Sources:

MS Bing Chat Copilot

In JavaFX, an AnchorPane is a type of layout container that allows you to anchor child nodes to its edges. It's especially useful when you want to position elements relative to the container's boundaries without manually handling their layout in terms of pixel values.

Key Features of AnchorPane:

A white background with black text

AI-generated content may be incorrect.

A black text on a white background

AI-generated content may be incorrect.

A black text on a white background

AI-generated content may be incorrect.

\*About the position:fixed property from CSS:

JavaFX CSS does not support the position: fixed property as you might find in traditional web CSS. However, you can achieve similar effects using layout managers and controls available in JavaFX.

For example, you can use a Pane or AnchorPane to position nodes at fixed locations within the scene.

Here's a simple example using AnchorPane:

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.layout.AnchorPane;

import javafx.stage.Stage;

public class FixedPositionExample extends Application {

@Override

public void start(Stage primaryStage) {

Button button = new Button("Fixed Position");

// Anchor the button at a fixed position

AnchorPane.setTopAnchor(button, 50.0);

AnchorPane.setLeftAnchor(button, 50.0);

AnchorPane anchorPane = new AnchorPane(button);

Scene scene = new Scene(anchorPane, 300, 200);

primaryStage.setScene(scene);

primaryStage.setTitle("Fixed Position Example");

primaryStage.show();

}

public static void main(String[] args) {

launch(args);

}

}

In this example, the button will always be positioned 50 pixels from the top and left edges of the AnchorPane, effectively mimicking a "fixed" position.

Ex2:

**package** application;

**import** javafx.application.Application;

**import** javafx.scene.Scene;

**import** javafx.scene.control.Button;

**import** javafx.scene.layout.AnchorPane;

**import** javafx.stage.Stage;

**public** **class** **Main** **extends** Application {

**@Override**

**public** **void** **start**(**Stage** primaryStage) {

// Create an AnchorPane

**AnchorPane** **anchorPane** = **new** **AnchorPane**();

// Create Buttons

**Button** **topButton** = **new** **Button**(**"Top Button"**);

**Button** **bottomButton** = **new** **Button**(**"Bottom Button"**);

// Anchor the buttons to the edges

**AnchorPane**.***setTopAnchor***(topButton, 10.0); // 10 pixels from the top

**AnchorPane**.***setLeftAnchor***(topButton, 20.0); // 20 pixels from the left

**AnchorPane**.***setBottomAnchor***(bottomButton, 10.0); // 10 pixels from the bottom

**AnchorPane**.***setRightAnchor***(bottomButton, 20.0); // 20 pixels from the right

// Add buttons to the AnchorPane

anchorPane.**getChildren**().**addAll**(topButton, bottomButton);

// Create the Scene

**Scene** **scene** = **new** **Scene**(anchorPane, **300**, **200**);

primaryStage.**setTitle**(**"AnchorPane Example"**);

primaryStage.**setScene**(scene);

primaryStage.**show**();

}

**public** **static** **void** **main**(**String**[] args) {

***launch***(args);

}

}

A screenshot of a computer

AI-generated content may be incorrect.

Let’s resize the app’s window and you’ll see the these buttons inside this app are still fixed at their original position:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A white background with black text

AI-generated content may be incorrect.

\* JavaFX doesn't directly support position:absolute like in standard CSS for web development. However, you can achieve similar functionality by using layout containers like AnchorPane or Pane to add absolute positioning via setLayoutX() and setLayoutY() methods.

**\*Using AnchorPane is just like some example above**

\*To Use Pane to anchor nodes to specific positions relative to the container edges, mimicking the behavior of position:absolute, please visit:

**C:\Users\terro\OneDrive\Technology\Programming Languages\_CLI\Java\CISC\_190\_Java\_Programming\Basic\_Knowledge\Chapter12\_JavaFX\_GUI\_Programming\_Basic\_Controls\Pane**