

```

//
// ProfilePageViewController.swift
// apppresent
//
// Created by Macbook Pro on 01/01/2019.
// Copyright © 2019 Macbook Pro. All rights reserved.
//

import UIKit
import FirebaseDatabase
import FirebaseAuth
import FirebaseStorage
import SDWebImage

class ProfilePageViewController: UIViewController,
UICollectionViewDataSource, UICollectionViewDelegate {
    @IBOutlet weak var loadview: UIView!

    var activityindicator: UIActivityIndicatorView =
        UIActivityIndicatorView()

    var datasource = [PostData]() //data structure defined
    //collection = imagecollection
    //collectionViewCell = imageCollectionViewCell
    var ref = Database.database().reference()
    // let userid = Auth.auth().currentUser!.uid

    @IBOutlet weak var collectionview: UICollectionView!
    var customimageflowlayout : CustomImageFlowLayout!
    @IBOutlet weak var profileimageview: UIImageView!
    @IBOutlet weak var displaystatuslbl: UILabel!
    @IBOutlet weak var displaynamelbl: UILabel!

    @IBOutlet weak var displayagelbl: UILabel!
    var age = 0
    var stringage = ""
    var birthday = ""
    var status = ""
    var profileImageUrl = ""
    var firstname = ""
    var lastname = ""
    var fullname = ""
    var selectedimage : UIImage?

    @IBAction func editProfile(_ sender: Any) {
        let controller = EditProfileViewController.create()
        controller.profilePage = self
        present(controller, animated: true)
    }
}

```

```

override func viewWillAppear(_ animated: Bool) {

    super.viewWillAppear(animated)
    startloading()
    loadDB()
    stoploading()
    //startloading()
    // load db again

}

func loadDB() {

    customimageflowlayout = CustomImageFlowLayout()
    collectionview.collectionViewLayout = customimageflowlayout
    collectionview.backgroundColor = .white


    displaynamelbl.preferredMaxLayoutWidth = 150
    displayagelbl.preferredMaxLayoutWidth = 150
    displaystatuslbl.preferredMaxLayoutWidth = 150
    profileimageview.layer.cornerRadius = 40
    profileimageview.clipsToBounds = true


    //works
    //var hello = "hello"
    // self.displayagelbl?.text = hello
    // let ref = Database.database().reference()
    // Do any additional setup after /loading the view.

    let userid = Auth.auth().currentUser?.uid
    ref.child("users").child(userid ??
        "").observeSingleEvent(of: .value, with: { (snapshot) in

        let value = snapshot.value as? NSDictionary

        let firstname = value?["firstname"] as? String ?? ""
        let lastname = value?["lastname"] as? String ?? ""
        let age = value?["age"] as? Int ?? 0
        let status = value?["status"] as? String ?? ""
        let fullname = "\(firstname) \(lastname)"
        let imageurl = value?["profileImageUrl"] as? String ?? ""
        print("This is age" ,age)
        let stringage = String(age)
        print(stringage)
        print(imageurl)
        print("This is status ", status)
        print(" This is fullname " ,fullname)

        self.displaystatuslbl?.text = status
        self.displayagelbl?.text = stringage
        self.displaynamelbl?.text = fullname

        if let url = URL(string: imageurl) {

```

```

        do {
            let imageAsData = try Data(contentsOf: url)
            let image = UIImage(data: imageAsData)
            self.profileimageView.image = image
        } catch {
            print("imageURL was not able to be converted into data")
            // Assert or add an alert
        }

    }
    print(snapshot)

})
/* UP TILL HERE was in view did load*/

// connect to firebase database
var postref = Database.database().reference(withPath: "posts")
postref.observe(DataEventType.value) { (snapshot) in
    self.datasource = snapshot.children.map({ return postData(snapshot:
        $0 as! dataSnapshot) })

    // save to firebase
    //collection is a iterator from first to last element type
    //map is a high order function used on collection types
    //0$ is the data

    // var newposts = [postData].self
    //self.collectionView.reloadData()

}

downloadPost()

}

func startloading () {

    self.view.addSubview(activityindicator)
    activityindicator.center = self.view.center
    activityindicator.hidesWhenStopped = true
    activityindicator.style = UIActivityIndicatorView.Style.gray
    activityindicator.startAnimating()
    UIApplication.shared.beginIgnoringInteractionEvents()
}

func stoploading() {
    activityindicator.stopAnimating()
    UIApplication.shared.endIgnoringInteractionEvents()
}

```

```

func collectionView(_ collectionView: UICollectionView,
    numberOfItemsInSection section: Int) -> Int {
    return datasource.count
}

```

```

func collectionView(_ collectionView: UICollectionView,
                    viewForSupplementaryElementOfKind kind:
                        String,
                    at indexPath: IndexPath) ->
                        UICollectionViewReusableView {
    //1
    switch kind {
    //2
    case UICollectionView.elementKindSectionHeader:
        //3
        let headerView =
            collectionView.dequeueReusableSupplementaryView(ofKind: kind,

                                                            withReuseIdentifier:
                                                                "postHeaderView",

                                                            for: indexPath) as!
                                                                postHeaderView

        headerView.label.text = "header"
        return headerView
    default:
        //4
        assert(false, "Unexpected element kind")
    }
}

```

```

func downloadPost() {

    let poststorage = Storage.storage().reference(withPath: "posts")
    let postdatabase = Database.database().reference(withPath: "posts")
    let currentuser = Auth.auth().currentUser

    var newuserposts = [PostData]() //1
    postdatabase
        .queryOrdered(byChild: "userID").queryEqual(toValue:
            "BRugt1SWc3Ruxq5kyh8M3RG74Dg1") // 2
        .observe(.value, with: { (snapshot: DataSnapshot) in
// 3

            for userpostsnapshot in snapshot.children {
                let userpostobject = PostData(snapshot: userpostsnapshot as!
                    DataSnapshot )
                newuserposts.append(userpostobject)
            }
            // 4
            self.datasource = newuserposts
            self.collectionview.reloadData()
        })
}

```

```

    }

    // click on the image and see it in full view with a caption like instagram.
    //the general feed with the people I follow
    func collectionView(_ collectionView: UICollectionView, didSelectItemAt
        indexPath: IndexPath) {

        let controller = UIStoryboard(name: "profile", bundle:
            nil).instantiateViewController(withIdentifier: "PostDetail") as!
            PostDetail
        let data = datasource[indexPath.row]
        controller.newImageView.sd_setImage(with: URL(string: data.url),
            placeholderImage: UIImage(named:"image1"))

        //controller.captionlbl.text =

        //controller.captionlbl.text = data.
        navigationController?.pushViewController(controller, animated: true)
    }

    func collectionView(_ collection: UICollectionView, cellForItemAt
        indexPath: IndexPath) -> UICollectionViewCell {

        //what it will show image in collection loads
        let cell = collection.dequeueReusableCell(withReuseIdentifier:
            "cell", for: indexPath) as! CollectionViewCell
        //reference cell
        //find right image in array

        let image = datasource[indexPath.row]

        //what is the type of cell
        //loop through every child node and grab - pathToImage
        print("is the url here too?")
        print(image.url)
        cell.imageview.sd_setImage(with: URL(string: image.url),
            placeholderImage: UIImage(named:"image1"))

        return cell
    }

}

```