****

**COMSATS UNIVERSITY ISLAMABAD, LAHORE CAMPUS**

**Title:**

**Lab Assignment**

**Subject:**

**Object Oriented Programing**

**Submitted to:**

**Sir Shahid Bhatti**

**Class:**

**SP24-BSE-A**

**Submitted by:**

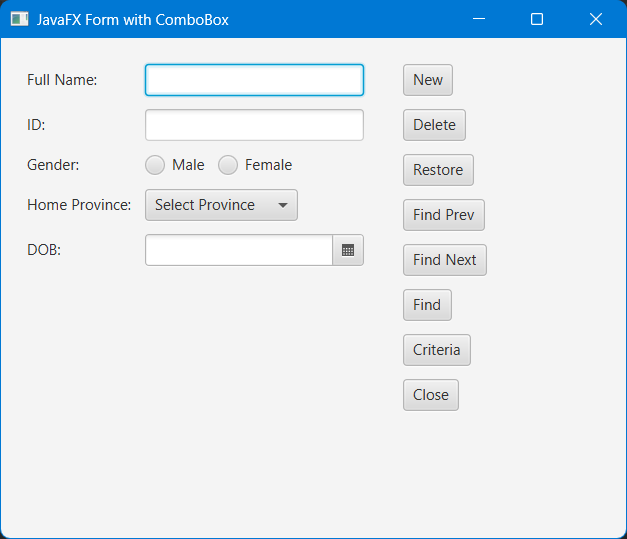
**MUDDASAR ALI GHAFOOR**

**Registration No:**

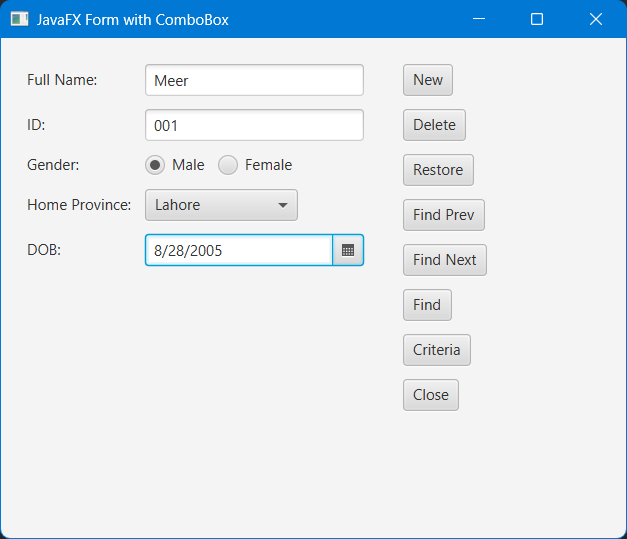
**SP24-BSE-062**

**Submission Date: November 29, 2024**

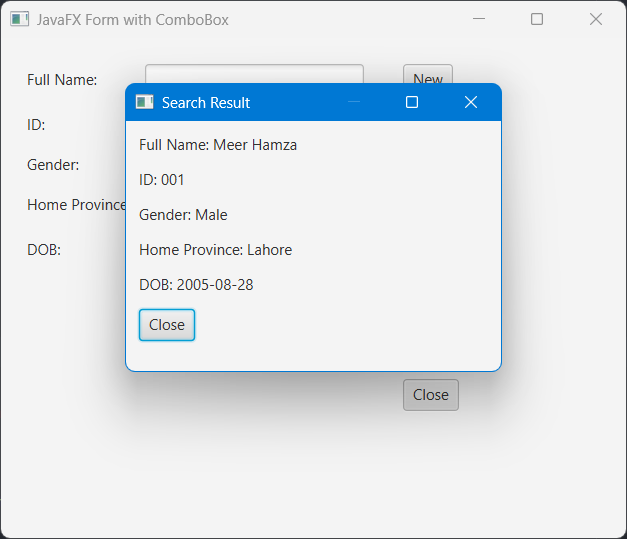
**Form:**

****

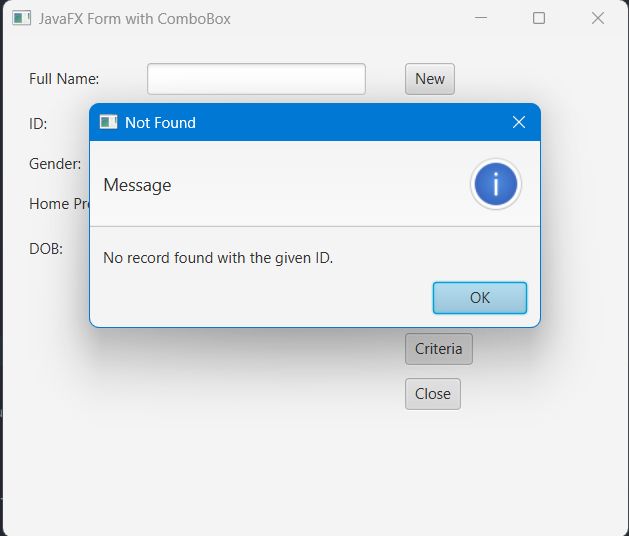
**Form Filling:**

****

**Search Result:**

****

**Not Found:**

****

**File Handling:**

**A screenshot of a computer

Description automatically generated**

**Code:**

*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* Main *extends* Application {  
 *private* TextField fullNameField, idField;  
 *private* ComboBox<String> homeProvinceComboBox;  
 *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();  
 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);  
  
 *// HomeProvince combo box* homeProvinceComboBox = *new* ComboBox<>();  
 homeProvinceComboBox.getItems().addAll("Lahore", "Karachi", "Faisalabad", "Multan", "Islamabad");  
 homeProvinceComboBox.setPromptText("Select Province");  
  
 *// Creating buttons* Button newButton = *new* Button("New");  
 Button deleteButton = *new* Button("Delete");  
 Button restoreButton = *new* Button("Restore");  
 Button findPrevButton = *new* Button("Find Prev");  
 Button findNextButton = *new* Button("Find Next");  
 Button criteriaButton = *new* Button("Criteria");  
 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(homeProvinceComboBox, 1, 3);  
 grid.add(*new* Label("DOB:"), 0, 4);  
 grid.add(dobPicker, 1, 4);  
  
 VBox buttonBox = *new* VBox(10, newButton, deleteButton, restoreButton, findPrevButton, findNextButton, findButton, criteriaButton, 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(stage));  
  
 *// Scene setup* Scene scene = *new* Scene(root, 500, 400);  
 stage.setScene(scene);  
 stage.setTitle("JavaFX Form with ComboBox");  
 stage.show();  
 }  
  
 *private void* saveAndClear() {  
 *try* (FileWriter writer = *new* FileWriter(file, *true*)) {  
 String gender = genderGroup.getSelectedToggle() != *null* ? ((RadioButton) genderGroup.getSelectedToggle()).getText()  
 : "N/A";  
 String homeProvince = homeProvinceComboBox.getValue() != *null* ? homeProvinceComboBox.getValue()  
 : "N/A";  
  
 String record = String.*format*("%s,%s,%s,%s,%s%n",  
 fullNameField.getText(),  
 idField.getText(),  
 gender,  
 homeProvince,  
 dobPicker.getValue());  
  
 writer.write(record);  
 clearFields();  
 } *catch* (IOException e) {  
 showAlert("Error", "Failed to save record.");  
 }  
 }  
  
 *private void* clearFields() {  
 fullNameField.clear();  
 idField.clear();  
 genderGroup.selectToggle(*null*);  
 homeProvinceComboBox.setValue(*null*);  
 dobPicker.setValue(*null*);  
 }  
  
 *private void* findRecord(Stage parentStage) {  
 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)) {  
 showSearchResult(parentStage, fields);  
 *return*;  
 }  
 }  
 showAlert("Not Found", "No record found with the given ID.");  
 } *catch* (IOException e) {  
 showAlert("Error", "Failed to read records.");  
 }  
 }  
  
 *private void* showSearchResult(Stage parentStage, String[] fields) {  
 Stage resultStage = *new* Stage();  
 resultStage.initOwner(parentStage);  
  
 VBox resultBox = *new* VBox(10);  
 resultBox.setPadding(*new* Insets(10));  
 resultBox.getChildren().addAll(  
 *new* Label("Full Name: " + fields[0]),  
 *new* Label("ID: " + fields[1]),  
 *new* Label("Gender: " + fields[2]),  
 *new* Label("Home Province: " + fields[3]),  
 *new* Label("DOB: " + fields[4])  
 );  
  
 Button closeButton = *new* Button("Close");  
 closeButton.setOnAction(e -> resultStage.close());  
  
 resultBox.getChildren().add(closeButton);  
  
 Scene scene = *new* Scene(resultBox, 300, 200);  
 resultStage.setScene(scene);  
 resultStage.setTitle("Search Result");  
 resultStage.show();  
 }  
  
 *private void* showAlert(String title, String message) {  
 Alert alert = *new* Alert(Alert.AlertType.INFORMATION);  
 alert.setTitle(title);  
 alert.setContentText(message);  
 alert.showAndWait();  
 }  
}