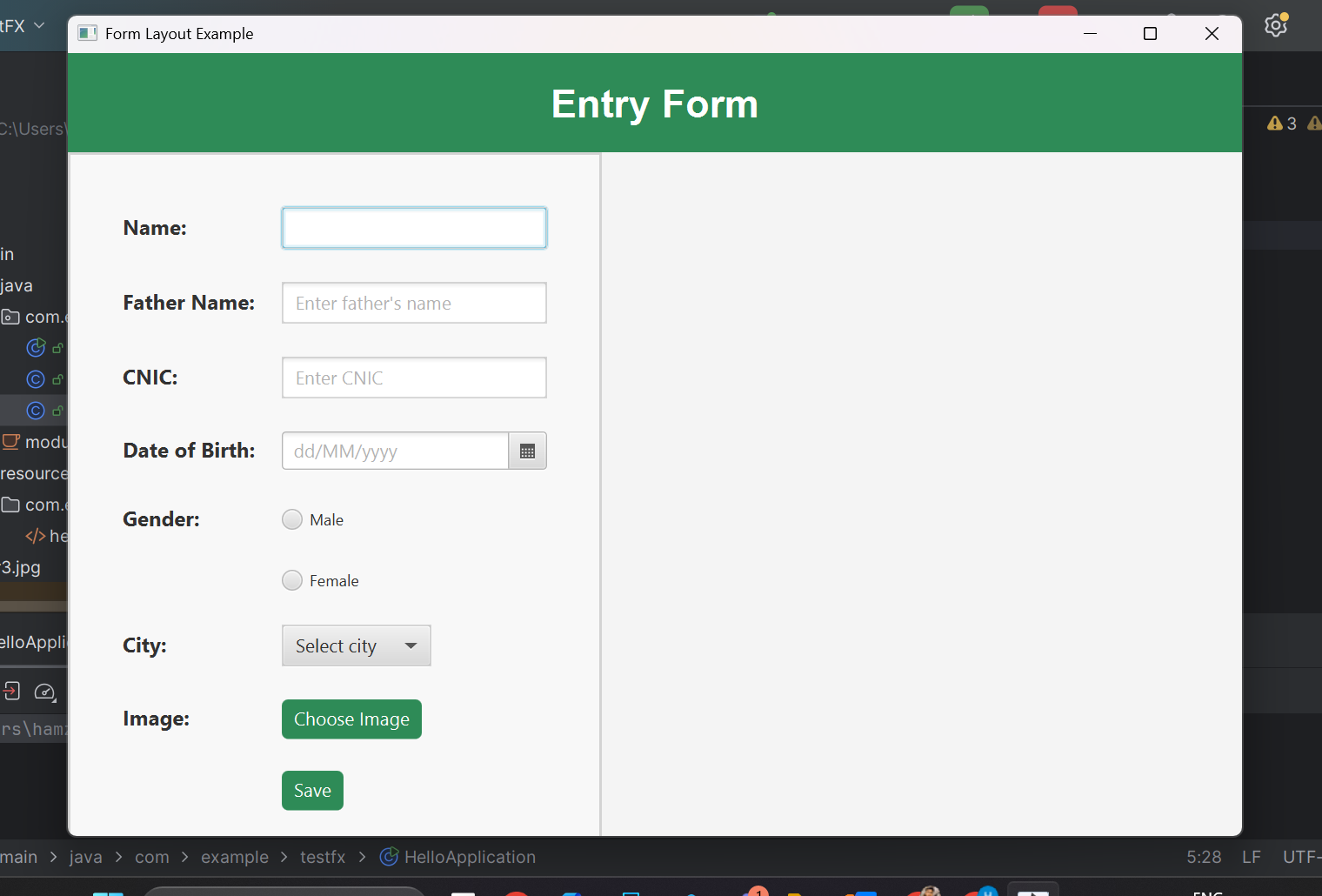
Code Review: Entry Form (JavaFX)

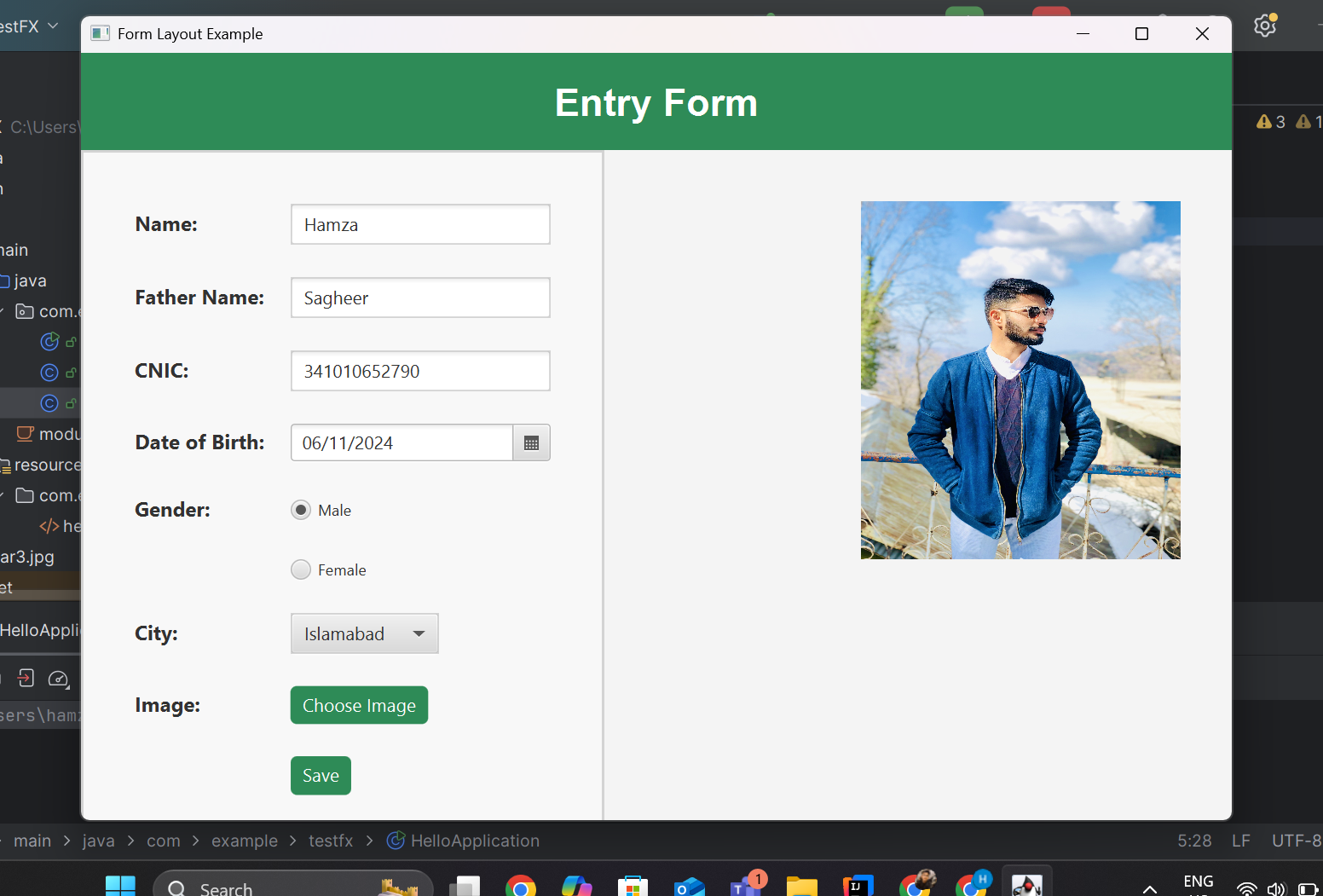
The HelloApplication class is a JavaFX application that provides a simple form to input personal information such as name, father's name, CNIC, date of birth, gender, city, and image. Upon filling the form, the details are saved into an ArrayList of Person objects. This document reviews the functionality of the key parts of the application.

# Output Review

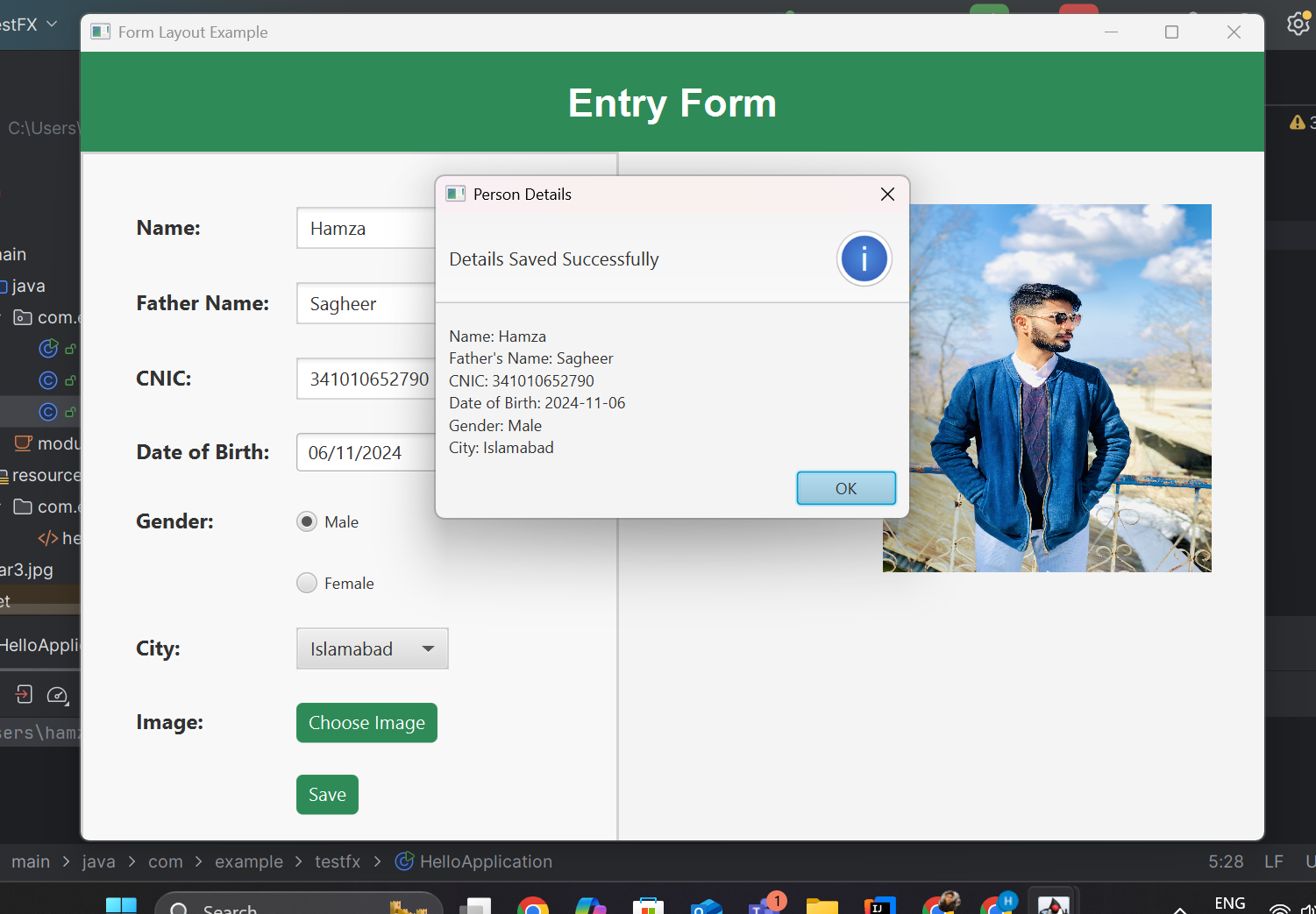
First of all , whenever I run my code , the box appear in this format given below :



And when I fill all the boxes and add image then the box changed into this format :



And when I click on Save button , form changed into this shape :

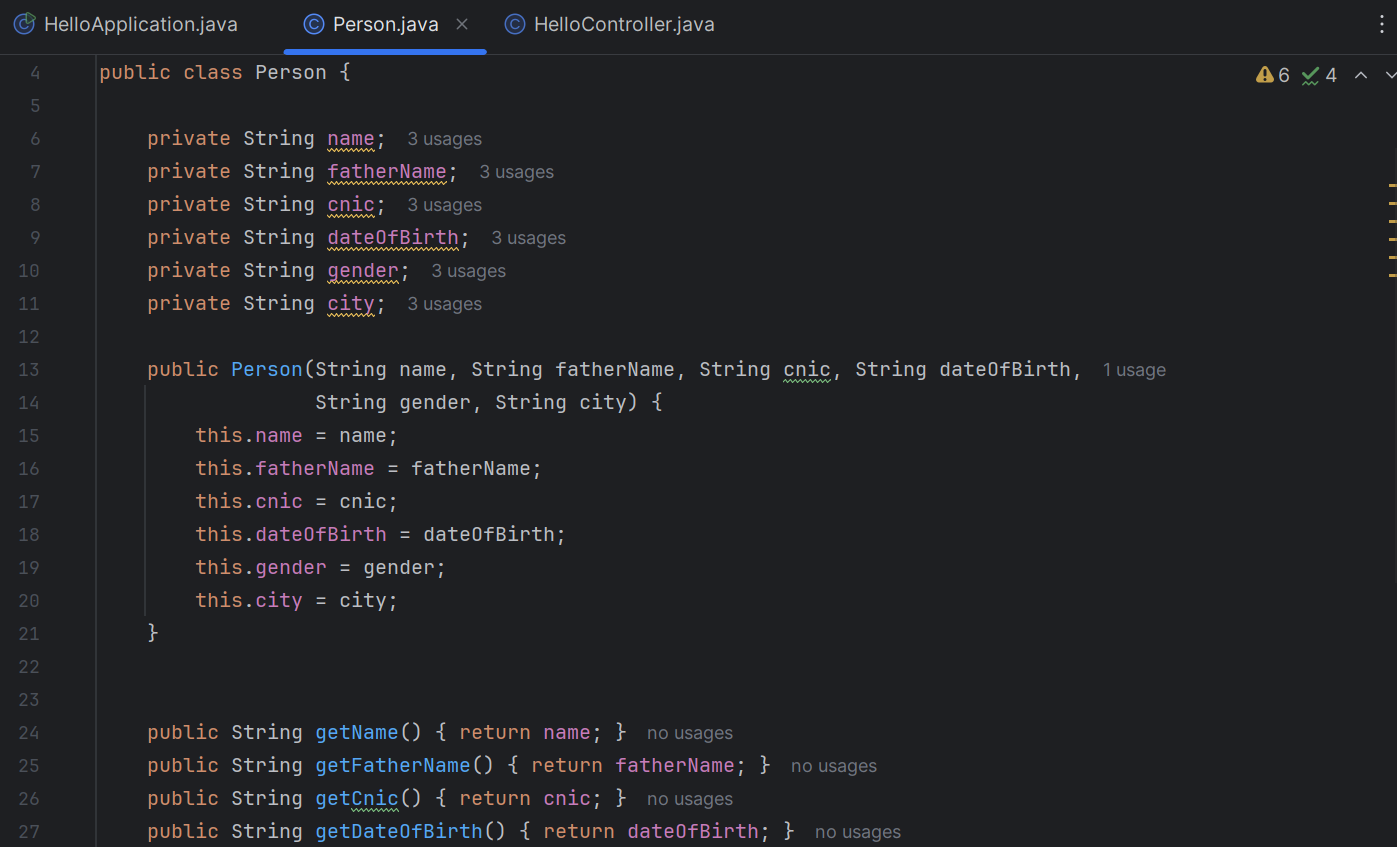


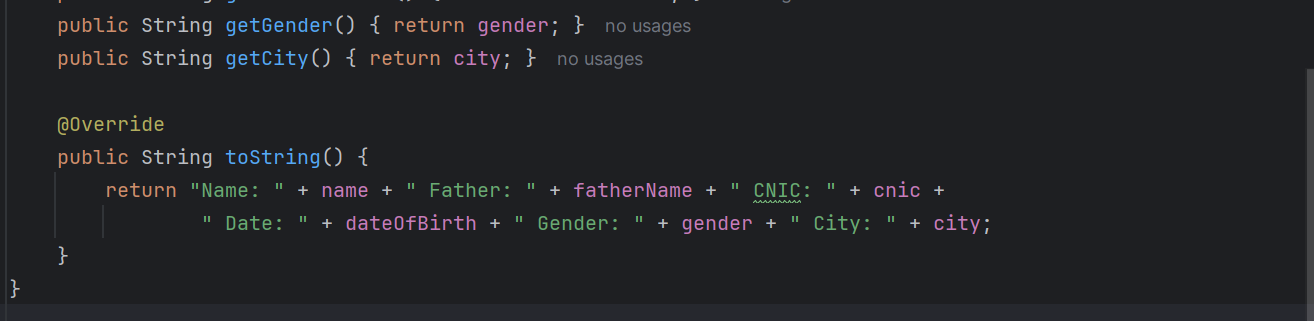
And it returns to default mode when I click on Ok .

# Functionality Review

## 1. Person List Initialization

An ArrayList called 'personList' is used to store Person objects. The 'Person' class holds individual user details such as name, father’s name, CNIC, date of birth, gender, and city. This list will collect all the entries submitted by the user.





## 2. BorderPane

The main layout of the application uses BorderPane, which organizes the content into a top section for the banner, left section for the form inputs, and right section for displaying the selected image. This structure allows for a clear, organized form.

## 3. Label Banner

The 'Entry Form' label at the top of the form is styled with custom font size, weight, and color. It serves as the heading for the form, making it clear for users that they are filling out an entry form.

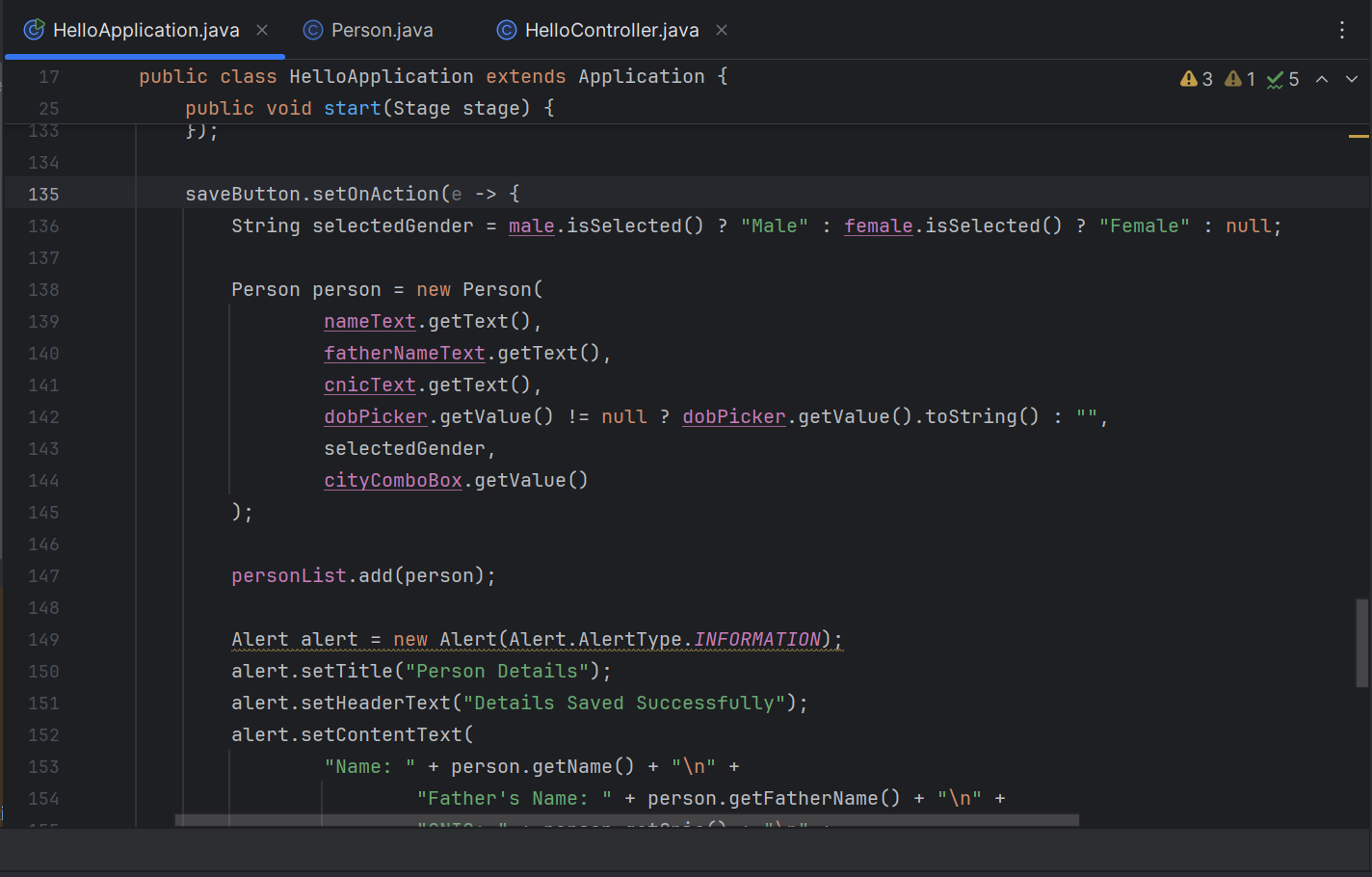
## 4. GridPane

The form inputs, including fields for name, father’s name, CNIC, date of birth, gender, and city, are placed in a GridPane. Each field is labeled clearly, and there are TextField, DatePicker, RadioButton, and ComboBox controls for user input.

## 5. ImageView

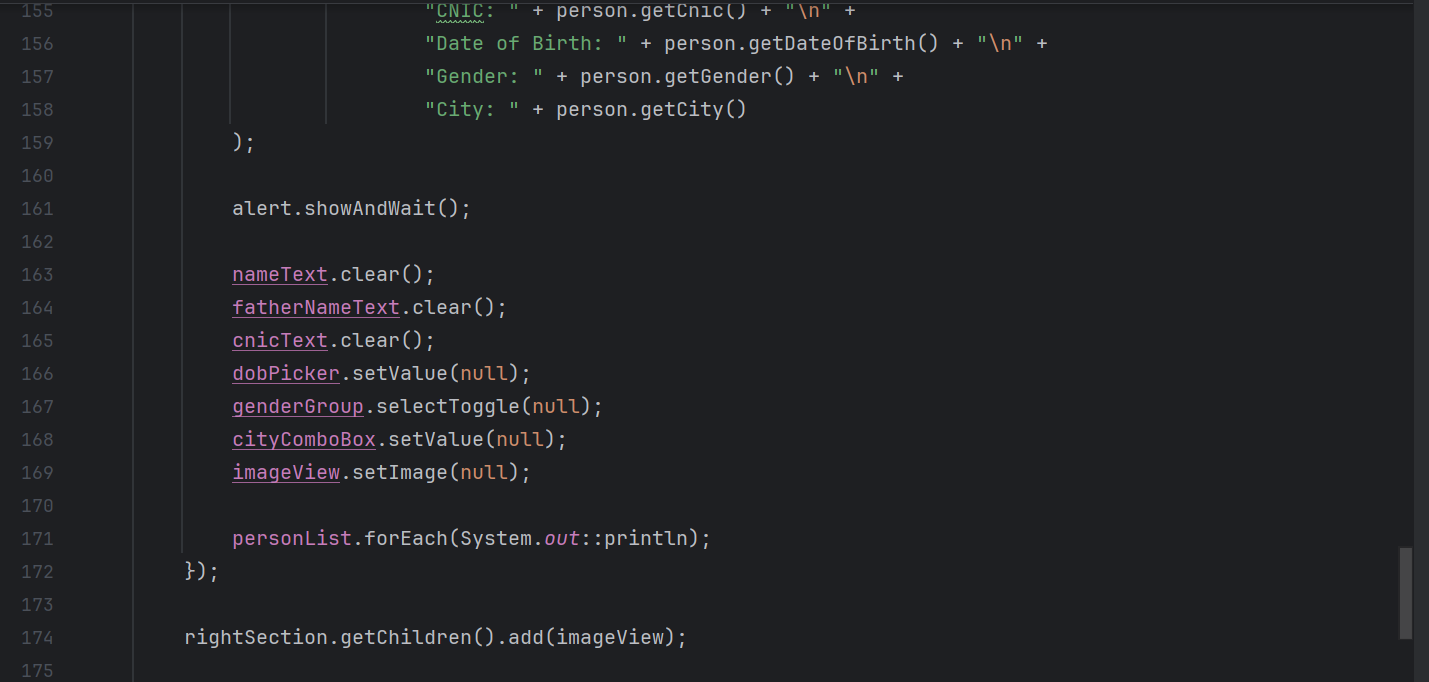
The user can select an image by clicking the 'Choose Image' button. The image is displayed in an ImageView in the right section of the window. The image is loaded using a FileChooser, which supports common image file types like PNG, JPG, and GIF.

## 6. Save Button Functionality

When the user fills out the form and clicks 'Save', the input data is collected and stored in a new Person object. This object is then added to the 'personList'. A success message is displayed using **an alert dialogue** , showing the details that were saved. The form fields are cleared after saving.

## 7. Form Resetting After Save

After the form data is saved, all the input fields (TextFields, ComboBox, RadioButtons, DatePicker) are cleared. The image selection is also reset to ensure the user can easily enter new data.



## 8. Displaying Saved Entries

Once the form is saved, all the entries in the 'personList' are printed to the console. This is useful for testing or debugging purposes.