Selected!", Alert.AlertType.WARNING);

} else {

String id = student.getId();

String name = student.getName();

String nsbmid = student.getNsbmId();

String email = student.getEmail();

String phonenumber = student.getPhoneNumber();

String nic = student.getNIC();

String address = student.getAddress();

String g\_name = student.getGuardName();

String g\_tel = student.getGuardTel();

reg\_txt\_id.setText(id);

reg\_txt\_username.setText(name);

reg\_txt\_nsbmid.setText(nsbmid);

reg\_txt\_email.setText(email);

reg\_txt\_phnmb.setText(phonenumber);

reg\_txt\_nic.setText(nic);

reg\_txt\_address.setText(address);

reg\_txt\_guardname.setText(g\_name);

reg\_txt\_guardtel.setText(g\_tel);

}

}

private void clearFields() {

reg\_txt\_username.setText("");

reg\_txt\_nsbmid.setText("");

reg\_txt\_email.setText("");

reg\_txt\_phnmb.setText("");

reg\_txt\_nic.setText("");

reg\_txt\_address.setText("");

reg\_txt\_guardname.setText("");

reg\_txt\_guardtel.setText("");

dateLeaved.setValue(null);

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Student/Student\_Menu.fxml"));

Scene scene = new Scene(root);

stu\_Menu.setScene(scene);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.show();

stu\_Menu.setResizable(false);

}

private void showAlert(String title, String content, Alert.AlertType alertType) {

Alert alert = new Alert(alertType);

alert.setTitle(title);

alert.setHeaderText(null);

alert.setContentText(content);

alert.showAndWait();

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Model;

import javafx.beans.property.SimpleStringProperty;

import javafx.beans.property.StringProperty;

public class EmployeeDetails {

// model class holding getters, setters and properties

private StringProperty id;

private StringProperty name;

private StringProperty nic;

private StringProperty tel;

private StringProperty emgTel;

public EmployeeDetails(String id,String name, String nic, String tel, String emgTel) {

this.id = new SimpleStringProperty(id);

this.name = new SimpleStringProperty(name);

this.nic = new SimpleStringProperty(nic);

this.tel = new SimpleStringProperty(tel);

this.emgTel = new SimpleStringProperty(emgTel);

}

public String getId() {

return id.get();

}

public String getName() {

return name.get();

}

public String getNic() {

return nic.get();

}

public String getTel() {

return tel.get();

}

public String getEmgTel() {

return emgTel.get();

}

public void setId(String value) {

id.set(value);

}

public void setName(String value) {

name.set(value);

}

public void setNic(String value) {

nic.set(value);

}

public void setTel(String value) {

tel.set(value);

}

public void seEmgTel(String value) {

emgTel.set(value);

}

public StringProperty idProperty() { return id; }

public StringProperty nameProperty() { return name; }

public StringProperty nicProperty() { return nic; }

public StringProperty telProperty() { return tel; }

public StringProperty emgTelProperty() { return emgTel; }

}

package Controllers.Employee;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ResourceBundle;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Button;

import javafx.scene.control.TextField;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

import javax.swing.JOptionPane;

import DBConnection.DBHandler;

import java.util.logging.Level;

import java.util.logging.Logger;

public class Employee\_feeController implements Initializable {

@FXML

private Button btn\_back;

@FXML

private TextField employeeID;

@FXML

private TextField employeeFee;

@FXML

private Button submit;

@FXML

private ComboBox<String> month;

@FXML

private ComboBox<String> year;

private Connection connection;

private DBHandler handler;

private PreparedStatement pst;

@Override

public void initialize(URL url, ResourceBundle rb) {

handler = new DBHandler();

ObservableList<String> months = FXCollections.observableArrayList(

"January", "February", "March", "April", "May", "June", "July", "August",

"September", "October", "November", "December"

);

month.setItems(months);

ObservableList<String> years = FXCollections.observableArrayList(

"2020", "2021", "2022", "2023", "2024", "2025", "2026", "2027", "2028",

"2029", "2030"

);

year.setItems(years);

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage emp\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Employee/Employee\_Menu.fxml"));

Scene scene = new Scene(root);

emp\_Menu.setScene(scene);

emp\_Menu.initStyle(StageStyle.TRANSPARENT);

emp\_Menu.show();

emp\_Menu.setResizable(false);

}

@FXML

private void submitButtonAction(MouseEvent event) {

String id = employeeID.getText();

String fee = employeeFee.getText();

String selectedMonth = month.getValue();

String selectedYear = year.getValue();

if (id == null || id.trim().isEmpty() || fee == null || fee.trim().isEmpty() || selectedMonth == null || selectedYear == null) {

showAlert("Warning", "Nothing Selected", Alert.AlertType.WARNING);

return;

}

try {

Double.parseDouble(fee);

} catch (NumberFormatException e) {

showAlert("Warning", "Salary must be a numeric value.", Alert.AlertType.WARNING);

return;

}

connection = handler.connectDB();

String checkEmployeeQuery = "SELECT \* FROM register\_employee WHERE id = ?"; // Assuming a table named 'employees'

try {

pst = connection.prepareStatement(checkEmployeeQuery);

pst.setString(1, id);

ResultSet rs = pst.executeQuery();

if (!rs.next()) {

showAlert("Error 404", "Employee ID doesn't exist", Alert.AlertType.WARNING);

return;

}

} catch (SQLException ex) {

Logger.getLogger(Employee\_feeController.class.getName()).log(Level.SEVERE, null, ex);

showAlert("Database Error", "Couldn't connect to database", Alert.AlertType.ERROR);

return;

}

String insertQuery = "INSERT INTO employee\_fee(employeeid, year, salary, month) VALUES(?, ?, ?, ?)";

try {

pst = connection.prepareStatement(insertQuery);

pst.setString(1, id);

pst.setString(2, selectedYear);

pst.setString(3, fee);

pst.setString(4, selectedMonth);

pst.executeUpdate();

showAlert("Message", "Salary record added successfully", Alert.AlertType.INFORMATION);

setTextRefresh();

} catch (SQLException ex) {

Logger.getLogger(Employee\_feeController.class.getName()).log(Level.SEVERE, null, ex);

showAlert("Error", "Error saving salary record", Alert.AlertType.ERROR);

} finally {

try {

if (pst != null) pst.close();

if (connection != null) connection.close();

} catch (SQLException ex) {

Logger.getLogger(Employee\_feeController.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

private void setTextRefresh() {

employeeID.clear();

employeeFee.clear();

month.setValue(null);

year.setValue(null);

}

private void showAlert(String title, String content, Alert.AlertType alertType) {

Alert alert = new Alert(alertType);

alert.setTitle(title);

alert.setHeaderText(null);

alert.setContentText(content);

alert.showAndWait();

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers.Employee;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.EventHandler;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

public class Employee\_MenuController implements Initializable {

double xoffset, yoffset;

@FXML

private Button btn\_back;

@FXML

private Button addNewEmployee;

@FXML

private Button updateEmployee;

@FXML

private Button deleteEmployee;

@FXML

private Button EmployeeFees;

@FXML

private Button allEmployeeLiving;

@FXML

private Button leavedEmployee;

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

}

@FXML

private void addNewEmployeeAction(MouseEvent event) throws IOException {

addNewEmployee.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Employee/New\_Employee.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

}

@FXML

private void updateEmployeeAction(MouseEvent event) throws IOException {

updateEmployee.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Employee/Update\_Employee.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

}

@FXML

private void deleteEmployeeAction(MouseEvent event) throws IOException {

deleteEmployee.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Employee/Delete\_Employee.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

}

@FXML

private void allEmployeeAction(MouseEvent event) throws IOException {

allEmployeeLiving.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Employee/All\_Employee\_Living.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

}

@FXML

private void leavedEmployeeAction(MouseEvent event) throws IOException {

leavedEmployee.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Employee/All\_Employee\_Leaved.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/MenuComponent.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

root.setOnMousePressed(new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event){

xoffset = event.getSceneX();

yoffset = event.getSceneY();

}

});

root.setOnMouseDragged(new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event){

login.setX(event.getScreenX() - xoffset);

login.setY(event.getScreenY() - yoffset);

}

});

}

@FXML

private void employee\_feeAction(MouseEvent event) throws IOException {

EmployeeFees.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Employee/Employee\_fee.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Model;

import javafx.beans.property.SimpleStringProperty;

import javafx.beans.property.StringProperty;

public class LeavedEmployeeDetails {

private StringProperty id;

private StringProperty name;

private StringProperty nic;

private StringProperty tel;

private StringProperty emgTel;

private StringProperty date;

public LeavedEmployeeDetails(String id,String name, String nic, String tel, String emgTel,String date) {

this.id = new SimpleStringProperty(id);

this.name = new SimpleStringProperty(name);

this.nic = new SimpleStringProperty(nic);

this.tel = new SimpleStringProperty(tel);

this.emgTel = new SimpleStringProperty(emgTel);

this.date = new SimpleStringProperty(date);

}

public String getId() {

return id.get();

}

public String getName() {

return name.get();

}

public String getNic() {

return nic.get();

}

public String getTel() {

return tel.get();

}

public String getEmgTel() {

return emgTel.get();

}

public String getDate() {

return date.get();

}

public void setId(String value) {

id.set(value);

}

public void setName(String value) {

name.set(value);

}

public void setNic(String value) {

nic.set(value);

}

public void setTel(String value) {

tel.set(value);

}

public void seEmgTel(String value) {

emgTel.set(value);

}

public void setDate(String value) {

date.set(value);

}

public StringProperty idProperty() { return id; }

public StringProperty nameProperty() { return name; }

public StringProperty nicProperty() { return nic; }

public StringProperty telProperty() { return tel; }

public StringProperty emgTelProperty() { return emgTel; }

public StringProperty DateProperty() { return date; }

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Model;

import javafx.beans.property.SimpleStringProperty;

import javafx.beans.property.StringProperty;

public class LeavedStudentDetails {

private StringProperty id;

private StringProperty name;

private StringProperty nsbmid;

private StringProperty email;

private StringProperty phoneNumber;

private StringProperty nic;

private StringProperty address;

private StringProperty guardName;

private StringProperty guardTel;

private StringProperty date;

public LeavedStudentDetails(String id,String name, String nsbmid, String email, String phoneNumber, String nic, String address, String guardName, String guardTel,String date)

{

this.id = new SimpleStringProperty(id);

this.name = new SimpleStringProperty(name);

this.nsbmid = new SimpleStringProperty(nsbmid);

this.email = new SimpleStringProperty(email);

this.phoneNumber = new SimpleStringProperty(phoneNumber);

this.nic = new SimpleStringProperty(nic);

this.address = new SimpleStringProperty(address);

this.guardName = new SimpleStringProperty(guardName);

this.guardTel = new SimpleStringProperty(guardTel);

this.date = new SimpleStringProperty(date);

}

public String getId() {

return id.get();

}

public String getName() {

return name.get();

}

public String getNsbmId() {

return nsbmid.get();

}

public String getEmail() {

return email.get();

}

public String getPhoneNumber() {

return phoneNumber.get();

}

public String getNIC() {

return nic.get();

}

public String getAddress() {

return address.get();

}

public String getGuardName() {

return guardName.get();

}

public String getGuardTel() {

return guardTel.get();

}

public String getDate() {

return date.get();

}

public void setId(String value) {

id.set(value);

}

public void setName(String value) {

name.set(value);

}

public void setNsbmid(String value) {

nsbmid.set(value);

}

public void setEmail(String value) {

email.set(value);

}

public void setPhoneNumber(String value) {

phoneNumber.set(value);

}

public void setNic(String value) {

nic.set(value);

}

public void setAddress(String value) {

address.set(value);

}

public void setGuardName(String value) {

guardName.set(value);

}

public void setGuardTel(String value) {

guardTel.set(value);

}

public void setDate(String value) {

date.set(value);

}

public StringProperty idProperty() { return id; }

public StringProperty nameProperty() { return name; }

public StringProperty nsbmIdProperty() { return nsbmid; }

public StringProperty emailProperty() { return email; }

public StringProperty phoneNumberProperty() { return phoneNumber; }

public StringProperty nicProperty() { return nic; }

public StringProperty addressProperty() { return address; }

public StringProperty guardNameProperty() { return guardName; }

public StringProperty guardTelProperty() { return guardTel; }

public StringProperty DateProperty() { return date; }

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import java.util.logging.Level;

import java.util.logging.Logger;

import javafx.application.Platform;

import javafx.event.EventHandler;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.AnchorPane;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

public class LoadingController implements Initializable {

double xoffset, yoffset;

@FXML

AnchorPane ap;

class ShowSplashScreen extends Thread {

@Override

public void run() {

try {

Thread.sleep(2000);

Platform.runLater(() -> {

Stage stage = new Stage();

Parent root = null;

try {

root = FXMLLoader.load(getClass().getResource("/FXML/Login.fxml"));

} catch (IOException ex) {

System.out.println(ex.getMessage());

}

Scene scene = new Scene(root);

stage.setScene(scene);

stage.initStyle(StageStyle.TRANSPARENT);

stage.setResizable(false);

stage.show();

ap.getScene().getWindow().hide();

root.setOnMousePressed(new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event){

xoffset = event.getSceneX();

yoffset = event.getSceneY();

}

});

root.setOnMouseDragged(new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event){

stage.setX(event.getScreenX() - xoffset);

stage.setY(event.getScreenY() - yoffset);

}

});

});

} catch (InterruptedException ex) {

System.out.println(ex.getMessage());

}

}

}

@Override

public void initialize(URL url, ResourceBundle rb) {

new ShowSplashScreen().start();

}

}

package Controllers;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

public class LoginController implements Initializable {

private static final String DB\_URL = "jdbc:mysql://localhost:3306/hostel\_management";

private static final String DB\_USER = "root";

private static final String DB\_PASSWORD = "zainyaqoob";

@FXML

private Button btn\_signIn;

@FXML

private Button btn\_signUp;

@FXML

private TextField txt\_username;

@FXML

private PasswordField txt\_password;

@FXML

private CheckBox pass\_toggle;

@FXML

private TextField txt\_pword;

@FXML

private Button btnClose;

@Override

public void initialize(URL url, ResourceBundle rb) {

this.togglevisiblePassword(null);

}

@FXML

private void signInButtonAction(MouseEvent event) {

String username = txt\_username.getText();

String password = txt\_password.getText();

if (username.isEmpty() || password.isEmpty()) {

showAlert("Input Error", "Username and password fields cannot be empty", Alert.AlertType.WARNING);

return;

}

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

PreparedStatement stmt = conn.prepareStatement("SELECT \* FROM users WHERE username = ? AND password = ?")) {

stmt.setString(1, username);

stmt.setString(2, password);

ResultSet rs = stmt.executeQuery();

if (rs.next()) {

showAlert("Login Successful", "Welcome " + username, Alert.AlertType.INFORMATION);

try {

FXMLLoader loader = new FXMLLoader(getClass().getResource("/FXML/MenuComponent.fxml"));

Parent root = loader.load();

Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();

stage.setScene(new Scene(root));

stage.show();

} catch (IOException e) {

System.out.println(e.getMessage());

showAlert("Error", "An error occurred while loading the next screen", Alert.AlertType.ERROR);

}

} else {

showAlert("Login Failed", "Invalid username or password", Alert.AlertType.ERROR);

}

} catch (SQLException e) {

System.out.println(e.getMessage());

showAlert("Database Error", "An error occurred while connecting to the database", Alert.AlertType.ERROR);

}

}

@FXML

private void signUpButtonAction(MouseEvent event) {

String username = txt\_username.getText();

String password = txt\_password.getText();

if (username.isEmpty() || password.isEmpty()) {

showAlert("Input Error", "Username and password fields cannot be empty", Alert.AlertType.WARNING);

return;

}

try (Connection conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

PreparedStatement checkStmt = conn.prepareStatement("SELECT \* FROM users WHERE username = ?");

PreparedStatement insertStmt = conn.prepareStatement("INSERT INTO users (username, password) VALUES (?, ?)")) {

checkStmt.setString(1, username);

ResultSet rs = checkStmt.executeQuery();

if (rs.next()) {

showAlert("Signup Failed", "Username already exists", Alert.AlertType.ERROR);

} else {

insertStmt.setString(1, username);

insertStmt.setString(2, password);

insertStmt.executeUpdate();

showAlert("Signup Successful", "User created successfully", Alert.AlertType.INFORMATION);

}

} catch (SQLException e) {

System.out.println(e.getMessage());

showAlert("Database Error", "An error occurred while connecting to the database", Alert.AlertType.ERROR);

}

}

@FXML

private void togglevisiblePassword(ActionEvent event) {

if (pass\_toggle.isSelected()) {

txt\_pword.setText(txt\_password.getText());

txt\_password.setVisible(false);

txt\_pword.setVisible(true);

} else {

txt\_password.setText(txt\_pword.getText());

txt\_password.setVisible(true);

txt\_pword.setVisible(false);

}

}

@FXML

public void closeLogin(ActionEvent event) {

Stage stage = (Stage) btnClose.getScene().getWindow();

stage.close();

}

private void showAlert(String title, String content, Alert.AlertType alertType) {

Alert alert = new Alert(alertType);

alert.setTitle(title);

alert.setHeaderText(null);

alert.setContentText(content);

alert.showAndWait();

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package MainApp;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.layout.StackPane;

import javafx.stage.Stage;

import javafx.scene.input.MouseEvent;

import java.io.File;

import java.util.Objects;

import javafx.scene.paint.Color;

import javafx.stage.StageStyle;

public class Main extends Application {

double xoffset, yoffset;

@Override

public void start(Stage primaryStage) {

try {

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Loading.fxml"));

Scene scene = new Scene(root);

primaryStage.initStyle(StageStyle.TRANSPARENT);

primaryStage.setScene(scene);

//scene.setFill(Color.TRANSPARENT);

primaryStage.show();

primaryStage.setResizable(false);

root.setOnMousePressed(new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event){

xoffset = event.getSceneX();

yoffset = event.getSceneY();

}

});

root.setOnMouseDragged(new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event){

primaryStage.setX(event.getScreenX() - xoffset);

primaryStage.setY(event.getScreenY() - yoffset);

}

});

} catch (Exception ex) {

System.out.println(ex.getMessage());

}

}

public static void main(String[] args) {

launch(args);

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.EventHandler;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

public class MenuComponentController {

double xoffset, yoffset;

@FXML

private Button btn\_student;

@FXML

private Button btn\_employee;

@FXML

private Button btn\_back;

@FXML

private void student\_btn\_clicked(MouseEvent event) throws IOException {

btn\_student.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Student/Student\_Menu.fxml"));

Scene scene = new Scene(root);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.setScene(scene);

stu\_Menu.show();

stu\_Menu.setFullScreen(false);

stu\_Menu.setResizable(false);

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Login.fxml"));

Scene scene = new Scene(root);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.setScene(scene);

stu\_Menu.show();

stu\_Menu.setResizable(false);

}

@FXML

private void employee\_btn\_clicked(MouseEvent event) throws IOException {

btn\_employee.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Employee/Employee\_Menu.fxml"));

Scene scene = new Scene(root);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.setScene(scene);

stu\_Menu.show();

stu\_Menu.setResizable(false);

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers.Employee;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ResourceBundle;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Button;

import javafx.scene.control.TextField;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

import javax.swing.JOptionPane;

import DBConnection.DBHandler;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javafx.stage.StageStyle;

public class New\_EmployeeController implements Initializable {

@FXML

private Button btn\_back;

@FXML

private TextField reg\_txt\_emp\_username;

@FXML

private TextField reg\_txt\_emp\_phnmb;

@FXML

private TextField reg\_txt\_emp\_nic;

@FXML

private Button btn\_reg\_employee;

@FXML

private TextField reg\_txt\_emp\_emgtel;

private Connection connection;

private DBHandler handler;

private PreparedStatement pst;

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

handler = new DBHandler();

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Employee/Employee\_Menu.fxml"));

Scene scene = new Scene(root);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.setScene(scene);

stu\_Menu.show();

stu\_Menu.setResizable(false);

}

@FXML

private void registerButtonAction(MouseEvent event) {

String name = reg\_txt\_emp\_username.getText();

String nic = reg\_txt\_emp\_nic.getText();

String tel = reg\_txt\_emp\_emgtel.getText();

String emgTel = reg\_txt\_emp\_phnmb.getText();

if (name.equals("")

|| nic.equals("")

|| tel.equals("")

|| emgTel.equals("")) {

setTExtRefresh();

} else {

String insert = "INSERT INTO register\_Employee(name,nic,tel,emg\_tel)" + "VALUES(?,?,?,?)";

connection = handler.connectDB();

try {

pst = connection.prepareStatement(insert);

} catch (SQLException ex) {

ex.printStackTrace();

}

try {

pst.setString(1, reg\_txt\_emp\_username.getText());

pst.setString(2, reg\_txt\_emp\_nic.getText());

pst.setString(3, reg\_txt\_emp\_emgtel.getText());

pst.setString(4, reg\_txt\_emp\_phnmb.getText());

pst.executeUpdate();

showAlert("Message", "Registered", Alert.AlertType.INFORMATION);

setTExtRefresh();

} catch (SQLException ex) {

Logger.getLogger(New\_EmployeeController.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, ex.getMessage());

}

}

}

private void setTExtRefresh() {

reg\_txt\_emp\_username.setText("");

reg\_txt\_emp\_nic.setText("");

reg\_txt\_emp\_emgtel.setText("");

reg\_txt\_emp\_phnmb.setText("");

}

private void showAlert(String title, String content, Alert.AlertType alertType) {

Alert alert = new Alert(alertType);

alert.setTitle(title);

alert.setHeaderText(null);

alert.setContentText(content);

alert.showAndWait();

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers.Student;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Alert;

import javafx.scene.control.Button;

import javafx.scene.control.TextField;

import javafx.scene.input.MouseEvent;

import DBConnection.DBHandler;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

public class New\_StudentController implements Initializable {

@FXML

private TextField reg\_txt\_username;

@FXML

private TextField reg\_txt\_nsbmid;

@FXML

private TextField reg\_txt\_email;

@FXML

private TextField reg\_txt\_phnmb;

@FXML

private TextField reg\_txt\_nic;

@FXML

private TextField reg\_txt\_address;

@FXML

private TextField reg\_txt\_guardname;

@FXML

private TextField reg\_txt\_guardtel;

private Connection connection;

private DBHandler handler;

private PreparedStatement pst;

@FXML

private Button btn\_back;

@FXML

private Button btn\_reg\_student;

@Override

public void initialize(URL url, ResourceBundle rb) {

handler = new DBHandler();

}

@FXML

private void registerButtonAction(MouseEvent event) {

String userName = reg\_txt\_username.getText();

String nsbmID = reg\_txt\_nsbmid.getText();

String email = reg\_txt\_email.getText();

String phoneNumber = reg\_txt\_phnmb.getText();

String nic = reg\_txt\_nic.getText();

String address = reg\_txt\_address.getText();

String guardName = reg\_txt\_guardname.getText();

String guardTel = reg\_txt\_guardtel.getText();

if (userName.isEmpty() || nsbmID.isEmpty() || email.isEmpty() || phoneNumber.isEmpty() ||

nic.isEmpty() || address.isEmpty() || guardName.isEmpty() || guardTel.isEmpty()) {

showAlert("Validation Error", "All fields are required!", Alert.AlertType.ERROR);

return;

}

if (!nsbmID.matches("\\d+")) {

showAlert("Validation Error", "NSBM ID must be a number.", Alert.AlertType.ERROR);

return;

}

if (!email.matches("^[a-zA-Z0-9.\_%+-]+@gmail\\.com$")) {

showAlert("Validation Error", "Email must be a valid Gmail address.", Alert.AlertType.ERROR);

return;

}

if (!phoneNumber.matches("\\+92\\d{10}")) {

showAlert("Validation Error", "Phone number must begin with +92.", Alert.AlertType.ERROR);

return;

}

if (!guardTel.matches("\\+92\\d{10}")) {

showAlert("Validation Error", "Guardian's contact must must begin with +92", Alert.AlertType.ERROR);

return;

}

if (!nic.matches("\\d{5}-\\d{7}-\\d")) {

showAlert("Validation Error", "NIC must be in the format 12345-1234567-1.", Alert.AlertType.ERROR);

return;

}

String checkQuery = "SELECT COUNT(\*) FROM register\_Students WHERE nsbmID = ? OR nic = ?";

String insertQuery = "INSERT INTO register\_Students(name, nsbmID, email, phoneNumber, nic, address, guardName, guardTel) VALUES(?, ?, ?, ?, ?, ?, ?, ?)";

try (Connection connection = handler.connectDB();

PreparedStatement checkStmt = connection.prepareStatement(checkQuery);

PreparedStatement insertStmt = connection.prepareStatement(insertQuery)) {

checkStmt.setString(1, nsbmID);

checkStmt.setString(2, nic);

try (java.sql.ResultSet resultSet = checkStmt.executeQuery()) {

if (resultSet.next() && resultSet.getInt(1) > 0) {

showAlert("Validation Error", "Roll No or CNIC already exists.", Alert.AlertType.ERROR);

return;

}

}

insertStmt.setString(1, userName);

insertStmt.setString(2, nsbmID);

insertStmt.setString(3, email);

insertStmt.setString(4, phoneNumber);

insertStmt.setString(5, nic);

insertStmt.setString(6, address);

insertStmt.setString(7, guardName);

insertStmt.setString(8, guardTel);

insertStmt.executeUpdate();

showAlert("Success", "Student registered successfully!", Alert.AlertType.INFORMATION);

setTExtRefresh();

} catch (SQLException ex) {

System.out.println(ex.getMessage());

showAlert("Database Error", "Failed to register student. Please try again.", Alert.AlertType.ERROR);

}

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Student/Student\_Menu.fxml"));

Scene scene = new Scene(root);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.setScene(scene);

stu\_Menu.show();

stu\_Menu.setResizable(false);

}

@FXML

private void setTExtRefresh() {

reg\_txt\_username.setText("");

reg\_txt\_nsbmid.setText("");

reg\_txt\_email.setText("");

reg\_txt\_phnmb.setText("");

reg\_txt\_nic.setText("");

reg\_txt\_address.setText("");

reg\_txt\_guardname.setText("");

reg\_txt\_guardtel.setText("");

}

private void showAlert(String title, String content, Alert.AlertType alertType) {

Alert alert = new Alert(alertType);

alert.setTitle(title);

alert.setHeaderText(null);

alert.setContentText(content);

alert.showAndWait();

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Model;

import javafx.beans.property.SimpleStringProperty;

import javafx.beans.property.StringProperty;

public class StudentDetails {

// model class holding getters, setters and properties

private StringProperty id;

private StringProperty name;

private StringProperty nsbmid;

private StringProperty email;

private StringProperty phoneNumber;

private StringProperty nic;

private StringProperty address;

private StringProperty guardName;

private StringProperty guardTel;

public StudentDetails(String id,String name, String nsbmid, String email, String phoneNumber, String nic, String address, String guardName, String guardTel) {

this.id = new SimpleStringProperty(id);

this.name = new SimpleStringProperty(name);

this.nsbmid = new SimpleStringProperty(nsbmid);

this.email = new SimpleStringProperty(email);

this.phoneNumber = new SimpleStringProperty(phoneNumber);

this.nic = new SimpleStringProperty(nic);

this.address = new SimpleStringProperty(address);

this.guardName = new SimpleStringProperty(guardName);

this.guardTel = new SimpleStringProperty(guardTel);

}

public String getId() {

return id.get();

}

public String getName() {

return name.get();

}

public String getNsbmId() {

return nsbmid.get();

}

public String getEmail() {

return email.get();

}

public String getPhoneNumber() {

return phoneNumber.get();

}

public String getNIC() {

return nic.get();

}

public String getAddress() {

return address.get();

}

public String getGuardName() {

return guardName.get();

}

public String getGuardTel() {

return guardTel.get();

}

public void setId(String value) {

id.set(value);

}

public void setName(String value) {

name.set(value);

}

public void setNsbmid(String value) {

nsbmid.set(value);

}

public void setEmail(String value) {

email.set(value);

}

public void setPhoneNumber(String value) {

phoneNumber.set(value);

}

public void setNic(String value) {

nic.set(value);

}

public void setAddress(String value) {

address.set(value);

}

public void setGuardName(String value) {

guardName.set(value);

}

public void setGuardTel(String value) {

guardTel.set(value);

}

// Propert values

public StringProperty idProperty() { return id; }

public StringProperty nameProperty() { return name; }

public StringProperty nsbmIdProperty() { return nsbmid; }

public StringProperty emailProperty() { return email; }

public StringProperty phoneNumberProperty() { return phoneNumber; }

public StringProperty nicProperty() { return nic; }

public StringProperty addressProperty() { return address; }

public StringProperty guardNameProperty() { return guardName; }

public StringProperty guardTelProperty() { return guardTel; }

}

package Controllers.Student;

import java.awt.\*;

import java.io.IOException;

import java.io.UnsupportedEncodingException;

import java.net.URI;

import java.net.URL;

import java.net.URLEncoder;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ResourceBundle;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Button;

import javafx.scene.control.TextField;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

import javax.swing.JOptionPane;

import DBConnection.DBHandler;

import java.util.logging.Level;

import java.util.logging.Logger;

public class Student\_feeController implements Initializable {

@FXML

private Button btn\_back;

@FXML

private TextField studentID;

@FXML

private TextField studentFee;

@FXML

private Button submit;

@FXML

private ComboBox<String> month;

@FXML

private ComboBox<String> year;

private Connection connection;

private DBHandler handler;

private PreparedStatement pst;

@Override

public void initialize(URL url, ResourceBundle rb) {

handler = new DBHandler();

ObservableList<String> months = FXCollections.observableArrayList("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December");

month.setItems(months);

ObservableList<String> years = FXCollections.observableArrayList("2020", "2021", "2022", "2023", "2024", "2025", "2026", "2027", "2028", "2029", "2030");

year.setItems(years);

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Student/Student\_Menu.fxml"));

Scene scene = new Scene(root);

stu\_Menu.setScene(scene);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.show();

stu\_Menu.setResizable(false);

}

@FXML

private void submitButtonAction(MouseEvent event) {

String id = studentID.getText();

String fee = studentFee.getText();

String selectedMonth = month.getValue();

String selectedYear = year.getValue();

if (id == null || id.trim().isEmpty() || fee == null || fee.trim().isEmpty() || selectedMonth == null || selectedYear == null) {

showAlert("Validation Error", "All Fields Are Required!", Alert.AlertType.ERROR);

return;

}

try {

Double.parseDouble(fee);

} catch (NumberFormatException e) {

showAlert("Warning", "Fee must be a numeric value", Alert.AlertType.WARNING);

return;

}

connection = handler.connectDB();

String checkStudentQuery = "SELECT email FROM register\_students WHERE nsbmId = ?";

String studentEmail = null;

try {

pst = connection.prepareStatement(checkStudentQuery);

pst.setString(1, id);

ResultSet rs = pst.executeQuery();

if (!rs.next()) {

showAlert("Error Code 404", "Student ID doesn't exist", Alert.AlertType.ERROR);

return;

}

studentEmail = rs.getString("email");

} catch (SQLException ex) {

System.out.println(ex.getMessage());

showAlert("Database Error", ex.getMessage(), Alert.AlertType.ERROR);

return;

}

String insertQuery = "INSERT INTO student\_fee(studentid, year, fee, month) VALUES(?, ?, ?, ?)";

try {

pst = connection.prepareStatement(insertQuery);

pst.setString(1, id);

pst.setString(2, selectedYear);

pst.setString(3, fee);

pst.setString(4, selectedMonth);

pst.executeUpdate();

showAlert("Message", "Fee Record Added Successfully", Alert.AlertType.INFORMATION);

sendEmail(studentEmail, selectedMonth, selectedYear, fee);

setTExtRefresh();

} catch (SQLException ex) {

Logger.getLogger(Student\_feeController.class.getName()).log(Level.SEVERE, null, ex);

showAlert("Error", "Error Saving Fee Record", Alert.AlertType.ERROR);

} finally {

try {

if (pst != null) pst.close();

if (connection != null) connection.close();

} catch (SQLException ex) {

System.out.println(ex.getMessage());

showAlert("Database Error", "An error occurred while connecting to the database", Alert.AlertType.ERROR);

}

}

}

private void sendEmail(String studentEmail, String month, String year, String fee) {

try {

String subject = "Fee Payment Pending for " + month + " " + year;

String body = "Dear Student,\n\nYour fee of " + fee + " for the month of " + month + " " + year + " is still pending.\n\nPlease make the payment at the earliest.\n\nBest regards,\nYour p2University";

String encodedSubject = URLEncoder.encode(subject, "UTF-8");

String encodedBody = URLEncoder.encode(body, "UTF-8");

String mailto = "https://mail.google.com/mail/?view=cm&fs=1&to=" + studentEmail + "&su=" + encodedSubject + "&body=" + encodedBody;

if (Desktop.isDesktopSupported()) {

URI mailtoURI = new URI(mailto);

Desktop.getDesktop().browse(mailtoURI);

} else {

showAlert("Error", "Desktop email client not supported", Alert.AlertType.ERROR);

}

} catch (UnsupportedEncodingException e) {

System.out.println(e.getMessage());

showAlert("Encoding Error", "Failed to encode the email content", Alert.AlertType.ERROR);

} catch (java.net.URISyntaxException e) {

System.out.println(e.getMessage());

showAlert("URI Error", "Invalid URI format for mailto link", Alert.AlertType.ERROR);

} catch (IOException e) {

System.out.println(e.getMessage());

showAlert("Error", "Failed to open web browser", Alert.AlertType.ERROR);

}

}

private void setTExtRefresh() {

studentID.clear();

studentFee.clear();

month.setValue(null);

year.setValue(null);

}

private void showAlert(String title, String content, Alert.AlertType alertType) {

Alert alert = new Alert(alertType);

alert.setTitle(title);

alert.setHeaderText(null);

alert.setContentText(content);

alert.showAndWait();

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers.Student;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.EventHandler;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

public class Student\_MenuController implements Initializable {

double xoffset, yoffset;

@FXML

private Button addNewStudent;

@FXML

private Button updateStudent;

@FXML

private Button deleteStudent;

@FXML

private Button allStudentLiving;

@FXML

private Button leavedStudent;

@FXML

private Button btn\_back;

@FXML

private Button studentFees;

@Override

public void initialize(URL url, ResourceBundle rb) {

}

@FXML

private void addNewStudentAction(MouseEvent event) throws IOException {

addNewStudent.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Student/New\_Student.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

}

@FXML

private void updateStudentAction(MouseEvent event) throws IOException {

updateStudent.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Student/Update\_Student.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

}

@FXML

private void deleteStudentAction(MouseEvent event) throws IOException {

deleteStudent.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Student/Delete\_Student.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

}

@FXML

private void studentFeeAction(MouseEvent event) throws IOException {

deleteStudent.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Student/Student\_fee.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

}

@FXML

private void allStudentAction(MouseEvent event) throws IOException {

allStudentLiving.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Student/All\_Student\_Living.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

}

@FXML

private void leavedStudentAction(MouseEvent event) throws IOException {

leavedStudent.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Student/All\_Student\_Leaved.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage login = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/MenuComponent.fxml"));

Scene scene = new Scene(root);

login.initStyle(StageStyle.TRANSPARENT);

login.setScene(scene);

login.show();

login.setResizable(false);

root.setOnMousePressed(new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event){

xoffset = event.getSceneX();

yoffset = event.getSceneY();

}

});

root.setOnMouseDragged(new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event){

login.setX(event.getScreenX() - xoffset);

login.setY(event.getScreenY() - yoffset);

}

});

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers.Employee;

import Controllers.Student.New\_StudentController;

import Model.EmployeeDetails;

import java.net.URL;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ResourceBundle;

import javafx.collections.ObservableList;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.\*;

import javafx.scene.input.MouseEvent;

import DBConnection.DBHandler;

import Model.StudentDetails;

import java.io.IOException;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javafx.collections.FXCollections;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

import javax.swing.JOptionPane;

public class Update\_EmployeeController implements Initializable {

@FXML

private Button btn\_refersh;

@FXML

private Button btn\_back;

@FXML

private TableView<EmployeeDetails> tableEmployee;

@FXML

private TableColumn<EmployeeDetails, String> col\_id;

@FXML

private TableColumn<EmployeeDetails, String> col\_name;

@FXML

private TableColumn<EmployeeDetails, String> col\_phonenumber;

@FXML

private TableColumn<EmployeeDetails, String> col\_emgtel;

private Connection connection;

private DBHandler handler;

private PreparedStatement pst;

private ObservableList<EmployeeDetails> data;

@FXML

private Button btn\_update\_employee;

@FXML

private TextField emp\_id;

@FXML

private TextField reg\_txt\_emp\_emgtel;

@FXML

private TextField reg\_txt\_emp\_username;

@FXML

private TextField reg\_txt\_emp\_nic;

@FXML

private TextField reg\_txt\_emp\_phnmb;

@FXML

private TableColumn<EmployeeDetails, String> col\_nic;

@Override

public void initialize(URL url, ResourceBundle rb) {

handler = new DBHandler();

}

@FXML

private void updateEmployeeButtonAction(MouseEvent event) {

String id = emp\_id.getText();

String name = reg\_txt\_emp\_username.getText();

String nic = reg\_txt\_emp\_nic.getText();

String tel = reg\_txt\_emp\_phnmb.getText();

String emg\_tel = reg\_txt\_emp\_emgtel.getText();

if (name.equals("")

|| id.equals("")

|| nic.equals("")

|| emg\_tel.equals("")

|| tel.equals("")) {

} else {

String update = "UPDATE register\_Employee set id = ?, name = ?, nic = ?,tel = ?,emg\_tel = ? where id = '" + id + "' ";

connection = handler.connectDB();

try {

pst = connection.prepareStatement(update);

} catch (SQLException ex) {

ex.printStackTrace();

}

try {

pst.setString(1, emp\_id.getText());

pst.setString(2, reg\_txt\_emp\_username.getText());

pst.setString(3, reg\_txt\_emp\_nic.getText());

pst.setString(4, reg\_txt\_emp\_phnmb.getText());

pst.setString(5, reg\_txt\_emp\_emgtel.getText());

pst.executeUpdate();

showAlert("Message", "Student, Updated", Alert.AlertType.INFORMATION);

autoRefresh();

} catch (SQLException ex) {

Logger.getLogger(New\_StudentController.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, ex.getMessage());

}

}

}

@FXML

private void refreshButtionClickAction(MouseEvent event) {

connection = handler.connectDB();

data = FXCollections.observableArrayList();

try {

ResultSet rs = connection.createStatement().executeQuery("SELECT \* FROM register\_employee");

while (rs.next()) {

data.add(new EmployeeDetails(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5)));

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

}

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

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

col\_nic.setCellValueFactory(new PropertyValueFactory<>("nic"));

col\_phonenumber.setCellValueFactory(new PropertyValueFactory<>("tel"));

col\_emgtel.setCellValueFactory(new PropertyValueFactory<>("emgTel"));

tableEmployee.setItems(null);

tableEmployee.setItems(data);

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Employee/Employee\_Menu.fxml"));

Scene scene = new Scene(root);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.setScene(scene);

stu\_Menu.show();

stu\_Menu.setResizable(false);

}

@FXML

private void displaySelectedAction(MouseEvent event) {

EmployeeDetails Employee = tableEmployee.getSelectionModel().getSelectedItem();

if (Employee == null) {

showAlert("Warning", "Nothing Selected", Alert.AlertType.WARNING);

} else {

String id = Employee.getId();

String name = Employee.getName();

String nic = Employee.getNic();

String tel = Employee.getTel();

String emg\_tel = Employee.getEmgTel();

emp\_id.setText(id);

reg\_txt\_emp\_username.setText(name);

reg\_txt\_emp\_nic.setText(nic);

reg\_txt\_emp\_phnmb.setText(tel);

reg\_txt\_emp\_emgtel.setText(emg\_tel);

}

}

private void autoRefresh() {

connection = handler.connectDB();

data = FXCollections.observableArrayList();

try {

ResultSet rs = connection.createStatement().executeQuery("SELECT \* FROM register\_employee");

while (rs.next()) {

data.add(new EmployeeDetails(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5)));

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

}

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

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

col\_nic.setCellValueFactory(new PropertyValueFactory<>("nic"));

col\_phonenumber.setCellValueFactory(new PropertyValueFactory<>("tel"));

col\_emgtel.setCellValueFactory(new PropertyValueFactory<>("emgTel"));

tableEmployee.setItems(null);

tableEmployee.setItems(data);

}

private void showAlert(String title, String content, Alert.AlertType alertType) {

Alert alert = new Alert(alertType);

alert.setTitle(title);

alert.setHeaderText(null);

alert.setContentText(content);

alert.showAndWait();

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controllers.Student;

import Model.StudentDetails;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.collections.ObservableList;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.\*;

import javafx.scene.input.MouseEvent;

import DBConnection.DBHandler;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javafx.collections.FXCollections;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

import javax.swing.JOptionPane;

public class Update\_StudentController implements Initializable {

@FXML

private TextField reg\_txt\_username;

@FXML

private TextField reg\_txt\_nsbmid;

@FXML

private TextField reg\_txt\_email;

@FXML

private TextField reg\_txt\_phnmb;

@FXML

private TextField reg\_txt\_nic;

@FXML

private TextField reg\_txt\_address;

@FXML

private TextField reg\_txt\_guardname;

@FXML

private TextField reg\_txt\_guardtel;

@FXML

private Button btn\_update\_student;

@FXML

private Button btn\_refersh;

private ObservableList<StudentDetails> data;

@FXML

private TableView<StudentDetails> tableStudent;

@FXML

private TableColumn<StudentDetails, String> col\_name;

@FXML

private TableColumn<StudentDetails, String> col\_nsbmid;

@FXML

private TableColumn<StudentDetails, String> col\_email;

@FXML

private TableColumn<StudentDetails, String> col\_phonenumber;

@FXML

private TableColumn<StudentDetails, String> col\_nic;

@FXML

private TableColumn<StudentDetails, String> col\_address;

@FXML

private TableColumn<StudentDetails, String> col\_g\_name;

@FXML

private TableColumn<StudentDetails, String> col\_g\_tel;

private Connection connection;

private DBHandler handler;

private PreparedStatement pst;

@FXML

private TextField reg\_txt\_id;

@FXML

private TableColumn<StudentDetails, String> col\_id;

@FXML

private Button btn\_back;

@Override

public void initialize(URL url, ResourceBundle rb) {

handler = new DBHandler();

}

private void autoRefresh() {

connection = handler.connectDB();

data = FXCollections.observableArrayList();

try {

// Execure query

ResultSet rs = connection.createStatement().executeQuery("SELECT \* FROM register\_students");

while (rs.next()) {

// get string from db

data.add(new StudentDetails(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(6), rs.getString(7), rs.getString(8), rs.getString(9)));

}

} catch (SQLException ex) {

System.out.println(ex.getMessage());

showAlert("Database Error", "An error occurred while connecting to the database", Alert.AlertType.ERROR);

}

// set cell values

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

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

col\_nsbmid.setCellValueFactory(new PropertyValueFactory<>("nsbmId"));

col\_email.setCellValueFactory(new PropertyValueFactory<>("email"));

col\_phonenumber.setCellValueFactory(new PropertyValueFactory<>("phoneNumber"));

col\_nic.setCellValueFactory(new PropertyValueFactory<>("nic"));

col\_address.setCellValueFactory(new PropertyValueFactory<>("address"));

col\_g\_name.setCellValueFactory(new PropertyValueFactory<>("guardName"));

col\_g\_tel.setCellValueFactory(new PropertyValueFactory<>("guardTel"));

tableStudent.setItems(null);

tableStudent.setItems(data);

}

private boolean isNsbmIdExists(String nsbmID, String currentId) {

String query = "SELECT id FROM register\_students WHERE nsbmID = ? AND id != ?";

try {

pst = connection.prepareStatement(query);

pst.setString(1, nsbmID);

pst.setString(2, currentId);

ResultSet rs = pst.executeQuery();

return rs.next(); // If a record exists, nsbmID is already in use

} catch (SQLException e) {

showAlert("Database Error", "An error occurred while checking NSBM ID.", Alert.AlertType.ERROR);

System.out.println(e.getMessage());

}

return false;

}

private boolean isNicExists(String nic, String currentId) {

String query = "SELECT id FROM register\_students WHERE nic = ? AND id != ?";

try {

pst = connection.prepareStatement(query);

pst.setString(1, nic);

pst.setString(2, currentId);

ResultSet rs = pst.executeQuery();

return rs.next(); // If a record exists, NIC is already in use

} catch (SQLException e) {

showAlert("Database Error", "An error occurred while checking NIC.", Alert.AlertType.ERROR);

System.out.println(e.getMessage());

}

return false;

}

@FXML

private void updateStudentButtonAction(MouseEvent event) {

String id = reg\_txt\_id.getText();

String userName = reg\_txt\_username.getText();

String nsbmID = reg\_txt\_nsbmid.getText();

String email = reg\_txt\_email.getText();

String phoneNumber = reg\_txt\_phnmb.getText();

String nic = reg\_txt\_nic.getText();

String address = reg\_txt\_address.getText();

String guardName = reg\_txt\_guardname.getText();

String guardTel = reg\_txt\_guardtel.getText();

// Validate inputs

if (userName.isEmpty() || nsbmID.isEmpty() || email.isEmpty() || phoneNumber.isEmpty() ||

nic.isEmpty() || address.isEmpty() || guardName.isEmpty() || guardTel.isEmpty()) {

showAlert("Validation Error", "All fields are required!", Alert.AlertType.ERROR);

return;

}

if (!nsbmID.matches("\\d+")) {

showAlert("Validation Error", "NSBM ID must be numeric.", Alert.AlertType.ERROR);

return;

}

if (!email.endsWith("@gmail.com")) {

showAlert("Validation Error", "Email must end with '@gmail.com'.", Alert.AlertType.ERROR);

return;

}

if (!phoneNumber.matches("\\+92\\d{10}")) {

showAlert("Validation Error", "Phone number must start with +92 and contain exactly 10 digits.", Alert.AlertType.ERROR);

return;

}

if (!guardTel.matches("\\+92\\d{10}")) {

showAlert("Validation Error", "Guardian's contact must start with +92 and contain exactly 10 digits.", Alert.AlertType.ERROR);

return;

}

if (!nic.matches("\\d{5}-\\d{7}-\\d")) {

showAlert("Validation Error", "CNIC must be in the format 12345-1234567-1.", Alert.AlertType.ERROR);

return;

}

if (isNsbmIdExists(nsbmID, id)) {

showAlert("Duplicate Error", "The Roll No already exists.", Alert.AlertType.ERROR);

return;

}

if (isNicExists(nic, id)) {

showAlert("Duplicate Error", "The CNIC already exists.", Alert.AlertType.ERROR);

return;

}

String updateQuery = "UPDATE register\_students SET name = ?, nsbmID = ?, email = ?, phoneNumber = ?, nic = ?, address = ?, guardName = ?, guardTel = ? WHERE id = ?";

connection = handler.connectDB();

try {

pst = connection.prepareStatement(updateQuery);

pst.setString(1, userName);

pst.setString(2, nsbmID);

pst.setString(3, email);

pst.setString(4, phoneNumber);

pst.setString(5, nic);

pst.setString(6, address);

pst.setString(7, guardName);

pst.setString(8, guardTel);

pst.setString(9, id);

int rowsUpdated = pst.executeUpdate();

if (rowsUpdated > 0) {

showAlert("Success", "Student record updated successfully!", Alert.AlertType.INFORMATION);

autoRefresh();

} else {

showAlert("Update Failed", "No record found with the given ID.", Alert.AlertType.WARNING);

}

} catch (SQLException e) {

showAlert("Database Error", "An error occurred while updating the student record.", Alert.AlertType.ERROR);

System.out.println(e.getMessage());

} finally {

try {

if (pst != null) pst.close();

if (connection != null) connection.close();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

}

}

@FXML

private void refreshButtionClickAction(MouseEvent event) {

connection = handler.connectDB();

data = FXCollections.observableArrayList();

try {

ResultSet rs = connection.createStatement().executeQuery("SELECT \* FROM register\_students");

while (rs.next()) {

data.add(new StudentDetails(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(6), rs.getString(7), rs.getString(8), rs.getString(9)));

}

} catch (SQLException ex) {

System.out.println(ex.getMessage());

}

// set cell values

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

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

col\_nsbmid.setCellValueFactory(new PropertyValueFactory<>("nsbmId"));

col\_email.setCellValueFactory(new PropertyValueFactory<>("email"));

col\_phonenumber.setCellValueFactory(new PropertyValueFactory<>("phoneNumber"));

col\_nic.setCellValueFactory(new PropertyValueFactory<>("nic"));

col\_address.setCellValueFactory(new PropertyValueFactory<>("address"));

col\_g\_name.setCellValueFactory(new PropertyValueFactory<>("guardName"));

col\_g\_tel.setCellValueFactory(new PropertyValueFactory<>("guardTel"));

tableStudent.setItems(null);

tableStudent.setItems(data);

}

@FXML

private void displaySelectedAction(MouseEvent event) {

StudentDetails student = tableStudent.getSelectionModel().getSelectedItem();

if (student == null) {

showAlert("Warning", "Nothing Selected", Alert.AlertType.WARNING);

} else {

String id = student.getId();

String name = student.getName();

String nsbmid = student.getNsbmId();

String email = student.getEmail();

String phonenumber = student.getPhoneNumber();

String nic = student.getNIC();

String address = student.getAddress();

String g\_name = student.getGuardName();

String g\_tel = student.getGuardTel();

reg\_txt\_id.setText(id);

reg\_txt\_username.setText(name);

reg\_txt\_nsbmid.setText(nsbmid);

reg\_txt\_email.setText(email);

reg\_txt\_phnmb.setText(phonenumber);

reg\_txt\_nic.setText(nic);

reg\_txt\_address.setText(address);

reg\_txt\_guardname.setText(g\_name);

reg\_txt\_guardtel.setText(g\_tel);

}

}

@FXML

private void back\_btn\_clicked(MouseEvent event) throws IOException {

btn\_back.getScene().getWindow().hide();

Stage stu\_Menu = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("/FXML/Student/Student\_Menu.fxml"));

Scene scene = new Scene(root);

stu\_Menu.setScene(scene);

stu\_Menu.initStyle(StageStyle.TRANSPARENT);

stu\_Menu.show();

stu\_Menu.setResizable(false);

}

private void showAlert(String title, String content, Alert.AlertType alertType) {

Alert alert = new Alert(alertType);

alert.setTitle(title);

alert.setHeaderText(null);

alert.setContentText(content);

alert.showAndWait();

}

}





