**Name= Parmjeet Singh**

**Student ID= 23099980**

Main

// LoginScene.java

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.layout.\*;

import javafx.geometry.\*;

import javafx.stage.Stage;

public class Main extends Application {

public static Scene getLoginScene(Stage stage) {

Label title = new Label("Login Page");

title.setStyle("-fx-font-size: 20px; -fx-font-weight: bold;");

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

TextField usernameField = new TextField();

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

PasswordField passwordField = new PasswordField();

Button loginButton = new Button("Login");

Label errorLabel = new Label();

loginButton.setOnAction(e -> {

if (usernameField.getText().equals("Parmjeet") && passwordField.getText().equals("Bhathal")) {

stage.setScene(CrudScene.getCrudScene(stage));

} else {

errorLabel.setText("Invalid credentials.");

}

});

VBox layout = new VBox(10, title, usernameLabel, usernameField, passwordLabel, passwordField, loginButton, errorLabel);

layout.setAlignment(Pos.CENTER);

layout.setPadding(new Insets(20));

return new Scene(layout, 400, 300);

}

@Override

public void start(Stage primaryStage) {

primaryStage.setTitle("JavaFX Application");

primaryStage.setScene(getLoginScene(primaryStage));

primaryStage.show();

}

public static void main(String[] args) {

launch(args);

}

}

Crudscene

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.layout.\*;

import javafx.geometry.\*;

import javafx.collections.\*;

import javafx.stage.Stage;

import javafx.scene.control.cell.PropertyValueFactory;

public class CrudScene {

private static ObservableList<Person> personList = FXCollections.observableArrayList();

private static TableView<Person> table;

private static TextField nameField;

private static TextField emailField;

public static Scene getCrudScene(Stage stage) {

Label title = new Label("CRUD Operations");

title.setStyle("-fx-font-size: 20px; -fx-font-weight: bold;");

nameField = new TextField();

nameField.setPromptText("Name");

emailField = new TextField();

emailField.setPromptText("Email");

Button addButton = new Button("Add");

Button updateButton = new Button("Update");

Button deleteButton = new Button("Delete");

Button clearButton = new Button("Clear");

table = new TableView<>();

table.setItems(personList);

TableColumn<Person, String> nameCol = new TableColumn<>("Name");

nameCol.setCellValueFactory(new PropertyValueFactory<>("name"));

nameCol.setPrefWidth(200);

TableColumn<Person, String> emailCol = new TableColumn<>("Email");

emailCol.setCellValueFactory(new PropertyValueFactory<>("email"));

emailCol.setPrefWidth(250);

table.getColumns().addAll(nameCol, emailCol);

// Add some sample data

personList.addAll(

new Person("basanta", "Aujla467@email.com"),

new Person("ghugu", "Reigns1316@email.com"),

new Person("ali", "Lesnar12@email.com")

);

// Event handlers

addButton.setOnAction(e -> addPerson());

updateButton.setOnAction(e -> updatePerson());

deleteButton.setOnAction(e -> deletePerson());

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

// Table selection handler

table.getSelectionModel().selectedItemProperty().addListener((obs, oldSelection, newSelection) -> {

if (newSelection != null) {

nameField.setText(newSelection.getName());

emailField.setText(newSelection.getEmail());

}

});

HBox inputBox = new HBox(10, nameField, emailField, addButton, updateButton, deleteButton, clearButton);

inputBox.setAlignment(Pos.CENTER);

VBox layout = new VBox(10, title, inputBox, table);

layout.setPadding(new Insets(20));

layout.setAlignment(Pos.CENTER);

return new Scene(layout, 700, 500);

}

private static void addPerson() {

String name = nameField.getText().trim();

String email = emailField.getText().trim();

if (!name.isEmpty() && !email.isEmpty()) {

personList.add(new Person(name, email));

clearFields();

} else {

showAlert("Please enter both name and email.");

}

}

private static void updatePerson() {

Person selected = table.getSelectionModel().getSelectedItem();

if (selected != null) {

String name = nameField.getText().trim();

String email = emailField.getText().trim();

if (!name.isEmpty() && !email.isEmpty()) {

int index = personList.indexOf(selected);

personList.set(index, new Person(name, email));

clearFields();

} else {

showAlert("Please enter both name and email.");

}

} else {

showAlert("Please select a person to update.");

}

}

private static void deletePerson() {

Person selected = table.getSelectionModel().getSelectedItem();

if (selected != null) {

personList.remove(selected);

clearFields();

} else {

showAlert("Please select a person to delete.");

}

}

private static void clearFields() {

nameField.clear();

emailField.clear();

table.getSelectionModel().clearSelection();

}

private static void showAlert(String message) {

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

alert.setTitle("Warning");

alert.setHeaderText(null);

alert.setContentText(message);

alert.showAndWait();

}

}

Person.java

public class Person {

private String name;

private String email;

public Person(String name, String email) {

this.name = name;

this.email = email;

}

public String getName() { return name; }

public String getEmail() { return email; }

}

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.