**OOP Lab Assignment 2**

**Submitted by:** Aleesha Batool

**Registration No:** SP24-BSE-131-B

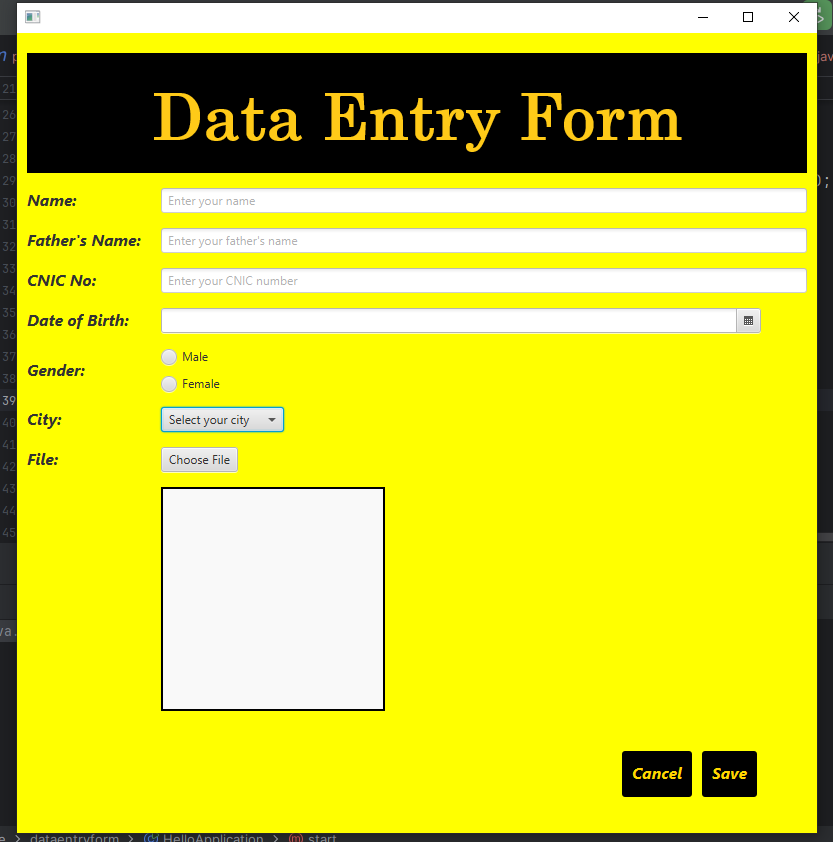
**JavaFX Data Entry Form Application**

**Purpose**

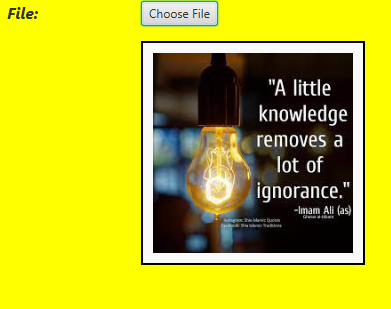
The application is designed as a **Data Entry Form** using JavaFX. Users can input personal information, including their name, father's name, CNIC number, gender, date of birth, city, and upload a file (image). The data is validated and stored in a list of Person objects, which can be accessed later.

**Features**

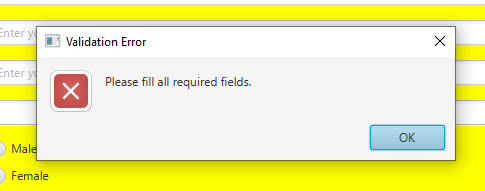
1. **Responsive Form Layout**:
   * Uses a GridPane for organized alignment of labels, input fields, and buttons.
   * Integrated visual elements like background color and banner images.



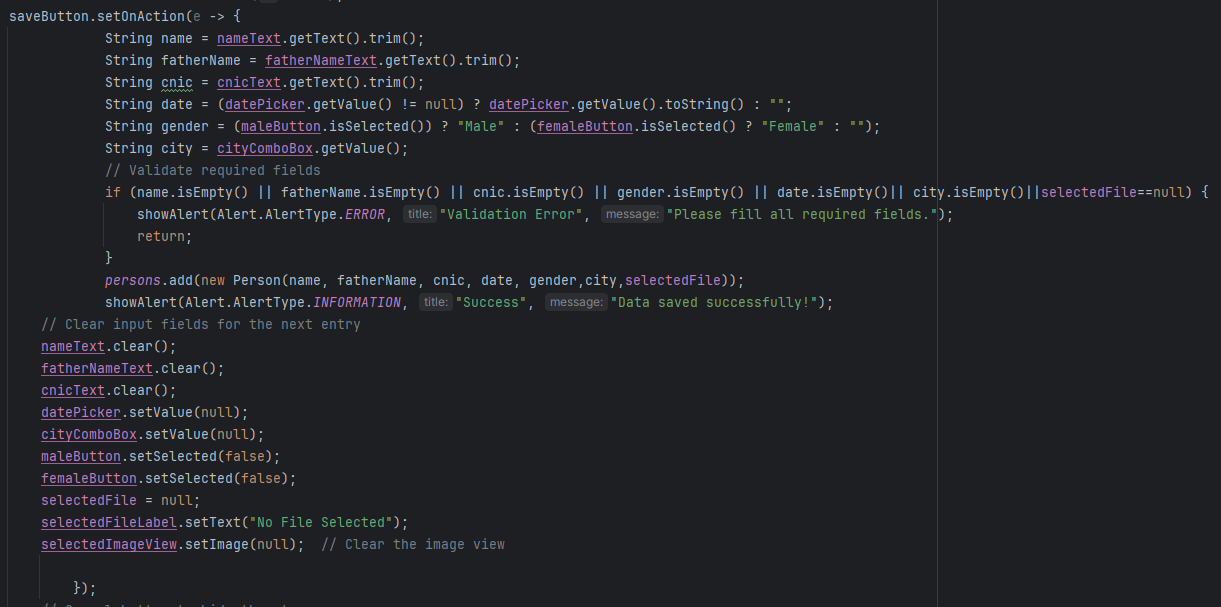
1. **User Input Fields**:
   * **TextFields** for name, father's name, and CNIC input.
   * **DatePicker** for date selection.
   * **ComboBox** for city selection.
   * **RadioButtons** with a ToggleGroup for selecting gender.
   * **FileChooser** for uploading an image file.
2. **File Handling**:
   * Displays the uploaded image in an ImageView.
   * Restricts file selection to specific image formats (\*.png, \*.jpeg, \*.jpg).



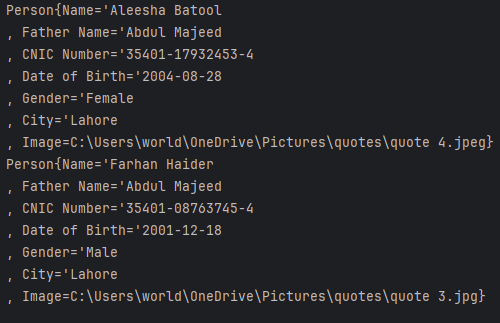
1. **Data Validation**:
   * Ensures that all fields are filled before saving data.
   * Alerts users if required fields are missing or invalid.



1. **Data Storage**:
   * Captures user input and stores it in a Person object.
   * Maintains a list of Person objects for easy retrieval.



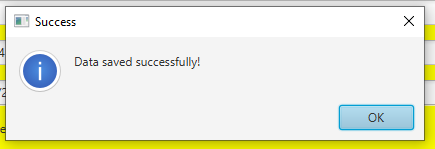
1. **Styling and Usability**:
   * Includes custom styles for labels, buttons, and the background.
   * Implements keyboard navigation with ENTER key to switch between fields.
   * Provides clear error messages and success notifications using alerts.
2. **Image Display**:
   * Dynamically updates the ImageView to show the selected image.
   * Clears the view upon saving or resetting the form.
3. **Cancel Button**:
   * Closes the form and prints all saved Person objects to the console.

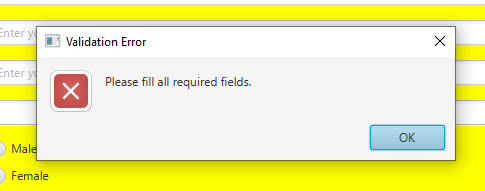


**Code Walkthrough**

**Main Class: HelloApplication**

1. **Attributes**:
   * Maintains a static ArrayList (persons) to store Person objects.
   * Uses a File object to track the selected image file.
2. **start(Stage stage) Method**:
   * Sets up the JavaFX UI components and their layout.
   * Initializes and styles the form elements.
   * Handles button actions (Save, Cancel, and Choose File).
3. **Helper Methods**:
   * showAlert: Displays alerts for success or error messages.





* + createStyledLabel: Generates uniformly styled labels for the form.

**Supporting Class: Person**

1. **Attributes**:
   * Represents user details such as name, father's name, CNIC, gender, date, city, and the uploaded image.
   * Encapsulated to ensure proper data management.
2. **Constructor**:
   * Initializes all fields of the Person class with provided values.
3. **toString Method**:
   * Returns a formatted string representation of the person's details for easy viewing.

**Key Functionalities**

1. **Save Button**:
   * Collects and validates user inputs.
   * Stores validated data into the persons list as a Person object.
   * Resets the form for the next entry.
2. **Cancel Button**:
   * Closes the application and prints saved data to the console.
3. **File Chooser**:
   * Opens a file dialog for image selection.
   * Displays the selected image in the form for verification.

**Validation**

The application ensures:

* All fields are filled before saving data.
* Only valid image files can be selected.
* Errors and success are clearly communicated using alerts.

**Visual Design**

* **Yellow Background**: Adds vibrancy to the UI.
* **Custom Button Styling**: Bold, italicized, and color-themed buttons enhance usability and appearance.
* **Dynamic ImageView**: Displays a preview of the uploaded image.

This application provides a complete framework for collecting and managing personal data using JavaFX. Its well-structured layout and validation mechanisms make it reliable and user-friendly.