**Task1:**

**Main:**

package application;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.stage.Stage;

import javafx.scene.Parent;

import javafx.scene.Scene;

public class Main extends Application {

@Override

public void start(Stage stage) {

try {

Parent root = FXMLLoader.load(getClass().getResource("Login.fxml"));

Scene scene = new Scene(root);

stage.setScene(scene);

stage.show();

} catch(Exception e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

launch(args);

}

}

**ControllerClass:**

package application;

import java.io.IOException;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.fxml.FXMLLoader;

import javafx.stage.Stage;

import javafx.scene.Scene;

import javafx.scene.\*;

import javafx.scene.Parent;

import javafx.scene.layout.BorderPane;

import javafx.scene.Node;

public class SceneController {

Stage stage = new Stage();

public void LoginWindow(ActionEvent event) throws IOException {

Parent root = FXMLLoader.load(getClass().getResource("Login.fxml"));

Scene scene = new Scene(root);

stage.setScene(scene);

stage.show();

}

public void RegisterWindow(ActionEvent event) throws IOException {

Parent root = FXMLLoader.load(getClass().getResource("Register.fxml"));

Scene scene = new Scene(root);

stage.setScene(scene);

stage.show();

}

public void Logout(ActionEvent event) throws IOException {

Parent root = FXMLLoader.load(getClass().getResource("Logout.fxml"));

Scene scene = new Scene(root);

stage.setScene(scene);

stage.show();

}

}

**Login fxml:**

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.layout.AnchorPane?>

<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/20.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.SceneController">

<children>

<Button layoutX="190.0" layoutY="273.0" mnemonicParsing="false" onAction="#Register" prefHeight="25.0" prefWidth="85.0" text="Register" />

<Label layoutX="260.0" layoutY="48.0" prefHeight="55.0" prefWidth="110.0" text="Login ">

<font>

<Font name="Tw Cen MT" size="38.0" />

</font>

</Label>

<Label layoutX="138.0" layoutY="131.0" prefHeight="41.0" prefWidth="104.0" text="Username">

<font>

<Font name="Tw Cen MT" size="23.0" />

</font>

</Label>

<TextField layoutX="138.0" layoutY="172.0" />

<Label layoutX="353.0" layoutY="131.0" prefHeight="41.0" prefWidth="104.0" text="Password">

<font>

<Font name="Tw Cen MT" size="23.0" />

</font>

</Label>

<TextField layoutX="353.0" layoutY="172.0" />

<Button layoutX="315.0" layoutY="273.0" mnemonicParsing="false" onAction="#Logout" prefHeight="25.0" prefWidth="85.0" text="Login" />

</children>

</AnchorPane>

**Logout fxml:**

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.layout.AnchorPane?>

<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/20.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.SceneController">

<children>

<Label layoutX="139.0" layoutY="111.0" prefHeight="40.0" prefWidth="349.0" text="Congrats You Have Logged In">

<font>

<Font name="Tw Cen MT" size="27.0" />

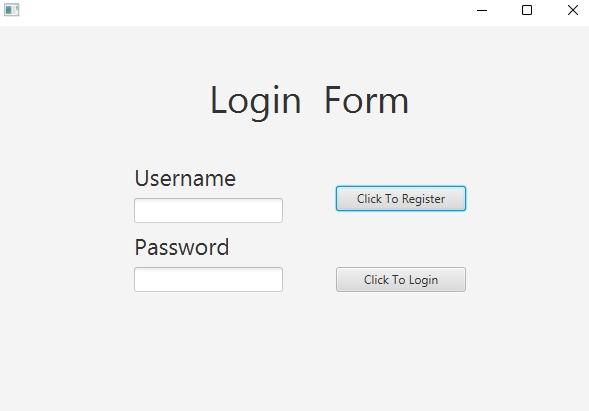
</font>

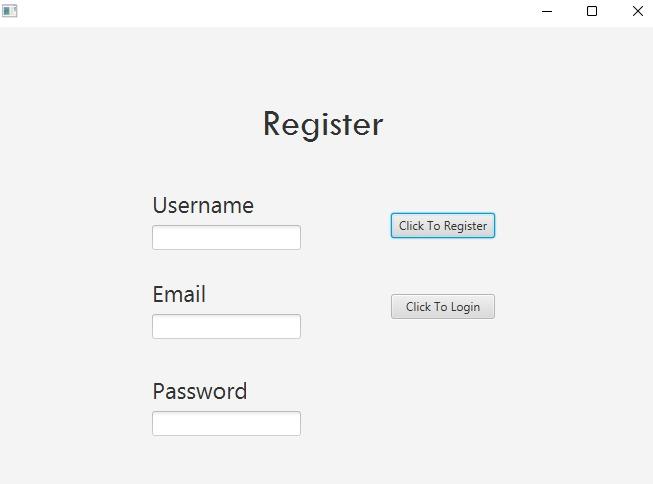
</Label>

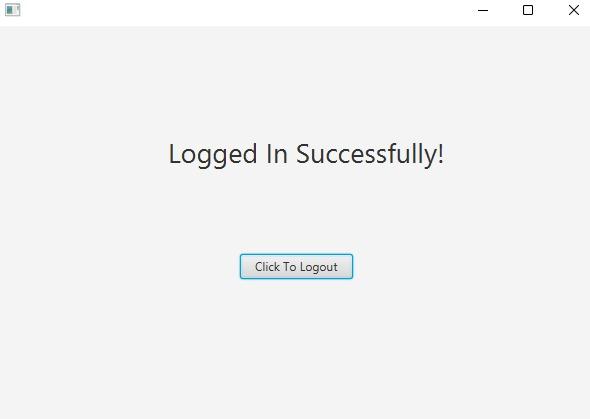
<Button layoutX="258.0" layoutY="231.0" mnemonicParsing="false" onAction="#Login" prefHeight="25.0" prefWidth="85.0" text="Logout" />

</children>

</AnchorPane>







**Task2:**

**Main:**

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.stage.Stage;

public class Main extends Application {

@Override

public void start(Stage primaryStage) throws Exception {

Parent root = FXMLLoader.load(getClass().getResource("LibraryCatalog.fxml"));

primaryStage.setTitle("Library Catalog System");

primaryStage.setScene(new Scene(root, 800, 600));

primaryStage.show();

}

public static void main(String[] args) {

launch(args);

}

}

**Controller:**

import javafx.fxml.FXML;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

public class LibraryCatalogController {

@FXML

private TextField searchField;

@FXML

private ComboBox<String> searchCategory;

@FXML

private TableView<Book> tableView;

@FXML

private TableColumn<Book, String> titleColumn;

@FXML

private TableColumn<Book, String> authorColumn;

@FXML

private TableColumn<Book, String> genreColumn;

@FXML

private TableColumn<Book, Integer> yearColumn;

public void initialize() {

titleColumn.setCellValueFactory(new PropertyValueFactory<>("title"));

authorColumn.setCellValueFactory(new PropertyValueFactory<>("author"));

genreColumn.setCellValueFactory(new PropertyValueFactory<>("genre"));

yearColumn.setCellValueFactory(new PropertyValueFactory<>("year"));

searchCategory.getItems().addAll("Title", "Author", "Genre");

searchCategory.setValue("Title");

}

@FXML

private void searchButtonClicked() {

}

}

**Book Class:**

public class Book {

private String title;

private String author;

private String genre;

private int year;

public Book(String title, String author, String genre, int year) {

this.title = title;

this.author = author;

this.genre = genre;

this.year = year;

}

}

**Task3:**

**Main:**

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.stage.Stage;

public class Main extends Application {

@Override

public void start(Stage primaryStage) throws Exception {

Parent root = FXMLLoader.load(getClass().getResource("TodoList.fxml"));

primaryStage.setTitle("To-Do List Application");

primaryStage.setScene(new Scene(root, 400, 400));

primaryStage.show();

}

public static void main(String[] args) {

launch(args);

}

}

**Controller:**

import javafx.fxml.FXML;

import javafx.scene.control.\*;

public class TodoListController {

@FXML

private TextField taskInput;

@FXML

private ListView<String> taskList;

@FXML

private Button addButton;

@FXML

private Button removeButton;

@FXML

private Button markCompleteButton;

public void initialize() {

addButton.setOnAction(event -> addTask());

removeButton.setOnAction(event -> removeTask());

markCompleteButton.setOnAction(event -> markTaskComplete());

}

private void addTask() {

String task = taskInput.getText();

if (!task.isEmpty()) {

taskList.getItems().add(task);

taskInput.clear();

}

}

private void removeTask() {

int selectedIndex = taskList.getSelectionModel().getSelectedIndex();

if (selectedIndex >= 0) {

taskList.getItems().remove(selectedIndex);

}

}

private void markTaskComplete() {

int selectedIndex = taskList.getSelectionModel().getSelectedIndex();

if (selectedIndex >= 0) {

String task = taskList.getItems().get(selectedIndex);

taskList.getItems().set(selectedIndex, "[Completed] " + task);

}

}

}

**Task#4 Main**

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.stage.Stage;

public class Main extends Application {

@Override

public void start(Stage primaryStage) throws Exception {

Parent root = FXMLLoader.load(getClass().getResource("ContactManager.fxml"));

primaryStage.setTitle("Contact Manager");

primaryStage.setScene(new Scene(root, 600, 400));

primaryStage.show();

}

public static void main(String[] args) {

launch(args);

}

}

**Controller:**

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.fxml.FXML;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

public class ContactManagerController {

@FXML

private TextField nameField;

@FXML

private TextField phoneNumberField;

@FXML

private TextField emailField;

@FXML

private TableView<Contact> contactTable;

@FXML

private TableColumn<Contact, String> nameColumn;

@FXML

private TableColumn<Contact, String> phoneNumberColumn;

@FXML

private TableColumn<Contact, String> emailColumn;

private ObservableList<Contact> contacts = FXCollections.observableArrayList();

public void initialize() {

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

phoneNumberColumn.setCellValueFactory(new PropertyValueFactory<>("phoneNumber"));

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

contactTable.setItems(contacts);

}

@FXML

private void addContact() {

String name = nameField.getText();

String phoneNumber = phoneNumberField.getText();

String email = emailField.getText();

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

Contact contact = new Contact(name, phoneNumber, email);

contacts.add(contact);

nameField.clear();

phoneNumberField.clear();

emailField.clear();

}

}

@FXML

private void editContact() {

Contact selectedContact = contactTable.getSelectionModel().getSelectedItem();

if (selectedContact != null) {

String newName = nameField.getText();

String newPhoneNumber = phoneNumberField.getText();

String newEmail = emailField.getText();

if (!newName.isEmpty() && !newPhoneNumber.isEmpty() && !newEmail.isEmpty()) {

selectedContact.setName(newName);

selectedContact.setPhoneNumber(newPhoneNumber);

selectedContact.setEmail(newEmail);

contactTable.refresh();

nameField.clear();

phoneNumberField.clear();

emailField.clear();

}

}

}

@FXML

private void deleteContact() {

Contact selectedContact = contactTable.getSelectionModel().getSelectedItem();

if (selectedContact != null) {

contacts.remove(selectedContact);

nameField.clear();

phoneNumberField.clear();

emailField.clear();

}

}

}

**Contact class:**

public class Contact {

private String name;

private String phoneNumber;

private String email;

public Contact(String name, String phoneNumber, String email) {

this.name = name;

this.phoneNumber = phoneNumber;

this.email = email;

}

}