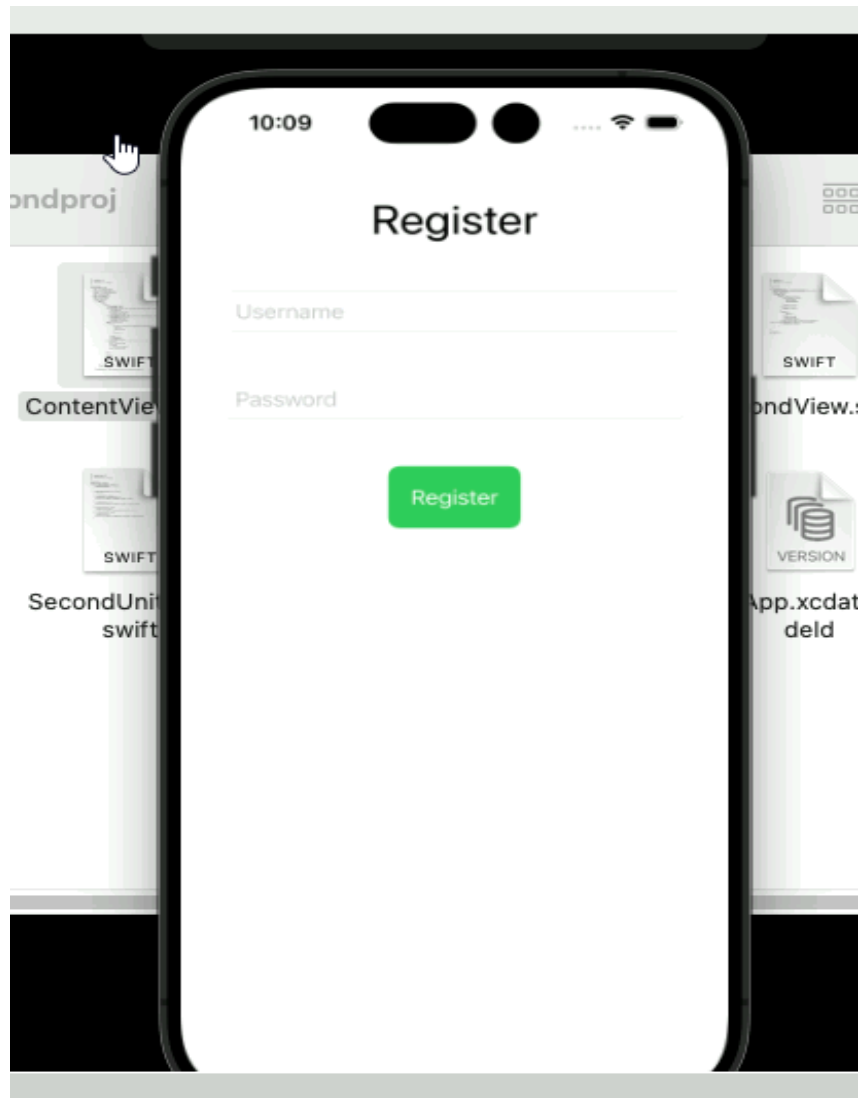


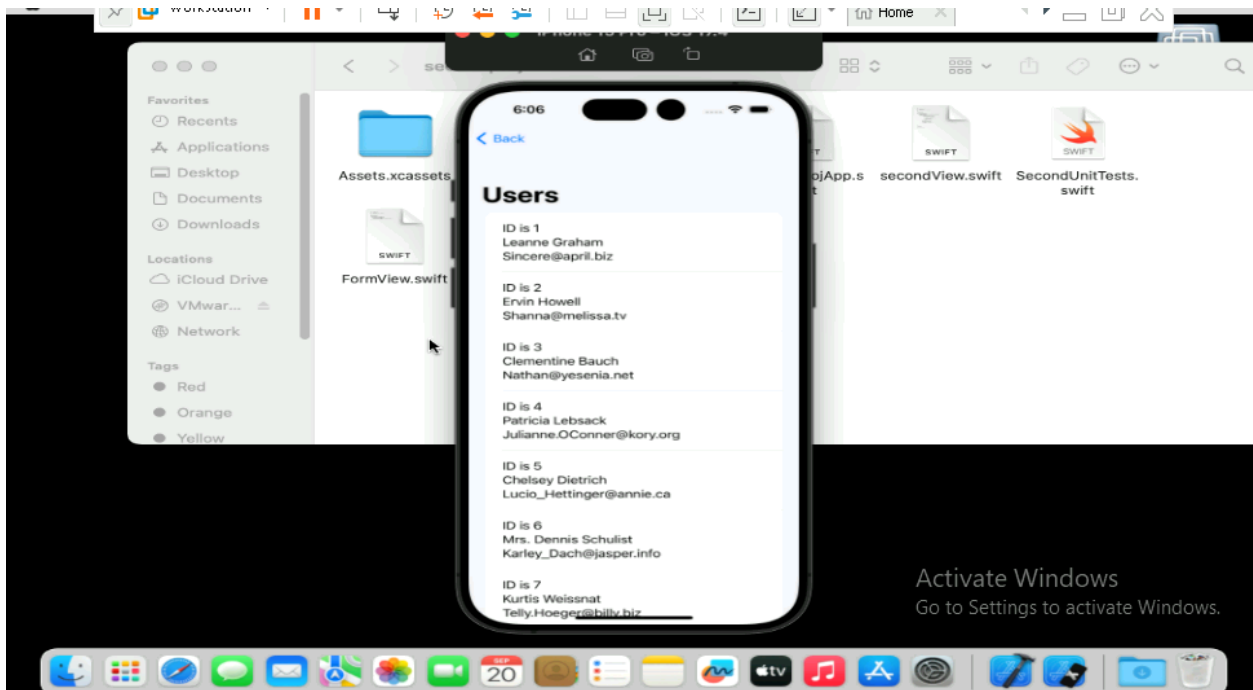
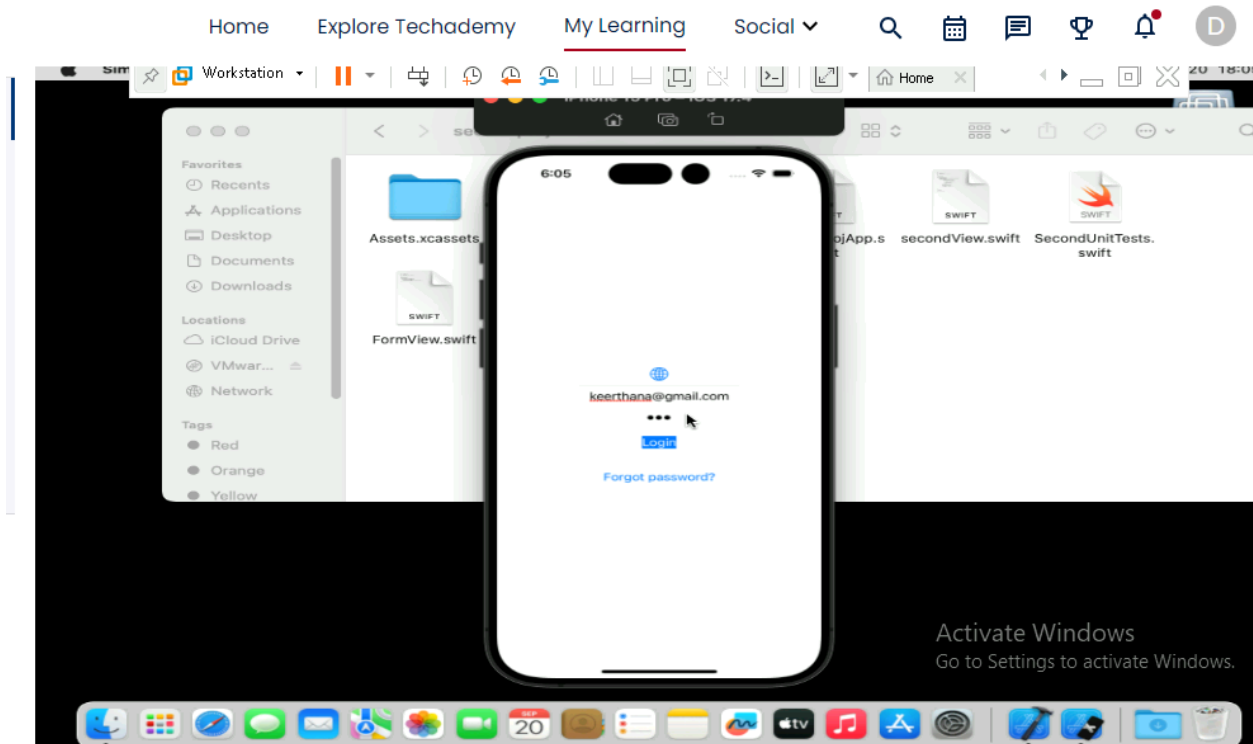
CASE STUDY

OUTPUTS

Registration form



Login form



secondprojApp

```
8 import SwiftUI
9 struct Content2View:View {
10     @State private var isRegistered:Bool=false
11     @State private var storedUsername:String=""
12     @State private var storedPassword:String=""
13     var body: some View {
14         NavigationView{
15             if isRegistered{
16                 ContentView(storedUsername:$storedUsername,storedPassword:$storedPassword)
17             }
18             else{
19                 RegistrationView(isRegistered: $isRegistered,storedUsername: $storedUsername,storedPassword: $storedPassword)
20             }
21         }
22     }
23 }
24 @main
25 struct secondprojApp: App {
26     var body: some Scene {
27         WindowGroup {
28             Content2View()
29         }
30     }
31 }
32
```

CONTENTVIEW CLASS

secondproj » secondproj » ContentView » body

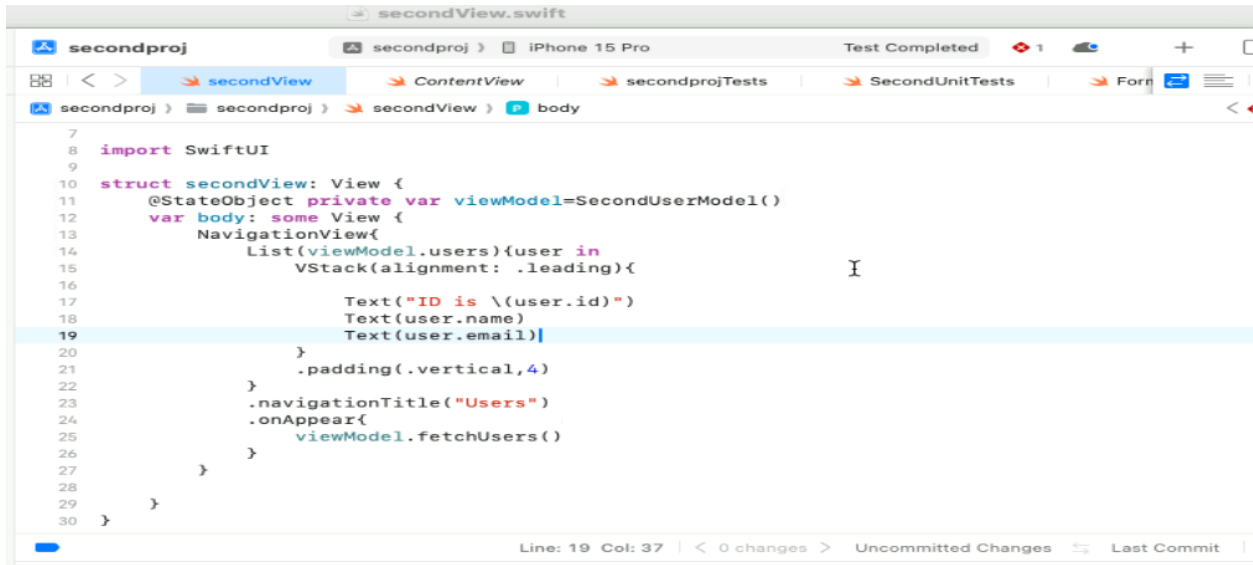
```
8 import SwiftUI
9 struct ContentView: View{
10     @State var username:String=""
11     @State var password:String=""
12     @Binding var storedUsername:String
13     @Binding var storedPassword:String
14     @State var myalert=false
15     @State var showRes=false
16     @State var myCV=false
17     @State var goprev=false
18
19     var body: some View {
20         NavigationView{
21             HStack{
22                 VStack{
23                     Image(systemName: "globe")
24                         .imageScale(.large)
25                         .foregroundColor(.tint)
26                     TextField("Enter username", text: $username).multilineTextAlignment(.center)
27                         .frame(width:200)
28                         .foregroundColor(Color.black)
29                     SecureField("Enter password", text: $password).multilineTextAlignment(.center)
30                         .lineSpacing(30)
31                         .foregroundColor(Color.black)
32                         .disableAutocorrection(true)
33                     Spacer().frame(height:20)
34                     Button(
35                         action:{
36                             if(username=="storedUsername" && password=="storedPassword"){
37                                 myalert=true
38                                 self.goprev=true
39                             }
40                         }
41                     )
42                 }
43             }
44         }
45     }
46 }
```

```

45         else{
46             showRes=true
47         }
48         },label:{
49             Text("Login")
50         }).background(.blue)
51         .foregroundColor(.white)
52         NavigationLink(destination: secondView(), isActive: $goprev){
53             EmptyView()
54
55             Spacer().frame(height:20)
56             if(showRes){
57                 Text("Incorrect details")
58             }
59             Spacer().frame(height:20)
60         }
61         Button{
62             }label:{
63                 Text("Forgot password?")
64             }
65         }
66         .padding()
67         .background().frame(maxWidth: .infinity)
68     }
69 }
70 .frame(maxHeight: .infinity)
71 .fixedSize(horizontal: false, vertical: true)
72 }
73 // KeyChainHelper().saveCredentials(username:username,password:password)
74 }
75 }
76

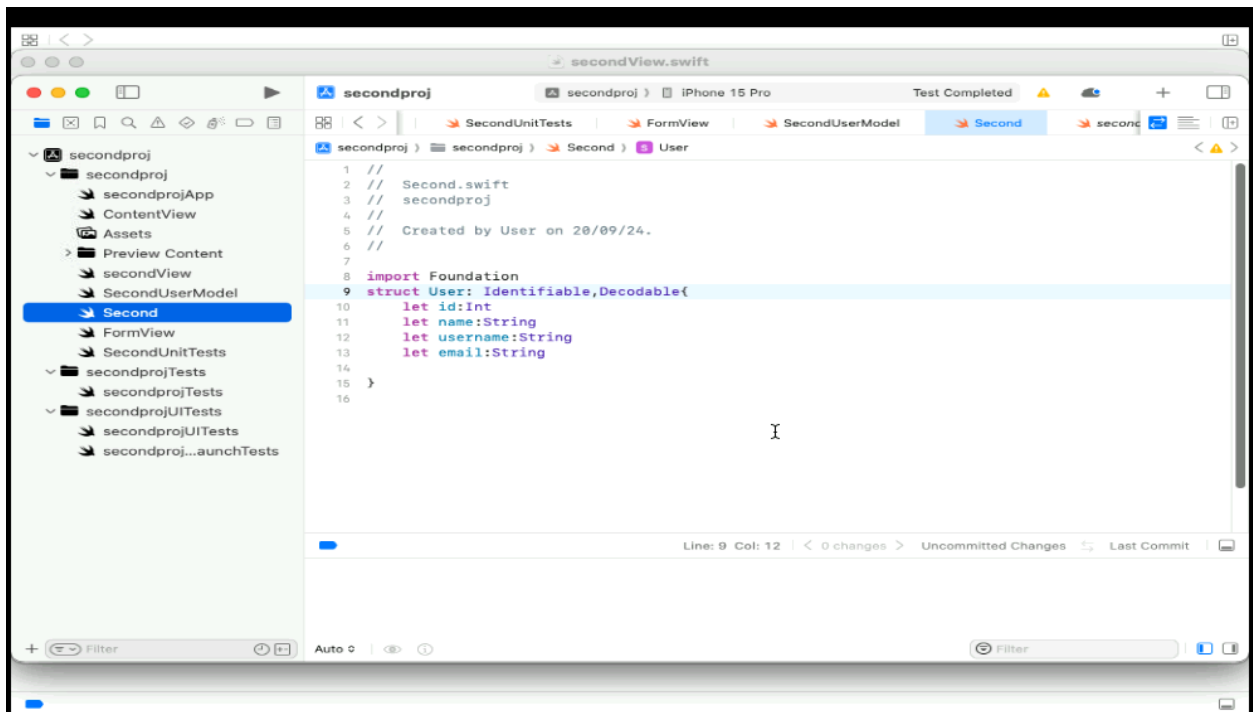
```

SECONDVIEW CLASS



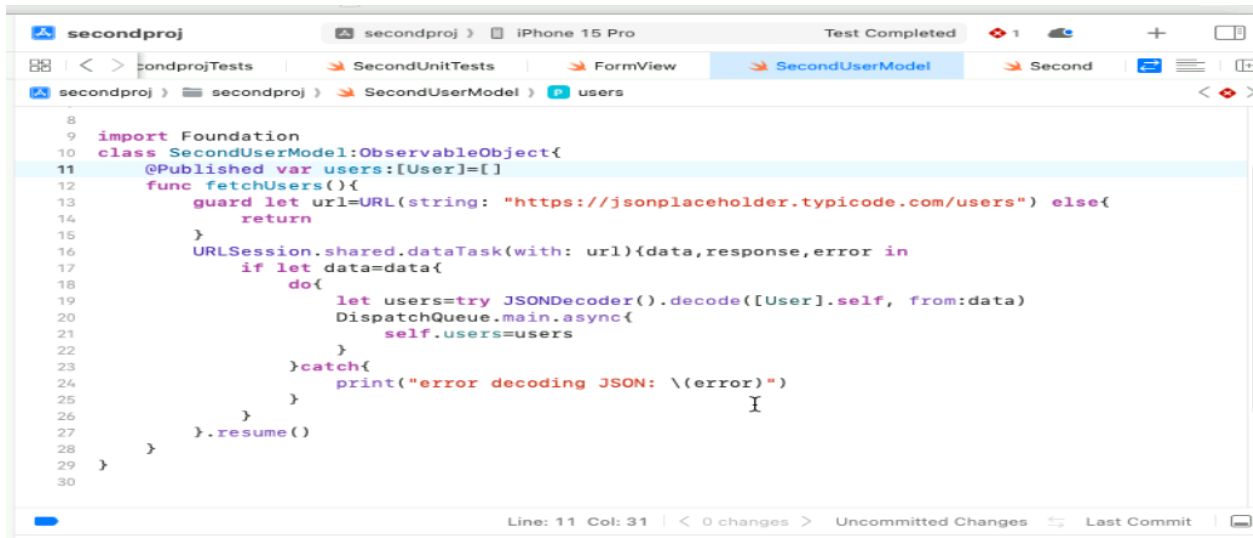
```
secondView.swift
secondproj › iPhone 15 Pro Test Completed 1
secondproj › ContentView › secondprojTests › SecondUnitTests › FormView
secondproj › secondproj › secondView › body
7
8 import SwiftUI
9
10 struct secondView: View {
11     @StateObject private var viewModel=SecondUserModel()
12     var body: some View {
13         NavigationView{
14             List(viewModel.users){user in
15                 VStack(alignment: .leading){
16
17                     Text("ID is \(user.id)")
18                     Text(user.name)
19                     Text(user.email)|
20                 }
21                 .padding(.vertical,4)
22             }
23             .navigationTitle("Users")
24             .onAppear{
25                 viewModel.fetchUsers()
26             }
27         }
28     }
29 }
30
Line: 19 Col: 37 | < 0 changes > Uncommitted Changes Last Commit
```

SECONDCLASS



```
secondView.swift
secondproj › iPhone 15 Pro Test Completed
secondproj › SecondUnitTests › FormView › SecondUserModel › Second › second
secondproj › secondproj › Second › User
1 //
2 // Second.swift
3 // secondproj
4 // Created by User on 20/09/24.
5 //
6 //
7
8 import Foundation
9 struct User: Identifiable,Decodable{
10
11     let id:Int
12     let name:String
13     let username:String
14     let email:String
15 }
16
Line: 9 Col: 12 | < 0 changes > Uncommitted Changes Last Commit
```

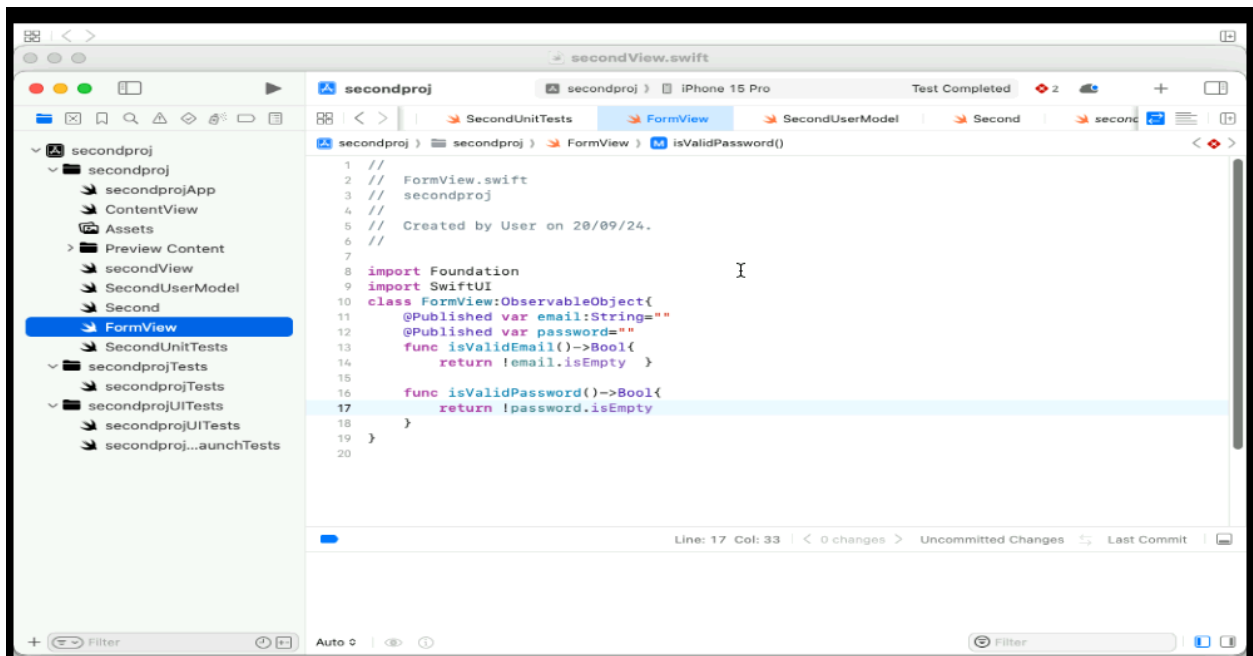
SECONDUSERMODEL CLASS



```
8
9 import Foundation
10 class SecondUserModel:ObservableObject{
11     @Published var users:[User]=[]
12     func fetchUsers(){
13         guard let url=URL(string: "https://jsonplaceholder.typicode.com/users") else{
14             return
15         }
16         URLSession.shared.dataTask(with: url){data,response,error in
17             if let data=data{
18                 do{
19                     let users=try JSONDecoder().decode([User].self, from:data)
20                     DispatchQueue.main.async{
21                         self.users=users
22                     }
23                 }catch{
24                     print("error decoding JSON: \(error)")
25                 }
26             }
27         }.resume()
28     }
29 }
30
```

Line: 11 Col: 31 | < 0 changes > Uncommitted Changes Last Commit

FORMVIEW



```
1 //
2 // FormView.swift
3 // secondproj
4 // Created by User on 20/09/24.
5 //
6 //
7
8 import Foundation
9 import SwiftUI
10 class FormView:ObservableObject{
11     @Published var email:String=""
12     @Published var password=""
13     func isValidEmail()->Bool{
14         return !email.isEmpty }
15
16     func isValidPassword()->Bool{
17         return !password.isEmpty
18     }
19 }
20
```

Line: 17 Col: 33 | < 0 changes > Uncommitted Changes Last Commit

REGISTRATIONVIEW

secondproj > secondproj > RegistrationView > S RegistrationView

```
8 import SwiftUI
9
10 struct RegistrationView: View {
11     @State private var username:String=""
12     @State private var password:String=""
13     @State private var registrationMessage=""
14     @Binding var isRegistered:Bool
15     @Binding var storedUsername:String
16     @Binding var storedPassword:String
17     var body: some View {
18         VStack{
19             Text("Register")
20                 .font(.largeTitle)
21                 .padding()
22
23             TextField("Username",text:$username)
24                 .textFieldStyle(RoundedBorderTextFieldStyle())
25                 .padding()
26
27             SecureField("Password",text: $password)
28                 .textFieldStyle(RoundedBorderTextFieldStyle())
29                 .padding()
30
31             Button(action:
32                 {register()}
33             ){
34                 Text("Register")
35                     .foregroundColor(.white)
36                     .padding()
37                     .background(Color.green)
38                     .cornerRadius(8)
39             }
40                 .padding()
41             Text(registrationMessage)
42                 .foregroundColor(.red)
43                 .padding()
44             Spacer()
45         }
```

```
46         .padding()
47     }
48     func register(){
49         if username.isEmpty || password.isEmpty{
50             registrationMessage="Both fields are required!"
51             return
52         }
53         storedUsername=username
54         storedPassword=password
55         isRegistered=true
56     }
57 }
58
59 //Preview {
60 //     RegistrationView()
61 //}
62
```


TEST CASES

