**Documentation of the "Comsats Data Entry Form" JavaFX Application**

**1. Introduction**

The "Comsats Data Entry Form" is a simple JavaFX application designed to create a personal data entry form. The user can input their details, including Name, Father’s Name, CNIC Number, Date of Birth, Gender, and City. In addition to the text fields, the form allows the user to upload an image file. Once the user clicks the "Save" button, the form data is printed to the console for further use.

**2. Code Overview**

This application leverages the JavaFX framework to create an intuitive user interface with various components like TextField, ComboBox, RadioButton, DatePicker, Button, and ImageView. These components allow the user to input information and select an image, which is then displayed in the form.

The program consists of one primary class, HelloApplication, which contains the logic for creating the form, managing user input, and displaying the output.

**3. Class and Method Explanation**

HelloApplication Class

* Purpose: This is the main class of the application. It extends the Application class and overrides the start(Stage stage) method, which is used to initialize the JavaFX UI components and set up the layout.
* start(Stage stage):
  + Initializes UI components such as labels, text fields, radio buttons, combo box, and buttons.
  + Sets up layout using GridPane and adds styling to each component.
  + Configures the event handling for button actions (file selection and saving data).

**4. User Interface (UI) Components**

The form is composed of various JavaFX UI components, each serving a specific purpose:

* GridPane: Used to structure the layout in rows and columns. Padding and gaps between elements are set for better spacing.
* Label: Text labels such as "Name", "Father's Name", "CNIC Number", etc., are displayed next to the corresponding input fields.
* TextField: For text input fields like Name, Father’s Name, and CNIC. These fields use setPromptText() to show placeholder text.
* DatePicker: A user-friendly date picker allows the user to select their Date of Birth.
* RadioButton: The user can choose between "Male" and "Female" for gender. A ToggleGroup ensures only one gender is selected at a time.
* ComboBox: Allows the user to select a city from a list of predefined options (Lahore, Islamabad, Wah).
* ImageView: Displays the selected image with fixed dimensions (80x80).
* Button: There are two buttons:
  1. "Choose File": Opens a file chooser dialog to select an image.
  2. "Save": Prints the entered data to the console.

**5. Handling User Inputs**

Each input field is associated with a UI component:

* TextFields capture text for Name, Father’s Name, and CNIC.
* DatePicker captures the Date of Birth.
* RadioButton captures the gender, with only one selected at any time due to the use of a ToggleGroup.
* ComboBox captures the city selection.

Once the user clicks the "Save" button, the data is retrieved from the fields and printed in the console. If the image is selected, its path is printed as well.

**6. File Selection Feature**

The FileChooser is used for selecting an image file. It supports common image formats like .jpg, .png, and .jpeg. The selected file’s path is then used to create an Image object, which is displayed in the ImageView.

Steps for file selection:

* The user clicks the "Choose File" button.
* A file chooser dialog opens, allowing the user to navigate and select an image.
* The image is displayed in the ImageView.

**7. Save Button Action**

When the "Save" button is clicked:

1. The entered data from the fields (Name, Father’s Name, CNIC, Date of Birth, Gender, City) is retrieved.
2. The data is printed to the console.
3. If an image is selected, the path of the image is also printed; otherwise, a message indicating no image was selected is displayed.

**8. Styling and Layout**

To make the form visually appealing, the UI components are styled with custom CSS:

* Labels have bold text and a font size of 14px.
* The GridPane layout is used to neatly arrange the components in rows and columns, with padding and gaps for better spacing.
* The ImageView for displaying the image has a fixed width and height.

**9. Conclusion**

The "Comsats Data Entry Form" application demonstrates how to create a simple, functional data entry form using JavaFX. It captures various user inputs, displays a file selection dialog for images, and prints the entered data to the console. The application makes use of several JavaFX components and basic event handling to create an effective user interface.

