

What's new on iPadOS 18

By Paul Peelen

Hi! I'm Paul!

iOS Developer since ~2009

Made a lot of apps, stopped counting at ~30

Worked as freelance dev for many years

Full time Staff Engineer at PayPal



Co-organiser at Cocoaheads Stockholm



Hi! I'm Paul!



From Nijmegen, the Netherlands

Moved to Stockholm, Sweden 23 years ago



Getting started

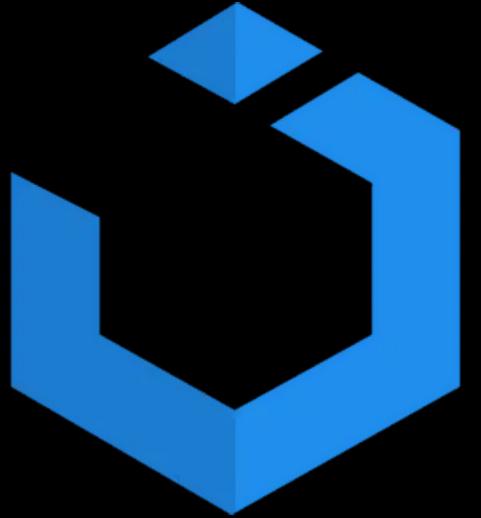
Get the workshop folder from USB

Open the project SwiftIslandApp.xcodeproj

If you like, perform a git init in the project to follow changes

Good to know

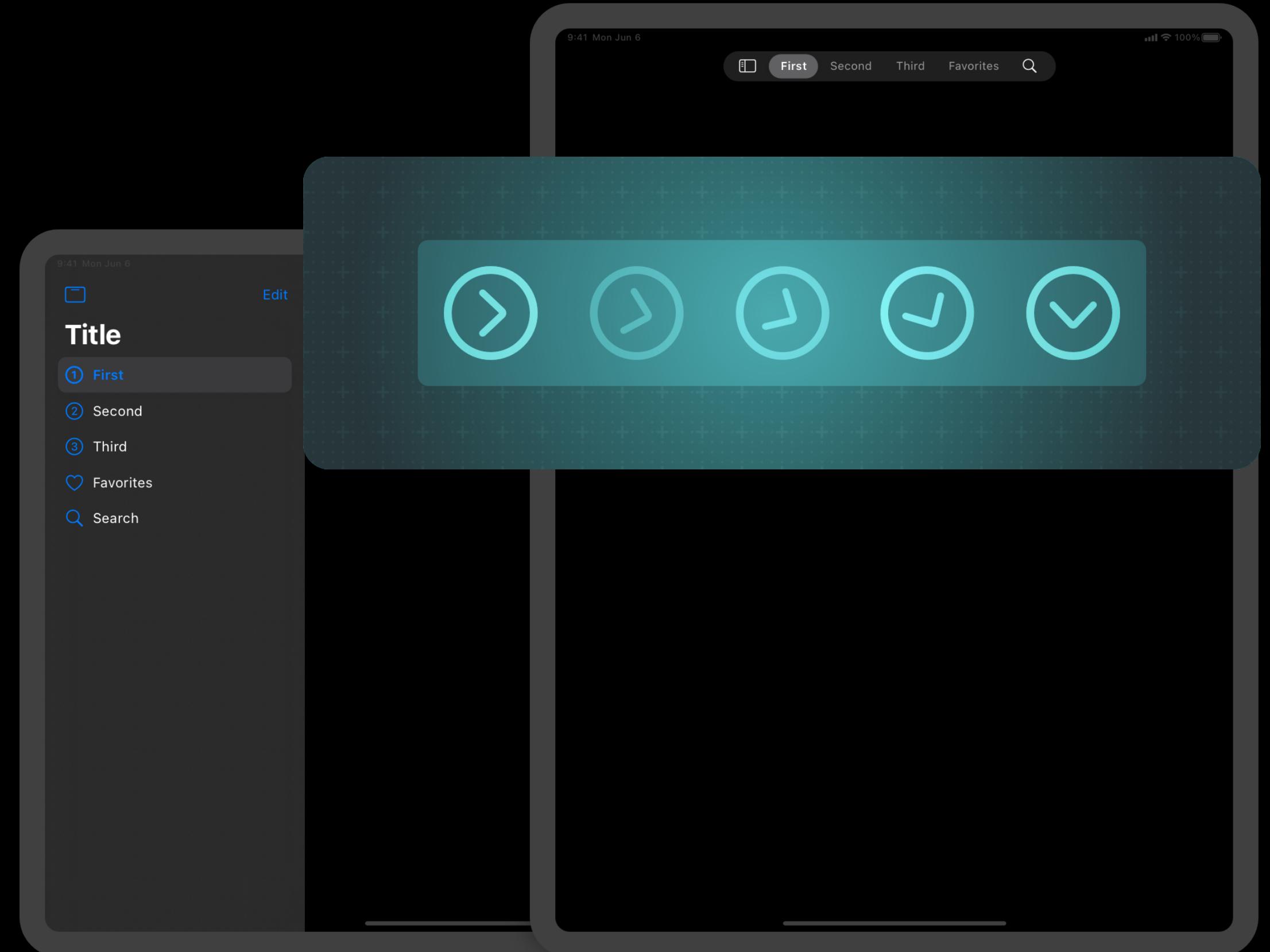
This workshop uses SwiftUI for code examples, however the API changes are also available in UIKit



iPadOS 18

Redesigned TabBar and Sidebar

Zoom animation



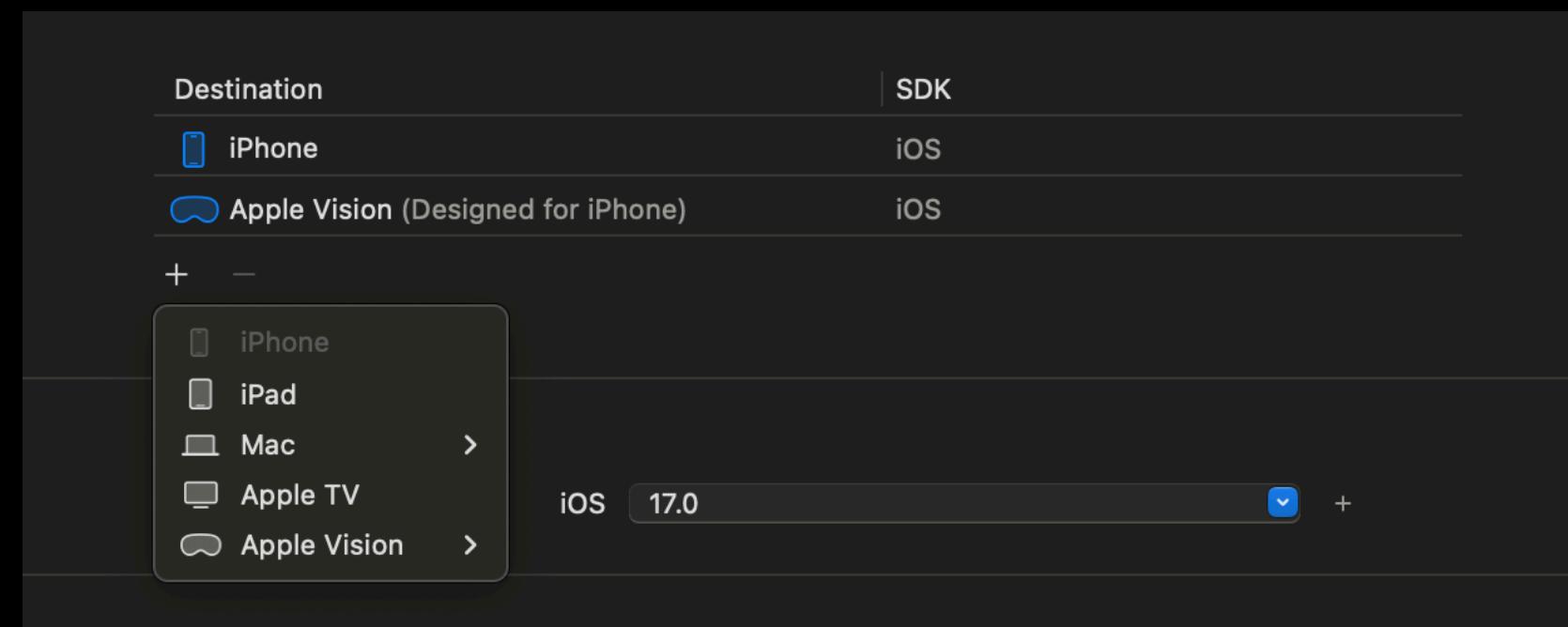
iPadOS 18

Redesigned TabBar and Sidebar

Zoom animation

Redesigned TabBar and Sidebar

Lets start off with adding iPad support.



Redesigned TabBar and Sidebar

Lets launch the app for iPad on iPadOS 17, select e.g. iPad Pro 13-inch (M4)

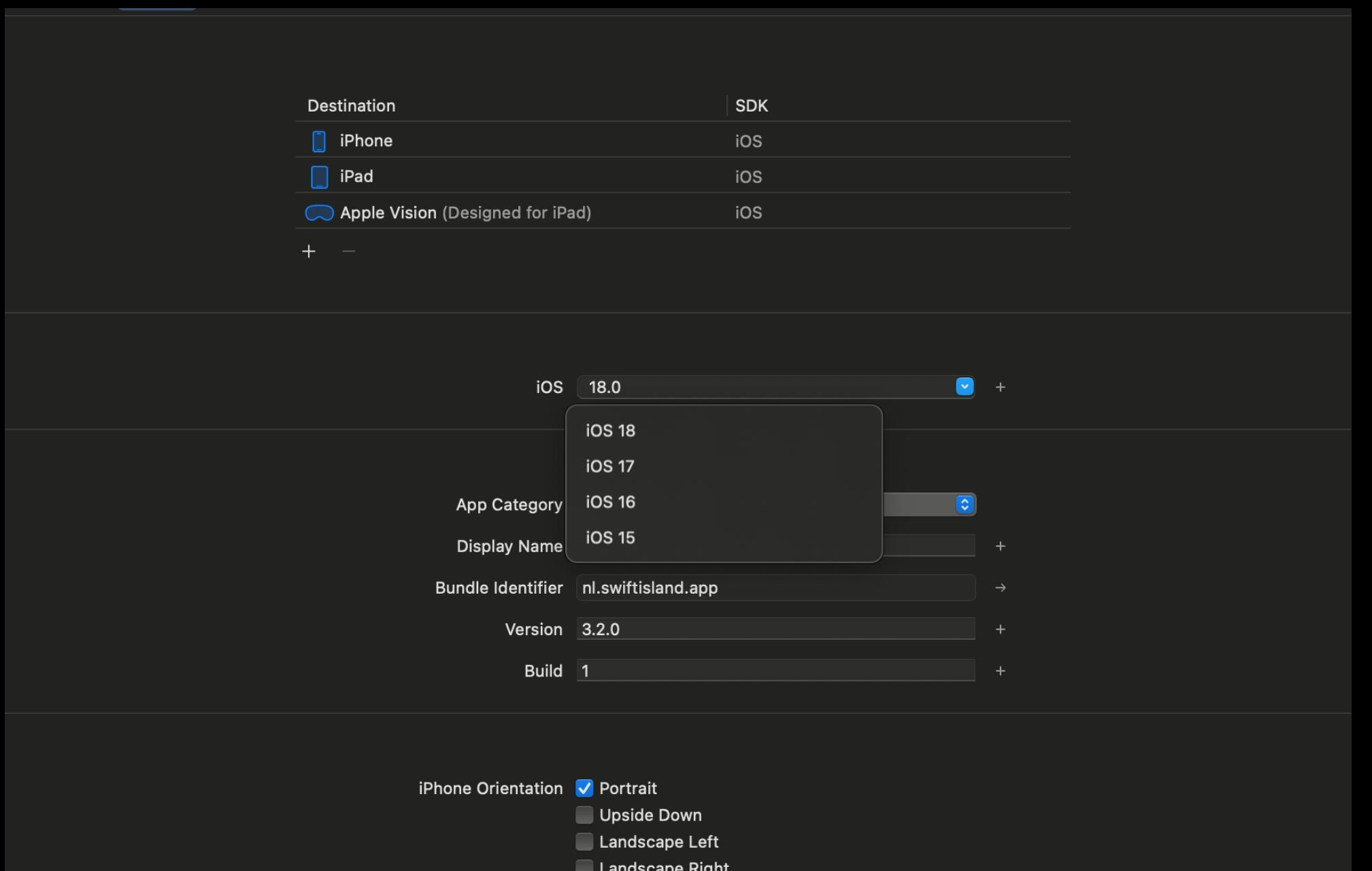
The new TabBar style is used on iOS 18, not on iOS 17

Now lets launch it for iOS18

Looks the same on iOS as before

Redesigned TabBar and Sidebar

For ease sake, lets set the project to iOS 18.



Redesigned TabBar and Sidebar

Lets start with opening TabBarView.swift.

Rename Tab enum to “Tabs” since “Tab” is a structure in SwiftUI now.

New syntax for TabView. Lets update TabBarView.swift:

```
TabView(selection: $selectedItem) {
    Tab("Conference", systemImage: "person.3", value: .home) {
        ConferencePageView(namespace: namespace, isShowingMentor: $isShowingMentor)
    }

    Tab("Practical", systemImage: "wallet.pass", value: .practical) {
        PracticalPageView()
    }

    Tab("Schedule", systemImage: "calendar.day.timeline.left", value: .schedule) {
        NavigationStack {
            ScheduleView()
        }
    }
}
```

You can remove the tabs array at line 28

Redesigned TabBar and Sidebar

Selection is used for programatically changing tabs:

```
TabView(selection: $selectedItem) {
    Tab("Conference", systemImage: "person.3", value: .home) {
        ConferencePageView(namespace: namespace, isShowingMentor: $isShowingMentor)
    }

    Tab("Practical", systemImage: "wallet.pass", value: .practical) {
        PracticalPageView()
    }

    Tab("Schedule", systemImage: "calendar.day.timeline.left", value: .schedule) {
        NavigationStack {
            ScheduleView()
        }
    }
}
```

Redesigned TabBar and Sidebar

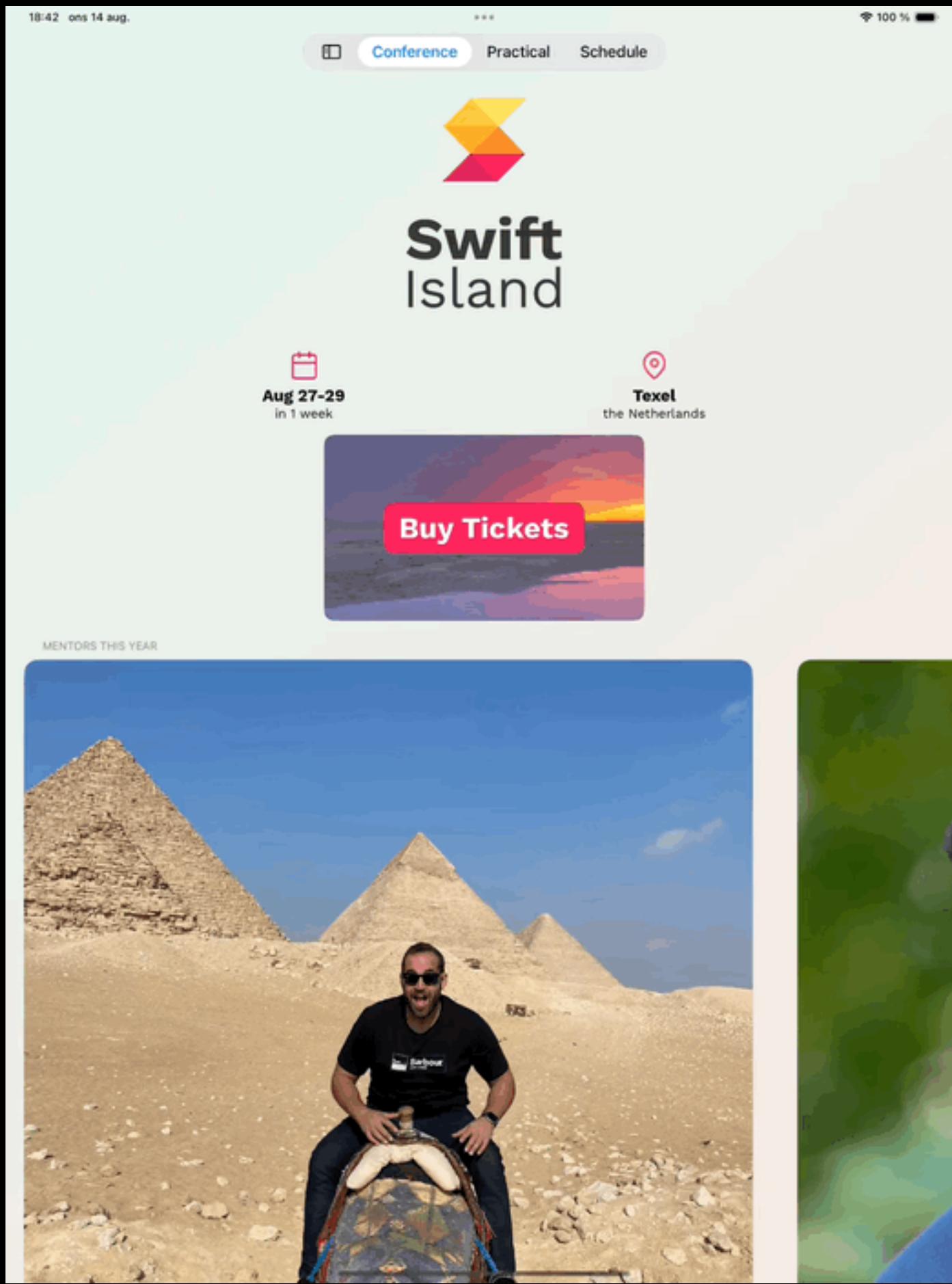
Enable sidebar support:

```
TabView(selection: $selectedItem) {
    Tab("Conference", systemImage: "person.3", value: .home) {
        ConferencePageView(namespace: namespace, isShowingMentor: $isShowingMentor)
    }

    Tab("Practical", systemImage: "wallet.pass", value: .practical) {
        PracticalPageView()
    }

    Tab("Schedule", systemImage: "calendar.day.timeline.left", value: .schedule) {
        NavigationStack {
            ScheduleView()
        }
    }
}.tabViewStyle(.sidebarAdaptable)
```

Redesigned TabBar and Sidebar



Sections in Sidebar

Add mentors item to tabbar

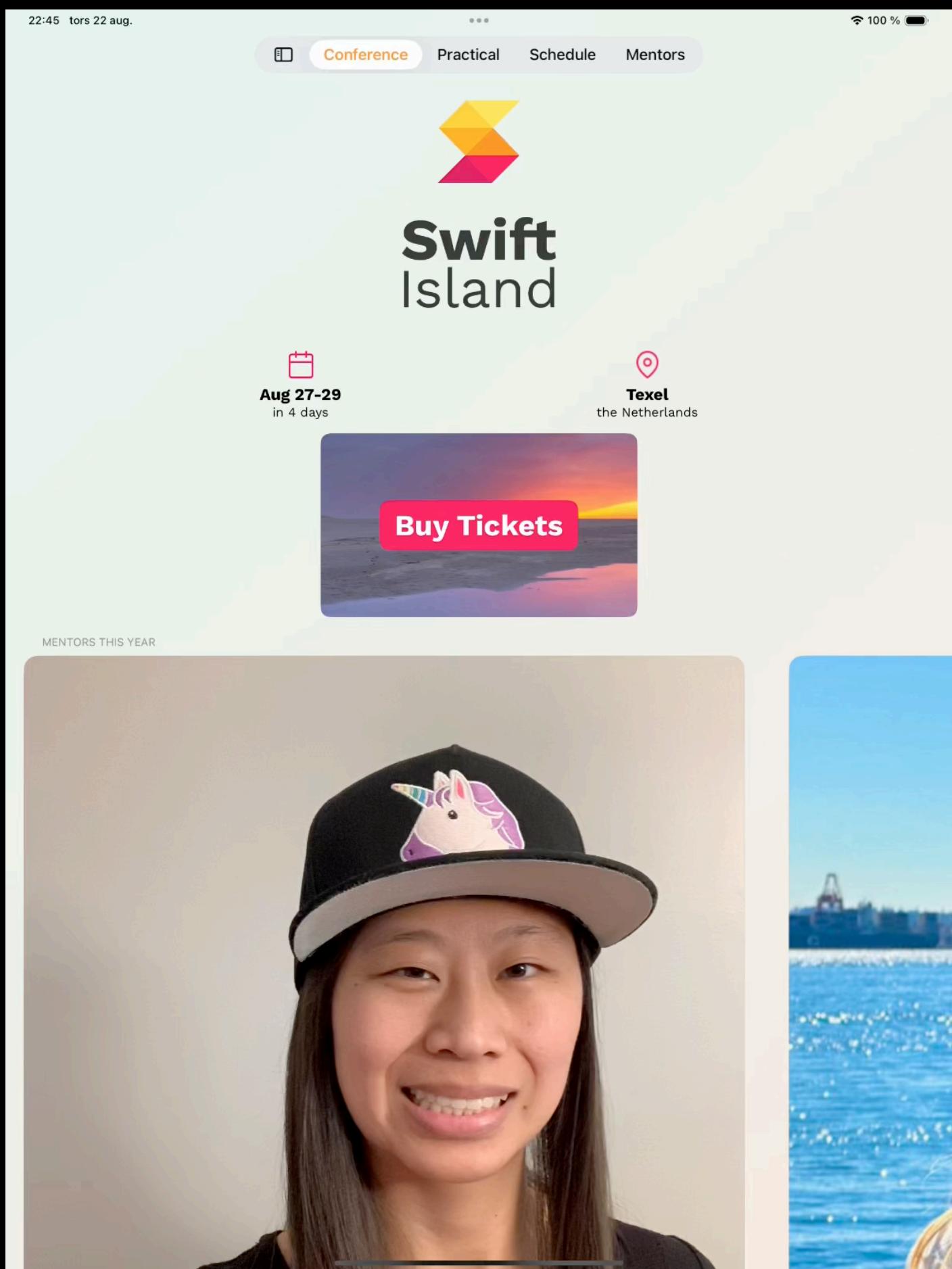
Add a new tabs item to the enum

```
private enum Tabs {  
    case home  
    case practical  
    case schedule  
    case mentors  
}
```

Add the tab

```
@State var mentorPath: NavigationPath = .init()  
  
// ...  
  
Tab("Mentors", systemImage: "graduationcap", value: .mentors) {  
    NavigationStack(path: $mentorPath) {  
        MentorListView()  
        .navigationDestination(for: Mentor.self) { mentor in  
            MentorPageView(mentor: mentor)  
        }  
    }  
}
```

Sections in Sidebar



Sections in Sidebar

Update the tabs for mentor to define a selected mentor

```
private enum Tabs: CaseIterable {
    case home
    case practical
    case schedule
    case mentors(Mentor)
}
```

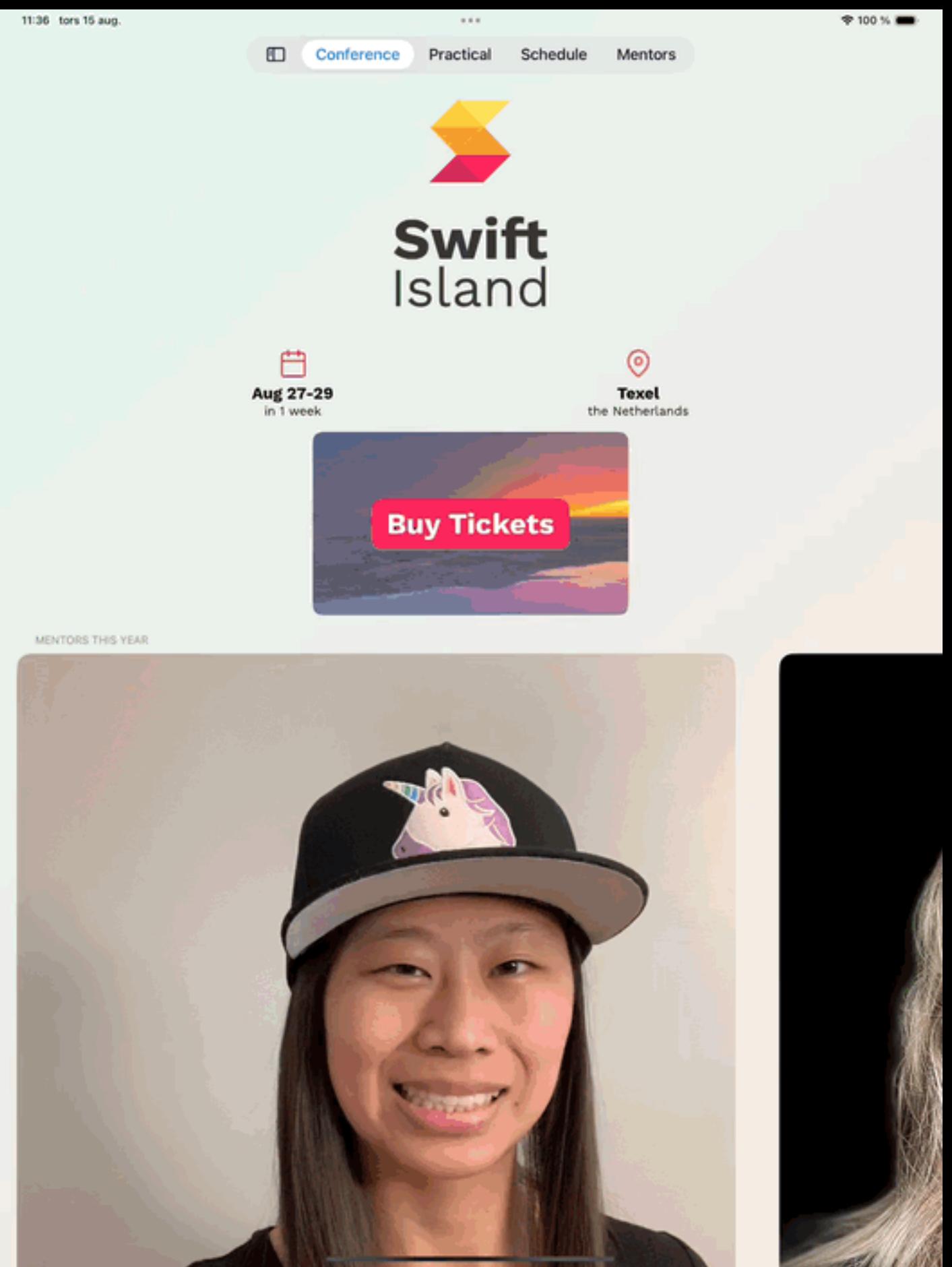
Replace Tab with TabSection

```
TabSection {
    ForEach(appDataModel.mentors) { mentor in
        Tab(mentor.name, systemImage: "graduationcap", value: Tabs.mentors(mentor)) {
            NavigationStack(path: $mentorPath) {
                MentorPageView(mentor: mentor)
            }
        }
    }
} header: {
    Label("Mentors", systemImage: "graduationcap")
}
```

Add Hashable support to Tabs

```
extension Tabs: Hashable {
    var id: String {
        switch self {
            case .home:
                return "home"
            case .practical:
                return "practical"
            case .schedule:
                return "schedule"
            case .mentors(let mentor):
                return mentor.id
        }
    }
}
```

Sections in Sidebar

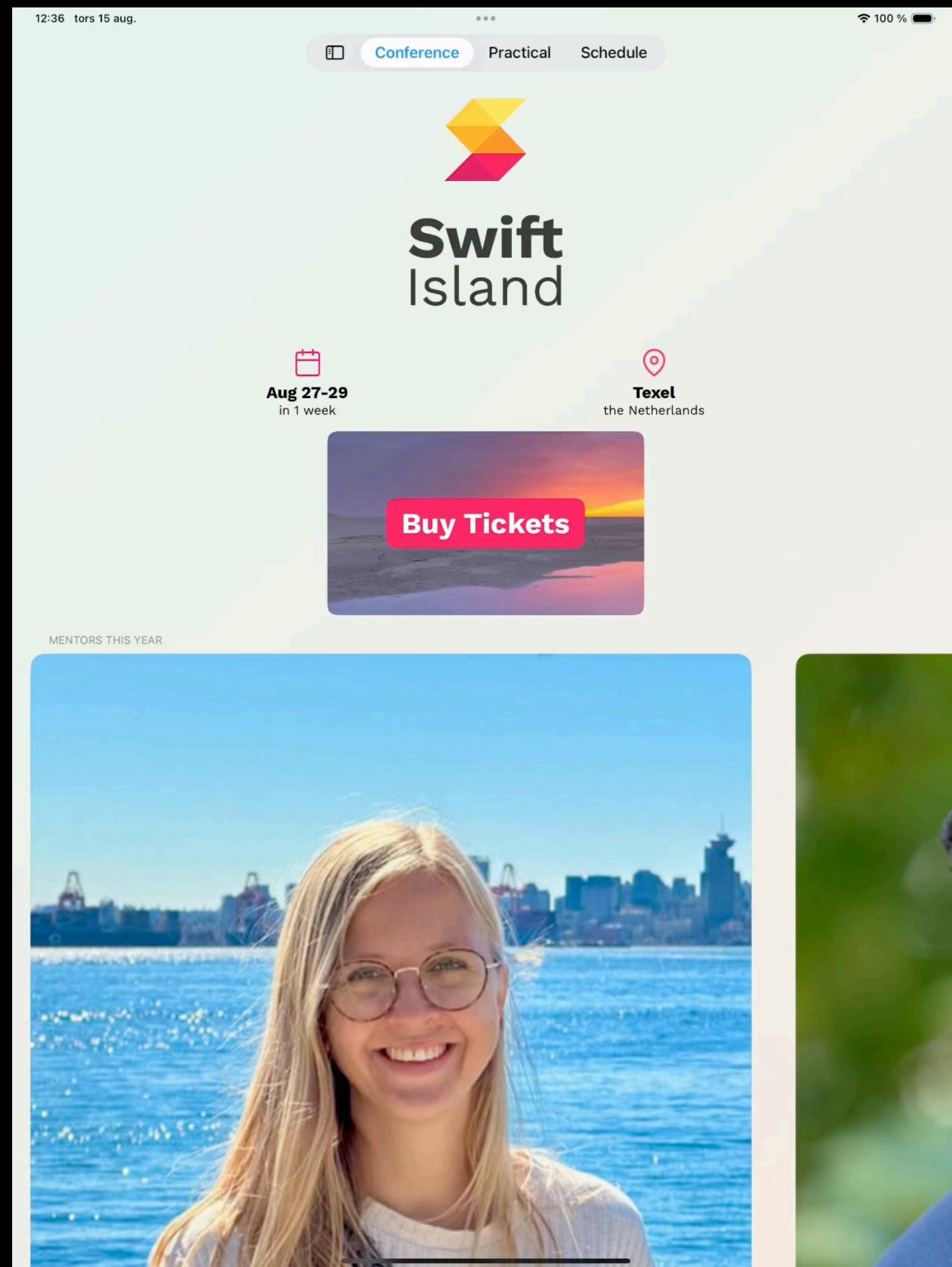


Sections in Sidebar

Hide the menu option from the TabBar

```
TabSection {  
    ForEach(appDataModel.mentors) { mentor in  
        Tab(mentor.name, systemImage: "graduationcap", value: Tabs.mentors(mentor)) {  
            NavigationStack(path: $mentorPath) {  
                MentorPageView(mentor: mentor)  
            }  
        }  
    }  
} header: {  
    Label("Mentors", systemImage: "graduationcap")  
}  
.defaultVisibility(.hidden, for: .tabBar)
```

Sections in Sidebar



User Customization

Customization allows users to:

- Hide tabs

- Reorder of groups

- Persist changes

User Customization

Start off with adding an AppStorage for persisting changes

```
struct TabBarView: View {  
    ...  
  
    @AppStorage("sidebarCustomizations") var tabViewCustomization: TabViewCustomization  
  
    ...  
}
```

Add TabView customization to the TabView

```
ZStack {  
    TabView(selection: $selectedItem) {  
        ...  
    }  
    .tabViewStyle(.sidebarAdaptable)  
    .tabViewCustomization($tabViewCustomization)  
}
```

User Customization

Add CustomizationId variable

```
private enum Tabs {
    case home
    case practical
    case schedule
    case mentorList
    case mentors(Mentor)

    var customizationID: String {
        switch self {
        case .mentors(let mentor):
            mentor.customizationID
        default:
            "SwiftIsland.\(self.hashValue)"
        }
    }
}

extension Mentor {
    var customizationID: String {
        "SwiftIsland.mentor." + self.name
    }
}
```

Disable customization behaviour

```
.customizationBehavior(.disabled, for: .sidebar, .tabBar)
```

Add CustomizationId to tabs

```
Tab("Conference", systemImage: "person.3", value: .home) {
    ConferencePageView(namespace: namespace, isShowingMentor: $isShowingMentor)
}

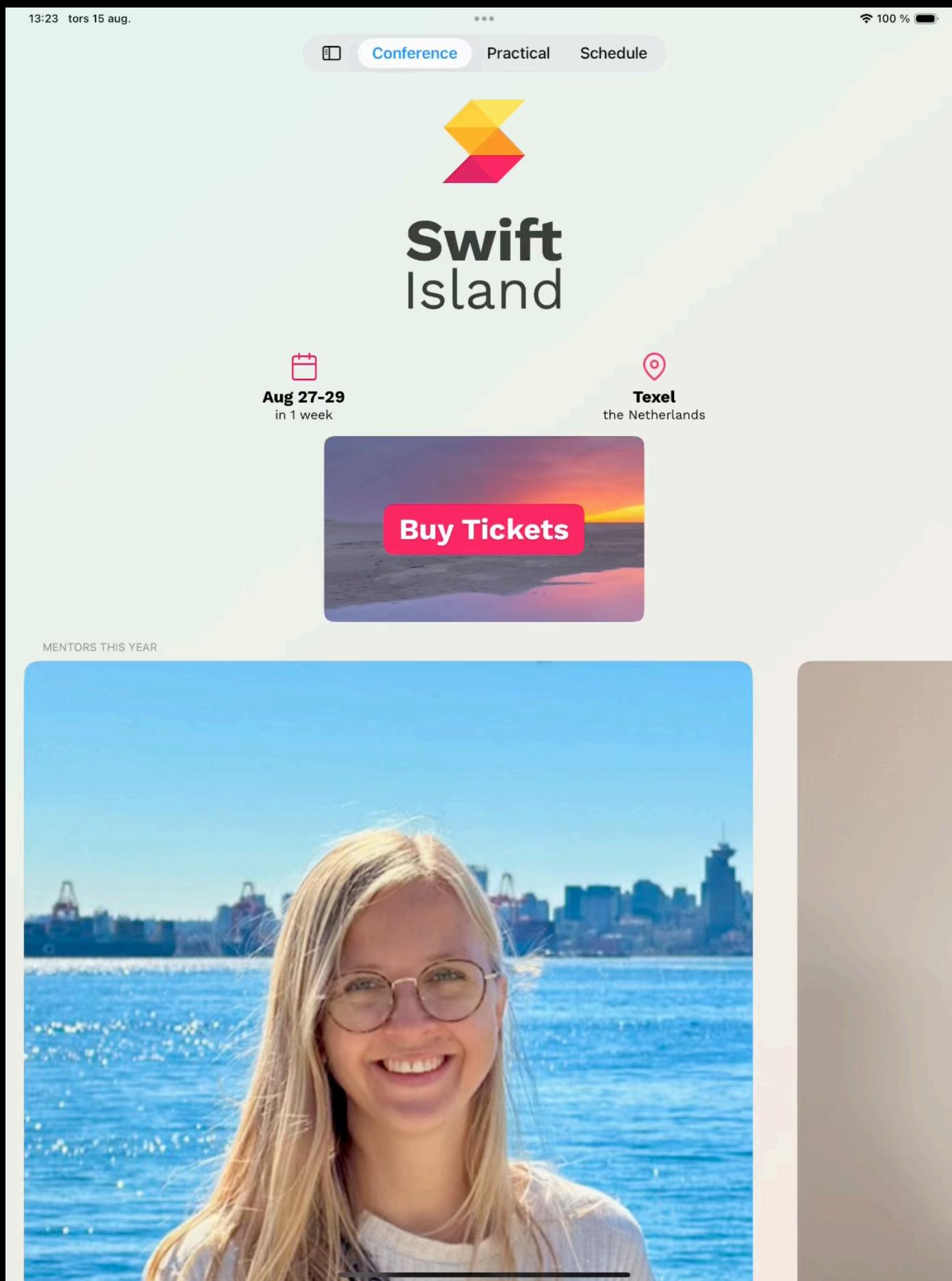
Tab("Practical", systemImage: "wallet.pass", value: .practical) {
    PracticalPageView()
    .customizationID(Tabs.practical.customizationID)
}

Tab("Schedule", systemImage: "calendar.day.timeline.left", value: .schedule) {
    NavigationStack {
        ScheduleView()
    }
    .customizationID(Tabs.schedule.customizationID)
}

ForEach(appDataModel.mentors) { mentor in
    Tab(mentor.name, systemImage: "graduationcap", value: Tabs.mentors(mentor)) {
        NavigationStack(path: $mentorPath) {
            MentorPageView(mentor: mentor)
        }
    }
    .customizationID(mentor.customizationID)
}
```

Notice that “Conference” did not get an ID
Why not?

User Customization



User Customization

Add order support

```
TabSection {  
    // ...  
} header: {  
    Label("Mentors", systemImage: "graduationcap")  
}  
.defaultVisibility(.hidden, for: .tabBar)  
.customizationID("mentorList")
```

Sidebar modifications

Enable search support:

```
private enum Tabs {  
    case home  
    case practical  
    case schedule  
    case mentors(Mentor)  
    case search  
}  
  
Tab(value: .search, role: .search) {  
    MentorListView()  
}
```

Search will always be pinned to the end of the sidebar and tabbar list

Sidebar modifications

Actions

```
TabSection {  
    // Tabs for section ...  
} header: {  
    Label("Mentors", systemImage: "graduationcap")  
}  
.defaultVisibility(.hidden, for: .tabBar)  
.customizationID("mentorList")  
.sectionActions {  
    Button {  
        // Logic for sending an email  
    } label: {  
        Label("Apply to be a mentor", systemImage: "square.and.pencil")  
    }  
}
```

Drop destination

```
@State private var favoriteMentors: [Mentor] = []  
  
TabSection {  
    ForEach(favoriteMentors) { mentor in  
        Tab(mentor.name, systemImage: "graduationcap", value: Tabs.mentors(mentor)) {  
            NavigationStack(path: $mentorPath) {  
                MentorPageView(mentor: mentor)  
            }  
        }  
    }  
.customizationID("favMentor.\\"(mentor.customizationID)")  
}  
  
Tab("Mentor list", systemImage: "graduationcap", value: .mentorList) {  
    NavigationStack {  
        MentorListView()  
    }  
}.customizationID(Tabs.mentorList.customizationID)  
.customizationBehavior(.disabled, for: .sidebar, .tabBar)  
} header: {  
    Label("Favorite mentors", systemImage: "heart")  
}  
.customizationBehavior(.disabled, for: .sidebar, .tabBar)  
.customizationID("favoriteMentorList")  
.dropDestination(for: Mentor.self) { mentors in  
    favoriteMentors.append(contentsOf: mentors)  
}  
.defaultVisibility(.hidden, for: .tabBar)
```

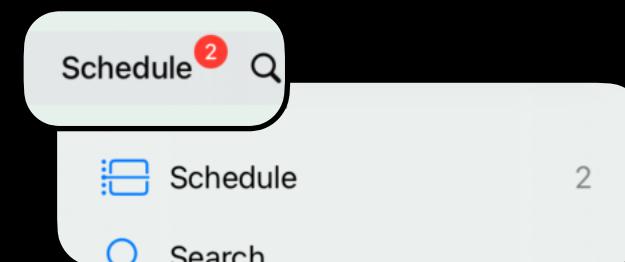
Sidebar modifications

Resetting sidebar

```
.sectionActions {  
    Button {  
        tabViewCustomization.resetSectionOrder()  
    } label: {  
        Label("Reset Section Order", systemImage: "arrow.counterclockwise")  
    }  
  
    Button {  
        tabViewCustomization.resetVisibility()  
    } label: {  
        Label("Reset Visibility", systemImage: "arrow.counterclockwise")  
    }  
}
```

Badge support

```
Tab("Schedule", systemImage: "calendar.day.timeline.left", value: .schedule) {  
    // ...  
}  
.customizationID(Tabs.schedule.customizationID)  
.badge(2)
```



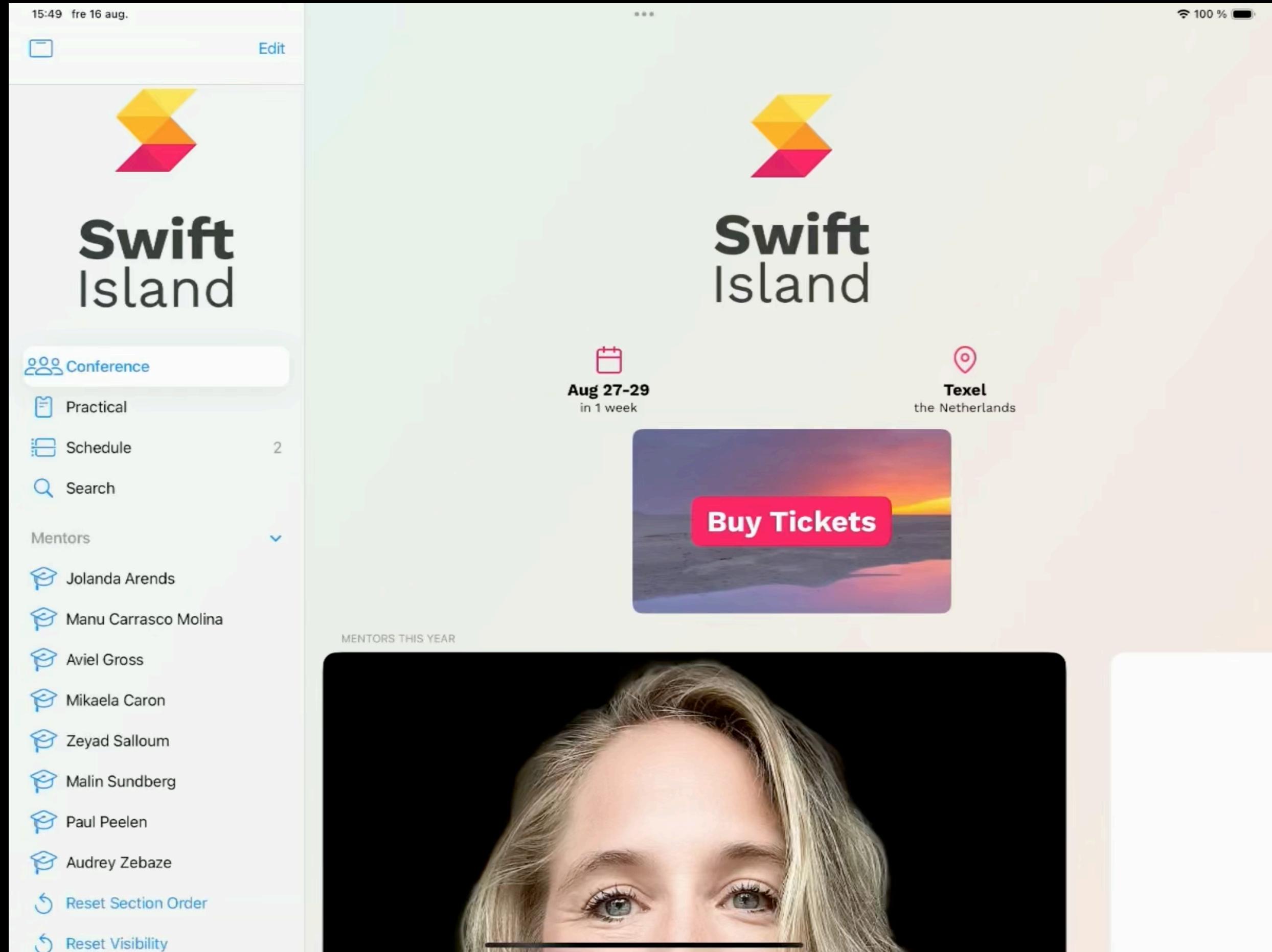
Header

```
.tabViewSidebarHeader {  
    Image("Logo")  
    VStack(spacing: 0) {  
        Text("Swift")  
            .font(.custom("WorkSans-Bold", size: 64))  
            .foregroundStyle(.logoText)  
        Text("Island")  
            .font(.custom("WorkSans-Regular", size: 60))  
            .foregroundStyle(.logoText)  
            .offset(CGSize(width: 0, height: -20))  
    }  
}
```

Footer

```
.tabViewSidebarFooter {  
    VStack(spacing: 8) {  
        Button {  
            tabViewCustomization.resetVisibility()  
            tabViewCustomization.resetSectionOrder()  
        } label: {  
            Text("Reset sidebar")  
                .font(.footnote)  
        }  
  
        Text("App version: \\" + (Bundle.main.appVersionLong) + "\\\\" + (Bundle.main.appBuild) + "\\")  
            .font(.footnote)  
            .foregroundStyle(.secondary)  
    }  
}
```

Sidebar modifications

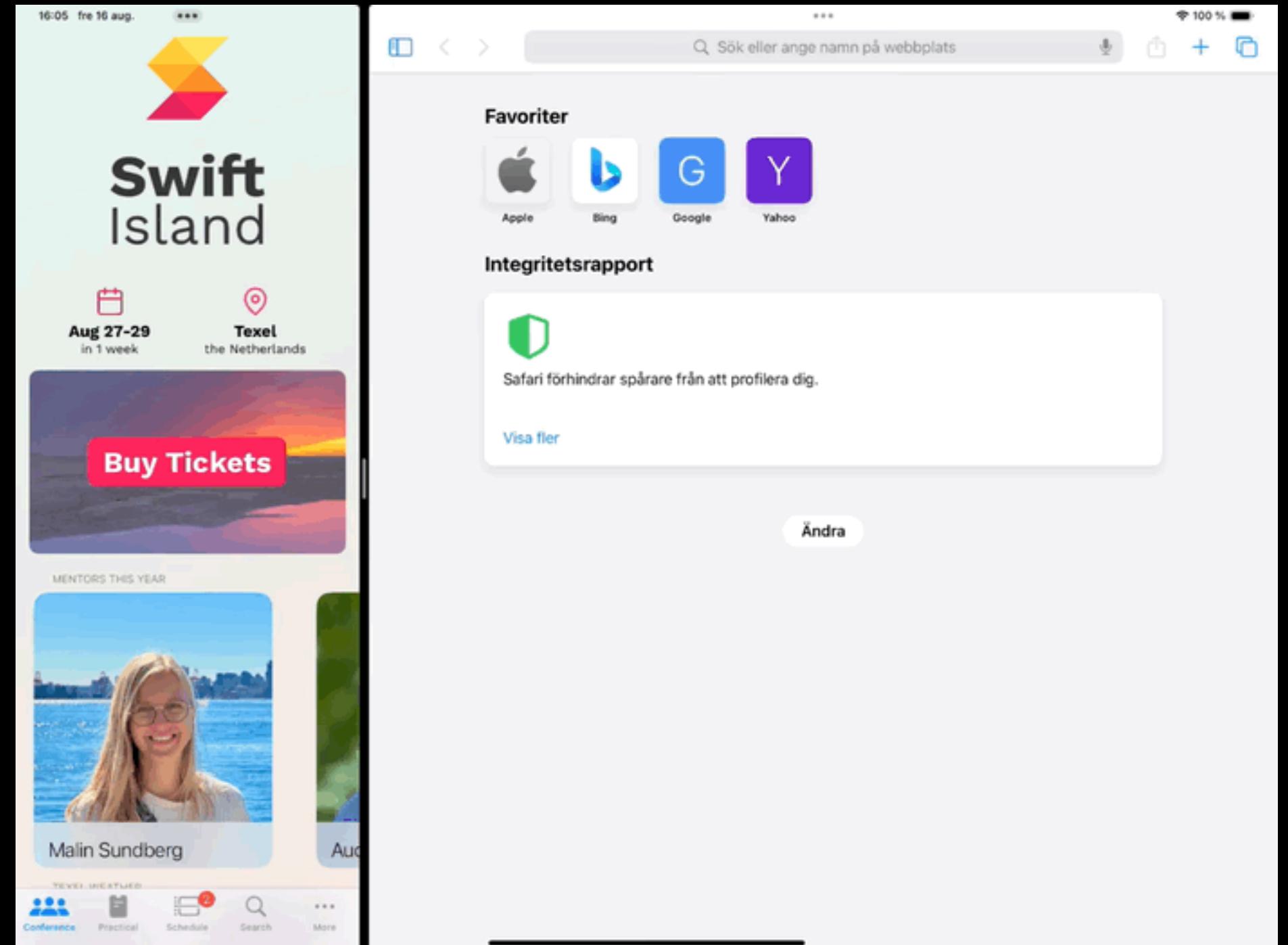


One more change...

The tabbar for compact size shows "more" item

```
@Environment(\.horizontalSizeClass) private var horizontalSizeClass  
...  
TabSection {  
    // ...  
} header: {  
    Label("Mentors", systemImage: "graduationcap")  
}  
.hidden(horizontalSizeClass == .compact)
```

Lets hide it for size class compact.



iPadOS 18

Redesigned TabBar and Sidebar

Zoom animation

iPadOS 18

Redesigned TabBar and Sidebar

Zoom animation

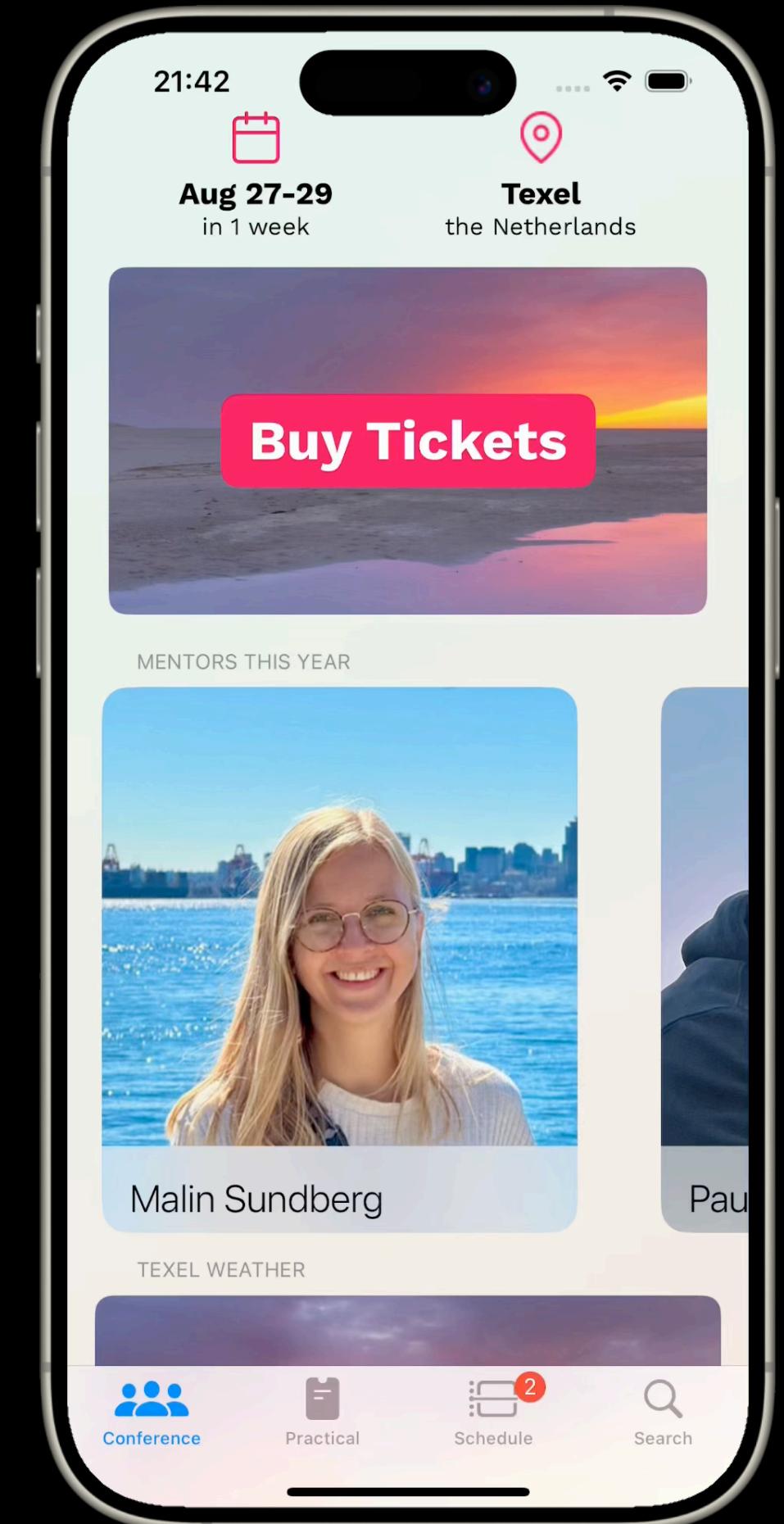
Zoom animation

Current implementation

ConferenceBoxMentors.swift

```
var namespace: Namespace.ID

ScrollView(.horizontal, showsIndicators: false) {
    LazyHStack {
        ForEach(appDataModel.mentors) { mentor in
            MentorView(namespace: namespace, mentor: mentor, isShowContent: $isShowingMentor)
                .matchedGeometryEffect(id: mentor.id, in: namespace)
                .mask {
                    RoundedRectangle(cornerRadius: 15, style: .continuous)
                }
                .padding(.horizontal, 20)
                .onTapGesture {
                    if mayShowMentorNextMentor {
                        mayShowMentorNextMentor = false
                        selectedMentor = mentor
                        withAnimation(.interactiveSpring(response: 0.55, dampingFraction: 0.8)) {
                            isShowingMentor = true
                        }
                    }
                }
                .frame(width: geo.size.width * 0.8)
                .frame(minHeight: geo.size.width * 0.80)
        }
    }
}
```



Zoom animation

Current implementation

ConferencePageView.swift

```
// The Mentor view when a mentor is selected
if let mentor = selectedMentor, isShowingMentor {
    MentorView(namespace: namespace, mentor: mentor, isShowCon
        .matchedGeometryEffect(id: mentor.id, in: namespace)
        .ignoresSafeArea()
        .onDisappear {
            mayShowMentorNextMentor = true
    }
}
```

matchedGeometryEffect(id:in:properties: anchor:isSource:)

Defines a group of views with synchronized geometry using an identifier and namespace that you provide.

```
nonisolated
func matchedGeometryEffect<ID>(
    id: ID,
    in namespace: Namespace.ID,
    properties: MatchedGeometryProperties = .frame,
    anchor: UnitPoint = .center,
    isSource: Bool = true
) -> some View where ID : Hashable
```

Parameters

id

The identifier, often derived from the identifier of the data being displayed by the view.

namespace

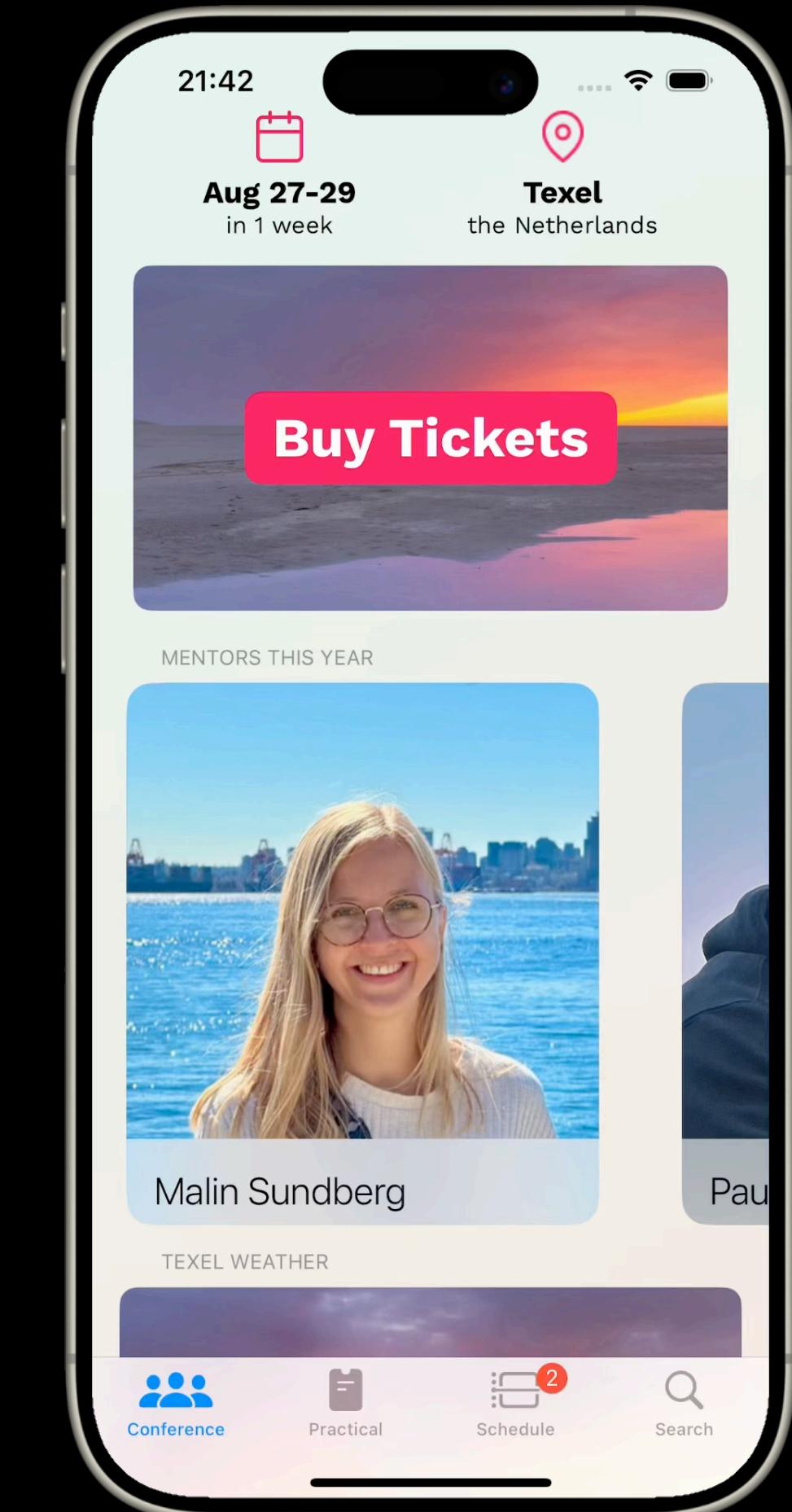
The namespace in which defines the id. New namespaces are created by adding an @Namespace variable to a [View](#) type and reading its value in the view's body method.

properties

The properties to copy from the source view.

anchor

The relative location in the view used to produce its shared position



Zoom animation

MatchedGeometryExample.swift

```
import SwiftUI
import SwiftIslandDataLogic

struct MatchedGeometryExample: View {
    @Namespace private var animationNamespace
    @State private var isExpanded = false

    var body: some View {
        VStack(alignment: .center) {
            if isExpanded {
                RoundedRectangle(cornerRadius: 25)
                    .fill(Color.blue)
                    .matchedGeometryEffect(id: "rectangle", in: animationNamespace)
                    .frame(width: 300, height: 300)
                    .onTapGesture {
                        withAnimation {
                            isExpanded.toggle()
                        }
                    }
            } else {
                RoundedRectangle(cornerRadius: 25)
                    .fill(Color.red)
                    .matchedGeometryEffect(id: "rectangle", in: animationNamespace)
                    .frame(width: 100, height: 100)
                    .onTapGesture {
                        withAnimation {
                            isExpanded.toggle()
                        }
                    }
            }
        }
    }
}

#Preview {
    MatchedGeometryExample()
}
```

Zoom animation

NavigationDestinationExample.swift

```
import SwiftUI
import SwiftIslandDataLogic

struct NavigationDestinationExample: View {
    private let mentor: Mentor = Mentor.forPreview()

    var body: some View {
        VStack {
            NavigationStack {
                List {
                    NavigationLink(value: mentor) {
                        HStack {
                            Image(mentor.imageName)
                                .resizable()
                                .scaledToFill()
                                .border(Color(.sRGB, red: 150 / 255, green: 150 / 255, blue: 150 / 255, opacity: 0.1), width: 1)
                                .aspectRatio(1.5, contentMode: .fit)
                                .frame(maxHeight: 100)
                                .cornerRadius(15)
                            Text(mentor.name)
                                .font(.title)
                        }
                    }
                }
                .navigationDestination(for: Mentor.self) { mentor in
                    MentorPageView(mentor: mentor)
                        .toolbarVisibility(.visible, for: .navigationBar)
                }
            }
        }
    }

    #Preview {
        NavigationDestinationExample()
    }
}
```

Zoom animation

```
NavigationStack {  
    ScrollView(.horizontal, showsIndicators: false) {  
        NavigationLink(value: myObject) {  
            MyView()  
        }  
    }  
    .navigationDestination(for: MyObject.self) { myObject in  
        MyOtherView(mentor: myObject)  
    }  
}
```

Zoom animation

```
NavigationStack {  
    ScrollView(.horizontal, showsIndicators: false) {  
        NavigationLink(value: myObject) {  
            MyView()  
                .matchedTransitionSource(id: "myView", in: namespace)  
        }  
    }  
    .navigationDestination(for: MyObject.self) { myObject in  
        MyOtherView(mentor: myObject)  
            .navigationTransition(.zoom(sourceID: "myView", in: namespace))  
    }  
}
```

```
#if os(macOS)
```

Zoom transition is not supported on MacOS

```
#endif
```

Zoom animation

Task

Add zoom transition support to ConferenceBoxMentorsWorkshop.swift

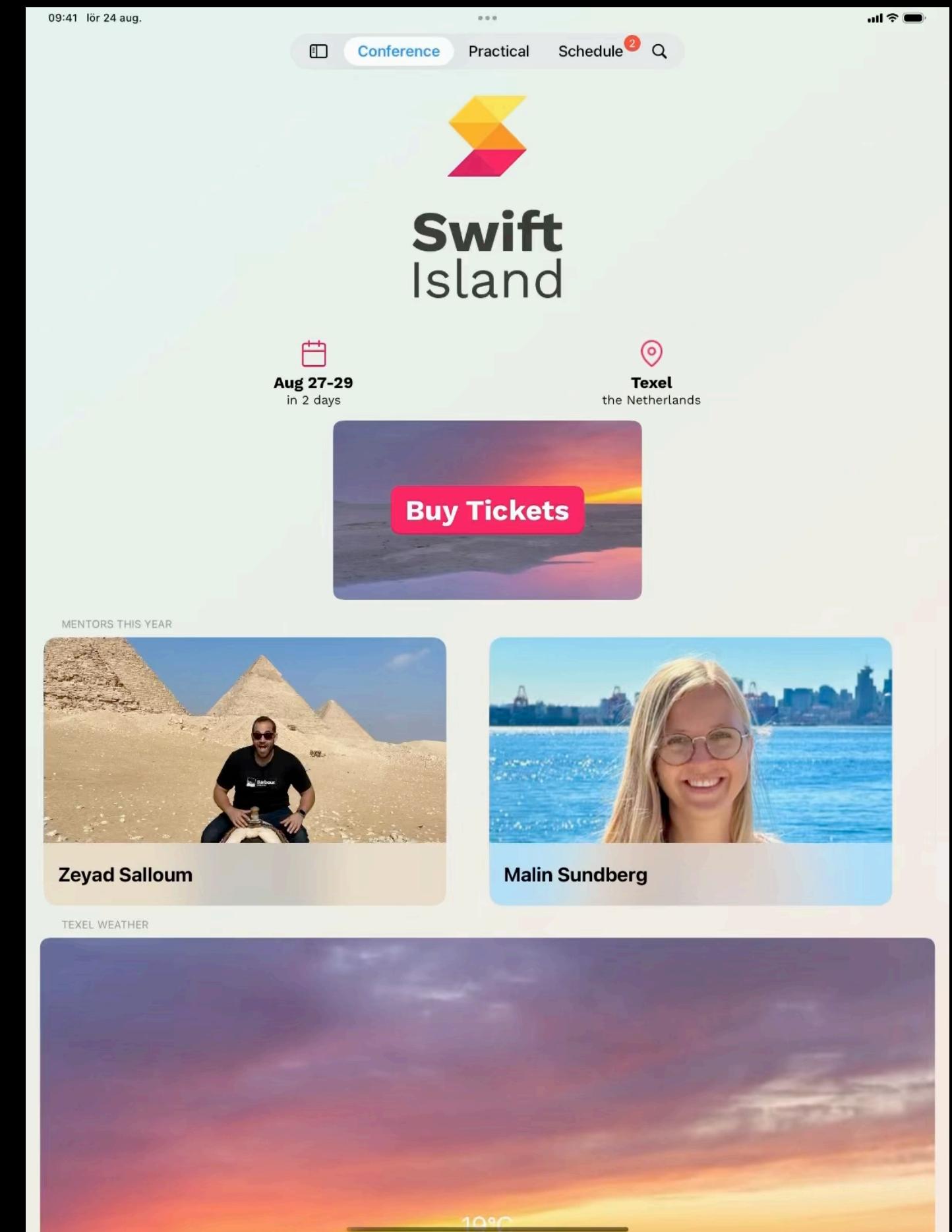
Update ConferencePageContentView to use ConferenceBoxMentorsWorkshop.swift

```
NavigationStack {  
    ScrollView(.horizontal, showsIndicators: false) {  
        NavigationLink(value: myObject) {  
            MyView()  
                .matchedTransitionSource(id: "myView", in: namespace)  
        }  
    }  
    .navigationDestination(for: MyObject.self) { myObject in  
        MyOtherView(mentor: myObject)  
            .navigationTransition(.zoom(sourceID: "myView", in: namespace))  
    }  
}
```

Zoom animation

Result

```
NavigationStack {  
    ScrollView(.horizontal, showsIndicators: false) {  
        LazyHStack {  
            ForEach(appDataModel.mentors) { mentor in  
                NavigationLink(value: mentor) {  
                    Image(mentor.imageName)  
                        .resizable()  
                        .scaledToFill()  
                        .border(Color(.sRGB, red: 150 / 255, green: 150 / 255, blue: 150 / 255,  
                            opacity: 0.1), width: 1)  
                        .cornerRadius(15)  
                        .aspectRatio(aspectRatio, contentMode: .fit)  
                        .frame(maxHeight: maxHeight)  
                        .overlay(  
                            ExcerptView(mentor: mentor)  
                        )  
                }  
                .matchedTransitionSource(id: mentor.id, in: namespace)  
                .aspectRatio(aspectRatio, contentMode: .fit)  
                .frame(maxHeight: maxHeight)  
                .mask {  
                    RoundedRectangle(cornerRadius: 15, style: .continuous)  
                }  
                .padding(.horizontal, 20)  
            }  
        }  
    }  
    .navigationDestination(for: Mentor.self) { mentor in  
        MentorPageView(mentor: mentor)  
            .navigationTransition(.zoom(sourceID: mentor.id, in: namespace))  
    }  
}
```



iPadOS 18

Redesigned TabBar and Sidebar

Zoom animation

iPadOS 18

Redesigned TabBar and Sidebar

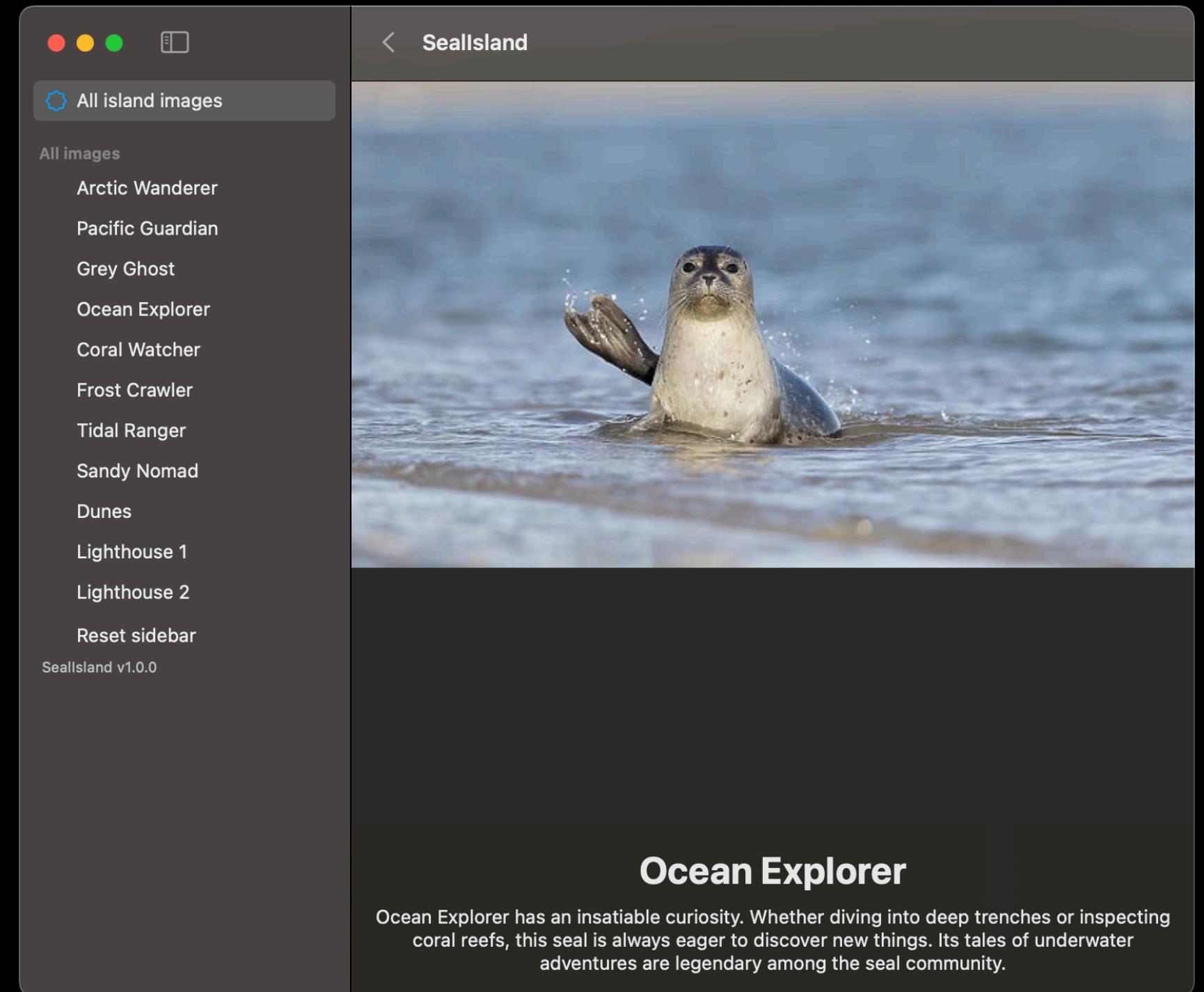
Zoom animation

Extra task

Extra task

Add the following support to the Seal Island app

- MacOS support
- Update the app and use TabView
- Add sidebar and section for all images
- When opening images from the list, use the zoom transition on iPadOS.
- Add your own flare to it, maybe using TabViewHeader?



Done.



@ppeelen



@ppeelen@mastodon.nu