Getting Started Tutorial (30%)

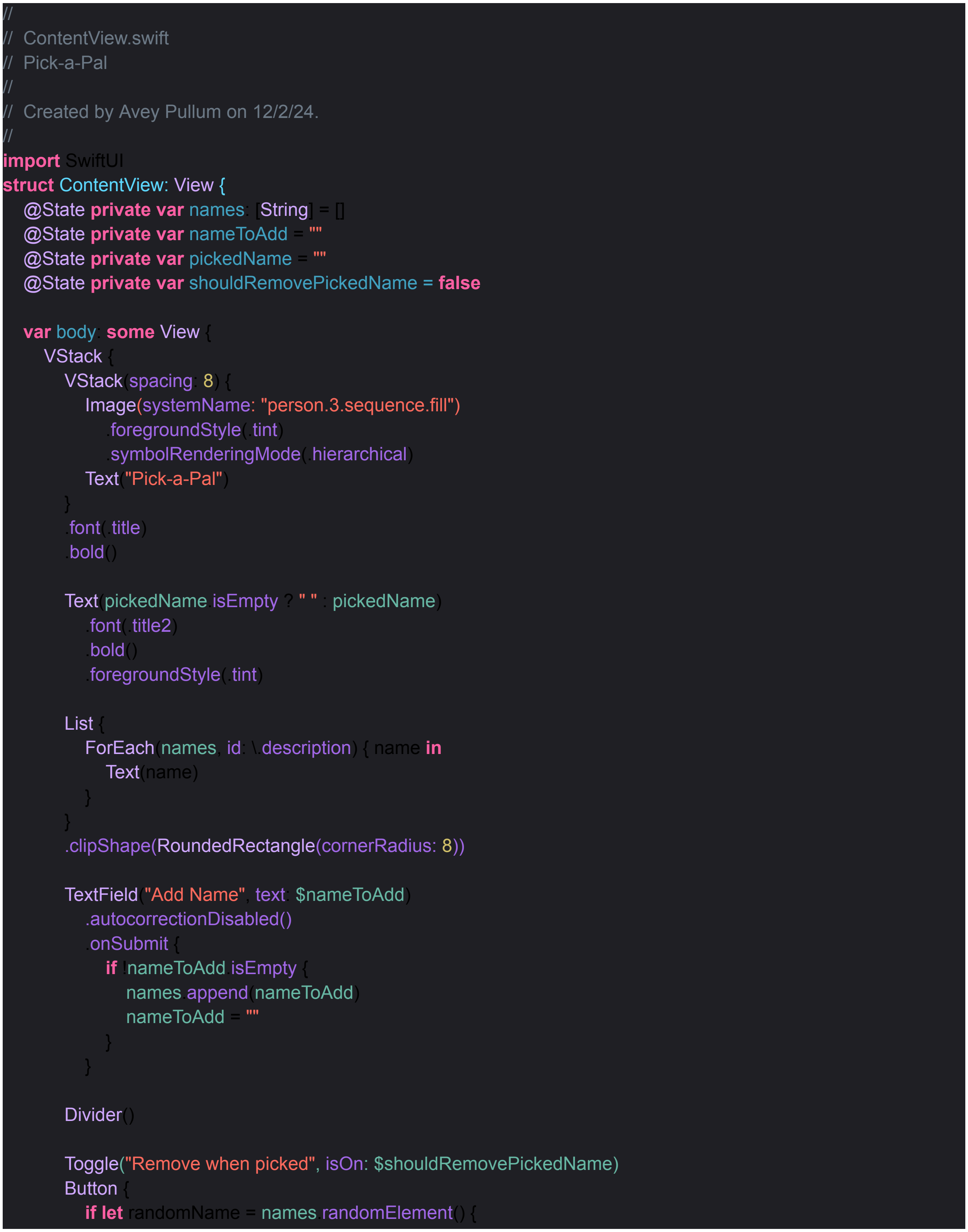
One of our main objectives for this project was to ensure that students have the chance to actually get started learning the programming languages and tools needed to get started on the project you have designed! Of course, given the large variety of programming languages and tools, you may not have the chance to learn these things in the traditional CS curriculum. Self-learning and teaching is one of the most valuable skills for software engineers, as it is something you will need to do throughout your entire future careers. When getting started in the real world, you will often have to complete a real "getting started" tutorial for the programming language or tool you want to learn. This section of the project will give you the time to complete such a tutorial for the type of project you want to create! If you have the chance to begin learning now, the chance that you make meaningful progress on these projects after this semester increases significantly. We also hope that you have fun, since you get to dive into a tool of your choice!

Now that you have identified the programming languages and tools you need to learn to complete your projects in the section above, to receive credit for this section, you will need to complete an Official Getting Started tutorial made for the language you need to learn. There are so many resources online that things can become overwhelming, so do not worry. Depending on the different project proposals you selected in the project proposal, choose the appropriate "concentration" below and complete the items listed inside of them! You will have until LDOC to complete this, so take your time, and have fun!

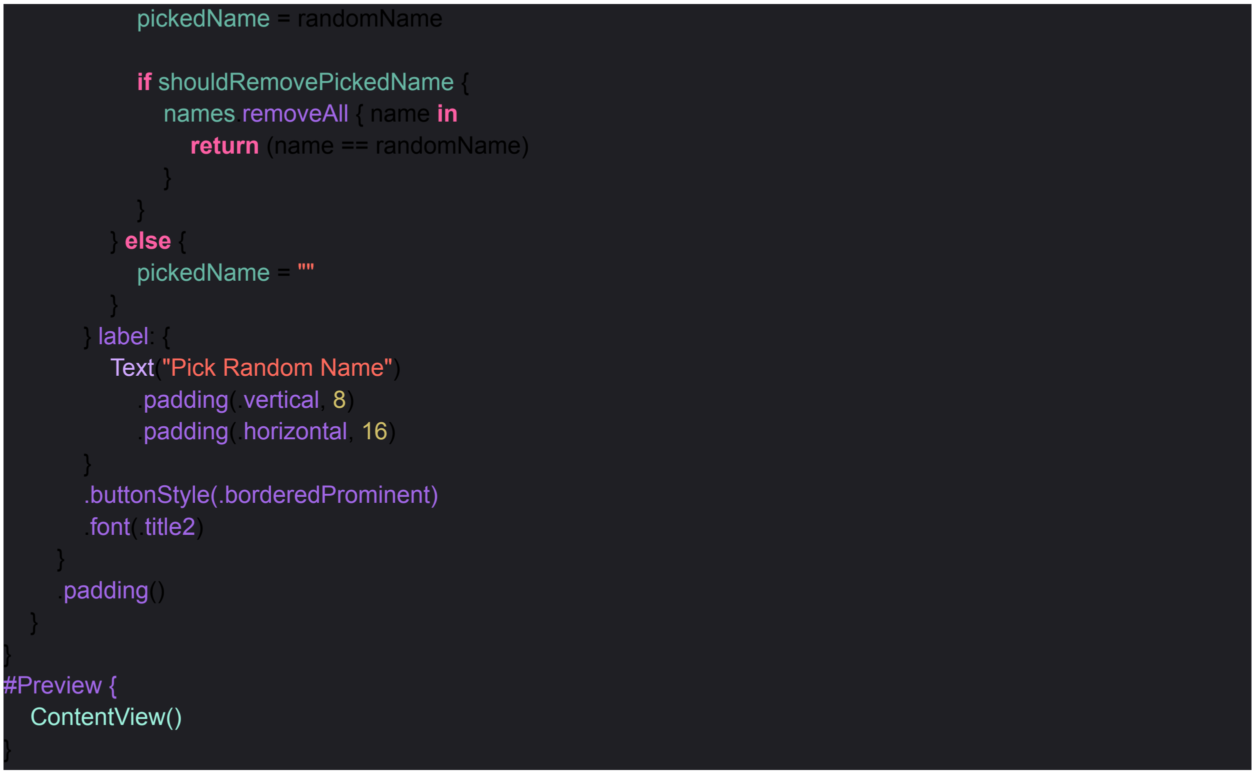
IMPORTANT: If you chose a project that is more niche / is not adequately represented in the tracks below, OR if you already have some experience in the track you chose, please reach out to either Ajay or Noah AS SOON AS POSSIBLE so we can develop a personalized track for you!

Track	Tutorials
iOS Development Track (for iOS / iPadOS / macOS apps) *Note: This track requires access to a Mac computer.*	**Official Develop in Swift Tutorial Series (here)** Created by Apple Complete the the following chapters: ● Explore Xcode ● Views, Structures, and Properties ● Layout and Style ● Buttons and State ● Lists and Text Fields **To Submit**: GitHub repository containing your complete code after following these tutorials.

Lists and Text Fields:

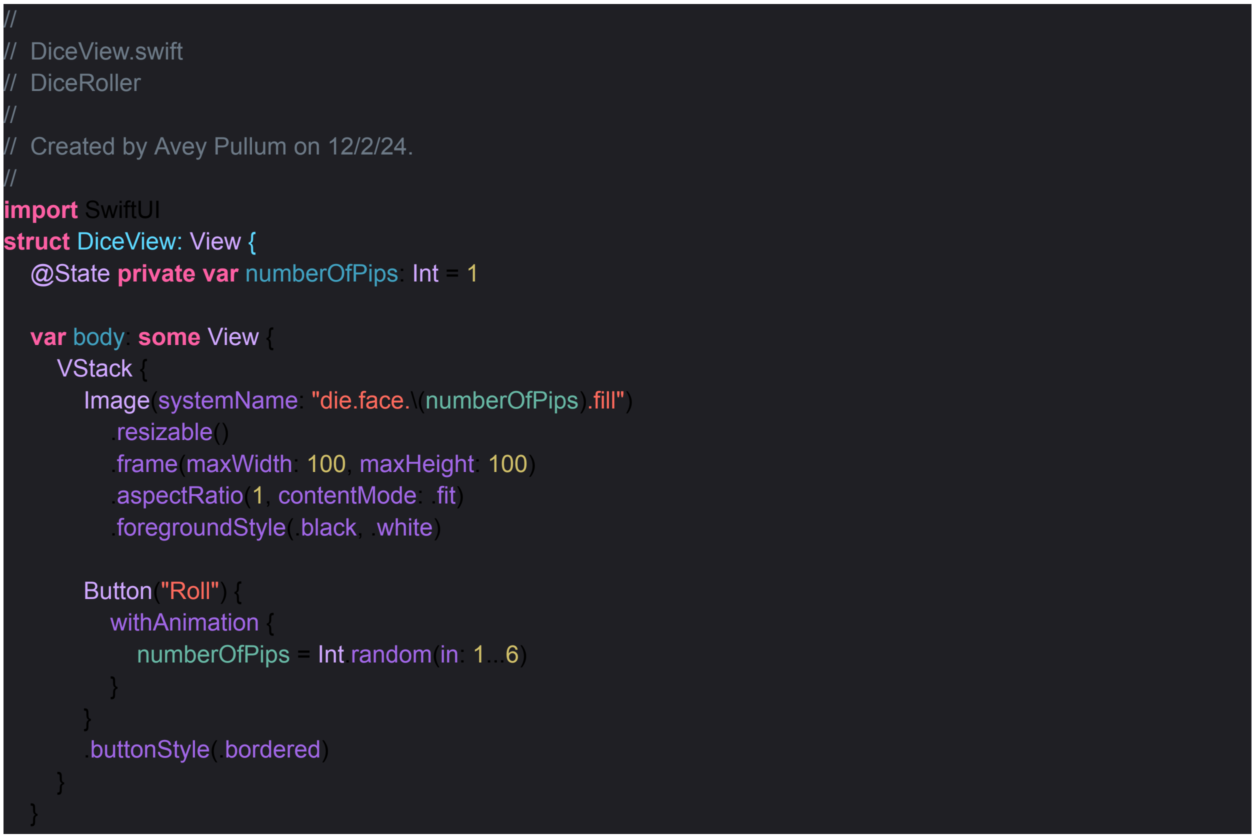


```
//
//  ContentView.swift
//  Pick-a-Pal
//
//  Created by Avey Pullum on 12/2/24.
//
import SwiftUI
struct ContentView: View {
    @State private var names: [String] = []
    @State private var nameToAdd = ""
    @State private var pickedName = ""
    @State private var shouldRemovePickedName = false

    var body: some View {
        VStack {
            VStack(spacing: 8) {
                Image(systemName: "person.3.sequence.fill")
                    .foregroundStyle(.tint)
                    .symbolRenderingMode(.hierarchical)
                Text("Pick-a-Pal")
            }
            .font(.title)
            .bold()

            Text(pickedName.isEmpty ? " " : pickedName)
                .font(.title2)
                .bold()
                .foregroundStyle(.tint)

            List {
                ForEach(names, id: \.description) { name in
                    Text(name)
                }
            }
            .clipShape(RoundedRectangle(cornerRadius: 8))

            TextField("Add Name", text: $nameToAdd)
                .autocorrectionDisabled()
                .onSubmit {
                    if !nameToAdd.isEmpty {
                        names.append(nameToAdd)
                        nameToAdd = ""
                    }
                }

            Divider()

            Toggle("Remove when picked", isOn: $shouldRemovePickedName)
            Button {
                if let randomName = names.randomElement() {
```



```swift
                    pickedName = randomName

                    if shouldRemovePickedName {
                        names.removeAll { name in
                            return (name == randomName)
                        }
                    }
                } else {
                    pickedName = ""
                }
            } label: {
                Text("Pick Random Name")
                    .padding(.vertical, 8)
                    .padding(.horizontal, 16)
            }
            .buttonStyle(.borderedProminent)
            .font(.title2)
        }
        .padding()
    }
}
#Preview {
    ContentView()
}
```

Buttons and State:



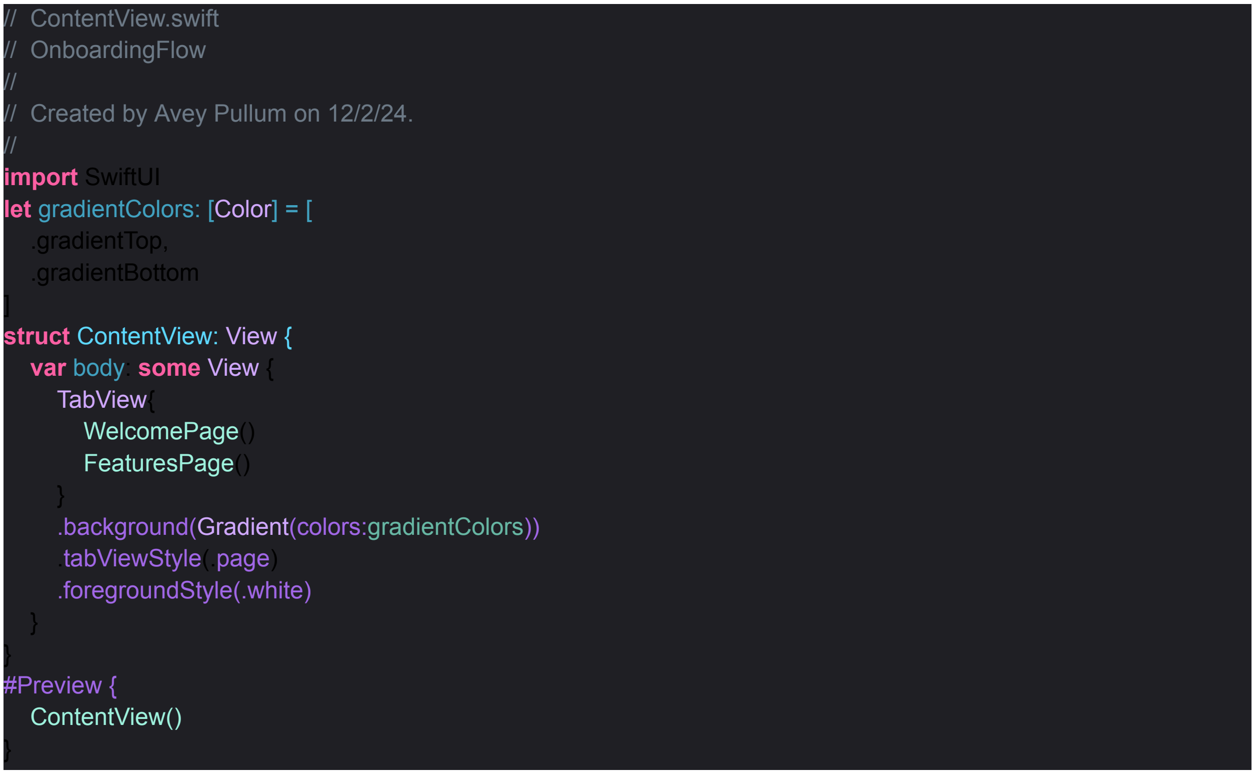
```swift
//
//  DiceView.swift
//  DiceRoller
//
//  Created by Avey Pullum on 12/2/24.
//
import SwiftUI
struct DiceView: View {
    @State private var numberOfPips: Int = 1

    var body: some View {
        VStack {
            Image(systemName: "die.face.\(numberOfPips).fill")
                .resizable()
                .frame(maxWidth: 100, maxHeight: 100)
                .aspectRatio(1, contentMode: .fit)
                .foregroundStyle(.black, .white)

            Button("Roll") {
                withAnimation {
                    numberOfPips = Int.random(in: 1...6)
                }
            }
            .buttonStyle(.bordered)
        }
    }
}
#Preview {
    DiceView()
}
```

```swift
//
//  ContentView.swift
//  DiceRoller
//
//  Created by Avey Pullum on 12/2/24.
//
import SwiftUI
struct ContentView: View {
    @State private var numberOfDice: Int = 1

    var body: some View {
        VStack {
            Text("Dice Roller")
                .font(.largeTitle.lowercaseSmallCaps())
                .foregroundStyle(.white)

            HStack {
                ForEach(1...numberOfDice, id: \.description) { _ in
                    DiceView()
                }
            }

            HStack {
                Button("Remove Dice", systemImage: "minus.circle.fill") {
                    withAnimation {
                        numberOfDice -= 1
                    }
                }
                .disabled(numberOfDice == 1)

                Button("Add Dice", systemImage: "plus.circle.fill") {
                    withAnimation {
                        numberOfDice += 1
                    }
                }
                .disabled(numberOfDice == 5)
            }
            .padding()
            .labelStyle(.iconOnly)
            .font(.title)
        }
        .padding()
        .frame(maxWidth: .infinity, maxHeight: .infinity)
        .background(.appBackground)
        .tint(.white)
    }
}
#Preview {
    ContentView()
}
```

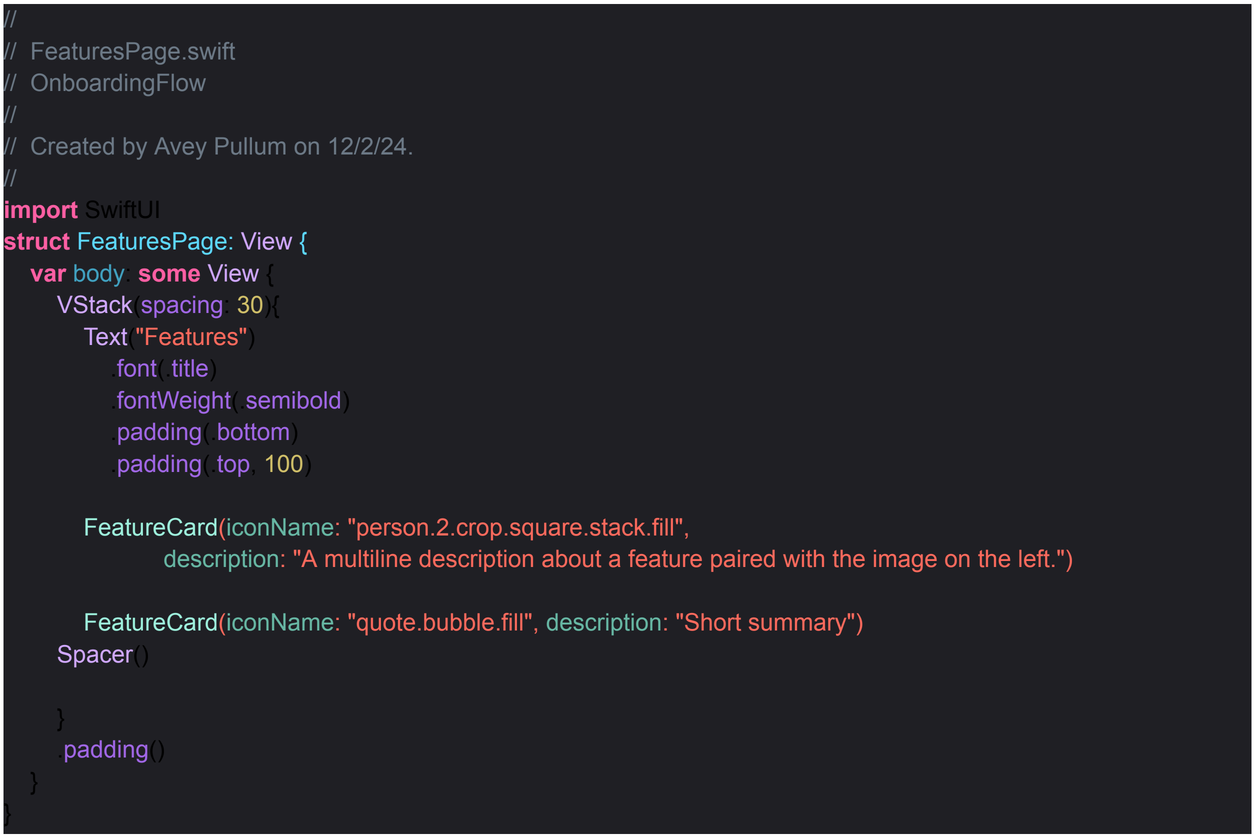
Layout and style:

```
//
//  WelcomePage.swift
//  OnboardingFlow
//
//  Created by Avey Pullum on 12/2/24.
//
import SwiftUI
struct WelcomePage: View {
    var body: some View {
        VStack {
            ZStack{
                RoundedRectangle(cornerRadius: 30)
                    .frame(width: 150, height: 150)
                    .foregroundStyle(.tint)

                Image(systemName: "figure.2.and.child.holdinghands")
                    .font(.system(size: 70))
                    .foregroundStyle(.white)
            }


            Text("Welcome to MyApp")
                .font(.title)
                .fontWeight(.semibold)
                .padding(.top)


            Text("Description Text")
                .font(.title2)

        }

        .padding()

    }
}
#Preview {
    WelcomePage()
}
```

```
//
```



```swift
//  ContentView.swift
//  OnboardingFlow
//
//  Created by Avey Pullum on 12/2/24.
//
import SwiftUI
let gradientColors: [Color] = [
    .gradientTop,
    .gradientBottom
]
struct ContentView: View {
    var body: some View {
        TabView{
            WelcomePage()
            FeaturesPage()
        }
        .background(Gradient(colors:gradientColors))
        .tabViewStyle(.page)
        .foregroundStyle(.white)
    }
}
#Preview {
    ContentView()
}
```



```swift
//
//  FeaturesPage.swift
//  OnboardingFlow
//
//  Created by Avey Pullum on 12/2/24.
//
import SwiftUI
struct FeaturesPage: View {
    var body: some View {
        VStack(spacing: 30){
            Text("Features")
                .font(.title)
                .fontWeight(.semibold)
                .padding(.bottom)
                .padding(.top, 100)

            FeatureCard(iconName: "person.2.crop.square.stack.fill",
                        description: "A multiline description about a feature paired with the image on the left.")

            FeatureCard(iconName: "quote.bubble.fill", description: "Short summary")
        Spacer()

        }
        .padding()
    }
}
```

```
#Preview {
    FeaturesPage()
        .frame(maxHeight: .infinity)
        .background(Gradient(colors: gradientColors))
        .foregroundStyle(.white)
}
```

```
//
//  FeatureCard.swift
//  OnboardingFlow
//
//  Created by Avey Pullum on 12/2/24.
//
import SwiftUI
struct FeatureCard: View {
    let iconName: String
    let description: String


    var body: some View {
        HStack {
            Image(systemName: iconName)
                .font(.largeTitle)
                .frame(width: 50)
                .padding(.trailing, 10)

            Text(description)

            Spacer()
        }
        .padding()
        .background {
            RoundedRectangle(cornerRadius: 12)
                .foregroundStyle(.tint)
                .opacity(0.25)
                .brightness(-0.4)
        }

        .foregroundStyle(.white)
    }
}
#Preview {
    FeatureCard(iconName: "person.2.crop.square.stack.fill",
            description: "A multi-line description about a feature paired with the image on the left")
}
```

Views, Structures, and Properties:

```swift
//
//  ContentView.swift
//  WeatherForcast
//
//  Created by Avey Pullum on 11/26/24.
//
import SwiftUI
struct ContentView: View {
    var body: some View {
        HStack {
            DayForecast(day: "Mon", isRainy: false, high: 70, low: 50)
            DayForecast(day: "Tue", isRainy: true, high: 60, low: 40)
        }
    }
}
#Preview {
    ContentView()
}
struct DayForecast: View {
    let day: String
    let isRainy: Bool
    let high: Int
    let low: Int

    var iconName: String {
        if isRainy {
            return "cloud.rain.fill"
        } else {
            return "sun.max.fill"

        }
    }
    var iconColor: Color {
        if isRainy {
            return Color.blue
        } else {
            return Color.yellow

        }
    }

    var body: some View {
        VStack {
            Text(day)
                .font(Font.headline)
            Image(systemName: iconName)
                .foregroundStyle(iconColor)
                .font(Font.largeTitle)
                .padding(5)
            Text("High: \(high)°F")
                .fontWeight(Font.Weight.semibold)
            Text("Low: \(low)°F")
                .fontWeight(Font.Weight.medium)
                .foregroundStyle(.secondary).foregroundStyle(Color.secondary)
        }
        .padding()
    }
}
```

Explore XCode:

```swift
//
//  ContentView.swift
//  ChatPrototype
//
//  Created by Avey Pullum on 11/22/24.
//
import SwiftUI
struct ContentView: View {
    var body: some View {
        VStack {
            Text("Hello, World!")
                .padding()
                .background(Color.yellow, in: RoundedRectangle(cornerRadius: 8))
            Text("Surely this place is not dangerous right??? Hahahah")
                .padding()
                .background(Color.teal, in: RoundedRectangle(cornerRadius: 8))
        }
        .padding()
    }
}
#Preview {
    ContentView()
}
```