### **Lab Experiment: Hide and Show Keyboard Programmatically in iOS Swift**

### **Objective:**

Learn how to programmatically hide and show the keyboard in an iOS app using Swift.

**Steps:**

**Set Up the Project:**

* Create a new Xcode project named KeyboardHandlingDemo.
* Choose "App" under iOS and set the language to Swift.

**Implement the View Controller:**

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

import UIKit

class ViewController: UIViewController, UITextFieldDelegate {

let textField: UITextField = {

let textField = UITextField()

textField.borderStyle = .roundedRect

textField.placeholder = "Enter text"

textField.translatesAutoresizingMaskIntoConstraints = false

return textField

}()

let showKeyboardButton: UIButton = {

let button = UIButton(type: .system)

button.setTitle("Show Keyboard", for: .normal)

button.addTarget(self, action: #selector(showKeyboard), for: .touchUpInside)

button.translatesAutoresizingMaskIntoConstraints = false

return button

}()

let hideKeyboardButton: UIButton = {

let button = UIButton(type: .system)

button.setTitle("Hide Keyboard", for: .normal)

button.addTarget(self, action: #selector(hideKeyboard), for: .touchUpInside)

button.translatesAutoresizingMaskIntoConstraints = false

return button

}()

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view.

// Add text field to the view

view.addSubview(textField)

view.addSubview(showKeyboardButton)

view.addSubview(hideKeyboardButton)

// Configure layout constraints

NSLayoutConstraint.activate([

textField.topAnchor.constraint(equalTo: view.safeAreaLayoutGuide.topAnchor, constant: 100),

textField.leadingAnchor.constraint(equalTo: view.leadingAnchor, constant: 20),

textField.trailingAnchor.constraint(equalTo: view.trailingAnchor, constant: -20),

textField.heightAnchor.constraint(equalToConstant: 40),

showKeyboardButton.topAnchor.constraint(equalTo: textField.bottomAnchor, constant: 20),

showKeyboardButton.leadingAnchor.constraint(equalTo: view.leadingAnchor, constant: 20),

hideKeyboardButton.topAnchor.constraint(equalTo: textField.bottomAnchor, constant: 20),

hideKeyboardButton.leadingAnchor.constraint(equalTo: showKeyboardButton.trailingAnchor, constant: 20)

])

// Set text field delegate

textField.delegate = self

}

@objc func showKeyboard() {

textField.becomeFirstResponder()

}

@objc func hideKeyboard() {

textField.resignFirstResponder()

}

// MARK: - UITextFieldDelegate

func textFieldShouldReturn(\_ textField: UITextField) -> Bool {

textField.resignFirstResponder()

return true

}

}

**Run the App:**

* Run the app on a simulator or a physical device.
* Tap the "Show Keyboard" button to bring up the keyboard.
* Tap the "Hide Keyboard" button to dismiss the keyboard.
* Verify that the keyboard is shown and hidden programmatically as expected.

**Summary:**

This lab experiment provides hands-on experience with hiding and showing the keyboard programmatically in an iOS app using Swift. By completing the experiment