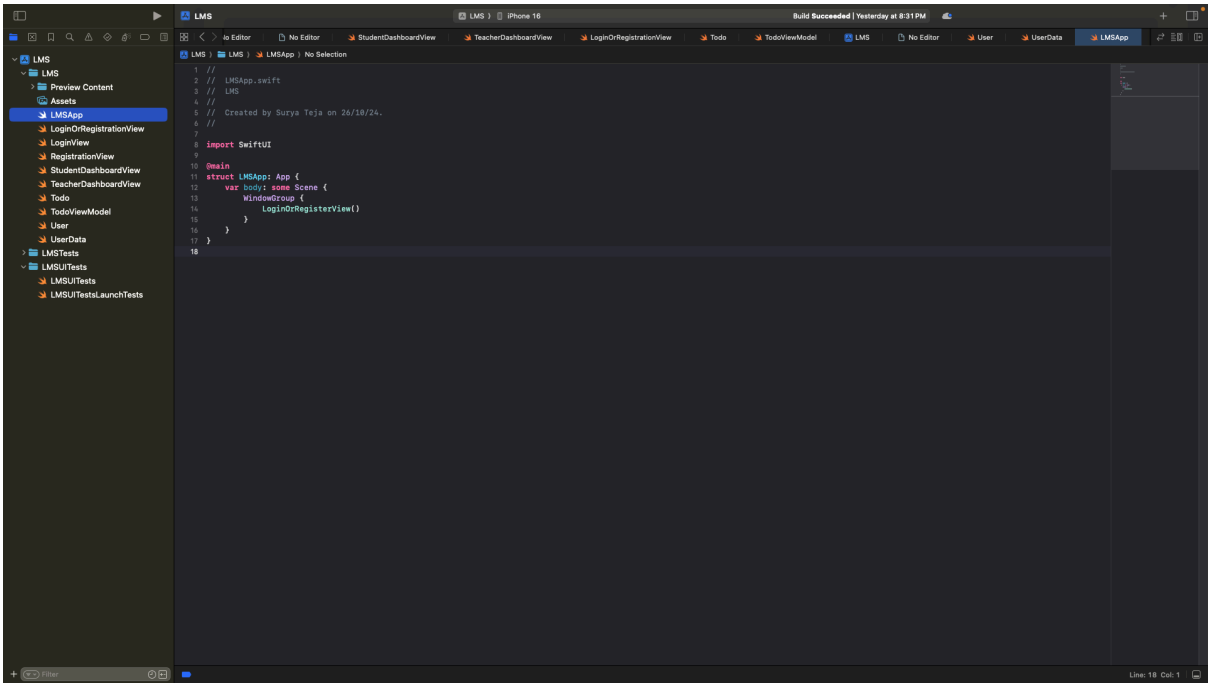
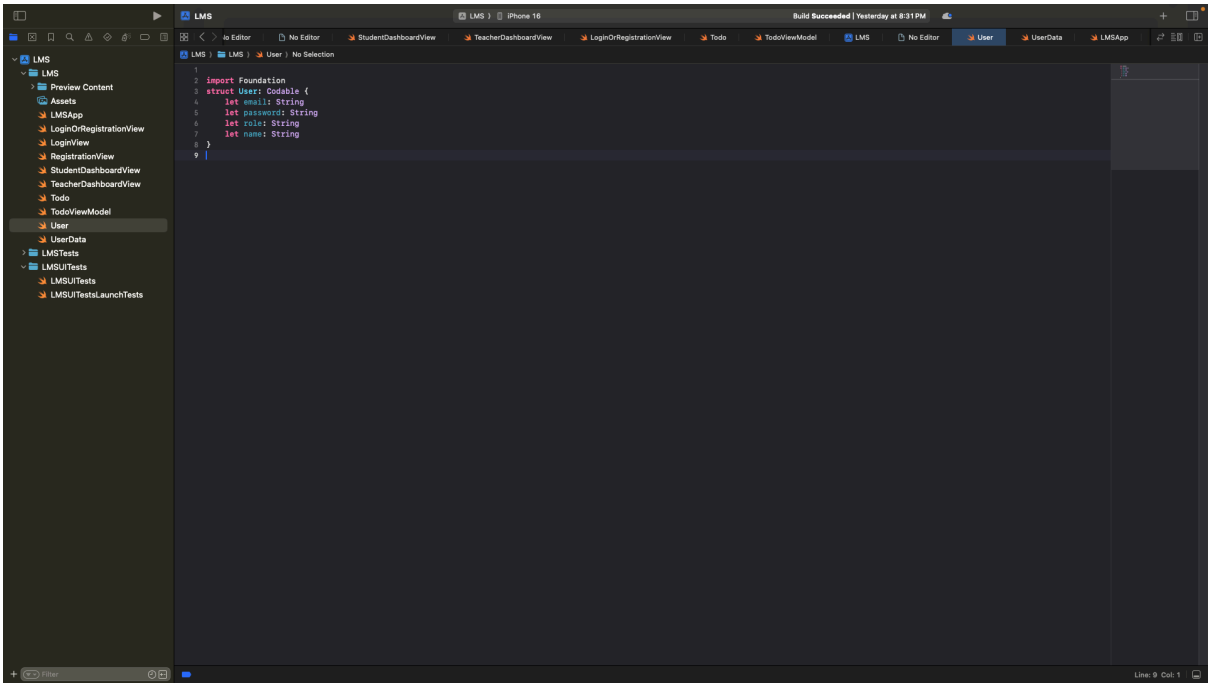
iOS application(Capstone)

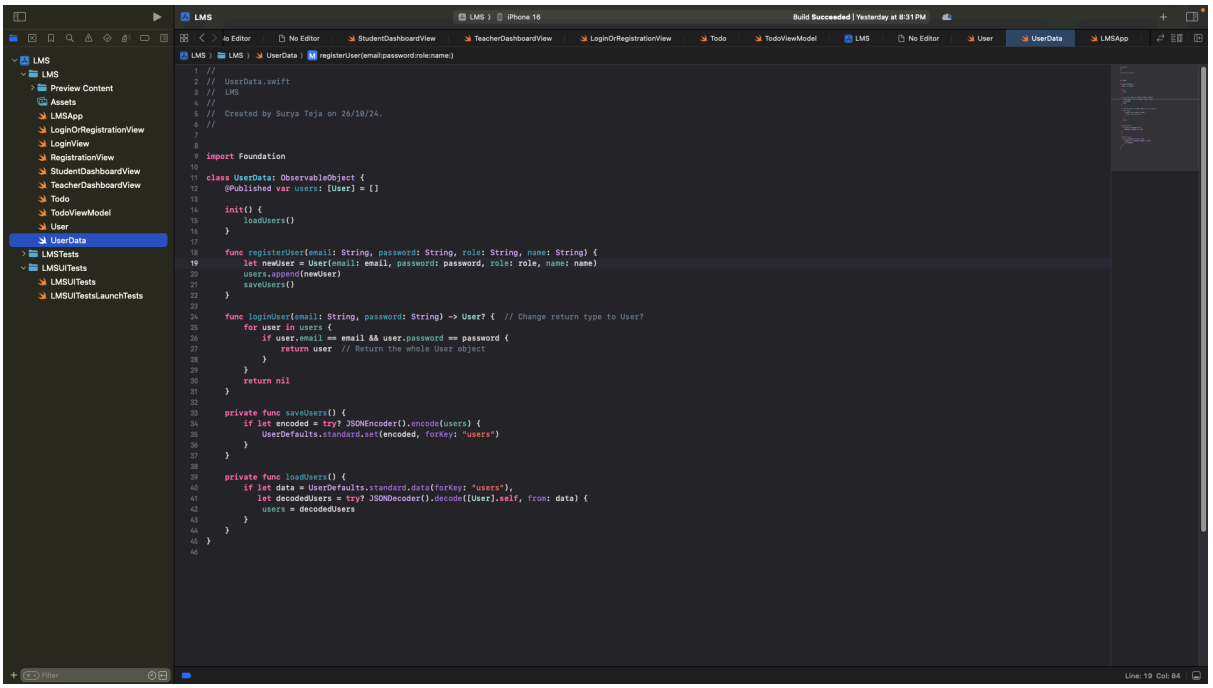
LMSApp:



User.swift:

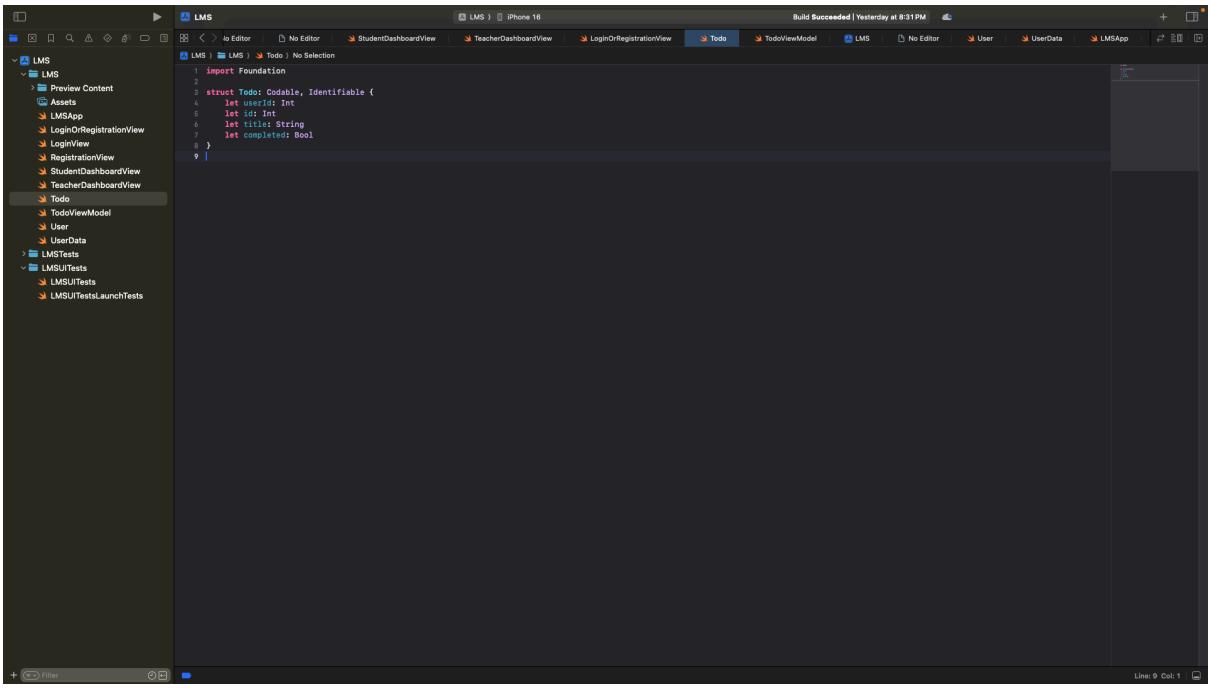


UserData.swift



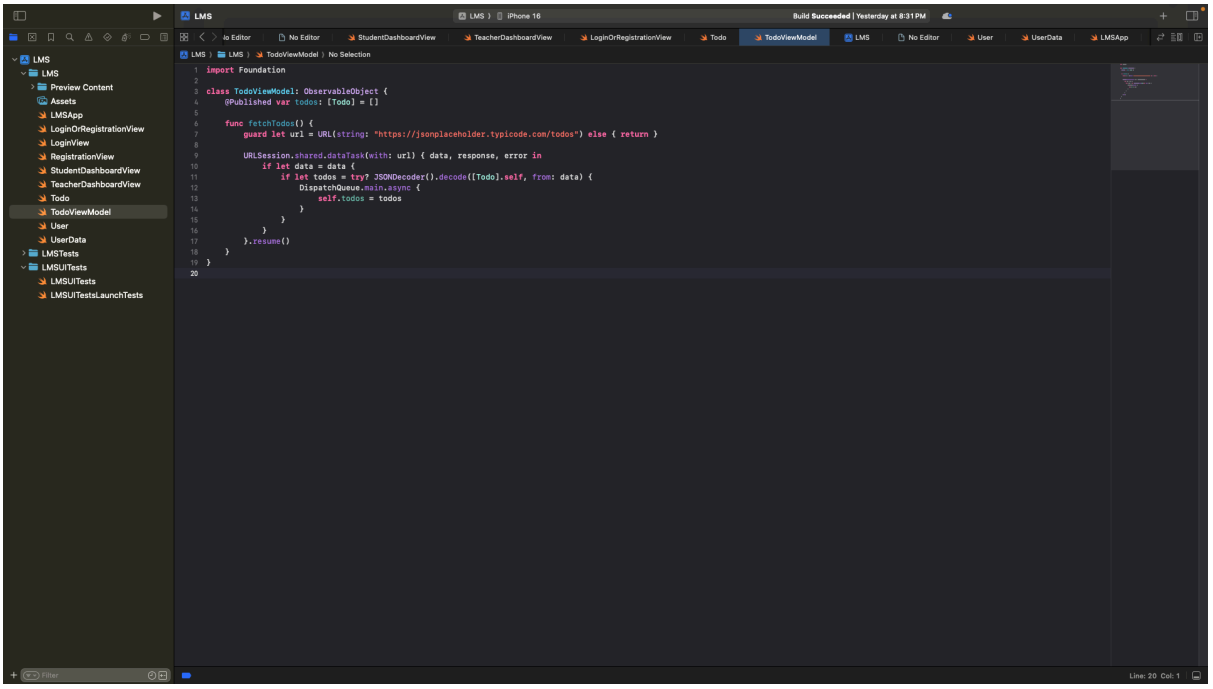
```swift
//
//  UserData.swift
//  LMS
//
//  Created by Surya Teja on 26/10/24.
//


import Foundation

class UserData: ObservableObject {
    @Published var users: [User] = []

    init() {
        loadUsers()
    }

    func registerUser(email: String, password: String, role: String, name: String) {
        let newUser = User(email: email, password: password, role: role, name: name)
        users.append(newUser)
        saveUsers()
    }

    func loginUser(email: String, password: String) -> User? {  // Change return type to User?
        for user in users {
            if user.email == email && user.password == password {
                return user  // Return the whole User object
            }
        }
        return nil
    }

    private func saveUsers() {
        if let encoded = try? JSONEncoder().encode(users) {
            UserDefaults.standard.set(encoded, forKey: "users")
        }
    }

    private func loadUsers() {
        if let data = UserDefaults.standard.data(forKey: "users"),
           let decodedUsers = try? JSONDecoder().decode([User].self, from: data) {
            users = decodedUsers
        }
    }
}

```

Todo.swift:



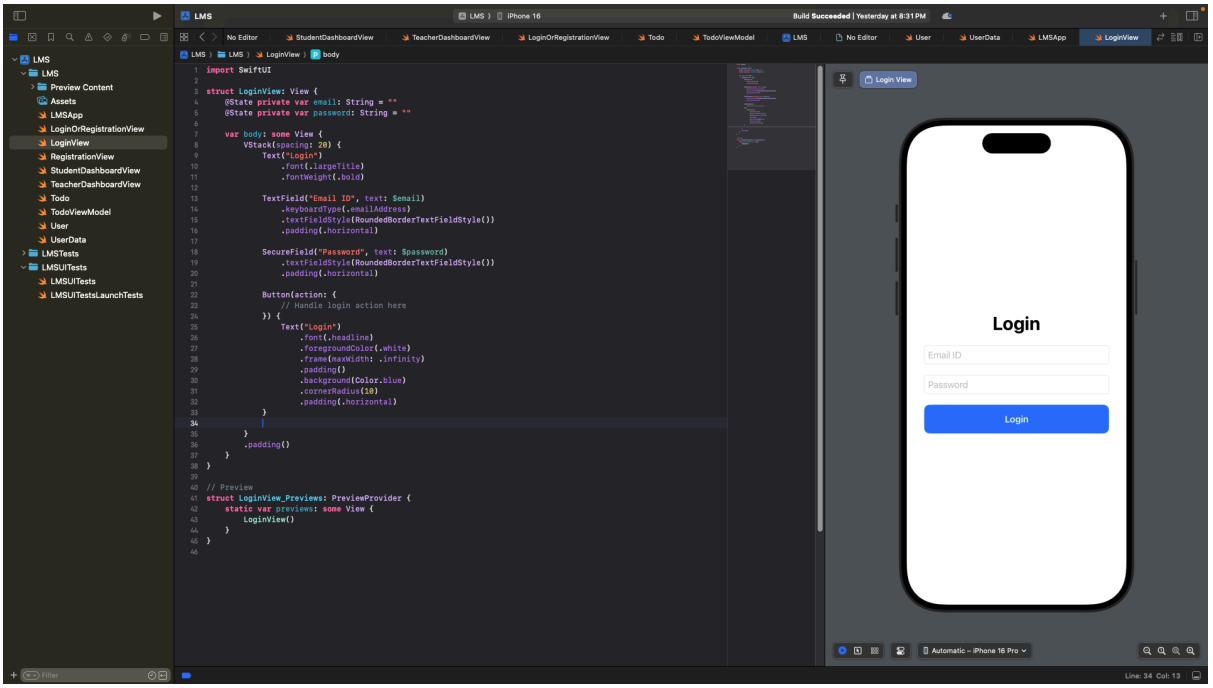
```swift
import Foundation

struct Todo: Codable, Identifiable {
    let userId: Int
    let id: Int
    let title: String
    let completed: Bool
}

```

TodoViewModel.swift:



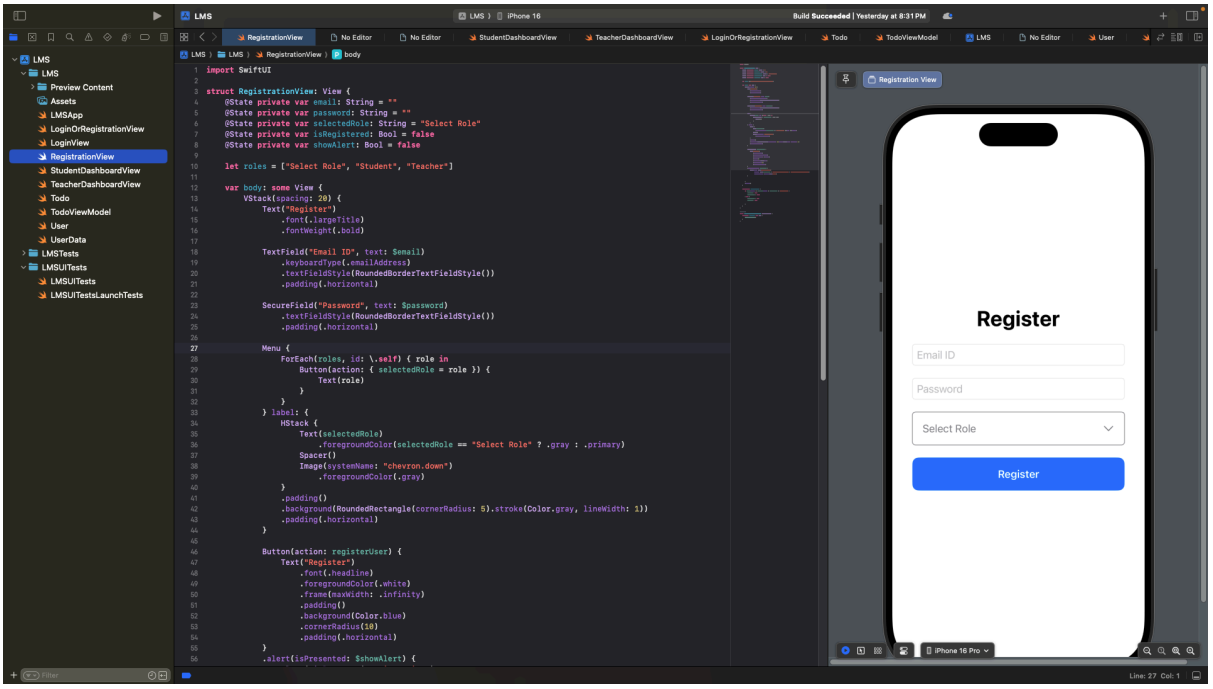
```swift
import Foundation

class TodoViewModel: ObservableObject {
    @Published var todos: [Todo] = []

    func fetchTodos() {
        guard let url = URL(string: "https://jsonplaceholder.typicode.com/todos") else { return }

        URLSession.shared.dataTask(with: url) { data, response, error in
            if let data = data {
                if let todos = try? JSONDecoder().decode([Todo].self, from: data) {
                    DispatchQueue.main.async {
                        self.todos = todos
                    }
                }
            }
        }.resume()
    }
}
```

LoginView.swift:



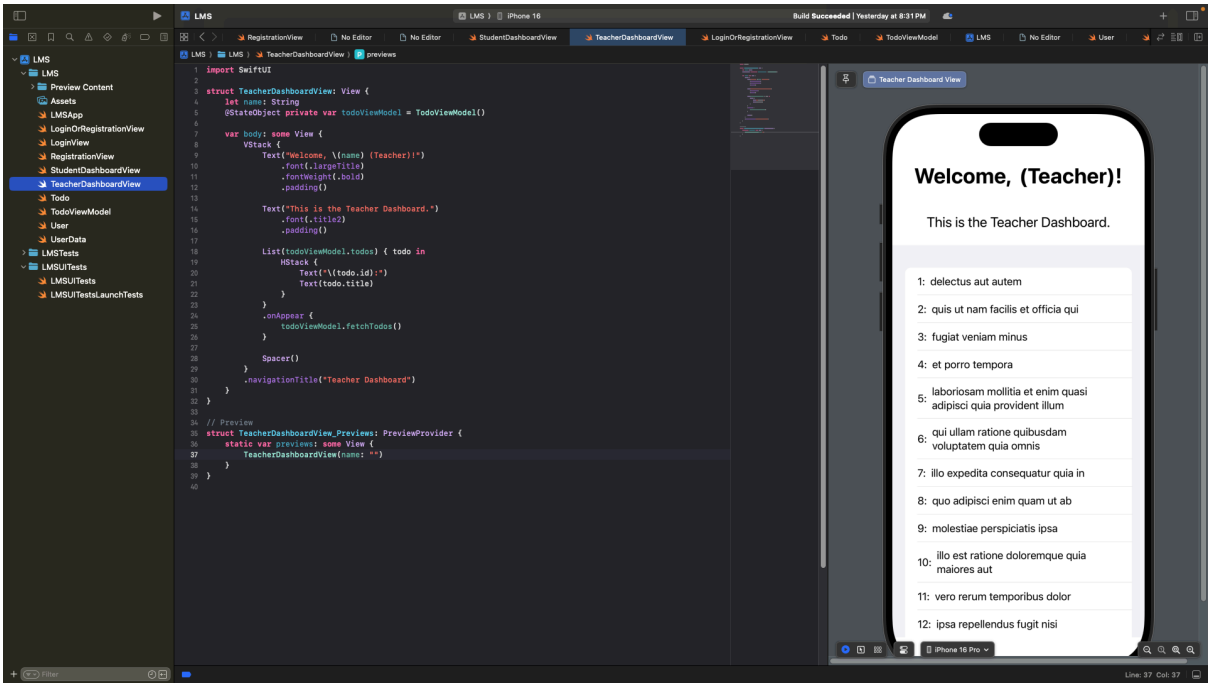
```swift
import SwiftUI

struct LoginView: View {
    @State private var email: String = ""
    @State private var password: String = ""

    var body: some View {
        VStack(spacing: 20) {
            Text("Login")
                .font(.largeTitle)
                .fontWeight(.bold)

            TextField("Email ID", text: $email)
                .keyboardType(.emailAddress)
                .textFieldStyle(RoundedBorderTextFieldStyle())
                .padding(.horizontal)

            SecureField("Password", text: $password)
                .textFieldStyle(RoundedBorderTextFieldStyle())
                .padding(.horizontal)

            Button(action: {
                // Handle login action here
            }) {
                Text("Login")
                    .font(.headline)
                    .foregroundColor(.white)
                    .frame(maxWidth: .infinity)
                    .padding()
                    .background(Color.blue)
                    .cornerRadius(10)
                    .padding(.horizontal)
            }

        }
        .padding()
    }
}

// Preview
struct LoginView_Previews: PreviewProvider {
    static var previews: some View {
        LoginView()
    }
}
```

RegistrationView.swift:

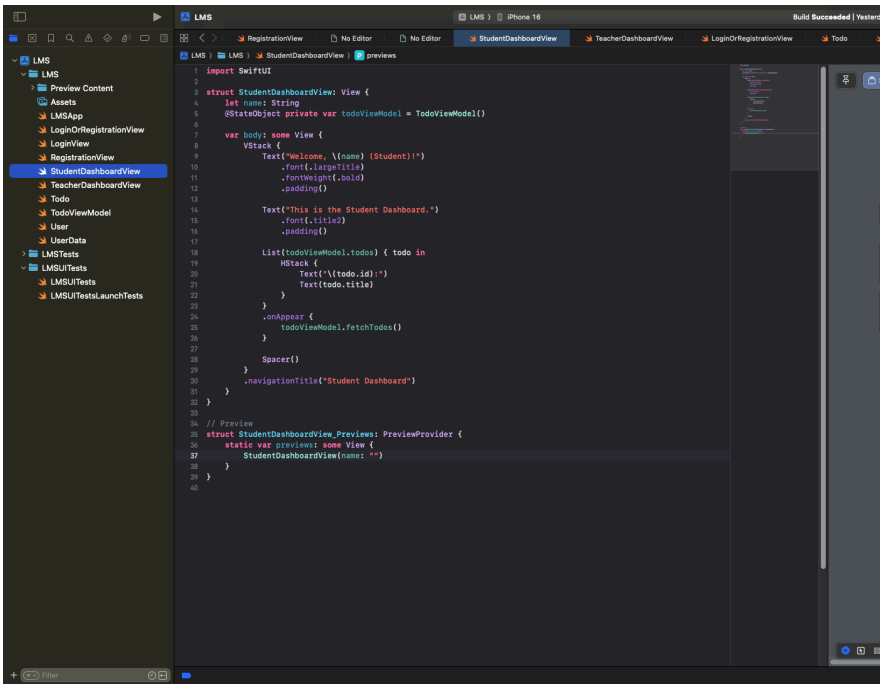


```swift
import SwiftUI

struct RegistrationView: View {
    @State private var email: String = ""
    @State private var password: String = ""
    @State private var selectedRole: String = "Select Role"
    @State private var isRegistered: Bool = false
    @State private var showAlert: Bool = false

    let roles = ["Select Role", "Student", "Teacher"]

    var body: some View {
        VStack(spacing: 20) {
            Text("Register")
                .font(.largeTitle)
                .fontWeight(.bold)

            TextField("Email ID", text: $email)
                .keyboardType(.emailAddress)
                .textFieldStyle(RoundedBorderTextFieldStyle())
                .padding(.horizontal)

            SecureField("Password", text: $password)
                .textFieldStyle(RoundedBorderTextFieldStyle())
                .padding(.horizontal)

            Menu {
                ForEach(roles, id: \.self) { role in
                    Button(action: { selectedRole = role }) {
                        Text(role)
                    }
                }
            } label: {
                HStack {
                    Text(selectedRole)
                        .foregroundColor(selectedRole == "Select Role" ? .gray : .primary)
                    Spacer()
                    Image(systemName: "chevron.down")
                        .foregroundColor(.gray)
                }
                .padding()
                .background(RoundedRectangle(cornerRadius: 5).stroke(Color.gray, lineWidth: 1))
                .padding(.horizontal)
            }

            Button(action: registerUser) {
                Text("Register")
                    .font(.headline)
                    .foregroundColor(.white)
                    .frame(maxWidth: .infinity)
                    .padding()
                    .background(Color.blue)
                    .cornerRadius(10)
                    .padding(.horizontal)
            }
            .alert(isPresented: $showAlert) {
                Alert(title: Text("Registration"),
                      message: Text(isRegistered ? "Registration Successful" : "Please fill in all fields"),
                      dismissButton: .default(Text("OK")))
            }


        }
        .padding()
    }

    private func registerUser() {
        if email.isEmpty || password.isEmpty || selectedRole == "Select Role" {
            showAlert = true
            isRegistered = false
        } else {
            isRegistered = true
            showAlert = true


        }
    }
}

// Preview
struct RegistrationView_Previews: PreviewProvider {
    static var previews: some View {
        RegistrationView()
    }
}
```

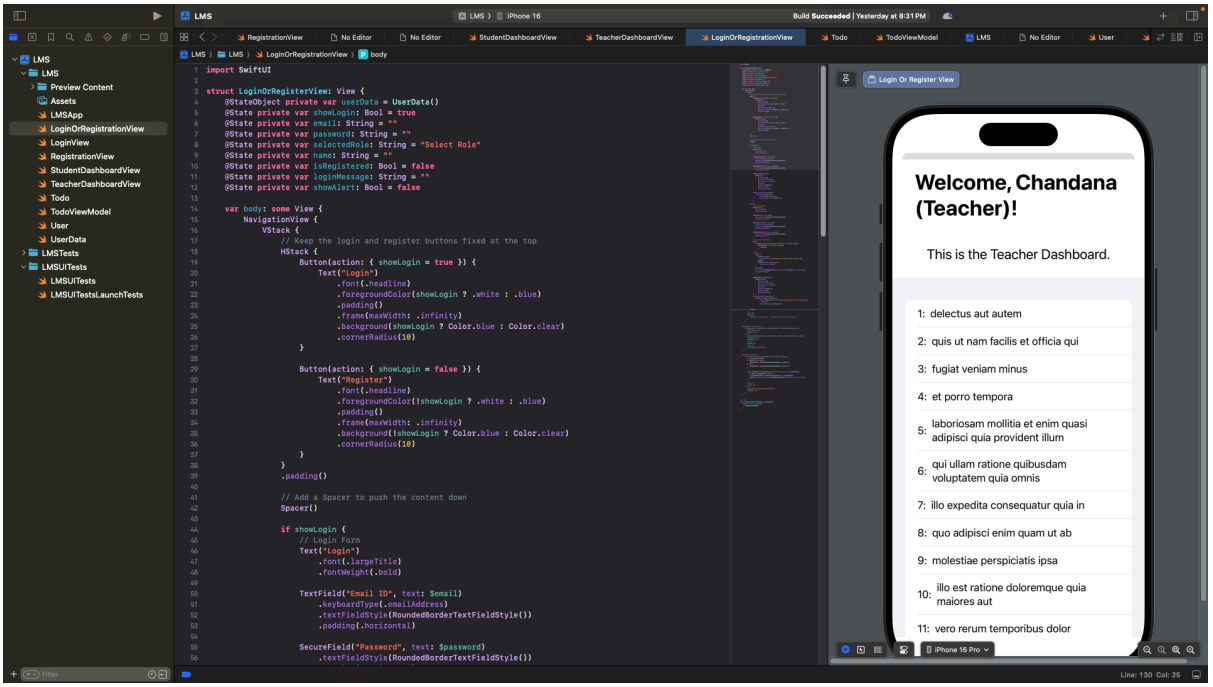
TeacherdashboardView.swift:

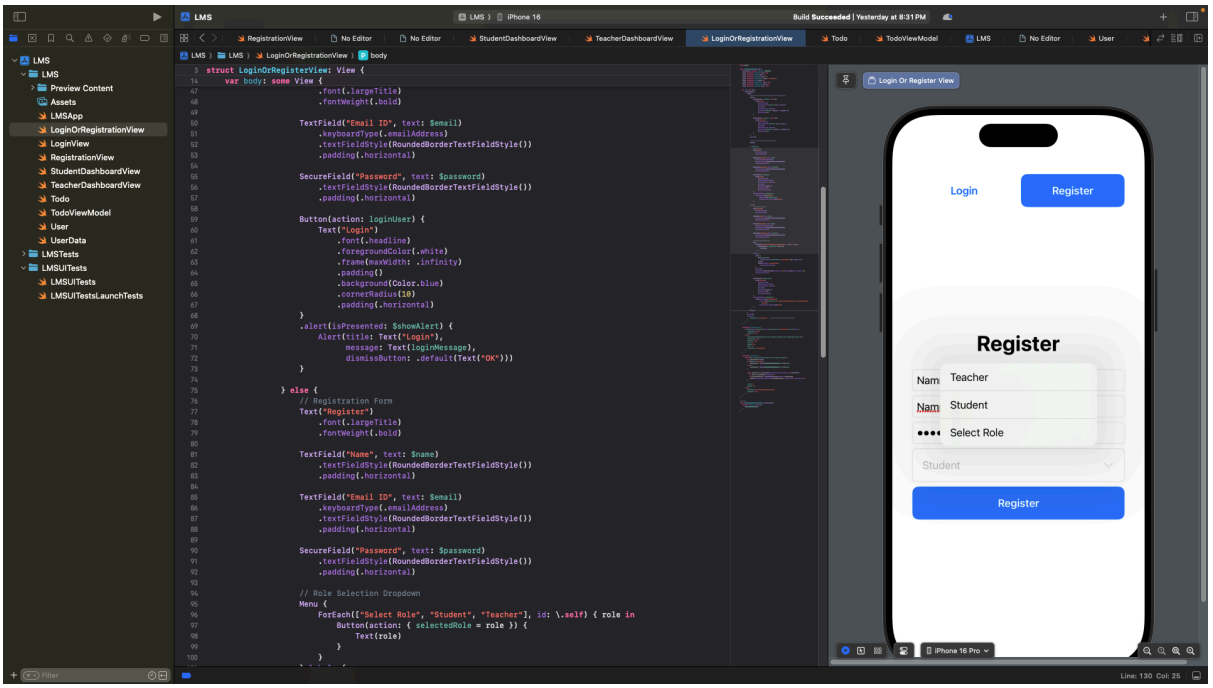


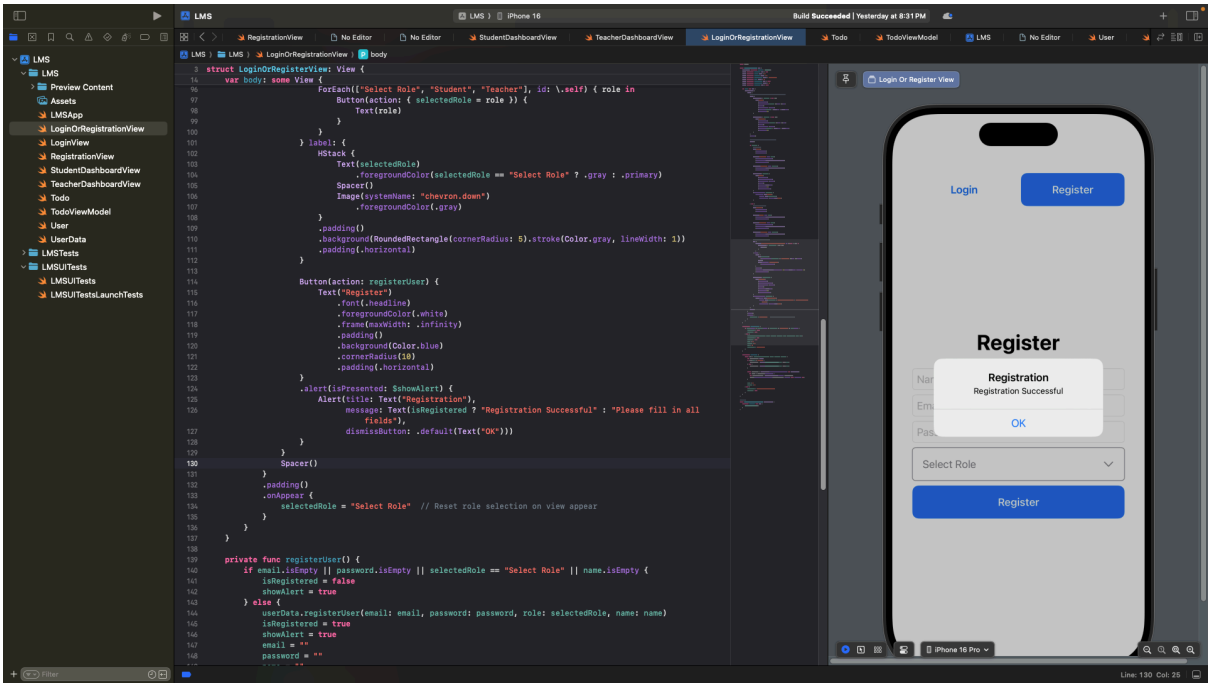
StudentDashboardView.swift:

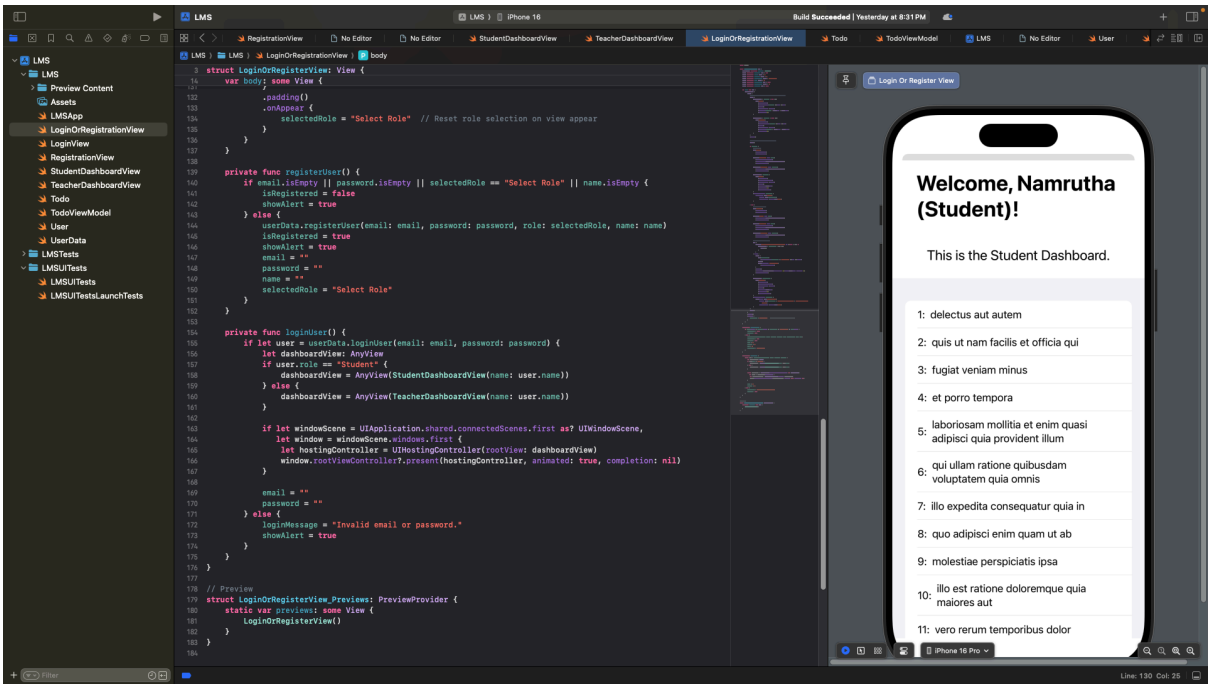


LoginOrRegistrationView.swift:


Welcome, Chandana (Teacher)!
This is the Teacher Dashboard.
1: delectus aut autem
2: quis ut nam facilis et officia qui
3: fugiat veniam minus
4: et porro tempora
5: laboriosam mollitia et enim quasi adipisci quia provident illum
6: qui ullam ratione quibusdam voluptatem quia omnis
7: illo expedita consequatur quia in
8: quo adipisci enim quam ut ab
9: molestiae perspiciatis ipsa
10: illo est ratione doloremque quia maiores aut
11: vero rerum temporibus dolor


Login
Register
Register
Teacher
Student
Select Role
Student
Register


LMS
Build Succeeded | Yesterday at 8:31 PM
Login
Register
Register
Registration
Registration Successful
OK
Select Role
Register


LMS
Build Succeeded | Yesterday at 8:31 PM
Welcome, Namrutha (Student)!
This is the Student Dashboard.
1: delectus aut autem
2: quis ut nam facilis et officia qui
3: fugiat veniam minus
4: et porro tempora
5: laboriosam mollitia et enim quasi adipisci quia provident illum
6: qui ullam ratione quibusdam voluptatem quia omnis
7: illo expedita consequatur quia in
8: quo adipisci enim quam ut ab
9: molestiae perspiciatis ipsa
10: illo est ratione doloremque quia maiores aut
11: vero rerum temporibus dolor