

**COMSATS University Islamabad, Lahore Campus**

**Subject:**

Object Oriented Programming.

**Submitted to**:

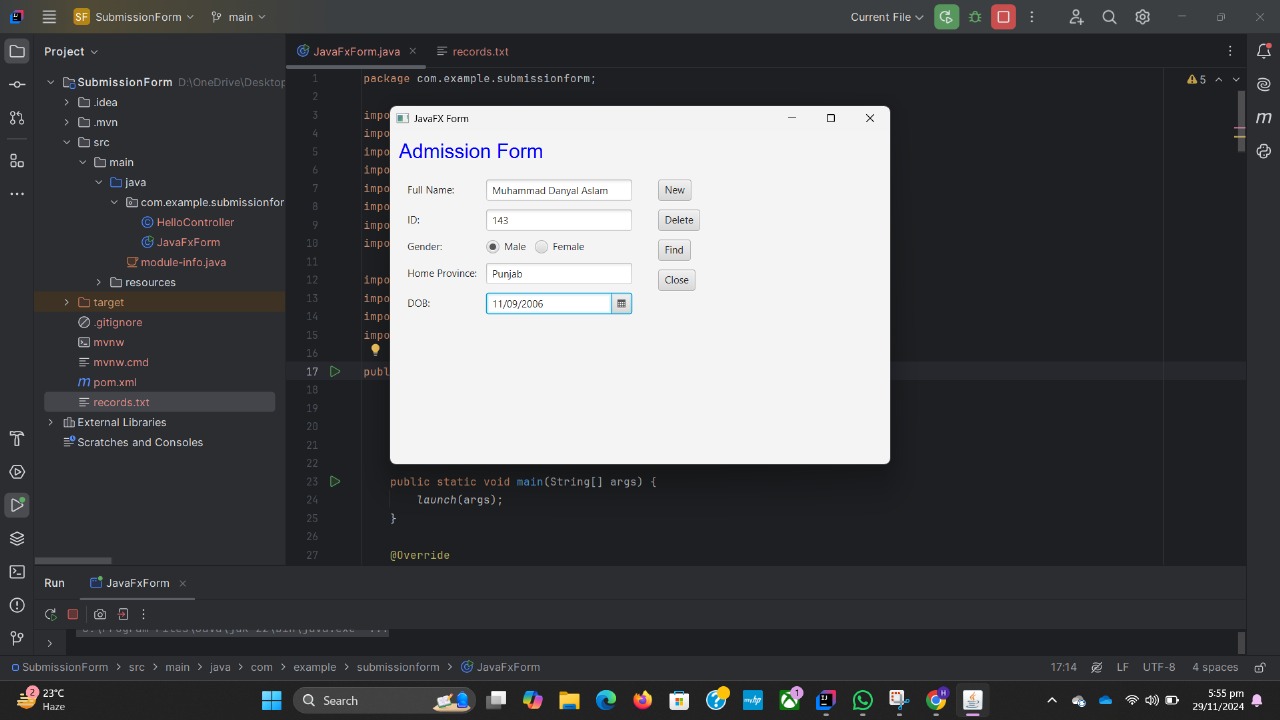
Sir Shahid Bhatti.

**Submitted by**:

Muhammad Danyal Aslam (SP24-BSE-143)

**Section**:

‘A’



Enter the information and click on the new button to write the data into records.txt file.

A screenshot of a computer

Description automatically generated

If user wants to find the information of any person, write the id of the person If it would be available Screen will pop up with information **if not found message with not found will pop**

A screenshot of a computer

Description automatically generated

The data on the form will be written in records.txt file.

**Source Code:**

package com.example.submissionform;

import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.layout.\*;

import javafx.scene.paint.Color;

import javafx.scene.text.Font;

import javafx.stage.Stage;

import java.io.\*;

import java.nio.file.Files;

import java.nio.file.Paths;

import java.util.List;

public class JavaFxForm extends Application {

private TextField fullNameField, idField, homeProvinceField;

private ToggleGroup genderGroup;

private DatePicker dobPicker;

private File file = new File("records.txt");

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage stage) {

// Create heading

Label heading = new Label("Admission Form");

heading.setFont(new Font("Arial", 24));

heading.setTextFill(Color.BLUE);

// Creating input fields

fullNameField = new TextField();

idField = new TextField();

homeProvinceField = new TextField();

dobPicker = new DatePicker();

// Gender radio buttons

RadioButton maleButton = new RadioButton("Male");

RadioButton femaleButton = new RadioButton("Female");

genderGroup = new ToggleGroup();

maleButton.setToggleGroup(genderGroup);

femaleButton.setToggleGroup(genderGroup);

// Creating buttons

Button newButton = new Button("New");

Button deleteButton = new Button("Delete");

Button findButton = new Button("Find");

Button closeButton = new Button("Close");

// Layout setup

GridPane grid = new GridPane();

grid.setPadding(new Insets(10));

grid.setHgap(10);

grid.setVgap(10);

grid.add(new Label("Full Name:"), 0, 0);

grid.add(fullNameField, 1, 0);

grid.add(new Label("ID:"), 0, 1);

grid.add(idField, 1, 1);

grid.add(new Label("Gender:"), 0, 2);

HBox genderBox = new HBox(10, maleButton, femaleButton);

grid.add(genderBox, 1, 2);

grid.add(new Label("Home Province:"), 0, 3);

grid.add(homeProvinceField, 1, 3);

grid.add(new Label("DOB:"), 0, 4);

grid.add(dobPicker, 1, 4);

VBox buttonBox = new VBox(10, newButton, deleteButton, findButton, closeButton);

buttonBox.setPadding(new Insets(10));

VBox mainLayout = new VBox(10, heading, new HBox(10, grid, buttonBox));

mainLayout.setPadding(new Insets(10));

// Button actions

newButton.setOnAction(e -> saveAndClear());

deleteButton.setOnAction(e -> deleteRecord());

findButton.setOnAction(e -> findRecord(stage));

closeButton.setOnAction(e -> stage.close());

// Scene setup

Scene scene = new Scene(mainLayout, 600, 400);

stage.setScene(scene);

stage.setTitle("JavaFX Form");

stage.show();

}

private void saveAndClear() {

try (FileWriter writer = new FileWriter(file, true)) {

String gender = ((RadioButton) genderGroup.getSelectedToggle()).getText();

String record = String.format("%s,%s,%s,%s,%s",

fullNameField.getText(),

idField.getText(),

gender,

homeProvinceField.getText(),

dobPicker.getValue());

writer.write(record + System.lineSeparator());

clearFields();

} catch (IOException e) {

showAlert("Error", "Failed to save record.");

}

}

private void clearFields() {

fullNameField.clear();

idField.clear();

homeProvinceField.clear();

genderGroup.selectToggle(null);

dobPicker.setValue(null);

}

private void deleteRecord() {

String searchID = idField.getText();

boolean found = false;

try {

List<String> allRecords = Files.readAllLines(Paths.get(file.toURI()));

try (FileWriter writer = new FileWriter(file, false)) {

for (String record : allRecords) {

if (!record.contains("," + searchID + ",")) {

writer.write(record + System.lineSeparator());

} else {

found = true;

}

}

}

if (found) {

showAlert("Info", "Record deleted successfully.");

clearFields();

} else {

showAlert("Info", "Record not found.");

}

} catch (IOException e) {

showAlert("Error", "Failed to delete record.");

}

}

private void findRecord(Stage primaryStage) {

String searchID = idField.getText();

boolean found = false;

try {

List<String> allRecords = Files.readAllLines(Paths.get(file.toURI()));

for (String record : allRecords) {

String[] fields = record.split(",");

if (fields[1].equals(searchID)) { // Compare ID

found = true;

openRecordWindow(primaryStage, record);

break;

}

}

if (!found) {

showAlert("Info", "Record not found.");

}

} catch (IOException e) {

showAlert("Error", "Failed to read records.");

}

}

private void openRecordWindow(Stage primaryStage, String record) {

String[] fields = record.split(",");

Stage recordStage = new Stage();

recordStage.initOwner(primaryStage);

recordStage.setTitle("Record Details");

GridPane grid = new GridPane();

grid.setPadding(new Insets(10));

grid.setHgap(10);

grid.setVgap(10);

grid.add(new Label("Full Name:"), 0, 0);

grid.add(new Label(fields[0]), 1, 0);

grid.add(new Label("ID:"), 0, 1);

grid.add(new Label(fields[1]), 1, 1);

grid.add(new Label("Gender:"), 0, 2);

grid.add(new Label(fields[2]), 1, 2);

grid.add(new Label("Home Province:"), 0, 3);

grid.add(new Label(fields[3]), 1, 3);

grid.add(new Label("DOB:"), 0, 4);

grid.add(new Label(fields[4]), 1, 4);

Scene scene = new Scene(grid, 300, 200);

recordStage.setScene(scene);

recordStage.show();

}

private void showAlert(String title, String message) {

Alert alert = new Alert(Alert.AlertType.INFORMATION);

alert.setTitle(title);

alert.setContentText(message);

alert.showAndWait();

}

}