



AOOP Assignment Submission Report

[Submitted as part of CTA Assignment No-2]

Course:	Advanced Object-Oriented Programming	Course Code:	18UCSE508
Semester:	V	Division:	A

Submitted by:

USN:	2SD20CS058	Name:	NAMAN N KABADI	
------	------------	-------	----------------	--

AOOPASSIGNMENT 2:

TERM WORK 01:

1. Problem Definition:

Write a Java program to build the GUI application using JavaFX for the following requirements:

- a) Read user name and password using appropriate JavaFX controls.
- b) Validate the input. If user name and password are matched with the assumed values, then display the welcome scene with proper text.
- c) If user name and password don't match, then raise appropriate exception.

2. Java Program:

```
/*
USER NAME AND PASSWORD VALIDATION SYSTEM
Author: NAMAN KABADI
USN:2SD20CS058
PROBLEM STATEMENT:
Write a Java program to build the GUI application using JavaFX for the
following requirements:
a) Read user name and password using appropriate JavaFX controls.
b) Validate the input. If user name and password are matched with the
assumed values, then
display the welcome scene with proper text.
c) If user name and password don't match, then raise appropriate
exception.

*/

package application;
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.HBox;
import javafx.scene.text.Font;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
import javafx.scene.text.TextAlignment;
import javafx.stage.Stage;
```

```
public class Question1 extends Application {
    //Here we are assuming the user name and password as admin and
    proceeding further
    private String adminUsername = "admin";
    private String adminPassword = "admin";

    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("JavaFX Login Example Demo");
        GridPane grid = new GridPane();
        grid.setAlignment(Pos.CENTER);
        grid.setHgap(10);
        grid.setVgap(10);
        grid.setPadding(new Insets(25, 25, 25, 25));

        Text sceneTitle = new Text("Login");
        sceneTitle.setTextAlignment(TextAlignment.CENTER);
        sceneTitle.setFont(Font.font("Tahoma", FontWeight.NORMAL, 40));
        grid.add(sceneTitle, 0, 0, 2, 1);

        Label userName = new Label("User Name:");
        grid.add(userName, 0, 1);
        TextField userTextField = new TextField();
        grid.add(userTextField, 1, 1);

        Label pw = new Label("Password:");
        grid.add(pw, 0, 2);

        PasswordField passwordBox = new PasswordField();
        grid.add(passwordBox, 1, 2);

        Button btn = new Button("Sign in");
        HBox hbBtn = new HBox(10);
        hbBtn.setAlignment(Pos.BOTTOM_RIGHT);
        hbBtn.getChildren().add(btn);
        grid.add(hbBtn, 1, 4);

        final Text actiontarget = new Text();
        grid.add(actiontarget, 1, 6);

        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent e) {
                String username = userTextField.getText().toString();
                String password = passwordBox.getText().toString();

                if(userTextField.getText().isEmpty()) {
                    showAlert(Alert.AlertType.ERROR, "Form Error!",
```

```
                "Please enter your correct user name");
            return;
        }
        if(passwordBox.getText().isEmpty()) {
            showAlert(Alert.AlertType.ERROR, "Form Error!",
                "Please enter a valid password");
            return;
        }

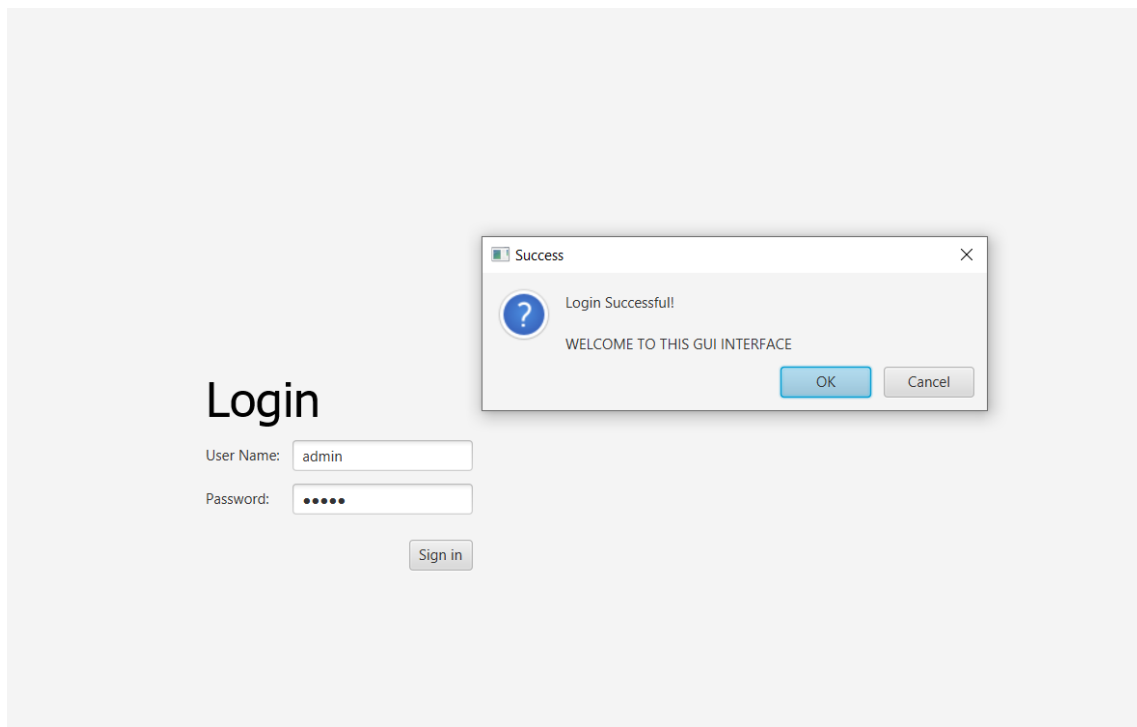
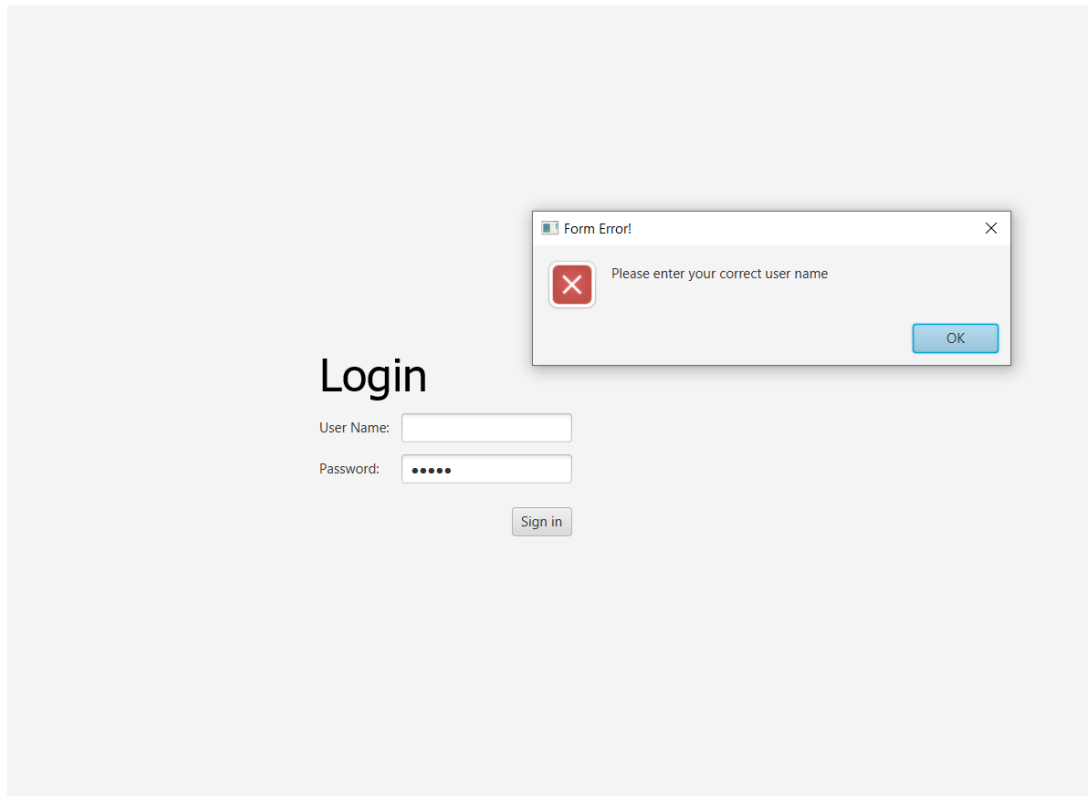
        if(adminUsername.equals(username) &&
adminPassword.equals(password)){
            infoBox("Login Successful!\n\nWELCOME TO THIS GUI
INTERFACE", null, "Success");
        } else{
            infoBox("Please enter correct User Name and Password",
null, "Failed");
        }
    }
    });

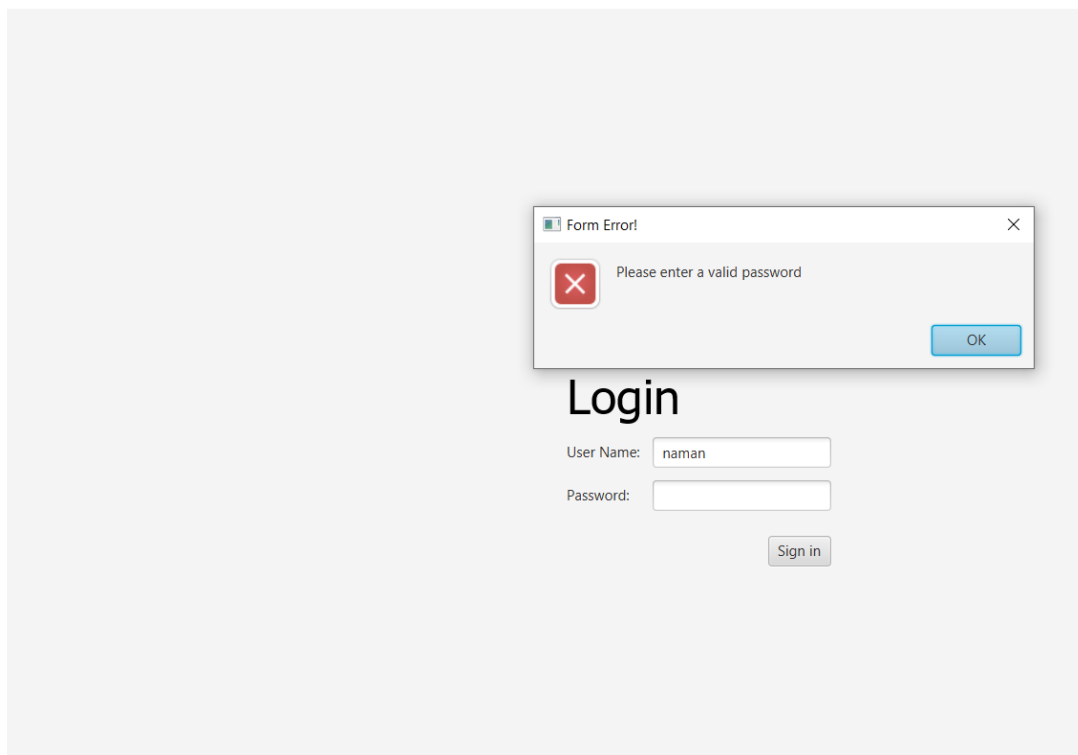
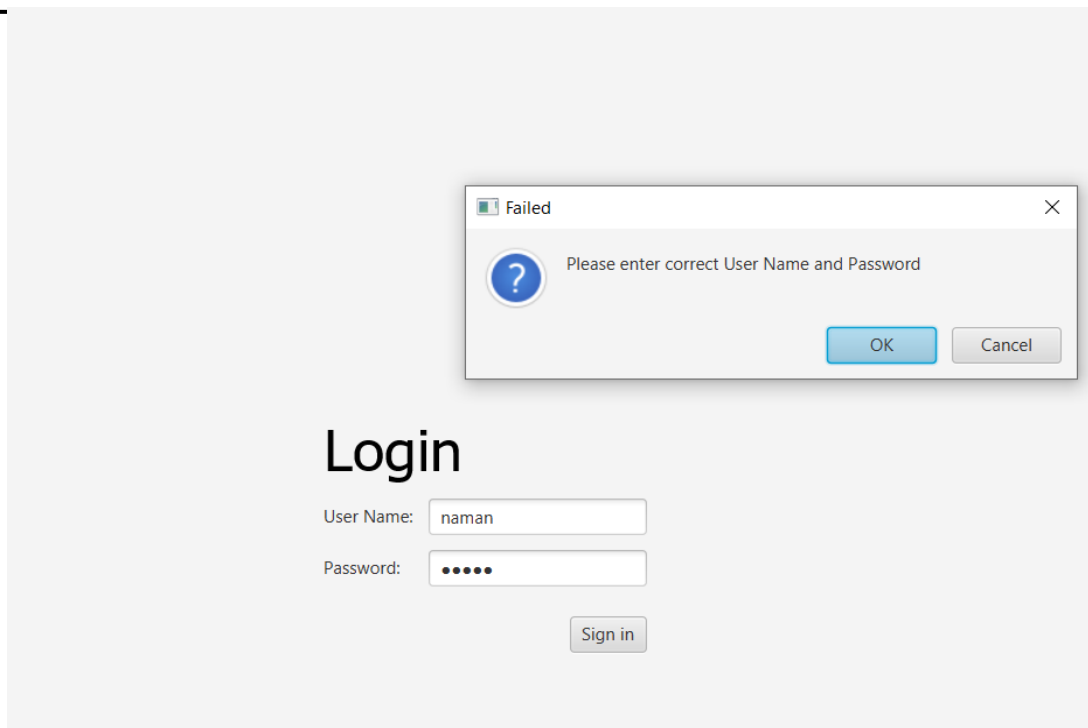
    Scene scene = new Scene(grid, 300, 275);
    primaryStage.setScene(scene);
    primaryStage.show();
}

private static void showAlert(Alert.AlertType alertType, String title,
String message) {
    Alert alert = new Alert(alertType);
    alert.setTitle(title);
    alert.setHeaderText(null);
    alert.setContentText(message);
    alert.show();
}

public static void infoBox(String infoMessage, String headerText,
String title){
    Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
    alert.setContentText(infoMessage);
    alert.setTitle(title);
    alert.setHeaderText(headerText);
    alert.showAndWait();
}
}
```

3. Screen Shots of Execution:





TERM WORK 02:

1. Problem Definition:

Write a Java program to build the GUI application using JavaFX for the following requirements:

- a) Create a Menu control to display the menu items: File, Edit & Help.
- b) Create sub menus in the order: File → New, Open & Save. Edit → Cut, Copy & Paste.

Help → Help Centre, About Us

The program must use Mnemonics and Accelerators (wherever appropriate) to Menu Items.

2. Java Program:

```
/*
Menu Control:
Author: NAMAN KABADI
USN:2SD20CS058
PROBLEM STATEMENT:
Write a Java program to build the GUI application using JavaFX for the
following requirements:
a) Create a Menu control to display the menu items: File, Edit & Help.
b) Create sub menus in the order: File → New, Open & Save. Edit → Cut,
Copy & Paste.
Help → Help Center, About Us
The program must use Mnemonics and Accelerators (wherever appropriate) to
Menu Items.
*/
package application;
//Demonstrate Menus
import javafx.application.*;
import javafx.scene.*;
import javafx.stage.*;
import javafx.scene.layout.*;
import javafx.scene.control.*;
import javafx.event.*;
import javafx.geometry.Pos;

public class Question2 extends Application {
    Label response;

    public static void main(String[] args) {
        // Start the JavaFX application by calling launch().
    }
}
```

```
        launch(args);
    }

    // Override the start() method.
    public void start(Stage myStage) {

        // Give the stage a title.
        myStage.setTitle("Menu Items Demo");

        // Use a BorderPane for the root node.
        BorderPane rootNode = new BorderPane();

        // Create a scene.
        Scene myScene = new Scene(rootNode, 300, 300);

        // Set the scene on the stage.
        myStage.setScene(myScene);

        // Create a label that will report the selection.
        response = new Label("Menu Demo");

        // Create the menu bar.
        MenuBar mb = new MenuBar();

        // Create the File menu.
        Menu fileMenu = new Menu("File");
        MenuItem open = new MenuItem("Open");
        MenuItem save = new MenuItem("Save");
        MenuItem exit = new MenuItem("Exit");
        MenuItem close = new MenuItem("Close");
        fileMenu.getItems().addAll(open, close, save, new
SeparatorMenuItem(), exit);

        // Add File menu to the menu bar.
        mb.getMenus().add(fileMenu);

        // Create the Edit menu.
        Menu EditMenu = new Menu("Edit");
        // Create the Edit sub-menu.
        MenuItem cutSubmenu = new MenuItem("Cut");
        MenuItem copySubmenu = new MenuItem("Copy");
        MenuItem pasteSubmenu = new MenuItem("Paste");
        EditMenu.getItems().addAll(cutSubmenu, copySubmenu, pasteSubmenu);
        mb.getMenus().add(EditMenu);

        // Create the Help menu.
        Menu helpMenu = new Menu("Help");
        MenuItem helpCentre = new MenuItem("Help Centre");
        MenuItem about = new MenuItem("About Us");
        helpMenu.getItems().addAll(helpCentre, about);

        // Add Help menu to the menu bar.
        mb.getMenus().add(helpMenu);
    }
}
```



```
// Create the context menu items
MenuItem cut = new MenuItem("Cut");
MenuItem copy = new MenuItem("Copy");
MenuItem paste = new MenuItem("Paste");

// Create a context (i.e., popup) menu that shows edit options.
final ContextMenu editMenu = new ContextMenu(cut, copy, paste);

// Create one event handler that will handle menu action events.
EventHandler<ActionEvent> MEHandler = new
EventHandler<ActionEvent>() {
    public void handle(ActionEvent ae) {
        String name = ((MenuItem) ae.getTarget()).getText();
        // If Exit is chosen, the program is terminated.
        if (name.equals("Exit"))
            Platform.exit();
        response.setText(name + " selected");
    }
};

// Set action event handlers for the menu items.
open.setOnAction(MEHandler);
close.setOnAction(MEHandler);
save.setOnAction(MEHandler);
exit.setOnAction(MEHandler);
about.setOnAction(MEHandler);
cutSubmenu.setOnAction(MEHandler);
copySubmenu.setOnAction(MEHandler);
helpCentre.setOnAction(MEHandler);
pasteSubmenu.setOnAction(MEHandler);
cut.setOnAction(MEHandler);
copy.setOnAction(MEHandler);
paste.setOnAction(MEHandler);

// Create a text field and set its column width to 20.
TextField tf = new TextField();
tf.setPrefColumnCount(20);

// Add the context menu to the text field.
tf.setContextMenu(editMenu);

// Add the menu bar to the top of the border pane and
// the response label to the center position.
rootNode.setTop(mb);

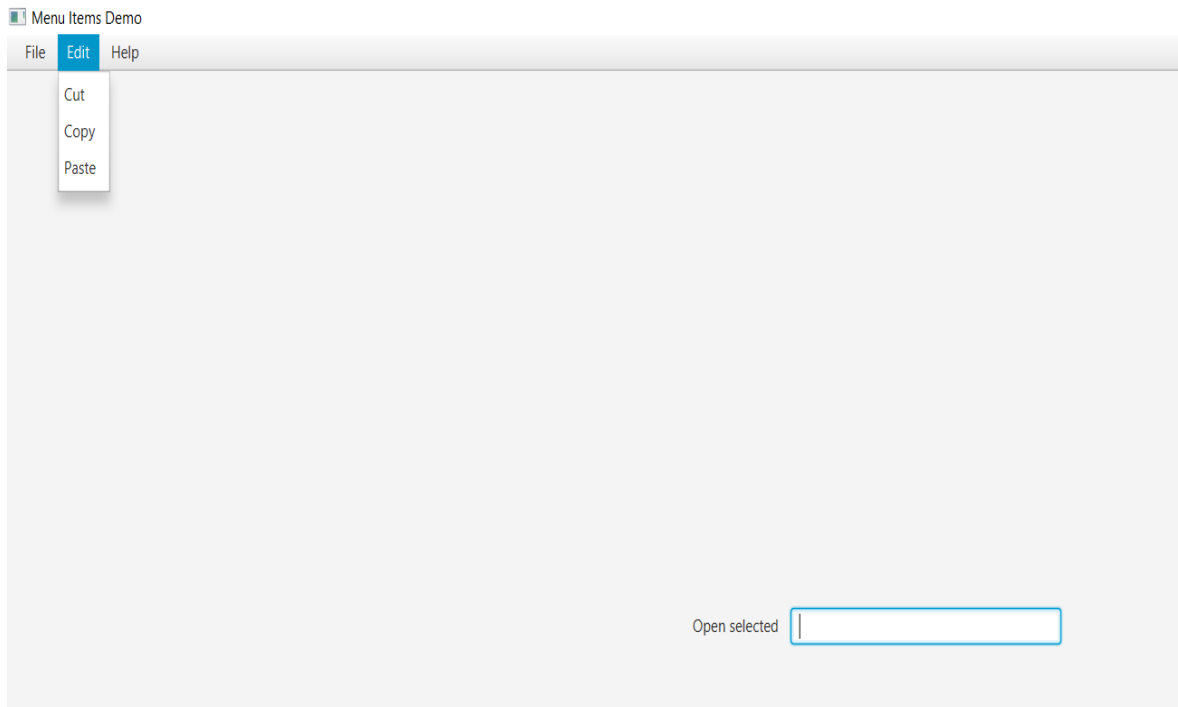
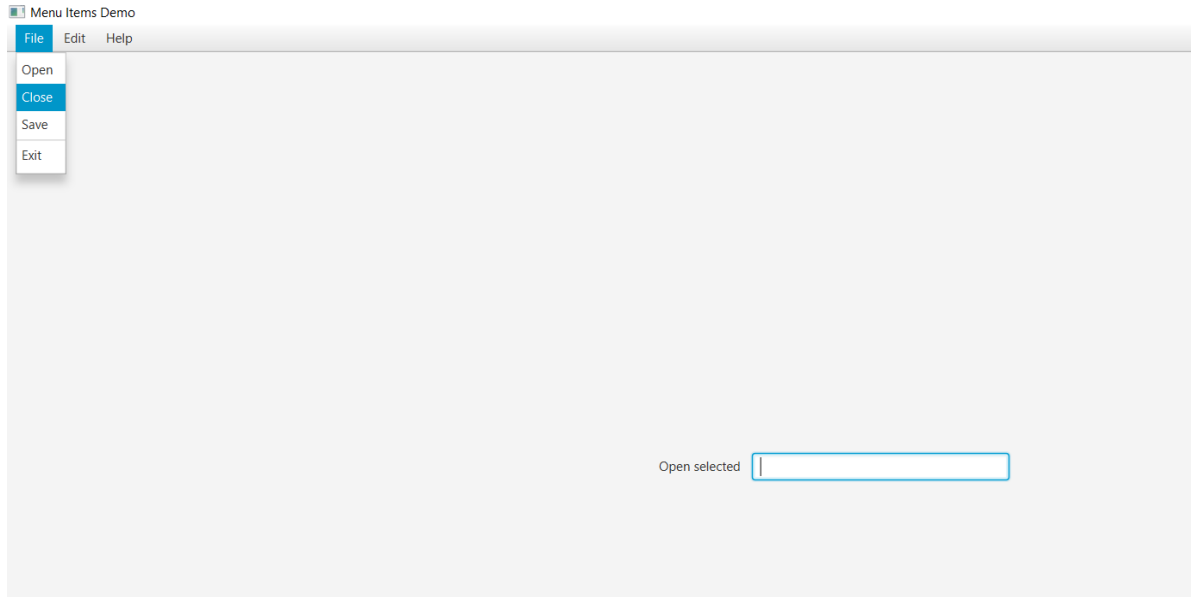
// Create a flow pane that will hold both the response
// label and the text field.
FlowPane fpRoot = new FlowPane(10, 10);

// Center the controls in the scene.
fpRoot.setAlignment(Pos.CENTER);

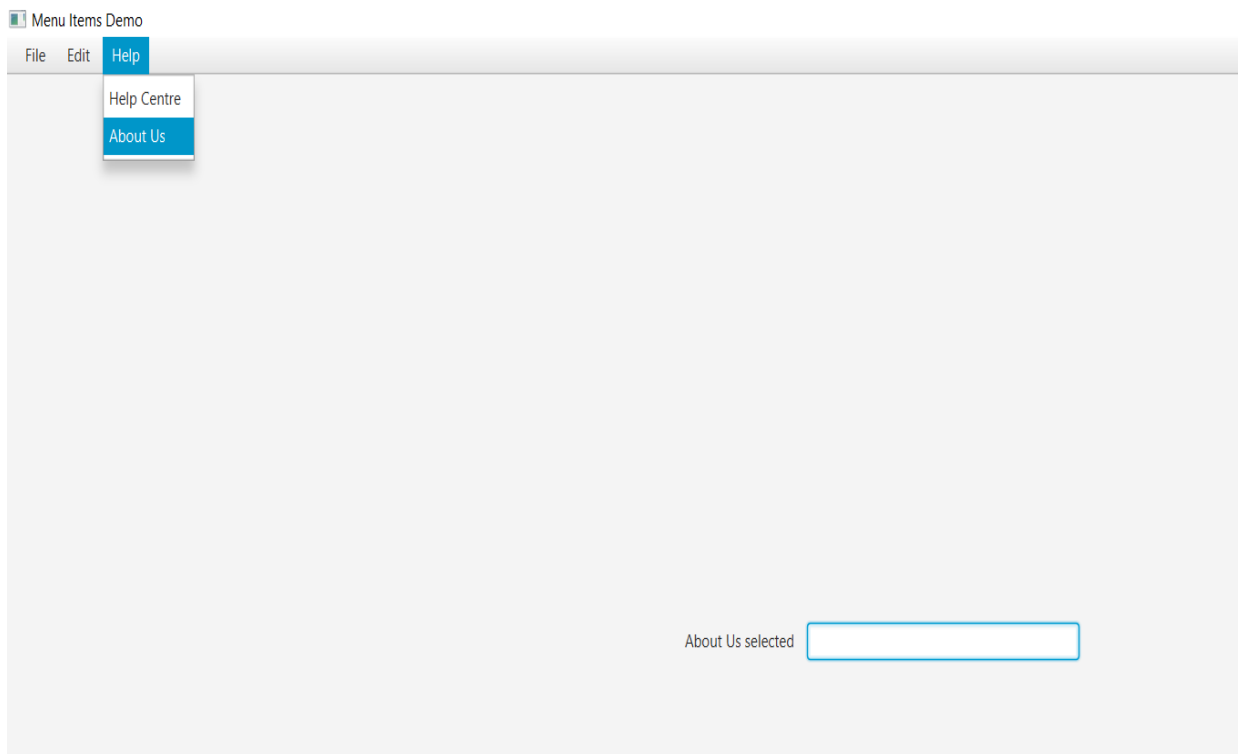
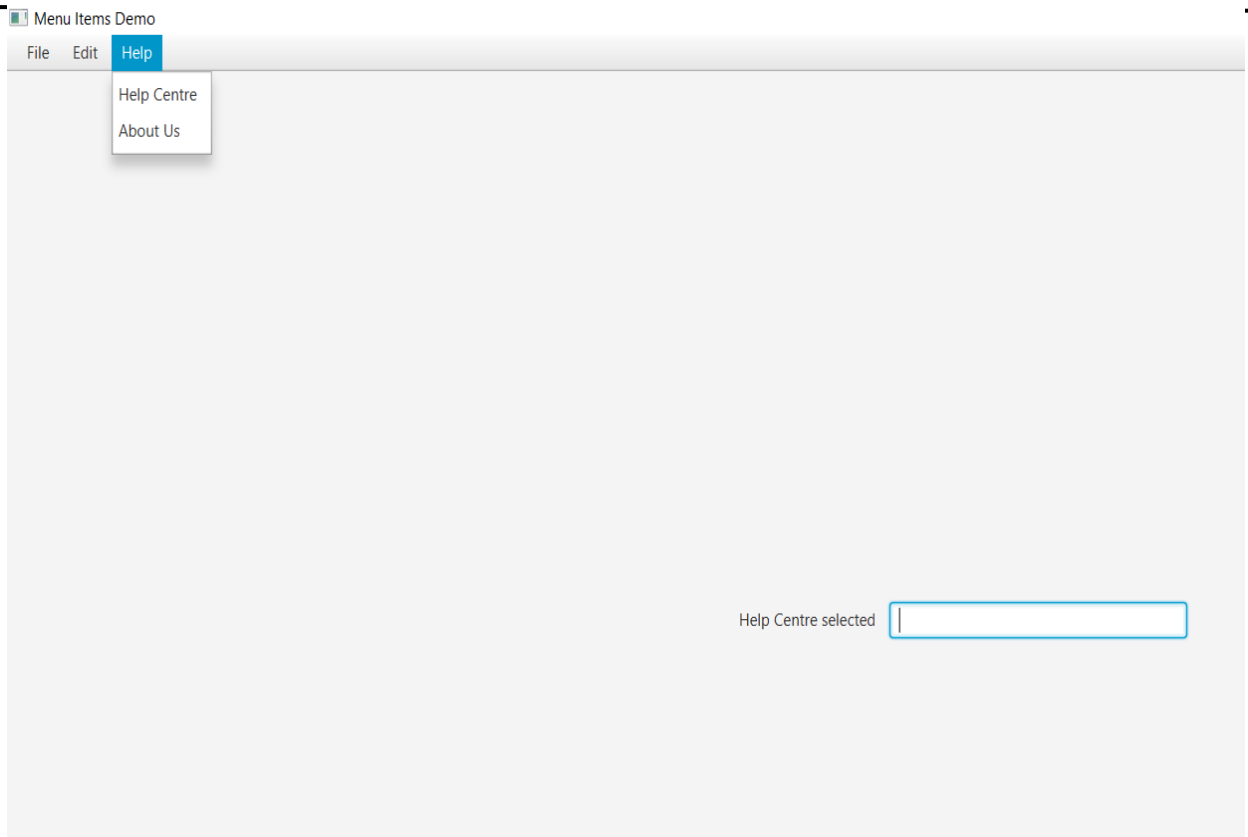
// Add both the label and the text field to the flow pane.
fpRoot.getChildren().addAll(response, tf);
```

```
// Add the flow pane to the center of the border layout.  
rootNode.setCenter(fpRoot);  
  
// Show the stage and its scene.  
myStage.show();  
}  
}
```

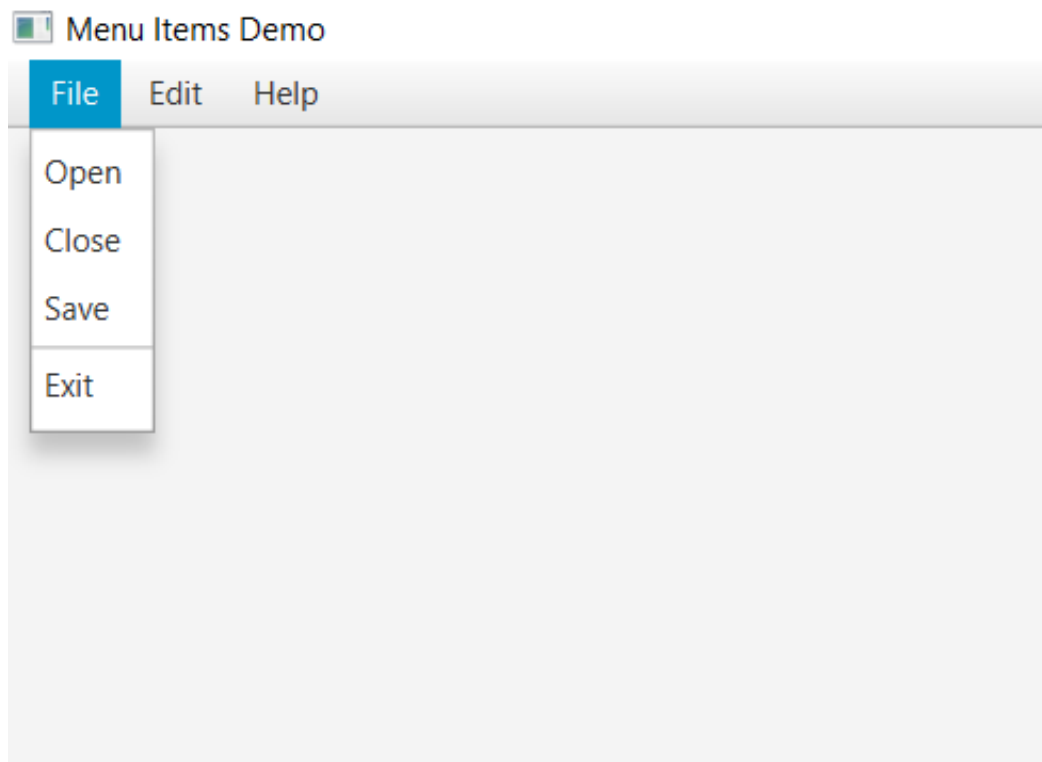
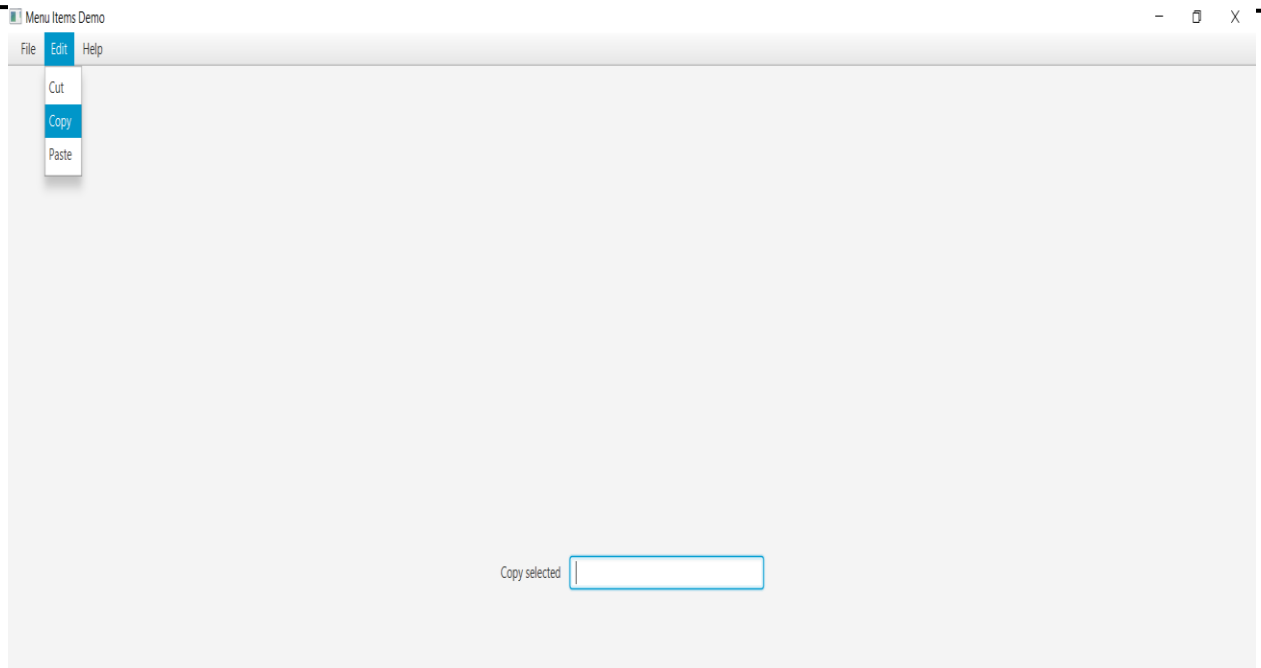
2. Screen Shots of Execution:



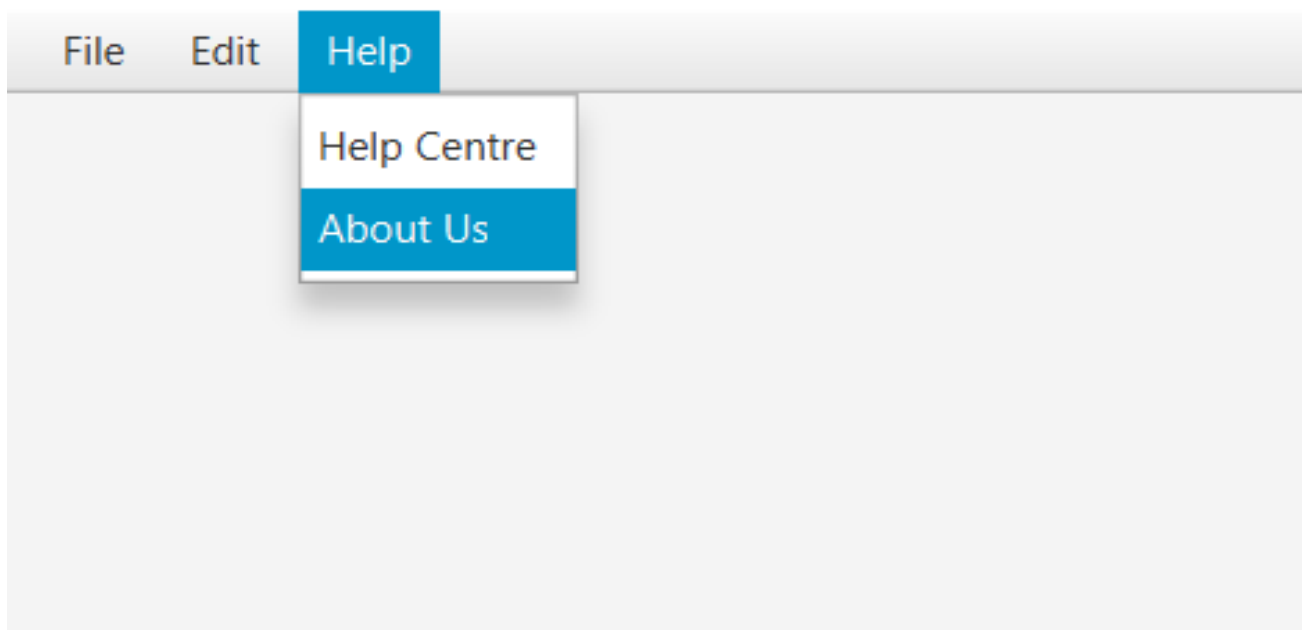
AOOP Assignment Submission Report



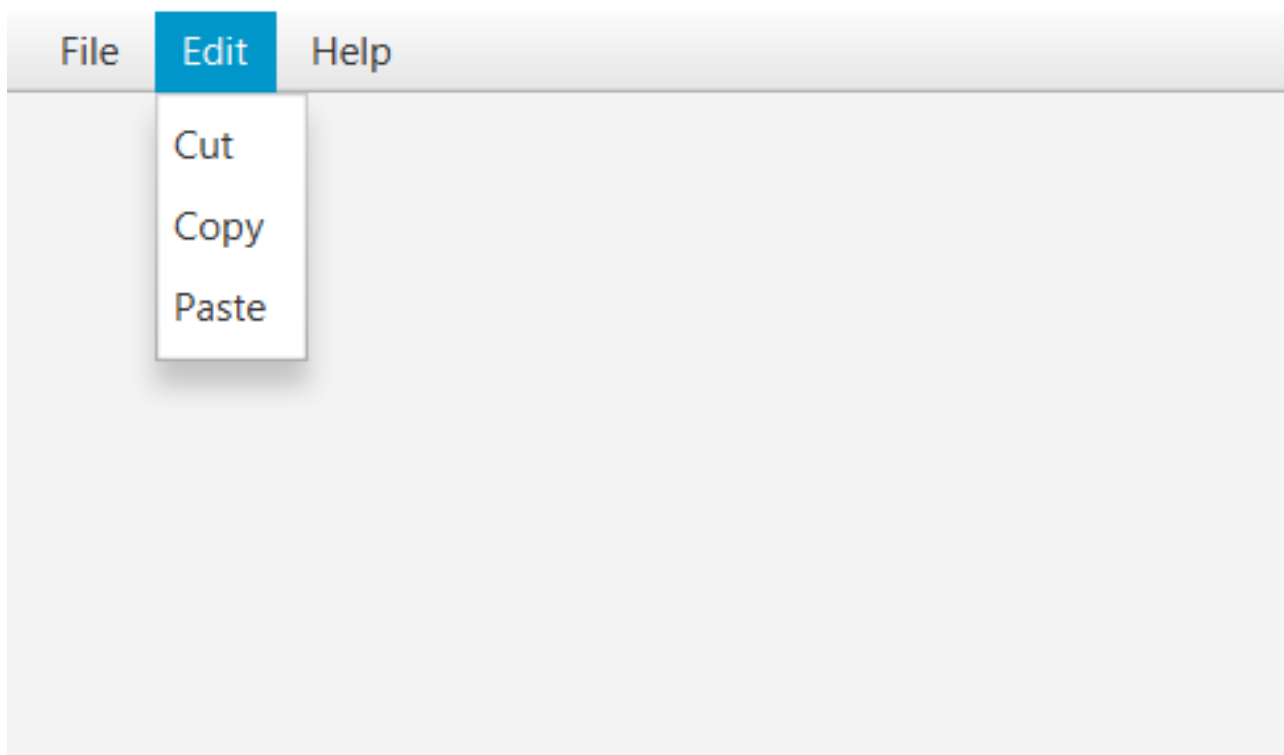
AOOP Assignment Submission Report



Menu Items Demo



Menu Items Demo



TERM WORK 03:

1. Problem Definition:

Write a Java program to build the GUI application using JavaFX for the following requirements:

- a) Create Context menu involving the menu items in the order: New & View.
- b) Create sub menus for the above main context menu: New → File, Folder & Image.
View → Large, Medium & Small.

The context menu must be displayed on right-click of the mouse button.

2. Java Program:

```
/*
Menu Control:
Author: NAMAN KABADI
USN:2SD20CS058
PROBLEM STATEMENT:
Write a Java program to build the GUI application using JavaFX for the
following requirements:
a) Create Context menu involving the menu items in the order: New & View.
b) Create sub menus for the above main context menu: New → File, Folder &
Image.
View → Large, Medium & Small.
The context menu must be displayed on right-click of the mouse button.
*/
package application;

//Demonstrate Menus
import javafx.application.*;
import javafx.scene.*;
import javafx.stage.*;
import javafx.scene.layout.*;
import javafx.scene.control.*;
import javafx.event.*;
import javafx.geometry.Pos;
//Demo 2
public class Question3 extends Application {
    Label response;

    public static void main(String[] args) {
        // Start the JavaFX application by calling launch().
        launch(args);
    }
}
```

```
// Override the start() method.
public void start(Stage myStage) {

    // Give the stage a title.
    myStage.setTitle("Demonstrate Menus");

    // Use a BorderPane for the root node.
    BorderPane rootNode = new BorderPane();

    // Create a scene.
    Scene myScene = new Scene(rootNode, 300, 300);

    // Set the scene on the stage.
    myStage.setScene(myScene);

    // Create a label that will report the selection.
    response = new Label("Menu Demo");

    // Create the menu bar.
    MenuBar mb = new MenuBar();
    //Creating the context menu New
    Menu New = new Menu("New");
    MenuItem File = new MenuItem("File");
    MenuItem Folder = new MenuItem("Folder");
    MenuItem Image = new MenuItem("Image");
    New.getItems().addAll(File, Folder, Image);
    mb.getMenus().add(New);

    Menu View = new Menu("View");
    MenuItem Large = new MenuItem("Large");
    MenuItem Medium = new MenuItem("Medium");
    MenuItem Small = new MenuItem("Small");
    View.getItems().addAll(Large, Medium, Small);
    mb.getMenus().add(View);

    // Create the context menu items
    MenuItem cut = new MenuItem("Cut");
    MenuItem copy = new MenuItem("Copy");
    MenuItem paste = new MenuItem("Paste");

    // Create a context (i.e., pop up) menu that shows edit options.
    final ContextMenu editMenu = new ContextMenu(cut, copy, paste);

    // Create one event handler that will handle menu action events.
    EventHandler<ActionEvent> MEHandler = new
    EventHandler<ActionEvent>() {
        public void handle(ActionEvent ae) {
            String name = ((MenuItem) ae.getTarget()).getText();
            // If Exit is chosen, the program is terminated.
            if (name.equals("Exit"))
                Platform.exit();
        }
    };
}
```

```
        response.setText(name + " selected");
    }
};

// Set action event handlers for the menu items.
New.setOnAction(MEHandler);
View.setOnAction(MEHandler);
cut.setOnAction(MEHandler);
copy.setOnAction(MEHandler);
paste.setOnAction(MEHandler);

// Create a text field and set its column width to 20.
TextField tf = new TextField();
tf.setPrefColumnCount(20);

// Add the context menu to the text field.
tf.setContextMenu(editMenu);

// Add the menu bar to the top of the border pane and
// the response label to the center position.
rootNode.setTop(mb);

// Create a flow pane that will hold both the response
// label and the text field.
FlowPane fpRoot = new FlowPane(10, 10);


// Center the controls in the scene.
fpRoot.setAlignment(Pos.CENTER);

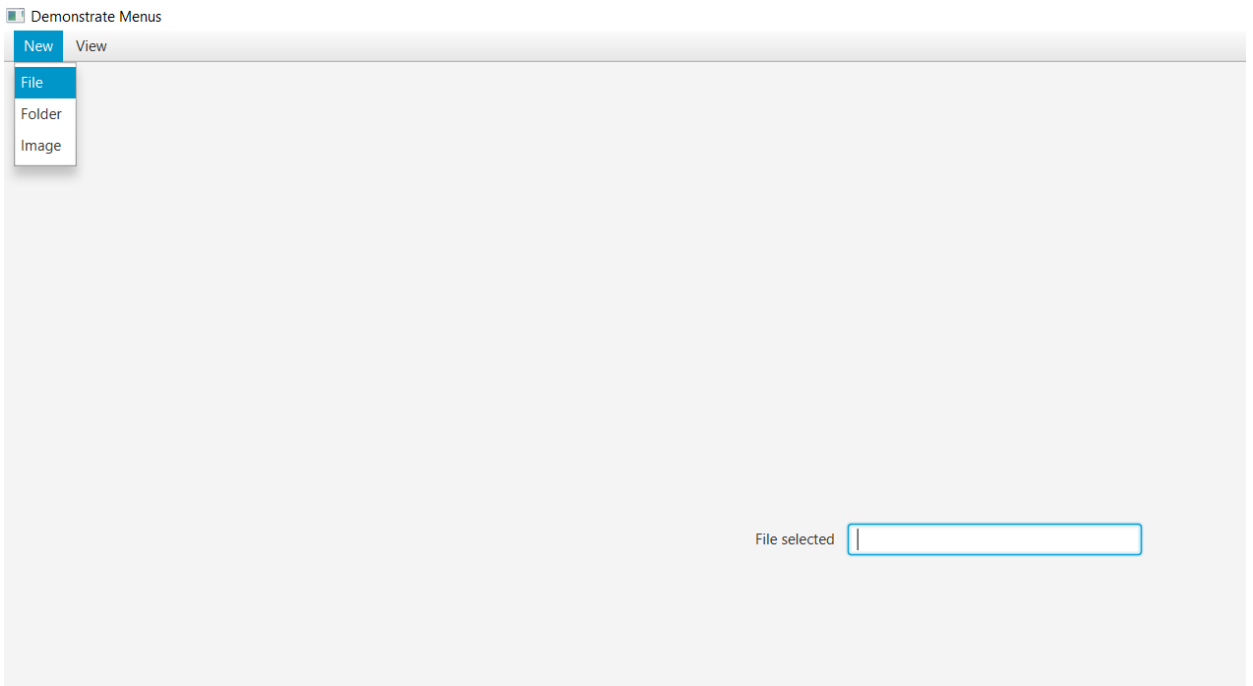
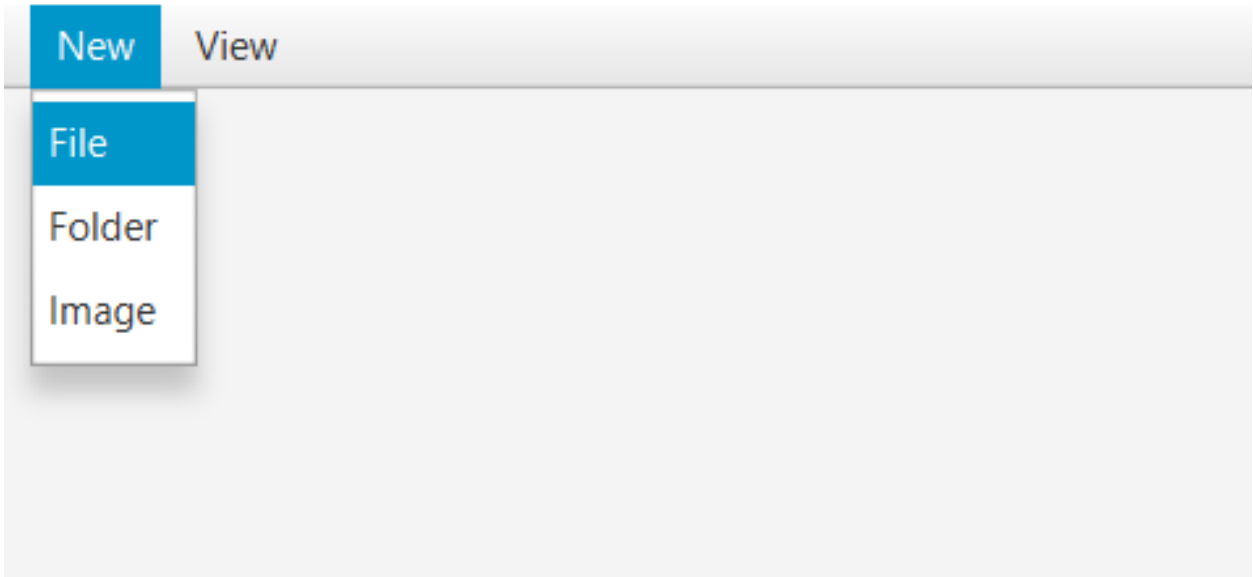
// Add both the label and the text field to the flow pane.
fpRoot.getChildren().addAll(response, tf);

// Add the flow pane to the center of the border layout.
rootNode.setCenter(fpRoot);

// Show the stage and its scene.
myStage.show();
}
```


3. Screen Shots of Execution:

 Demonstrate Menus



AOOP Assignment Submission Report

Demonstrate Menus

New View
File
Folder
Image

Image selected CONEXT MENU

Demonstrate Menus

New View

File
Folder
Image

AOOP Assignment Submission Report

Demonstrate Menus

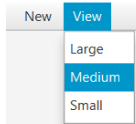
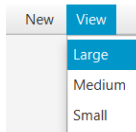


Image selected CONEXT MENU

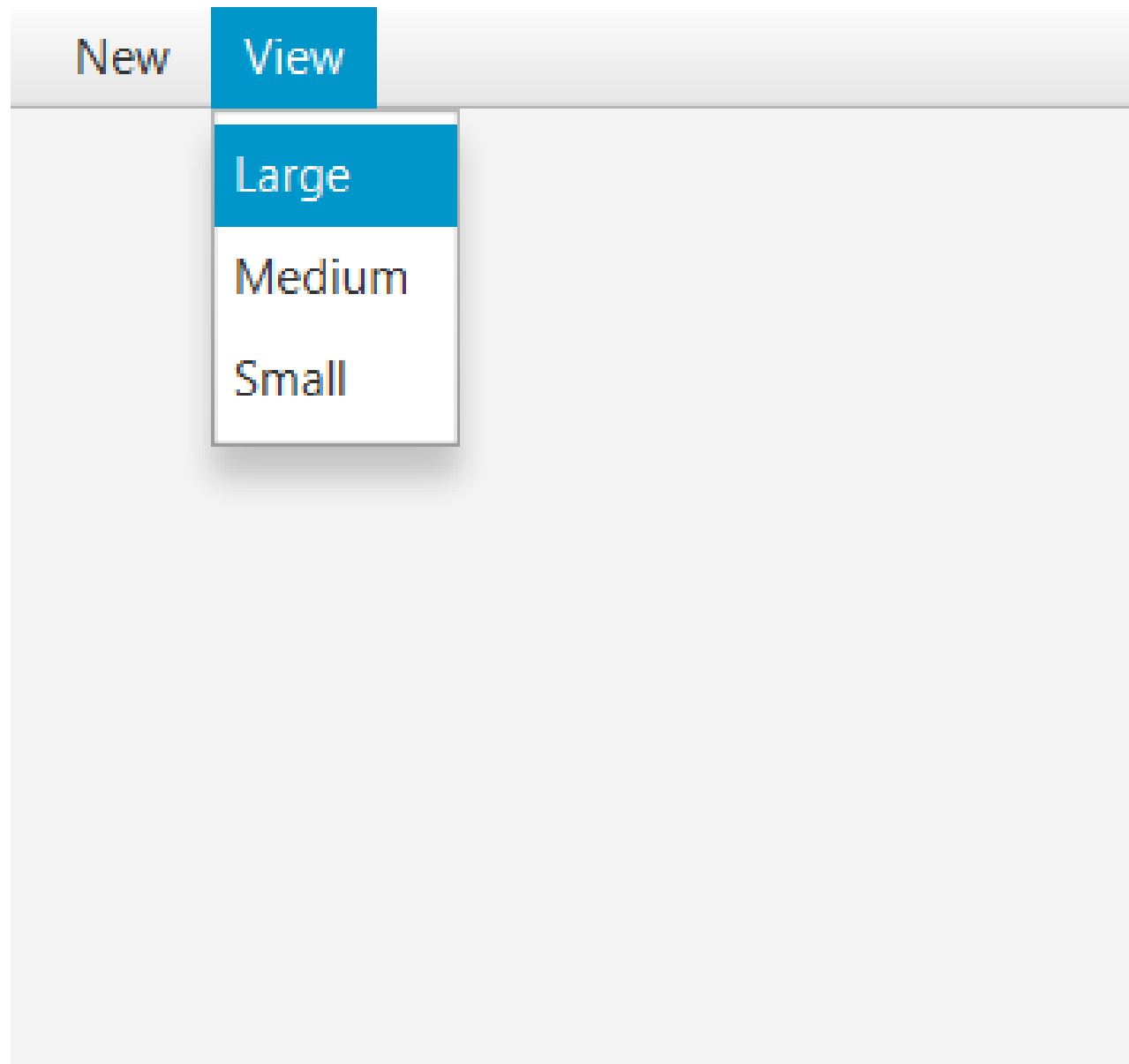
Demonstrate Menus



Large selected CONEXT MENU



Demonstrate Menus

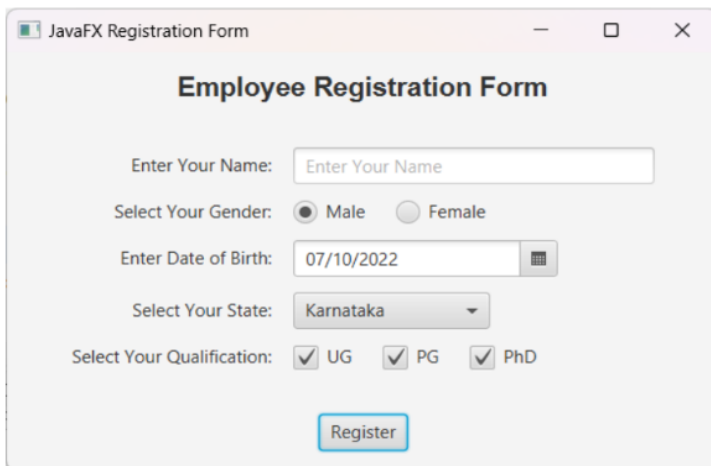


TERM WORK 04:

1. Problem Definition:

Write a JavaFX program that produces the following output when executed and displays Dialog Box

(as shown in Figure.2) on click of Register button (as shown in Figure.1):



The screenshot shows a JavaFX window titled "JavaFX Registration Form". Inside, the title "Employee Registration Form" is centered. The form contains the following fields and controls:

- "Enter Your Name:" followed by a text input field containing "Enter Your Name".
- "Select Your Gender:" with two radio buttons, "Male" (selected) and "Female".
- "Enter Date of Birth:" followed by a date input field containing "07/10/2022" and a calendar icon.
- "Select Your State:" followed by a dropdown menu showing "Karnataka".
- "Select Your Qualification:" with three checked checkboxes: "UG", "PG", and "PhD".
- A blue "Register" button at the bottom right.

Figure.1

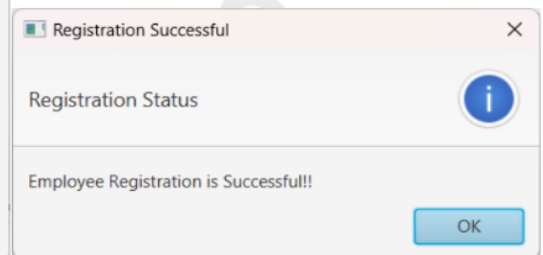


Figure. 2

2. Java Program:

```
package application;

import javafx.application.Application;
import javafx.geometry.HPos;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Button;
import javafx.scene.control.ComboBox;
import javafx.scene.control.DatePicker;
import javafx.scene.control.Label;
import javafx.scene.control.RadioButton;
import javafx.scene.control.TextField;
import javafx.scene.control.ToggleGroup;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.scene.text.Font;
import javafx.stage.Stage;

public class Q4 extends Application {

    @Override
    public void start(Stage primaryStage) {
        try {

            String state[] = {
                "Maharashtra", "Kerala", "Karnataka", "Haryana", "Andhra Pradesh", "Arunachal Pradesh", "Assam", "Bihar", "Chhattisgarh",
                "Goa", "Gujarat", "PUNJAB"};
            VBox cBox = new VBox();

            Scene scene = new Scene(cBox, 450, 300);

            scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
            primaryStage.setScene(scene);

            Label headingLabel = new Label("Employee Registration Form");
            headingLabel.setFont(Font.font(26));
            Label nameLabel = new Label("Enter Your Name:");
            Label genderLabel = new Label("Enter Your Gender");
            Label dobLabel = new Label("Enter Date of Birth");
            Label stateLabel = new Label("Select Your State");

            TextField nameField = new TextField();
```

```
nameField.setPrefWidth(250);
nameField.setPromptText("Enter Your Name");
nameField.setFocusTraversable(false);

RadioButton maleButton = new RadioButton("Male");
RadioButton femaleButton = new RadioButton("Female");

ToggleGroup genderGroup = new ToggleGroup();
maleButton.setToggleGroup(genderGroup);
femaleButton.setToggleGroup(genderGroup);

DatePicker dobDatePicker = new DatePicker();

ComboBox<String> stateComboBox = new ComboBox<>();
stateComboBox.setPrefWidth(130);
stateComboBox.getItems().addAll(state);
Button registerButton = new Button("Register");

registerButton.setOnAction((ae)->{

    Alert registerDialog = new Alert(AlertType.INFORMATION);
    registerDialog.setTitle("Registration Successful");

    registerDialog.setHeaderText("Registration Status");
    registerDialog.setContentText("Employee Registration
Successful!!");
    registerDialog.showAndWait();

} );
cBox.setAlignment(Pos.CENTER);
cBox.setSpacing(20);
cBox.setPadding(new Insets(0, 0, 0, 20));

GridPane cPane = new GridPane();
cPane.setHgap(10);
cPane.setVgap(10);

cPane.setAlignment(Pos.CENTER);

cPane.add(nameLabel, 0, 0);

GridPane.setHalignment(nameLabel, HPos.RIGHT);
cPane.add(nameField, 1, 0);

cPane.add(genderLabel, 0, 1);
GridPane.setHalignment(genderLabel, HPos.RIGHT);

HBox genderBox = new HBox();
genderBox.setSpacing(10);
genderBox.getChildren().addAll(maleButton, femaleButton);

cPane.add(genderBox, 1, 1);
```

```

cPane.add(dobLabel, 0, 2);
GridPane.setHalignment(dobLabel, HPos.RIGHT);

cPane.add(dobDatePicker, 1, 2);

cPane.add(stateLabel, 0, 3);
GridPane.setHalignment(stateLabel, HPos.RIGHT);

cPane.add(stateComboBox, 1, 3);


cBox.getChildren().addAll(headingLabel, cPane, registerButton);


primaryStage.show();

} catch (Exception e) {
    System.err.println(e);
}

}

public static void main(String[] args) {
    launch(args);
}

}

```

3. Screen Shots of Execution:

