

Proje - Öğün Takibi

Bu projenin amacı `TableView` elemanının kullanımını pekiştirmektir. Yemek listesini gösteren ve bu listeyi üç öğün olarak gruplandıran bir uygulama yapacaksınız.

"Meal Tracker" adında bir "Single View App" projesi oluşturun.

Storyboard'u Düzenlemek

Adım 1: Storyboard'unuzda otomatik olarak beliren `UIViewController` elemanını silin. Onun yerine `UITableViewController` elemanını ekleyin.

Adım 2: Elemanınıza `Navigation Controller` ekleyin. Başlangıç ekranı olarak ayarlayın.

Adım 3: `Table View` elemanınızın otomatik olarak bir prototip hücresi olacaktır. Bu hücrenin "Style" özelliğini "Subtitle" olarak ayarlayın.

Adım 4: "reuse identifier" tanımını "foodCell" olarak değiştirin.

Adım 5: `Table View` elemanını seçin ve "Style" özelliğini "Grouped" olarak ayarlayın.

Adım 6: `UITableViewController` elemanına `FoodTableViewController` adında yeni bir sınıf (kod dosyası) ekleyin.

Örnek Objeler Oluşturmak

Adım 7: İki tane Swift dosyası oluşturun. Birinin adını "Meal" (öğün) diğerrinin adını "Food" (yemek) yapın.

Adım 8: "Food" dosyasının içinde bir tane "Food" struct'ı oluşturun. Struct'ınızın iki tane değişkeni olsun: "name" (ad) ve "description" (tanım). İkisi de `String` türünden olsun.

Adım 9: "Meal" dosyasının içinde bir tane "Meal" struct'ı oluşturun. Struct'ınızın iki tane değişkeni olsun: "name" (ad) ve "food" (yemek). "name" değişkeni `String` türünden, "food" değişkeni ise `[Food]` türünden olmalıdır.

`UITableViewDataSource` Elemanını Kullanmak

Adım 10: `FoodTableViewController` kod dosyasının içinde bulunan tüm kodları silin. Sadece belirtilen kodlar kalsın: `numberOfSections(in:)`, `tableView(_:numberOfRowsInSection)`, `tableView(_:cellForRowAt:)` ve `tableView(_:titleForHeaderInSection)`.

Adım 11: Başlangıç değeri olmayan bir "meals" değişkeni oluşturun. Bu değişken `[Meals]` türünden olsun.

Adım 12: Değişkenin gövdesinde üç tane `Meal` objesi oluşturun. Bu objeleri "breakfast", "lunch" ve "dinner" olarak adlandırın.

Adım 13: Her `Meal` objesi için bir `Food` objesi oluşturun. Bu objeler önceden tanımladığınız "food" arrayinden gelsin. Her `Food` objesi için bu objenin "meal" özelliğine göre bir ad verin.

Adım 14: Üç tane `Meal` objesini array halinde return edin.

Adım 15: `numberOfSections(in:)` fonksiyonunun içinde öğün sayınızı "meal" arrayi halinde return edin.

Adım 16: `tableView(_:numberOfRowsInSection:)` fonksiyonunun içinde belirlenen bölüm için tanımlanan "meal" değişkenine erişin ve o öğünün içinde "food" elemanlarının sayısını return edin.

Adım 17: `tableView(_:cellForRowAt:)` fonksiyonunun içinde "foodCell" adındaki hücreyi ayırın.

Adım 18: Return edilen "meal" değişkeninden karşılık gelen "food" değişkenlerine `indexPath.section` kullanarak erişin. Bu listenizin sırasını ayarlayacaktır.

Adım 19: Sıraya konulan "food" değişkenlerine erişebilmek için `indexPath.row` kodunu kullanın.

Adım 20: Hücrenizin başlığını ve alt başlığını "name" değişkeni ve bu değişkene uyan "description" değişkeni olarak ayarlayın.

Adım 21: `tableView(_:titleForHeaderInSection:)` fonksiyonunun içinde belirtilen bölüme karşılık gelen "meal" değişkeninin "name" özelliğini return edin.

Tebrikler 🎉👏 Bir `Table View` elemanının içinde günlük öğünleri ve öğünlere karşılık gelen yemekleri gösteren bir uygulama yaptınız. Projenizi kaydedebilir ve gelecekteki projelerinizi yaparken örnek alabilirsiniz.