**T.C. İSTANBUL Okan Üniversitesi**

**Meslek Yüksek Okulu**

**Elektronik ve Otomasyon Bölümü**

**Mobil Teknolojileri Programı**

****

**IOS UYGULAMA GELİŞTİRME**

**PROJE ÖDEVİ**

**PROJE İSMİ: TEST UYGULAMASI**

**Hazırlayan**

**18MY93002 GÜLER FATMA AÇIKGÖZ**

**18MY93003 ÖMER FARUK ERTÜRK**

İçindekiler

[1.xCode Nedir? 3](#_Toc39159849)

[2.Apple’ın Swift Programlama Dili Nedir, Neler Yapılabilir? 4](#_Toc39159850)

[2.1.Swift’in Gelişme Süreci 4](#_Toc39159851)

[2.2.Swift İle Neler Yapılabilir? 4](#_Toc39159852)

[Test Uygulaması Hakkında 4](#_Toc39159853)

[1. PROJENİN KONUSU 4](#_Toc39159854)

[2. PROJENİN AMACI 4](#_Toc39159855)

[3. UYGULAMADA KULLANILAN SAYFALAR 5](#_Toc39159856)

[a. VievController.swift 5](#_Toc39159857)

[b. QuizCVCell.swift 5](#_Toc39159858)

[c. QuizVC.swift 5](#_Toc39159859)

[d. ResultVC.swift 5](#_Toc39159860)

[4. UYGULAMADA KULLANILAN EKRANLAR 5](#_Toc39159861)

[a. GİRİŞ EKRANI 6](#_Toc39159862)

[b. SORU SAYFALARI 6](#_Toc39159863)

[c. SONUÇ SAYFASI 6](#_Toc39159864)

[KODLAR 6](#_Toc39159865)

[a. VievController.swift 6](#_Toc39159866)

[b. QuizCVCell.swift 8](#_Toc39159867)

[c. QuizVC.swift 13](#_Toc39159868)

[d. ResultVC.swift 21](#_Toc39159869)

# 1.xCode Nedir?

  
iOS uygulamaları geliştirmek için bir Mac’e ve Apple Geliştirici Programı’na Apple Kimliğinizi tanımlamış olmamız gerekmektedir. Daha sonrasında eğer yazılım geliştirmek istiyorsak bir geliştirici aracı olan Xcode aracını yüklememiz gerekmektedir.

Apple’ın tasarladığı ve sadece Mac’e yüklenebilen Xcode’da bir uygulama tasarlama-geliştirme ve hata ayıklaması yapılabilmektedir. Xcode uygulama hazırlama esnasında ihtiyaç duyacağınız araçları, kitleri ve derleyicileri içerecek iOS SDK’ya sahiptir.

Xcode, uygulama arayüzü oluşturmak için iki seçenek sunar;

* Programmatic
* Interface Builder: Storyboards ve XML Interface Builder (XIBs)

Her iki seçeneğin çok sayıda artı ve eksileri var. Bunlara değinmenin şimdilik gerektiğini düşünmediğim için bu kısmı geçiyorum.

Xcode içerisinde uygulama geliştirmek için iki adet dil bulunmaktadır : Swift ve Objective-C. Sadece bir dil kullanarak veya her ikisinin bir karışımını kullanarak uygulama geliştirebilirsiniz.



# 2.Apple’ın Swift Programlama Dili Nedir, Neler Yapılabilir?

## 2.1.Swift’in Gelişme Süreci

Apple Worldwide Developers Conference 2014 isimli etkinlikte Swift dilini duyurdu. Apple’ın üst düzey yöneticilerinden biri olan Chris Lattner, programlama dilini 2010 yılında geliştirmeye başladı. Apple 2013 yılına kadar karşılarına çıkması muhtemel bir soruna cevap aradı. O sorun bu döneme kadar uygulama geliştirmek için kullanılmış olan Objective-C ekosisteminde kendine nasıl yer bulacağıydı. Teknoloji devi, geliştiricileri Swift kullanmaya zorlamak istemiyordu. Çünkü böyle bir hareket çok sert tepkiler alabilirdi. Apple bu sebeple mevcut Objective-C dilini aynı şekilde geliştirmeye devam etme kararı aldı. Ancak bir yandan da yeni ve güvenilir bir dil geliştirdiğinin mesajlarını vermeye başladı. Bir yıl sonra ise Apple Swift’in betasını kullanıcılara sundu. Swift, ilk ay 11 milyon kişi tarafından indirildi.

Bir süre sonra Swift dili piyasa da kendine yer etmeye başladı. Görüşler başlarda çok karmaşıktı. İki farklı düşünce vardı. Kullananların yarısı kullanım kolaylığı sebebi ile daha kullanışlı ve daha iyi olduğunu düşünüyor. Diğer yarısı ise kullanım kolaylığından ötürü basit ve yetersiz olduğunu düşünüyordu. Ancak iki taraf da henüz Swift kullanmak için çok **erken olduğu** konusunda ortak fikre sahipti.

2.2.Swift İle Neler Yapılabilir?

Swift, yalnızca Apple ürünlerine uygulama yazmak için kullanılabilmektedir. Yani Swift kullanarak macOS, tvOS, iPadOS, watchOS ve iOS yüklü tüm cihazlarda sorunsuzca çalışabilecek uygulamalar geliştirmek mümkün. Bu konuda yalnızca tek bir istisna bulunuyor.

Swift temelinde açık kaynak kodlu bir programlama dili olduğu için Linux işletim sisteminde çalışabilecek uygulamalar da yazılabilmektedir.

**Linux’ta çalışan uygulamalar**yazmak mümkün olsa da Swift esas olarak Apple ürünlerinde çalışması için üretilmiştir. Bu yüzden Swift, Apple marka ürünlerin donanımlarını en verimli şekilde kullanabilen programlama dili. Yani Apple yıllar boyunca inşa ettiği ekosistemini burada da sürdürmeyi başarıyor.

# Test Uygulaması Hakkında

## PROJENİN KONUSU

Projemiz program üzerinden öğrencilere sınav yapabilmek için hazırlanmıştır.

## PROJENİN AMACI

Projemiz kâğıt üzerinde yapılan hazırlanması, cevaplanması ve kontrol edilmesi saatler sürebilecek sınav işlemlerini daha hızlı bir şekilde gerçekleştirilebilmesi için yapılmıştır. Eğitim verenlerin sadece soruları ve doğru cevapları programa girerek öğrencilere vermesi yeterlidir. Verilen cevapların doğru olup olmadığını ve öğrencinin aldığı puanı programımız otomatik olarak hesaplayabilmektedir.

## UYGULAMADA KULLANILAN SAYFALAR

### VievController.swift

İçerisinde ekranda bulunan label ve buttonun ne yapacağı yazmaktadır. Rengi konumu gibi özellikler bulunmaktadır.

### QuizCVCell.swift

İçerisinde soruların cevaplarının seçilme durumu kontrol edilmektedir. Örneğin bir seçenek seçildiğinde diğer butonlara tıklanma iptal olur.

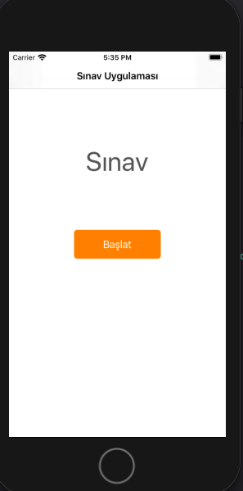
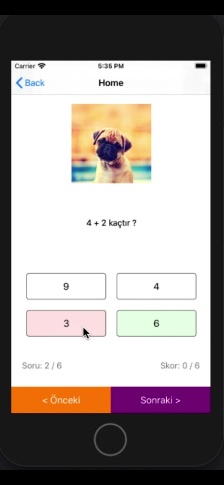
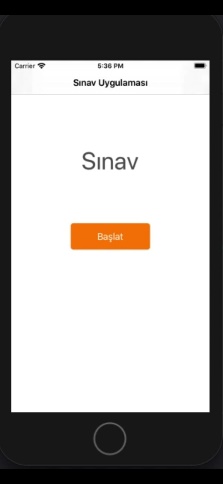
### QuizVC.swift

İçerisinde soru sayfasında olacaklar yazmaktadır. Sorular, soruların konumu ve soruların cevapları gibi işlemler bu sayfa içerisinde gerçekleşmektedir.

### ResultVC.swift

İçerisinde sonuç ekranında neler yapılacağı yazmaktadır. Skorun hesaplanması ve alınan puana göre durumun değerlendirilmesi yapılmaktadır.

## UYGULAMADA KULLANILAN EKRANLAR

### GİRİŞ EKRANI

Giriş ekranında Teste başlama butonu ve ‘Sınav’ yazan bir label bulunmaktadır.

### SORU SAYFALARI

Soru sayfalarında bir tane rastgele görsel, soru ve 4 adet şık bulunmaktadır. Sayfanın sol altında önceki soru butonu sağ altında sonraki soru butonu bulunmaktadır. Bunların hemen üstünde kaçıncı soruda olunduğu ve şu ana kadar kaç tane doğru cevap verildiği yazmaktadır.

### SONUÇ SAYFASI

6 adet sorunun sonunda kaç doğru cevap verildiği yazan bir label ve sınava tekrardan başlamak için de bir adet buton bulunmaktadır.

# KODLAR

### VievController.swift

// 18MY93003 Ömer Faruk ERTÜRK

// 18MY93002 Güler Fatma AÇIKGÖZ

import UIKit

class ViewController: UIViewController {

var window: UIWindow?

override func viewDidLoad() {

super.viewDidLoad()

self.title="Sınav Uygulaması "

self.view.backgroundColor=UIColor.white

setupViews()

}

@objc func btnGetStartedAction() {

let v=QuizVC()

self.navigationController?.pushViewController(v, animated: true)

}

func setupViews() {

self.view.addSubview(lblTitle)

lblTitle.topAnchor.constraint(equalTo: self.view.topAnchor, constant: 150).isActive=true

lblTitle.centerXAnchor.constraint(equalTo: self.view.centerXAnchor).isActive=true

lblTitle.widthAnchor.constraint(equalToConstant: 250).isActive=true

lblTitle.heightAnchor.constraint(equalToConstant: 80).isActive=true

self.view.addSubview(btnGetStarted)

btnGetStarted.heightAnchor.constraint(equalToConstant: 50).isActive=true

btnGetStarted.widthAnchor.constraint(equalToConstant: 150).isActive=true

btnGetStarted.centerXAnchor.constraint(equalTo: self.view.centerXAnchor).isActive=true

btnGetStarted.centerYAnchor.constraint(equalTo: self.view.centerYAnchor, constant: 0).isActive=true

}

let lblTitle: UILabel = {

let lbl=UILabel()

lbl.text="Sınav "

lbl.textColor=UIColor.darkGray

lbl.textAlignment = .center

lbl.font = UIFont.systemFont(ofSize: 46)

lbl.numberOfLines=2

lbl.translatesAutoresizingMaskIntoConstraints=false

return lbl

}()

let btnGetStarted: UIButton = {

let btn=UIButton()

btn.setTitle("Başlat", for: .normal)

btn.setTitleColor(UIColor.white, for: .normal)

btn.backgroundColor=UIColor.orange

btn.layer.cornerRadius=5

btn.layer.masksToBounds=true

btn.translatesAutoresizingMaskIntoConstraints=false

btn.addTarget(self, action: #selector(btnGetStartedAction), for: .touchUpInside)

return btn

}()

}

### QuizCVCell.swift

// 18MY93003 Ömer Faruk ERTÜRK

// 18MY93002 Güler Fatma AÇIKGÖZ

import UIKit

protocol SoruApp: class {

func didChooseAnswer(btnIndex: Int)

}

class SorguUygulamasi: UICollectionViewCell {

var btn1: UIButton!

var btn2: UIButton!

var btn3: UIButton!

var btn4: UIButton!

var btnsArray = [UIButton]()

weak var delegate: SoruApp?

var question: Question? {

didSet {

guard let unwrappedQue = question else { return }

imgView.image = UIImage(named: unwrappedQue.imgName)

lblQue.text = unwrappedQue.questionText

btn1.setTitle(unwrappedQue.options[0], for: .normal)

btn2.setTitle(unwrappedQue.options[1], for: .normal)

btn3.setTitle(unwrappedQue.options[2], for: .normal)

btn4.setTitle(unwrappedQue.options[3], for: .normal)

if unwrappedQue.isAnswered {

btnsArray[unwrappedQue.correctAns].backgroundColor=UIColor.green

if unwrappedQue.wrongAns >= 0 {

btnsArray[unwrappedQue.wrongAns].backgroundColor=UIColor.red

}

}

}

}

override init(frame: CGRect) {

super.init(frame: frame)

setupViews()

btnsArray = [btn1, btn2, btn3, btn4]

}

@objc func btnOptionAction(sender: UIButton) {

guard let unwrappedQue = question else { return }

if !unwrappedQue.isAnswered {

delegate?.didChooseAnswer(btnIndex: sender.tag)

}

}

override func prepareForReuse() {

btn1.backgroundColor=UIColor.white

btn2.backgroundColor=UIColor.white

btn3.backgroundColor=UIColor.white

btn4.backgroundColor=UIColor.white

}

func setupViews() {

addSubview(imgView)

imgView.topAnchor.constraint(equalTo: self.topAnchor, constant: 50).isActive=true

imgView.centerXAnchor.constraint(equalTo: self.centerXAnchor).isActive=true

imgView.widthAnchor.constraint(equalToConstant: 150).isActive=true

imgView.heightAnchor.constraint(equalTo: imgView.widthAnchor).isActive=true

addSubview(lblQue)

lblQue.topAnchor.constraint(equalTo: imgView.bottomAnchor).isActive=true

lblQue.leftAnchor.constraint(equalTo: self.leftAnchor, constant: 12).isActive=true

lblQue.rightAnchor.constraint(equalTo: self.rightAnchor, constant: -12).isActive=true

lblQue.heightAnchor.constraint(equalToConstant: 150).isActive=true

let btnWidth: CGFloat = 150

let btnHeight: CGFloat = 50

btn1 = getButton(tag: 0)

addSubview(btn1)

NSLayoutConstraint.activate([btn1.topAnchor.constraint(equalTo: lblQue.bottomAnchor, constant: 20), btn1.rightAnchor.constraint(equalTo: self.centerXAnchor, constant: -10), btn1.widthAnchor.constraint(equalToConstant: btnWidth), btn1.heightAnchor.constraint(equalToConstant: btnHeight)])

btn1.addTarget(self, action: #selector(btnOptionAction), for: .touchUpInside)

btn2 = getButton(tag: 1)

addSubview(btn2)

NSLayoutConstraint.activate([btn2.topAnchor.constraint(equalTo: btn1.topAnchor), btn2.leftAnchor.constraint(equalTo: self.centerXAnchor, constant: 10), btn2.widthAnchor.constraint(equalToConstant: btnWidth), btn2.heightAnchor.constraint(equalToConstant: btnHeight)])

btn2.addTarget(self, action: #selector(btnOptionAction), for: .touchUpInside)

btn3 = getButton(tag: 2)

addSubview(btn3)

NSLayoutConstraint.activate([btn3.topAnchor.constraint(equalTo: btn1.bottomAnchor, constant: 20), btn3.rightAnchor.constraint(equalTo: self.centerXAnchor, constant: -10), btn3.widthAnchor.constraint(equalToConstant: btnWidth), btn3.heightAnchor.constraint(equalToConstant: btnHeight)])

btn3.addTarget(self, action: #selector(btnOptionAction), for: .touchUpInside)

btn4 = getButton(tag: 3)

addSubview(btn4)

NSLayoutConstraint.activate([btn4.topAnchor.constraint(equalTo: btn3.topAnchor), btn4.leftAnchor.constraint(equalTo: self.centerXAnchor, constant: 10), btn4.widthAnchor.constraint(equalToConstant: btnWidth), btn4.heightAnchor.constraint(equalToConstant: btnHeight)])

btn4.addTarget(self, action: #selector(btnOptionAction), for: .touchUpInside)

}

func getButton(tag: Int) -> UIButton {

let btn=UIButton()

btn.tag=tag

btn.setTitle("Seçenek", for: .normal)

btn.setTitleColor(UIColor.black, for: .normal)

btn.backgroundColor=UIColor.white

btn.layer.borderWidth=1

btn.layer.borderColor=UIColor.darkGray.cgColor

btn.layer.cornerRadius=5

btn.clipsToBounds=true

btn.translatesAutoresizingMaskIntoConstraints=false

return btn

}

let imgView: UIImageView = {

let v=UIImageView()

v.image = #imageLiteral(resourceName: "img2")

v.contentMode = .scaleAspectFit

v.translatesAutoresizingMaskIntoConstraints=false

return v

}()

let lblQue: UILabel = {

let lbl=UILabel()

lbl.text="Bu bir soru ve cevaplamanız gerekiyor?"

lbl.textColor=UIColor.black

lbl.textAlignment = .center

lbl.font = UIFont.systemFont(ofSize: 16)

lbl.numberOfLines=4

lbl.translatesAutoresizingMaskIntoConstraints=false

return lbl

}()

required init?(coder aDecoder: NSCoder) {

fatalError("Hata")

}

}

### QuizVC.swift

// 18MY93003 Ömer Faruk ERTÜRK

// 18MY93002 Güler Fatma AÇIKGÖZ

import UIKit

struct Question {

let Resim: String

let soruYazisi: String

let secenek: [String]

let dogruCevap: Int

var yanlisCevap: Int

var cevap: Bool

}

class SoruFO: UIViewController, UICollectionViewDelegate, UICollectionViewDataSource, UICollectionViewDelegateFlowLayout {

var myCollectionView: UICollectionView!

var soruArray = [Question]()

var skor: Int = 0

var soruNumarasi = 1

var window: UIWindow?

override func viewDidLoad() {

super.viewDidLoad()

self.title="Home"

self.view.backgroundColor=UIColor.white

let layout = UICollectionViewFlowLayout()

layout.sectionInset = UIEdgeInsets(top: 0, left: 0, bottom: 0, right: 0)

layout.itemSize = CGSize(width: self.view.frame.width, height: self.view.frame.height)

layout.scrollDirection = .horizontal

layout.minimumLineSpacing = 1

layout.minimumInteritemSpacing = 1

myCollectionView=UICollectionView(frame: CGRect(x: 0, y: 0, width: self.view.frame.width, height: self.view.frame.height), collectionViewLayout: layout)

myCollectionView.delegate=self

myCollectionView.dataSource=self

myCollectionView.register(QuizCVCell.self, forCellWithReuseIdentifier: "Cell")

myCollectionView.showsHorizontalScrollIndicator = false

myCollectionView.translatesAutoresizingMaskIntoConstraints=false

myCollectionView.backgroundColor=UIColor.white

myCollectionView.isPagingEnabled = true

self.view.addSubview(myCollectionView)

let que1 = Question(Resim: "img1", soruYazisi: "2 x 2 kaçtır ?", secenek: ["2", "4", "8", "6"], dogruCevap: 1, yanlisCevap: -1, cevap: false)

let que2 = Question(Resim: "img2", soruYazisi: "4 + 2 kaçtır ?", secenek: ["9", "4", "3", "6"], dogruCevap: 3, yanlisCevap: -1, cevap: false)

let que3 = Question(Resim: "img1", soruYazisi: "6 / 2 kaçtır ?", secenek: ["2", "4", "3", "5"], dogruCevap: 2, yanlisCevap: -1, cevap: false)

let que4 = Question(Resim: "img2", soruYazisi: "2 - 2 kaçtır ?", secenek: ["2", "4", "1", "0"], dogruCevap: 3, yanlisCevap: -1, cevap: false)

let que5 = Question(Resim: "img1", soruYazisi: "12 x 2 kaçtır ?", secenek: ["24", "40", "26", "34"], dogruCevap: 0, yanlisCevap: -1, cevap: false)

let que6 = Question(Resim: "img2", soruYazisi: "Gökyüzünün rengi nedir?", secenek: ["Mor", "Sarı", "Mavi", "Beyaz"], dogruCevap: 2, yanlisCevap: -1, cevap: false)

soruArray = [que1, que2, que3, que4, que5, que6]

setupViews()

}

func collectionView(\_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {

return soruArray.count

}

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

let cell=collectionView.dequeueReusableCell(withReuseIdentifier: "Cell", for: indexPath) as! QuizCVCell

cell.question=soruArray[indexPath.row]

cell.delegate=self

return cell

}

func scrollViewDidEndDecelerating(\_ scrollView: UIScrollView) {

setQuestionNumber()

}

func setQuestionNumber() {

let x = myCollectionView.contentOffset.x

let w = myCollectionView.bounds.size.width

let currentPage = Int(ceil(x/w))

if currentPage < soruArray.count {

lblQueNumber.text = "Soru: \(currentPage+1) / \(soruArray.count)"

soruNumarasi = currentPage + 1

}

}

@objc func btnPrevNextAction(sender: UIButton) {

if sender == btnNext && soruNumarasi == soruArray.count {

let v=ResultVC()

v.skor = skor

v.totalskor = soruArray.count

self.navigationController?.pushViewController(v, animated: false)

return

}

let collectionBounds = self.myCollectionView.bounds

var contentOffset: CGFloat = 0

if sender == btnNext {

contentOffset = CGFloat(floor(self.myCollectionView.contentOffset.x + collectionBounds.size.width))

soruNumarasi += soruNumarasi >= soruArray.count ? 0 : 1

} else {

contentOffset = CGFloat(floor(self.myCollectionView.contentOffset.x - collectionBounds.size.width))

soruNumarasi -= soruNumarasi <= 0 ? 0 : 1

}

self.moveToFrame(contentOffset: contentOffset)

lblQueNumber.text = "Soru: \(soruNumarasi) / \(soruArray.count)"

}

func moveToFrame(contentOffset : CGFloat) {

let frame: CGRect = CGRect(x : contentOffset ,y : self.myCollectionView.contentOffset.y ,width : self.myCollectionView.frame.width,height : self.myCollectionView.frame.height)

self.myCollectionView.scrollRectToVisible(frame, animated: true)

}

func setupViews() {

myCollectionView.topAnchor.constraint(equalTo: self.view.topAnchor).isActive=true

myCollectionView.leftAnchor.constraint(equalTo: self.view.leftAnchor).isActive=true

myCollectionView.rightAnchor.constraint(equalTo: self.view.rightAnchor).isActive=true

myCollectionView.bottomAnchor.constraint(equalTo: self.view.bottomAnchor).isActive=true

self.view.addSubview(btnPrev)

btnPrev.heightAnchor.constraint(equalToConstant: 50).isActive=true

btnPrev.widthAnchor.constraint(equalTo: self.view.widthAnchor, multiplier: 0.5).isActive=true

btnPrev.leftAnchor.constraint(equalTo: self.view.leftAnchor).isActive=true

btnPrev.bottomAnchor.constraint(equalTo: self.view.bottomAnchor, constant: 0).isActive=true

self.view.addSubview(btnNext)

btnNext.heightAnchor.constraint(equalTo: btnPrev.heightAnchor).isActive=true

btnNext.widthAnchor.constraint(equalTo: btnPrev.widthAnchor).isActive=true

btnNext.rightAnchor.constraint(equalTo: self.view.rightAnchor).isActive=true

btnNext.bottomAnchor.constraint(equalTo: self.view.bottomAnchor, constant: 0).isActive=true

self.view.addSubview(lblQueNumber)

lblQueNumber.heightAnchor.constraint(equalToConstant: 20).isActive=true

lblQueNumber.widthAnchor.constraint(equalToConstant: 150).isActive=true

lblQueNumber.leftAnchor.constraint(equalTo: self.view.leftAnchor, constant: 20).isActive=true

lblQueNumber.bottomAnchor.constraint(equalTo: self.view.bottomAnchor, constant: -80).isActive=true

lblQueNumber.text = "Soru: \(1) / \(soruArray.count)"

self.view.addSubview(lblskor)

lblskor.heightAnchor.constraint(equalTo: lblQueNumber.heightAnchor).isActive=true

lblskor.widthAnchor.constraint(equalTo: lblQueNumber.widthAnchor).isActive=true

lblskor.rightAnchor.constraint(equalTo: self.view.rightAnchor, constant: -20).isActive=true

lblskor.bottomAnchor.constraint(equalTo: lblQueNumber.bottomAnchor).isActive=true

lblskor.text = "Skor: \(skor) / \(soruArray.count)"

}

let btnPrev: UIButton = {

let btn=UIButton()

btn.setTitle("< Önceki", for: .normal)

btn.setTitleColor(UIColor.white, for: .normal)

btn.backgroundColor=UIColor.orange

btn.translatesAutoresizingMaskIntoConstraints=false

btn.addTarget(self, action: #selector(btnPrevNextAction), for: .touchUpInside)

return btn

}()

let btnNext: UIButton = {

let btn=UIButton()

btn.setTitle("Sonraki >", for: .normal)

btn.setTitleColor(UIColor.white, for: .normal)

btn.backgroundColor=UIColor.purple

btn.translatesAutoresizingMaskIntoConstraints=false

btn.addTarget(self, action: #selector(btnPrevNextAction), for: .touchUpInside)

return btn

}()

let lblQueNumber: UILabel = {

let lbl=UILabel()

lbl.text="0 / 0"

lbl.textColor=UIColor.gray

lbl.textAlignment = .left

lbl.font = UIFont.systemFont(ofSize: 16)

lbl.translatesAutoresizingMaskIntoConstraints=false

return lbl

}()

let lblskor: UILabel = {

let lbl=UILabel()

lbl.text="0 / 0"

lbl.textColor=UIColor.gray

lbl.textAlignment = .right

lbl.font = UIFont.systemFont(ofSize: 16)

lbl.translatesAutoresizingMaskIntoConstraints=false

return lbl

}()

}

extension SoruFO: QuizCVCellDelegate {

func didChooseAnswer(btnIndex: Int) {

let centerIndex = getCenterIndex()

guard let index = centerIndex else { return }

soruArray[index.item].cevap=true

if soruArray[index.item].dogruCevap != btnIndex {

soruArray[index.item].yanlisCevap = btnIndex

skor -= 1

} else {

skor += 1

}

lblskor.text = "Skor: \(skor) / \(soruArray.count)"

myCollectionView.reloadItems(at: [index])

}

func getCenterIndex() -> IndexPath? {

let center = self.view.convert(self.myCollectionView.center, to: self.myCollectionView)

let index = myCollectionView!.indexPathForItem(at: center)

print(index ?? "index not found")

return index

}

}

### ResultVC.swift

// 18MY93003 Ömer Faruk ERTÜRK

// 18MY93002 Güler Fatma AÇIKGÖZ

import UIKit

class ResultFO: UIViewController {

var skor: Int?

var toplamSkor: Int?

override func viewDidLoad() {

super.viewDidLoad()

navigationItem.hidesBackButton = true

setupViews()

}

func showRating() {

var rating = ""

var color = UIColor.black

guard let sc = skor, let tc = toplamSkor else { return }

let s = sc \* 100 / tc

if s < 10 {

rating = "Kötü"

color = UIColor.darkGray

} else if s < 40 {

rating = "Ortalama"

color = UIColor.blue

} else if s < 60 {

rating = "İyi"

color = UIColor.yellow

} else if s < 80 {

rating = "Mükemmel"

color = UIColor.red

} else if s <= 100 {

rating = "Muhteşem"

color = UIColor.orange

}

lblRating.text = "\(rating)"

lblRating.textColor=color

}

@objc func btnRestartAction() {

self.navigationController?.popToRootViewController(animated: true)

}

func setupViews() {

self.view.addSubview(lblTitle)

lblTitle.topAnchor.constraint(equalTo: self.view.topAnchor, constant: 80).isActive=true

lblTitle.centerXAnchor.constraint(equalTo: self.view.centerXAnchor).isActive=true

lblTitle.widthAnchor.constraint(equalToConstant: 250).isActive=true

lblTitle.heightAnchor.constraint(equalToConstant: 80).isActive=true

self.view.addSubview(lblskor)

lblskor.topAnchor.constraint(equalTo: lblTitle.bottomAnchor, constant: 0).isActive=true

lblskor.centerXAnchor.constraint(equalTo: self.view.centerXAnchor).isActive=true

lblskor.widthAnchor.constraint(equalToConstant: 150).isActive=true

lblskor.heightAnchor.constraint(equalToConstant: 60).isActive=true

lblskor.text = "\(skor!) / \(toplamSkor!)"

self.view.addSubview(lblRating)

lblRating.topAnchor.constraint(equalTo: lblskor.bottomAnchor, constant: 40).isActive=true

lblRating.centerXAnchor.constraint(equalTo: self.view.centerXAnchor).isActive=true

lblRating.widthAnchor.constraint(equalToConstant: 150).isActive=true

lblRating.heightAnchor.constraint(equalToConstant: 60).isActive=true

showRating()

self.view.addSubview(btnRestart)

btnRestart.topAnchor.constraint(equalTo: lblRating.bottomAnchor, constant: 40).isActive=true

btnRestart.centerXAnchor.constraint(equalTo: self.view.centerXAnchor).isActive=true

btnRestart.widthAnchor.constraint(equalToConstant: 150).isActive=true

btnRestart.heightAnchor.constraint(equalToConstant: 50).isActive=true

btnRestart.addTarget(self, action: #selector(btnRestartAction), for: .touchUpInside)

}

let lblTitle: UILabel = {

let lbl=UILabel()

lbl.text="Puanın"

lbl.textColor=UIColor.darkGray

lbl.textAlignment = .center

lbl.font = UIFont.systemFont(ofSize: 46)

lbl.numberOfLines=2

lbl.translatesAutoresizingMaskIntoConstraints=false

return lbl

}()

let lblskor: UILabel = {

let lbl=UILabel()

lbl.text="0 / 0"

lbl.textColor=UIColor.black

lbl.textAlignment = .center

lbl.font = UIFont.boldSystemFont(ofSize: 24)

lbl.translatesAutoresizingMaskIntoConstraints=false

return lbl

}()

let lblRating: UILabel = {

let lbl=UILabel()

lbl.text="İyi"

lbl.textColor=UIColor.black

lbl.textAlignment = .center

lbl.font = UIFont.boldSystemFont(ofSize: 24)

lbl.translatesAutoresizingMaskIntoConstraints=false

return lbl

}()

let btnRestart: UIButton = {

let btn = UIButton()

btn.setTitle("Başlat", for: .normal)

btn.setTitleColor(UIColor.white, for: .normal)

btn.backgroundColor=UIColor.orange

btn.layer.cornerRadius=5

btn.clipsToBounds=true

btn.translatesAutoresizingMaskIntoConstraints=false

return btn

}()

}