Swift UI iOS App Development

Module 1: Introduction to SwiftUI

- Overview of SwiftUI: History, Framework, and Benefits
- Understanding SwiftUI's Declarative Syntax
- SwiftUI vs UIKit: Key Differences
- Structure of a SwiftUI app

Lab : Getting Started with SwiftUI

- Create a new SwiftUI project
- Modify the default ContentView
- Build a simple "Hello, World!" app

Module 2: Views and Modifiers

- Introduction to SwiftUI Views
- Common Views: Text, Image, Button, Spacer, Stack (VStack, HStack, ZStack)
- Using Modifiers for styling Views

Lab : Creating a Simple Layout

- Design a card layout with Text, Image, and Button
- Apply modifiers: Padding, Background, ForegroundColor, etc.

Module 3: Stacks and Layouts

- Understanding Stack Views (HStack, VStack, ZStack)
- Building layouts with Spacer and Alignment
- Using GeometryReader for responsive layouts

Lab : Building a Custom Layout

- Create a product card using VStack and HStack
- Add dynamic alignment and spacing

Module 4: Handling User Input with Forms

- Introduction to Form

- Working with TextFields, Toggles, Pickers, and Sliders
- Binding data with @State properties

Lab : Building a Simple Form

- Build a user profile form using TextField, Picker, and Toggle
- Validate and display user input

Module 5: Navigation in SwiftUI

- NavigationView and NavigationLink
- Creating a Master-Detail Interface
- NavigationStack for multi-level navigation

Lab : Creating a Navigation-based App

- Build an app with multiple screens using NavigationLink
- Implement navigation between product categories and details

Module 6: Lists and Dynamic Data

- Introduction to List View
- Displaying static and dynamic data
- Working with Arrays and Identifiable Protocol

Lab : Dynamic List with Data

- Create a dynamic list of items (e.g., laptops, books) from an array
- Add functionality to delete, move, and reorder items in the list

Module 7: Data Flow with State and Binding

- Managing Data with @State and @Binding
- Introduction to Observables: ObservableObject and @Published
- Introduction to Data Persistence with UserDefaults

Lab : Data Binding and Persistence

- Implement a simple counter app using @State and @Binding
- Persist data across app Modules with UserDefaults

Module 8: Advanced Data Flow

- Using @EnvironmentObject for global data management
- Passing data across multiple views

Lab: Global Data Management

- Create an app with global settings using @EnvironmentObject
- Share data between multiple views

Module 9: Customizing Lists

- Customizing the appearance of List Rows
- Building custom list item views
- Adding navigation and actions in list items

Lab : Building a Custom List View

- Create a custom List view with navigation to a detailed view for each item
- Customize list row appearance with images, text, and buttons

Module 10: Animation and Transitions

- Introduction to Animation in SwiftUI
- Implicit and Explicit Animations

Lab : Adding Animations

- Create an app with animation on button tap

Module 11: Working with Gestures

- Introduction to Tap, Drag, Rotation, and Magnification Gestures
- Handling multiple gestures in a single view
- Gesture composition

Lab : Implementing Gestures

- Build an interactive app with draggable, scalable, and rotatable views
- Combine multiple gestures in a single view

Module 12: Working with Grids and Collections

- Introduction to LazyVGrid and LazyHGrid
- Working with Grid layouts

- Building a dynamic Grid using arrays

Lab : Dynamic Grid Layout

- Display a grid of items (e.g., fruit or products)
- Make the grid scrollable and responsive using LazyVGrid

Module 13: Styling and Themes

- Customizing appearance with Themes and Styling
- Using Environment values to change global app appearance

Lab: Theme-based Customization

- Create a light and dark theme for the app
- Apply consistent styling across all views

Module 14: App Architecture

- Introduction to MVVM (Model-View-ViewModel) Architecture in SwiftUI
- Organizing code using ViewModels
- Separating concerns between Views, Models, and ViewModels

Lab: Implementing MVVM Architecture

- Build a weather app using MVVM architecture
- Organize data fetching and display using ViewModel

Module 15: Data Persistence with Core Data

- Introduction to Core Data in SwiftUI
- Saving and fetching data using Core Data

Lab: Building a Core Data App

- Create a task manager app that stores tasks using Core Data

Module 16: Networking and REST API Integration

- Introduction to URLModule
- Making GET and POST requests
- Parsing JSON data using Codable
- Handling network delays and errors

- Using AsyncImage for displaying images from URLs

Lab: Fetch Data from REST API

- Build an app that fetches data from an API
- Display the fetched data in a list format
- Implement error handling for failed network requests
- Build an app to fetch and display a list of remote images