**Sir Shahid Bhatti**

**SUBMITTED TO:**

**Muhibba Fatima (SP24-BSE-B-138)**

**SUBMITTED BY:**

**Object Oriented Programming**

**SUBJECT:**

**ASSIGNMENT # 2**

**Explanation of the output:**

When we run the code, it will appear:

This is the form with all different fields like name, father name, CNIC number, date, gender, city and choose image option. Now we will fill this form.

This is due to the code:

First of all, titleLabel creates a label for title applies styling such as font size, color, and border radius using CSS. VBox set padding and aligns the title. For name, father name, CNIC, labels are created. Each label is associated with a text field to accept user input.

**Date Picker:**

Now, after filling name, father name, CNIC, there is date picker. The user will pick a date and it will appear in the text field as it is shown above. It is due to the code, where we use DatePicker to accept user input.

After the date picker, there is an option for gender, which is showing two options: male and female. We will choose one. The code for this is:

This code creates two radio buttons, male and female and groups them into a toggleGroup, ensuring one option can be selected at a time.

**Combo Box:**

Then there is a combo box for selecting city which is shown below now:

The code for this combo box is:

This ensures the user to select an option from the list of city names. It is adding different city names in the combo box. The user will select only one option.

**Image selection:**

A button is provided to select an image from the files. The output is shown as:

This is due to the code:

The code creates a UI section where an image file may be selected for display. A Label to invite the user to select an image and a Button to open a FileChooser dialog for the selection of an image file is created. In a VBox, the selected image is placed inside an ImageView which maintains the aspect ratio of the image. The VBox has a border and padding applied for styling. The FileChooser filters for image files, and when an image is selected, it is displayed in the ImageView. This section is then added to the main layout, which places it alongside other form elements.

**Save Button:**

On clicking the save button, all the data is stored in the array list of erson, as we make the person class with such attributes and a toString method. Our data get stored in the array list. On clicking the save, a message “Form has been saved successfully” appears.

The code for the save button is:

This code creates a save button. When this button is clicked, it collects the data (name, father name, CNIC number, date, gender, city) entered by the user in the form fields. This data then creates a new person object which is added to the personList(an array list of person).

An information alert is displayed to confirm that the form has been saved correctly, displaying a message informing the user. Upon saving the form, all input fields such as name, father's name, date picker, gender radio buttons, and the city combo box are cleared. The ImageView is also reset to null to remove the image displayed in it.

Finally, the console prints the current list of persons that are added in the array list of person by filling the form. This allows debugging and verification of the added data.

**Explanation of code:**

This javaFX application creates a user form for collecting and managing personal information. The class Person models the user through fields like name, father's name, CNIC, date of birth, gender, and the city. A constructor and getter methods with a function for toString method for easy display have been used.

The HelloApplication class codes up the graphical interface. A VBox styled with a Label serves as the title. The form layout, built by a GridPane, consists of the following labeled input fields: TextField for name, father's name, and CNIC, the DatePicker for date, RadioButton for gender, and the ComboBox to select the city to live in.

The Save button collects input values, creates a Person object, and adds it to an ArrayList. An alert confirms that the form has been saved. After saving, all fields are cleared, and the displayed image (if any) is reset. The current list of saved Person objects is printed to the console.

The form features an image upload feature using a FileChooser. The user can choose an image, and then it will be displayed in an ImageView styled with a border and background. This section is structured in a VBox.

The HBox of this application positions the form and image sections horizontally. The application contains all its constituent parts wrapped in a root VBox that has padding and background color and is shown in a Scene. The application launches with a specified window size and title. This design is well structured and shows proper usage of JavaFX for form-based user interfaces.