

iOS Labor 4 - UINavigationController, TabView és Sheet

Ez az útmutató végigvezet egy komplett SwiftUI alkalmazás felépítésén, amely bemutatja a **TabView**, **NavigationStack** és a **sheet** modális ablak használatát. Az alkalmazás egy "Kedvenc Helyeim" lista lesz.

1. feladat: BestPlaces

Nyisd meg az Xcode-ot és a File > New > Project... menüben válaszd az iOS > App sablont. Nevezzük el az alkalmazást **BestPlaces**-nak!

A Model

Hozz létre egy új Swift fájlt **Place.swift** néven.

Definiálj benne egy **Place** nevű **struct**-ot a következő nevű és típusú property-kkel

- **id** (UUID)
- **name** (String)
- **description** (String)
- **icon** (String)

A **Place** feleljen meg az **Identifiable** és **Hashable** protokolloknak.

TabView

Most az alkalmazás fő, "tab-füles"  szerkezetét alakítjuk ki.

Nyisd meg a **ContentView.swift** fájlt

Cseréld le a meglévő **VStack**-et egy **TabView**-ra.

Helyezz el két ideiglenes nézetet (pl. **Text**) a **TabView**-n belül, és mindegyikhez adj hozzá egy **.tabItem** módosítót:

- Első tab: **Image(systemName: "map.fill")** ikonnal és **"Helyek"** *label*-lel.
- Második Tab: **Image(systemName: "person.fill")** ikonnal és **"Profil"** *label*-lel.

NavigationStack

Most pedig használjuk a **NavigationStack**-et a hierarchikus navigációhoz.

Hozz létre egy új **SwiftUI View** fájlt **PlacesListView.swift** néven.

Hozz létre egy **@State** property-t **places** néven, ami egy **[Place]** tömb.

Töltsd fel néhány példa adattal

```
@State private var places: [Place] = [  
    Place(name: "Eiffel-torony", description: "Párizs ikonikus tornya.",  
    icon: "🗼"),  
    Place(name: "Szabadság-szobor", description: "Neoklasszicista szobor
```

```
New York kikötőjében.", icon: "🗽"),
    Place(name: "Kolosszeum", description: "Ovális amfiteátrum Róma
központjában.", icon: "🏛️")
]
```

Hozz létre egy másik `@State` property-t, aminek a neve legyen `showingAddPlaceSheet`, a típusa `Bool` és a kezdőértéke `false`

A nézet `body`-ját (a `VStack`-et) cseréld le egy `NavigationStack`-re

A `NavigationStack`-en belül hozz létre egy `List`-et, ami a `places` tömbön iterál.

```
NavigationStack {
    List(places) { place in
        ...
    }
}
```

A lista minden eleméhez használj `NavigationLink(value: place)`-et. A link `label`-jeként jelenítsd meg a hely ikonját és nevét

```
List(places) { place in
    NavigationLink(value: place) {
        HStack {
            Text(place.icon)
            Text(place.name)
        }
    }
}
```

Érdemes kicsit csinosítani a megjelenésen, például a `font(...)`, a `foregroundColor(...)`, stb. modifier-ek használatával

A `List` után add hozzá a `.navigationDestination(for: Place.self)` modifier-t

Ez kezeli, hogy hova navigáljon az app egy `Place` objektumra kattintva

A `destination` blokkba egyelőre egy ideiglenes `Text("Részletek oldal")` kerüljön

Adj a `List`-hez egy `.navigationTitle("Kedvenc Helyeim")` modifier-t

Ezután adj hozzá (szintén a `List`-hez) egy `.toolbar`-t egy `Button`-nal. Ehhez egy `ToolbarItem`-et is létre kell hozni, aminek a `placement` paraméterének be kell állítani, hogy hová szeretnénk elhelyezni az elemet.

A `placement` egyenlőre legyen `.navigationBarTrailing`. iOS 26-tól viszont `.topBarTrailing`-et ajánlott már használni.

```
.toolbar {
    ToolbarItem(placement: .navigationBarTrailing) {
        Button{
            ...
        } label: {
            ...
        }
    }
}
```

A gomb `label`-je legyen egy plusz jel (`Image(systemName: "plus")`), és a lenyomására a `showingAddPlaceSheet` értéke váltson `true`-ra

Menj vissza a `ContentView.swift`-be, és az első fül ideiglenes `Text`-jét cseréld le a `PlacesListView()`-ra.

További nézetek

Most elkészítjük a hiányzó képernyőket.

PlaceDetailView

Hozz létre egy új **SwiftUI View** fájlt `PlaceDetailView.swift` néven

Adj a nézethez egy `place` nevű, `Place` típusú property-t

Egy `VStack` segítségével jelenítsd meg a kapott `place` ikonját, nevét és leírását

A `PlacesListView.swift`-ben a `.navigationDestination` módosítóban cseréld le az ideiglenes `Text`-et `PlaceDetailView(place: place)`-re.

AddPlaceView

Hozz létre egy új **SwiftUI View** fájlt `AddPlaceView.swift` néven

Ezen a nézeten lehet majd új helyet felvenni

Hozz létre `@State` property-eket az űrlap mezőinek (`name`, `description`, `icon`)

Adj hozzá egy `@Binding var places: [Place]` property-t, hogy a nézet módosítani tudja az eredeti listát

Adj hozzá egy `@Environment(\.dismiss)` változót a nézet bezárásához

```
@Environment(\.dismiss) var dismiss
```

A `body` tartamát csomagold be egy `NavigationStack`-be, hogy legyen fejléce

Hozz létre egy `Form`-ot, amibe helyezz el `TextField`-eket, amiket a helyi `@State` változókhoz kötsz.

```

NavigationStack {
    Form {
        Section(header: Text("Hely adatai")) {
            TextField("Hely neve", text: $name)
            TextField("Leírás", text: $description)
            TextField("Ikon (emoji)", text: $icon)
        }
    }
}

```

A `Form`-hoz adjunk hozzá egy `.navigationBarItems` modifier-t, amire helyezz el két gombot

```

.navigationBarItems(
    leading: Button("Mégse") {
        ...
    },
    trailing: Button("Mentés") {
        ...
    }
)

```

A "Mégse" gomb hívja meg a `dismiss()`-t

A "Mentés" gomb hozzon létre egy új `Place`-t, amit adjon hozzá a `places` tömbhöz, majd hívja meg szintén a `dismiss()`-t

A `PlacesListView.swift`-ben, a `NavigationStack`-en belül adj hozzá egy `.sheet(isPresented: $showingAddPlaceSheet)` modifier-t

A `content` blokkba add meg az `AddPlaceView(places: $places)`-t.

```

.sheet(isPresented: $showingAddPlaceSheet) {
    AddPlaceView(places: $places)
}

```

ProfileView

Végül elkészítjük a második tab tartalmát.

Hozz létre egy új **SwiftUI View** fájlt `ProfileView.swift` néven

Építs fel egy egyszerű profilt egy `VStack`-ben egy `Image(systemName: "person.crop.circle.fill")` ikon és néhány `Text` elem segítségével

A `ContentView.swift`-ben a második tab ideiglenes `Text`-ét cseréld le egy `NavigationStack`-re, ami a `ProfileView()`-t tartalmazza. Ez biztosít egy címsort a profiloldalnak.

A `ProfileView.swift`-ben adj a nézethez egy `.navigationTitle(...)` modifier-t megfelelő tartalommal.

Ezzel a feladat kész. Futtasd az alkalmazást, és teszteld a működését!

2. feladat

Hozz létre egy harmadik tab-ot, amin a kerülendő helyeket tudod nyilván tartani egy listában. Használhatod hozzá az előző labor anyagát is.