

**COMSATS University Islamabad, Lahore Campus**

**Subject:**

OOP.

**Submitted to**:

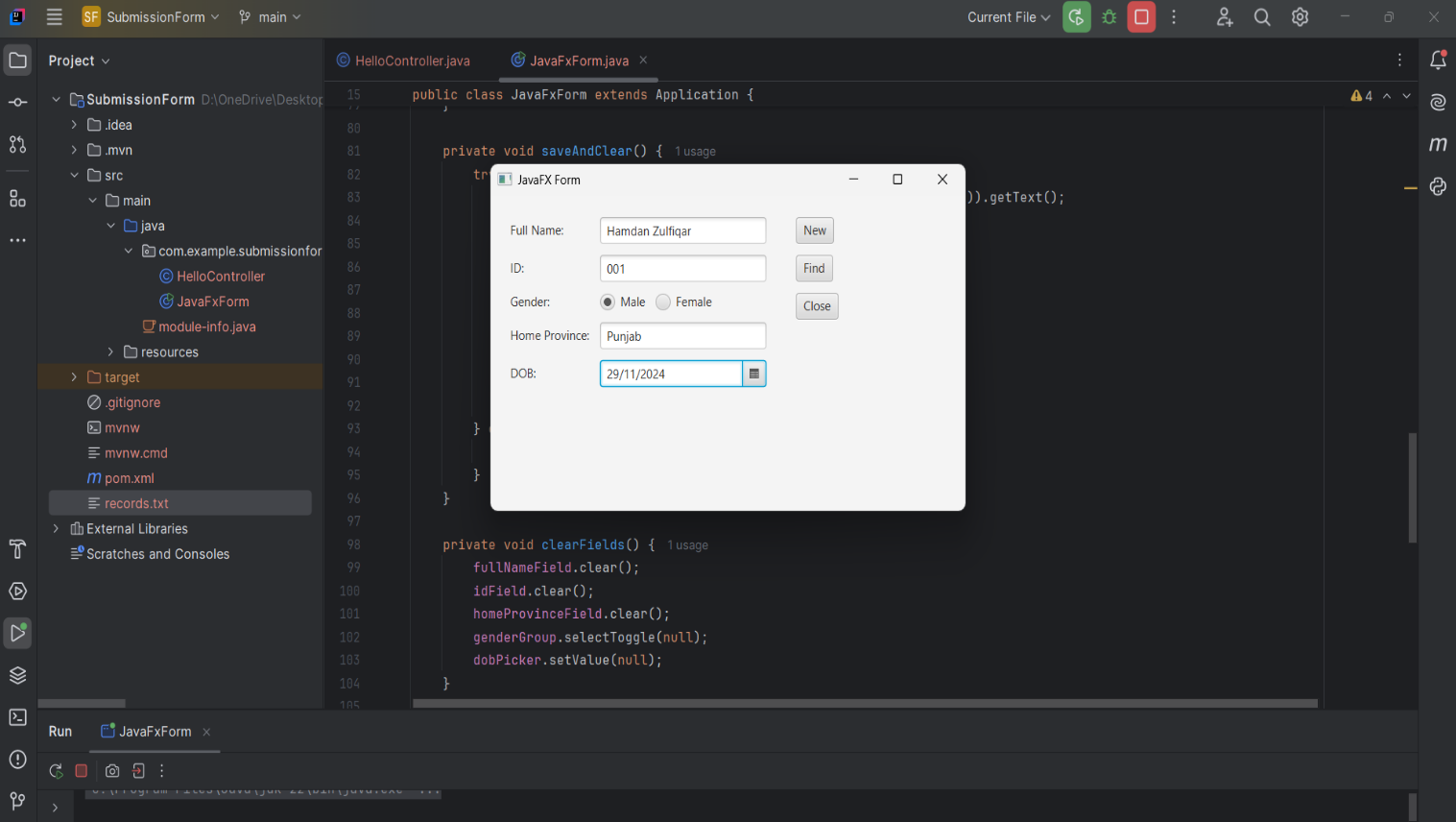
Sir Shahid Bhatti.

**Submitted by**:

Hamdan Zulfiqar (SP24-BSE-032)

**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 you entered earlier and then click on the find button.

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.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) {  
 // 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 closeButton = new Button("Close");  
 Button findButton = new Button("Find");  
  
 // 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, findButton, closeButton);  
 buttonBox.setPadding(new Insets(10));  
  
 HBox root = new HBox(10, grid, buttonBox);  
 root.setPadding(new Insets(10));  
  
 // Button actions  
 newButton.setOnAction(e -> saveAndClear());  
 closeButton.setOnAction(e -> stage.close());  
 findButton.setOnAction(e -> findRecord());  
  
 // Scene setup  
 Scene scene = new Scene(root, 500, 300);  
 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%n",  
 fullNameField.getText(),  
 idField.getText(),  
 gender,  
 homeProvinceField.getText(),  
 dobPicker.getValue());  
  
 writer.write(record);  
 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 findRecord() {  
 String idToFind = idField.getText();  
 if (idToFind.isEmpty()) {  
 showAlert("Error", "Please enter an ID to find.");  
 return;  
 }  
  
 try {  
 List<String> records = Files.*readAllLines*(Paths.*get*(file.toURI()));  
 for (String record : records) {  
 String[] fields = record.split(",");  
 if (fields[1].equals(idToFind)) {  
 fullNameField.setText(fields[0]);  
 idField.setText(fields[1]);  
 homeProvinceField.setText(fields[3]);  
 dobPicker.setValue(java.time.LocalDate.*parse*(fields[4]));  
 genderGroup.selectToggle(fields[2].equals("Male") ? genderGroup.getToggles().get(0) : genderGroup.getToggles().get(1));  
 return;  
 }  
 }  
 showAlert("Not Found", "No record found with the given ID.");  
 } catch (IOException e) {  
 showAlert("Error", "Failed to read records.");  
 }  
 }  
  
 private void showAlert(String title, String message) {  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle(title);  
 alert.setContentText(message);  
 alert.showAndWait();  
 }  
}