### **Lab Exercise: WKWebView Programmatically in iOS Swift**

**Objective:**

Learn how to create and use a WKWebView programmatically to load and display web content in an iOS app.

**Set Up the Project:**

Create a new Xcode project named WebViewDemo.

Choose "App" under iOS and set the language to Swift.

**Add WebKit Framework:**

* Open the project navigator and select the project file.
* Select the target and navigate to the "General" tab.
* Scroll down to "Frameworks, Libraries, and Embedded Content" and click the "+" button.
* Search for "WebKit.framework" and add it to the project.

**Implement the View Controller:**

* Open ViewController.swift.
* Replace the contents of ViewController.swift with the following code:

import UIKit

import WebKit

class ViewController: UIViewController {

var webView: WKWebView!

override func viewDidLoad() {

super.viewDidLoad()

// Initialize the web view

webView = WKWebView()

webView.translatesAutoresizingMaskIntoConstraints = false

// Add the web view to the main view

view.addSubview(webView)

// Set up the web view constraints

NSLayoutConstraint.activate([

webView.leadingAnchor.constraint(equalTo: view.leadingAnchor),

webView.trailingAnchor.constraint(equalTo: view.trailingAnchor),

webView.topAnchor.constraint(equalTo: view.topAnchor),

webView.bottomAnchor.constraint(equalTo: view.bottomAnchor)

])

// Load a URL

if let url = URL(string: "https://www.apple.com") {

let request = URLRequest(url: url)

webView.load(request)

}

}

}

**Run the App:**

* Run the app on a simulator or a physical device.
* Verify that the web view loads the specified URL and displays the web content.

**Summary:**

This lab exercise provides hands-on experience with implementing a WKWebView programmatically in an iOS app using Swift. By completing the exercise, you will gain practical knowledge of using web views to load and display web content dynamically in iOS apps.