```swift
//
//  ViewController.swift
//  specialTopicSem4
//
//  Created by Anantha Krishna on 02/03/20.
//  Copyright © 2020 Anantha Krishna. All rights reserved.
//

import UIKit
import FirebaseDatabase
import Firebase
import GoogleSignIn
import FirebaseAuth

class ViewController: UIViewController,GIDSignInDelegate{


    @IBOutlet weak var signInButton:GIDSignInButton!


    override func viewDidLoad() {
        super.viewDidLoad()
        GIDSignIn.sharedInstance().delegate = self
        // Do any additional setup after loading the view.
    }


    @IBAction func signInButtonPressed(_ sender: Any) {

        GIDSignIn.sharedInstance()?.presentingViewController = self
        GIDSignIn.sharedInstance().signIn()
    }

    func sign(_ signIn: GIDSignIn!, didSignInFor user: GIDGoogleUser!,
     withError error: Error?) {
         // ...
         if let error = error {
            print(error.localizedDescription)
           return
         }

         guard let authentication = user.authentication else { return }
         let credential = GoogleAuthProvider.credential(withIDToken:
          authentication.idToken,accessToken: authentication.accessToken)

        Auth.auth().signIn(with: credential) { (AuthDataResult, Error) in
            if let error = Error{
            print(error)
            }

//            print(user.profile.name)
//            print(user.profile.familyName)
        }
```

```
        print("Successfully signed in using google")

        self.dismiss(animated: true) {
         self.performSegue(withIdentifier: "Test", sender: self)
        }

           // MARK: Firebase Sign In

       }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        if segue.identifier == "Test"
        {
// no data required to be passed
//            let destinationVC = segue.destination as!
 AfterSignInViewController

        }


    }
    func moveToNextView()
    {
        self.performSegue(withIdentifier: "Test", sender: self)
    }


}
```

```
//
//  AfterSignInViewController.swift
//  specialTopicSem4
//
//  Created by Anantha Krishna on 22/03/20.
//  Copyright © 2020 Anantha Krishna. All rights reserved.
//

import UIKit
import Firebase
import FirebaseDatabase

class AfterSignInViewController: UIViewController {




    @IBOutlet weak var statusLabel: UILabel!
    @IBOutlet weak var powerValue: UILabel!

    @IBOutlet weak var toggleValue: UILabel!

    var ref:DatabaseReference!
    var data:FirebaseData?
    var powerString:String?
    var toggleString:String?
    var timer:Timer?
    override func viewDidLoad() {
        super.viewDidLoad()
        ref = Database.database().reference()
        getFirebaseData(ref)
        // change the delay to 1s but proceed with caution
        timer = Timer.scheduledTimer(withTimeInterval: 1.0, repeats: true,
         block: { (Timer) in
             print("Inside dispatch ",self.powerString)
                      if let powerString = self.powerString
                      {
                          print("typecasted\n")
                          self.powerValue.text! = powerString
                          self.toggleValue.text! = self.toggleString!
                          self.getStatus()


                      }
             self.getFirebaseData(self.ref)
        })

    }

    func getStatus(){
        if (Double(self.powerString!) == 0.0)
        {
            if(self.toggleString! == "On")
            {
                self.statusLabel.text! = "Fused"
```

```
            }
            else if(self.toggleString! == "Off"){
                self.statusLabel.text! = "Normal"
            }
            else {
                self.statusLabel.text! = "Functioning in Auto Mode"
            }
        }
        else
        {
            if(self.toggleString! == "On"){
                self.statusLabel.text! = "Normal"
            }
            else if(self.toggleString! == "Off"){
                self.statusLabel.text! = "Switch Faulty"
            }
            else {
                self.statusLabel.text! = "Functioning in Auto Mode"
            }
        }
    }


    @IBAction func buttonPressed(_ sender: UIButton) {
        if(sender.titleLabel?.text! != "Button Disabled")
        {
         performSegue(withIdentifier: "ChartViewSegue", sender: self)
        }
    }

    @IBAction func toggleSwitchPressed(_ sender: Any) {
        if(self.toggleString == "On")
        {
            self.ref.child("Power").setValue(0.0)
            self.ref.child("Toggle").setValue("Off")
        }
        else if(self.toggleString == "Off")
        {
            // change that value to float if needed
          self.ref.child("Power").setValue(5.0)
          self.ref.child("Toggle").setValue("Auto")
        }
        else
        {
            // change that value to float if needed
          self.ref.child("Power").setValue(5.0)
          self.ref.child("Toggle").setValue("On")
        }


    }
    @objc func display(){

//                    print("Inside dispatch ",self.powerString)
                  if let powerString = self.powerString
```

```
                    {
//                      print("typecasted\n")
                        self.powerValue.text! = powerString
                        self.toggleValue.text! = self.toggleString!

                    }
    }



    func getFirebaseData(_ ref:DatabaseReference)
    {
        ref.child("Power").observeSingleEvent(of: .value) { (DataSnapshot)
         in
            let val:Double?
//            print(DataSnapshot.value)
            val = DataSnapshot.value as? Double
            if let val = val{
                self.powerString = String(val)
            }

        }
        ref.child("Toggle").observeSingleEvent(of: .value) { (DataSnapshot)
         in

            let val:String?
//            print(DataSnapshot.value)
            val = DataSnapshot.value as? String
            if let val = val{
                self.toggleString = val
            }
        }
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

        if (segue.identifier == "ChartViewSegue")
        {
            // no data to be passed from this view to the other so this is
             empty
        }
    }



}
```

```swift
//
//  ChartViewController.swift
//  specialTopicSem4
//
//  Created by Anantha Krishna on 04/04/20.
//  Copyright © 2020 Anantha Krishna. All rights reserved.
//

import UIKit
import Charts
import TinyConstraints
import Firebase

class ChartViewController: UIViewController ,ChartViewDelegate{
    lazy var lineChartView: LineChartView = {
        let chartView = LineChartView()
        chartView.backgroundColor = .systemBlue;
        return chartView
    }()

    var deviceType:String?
    var counter:Double = 0.0

    var ref:DatabaseReference!

    override func viewDidLoad() {
        super.viewDidLoad()
        view.addSubview(lineChartView)
        lineChartView.centerInSuperview()
        lineChartView.width(to: view)
        lineChartView.heightToWidth(of: view)
        ref = Database.database().reference()
        getFirebaseData(ref)
        setData()
        DispatchQueue.main.asyncAfter(deadline: .now()+1) {
            self.viewDidLoad()
        }

        // Do any additional setup after loading the view.
    }

    func chartValueSelected(_ chartView: ChartViewBase, entry:
     ChartDataEntry, highlight: Highlight) {
        print(entry)
    }
    func setData()
    {
        let set1 = LineChartDataSet(entries: yValues, label: "Value over
         time")
        let data = LineChartData(dataSet:set1)
        lineChartView.data = data
    }

    var yValues:[ChartDataEntry] = []
```

```
    func getFirebaseData(_ ref:DatabaseReference)
    {

        ref.child("Power").observeSingleEvent(of: .value) { (DataSnapshot)
         in

            var rawData:Double?
            rawData = DataSnapshot.value as? Double

            if let doubleFromString = rawData
            {
                print(doubleFromString)
                self.yValues.append(ChartDataEntry(x:self
                 .counter,y:doubleFromString))
                self.counter+=1
            }
        }

    }

    /*
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a
     little preparation before navigation
    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        // Get the new view controller using segue.destination.
        // Pass the selected object to the new view controller.
    }
    */

}
```

```swift
//
//  AppDelegate.swift
//  specialTopicSem4
//
//  Created by Anantha krishna on 02/03/20.
//  Copyright © 2020 Anantha Krishna. All rights reserved.
//

import UIKit
import Firebase
import GoogleSignIn

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {


     func application(_ application: UIApplication,
      didFinishLaunchingWithOptions launchOptions:
      [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after application launch.
        FirebaseApp.configure()
        GIDSignIn.sharedInstance().clientID =
         FirebaseApp.app()?.options.clientID

        return true
    }

    // MARK: UISceneSession Lifecycle

    func application(_ application: UIApplication,
     configurationForConnecting connectingSceneSession: UISceneSession,
     options: UIScene.ConnectionOptions) -> UISceneConfiguration {
        // Called when a new scene session is being created.
        // Use this method to select a configuration to create the new
         scene with.
        return UISceneConfiguration(name: "Default Configuration",
         sessionRole: connectingSceneSession.role)
    }

    func application(_ application: UIApplication, didDiscardSceneSessions
     sceneSessions: Set<UISceneSession>) {
        // Called when the user discards a scene session.
        // If any sessions were discarded while the application was not
         running, this will be called shortly after
         application:didFinishLaunchingWithOptions.
        // Use this method to release any resources that were specific to
         the discarded scenes, as they will not return.
    }

    func application(_ app: UIApplication, open url: URL, options:
     [UIApplication.OpenURLOptionsKey : Any]) -> Bool {

        return GIDSignIn.sharedInstance().handle(url)
    }
```

```
}
```

```swift
//
//  SceneDelegate.swift
//  specialTopicSem4
//
//  Created by Anantha Krishna on 02/03/20.
//  Copyright © 2020 Anantha Krishna. All rights reserved.
//

import UIKit

class SceneDelegate: UIResponder, UIWindowSceneDelegate {

    var window: UIWindow?


    func scene(_ scene: UIScene, willConnectTo session: UISceneSession,
     options connectionOptions: UIScene.ConnectionOptions) {
        // Use this method to optionally configure and attach the UIWindow
         `window` to the provided UIWindowScene `scene`.
        // If using a storyboard, the `window` property will automatically
         be initialized and attached to the scene.
        // This delegate does not imply the connecting scene or session are
         new (see `application:configurationForConnectingSceneSession`
         instead).
        guard let _ = (scene as? UIWindowScene) else { return }
    }

    func sceneDidDisconnect(_ scene: UIScene) {
        // Called as the scene is being released by the system.
        // This occurs shortly after the scene enters the background, or
         when its session is discarded.
        // Release any resources associated with this scene that can be
         re-created the next time the scene connects.
        // The scene may re-connect later, as its session was not
         neccessarily discarded (see `application:didDiscardSceneSessions`
         instead).
    }

    func sceneDidBecomeActive(_ scene: UIScene) {
        // Called when the scene has moved from an inactive state to an
         active state.
        // Use this method to restart any tasks that were paused (or not
         yet started) when the scene was inactive.
    }

    func sceneWillResignActive(_ scene: UIScene) {
        // Called when the scene will move from an active state to an
         inactive state.
        // This may occur due to temporary interruptions (ex. an incoming
         phone call).
    }

    func sceneWillEnterForeground(_ scene: UIScene) {
        // Called as the scene transitions from the background to the
         foreground.
```

```
        // Use this method to undo the changes made on entering the
         background.
    }

    func sceneDidEnterBackground(_ scene: UIScene) {
        // Called as the scene transitions from the foreground to the
         background.
        // Use this method to save data, release shared resources, and
         store enough scene-specific state information
        // to restore the scene back to its current state.
    }


}
```

```
//
//  FirebaseData.swift
//  specialTopicSem4
//
//  Created by Anantha krishna on 24/03/20.
//  Copyright © 2020 Anantha Krishna. All rights reserved.
//

import Foundation
struct FirebaseData{
    var devicePower:String
    var deviceToggle:String
}
```

```
# Uncomment the next line to define a global platform for your project
platform :ios, '13.0'

target 'specialTopicSem4' do
  # Comment the next line if you don't want to use dynamic frameworks
  use_frameworks!

  # Pods for specialTopicSem4
pod 'Firebase/Database'
pod 'Firebase/Auth'
pod 'GoogleSignIn'
pod 'Charts'
pod 'TinyConstraints'
# add pods for any other desired Firebase products
# https://firebase.google.com/docs/ios/setup#available-pods

end
```