RMIT University

COSC2659| iOS Development

Assignment 1

Student name: Tran Xuan Hoang Dat

SID: s3651550

App Name: VietNamLandmarksApp

Professor: Tom Huynh

Date: 25/07/2024

Contents

Introduction

Overview and Context

The VietNamLandmarksApp is designed to showcase the most famous landmarks in Vietnam, providing users with detailed information, images, and interactive maps. The app aims to offer a comprehensive and engaging experience for users interested in exploring Vietnam's rich cultural heritage and beautiful landmarks.

Goals:

- **Educational Resource:** To fix all the requirements of the app assignment and get into know how to setup a Swift app
- **Tourist Guide:** To provide tourists with a user-friendly guide to explore and appreciate Vietnam's famous sites.
- **Interactive Experience:** To offer an interactive platform where users can mark their favorite landmarks, view locations on a map, and read detailed descriptions.

Target Audience:

- **Tourists:** Both international and domestic tourists looking for reliable information on Vietnamese landmarks.
- **History and Culture Enthusiasts:** Individuals interested in learning more about Vietnam's historical and cultural sites.
- **Students and Educators:** A valuable resource for students and educators studying Vietnamese geography and history.

App Name and Logo Explanation:

- **App Name:** VietNamLandmarksApp
  - The name is straightforward and descriptive, making it clear that the app focuses on landmarks in Vietnam.
- **Logo:** I love Shiba so I chose it. Also, it makes users feel funny and relaxed when open my app.

Motivation and Inspiration:

- **Topic Choice:** The inspiration for this app comes from a deep appreciation for Vietnam's rich cultural heritage and history. The desire to make this information accessible and appealing to a wide audience is an important motivating factor

- **Motivation:** The driving force behind developing VietNamLandmarksApp was a personal interest in Vietnamese culture and history and recognizing a gap in the market for a user-friendly application, providing information highlighting Vietnam's landmarks.

By providing detailed descriptions, interactive maps, and high-quality images, VietNamLandmarksApp aims to enhance users' understanding and appreciation of Vietnam's landmarks, fostering connections between more deeply connected to the country's heritage

Implementation Details

Technical Features

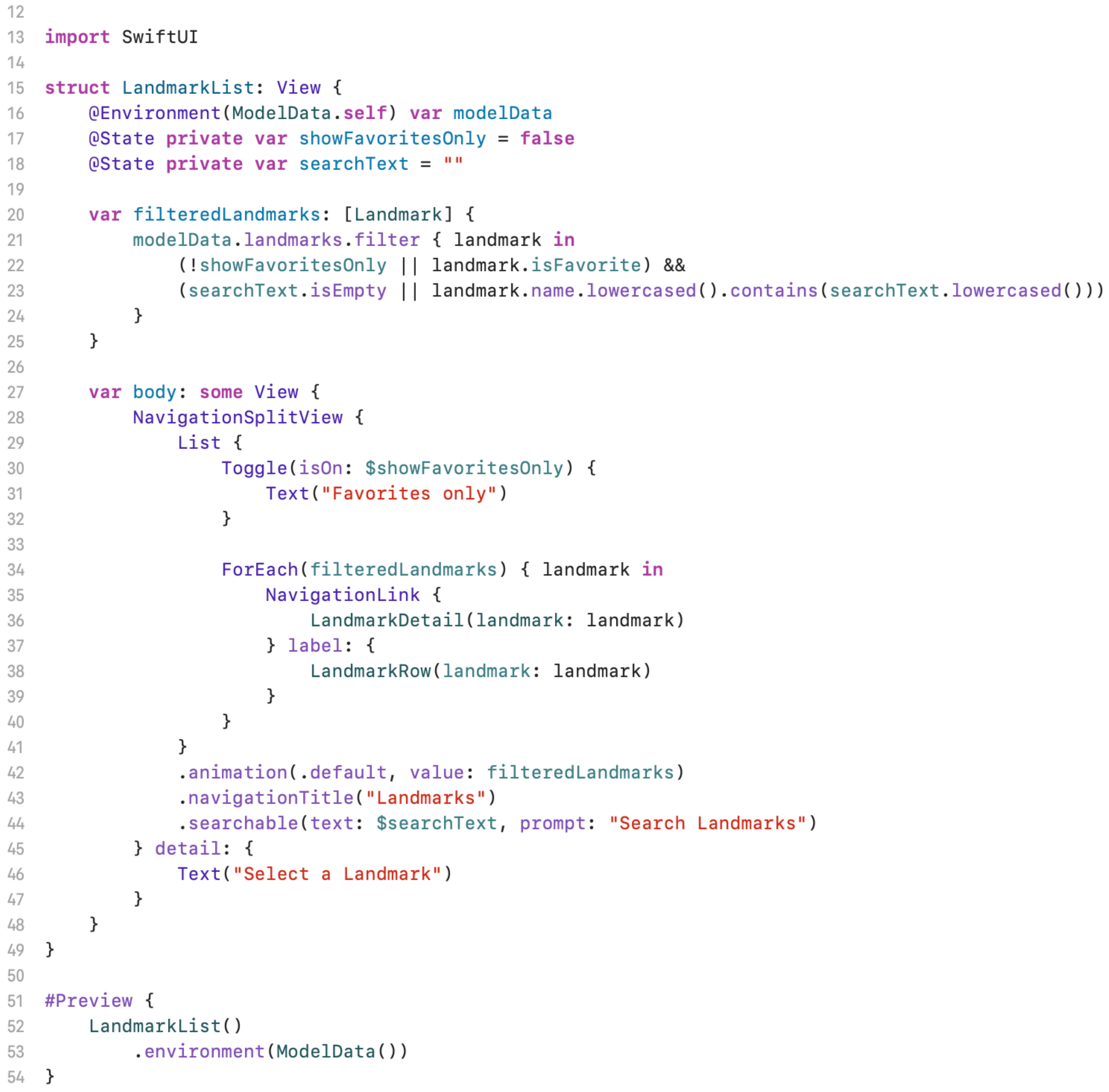
The application is built using Swift and Swift UI, on the latest iOS 15. The app not only includes several key features which follow the assignment requirements but also brings back a good user experience.

2.1 Landmark List

Feature Overview: The Landmark List displays a list of all the landmarks with a toggle to show only favorites and a search bar to filter landmarks by name.

Implementation:

- **UI Components:** SwiftUI ‘List’, ‘Toggle’, and ‘NavigationLink’.
- **State Management:** ‘@State’ properties for ‘showFavoritesOnly’ and ‘searchText’.
- **Filtering Logic:** A computed property ‘filteredLandmarks’ to filter landmarks based on the toggle and search text.



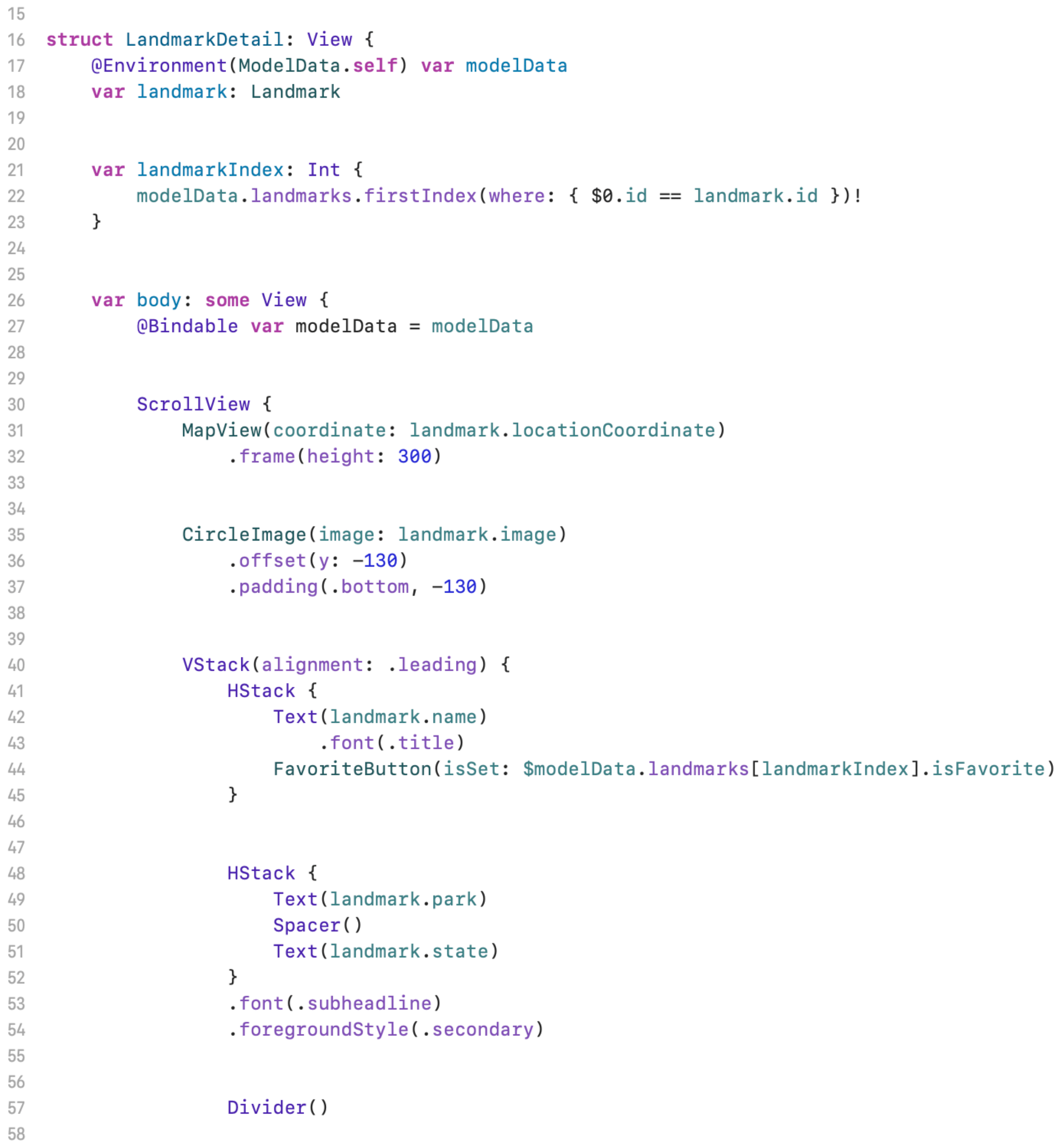
```swift
import SwiftUI

struct LandmarkList: View {
    @Environment(ModelData.self) var modelData
    @State private var showFavoritesOnly = false
    @State private var searchText = ""

    var filteredLandmarks: [Landmark] {
        modelData.landmarks.filter { landmark in
            (!showFavoritesOnly || landmark.isFavorite) &&
            (searchText.isEmpty || landmark.name.lowercased().contains(searchText.lowercased()))
        }
    }

    var body: some View {
        NavigationSplitView {
            List {
                Toggle(isOn: $showFavoritesOnly) {
                    Text("Favorites only")
                }

                ForEach(filteredLandmarks) { landmark in
                    NavigationLink {
                        LandmarkDetail(landmark: landmark)
                    } label: {
                        LandmarkRow(landmark: landmark)
                    }
                }
            }
            .animation(.default, value: filteredLandmarks)
            .navigationTitle("Landmarks")
            .searchable(text: $searchText, prompt: "Search Landmarks")
        } detail: {
            Text("Select a Landmark")
        }
    }
}

#Preview {
    LandmarkList()
        .environment(ModelData())
}
```

2.2 Landmark Detail View

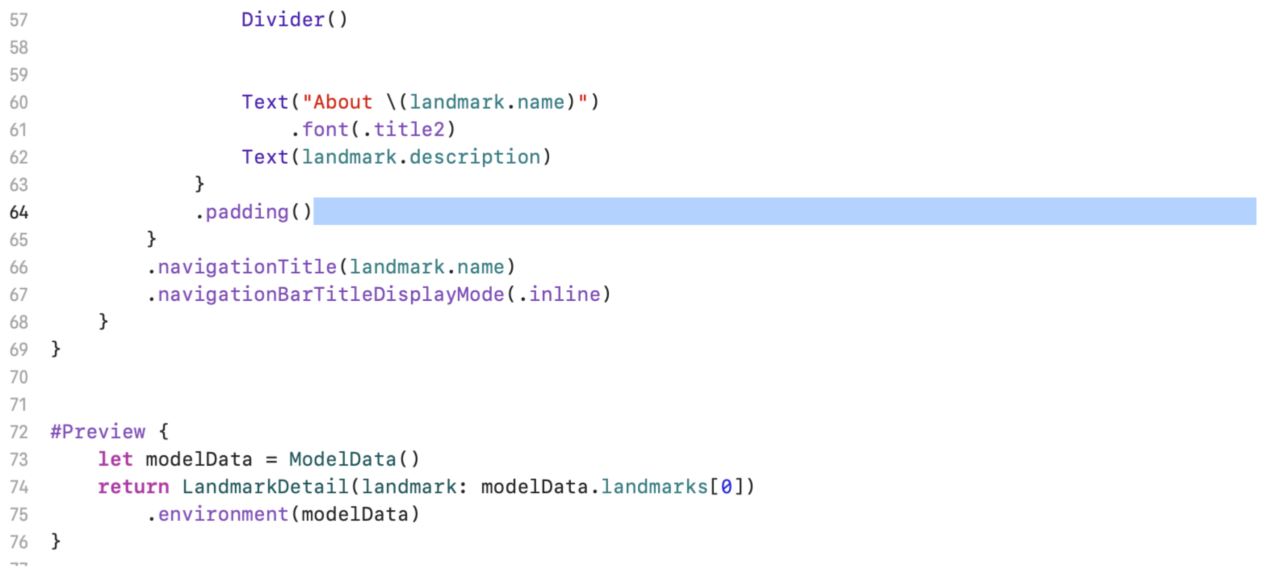
Feature Overview: The Landmark Detail View shows detailed information about a selected landmark, including its image, location on a map, and description. Users can also mark the landmark as a favorite.

Implementation:

- **UI Components:** SwiftUI ScrollView, Map, and CircleImage.
- **State Management:** @Environment and @Binding to manage data and state.
- **Favorite Toggle:** A FavoriteButton to mark the landmark as a favorite.



```swift

struct LandmarkDetail: View {
    @Environment(ModelData.self) var modelData
    var landmark: Landmark


    var landmarkIndex: Int {
        modelData.landmarks.firstIndex(where: { $0.id == landmark.id })!
    }


    var body: some View {
        @Bindable var modelData = modelData


        ScrollView {
            MapView(coordinate: landmark.locationCoordinate)
                .frame(height: 300)


            CircleImage(image: landmark.image)
                .offset(y: -130)
                .padding(.bottom, -130)


            VStack(alignment: .leading) {
                HStack {
                    Text(landmark.name)
                        .font(.title)
                    FavoriteButton(isSet: $modelData.landmarks[landmarkIndex].isFavorite)
                }


                HStack {
                    Text(landmark.park)
                    Spacer()
                    Text(landmark.state)
                }
                .font(.subheadline)
                .foregroundStyle(.secondary)


                Divider()

```



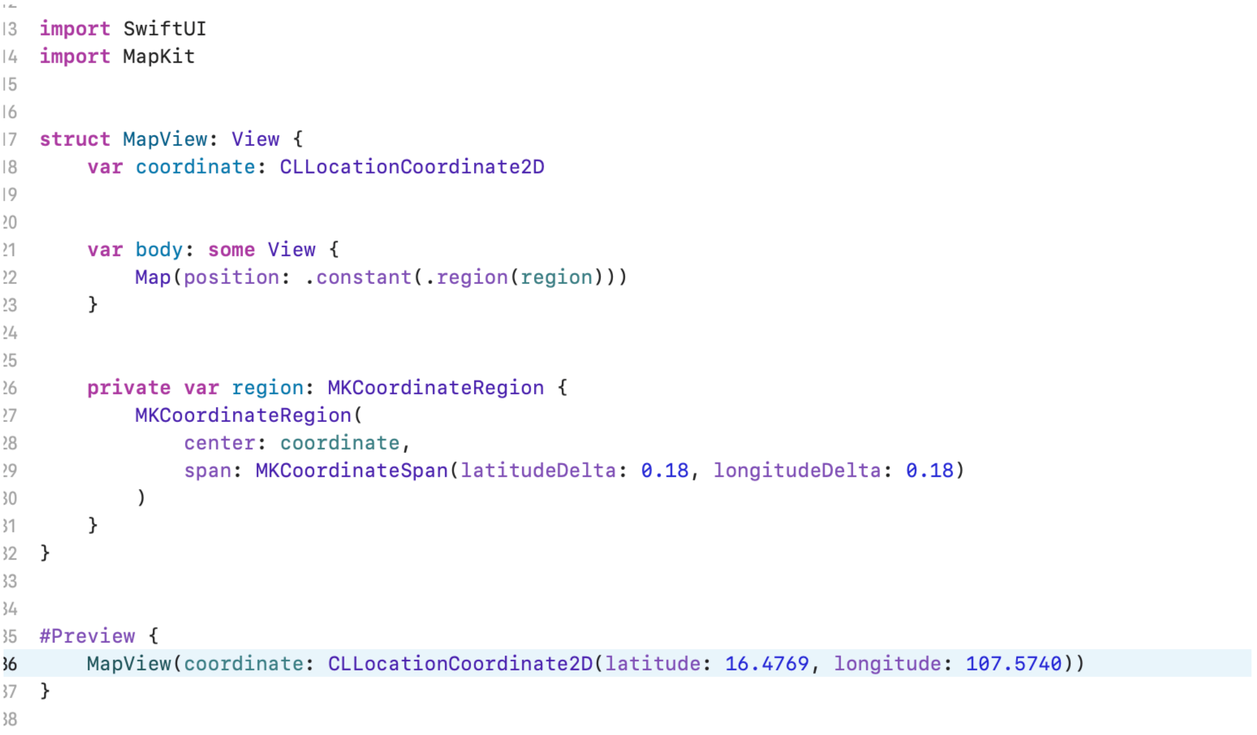
```
                Divider()


                Text("About \(landmark.name)")
                    .font(.title2)
                Text(landmark.description)
            }
            .padding()
        }
        .navigationTitle(landmark.name)
        .navigationBarTitleDisplayMode(.inline)
    }
}


#Preview {
    let modelData = ModelData()
    return LandmarkDetail(landmark: modelData.landmarks[0])
        .environment(modelData)
}
```

2.3 Map View

Feature Overview: The Map View displays the location of a landmark using Apple Maps.

Implementation:

- **UI Components:** SwiftUI Map.
- **Map Configuration:** Setting the map region based on the landmark's coordinates



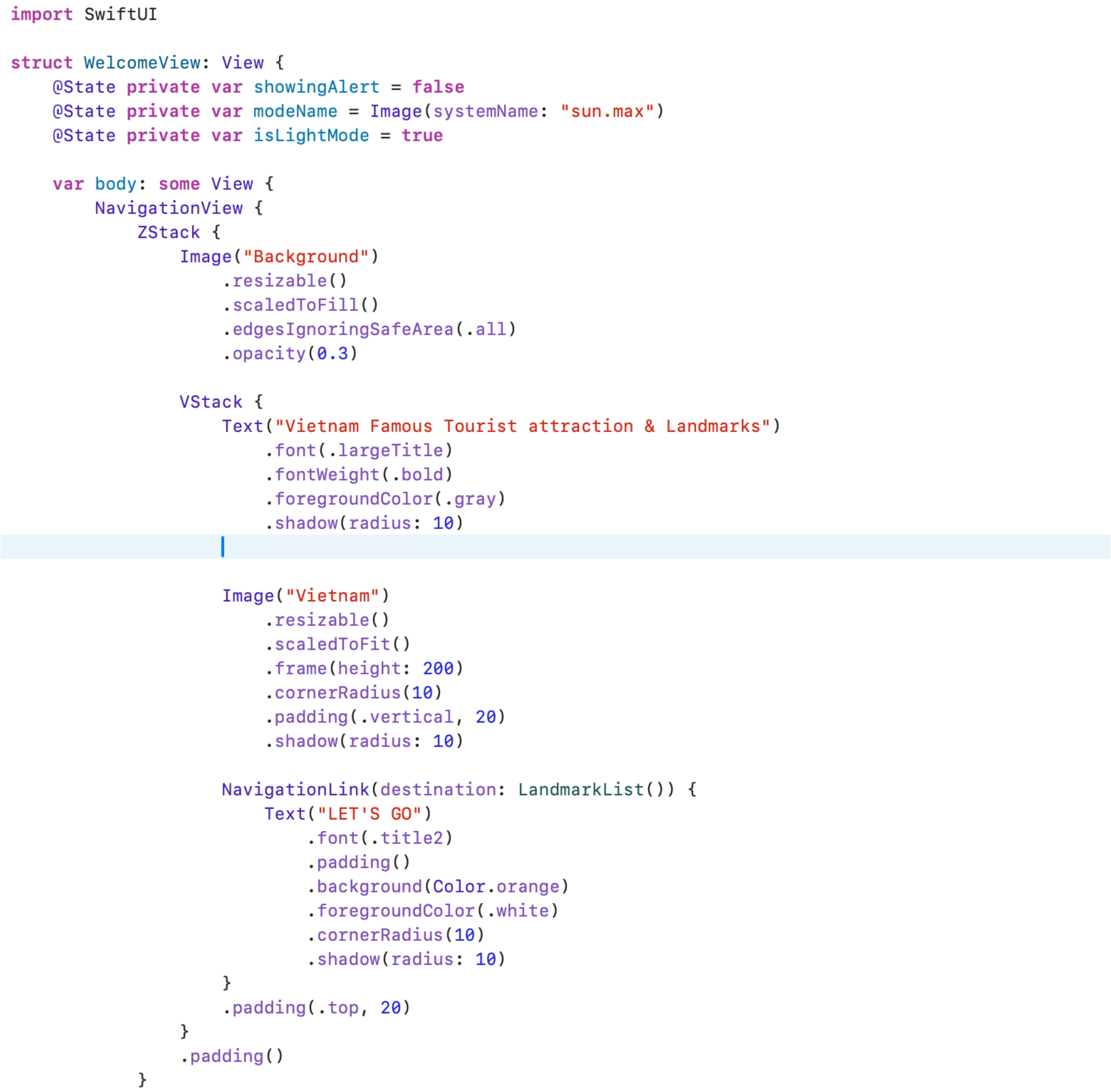
```
import SwiftUI
import MapKit


struct MapView: View {
    var coordinate: CLLocationCoordinate2D


    var body: some View {
        Map(position: .constant(.region(region)))
    }


    private var region: MKCoordinateRegion {
        MKCoordinateRegion(
            center: coordinate,
            span: MKCoordinateSpan(latitudeDelta: 0.18, longitudeDelta: 0.18)
        )
    }
}


#Preview {
    MapView(coordinate: CLLocationCoordinate2D(latitude: 16.4769, longitude: 107.5740))
}
```

2.4 Welcome View

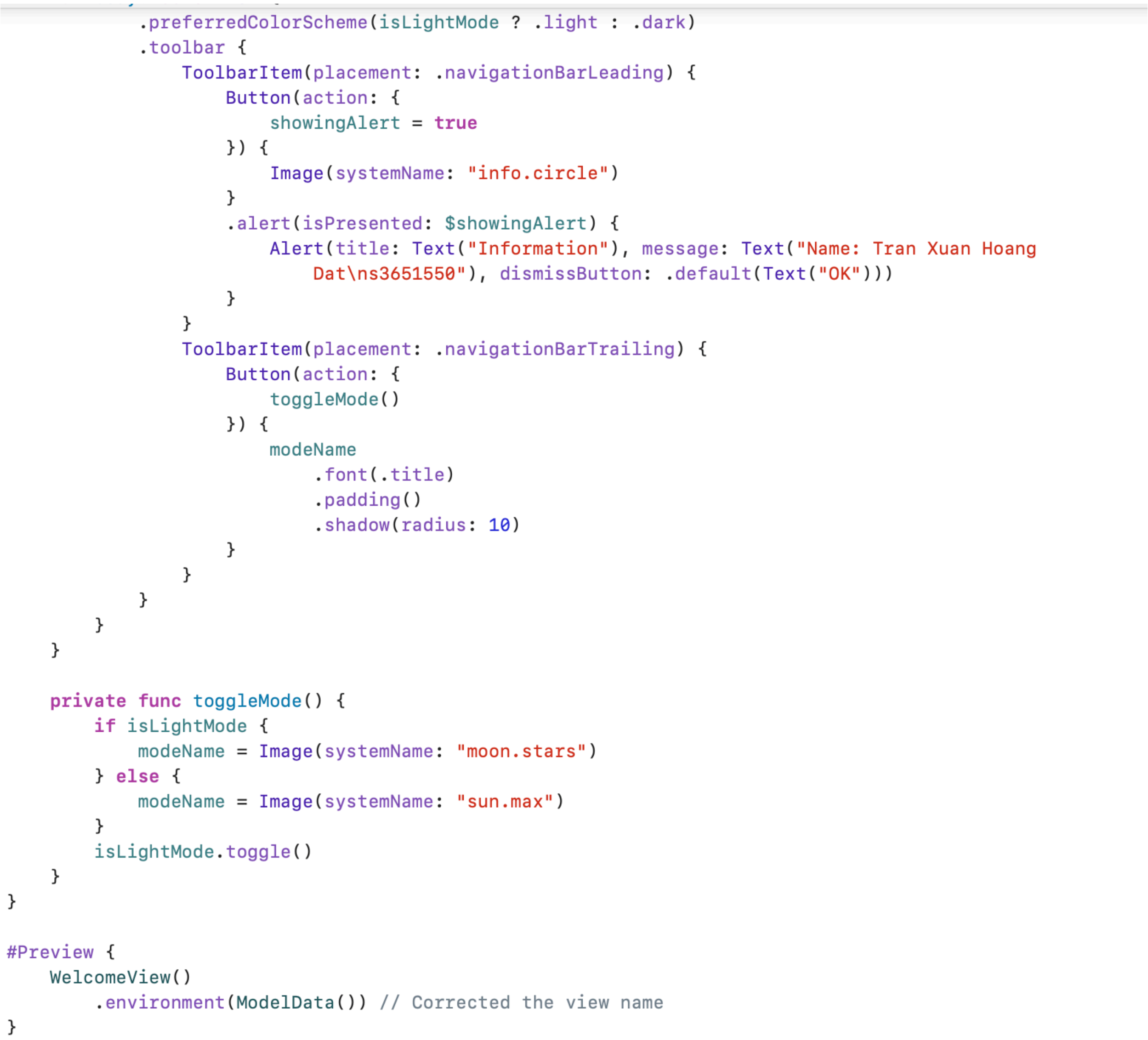
Feature Overview: The Welcome View serves as the entry point of the app, providing a brief introduction and navigation to the landmark list.

Implementation:

- **UI Components:** SwiftUI NavigationView, ZStack, and VStack.
- **Dark/Light Mode Toggle:** A button to switch between light and dark modes.



```swift
import SwiftUI

struct WelcomeView: View {
    @State private var showingAlert = false
    @State private var modeName = Image(systemName: "sun.max")
    @State private var isLightMode = true

    var body: some View {
        NavigationView {
            ZStack {
                Image("Background")
                    .resizable()
                    .scaledToFill()
                    .edgesIgnoringSafeArea(.all)
                    .opacity(0.3)

                VStack {
                    Text("Vietnam Famous Tourist attraction & Landmarks")
                        .font(.largeTitle)
                        .fontWeight(.bold)
                        .foregroundColor(.gray)
                        .shadow(radius: 10)


                    Image("Vietnam")
                        .resizable()
                        .scaledToFit()
                        .frame(height: 200)
                        .cornerRadius(10)
                        .padding(.vertical, 20)
                        .shadow(radius: 10)

                    NavigationLink(destination: LandmarkList()) {
                        Text("LET'S GO")
                            .font(.title2)
                            .padding()
                            .background(Color.orange)
                            .foregroundColor(.white)
                            .cornerRadius(10)
                            .shadow(radius: 10)
                    }
                    .padding(.top, 20)
                }
                .padding()
            }
```



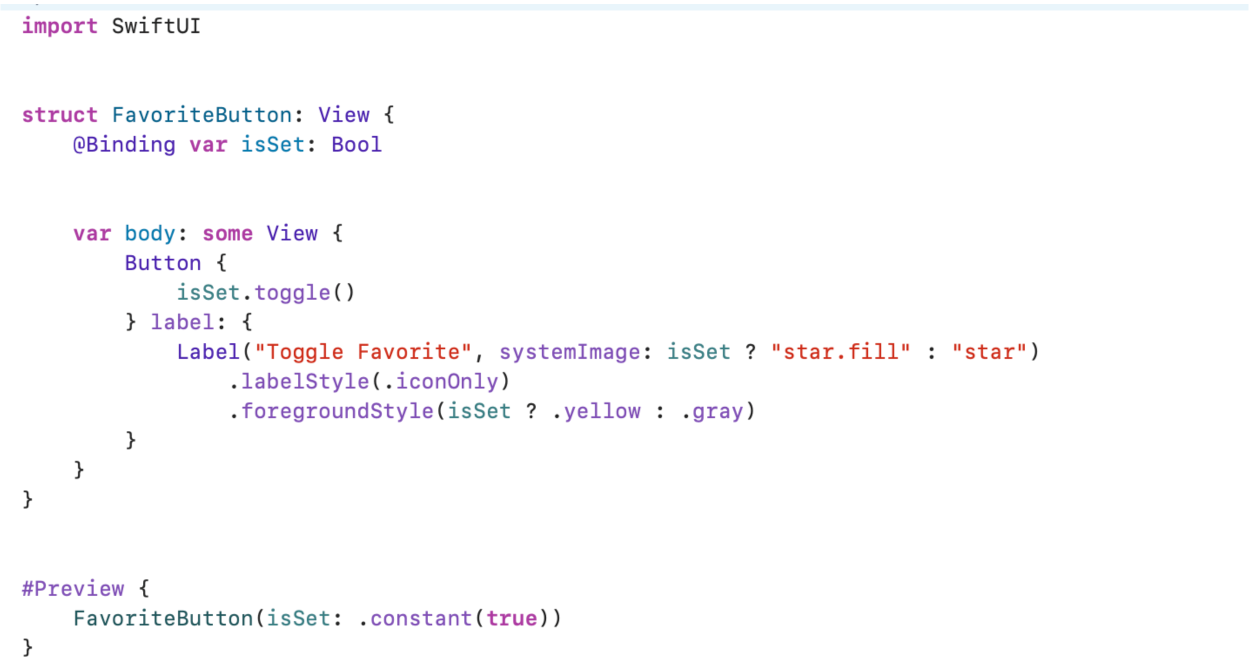
```
            .preferredColorScheme(isLightMode ? .light : .dark)
            .toolbar {
                ToolbarItem(placement: .navigationBarLeading) {
                    Button(action: {
                        showingAlert = true
                    }) {
                        Image(systemName: "info.circle")
                    }
                    .alert(isPresented: $showingAlert) {
                        Alert(title: Text("Information"), message: Text("Name: Tran Xuan Hoang
                            Dat\ns3651550"), dismissButton: .default(Text("OK")))
                    }
                }
                ToolbarItem(placement: .navigationBarTrailing) {
                    Button(action: {
                        toggleMode()
                    }) {
                        modeName
                            .font(.title)
                            .padding()
                            .shadow(radius: 10)
                    }
                }
            }
        }
    }

    private func toggleMode() {
        if isLightMode {
            modeName = Image(systemName: "moon.stars")
        } else {
            modeName = Image(systemName: "sun.max")
        }
        isLightMode.toggle()
    }
}

#Preview {
    WelcomeView()
        .environment(ModelData()) // Corrected the view name
}
```

2.5 Favorite Button

Feature Overview: The Favorite Button allows users to mark a landmark as a favorite.

Implementation:

- **UI Components:** SwiftUI Button and Label.
- **State Management:** @Binding to toggle the favorite status.



```swift
import SwiftUI


struct FavoriteButton: View {
    @Binding var isSet: Bool


    var body: some View {
        Button {
            isSet.toggle()
        } label: {
            Label("Toggle Favorite", systemImage: isSet ? "star.fill" : "star")
                .labelStyle(.iconOnly)
                .foregroundStyle(isSet ? .yellow : .gray)
        }
    }
}


#Preview {
    FavoriteButton(isSet: .constant(true))
}
```

Known Bugs/Problems

- **Duplicate Back Buttons on Landmark Detail View:**

- **Issue:** Currently, the Landmark Detail view displays two back buttons on the navigation bar, which can confuse users and clutter the interface.

- **Cause:** This issue is likely due to overlapping navigation settings or redundant navigation views within the code.

Design Elements and User Experience

Visual Appeal

Color Schemes:

- **Light Mode:** The app uses a bright and welcoming color palette for light mode, featuring white backgrounds with accent colors such as orange for buttons and gray for text.
- **Dark Mode:** In dark mode, the app switches to a dark background with white text and subdued colors to reduce eye strain and provide a pleasant user experience in low-light environments.

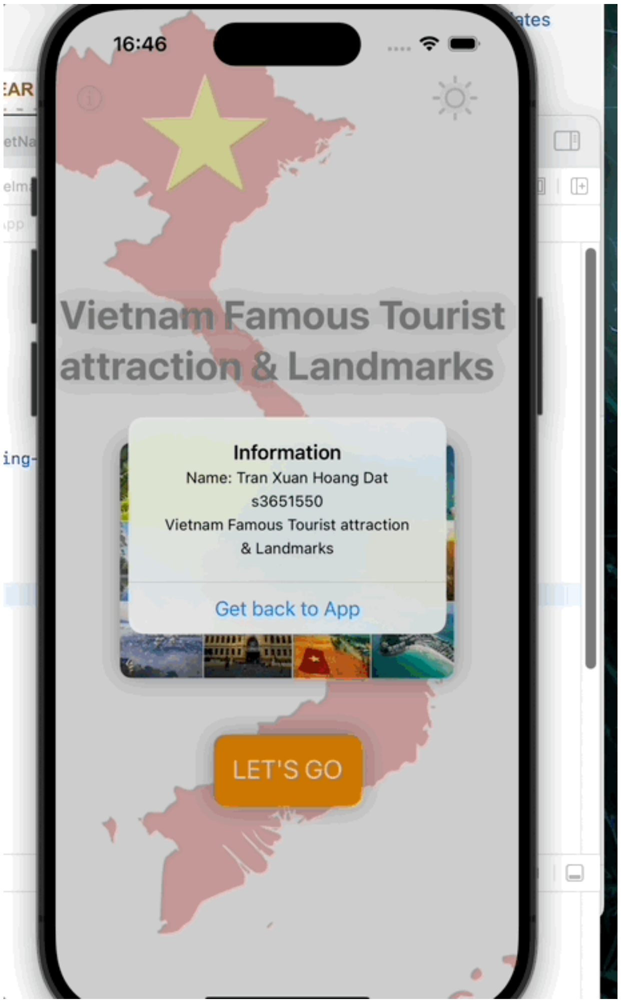
Fonts:

- The app uses the default system font to maintain a consistent and clean look across different devices and iOS versions. The text size and weight are carefully chosen to ensure readability and emphasis where necessary.

Layout:

- The layout is designed to be clean and uncluttered, with sufficient spacing between elements to ensure a comfortable reading experience. Visual hierarchy is established through the use of different font sizes and weights.

Screenshots:



Vietnam Famous Tourist attraction & Landmarks

Information

Name: Tran Xuan Hoang Dat

s3651550

Vietnam Famous Tourist attraction & Landmarks

Intuitive User Interface

Navigation:

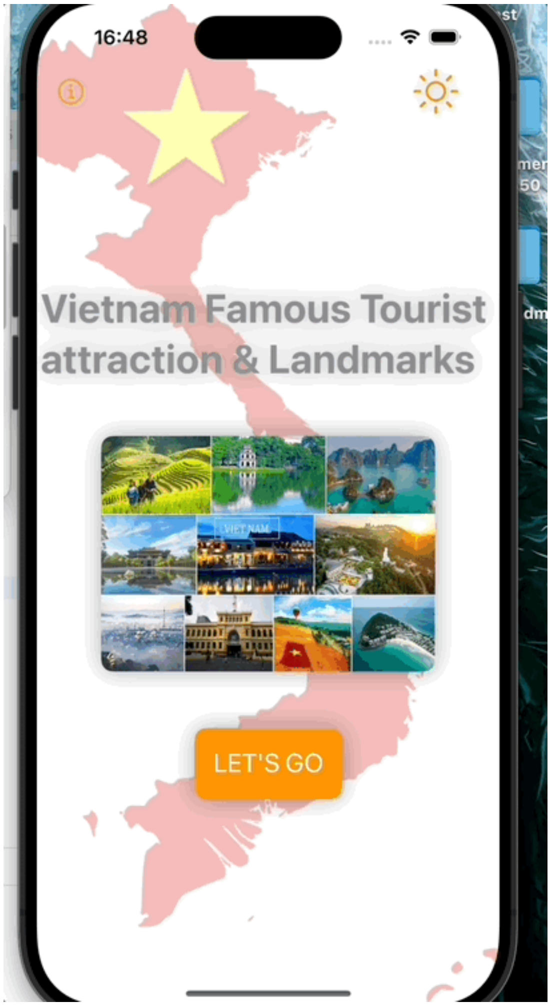
- The app uses a NavigationView and NavigationLink for easy navigation between the welcome screen, landmark list, and landmark detail views. This familiar navigation pattern ensures that users can easily find and access different parts of the app.

Buttons and Interactions:

- Buttons are designed to be easily tappable, with a clear visual indication when they are pressed. The LET'S GO button on the welcome screen is prominent and inviting, encouraging users to start exploring the landmarks.

Search and Filtering:

- The landmark list includes a search bar and a toggle switch for filtering favorites, providing users with quick and easy ways to find specific landmarks. The search functionality is responsive, updating the list in real-time as the user types.



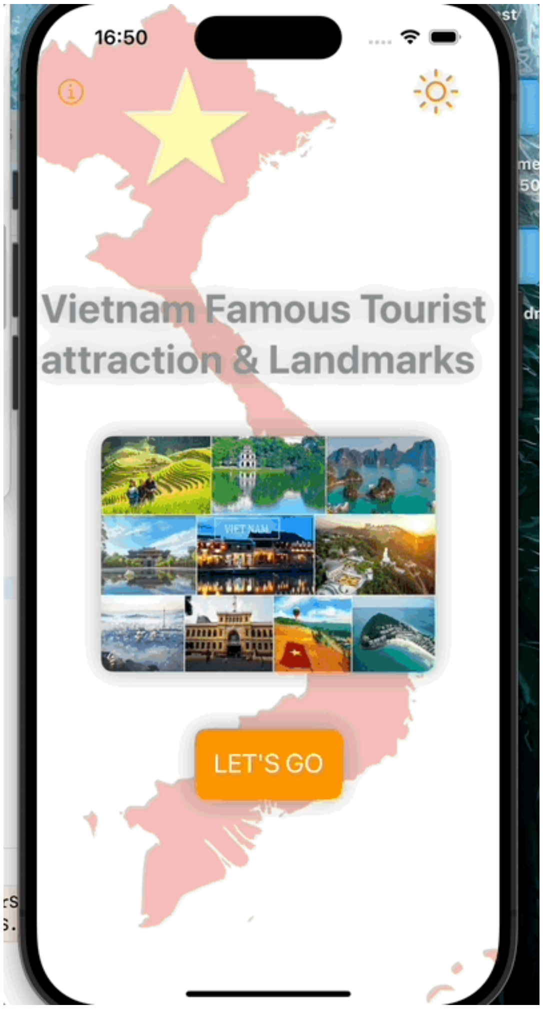
Consistency

Uniform Design Elements:

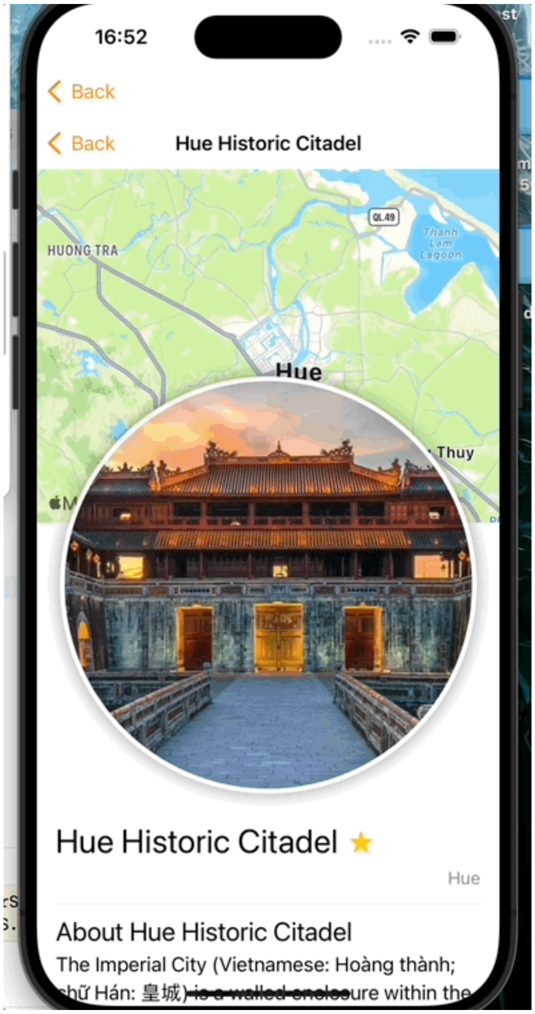
- Consistent use of fonts, colors, and UI components throughout the app ensures a cohesive look and feel. Buttons, text styles, and image displays follow a uniform design language, making the app aesthetically pleasing and easy to use.

Imagery:

- Landmark images are displayed in a consistent manner, using the same size and aspect ratio to ensure visual harmony. The use of CircleImage for profile-like displays and MapView for location visualization provides a uniform presentation of landmarks.



Vietnam Famous Tourist attraction & Landmarks



Hue Historic Citadel ★

Hue

About Hue Historic Citadel

The Imperial City (Vietnamese: Hoàng thành; chữ Hán: 皇城) ~~is a walled enclosure~~ure within the

Conclusion

Building the Vietnam Landmarks app was a rewarding experience that helped improve my skills in SwiftUI and iOS development. I gained valuable insights into state management, data processing, and creating a user interface that can adapt to both light and dark modes. As for future improvements, I plan to optimize performance, add features like user reviews and ratings, and fix known bugs like duplicate back button issue, ensure app more powerful and user friendly.

Reference: https://developer.apple.com/tutorials/swiftui/handling-user-input

iOS lecturer and tutor Week 4 - Shapes, Navigation List & Map