### **Lab Exercise: Auto Layout Programmatically in iOS Swift**

### **Objective:**

Learn how to create and apply Auto Layout constraints programmatically to build a simple user interface in an iOS app.

**Tasks:**

**Set Up the Project:**

* Create a new Xcode project named AutoLayoutDemo.
* 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 {

let redView: UIView = {

let view = UIView()

view.backgroundColor = .red

view.translatesAutoresizingMaskIntoConstraints = false

return view

}()

let blueView: UIView = {

let view = UIView()

view.backgroundColor = .blue

view.translatesAutoresizingMaskIntoConstraints = false

return view

}()

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view.

// Add red view to the view

view.addSubview(redView)

// Add blue view to the view

view.addSubview(blueView)

// Configure layout constraints for red view

NSLayoutConstraint.activate([

redView.topAnchor.constraint(equalTo: view.safeAreaLayoutGuide.topAnchor, constant: 50),

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

redView.widthAnchor.constraint(equalToConstant: 100),

redView.heightAnchor.constraint(equalToConstant: 100)

])

// Configure layout constraints for blue view

NSLayoutConstraint.activate([

blueView.topAnchor.constraint(equalTo: redView.bottomAnchor, constant: 20),

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

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

blueView.bottomAnchor.constraint(equalTo: view.safeAreaLayoutGuide.bottomAnchor, constant: -50)

])

}

}

**Run the App:**

* Run the app on a simulator or a physical device.
* Verify that the red and blue views are displayed with the specified layout constraints.
* Verify that the views are positioned correctly and resized according to the constraints.

**Summary:**

This lab exercise provides hands-on experience with implementing Auto Layout programmatically in an iOS app using Swift. By completing the exercise, you will gain practical knowledge of creating and applying layout constraints to build user interfaces dynamically in iOS apps. wift. By completing the experiment