**Assignment 04**

**Notepad**

**Assigned date :** 06-Dec-2024

**Due Date :**  20-Dec-2024

**Student’s Details :** Abeera (SP24-BSE-004**)**

Nafeesa Mushtaq (SP24-BSE-096)

**Here is the whole code of our notepad app:**

**Main.java**

*//package com.noteapp;*import javafx.application.Application;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.Pane;  
import javafx.scene.control.Label;  
import javafx.scene.control.Button;  
import javafx.scene.text.Font;  
import javafx.scene.effect.DropShadow;  
import javafx.scene.paint.Color;  
import javafx.stage.Stage;  
import javafx.scene.layout.StackPane;  
import javafx.scene.layout.VBox;  
import java.io.IOException;  
  
public class Main extends Application {  
  
 public static void main(String[] args) {  
  
 *launch*(args);  
 }  
  
 @Override  
 public void start(Stage stage) {  
 showWelcomePage(stage);  
 }  
  
 public void showWelcomePage(Stage stage) {  
 Image backgroundImg = new Image("img.jpg");  
 ImageView backgroundView = new ImageView(backgroundImg);  
 backgroundView.fitWidthProperty().bind(stage.widthProperty());  
 backgroundView.fitHeightProperty().bind(stage.heightProperty());  
  
 Label label = new Label("WELCOME\tTO\n NOTEHUB");  
 label.setFont(new Font("Britannic Bold", 60));  
 label.setTextFill(Color.*web*("#81D4D8"));  
  
 DropShadow dropShadow = new DropShadow();  
 dropShadow.setColor(Color.*DARKBLUE*);  
 dropShadow.setOffsetX(6);  
 dropShadow.setOffsetY(6);  
 label.setEffect(dropShadow);  
  
 label.layoutXProperty().bind(stage.widthProperty().subtract(label.widthProperty()).divide(2));  
 label.layoutYProperty().bind(stage.heightProperty().subtract(label.heightProperty()).divide(2));  
  
 Button nextButton = new Button("Next");  
 nextButton.setFont(new Font("Britannic Bold", 16));  
 nextButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
  
 nextButton.setOnMouseEntered(e -> nextButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: white; -fx-padding: 10px 20px;"));  
 nextButton.setOnMouseExited(e -> nextButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;"));  
 nextButton.setLayoutX(750);  
 nextButton.setLayoutY(430);  
 nextButton.setEffect(dropShadow);  
 nextButton.setOnAction(e -> showLoginPage(stage));  
  
 Pane root = new Pane();  
 root.getChildren().addAll(backgroundView, label, nextButton);  
  
 Scene scene = new Scene(root, 1000, 600); *// Set initial window size* stage.setTitle("WELCOME TO NOTEHUB");  
 stage.setScene(scene);  
 stage.show();  
 }  
  
 public void showLoginPage(Stage stage) {  
 Image backgroundImg = new Image("img.jpg");  
 ImageView backgroundView = new ImageView(backgroundImg);  
 backgroundView.fitWidthProperty().bind(stage.widthProperty());  
 backgroundView.fitHeightProperty().bind(stage.heightProperty());  
  
  
 VBox root = new VBox(20); *// Root layout with spacing between buttons* DropShadow dropShadow = new DropShadow();  
 dropShadow.setColor(Color.*DARKBLUE*);  
 dropShadow.setOffsetX(3);  
 dropShadow.setOffsetY(3);  
  
 Button loginButton = new Button("Login");  
 loginButton.setFont(new Font("Britannic Bold", 16));  
 loginButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 loginButton.setOnMouseEntered(e -> loginButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: white; -fx-padding: 10px 20px;"));  
 loginButton.setOnMouseExited(e -> loginButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;"));  
  
 loginButton.setEffect(dropShadow);  
  
 Button signupButton = new Button("Signup");  
 signupButton.setFont(new Font("Britannic Bold", 16));  
 signupButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 signupButton.setOnMouseEntered(e -> signupButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: white; -fx-padding: 10px 20px;"));  
 signupButton.setOnMouseExited(e -> signupButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;"));  
  
 signupButton.setEffect(dropShadow);  
  
 Button backButton = new Button("Back");  
 backButton.setFont(new Font("Britannic Bold", 16));  
 backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 backButton.setOnMouseEntered(e -> backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: white; -fx-padding: 10px 20px;"));  
 backButton.setOnMouseExited(e -> backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;"));  
  
 backButton.setEffect(dropShadow);  
  
 backButton.setOnAction(e -> showWelcomePage(stage));  
 loginButton.setOnAction(e -> showLoginForm(stage));  
 signupButton.setOnAction(e -> showSignupForm(stage));  
  
 root.getChildren().addAll(loginButton, signupButton,backButton);  
 root.setAlignment(Pos.*CENTER*);  
 StackPane mainRoot = new StackPane();  
 mainRoot.getChildren().addAll(backgroundView, root);  
 Scene loginScene = new Scene(mainRoot,1000, 600);  
  
 stage.setScene(loginScene);  
 stage.show();  
 }  
 private void showLoginForm(Stage stage) {  
 LoginScreen loginScreen = new LoginScreen();  
 loginScreen.show(stage);  
 }  
  
 private void showSignupForm(Stage stage) {  
 SignupScreen signupScreen = new SignupScreen();  
 signupScreen.show(stage);  
 }  
}

**LoginScreen.java**

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.StackPane;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
import javafx.geometry.Pos;  
import javafx.scene.text.Font;  
import javafx.scene.paint.Color;  
  
import java.io.\*;  
  
public class LoginScreen {  
  
 public void show(Stage stage) {  
 Image backgroundImg = new Image("img.jpg");  
 ImageView backgroundView = new ImageView(backgroundImg);  
 backgroundView.fitWidthProperty().bind(stage.widthProperty());  
 backgroundView.fitHeightProperty().bind(stage.heightProperty());  
 VBox root = new VBox(20);  
  
 Label label = new Label("Login Page");  
 label.setFont(new Font("Britannic Bold", 50));  
 label.setTextFill(Color.*web*("#81D4D8"));  
  
 DropShadow dropShadow = new DropShadow();  
 dropShadow.setColor(Color.*DARKBLUE*);  
 dropShadow.setOffsetX(6);  
 dropShadow.setOffsetY(6);  
 label.setEffect(dropShadow);  
  
 *// Input fields for username and password* TextField usernameField = new TextField();  
 usernameField.setPromptText("Enter Username");  
 usernameField.setMaxWidth(300);  
  
 PasswordField passwordField = new PasswordField();  
 passwordField.setPromptText("Enter Password");  
 passwordField.setMaxWidth(300);  
  
 *// Login Button* Button loginButton = new Button("Login");  
 loginButton.setFont(new Font("Britannic Bold", 16));  
 loginButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 loginButton.setOnMouseEntered(e -> loginButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: white; -fx-padding: 10px 20px;"));  
 loginButton.setOnMouseExited(e -> loginButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;"));  
  
 loginButton.setEffect(dropShadow);  
  
 *// Back Button* Button backButton = new Button("Back");  
 backButton.setFont(new Font("Britannic Bold", 16));  
 backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 backButton.setOnMouseEntered(e -> backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: white; -fx-padding: 10px 20px;"));  
 backButton.setOnMouseExited(e -> backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;"));  
 backButton.setEffect(dropShadow);  
 backButton.setOnAction(e -> new Main().showLoginPage(stage));  
  
 *// Login button action* loginButton.setOnAction(e -> {  
 String username = usernameField.getText().trim();  
 String password = passwordField.getText().trim();  
  
 if (username.isEmpty() || password.isEmpty()) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "Username and Password cannot be empty!");  
 return;  
 }  
  
 if (isExistingUser(username, password)) {  
 showAlert(Alert.AlertType.*ERROR*, "Login Error", "Username or Password already exists!");  
 } else {  
 createUserFiles(username, password);  
 new NotesScreen().show(stage, username); *// Open notes screen* }  
 });  
  
 root.getChildren().addAll(label, usernameField, passwordField, loginButton, backButton);  
 root.setAlignment(Pos.*CENTER*);  
  
 StackPane mainroot = new StackPane();  
 mainroot.getChildren().addAll(backgroundView, root);  
  
 Scene scene = new Scene(mainroot, 1000, 600);  
 stage.setScene(scene);  
 stage.show();  
 }  
  
 private boolean isExistingUser(String username, String password) {  
 File userFolder = new File("users/" + username);  
 File credentialsFile = new File(userFolder, "credentials.txt");  
  
 if (credentialsFile.exists()) {  
 try (BufferedReader reader = new BufferedReader(new FileReader(credentialsFile))) {  
 String savedUsername = reader.readLine().split(": ")[0];  
 String savedPassword = reader.readLine().split(": ")[0];  
  
 return username.equals(savedUsername) && password.equals(savedPassword);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 return false;  
 }  
  
 private void createUserFiles(String username, String password) {  
 try {  
 *// Create a folder for the user* File userFolder = new File("users/" + username);  
 if (!userFolder.exists()) {  
 userFolder.mkdirs();  
 }  
  
 *// Save the user's credentials* File credentialsFile = new File(userFolder, "credentials.txt");  
 try (FileWriter writer = new FileWriter(credentialsFile)) {  
 writer.write(username + "\n");  
 writer.write(password + "\n");  
 }  
  
 System.*out*.println("User files created for: " + username);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
  
  
 private void showAlert(Alert.AlertType alertType, String title, String message) {  
 Alert alert = new Alert(alertType);  
 alert.setTitle(title);  
 alert.setHeaderText(null);  
 alert.setContentText(message);  
 alert.showAndWait();  
 }  
}

**SignUpScreen.java**

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.StackPane;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
import javafx.geometry.Pos;  
import javafx.scene.text.Font;  
import javafx.scene.paint.Color;  
  
import java.io.\*;  
import java.nio.file.\*;  
  
public class SignupScreen {  
 public void show(Stage stage) {  
  
 Image backgroundImg = new Image("img.jpg");  
 ImageView backgroundView = new ImageView(backgroundImg);  
 backgroundView.fitWidthProperty().bind(stage.widthProperty());  
 backgroundView.fitHeightProperty().bind(stage.heightProperty());  
 VBox root = new VBox(20);  
  
 Label titleLabel = new Label("Signup Page");  
 titleLabel.setFont(new Font("Britannic Bold", 40));  
 titleLabel.setTextFill(Color.*SKYBLUE*);  
  
 DropShadow dropShadow = new DropShadow();  
 dropShadow.setColor(Color.*DARKBLUE*);  
 dropShadow.setOffsetX(3);  
 dropShadow.setOffsetY(3);  
 titleLabel.setEffect(dropShadow);  
  
 TextField usernameField = new TextField();  
 usernameField.setPromptText("Enter your username");  
 usernameField.setMaxWidth(300);  
  
 PasswordField passwordField = new PasswordField();  
 passwordField.setPromptText("Enter your password");  
 passwordField.setMaxWidth(300);  
  
 Button signupButton = new Button("Signup");  
 signupButton.setFont(new Font("Britannic Bold", 16));  
 signupButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 signupButton.setOnMouseEntered(e -> signupButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: white; -fx-padding: 10px 20px;"));  
 signupButton.setOnMouseExited(e -> signupButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;"));  
 signupButton.setEffect(dropShadow);  
 signupButton.setOnAction(e -> {  
 String username = usernameField.getText().trim();  
 String password = passwordField.getText().trim();  
  
 if (username.isEmpty() || password.isEmpty()) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "Username and password cannot be empty!");  
 return;  
 }  
  
 handleSignup(stage, username, password);  
 });  
  
 Button backButton = new Button("Back");  
 backButton.setFont(new Font("Britannic Bold", 16));  
 backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 backButton.setOnMouseEntered(e -> backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: white; -fx-padding: 10px 20px;"));  
 backButton.setOnMouseExited(e -> backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;"));  
 backButton.setOnAction(e -> new Main().showLoginPage(stage));  
 backButton.setEffect(dropShadow);  
  
 root.getChildren().addAll(titleLabel, usernameField, passwordField, signupButton, backButton);  
 root.setAlignment(Pos.*CENTER*);  
  
  
 StackPane stackPane = new StackPane();  
 stackPane.getChildren().addAll(backgroundView, root);  
  
 Scene scene = new Scene(stackPane, 1000, 600);  
 stage.setScene(scene);  
 stage.show();  
 }  
  
 private void handleSignup(Stage stage, String username, String password) {  
 Path userDirectory = Paths.*get*("users");  
  
 try {  
 if (!Files.*exists*(userDirectory)) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "No users exist in the system!");  
 return;  
 }  
  
 boolean userExists = false;  
 boolean credentialsMatch = false;  
  
 *// Check all user folders* for (File userFolder : userDirectory.toFile().listFiles()) {  
 if (userFolder.isDirectory()) {  
 Path credentialsFile = userFolder.toPath().resolve("credentials.txt");  
 if (Files.*exists*(credentialsFile)) {  
 try (BufferedReader reader = new BufferedReader(new FileReader(credentialsFile.toFile()))) {  
 String savedUsername = reader.readLine();  
 String savedPassword = reader.readLine();  
  
 if (username.equals(savedUsername)) {  
 userExists = true;  
 if (password.equals(savedPassword)) {  
 credentialsMatch = true;  
 break;  
 }  
 }  
 }  
 }  
 }  
 }  
  
 if (credentialsMatch) {  
 new NotesScreen().show(stage, username); *// Redirect to saved notes* } else if (userExists) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "Password is incorrect for the existing username!");  
 } else {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "User with this username does not exist!");  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 showAlert(Alert.AlertType.*ERROR*, "Error", "An error occurred during signup.");  
 }  
 }  
  
 private void showAlert(Alert.AlertType alertType, String title, String message) {  
 Alert alert = new Alert(alertType);  
 alert.setTitle(title);  
 alert.setHeaderText(null);  
 alert.setContentText(message);  
 alert.showAndWait();  
 }  
}

**NoteScreen.java**

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.HBox;  
import javafx.scene.layout.StackPane;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
import javafx.geometry.Pos;  
import javafx.scene.text.Font;  
import javafx.scene.paint.Color;  
  
import java.io.\*;  
import java.nio.file.\*;  
import java.util.List;  
import java.util.stream.Collectors;  
  
public class NotesScreen {  
 public void show(Stage stage, String username) {  
 Image backgroundImg = new Image("img.jpg");  
 ImageView backgroundView = new ImageView(backgroundImg);  
 backgroundView.fitWidthProperty().bind(stage.widthProperty());  
 backgroundView.fitHeightProperty().bind(stage.heightProperty());  
  
 VBox root = new VBox(20);  
  
 *// Welcome Label* Label welcomeLabel = new Label("Welcome " + username + "!\n\n");  
 welcomeLabel.setFont(new Font("Britannic Bold", 60));  
 welcomeLabel.setTextFill(Color.*web*("#81D4D8"));  
 DropShadow dropShadow = new DropShadow();  
 dropShadow.setColor(Color.*DARKBLUE*);  
 dropShadow.setOffsetX(6);  
 dropShadow.setOffsetY(6);  
 welcomeLabel.setEffect(dropShadow);  
  
 *// Button to add a new note* Button addNoteButton = new Button("Add Note");  
 addNoteButton.setFont(new Font("Britannic Bold", 16));  
 addNoteButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 addNoteButton.setOnMouseEntered(e -> addNoteButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: white; -fx-padding: 10px 20px;"));  
 addNoteButton.setOnMouseExited(e -> addNoteButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;"));  
 addNoteButton.setOnAction(e -> new AddNotesScreen().show(stage, username));  
  
 *// Logout Button* Button logoutButton = new Button("Logout");  
 logoutButton.setFont(new Font("Britannic Bold", 16));  
 logoutButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 logoutButton.setOnMouseEntered(e -> logoutButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: white; -fx-padding: 10px 20px;"));  
 logoutButton.setOnMouseExited(e -> logoutButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;"));  
 logoutButton.setOnAction(e -> {  
 Alert logoutConfirmation = new Alert(Alert.AlertType.*CONFIRMATION*);  
 logoutConfirmation.setTitle("Logout Confirmation");  
 logoutConfirmation.setHeaderText("Are you sure you want to logout?");  
 logoutConfirmation.setContentText("Click OK to logout or Cancel to stay.");  
 logoutConfirmation.showAndWait().ifPresent(response -> {  
 if (response == ButtonType.*OK*) {  
 new Main().showWelcomePage(stage); *// Go back to the welcome page* }  
 });  
 });  
  
 *// View Notes Button* Button viewButton = new Button("View Notes");  
 viewButton.setFont(new Font("Britannic Bold", 16));  
 viewButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 viewButton.setOnMouseEntered(e -> viewButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: white; -fx-padding: 10px 20px;"));  
 viewButton.setOnMouseExited(e -> viewButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;"));  
 viewButton.setOnAction(e -> loadUserNotes(stage, username)); *// Load user's notes* root.getChildren().addAll(welcomeLabel, addNoteButton, logoutButton, viewButton);  
 root.setAlignment(Pos.*CENTER*);  
 addNoteButton.setEffect(dropShadow);  
 viewButton.setEffect(dropShadow);  
 logoutButton.setEffect(dropShadow);  
 StackPane mainRoot = new StackPane();  
 mainRoot.getChildren().addAll(backgroundView,root);  
  
 Scene scene = new Scene(mainRoot, 1000, 600);  
 stage.setScene(scene);  
 stage.show();  
 }  
  
 private void loadUserNotes(Stage stage, String username) {  
 Image backgroundImg = new Image("bg.jpg");  
 ImageView backgroundView = new ImageView(backgroundImg);  
 backgroundView.fitWidthProperty().bind(stage.widthProperty());  
 backgroundView.fitHeightProperty().bind(stage.heightProperty());  
  
 Path userFolder = Paths.*get*("users/" + username);  
 if (Files.*exists*(userFolder)) {  
 try {  
 *// Get all .txt files in the user's folder, excluding "credentials.txt"* List<Path> noteFiles = Files.*walk*(userFolder)  
 .filter(path -> path.toString().endsWith(".txt") && !path.getFileName().toString().equals("credentials.txt"))  
 .collect(Collectors.*toList*());  
  
 if (noteFiles.isEmpty()) {  
 showAlert(Alert.AlertType.*WARNING*, "No Notes Found", "You don't have any notes yet. Add some!");  
 return;  
 }  
  
 *// Create a VBox to display the list of note files* VBox notesListVBox = new VBox(10);  
 notesListVBox.setAlignment(Pos.*CENTER*);  
  
 *// Add a row with a note button and a delete button for each note file* for (Path noteFile : noteFiles) {  
 HBox noteRow = new HBox(10); *// Horizontal layout for note button and delete button* noteRow.setAlignment(Pos.*CENTER*);  
  
 *// Note button* Button noteButton = new Button(noteFile.getFileName().toString());  
 noteButton.setFont(new Font("Arial", 16));  
 noteButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 noteButton.setOnAction(e -> viewNoteContent(stage, noteFile));  
  
 *// Delete button* Image deleteImage = new Image(getClass().getResourceAsStream("/del.png")); *// Ensure the image is in the `src` directory or resources folder* ImageView deleteImageView = new ImageView(deleteImage);  
 deleteImageView.setFitWidth(60); *// Set desired width* deleteImageView.setFitHeight(55); *// Set desired height* Button deleteButton = new Button();  
 deleteButton.setGraphic(deleteImageView);  
 deleteButton.setStyle("-fx-background-color: transparent; -fx-padding: 10px;");  
 deleteButton.setOnAction(e -> {  
 Alert confirmDelete = new Alert(Alert.AlertType.*CONFIRMATION*);  
 confirmDelete.setTitle("Delete Confirmation");  
 confirmDelete.setHeaderText("Are you sure you want to delete this note?");  
 confirmDelete.setContentText("This action cannot be undone.");  
 confirmDelete.showAndWait().ifPresent(response -> {  
 if (response == ButtonType.*OK*) {  
 try {  
 Files.*delete*(noteFile); *// Delete the file* showAlert(Alert.AlertType.*INFORMATION*, "Success", "Note deleted successfully.");  
 loadUserNotes(stage, username); *// Reload the notes list* } catch (IOException ex) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "An error occurred while deleting the note.");  
 }  
 }  
 });  
 });  
  
 *// Edit button to switch to edit mode* Image editImage = new Image(getClass().getResourceAsStream("/edit1.png"));  
 ImageView editImageView = new ImageView(editImage);  
 editImageView.setFitWidth(50); *// Set desired width* editImageView.setFitHeight(50); *// Set desired height* Button editButton = new Button();  
 editButton.setGraphic(editImageView);  
 editButton.setOnAction(e -> {  
 String content = null;  
 try {  
 content = new String(Files.*readAllBytes*(noteFile));  
 } catch (IOException ex) {  
 throw new RuntimeException(ex);  
 }  
 *// Switch to an editable view* VBox editRoot = new VBox(20);  
 editRoot.setAlignment(Pos.*CENTER*);  
  
 *// Create a TextArea for editing the content* TextArea contentArea = new TextArea(content);  
 contentArea.setFont(new Font("Arial", 16));  
 contentArea.setMaxWidth(600);  
 contentArea.setWrapText(true);  
  
 *// Save button to save the edited content* Button saveButton = new Button("Save Note");  
 saveButton.setFont(new Font("Britannic Bold", 16));  
 saveButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 saveButton.setOnAction(saveEvent -> {  
 try {  
 *// Save the edited content back to the file* Files.*write*(noteFile, contentArea.getText().getBytes());  
 showAlert(Alert.AlertType.*INFORMATION*, "Success", "Note saved successfully.");  
 *// Reload the original view with updated content* viewNoteContent(stage, noteFile);  
 } catch (IOException ex) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "An error occurred while saving the note.");  
 }  
 });  
  
 *// Back button to cancel editing and return to the view mode* Button cancelButton = new Button("Cancel Editing");  
 cancelButton.setFont(new Font("Britannic Bold", 16));  
 cancelButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 cancelButton.setOnAction(cancelEvent -> {  
 *// Return to the original view mode without saving* viewNoteContent(stage, noteFile);  
 });  
  
 editRoot.getChildren().addAll(contentArea, saveButton, cancelButton);  
  
 StackPane s = new StackPane();  
 s.getChildren().addAll(backgroundView, editRoot); *// Add background image* Scene editScene = new Scene(s, 1000, 600);  
 stage.setScene(editScene);  
 stage.show();  
 });  
  
 *// Add the note button and delete button to the row* noteRow.getChildren().addAll(noteButton, editButton, deleteButton);  
 notesListVBox.getChildren().add(noteRow);  
 }  
  
 *// Create a scrollable pane to display the notes list* ScrollPane scrollPane = new ScrollPane();  
 scrollPane.setContent(notesListVBox);  
 scrollPane.setFitToWidth(true);  
  
 *// Create the scene for viewing the list of notes* VBox notesRoot = new VBox(20);  
 notesRoot.setAlignment(Pos.*CENTER*); *// Align buttons and content in the center  
  
 // Back button to return to main screen* Button backButton = new Button("Back to Main");  
 backButton.setFont(new Font("Britannic Bold", 16));  
 backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 backButton.setOnAction(e -> new NotesScreen().show(stage, username)); *// Go back to Notes Screen  
  
 // Add scrollPane and backButton to the notesRoot* notesRoot.getChildren().addAll(scrollPane, backButton);  
  
 StackPane noteRoot = new StackPane();  
 noteRoot.getChildren().addAll(backgroundView, notesRoot); *// Background with content* Scene notesScene = new Scene(noteRoot, 1000, 600);  
 stage.setScene(notesScene);  
 stage.show();  
  
 } catch (IOException e) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "An error occurred while loading the notes.");  
 }  
 } else {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "User folder does not exist.");  
 }  
 }  
  
 *// Method to view the content of a selected note* private void viewNoteContent(Stage stage, Path noteFile) {  
 Image backgroundImg = new Image("img.jpg");  
 ImageView backgroundView = new ImageView(backgroundImg);  
 backgroundView.fitWidthProperty().bind(stage.widthProperty());  
 backgroundView.fitHeightProperty().bind(stage.heightProperty());  
  
 try {  
 *// Read the content of the selected not* String content = new String(Files.*readAllBytes*(noteFile));  
 *// Create a VBox for the initial view* VBox viewRoot = new VBox(20);  
 viewRoot.setAlignment(Pos.*CENTER*);  
  
 *// Display the content of the note using a Label* Label contentLabel = new Label(content);  
 contentLabel.setFont(new Font("Arial", 16));  
 contentLabel.setWrapText(true);  
  
 *// Back button to return to the notes list* Button backButton = new Button("Back to Notes List");  
 backButton.setFont(new Font("Britannic Bold", 16));  
 backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 backButton.setOnAction(e -> loadUserNotes(stage, noteFile.getParent().getFileName().toString()));  
  
 viewRoot.getChildren().addAll(contentLabel, backButton);  
 StackPane mainroot = new StackPane();  
 mainroot.getChildren().addAll(backgroundView, viewRoot);  
  
 Scene viewScene = new Scene(mainroot, 1000, 600);  
  
 *// Set the scene to the stage* stage.setScene(viewScene);  
 stage.show();  
  
 *// Event handler for the Edit button* } catch (IOException e) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "An error occurred while loading the note content.");  
 }  
 }  
  
 *// Method to display alerts* private void showAlert(Alert.AlertType alertType, String title, String message) {  
 Alert alert = new Alert(alertType);  
 alert.setTitle(title);  
 alert.setHeaderText(null);  
 alert.setContentText(message);  
 alert.showAndWait();  
 }  
}

**AddNotesScreen.java**

import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextArea;  
import javafx.scene.control.TextField;  
import javafx.scene.effect.DropShadow;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.StackPane;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
import javafx.geometry.Pos;  
import javafx.scene.text.Font;  
import javafx.scene.paint.Color;  
  
import java.io.\*;  
import java.nio.file.Files;  
import java.nio.file.Path;  
import java.nio.file.Paths;  
import java.util.UUID;  
  
public class AddNotesScreen {  
 public void show(Stage stage, String username) {  
 Image backgroundImg = new Image("bg.jpg");  
 ImageView backgroundView = new ImageView(backgroundImg);  
 backgroundView.fitWidthProperty().bind(stage.widthProperty());  
 backgroundView.fitHeightProperty().bind(stage.heightProperty());  
 VBox root = new VBox(20);  
  
 Label titleLabel = new Label("Add Notes for " + username);  
 titleLabel.setFont(new Font("Britannic Bold", 55));  
 titleLabel.setTextFill(Color.*DARKBLUE*);  
  
 DropShadow dropShadow = new DropShadow();  
 dropShadow.setColor(Color.*SKYBLUE*);  
 dropShadow.setOffsetX(6);  
 dropShadow.setOffsetY(6);  
 titleLabel.setEffect(dropShadow);  
  
 Label Ntitle = new Label("Note's Title");  
 Ntitle.setFont(new Font("Britannic Bold", 30));  
 Ntitle.setTextFill(Color.*DARKBLUE*);  
  
 TextField noteTitleField = new TextField();  
 noteTitleField.setPromptText("Enter note title...");  
 noteTitleField.setMaxWidth(600);  
  
 Label content = new Label("Note's Content");  
 content.setFont(new Font("Britannic Bold", 30));  
 content.setTextFill(Color.*DARKBLUE*);  
  
 TextArea noteArea = new TextArea();  
 noteArea.setPromptText("Write your note here...");  
 noteArea.setWrapText(true);  
 noteArea.setMaxWidth(600);  
  
 Button saveButton = new Button("Save Note");  
 saveButton.setFont(new Font("Britannic Bold", 16));  
 saveButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 saveButton.setOnMouseEntered(e -> saveButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: white; -fx-padding: 10px 20px;"));  
 saveButton.setOnMouseExited(e -> saveButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;"));  
  
 saveButton.setEffect(dropShadow);  
  
 Button backButton = new Button("Back");  
 backButton.setFont(new Font("Britannic Bold", 16));  
 backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;");  
 backButton.setOnMouseEntered(e -> backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: white; -fx-padding: 10px 20px;"));  
 backButton.setOnMouseExited(e -> backButton.setStyle("-fx-background-color: darkblue; -fx-text-fill: skyblue; -fx-padding: 10px 20px;"));  
  
 *// Save note to user's file* saveButton.setOnAction(e -> {  
 String title = noteTitleField.getText();  
 String note = noteArea.getText();  
 if (!title.isEmpty() && !note.isEmpty()) {  
 saveNoteForUser(username, title, note);  
 noteArea.clear();  
 noteTitleField.clear();  
 } else {  
 System.*out*.println("Title and note cannot be empty!");  
 }  
 });  
  
 *// Go back to the NotesScreen* backButton.setEffect(dropShadow);  
 backButton.setOnAction(e -> new NotesScreen().show(stage, username));  
  
 root.getChildren().addAll(titleLabel,Ntitle, noteTitleField,content, noteArea, saveButton, backButton);  
 root.setAlignment(Pos.*CENTER*);  
  
 StackPane mainroot = new StackPane();  
 mainroot.getChildren().addAll(backgroundView, root);  
  
 Scene scene = new Scene(mainroot, 1000, 600);  
 stage.setScene(scene);  
 stage.show();  
 }  
  
 private void saveNoteForUser(String username, String title, String note) {  
 Path userFolder = Paths.*get*("users/" + username);  
 if (!Files.*exists*(userFolder)) {  
 System.*out*.println("User folder does not exist!");  
 return;  
 }  
  
 *// Generate a filename using the title* String noteFileName = title.replaceAll("\\s+", "\_") + ".txt";  
 File noteFile = new File(userFolder.toFile(), noteFileName);  
  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(noteFile))) {  
 writer.write(note);  
 System.*out*.println("Note saved successfully in file: " + noteFileName);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

**The End**