

**Course: Object Oriented Programming**

**Lab Assignment#4:**

**Title:**

***“A product implementing java fx, file handling and Object-Oriented Programming****”*

|  |  |
| --- | --- |
| Group Members: | Aisha  Aamna Tariq  Muhammad Sameer Yaseen |
| Registration number: | SP24-BSE-142-A  SP24-BSE-001-A  SP24-BSE-085-A |
| Assignment number: | 4 |
| Assigned date: | 10-12-2024 |
| Course instructor: | Sir Shahid Bhatti |
| Due Date: | 19-12-2024 |

***1.HelloApplication (Main File):***

package com.example.demo8;

import javafx.application.Application;

import javafx.scene.Cursor;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.effect.DropShadow;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.layout.\*;

import javafx.scene.paint.Color;

import javafx.scene.text.Font;

import javafx.stage.Stage;

import javafx.geometry.Pos;

import javafx.scene.shape.Line;

import javafx.scene.text.Text;

import java.io.\*;

import java.util.Objects;

public class HelloApplication extends Application {

    private VBox loginForm;

    private VBox newPasswordForm;

    private VBox createAccountForm;

    private TextField usernameField;

    private PasswordField passwordField;

    private PasswordField newPasswordField;

    private PasswordField confirmPasswordField;

    private TextField emailField;

    private TextField usernameFieldForReset;

    private TextField newUsernameFieldForAccount;

    private TextField emailFieldForAccount;

    private PasswordField newPasswordFieldForAccount;

    private PasswordField confirmNewPasswordFieldForAccount;

    private final String USER\_CREDENTIALS\_FILE = "user\_credentials.txt";

    @Override

    public void start(Stage stage) {

        Pane root = new Pane();

        Scene galleryScene = new Scene(root, 1100, 600, Color.PINK);

        stage.setScene(galleryScene);

        stage.setTitle("ART GALLERY");

        Image image = new Image(Objects.requireNonNull(getClass().getResource("/com/example/demo8/Pookie.png")).toExternalForm());

        stage.getIcons().add(image);

        stage.setFullScreen(true);

        loginForm = new VBox(10);

        loginForm.setAlignment(Pos.CENTER);

        loginForm.setStyle("-fx-background-color: transparent; -fx-padding: 30px; -fx-border-radius: 30px; -fx-background-radius: 30px;");

        loginForm.layoutXProperty().bind(root.widthProperty().subtract(loginForm.widthProperty()).divide(2));

        loginForm.layoutYProperty().bind(root.heightProperty().subtract(loginForm.heightProperty()).divide(2).add(50));

        BackgroundFill b1 = new BackgroundFill(Color.PINK, null, null);

        Background b2 = new Background(b1);

        root.setBackground(b2);

        Image image1 = new Image(Objects.requireNonNull(getClass().getResource("/com/example/demo8/img\_1.png")).toExternalForm());

        BackgroundImage backgroundImage = new BackgroundImage(

                image1,

                BackgroundRepeat.NO\_REPEAT,

                BackgroundRepeat.NO\_REPEAT,

                BackgroundPosition.CENTER,

                new BackgroundSize(

                        BackgroundSize.AUTO,

                        BackgroundSize.AUTO,

                        false,

                        false,

                        true,

                        true

                )

        );

        Background background = new Background(backgroundImage);

        root.setBackground(background);

        Text text = new Text("WELCOME TO ART GALLERY");

        text.setFont(Font.font("Algerian", 60));

        text.setFill(Color.BLACK);

        text.setX(250);

        text.setY(150);

        DropShadow shadow = new DropShadow();

        text.setEffect(shadow);

        Line line = new Line();

        line.setStartX(250);

        line.setStartY(180);

        line.setEndX(1050);

        line.setEndY(180);

        line.setStrokeWidth(5);

        HBox loginContainer = new HBox(20);

        loginContainer.setAlignment(Pos.CENTER);

        loginContainer.layoutXProperty().bind(root.widthProperty().subtract(loginContainer.widthProperty()).divide(2));

        loginContainer.layoutYProperty().bind(root.heightProperty().subtract(loginContainer.heightProperty()).divide(2).add(50));

        Image loginImage = new Image(Objects.requireNonNull(getClass().getResource("/com/example/demo8/img\_3.png")).toExternalForm());

        ImageView loginImageView = new ImageView(loginImage);

        loginImageView.setFitWidth(400);

        loginImageView.setFitHeight(400);

        loginImageView.setPreserveRatio(true);

        loginContainer.getChildren().addAll(loginImageView, loginForm);

        root.getChildren().add(loginContainer);

        Label usernameLabel = new Label("Username:");

        usernameLabel.setFont(Font.font("Arial", 16));

        usernameLabel.setTextFill(Color.BLACK);

        usernameField = new TextField();

        usernameField.setPromptText("Enter your username");

        Label passwordLabel = new Label("Password:");

        passwordLabel.setFont(Font.font("Arial", 16));

        passwordLabel.setTextFill(Color.BLACK);

        passwordField = new PasswordField();

        passwordField.setPromptText("Enter your password");

        Button loginButton = new Button("Login");

        loginButton.setFont(Font.font("Arial", 14));

        loginButton.setOnAction(e -> handleLogin(stage));

        loginButton.setCursor(Cursor.HAND);

        Hyperlink forgotPasswordLink = new Hyperlink("Forgot Password?");

        forgotPasswordLink.setTextFill(Color.BLACK);

        forgotPasswordLink.setOnAction(e -> toggleForms());

        forgotPasswordLink.setCursor(Cursor.HAND);

        Hyperlink createAccountLink = new Hyperlink("Create a New Account");

        createAccountLink.setTextFill(Color.BLACK);

        createAccountLink.setOnAction(e -> showCreateAccountForm());

        createAccountLink.setCursor(Cursor.HAND);

        loginForm.getChildren().addAll(usernameLabel, usernameField, passwordLabel, passwordField, loginButton, forgotPasswordLink, createAccountLink);

        newPasswordForm = new VBox(10);

        newPasswordForm.setAlignment(Pos.CENTER);

        newPasswordForm.setStyle("-fx-background-color: TRANSPARENT; -fx-padding: 20px; -fx-border-radius: 10px; -fx-background-radius: 10px;");

        newPasswordForm.setLayoutX(400);

        newPasswordForm.setLayoutY(250);

        newPasswordForm.setVisible(false);

        newPasswordForm.layoutXProperty().bind(root.widthProperty().subtract(newPasswordForm.widthProperty()).divide(2));

        newPasswordForm.layoutYProperty().bind(root.heightProperty().subtract(newPasswordForm.heightProperty()).divide(2).add(50));

        Label usernameLabelForReset = new Label("Username:");

        usernameLabelForReset.setFont(Font.font("Arial", 16));

        usernameLabelForReset.setTextFill(Color.BLACK);

        usernameFieldForReset = new TextField();

        usernameFieldForReset.setPromptText("Enter your username");

        Label emailLabelForReset = new Label("Email:");

        emailLabelForReset.setFont(Font.font("Arial", 16));

        emailLabelForReset.setTextFill(Color.BLACK);

        emailField = new TextField();

        emailField.setPromptText("Enter your email");

        Label newPasswordLabel = new Label("New Password:");

        newPasswordLabel.setFont(Font.font("Arial", 16));

        newPasswordLabel.setTextFill(Color.BLACK);

        newPasswordField = new PasswordField();

        newPasswordField.setPromptText("Enter your new password");

        Label confirmPasswordLabel = new Label("Confirm Password:");

        confirmPasswordLabel.setFont(Font.font("Arial", 16));

        confirmPasswordLabel.setTextFill(Color.BLACK);

        confirmPasswordField = new PasswordField();

        confirmPasswordField.setPromptText("Retype your new password");

        Button resetPasswordButton = new Button("Reset Password");

        resetPasswordButton.setFont(Font.font("Arial", 14));

        resetPasswordButton.setOnAction(e -> resetPassword());

        resetPasswordButton.setCursor(Cursor.HAND);

        Button backToLoginButtonFromReset = new Button("Back");

        backToLoginButtonFromReset.setFont(Font.font("Arial", 14));

        backToLoginButtonFromReset.setOnAction(e -> switchToLoginForm());

        backToLoginButtonFromReset.setCursor(Cursor.HAND);

        newPasswordForm.getChildren().addAll(usernameLabelForReset, usernameFieldForReset, emailLabelForReset, emailField, newPasswordLabel, newPasswordField, confirmPasswordLabel, confirmPasswordField, resetPasswordButton, backToLoginButtonFromReset);

        createAccountForm = new VBox(10);

        createAccountForm.setAlignment(Pos.CENTER);

        createAccountForm.setStyle("-fx-background-color: transparent; -fx-padding: 20px; -fx-border-radius: 10px; -fx-background-radius: 10px;");

        createAccountForm.setLayoutX(400);

        createAccountForm.setLayoutY(250);

        createAccountForm.setVisible(false);

        createAccountForm.layoutXProperty().bind(root.widthProperty().subtract(createAccountForm.widthProperty()).divide(2));

        createAccountForm.layoutYProperty().bind(root.heightProperty().subtract(createAccountForm.heightProperty()).divide(2).add(50));

        Label newUsernameLabel = new Label("Username:");

        newUsernameLabel.setFont(Font.font("Arial", 16));

        newUsernameLabel.setTextFill(Color.BLACK);

        newUsernameFieldForAccount = new TextField();

        newUsernameFieldForAccount.setPromptText("Enter your username");

        Label emailLabel = new Label("Email:");

        emailLabel.setFont(Font.font("Arial", 16));

        emailLabel.setTextFill(Color.BLACK);

        emailFieldForAccount = new TextField();

        emailFieldForAccount.setPromptText("Enter your email");

        Label newPasswordLabelForAccount = new Label("Password:");

        newPasswordLabelForAccount.setFont(Font.font("Arial", 16));

        newPasswordLabelForAccount.setTextFill(Color.BLACK);

        newPasswordFieldForAccount = new PasswordField();

        newPasswordFieldForAccount.setPromptText("Enter your password");

        Label confirmNewPasswordLabel = new Label("Confirm Password:");

        confirmNewPasswordLabel.setFont(Font.font("Arial", 16));

        confirmNewPasswordLabel.setTextFill(Color.BLACK);

        confirmNewPasswordFieldForAccount = new PasswordField();

        confirmNewPasswordFieldForAccount.setPromptText("Retype your password");

        Button createAccountButton = new Button("Create Account");

        createAccountButton.setFont(Font.font("Arial", 14));

        createAccountButton.setOnAction(e -> handleCreateAccount());

        createAccountButton.setCursor(Cursor.HAND);

        Button backToLoginButtonFromAccount = new Button("Back");

        backToLoginButtonFromAccount.setFont(Font.font("Arial", 14));

        backToLoginButtonFromAccount.setOnAction(e -> switchToLoginForm());

        backToLoginButtonFromAccount.setCursor(Cursor.HAND);

        createAccountForm.getChildren().addAll(newUsernameLabel, newUsernameFieldForAccount, emailLabel, emailFieldForAccount, newPasswordLabelForAccount, newPasswordFieldForAccount, confirmNewPasswordLabel, confirmNewPasswordFieldForAccount, createAccountButton, backToLoginButtonFromAccount);

        Button backToLoginButtonFromAdmin = new Button("Back");

        backToLoginButtonFromAdmin.setFont(Font.font("Arial", 14));

        backToLoginButtonFromAdmin.setOnAction(e -> switchToLoginForm());

        backToLoginButtonFromAdmin.setCursor(Cursor.HAND);

        root.getChildren().addAll(text, line, loginForm, newPasswordForm, createAccountForm);

        stage.show();

    }

    private void handleLogin(Stage stage) {

        String username = usernameField.getText();

        String password = passwordField.getText();

        if (validateLogin(username, password)) {

            System.out.println("Login successful!");

            openArtworkManagerUI();

            stage.close();

        } else {

            System.out.println("Invalid username or password.");

            showAlert("Login Failed", "Invalid username or password. Please try again.");

        }

    }

    private boolean validateLogin(String username, String password) {

        try (BufferedReader br = new BufferedReader(new FileReader(USER\_CREDENTIALS\_FILE))) {

            String line;

            while ((line = br.readLine()) != null) {

                String[] credentials = line.split(",");

                if (credentials[0].equals(username) && credentials[2].equals(password)) {

                    return true;

                }

            }

        } catch (IOException e) {

            e.printStackTrace();

        }

        return false;

    }

    private void openArtworkManagerUI() {

        ArtworkManagerUI artworkManagerUI = new ArtworkManagerUI();

        Stage artworkStage = new Stage();

        Scene artworkScene = new Scene(artworkManagerUI, 800, 600);

        artworkStage.setTitle("Artwork Manager");

        artworkStage.setScene(artworkScene);

    }

    private void handleCreateAccount() {

        String username = newUsernameFieldForAccount.getText();

        String email = emailFieldForAccount.getText();

        String password = newPasswordFieldForAccount.getText();

        String confirmPassword = confirmNewPasswordFieldForAccount.getText();

        if (!password.equals(confirmPassword)) {

            showAlert("Password Mismatch", "Passwords do not match. Please check your password and try again.");

            return;

        }

        try (BufferedWriter writer = new BufferedWriter(new FileWriter(USER\_CREDENTIALS\_FILE, true))) {

            writer.write(username + "," + email + "," + password);

            writer.newLine();

            showAlert("Account Created", "Your account has been created successfully.");

            switchToLoginForm();

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

    private void resetPassword() {

        String username = usernameFieldForReset.getText();

        String email = emailField.getText();

        String newPassword = newPasswordField.getText();

        String confirmPassword = confirmPasswordField.getText();

        if (newPassword.isEmpty() || confirmPassword.isEmpty()) {

            showAlert("Empty Fields", "Please fill in both password fields.");

            return;

        }

        if (!newPassword.equals(confirmPassword)) {

            showAlert("Password Mismatch", "Passwords do not match. Please check your password and try again.");

            return;

        }

        try (BufferedReader br = new BufferedReader(new FileReader(USER\_CREDENTIALS\_FILE))) {

            StringBuilder newFileContent = new StringBuilder();

            String line;

            boolean userFound = false;

            while ((line = br.readLine()) != null) {

                String[] credentials = line.split(",");

                if (credentials[0].equals(username) && credentials[1].equals(email)) {

                    newFileContent.append(username).append(",").append(email).append(",").append(newPassword).append("\n");

                    userFound = true;

                } else {

                    newFileContent.append(line).append("\n");

                }

            }

            if (!userFound) {

                showAlert("User Not Found", "The username or email does not match our records.");

                return;

            }

            try (BufferedWriter bw = new BufferedWriter(new FileWriter(USER\_CREDENTIALS\_FILE))) {

                bw.write(newFileContent.toString());

            }

            showAlert("Success", "Your password has been reset successfully.");

        } catch (IOException e) {

            e.printStackTrace();

            showAlert("Error", "An error occurred while resetting the password. Please try again.");

        }

    }

    private void toggleForms() {

        loginForm.setVisible(false);

        newPasswordForm.setVisible(true);

    }

    private void switchToLoginForm() {

        loginForm.setVisible(true);

        newPasswordForm.setVisible(false);

        createAccountForm.setVisible(false);

    }

    private void showCreateAccountForm() {

        loginForm.setVisible(false);

        createAccountForm.setVisible(true);

    }

    private void showAlert(String title, String message) {

        Alert alert = new Alert(Alert.AlertType.INFORMATION);

        alert.setTitle(title);

        alert.setHeaderText(null);

        alert.setContentText(message);

        alert.showAndWait();

    }

    public static void main(String[] args) {

        launch();

    }

}

***2.HelloController***

package com.example.demo8;

import javafx.fxml.FXML;

import javafx.scene.control.Label;

public class HelloController {

    @FXML

    private Label welcomeText;

    @FXML

    protected void onHelloButtonClick() {

        welcomeText.setText("Welcome to JavaFX Application!");

    }

}

***3.ArtWorkManage UI***

package com.example.demo8;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.layout.\*;

import javafx.scene.paint.Color;

import javafx.scene.text.Font;

import javafx.stage.Stage;

import java.util.ArrayList;

import java.util.List;

import java.util.Objects;

public class ArtworkManagerUI extends VBox {

    private List<Artwork> artworks;

    private VBox artworkList;

    private TextField searchField;

    private Button searchButton;

    private Button clearButton;

    public ArtworkManagerUI() {

        artworks = new ArrayList<>();

        artworks.add(new Artwork("Artwork 1", "A beautiful painting", "Wassily Kandinsky", "$1000", "Artwork\_1.jpg", 15));

        artworks.add(new Artwork("Artwork 2", "A stunning sculpture", "Pablo Picasso", "$2000", "Artwork\_2.jpg", 25));

        artworks.add(new Artwork("Artwork 3", "An abstract piece", "Wassily Kandinsky", "$1500", "Artwork\_3.png", 22));

        artworks.add(new Artwork("Artwork 4", "A realistic Painting", "Gustave Courbet", "$3000", "Artwork\_4.jpg", 63));

        artworks.add(new Artwork("Artwork 5", "Whistler's Mother", "James Mcneill Whistler", "$25", "Artwork\_5.png", 56));

        artworks.add(new Artwork("Artwork 6", "Museo Del Prado in Madrid", "Diego Velazquez", "$99", "Artwork\_6.png", 44));

        artworks.add(new Artwork("Artwork 7", "The Young Women of Avignon", "Pablo Picasso", "$950", "Artwork\_7.png", 85));

        artworks.add(new Artwork("Artwork 8", "Reproduction Oil Painting", "John William Waterhouse", "$60", "Artwork\_8.png", 80));

        artworks.add(new Artwork("Artwork 9", "Laughing Cavalier", "Frans Hals", "$300", "Artwork\_9.png", 46));

        artworks.add(new Artwork("Artwork 10", "Van Gogh Self Portrait", "Vincent Van Gogh", "$700", "Artwork\_10.png", 36));

        artworks.add(new Artwork("Artwork 11", "The Fifer", "Edouard Manet", "$600", "Artwork\_11.png", 5));

        artworks.add(new Artwork("Artwork 12", "View of Toledo", "El Greco", "$2000", "Artwork\_12.png", 41));

        artworks.add(new Artwork("Artwork 13", "Napoleon Crossing The Alps", "Jacques Louis David", "$7000", "Artwork\_13.png", 75));

        artworks.add(new Artwork("Artwork 14", "Cafe Terrace At Night", "Vincent Van Gogh", "$400", "Artwork\_14.png", 50));

        artworks.add(new Artwork("Artwork 15", "Barge Haulers On The Volga", "Ilya Repin", "$300", "Artwork\_15.png", 23));

        artworks.add(new Artwork("Artwork 16", "Girl With The Pearl Earring", "Johannes Vermeer", "$200", "Artwork\_16.png", 30));

        artworks.add(new Artwork("Artwork 17", "Irises", "Vincent Van Gogh", "$2000", "Artwork\_17.png", 13));

        BorderPane root = new BorderPane();

        HBox topNav = new HBox(20);

        topNav.setPadding(new Insets(10));

        topNav.setStyle("-fx-background-color: Pink;");

        topNav.setAlignment(Pos.CENTER);

        Label greetingLabel = new Label("Welcome to Art Gallery");

        greetingLabel.setFont(Font.font("Algerian", 40));

        topNav.getChildren().addAll(greetingLabel);

        root.setTop(topNav);

        VBox header = new VBox(10);

        header.setPadding(new Insets(10));

        HBox searchBox = new HBox(10);

        searchBox.setAlignment(Pos.CENTER\_RIGHT);

        searchField = new TextField();

        searchField.setPromptText("Search by title or artist...");

        searchButton = new Button("Search");

        searchButton.setOnAction(e -> searchArtworks());

        clearButton = new Button("Clear");

        clearButton.setOnAction(e -> clearSearch());

        searchBox.getChildren().addAll(searchField, searchButton, clearButton);

        Button addNewArtworkButton = new Button("Add New Artwork +");

        header.getChildren().addAll(searchBox, addNewArtworkButton);

        artworkList = new VBox(10);

        artworkList.setPadding(new Insets(10));

        for (int i = 0; i < artworks.size(); i++) {

            artworkList.getChildren().add(createArtworkItem(artworks.get(i), i));

        }

        ScrollPane scrollPane = new ScrollPane(artworkList);

        scrollPane.setFitToWidth(true);

        VBox centerContent = new VBox(10);

        centerContent.getChildren().addAll(header, scrollPane);

        root.setCenter(centerContent);

        Label footerLabel = new Label("© All rights reserved");

        footerLabel.setPadding(new Insets(10));

        footerLabel.setAlignment(Pos.CENTER);

        footerLabel.setStyle("-fx-background-color: Pink;");

        root.setBottom(footerLabel);

        Scene scene = new Scene(root, 800, 600);

        Stage artworkStage = new Stage();

        artworkStage.setTitle("Artwork Manager");

        artworkStage.setScene(scene);

        artworkStage.show();

        addNewArtworkButton.setOnAction(e -> showAddArtworkForm());

    }

    private void showAddArtworkForm() {

        Stage addArtworkStage = new Stage();

        addArtworkStage.setTitle("Add New Artwork");

        VBox formLayout = new VBox(10);

        formLayout.setPadding(new Insets(20));

        formLayout.setStyle("-fx-background-color: Pink;");

        formLayout.setAlignment(Pos.CENTER);

        TextField titleField = new TextField();

        titleField.setPromptText("Artwork Title");

        TextField descriptionField = new TextField();

        descriptionField.setPromptText("Artwork Description");

        TextField artistField = new TextField();

        artistField.setPromptText("Artist Name");

        TextField priceField = new TextField();

        priceField.setPromptText("Artwork Price");

        TextField imageUrlField = new TextField();

        imageUrlField.setPromptText("Image URL");

        TextField stockField = new TextField();

        stockField.setPromptText("Stock Available");

        Button addButton = new Button("Add Artwork");

        Button cancelButton = new Button("Cancel");

        addButton.setOnAction(e -> {

            String title = titleField.getText();

            String description = descriptionField.getText();

            String artist = artistField.getText();

            String price = priceField.getText();

            String imageUrl = imageUrlField.getText();

            String stockAvailable = stockField.getText();

            if (title.isEmpty() || description.isEmpty() || artist.isEmpty() || price.isEmpty() || imageUrl.isEmpty() || stockAvailable.isEmpty()) {

                showAlert("Input Error", "Please fill in all fields.");

            } else {

                try {

                    int stock = Integer.parseInt(stockAvailable);

                    Artwork newArtwork = new Artwork(title, description, artist, price, imageUrl, stock);

                    artworks.add(newArtwork);

                    addArtworkToGallery(newArtwork);

                    addArtworkStage.close();

                    showAlert("Artwork Added", "Artwork '" + title + "' has been added successfully.");

                } catch (NumberFormatException ex) {

                    showAlert("Invalid Stock", "Please enter a valid number for stock.");

                }

            }

        });

        cancelButton.setOnAction(e -> addArtworkStage.close());

        formLayout.getChildren().addAll(titleField, descriptionField, artistField, priceField, imageUrlField, stockField, addButton, cancelButton);

        Scene scene = new Scene(formLayout, 400, 400);

        addArtworkStage.setScene(scene);

        addArtworkStage.show();

    }

    private void addArtworkToGallery(Artwork artwork) {

        artworkList.getChildren().add(createArtworkItem(artwork, artworks.indexOf(artwork)));

    }

    private void searchArtworks() {

        String query = searchField.getText().toLowerCase();

        List<Artwork> filteredArtworks = new ArrayList<>();

        for (Artwork artwork : artworks) {

            if (artwork.getTitle().toLowerCase().contains(query) || artwork.getArtist().toLowerCase().contains(query)) {

                filteredArtworks.add(artwork);

            }

        }

        refreshGallery(filteredArtworks);

    }

    private void clearSearch() {

        searchField.clear();

        refreshGallery(artworks);

    }

    private void refreshGallery(List<Artwork> artworksToShow) {

        artworkList.getChildren().clear();

        for (Artwork artwork : artworksToShow) {

            artworkList.getChildren().add(createArtworkItem(artwork, artworks.indexOf(artwork)));

        }

    }

    private HBox createArtworkItem(Artwork artwork, int index) {

        HBox artworkItem = new HBox(10);

        artworkItem.setPadding(new Insets(10));

        artworkItem.setStyle("-fx-border-color: #ccc; -fx-border-width: 1; -fx-background-color: #f9f9f9;");

        ImageView artworkImage;

        try {

            artworkImage = new ImageView(new Image(Objects.requireNonNull(getClass().getResource(artwork.getImageUrl())).toExternalForm()));

            artworkImage.setFitWidth(100);

            artworkImage.setFitHeight(100);

        } catch (NullPointerException e) {

            artworkImage = new ImageView(new Image(getClass().getResource("/path/to/placeholder.png").toExternalForm()));

            artworkImage.setFitWidth(100);

            artworkImage.setFitHeight(100);

        }

        VBox details = new VBox(5);

        Label titleLabel = new Label(artwork.getTitle());

        Label descriptionLabel = new Label(artwork.getDescription());

        Label artistLabel = new Label("Artist's Name: " + artwork.getArtist());

        Label priceLabel = new Label("Artwork Price: " + artwork.getPrice());

        Label stockLabel = new Label("Stock Available: " + artwork.getStockAvailable());

        details.getChildren().addAll(titleLabel, descriptionLabel, artistLabel, priceLabel, stockLabel);

        VBox buttonBox = new VBox(10);

        buttonBox.setAlignment(Pos.CENTER\_RIGHT);

        Button editButton = new Button("Edit");

        Button deleteButton = new Button("Delete");

        editButton.setOnAction(e -> editArtwork(index));

        deleteButton.setOnAction(e -> deleteArtwork(index));

        buttonBox.getChildren().addAll(editButton, deleteButton);

        artworkItem.getChildren().addAll(artworkImage, details, buttonBox);

        return artworkItem;

    }

    private void editArtwork(int index) {

        Artwork artworkToEdit = artworks.get(index);

        Stage editArtworkStage = new Stage();

        editArtworkStage.setTitle("Edit Artwork");

        VBox formLayout = new VBox(10);

        formLayout.setPadding(new Insets(20));

        formLayout.setStyle("-fx-background-color: Pink;");

        formLayout.setAlignment(Pos.CENTER);

        TextField titleField = new TextField(artworkToEdit.getTitle());

        TextField descriptionField = new TextField(artworkToEdit.getDescription());

        TextField artistField = new TextField(artworkToEdit.getArtist());

        TextField priceField = new TextField(artworkToEdit.getPrice());

        TextField imageUrlField = new TextField(artworkToEdit.getImageUrl());

        TextField stockField = new TextField(String.valueOf(artworkToEdit.getStockAvailable()));

        Button saveButton = new Button("Save Changes");

        Button cancelButton = new Button("Cancel");

        saveButton.setOnAction(e -> {

            String title = titleField.getText();

            String description = descriptionField.getText();

            String artist = artistField.getText();

            String price = priceField.getText();

            String imageUrl = imageUrlField.getText();

            String stockAvailable = stockField.getText();

            if (title.isEmpty() || description.isEmpty() || artist.isEmpty() || price.isEmpty() || imageUrl.isEmpty() || stockAvailable.isEmpty()) {

                showAlert("Input Error", "Please fill in all fields.");

            } else {

                try {

                    int stock = Integer.parseInt(stockAvailable);

                    artworkToEdit.setTitle(title);

                    artworkToEdit.setDescription(description);

                    artworkToEdit.setArtist(artist);

                    artworkToEdit.setPrice(price);

                    artworkToEdit.setImageUrl(imageUrl);

                    artworkToEdit.setStockAvailable(stock);

                    editArtworkStage.close();

                    refreshGallery(artworks);

                    showAlert("Artwork Updated", "Artwork '" + title + "' has been updated successfully.");

                } catch (NumberFormatException ex) {

                    showAlert("Invalid Stock", "Please enter a valid number for stock.");

                }

            }

        });

        cancelButton.setOnAction(e -> editArtworkStage.close());

        formLayout.getChildren().addAll(titleField, descriptionField, artistField, priceField, imageUrlField, stockField, saveButton, cancelButton);

        Scene scene = new Scene(formLayout, 400, 400);

        editArtworkStage.setScene(scene);

        editArtworkStage.show();

    }

    private void deleteArtwork(int index) {

        Artwork artworkToDelete = artworks.get(index);

        Alert confirmationAlert = new Alert(Alert.AlertType.CONFIRMATION);

        confirmationAlert.setTitle("Confirm Deletion");

        confirmationAlert.setHeaderText("Are you sure you want to delete this artwork?");

        confirmationAlert.setContentText("Title: " + artworkToDelete.getTitle());

        ButtonType yesButton = new ButtonType("Yes");

        ButtonType noButton = new ButtonType("No");

        confirmationAlert.getButtonTypes().setAll(yesButton, noButton);

        confirmationAlert.showAndWait().ifPresent(response -> {

            if (response == yesButton) {

                artworks.remove(artworkToDelete);

                refreshGallery(artworks);

                showAlert("Artwork Deleted", "Artwork '" + artworkToDelete.getTitle() + "' has been deleted.");

            }

        });

    }

    private void showAlert(String title, String content) {

        Alert alert = new Alert(Alert.AlertType.INFORMATION);

        alert.setTitle(title);

        alert.setHeaderText(null);

        alert.setContentText(content);

        alert.showAndWait();

    }

}

***4.ArtWork***

package com.example.demo8;

public class Artwork {

    private String title;

    private String description;

    private String artist;

    private String price;

    private String imageUrl;

    private int stockAvailable;

    public Artwork(String title, String description, String artist, String price, String imageUrl, int stockAvailable) {

        this.title = title;

        this.description = description;

        this.artist = artist;

        this.price = price;

        this.imageUrl = imageUrl;

        this.stockAvailable = stockAvailable;

    }

    public void setTitle(String title) {

        this.title = title;

    }

    public void setDescription(String description) {

        this.description = description;

    }

    public void setArtist(String artist) {

        this.artist = artist;

    }

    public void setPrice(String price) {

        this.price = price;

    }

    public void setImageUrl(String imageUrl) {

        this.imageUrl = imageUrl;

    }

    public void setStockAvailable(int stockAvailable) {

        this.stockAvailable = stockAvailable;

    }

    public String getTitle() {

        return title;

    }

    public int getStockAvailable() {

        return stockAvailable;

    }

    public String getDescription() {

        return description;

    }

    public String getArtist() {

        return artist;

    }

    public String getPrice() {

        return price;

    }

    public String getImageUrl() {

        return imageUrl;} }