

```
import SwiftUI
```

```
import WebKit
```

```
struct ContentView: View {
```

```
    @State private var selectedSection: String = "소개"
```

```
    @State private var code: String = "<html><body><h1>Hello, Swift!</h1></body></html>"
```

```
    @State private var htmlCode: String = ""
```

```
    var body: some View {
```

```
        VStack {
```

```
            MenuTabView(selectedSection: $selectedSection)
```

```
            ScrollView {
```

```
                VStack(alignment: .leading, spacing: 20) {
```

```
                    if selectedSection == "소개" {
```

```
                        ProfileView()
```

```
                        AboutMeViewPart1()
```

```
                        AboutMeViewPart2()
```

```
                    } else if selectedSection == "기술 스택" {
```

```
                        TechStackView()
```

```
                    } else if selectedSection == "프로젝트" {
```

```
                        ProjectsListView()
```

```
                    } else if selectedSection == "예제" {
```

```
                        WebCodeEditorView(code: $code, htmlCode: $htmlCode)
```

```
                }
```

```
            }
```

```
            .padding()
```

```
        }
```

```
    }
```

```
    .padding(.horizontal)
```

```
    .background(Color(UIColor.systemBackground)) // Apple 메모장처럼 배경을 깔  
    끄하게
```

```
    .navigationTitle("My Notes") // 상단에 제목 추가
```

```
    .navigationBarTitleDisplayMode(.inline) // Apple 스타일의 네비게이션 타이틀
```

```
    }
```

```
}
```

```
struct MenuTabView: View {
```

```
    @Binding var selectedSection: String
```

```
    var body: some View {
```

```
        HStack(spacing: 15) {
```

```
            ForEach(["소개", "기술 스택", "프로젝트", "예제"], id: \.self) { section in
```

```
                Button(action: {
```

```
                    selectedSection = section
```

```
                }) {
```

```
                    Text(section)
```

```

                .font(.headline)
                .foregroundColor(selectedSection == section ? .blue : .primary)
                .padding(.vertical, 8)
                .padding(.horizontal, 12)
                .background(selectedSection == section ?
Color.blue.opacity(0.1) : Color.clear)
                .cornerRadius(8)
            }
        }
    }
    .padding()
    .background(Color(UIColor.systemGray6))
}
}

```

```

struct ProfileView: View {
    var body: some View {
        VStack(alignment: .leading, spacing: 10) {
            Image(systemName: "person.circle")
                .resizable()
                .frame(width: 80, height: 80)
                .foregroundColor(.blue)
            Text("강상윤")
                .font(.largeTitle)
            Text("Frontend Developer")
                .font(.subheadline)
                .foregroundColor(.secondary)
        }
        .padding(.bottom, 20)
    }
}

```

```

struct AboutMeViewPart1: View {
    var body: some View {
        VStack(alignment: .leading, spacing: 10) {
            Text("웹 개발")
                .font(.headline)
            Text("Hello World.")
                .font(.body)
        }
    }
}

```

```

struct AboutMeViewPart2: View {
    var body: some View {
        VStack(alignment: .leading, spacing: 10) {
            Text("자기소개")

```

```

        .font(.headline)
        Text("안녕하세요! 저는 [강상윤]입니다. 현재 [배재대학교]에서 프론트엔드 개발을 전
공하고 있습니다. 최신 기술과 트렌드에 대한 열정으로, 사용자 경험을 극대화하는 직관적이고 매
력적인 인터페이스를 만드는 데 중점을 두고 있습니다.")
        Text("저는 Java, HTML, CSS, JavaScript를 사용하여 다양한 프로젝트를 진행했
으며, 이를 통해 웹 애플리케이션과 모바일 애플리케이션의 설계 및 구현 능력을 키웠습니다. 특
히, Apple 제품에 대한 깊은 이해를 바탕으로, iOS 앱 개발에 대한 관심을 가지고 있습니다.")
        Text("제 포트폴리오에는 여러 프로젝트의 사례가 포함되어 있으며, 이들은 문제 해결
능력과 창의성을 보여주는 대표적인 예입니다. 앞으로도 기술을 지속적으로 발전시키고, 더 나은
사용자 경험을 제공하기 위해 노력하겠습니다.")
        Text("감사합니다!")
        .foregroundColor(.gray)
    }
}
}

```

```

struct TechStackView: View {
    let techStacks = [
        ("HTML", "🌐", "https://developer.mozilla.org/en-US/docs/Web/HTML"),
        ("CSS", "🎨", "https://developer.mozilla.org/en-US/docs/Web/CSS"),
        ("JavaScript", "💻", "https://developer.mozilla.org/en-US/docs/Web/
JavaScript"),
        ("C#", "🖥️", "https://learn.microsoft.com/en-us/dotnet/csharp/"),
        ("Unity", "🎮", "https://unity.com/")
    ]

    var body: some View {
        VStack(alignment: .leading, spacing: 15) {
            Text("기술 스택")
                .font(.headline)

            ForEach(techStacks, id: \.0) { tech, emoji, url in
                Link(destination: URL(string: url)!) {
                    HStack {
                        Text(emoji)
                        Text(tech)
                            .font(.subheadline)
                            .foregroundColor(.primary)
                    }
                    Spacer()
                }
                .padding(.vertical, 8)
                .padding(.horizontal, 12)
                .background(Color(UIColor.systemGray6))
                .cornerRadius(8)
            }
        }
    }
}

```

```
}  
}
```

```
struct ProjectsListView: View {  
    var body: some View {  
        VStack(alignment: .leading, spacing: 10) {  
            Text("프로젝트")  
                .font(.headline)  
  
            ForEach(1...4, id: \.self) { index in  
                Link(destination: URL(string: "https://www.notion.so/  
11f25fd1ee5580eab827d100adf1bb54?  
v=12025fd1ee55804082be000c6d00e6a7&pvs=4")!) {  
                    HStack {  
                        Image(systemName: "doc.text.image")  
                            .resizable()  
                            .aspectRatio(contentMode: .fit)  
                            .frame(width: 60, height: 60)  
                        VStack(alignment: .leading) {  
                            Text("프로젝트 \(index)")  
                                .font(.subheadline)  
                            Text("설명 내용")  
                                .font(.caption)  
                                .foregroundColor(.secondary)  
                        }  
                        Spacer()  
                    }  
                    .padding(.vertical, 8)  
                    .background(Color(UIColor.systemGray6))  
                    .cornerRadius(8)  
                }  
            }  
        }  
    }  
}
```

```
struct WebCodeEditorView: View {  
    @Binding var code: String  
    @Binding var htmlCode: String  
  
    var body: some View {  
        VStack(alignment: .leading, spacing: 15) {  
            Text("예제")  
                .font(.headline)  
  
            TextEditor(text: $code)  
                .frame(height: 150)  
        }  
    }  
}
```

```

        .border(Color(UIColor.systemGray3), width: 1)
        .padding(.bottom, 10)

        Button(action: {
            htmlCode = code
        }) {
            Text("코드 실행")
                .font(.headline)
                .padding()
                .background(Color.blue)
                .foregroundColor(.white)
                .cornerRadius(10)
        }

        Text("HTML 코드 실행 결과")
            .font(.subheadline)
            .padding(.top, 10)

        WebView(html: $htmlCode)
            .frame(height: 250)
            .border(Color(UIColor.systemGray3), width: 1)
    }
}

struct WebView: UIViewRepresentable {
    @Binding var html: String
    let webView = WKWebView()

    func makeUIView(context: Context) -> WKWebView {
        return webView
    }

    func updateUIView(_ uiView: WKWebView, context: Context) {
        uiView.loadHTMLString(html, baseURL: nil)
    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}

```