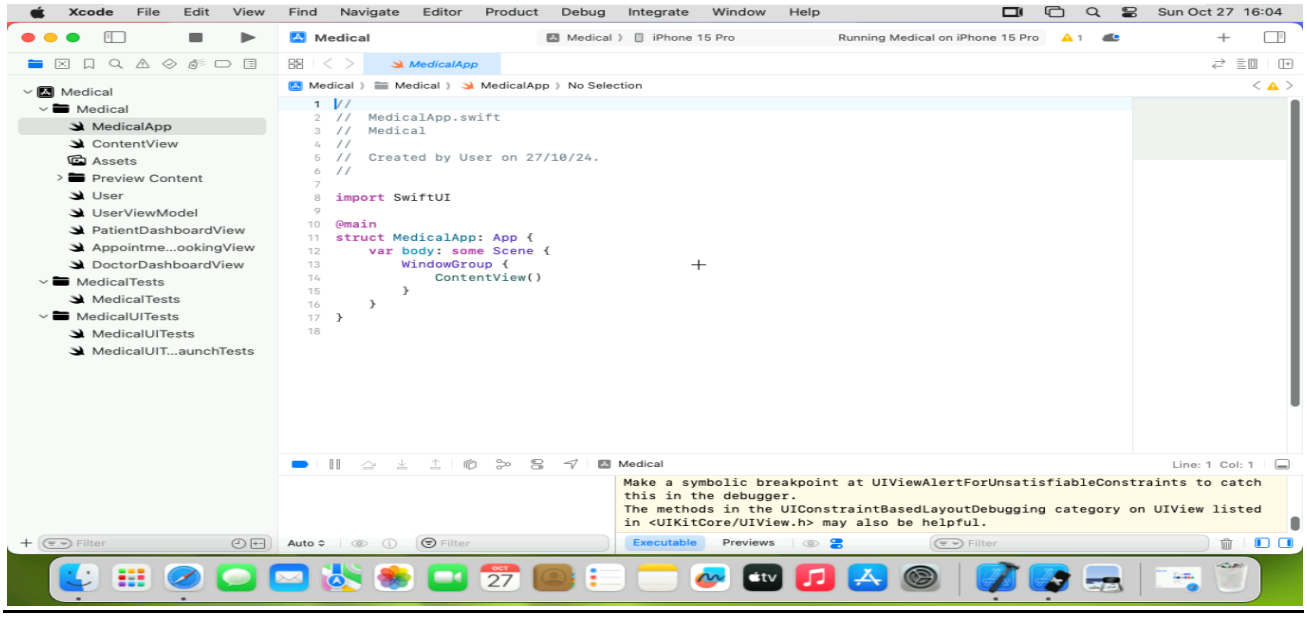
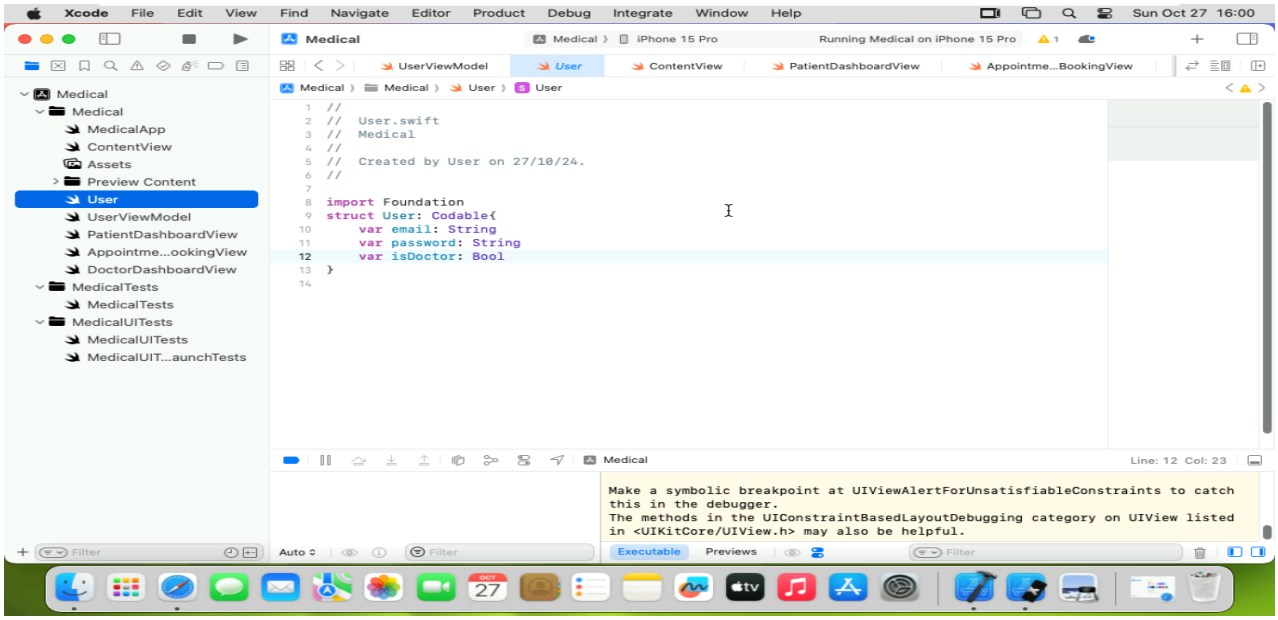
IOS Application

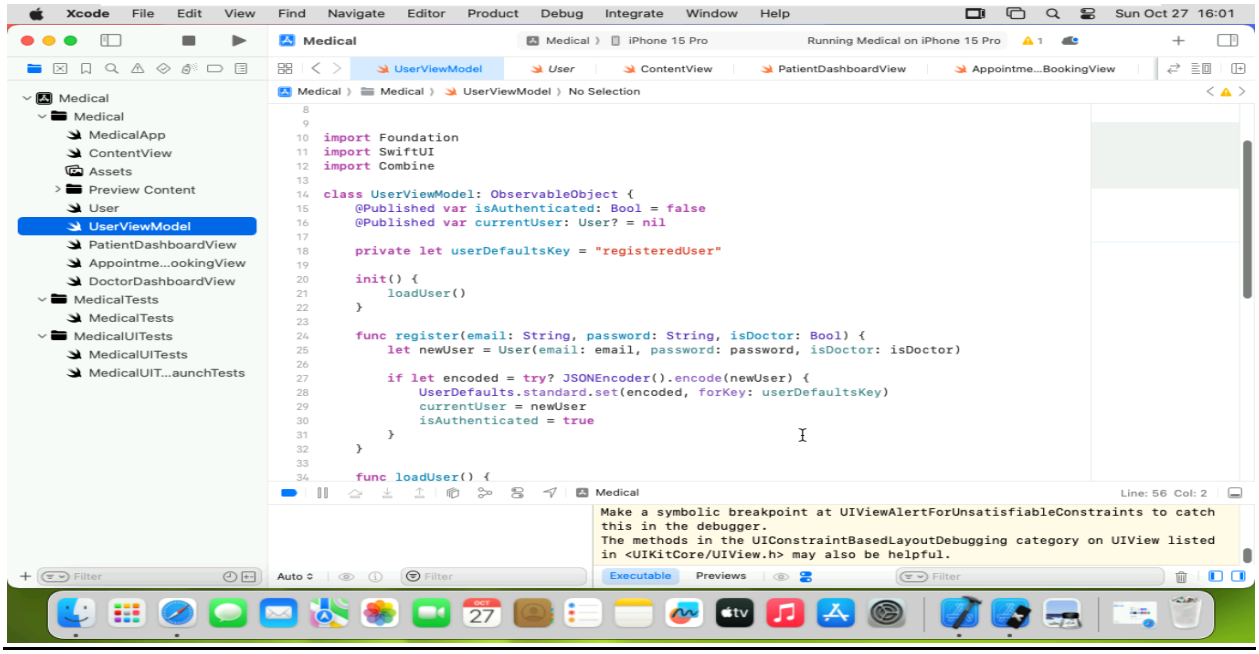
MedicalApp:



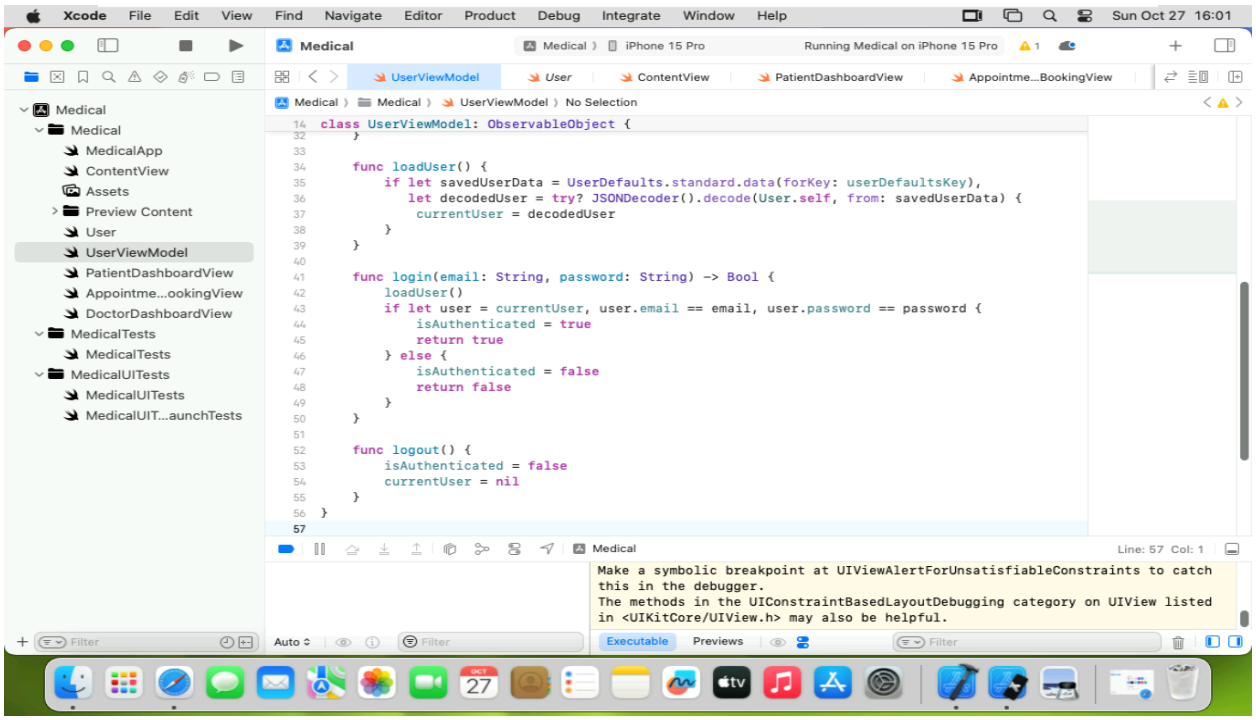
User:



UserViewModel:

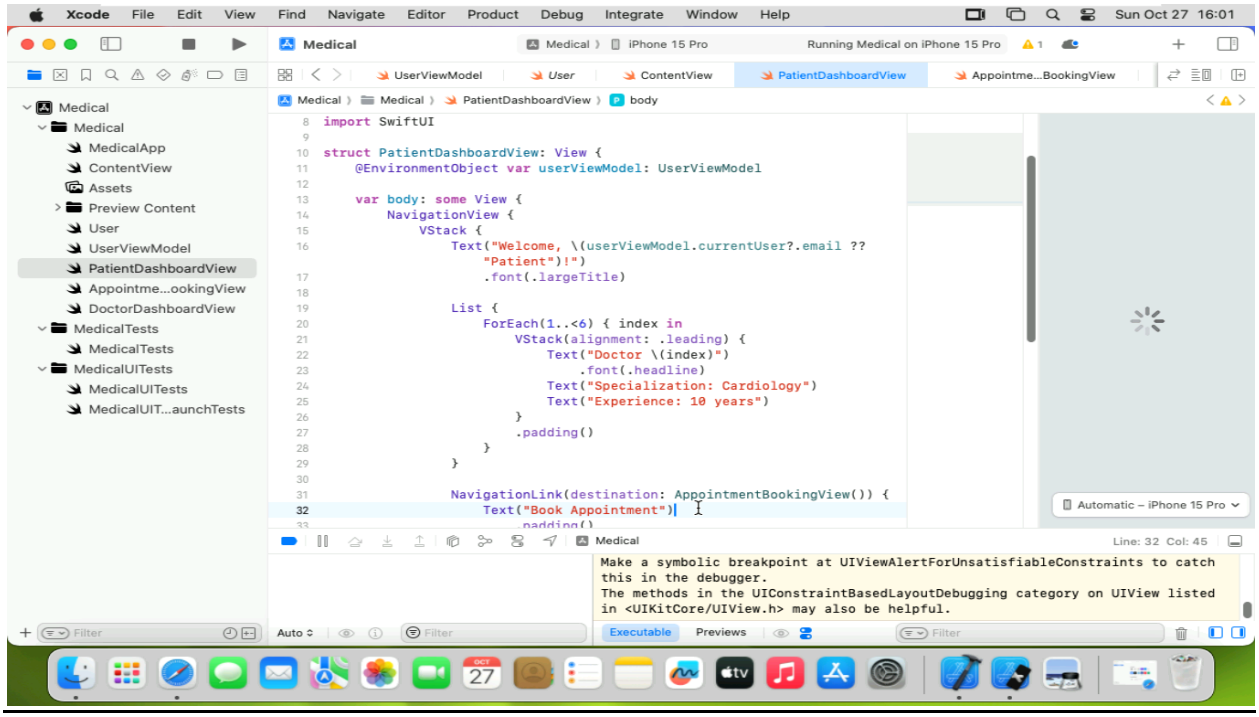


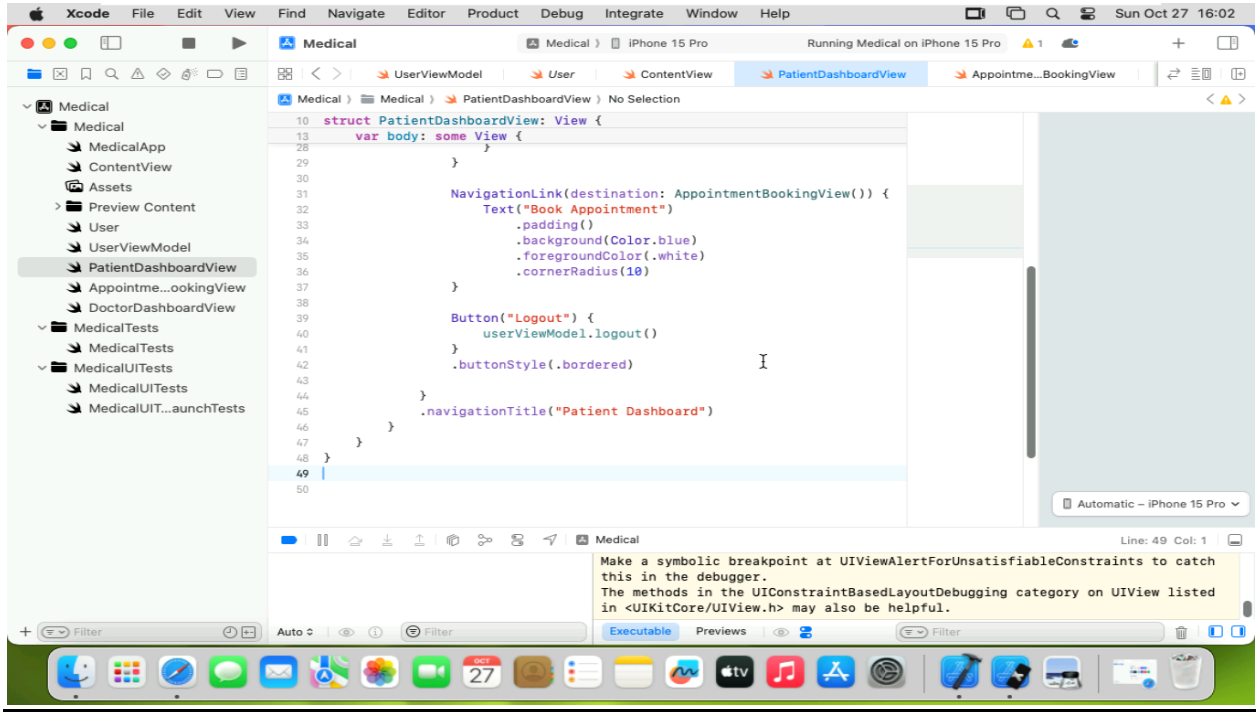
```
import Foundation
import SwiftUI
import Combine

class UserViewModel: ObservableObject {
    @Published var isAuthenticated: Bool = false
    @Published var currentUser: User? = nil

    private let userDefaultsKey = "registeredUser"

    init() {
        loadUser()
    }

    func register(email: String, password: String, isDoctor: Bool) {
        let newUser = User(email: email, password: password, isDoctor: isDoctor)

        if let encoded = try? JSONEncoder().encode(newUser) {
            UserDefaults.standard.set(encoded, forKey: userDefaultsKey)
            currentUser = newUser
            isAuthenticated = true
        }
    }

    func loadUser() {
```



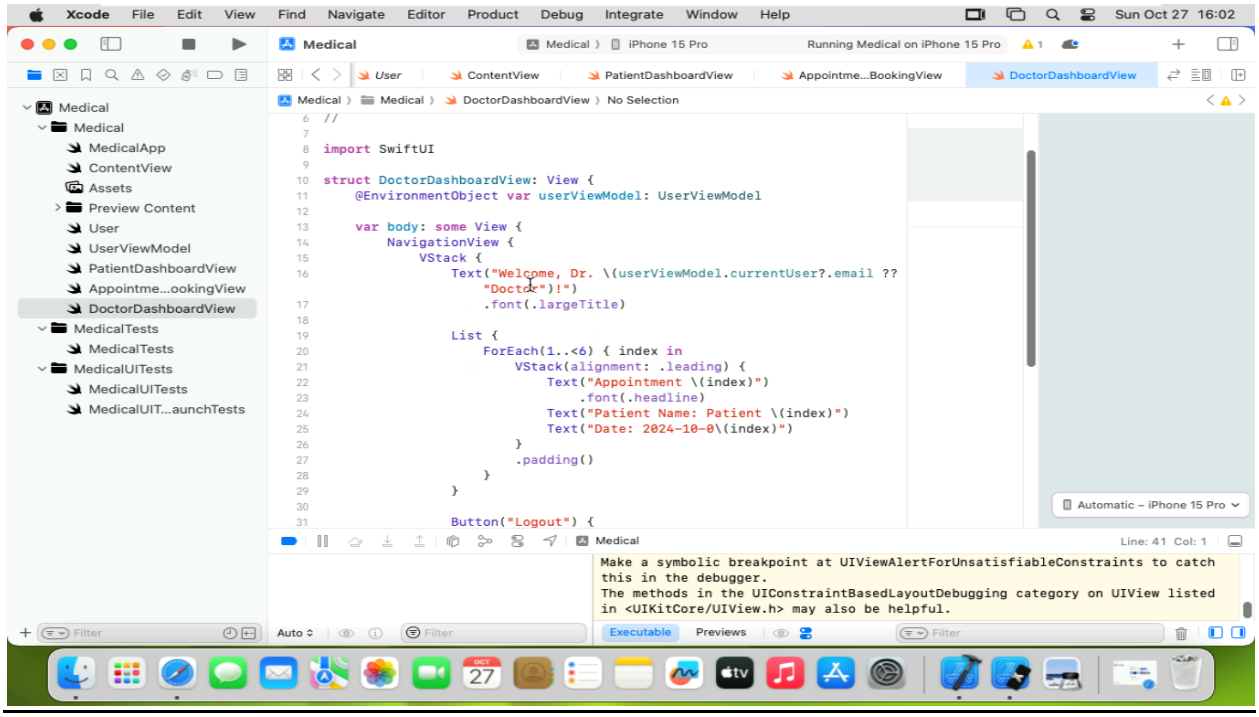
```
class UserViewModel: ObservableObject {
    }

    func loadUser() {
        if let savedUserData = UserDefaults.standard.data(forKey: userDefaultsKey),
           let decodedUser = try? JSONDecoder().decode(User.self, from: savedUserData) {
            currentUser = decodedUser
        }
    }

    func login(email: String, password: String) -> Bool {
        loadUser()
        if let user = currentUser, user.email == email, user.password == password {
            isAuthenticated = true
            return true
        } else {
            isAuthenticated = false
            return false
        }
    }

    func logout() {
        isAuthenticated = false
        currentUser = nil
    }
}
```

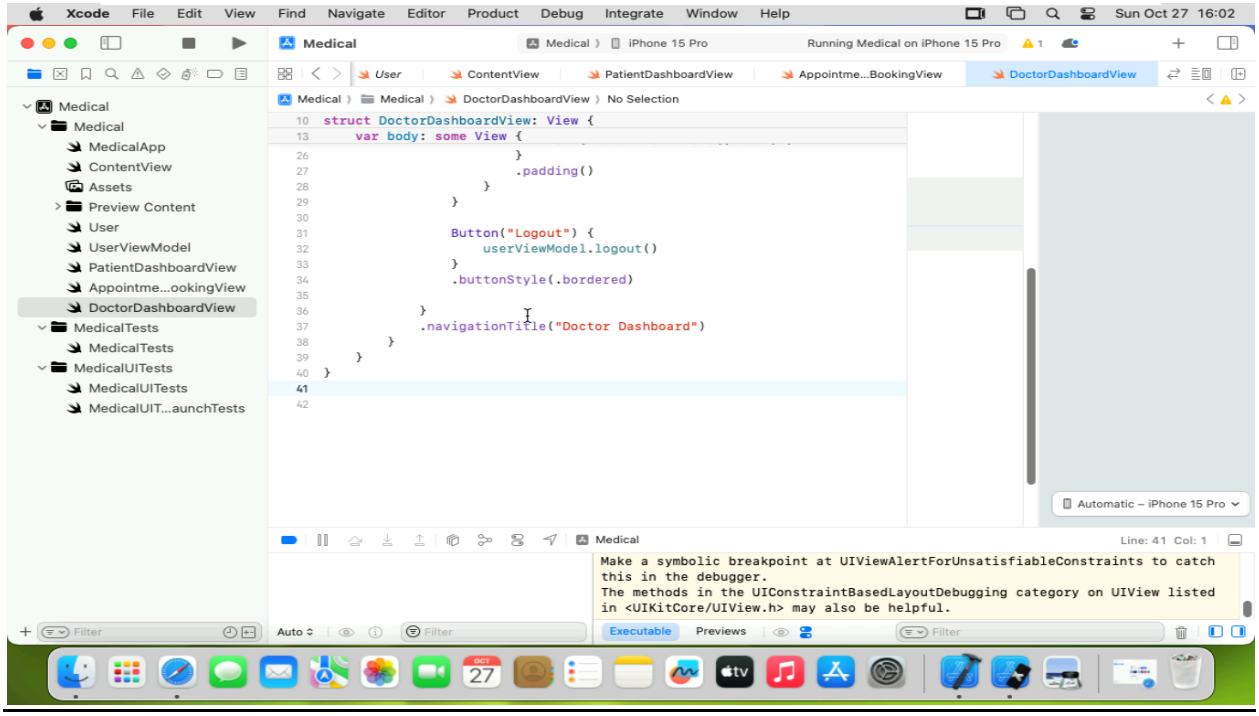
PatientViewDashboard:



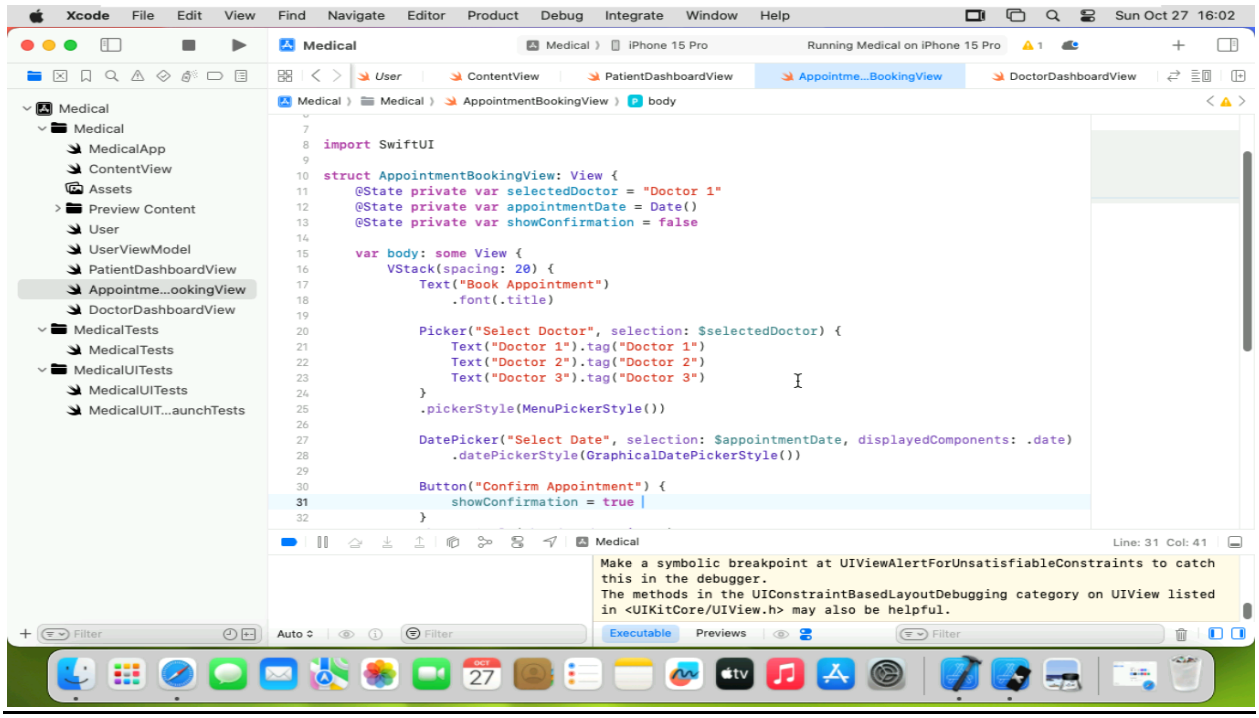


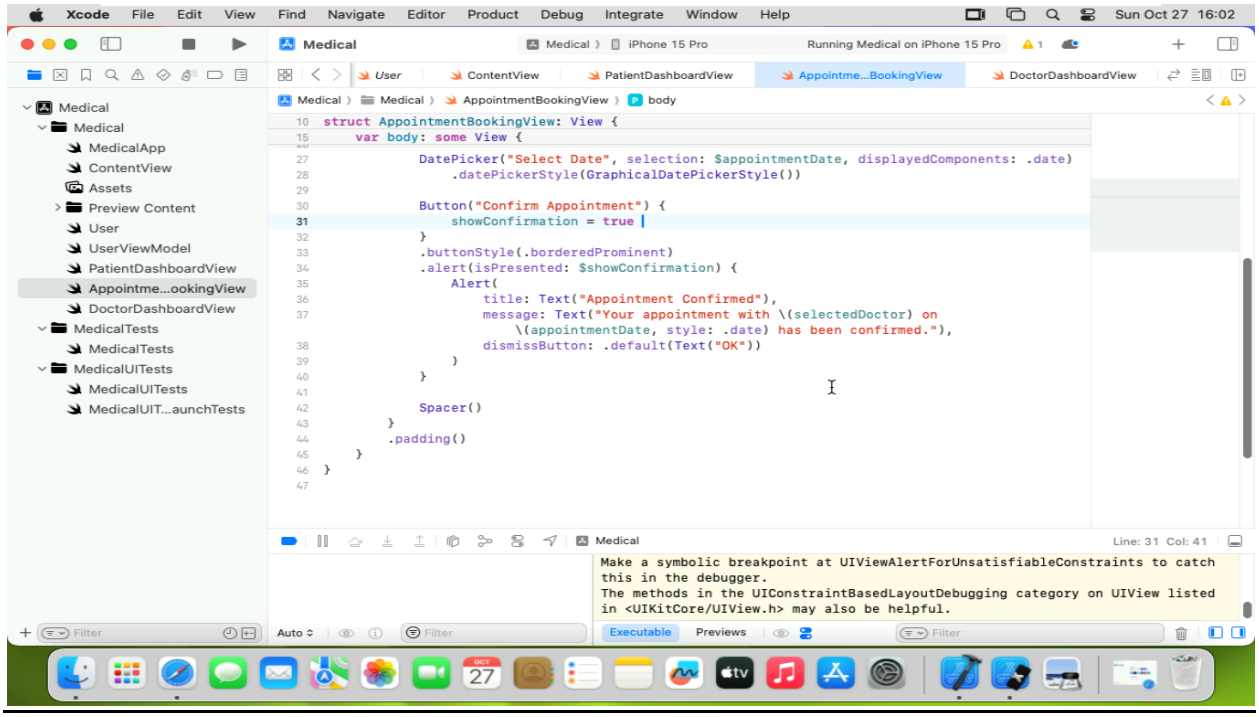
DoctorDashboardView:



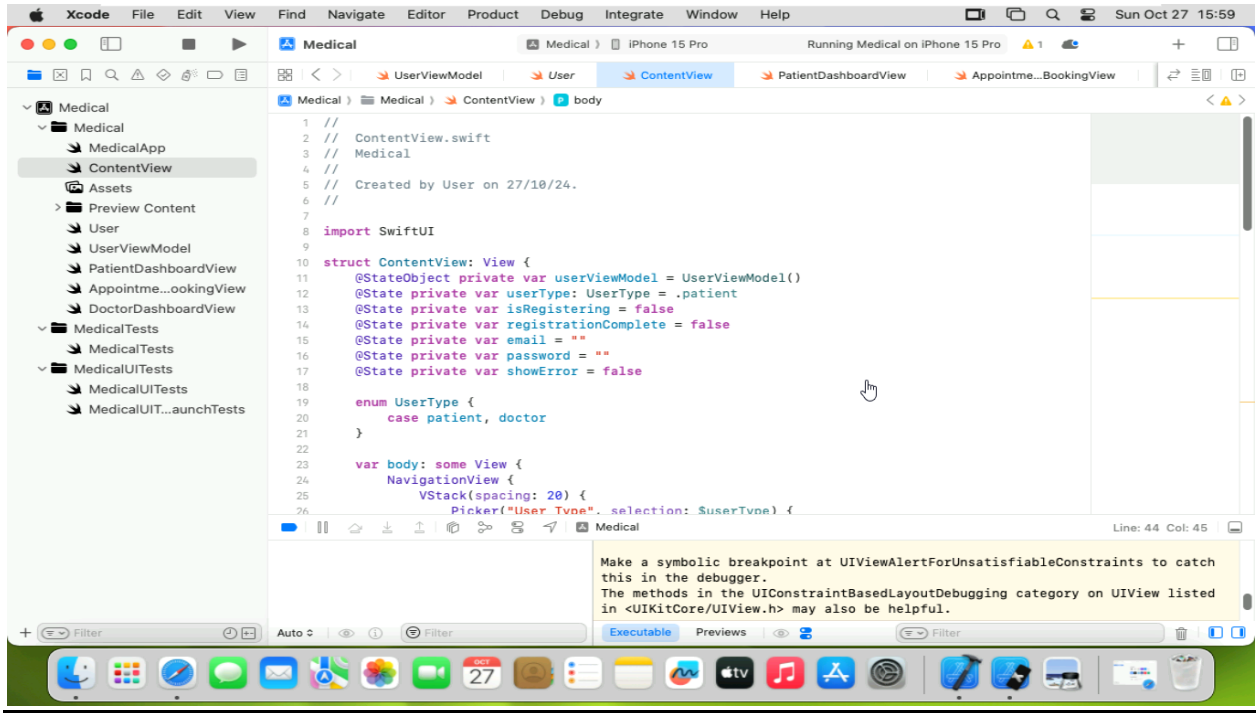


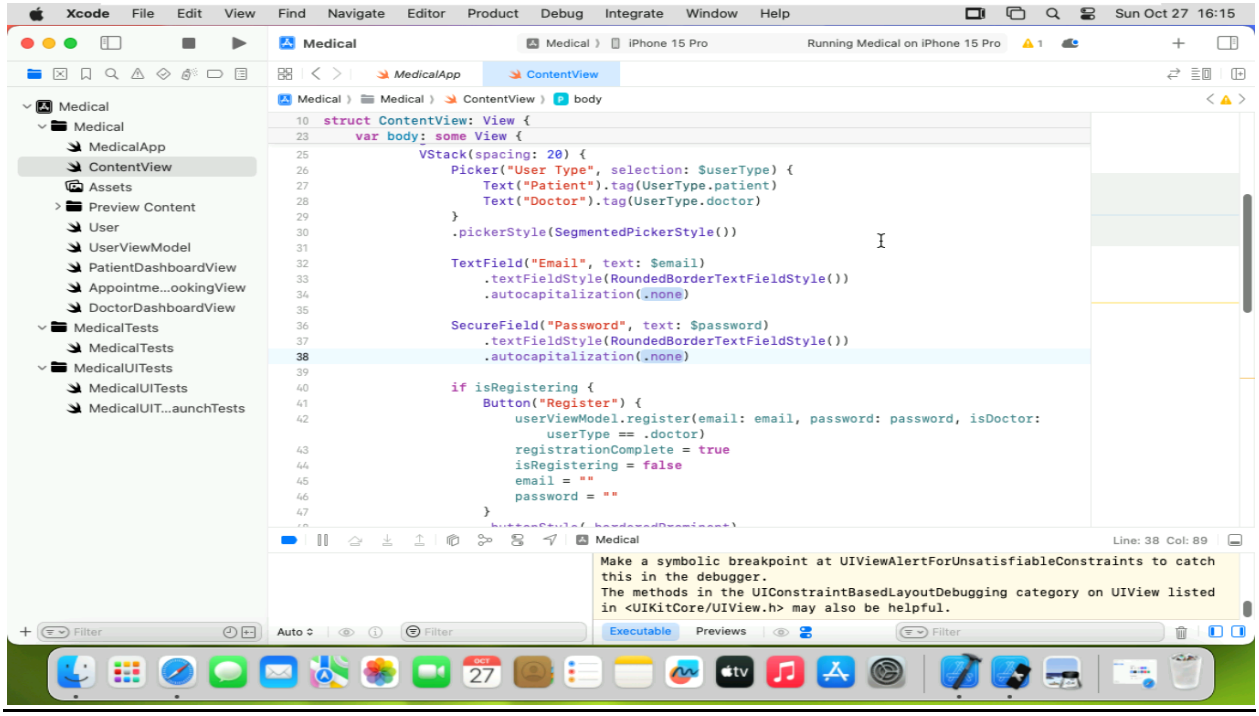
AppointmentBookingView:

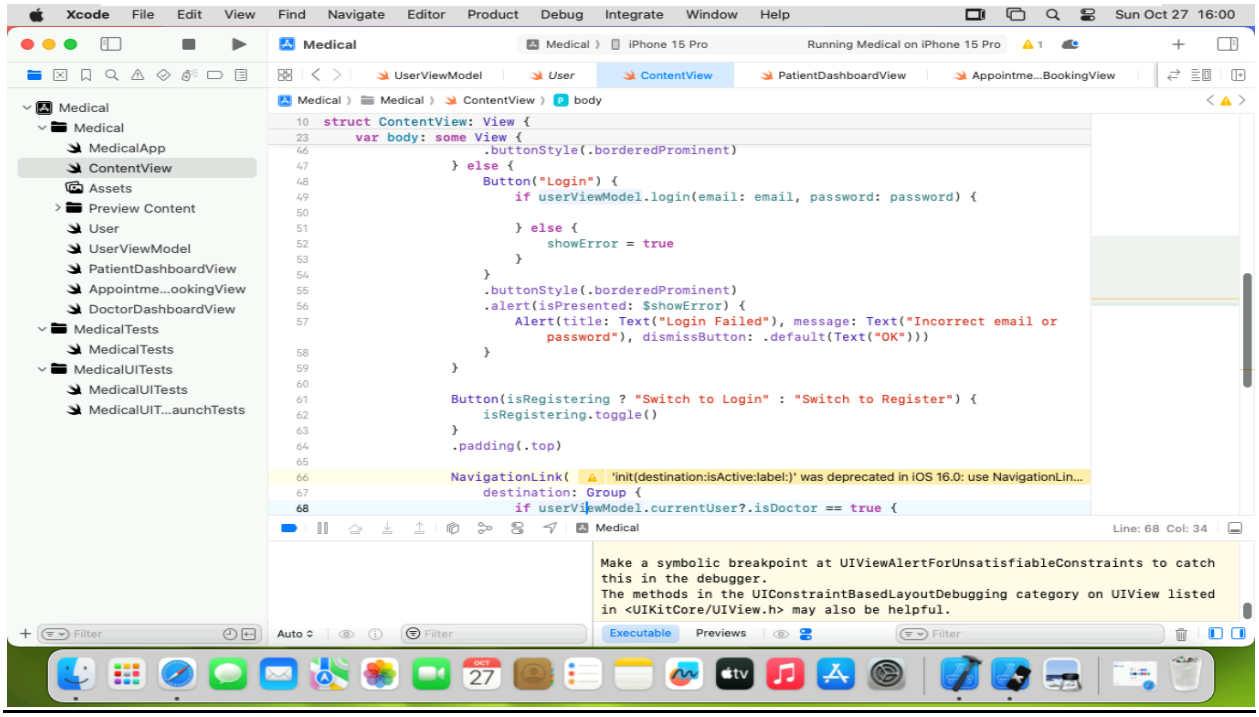


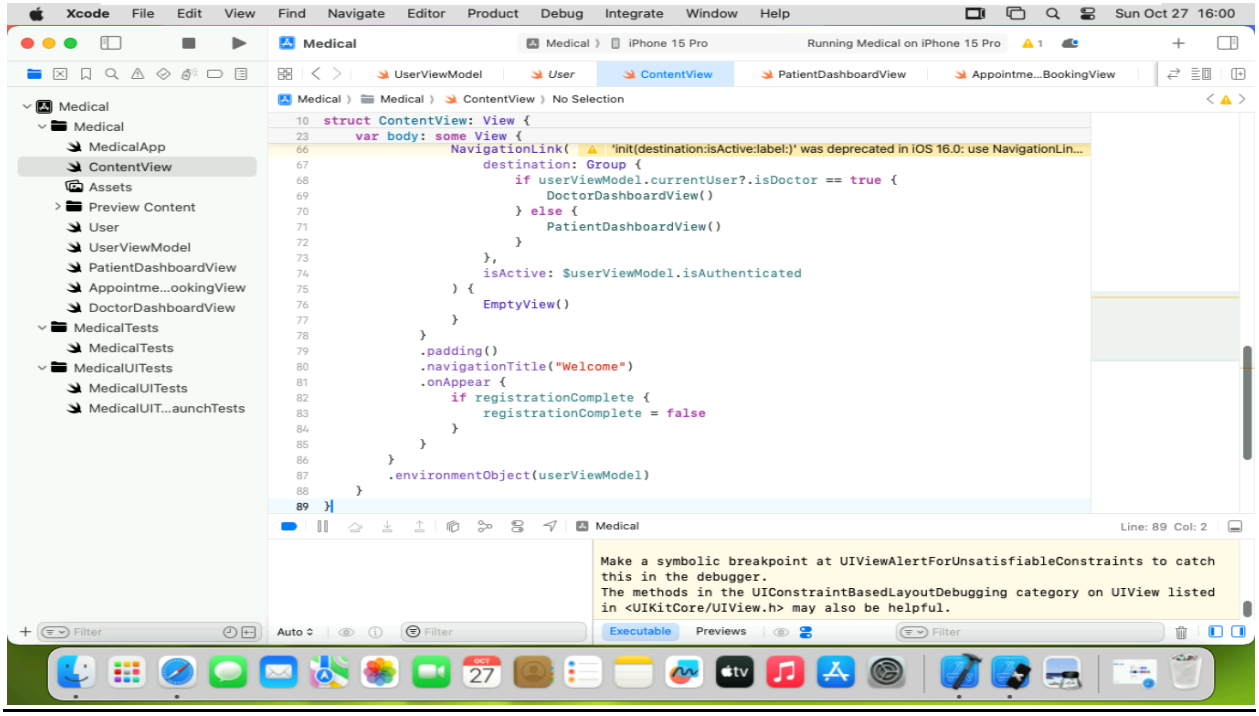


ContentView:

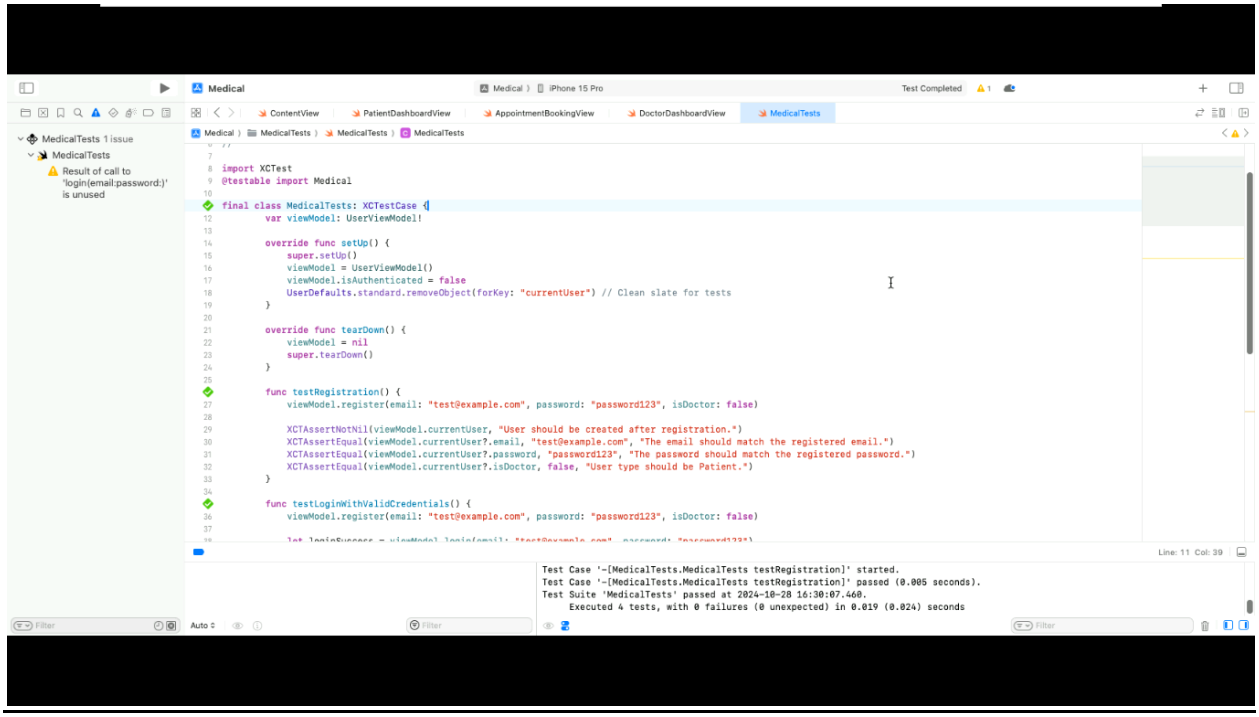


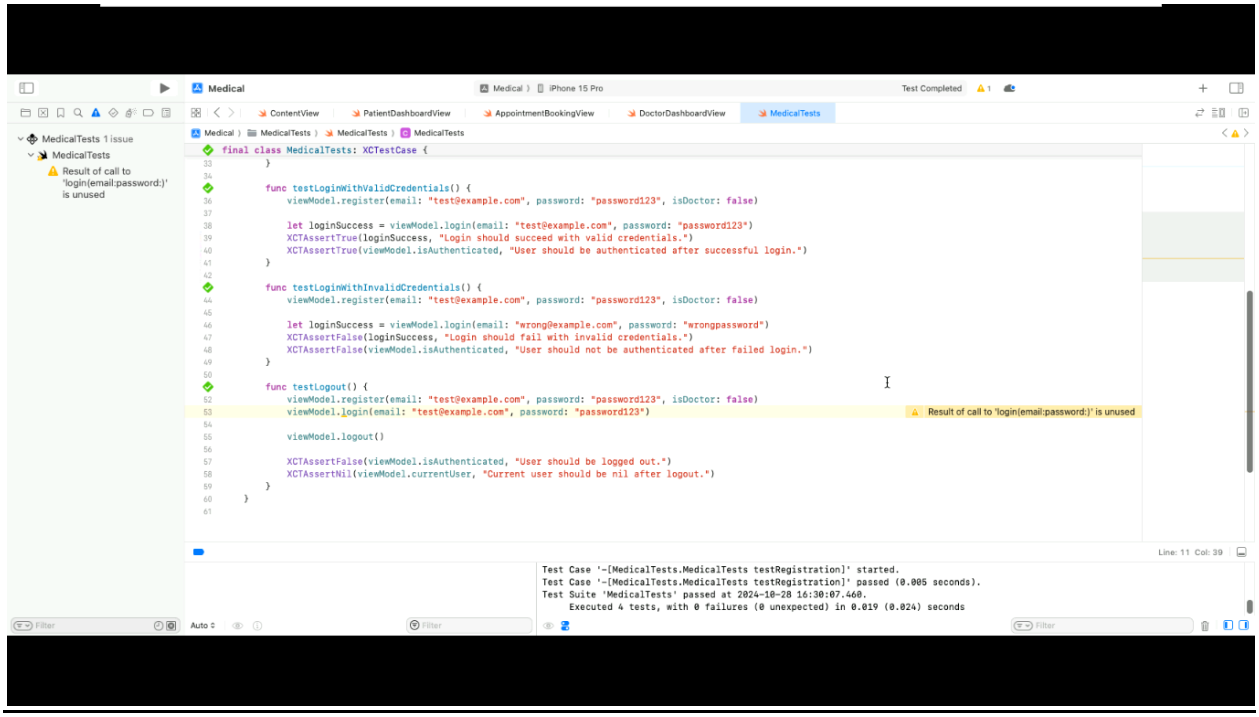



Xcode File Edit View Find Navigate Editor Product Debug Integrate Window Help
Sun Oct 27 16:00
Medical
Medical › iPhone 15 Pro
Running Medical on iPhone 15 Pro
UserViewModel
User
ContentView
PatientDashboardView
Appointme...BookingView
Medical › Medical › ContentView › body
struct ContentView: View {
var body: some View {
.buttonStyle(.borderedProminent)
} else {
Button("Login") {
if userViewModel.login(email: email, password: password) {
} else {
showError = true
}
}
.buttonStyle(.borderedProminent)
.alert(isPresented: $showError) {
Alert(title: Text("Login Failed"), message: Text("Incorrect email or password"), dismissButton: .default(Text("OK")))
}
}
Button(isRegistering ? "Switch to Login" : "Switch to Register") {
isRegistering.toggle()
}
.padding(.top)
NavigationLink('init(destination:isActive:label:)' was deprecated in iOS 16.0: use NavigationLin...
destination: Group {
if userViewModel.currentUser?.isDoctor == true {
Line: 68 Col: 34
Make a symbolic breakpoint at UIViewAlertForUnsatisfiableConstraints to catch this in the debugger.
The methods in the UIConstraintBasedLayoutDebugging category on UIView listed in <UIKitCore/UIView.h> may also be helpful.
Executable
Previews
Medical
MedicalApp
ContentView
Assets
Preview Content
User
UserViewModel
PatientDashboardView
Appointme...ookingView
DoctorDashboardView
MedicalTests
MedicalTests
MedicalUITests
MedicalUITests
MedicalUIT...aunchTests


Xcode File Edit View Find Navigate Editor Product Debug Integrate Window Help
Sun Oct 27 16:00
Medical
Medical › iPhone 15 Pro
Running Medical on iPhone 15 Pro
UserViewModel
User
ContentView
PatientDashboardView
Appointme...BookingView
Medical › Medical › ContentView › No Selection
struct ContentView: View {
var body: some View {
NavigationLink('init(destination:isActive:label:)' was deprecated in iOS 16.0: use NavigationLin...
destination: Group {
if userViewModel.currentUser?.isDoctor == true {
DoctorDashboardView()
} else {
PatientDashboardView()
}
},
isActive: $userViewModel.isAuthenticated
) {
EmptyView()
}
}
.padding()
.navigationTitle("Welcome")
.onAppear {
if registrationComplete {
registrationComplete = false
}
}
}
.environmentObject(userViewModel)
}
}
Line: 89 Col: 2
Make a symbolic breakpoint at UIViewAlertForUnsatisfiableConstraints to catch this in the debugger.
The methods in the UIConstraintBasedLayoutDebugging category on UIView listed in <UIKitCore/UIView.h> may also be helpful.
Executable
Previews
Medical
MedicalApp
ContentView
Assets
Preview Content
User
UserViewModel
PatientDashboardView
Appointme...ookingView
DoctorDashboardView
MedicalTests
MedicalTests
MedicalUITests
MedicalUITests
MedicalUIT...aunchTests

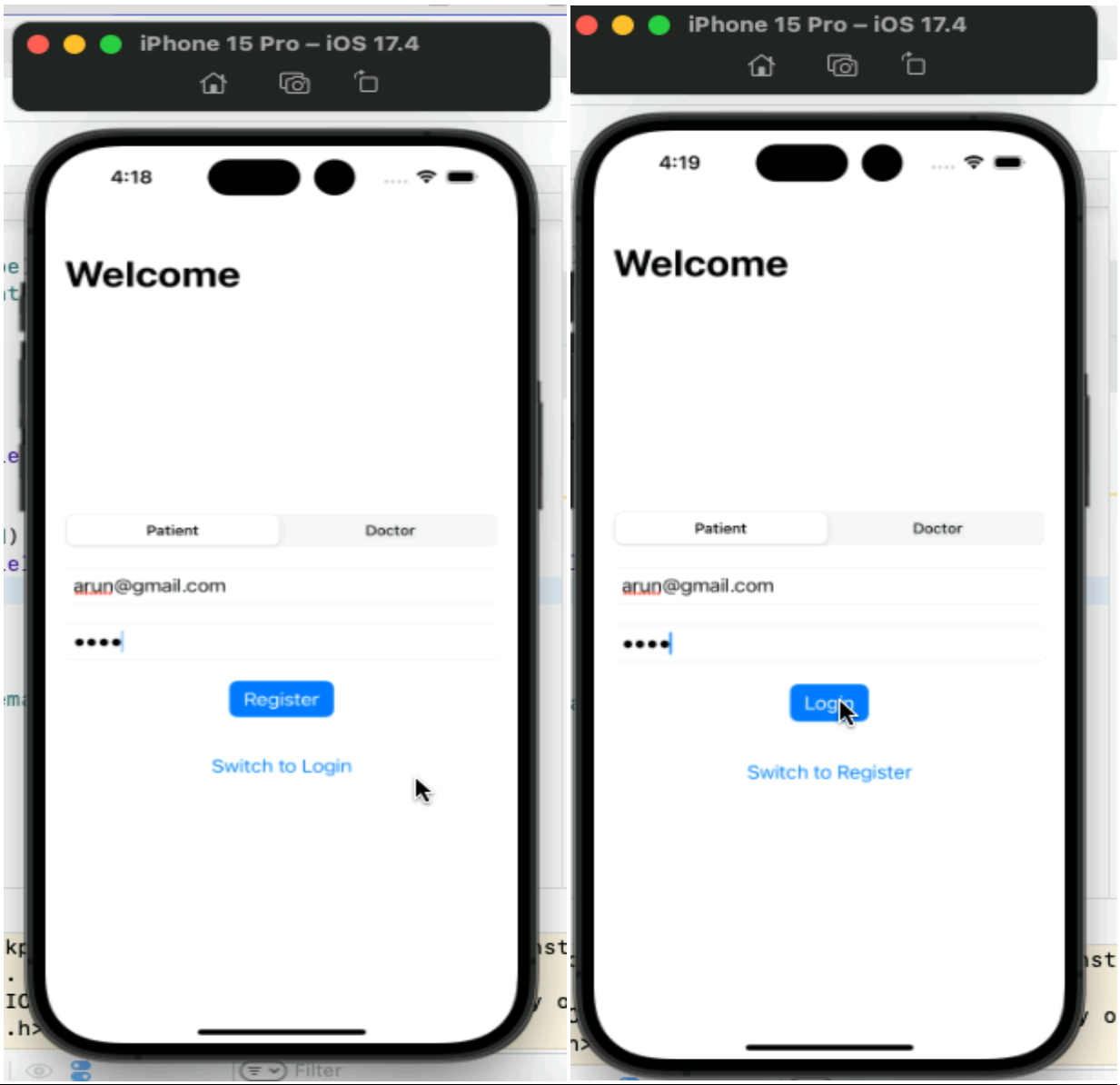
Testcases:


Medical
Medical) iPhone 15 Pro
Test Completed
ContentView
PatientDashboardView
AppointmentBookingView
DoctorDashboardView
MedicalTests
MedicalTests 1 issue
MedicalTests
Result of call to 'login(email:password:)' is unused
import XCTest
@testable import Medical
final class MedicalTests: XCTestCase {
var viewModel: UserViewModel!
override func setUp() {
super.setUp()
viewModel = UserViewModel()
viewModel.isAuthenticated = false
UserDefaults.standard.removeObject(forKey: "currentUser") // Clean slate for tests
}
override func tearDown() {
viewModel = nil
super.tearDown()
}
func testRegistration() {
viewModel.register(email: "test@example.com", password: "password123", isDoctor: false)
XCTAssertNotNil(viewModel.currentUser, "User should be created after registration.")
XCTAssertEqual(viewModel.currentUser?.email, "test@example.com", "The email should match the registered email.")
XCTAssertEqual(viewModel.currentUser?.password, "password123", "The password should match the registered password.")
XCTAssertEqual(viewModel.currentUser?.isDoctor, false, "User type should be Patient.")
}
func testLoginWithValidCredentials() {
viewModel.register(email: "test@example.com", password: "password123", isDoctor: false)
Test Case '-[MedicalTests.MedicalTests testRegistration]' started.
Test Case '-[MedicalTests.MedicalTests testRegistration]' passed (0.005 seconds).
Test Suite 'MedicalTests' passed at 2024-10-28 16:30:07.460.
Executed 4 tests, with 0 failures (0 unexpected) in 0.019 (0.024) seconds

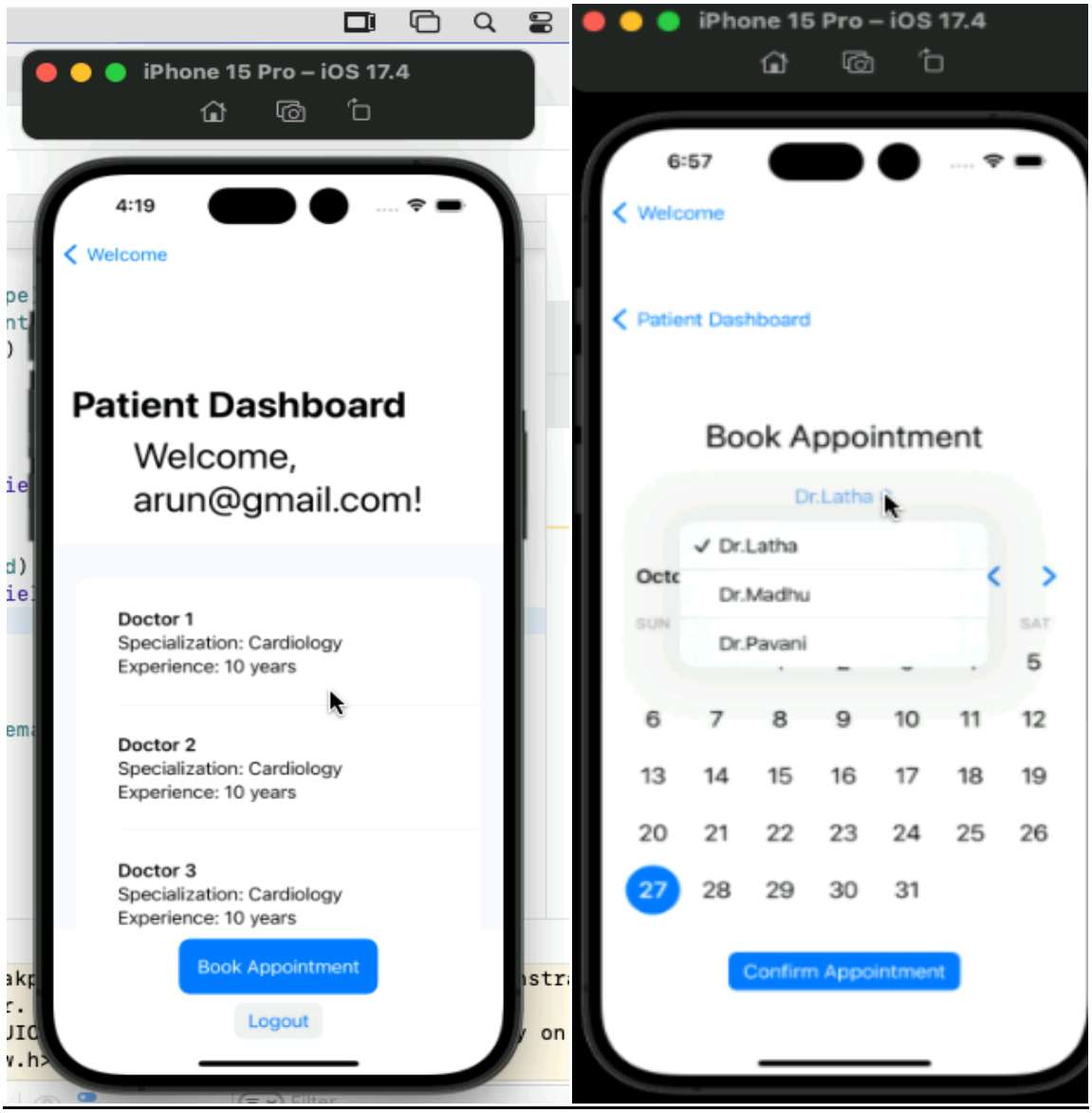

Medical
Medical) iPhone 15 Pro
Test Completed
ContentView
PatientDashboardView
AppointmentBookingView
DoctorDashboardView
MedicalTests
MedicalTests 1 issue
MedicalTests
Result of call to 'login(email:password:)' is unused
final class MedicalTests: XCTestCase {
func testLoginWithValidCredentials() {
viewModel.register(email: "test@example.com", password: "password123", isDoctor: false)
let loginSuccess = viewModel.login(email: "test@example.com", password: "password123")
XCTAssertTrue(loginSuccess, "Login should succeed with valid credentials.")
XCTAssertTrue(viewModel.isAuthenticated, "User should be authenticated after successful login.")
}
func testLoginWithInvalidCredentials() {
viewModel.register(email: "test@example.com", password: "password123", isDoctor: false)
let loginSuccess = viewModel.login(email: "wrong@example.com", password: "wrongpassword")
XCTAssertFalse(loginSuccess, "Login should fail with invalid credentials.")
XCTAssertFalse(viewModel.isAuthenticated, "User should not be authenticated after failed login.")
}
func testLogout() {
viewModel.register(email: "test@example.com", password: "password123", isDoctor: false)
viewModel.login(email: "test@example.com", password: "password123")
Result of call to 'login(email:password:)' is unused
viewModel.logout()
XCTAssertFalse(viewModel.isAuthenticated, "User should be logged out.")
XCTAssertNil(viewModel.currentUser, "Current user should be nil after logout.")
}
}
Test Case '-[MedicalTests.MedicalTests testRegistration]' started.
Test Case '-[MedicalTests.MedicalTests testRegistration]' passed (0.005 seconds).
Test Suite 'MedicalTests' passed at 2024-10-28 16:30:07.460.
Executed 4 tests, with 0 failures (0 unexpected) in 0.019 (0.024) seconds

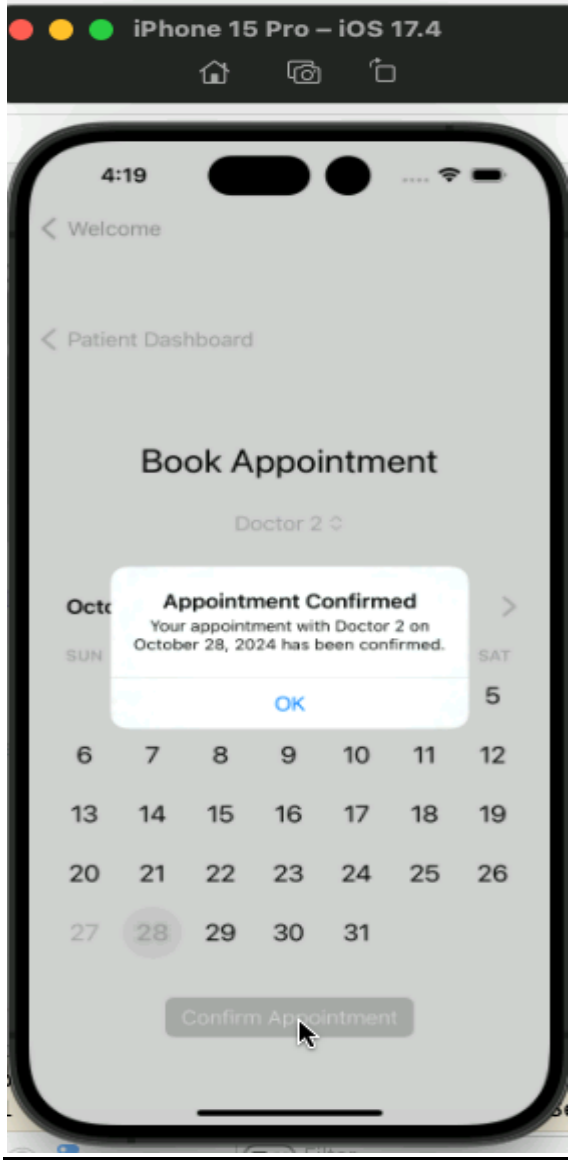
Outputs:

Patientlogin:

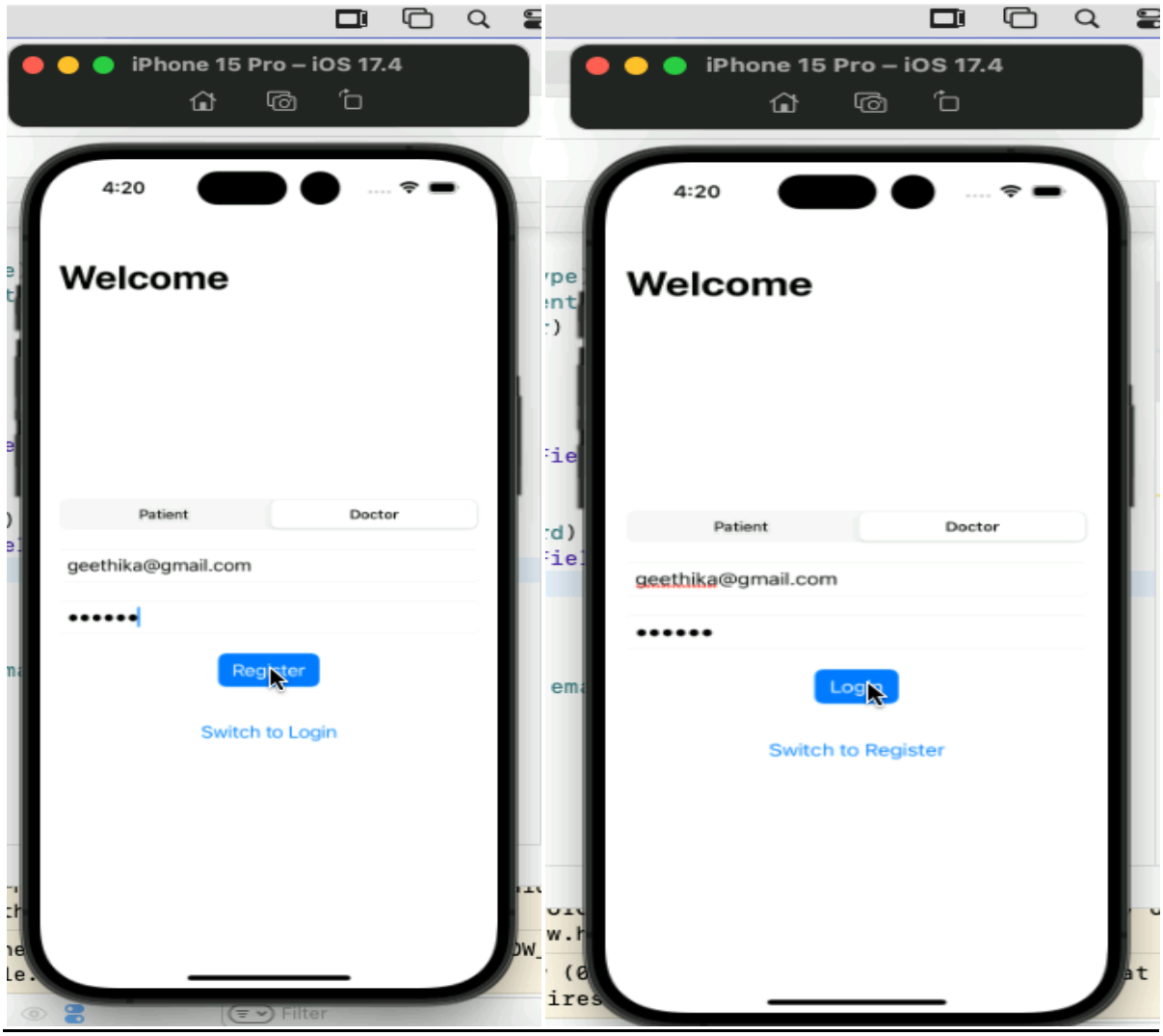


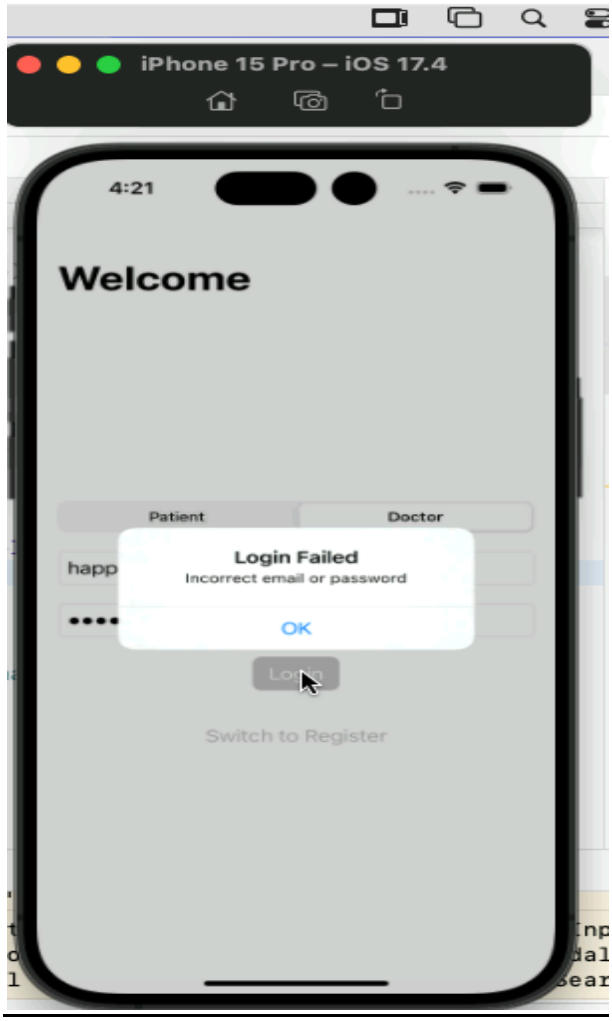
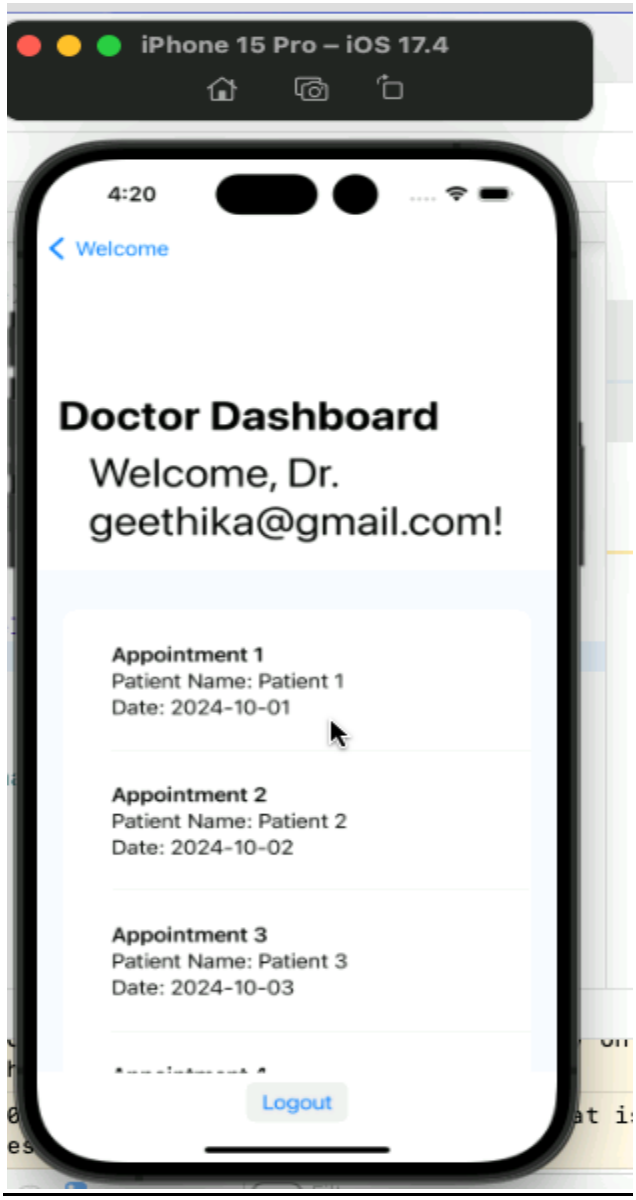
PatientDashboard:




iPhone 15 Pro – iOS 17.4
4:19
Welcome
Patient Dashboard
Book Appointment
Doctor 2
Appointment Confirmed
Your appointment with Doctor 2 on October 28, 2024 has been confirmed.
OK
SUN
SAT
5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
Confirm Appointment

DoctorLogin:




iPhone 15 Pro – iOS 17.4
4:21
Welcome
Patient
Doctor
happ
Login Failed
Incorrect email or password
OK
Switch to Register

DoctorDashboard: