บริษัท ศิลา นักปราชญ์ จำกัด

We believe that technology has the power to transform the classroom.

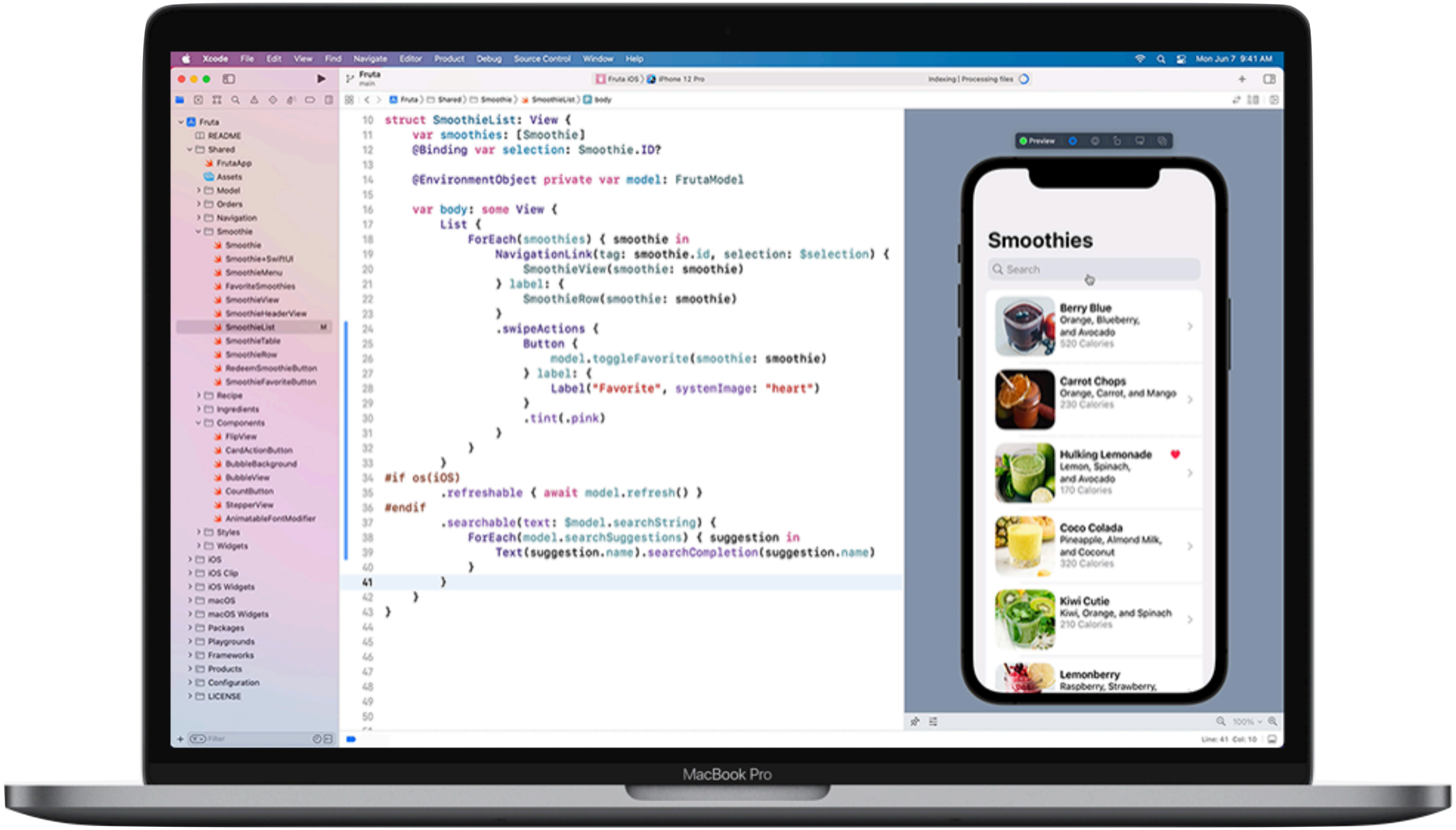
Hello, SwiftUI !

โดย ธิติ ธีระเธียร

Apple Certified Trainer (App Development with Swift)

Issue: January 2026


9:41
Clock
Camera
375/500CAL
19/30MIN
4/12HRS
TV
Stocks
Activity
Mail
Files
Photos
App Store
Weather
Health
Home
Chromatica
Lady Gaga
News
Photos
Music

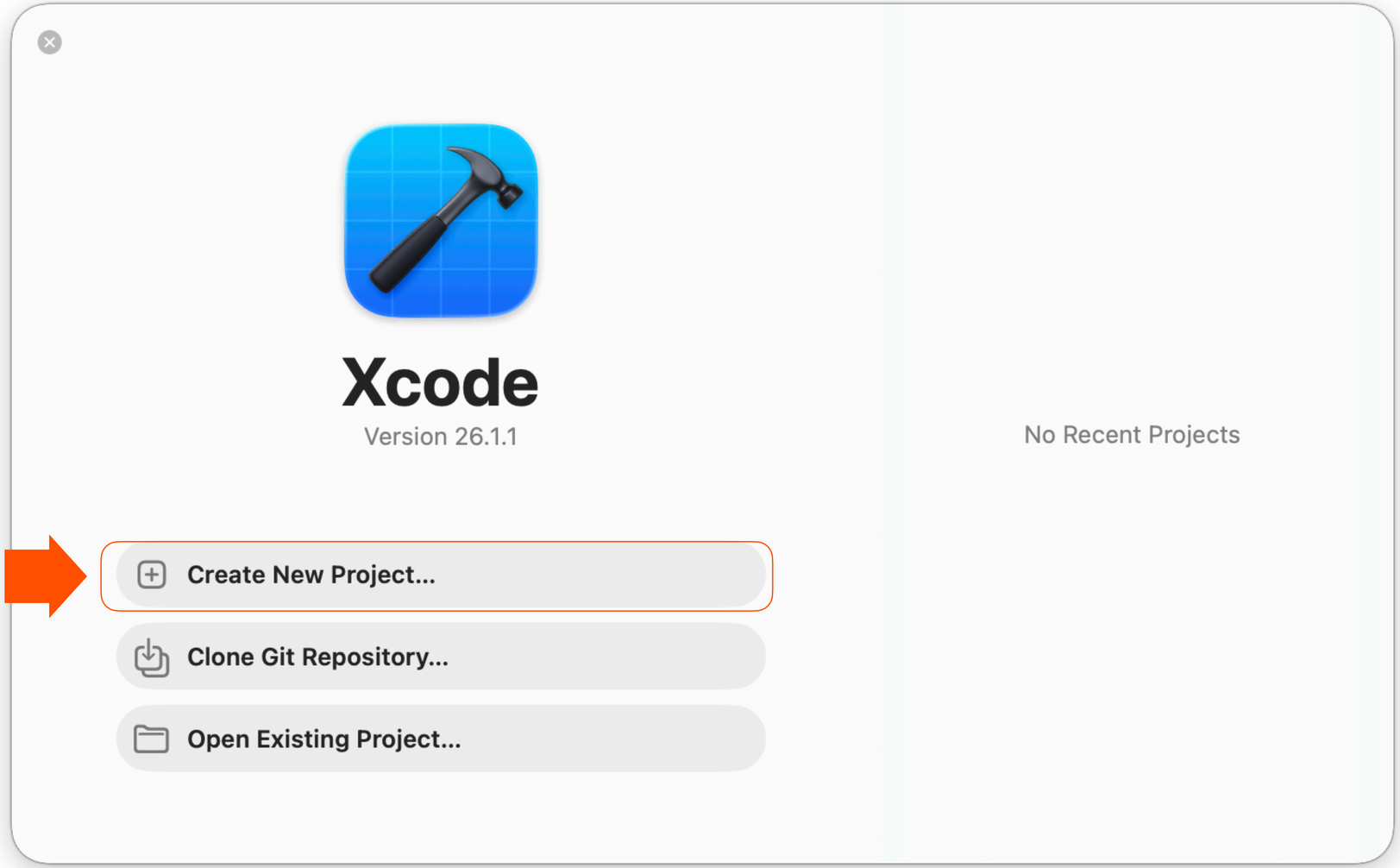

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Mon Jun 7 9:41 AM
Fruta
Fruta iOS iPhone 12 Pro
Indexing | Processing files
Fruta Shared Smoothie SmoothieList body
Fruta
README
Shared
FrutaApp
Assets
Model
Orders
Navigation
Smoothie
Smoothie
Smoothie+SwiftUI
SmoothieMenu
FavoriteSmoothies
SmoothieView
SmoothieHeaderView
SmoothieList
SmoothieTable
SmoothieRow
RedeemSmoothieButton
SmoothieFavoriteButton
Recipe
Ingredients
Components
FlipView
CardActionButton
BubbleBackground
BubbleView
CountButton
StepperView
AnimatableFontModifier
Styles
Widgets
iOS
iOS Clip
iOS Widgets
macOS
macOS Widgets
Packages
Playgrounds
Frameworks
Products
Configuration
LICENSE
10 struct SmoothieList: View {
11 var smoothies: [Smoothie]
12 @Binding var selection: Smoothie.ID?
13
14 @EnvironmentObject private var model: FrutaModel
15
16 var body: some View {
17 List {
18 ForEach(smoothies) { smoothie in
19 NavigationLink(tag: smoothie.id, selection: $selection) {
20 SmoothieView(smoothie: smoothie)
21 } label: {
22 SmoothieRow(smoothie: smoothie)
23 }
24 .swipeActions {
25 Button {
26 model.toggleFavorite(smoothie: smoothie)
27 } label: {
28 Label("Favorite", systemImage: "heart")
29 }
30 .tint(.pink)
31 }
32 }
33 }
34 #if os(iOS)
35 .refreshable { await model.refresh() }
36 #endif
37 .searchable(text: $model.searchString) {
38 ForEach(model.searchSuggestions) { suggestion in
39 Text(suggestion.name).searchCompletion(suggestion.name)
40 }
41 }
42 }
43 }
Preview
Smoothies
Search
Berry Blue
Orange, Blueberry, and Avocado
320 Calories
Carrot Chops
Orange, Carrot, and Mango
230 Calories
Hulking Lemonade
Lemon, Spinach, and Avocado
170 Calories
Coco Colada
Pineapple, Almond Milk, and Coconut
320 Calories
Kiwi Cutie
Kiwi, Orange, and Spinach
210 Calories
Lemonberry
Raspberry, Strawberry,
100%
Line: 41 Col: 10
MacBook Pro

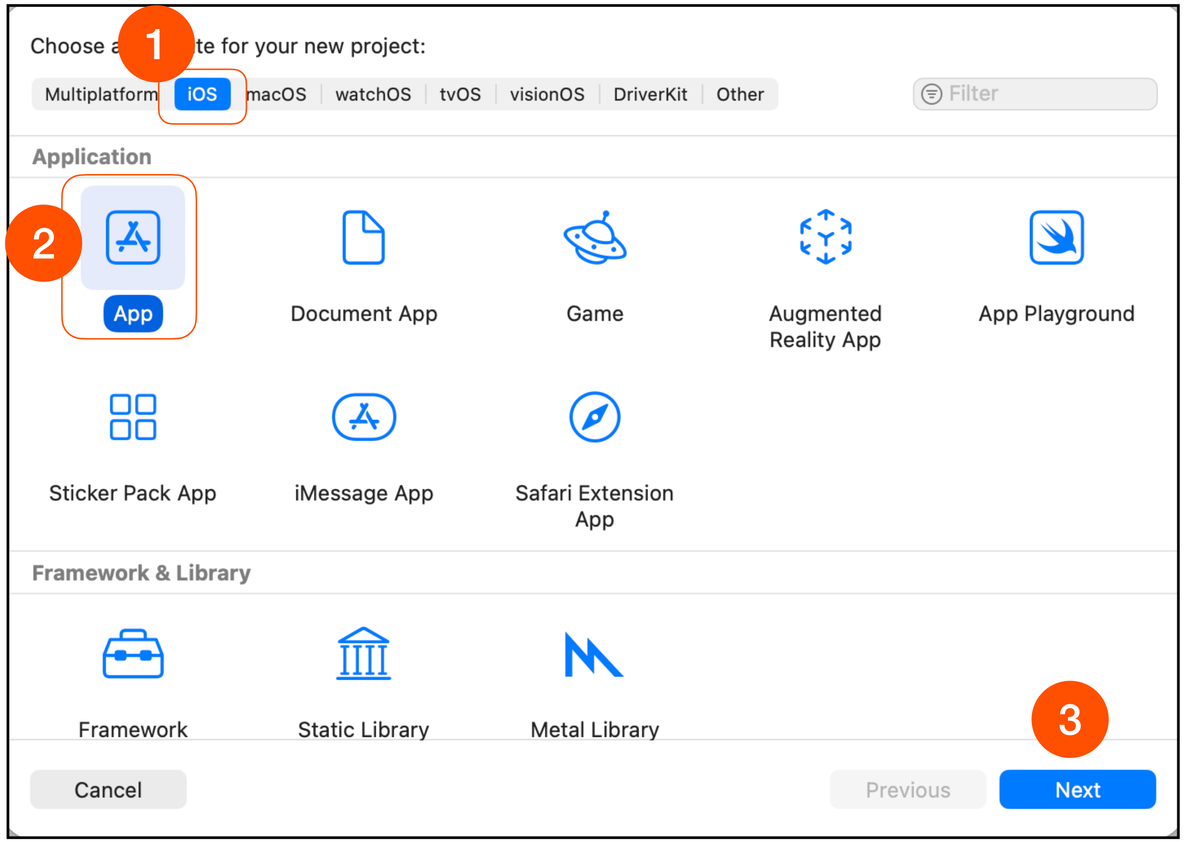
SwiftUI

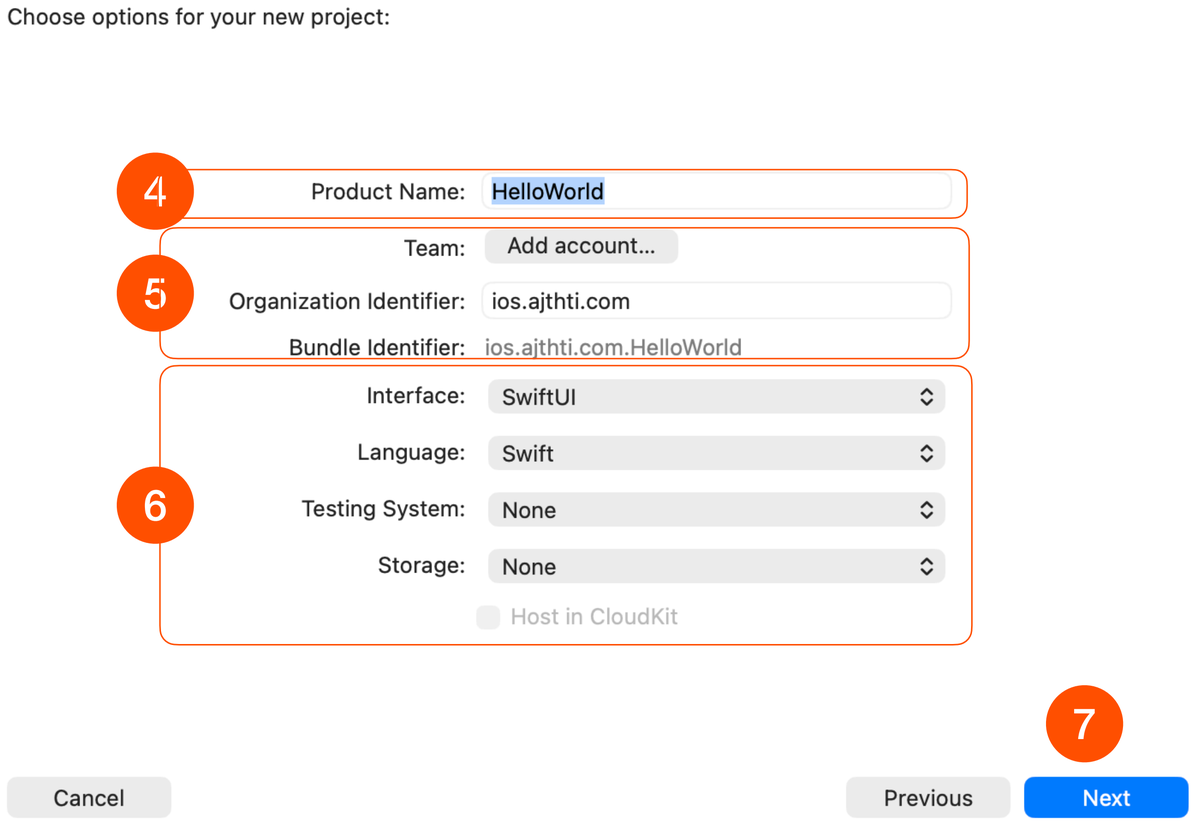
การเขียนคำสั่งแบบ Declarative เพื่อสร้างส่วนติดต่อกับผู้ใช้งาน (User Interface) สำหรับการพัฒนาแอปบนแพรตฟอร์มของ Apple ถูกประกาศใช้ครั้งแรก ในปี ค.ศ.2019 ซึ่งช่วยให้นักพัฒนาสามารถสร้างแอปได้อย่างรวดเร็วและใช้โค้ดน้อยที่สุด

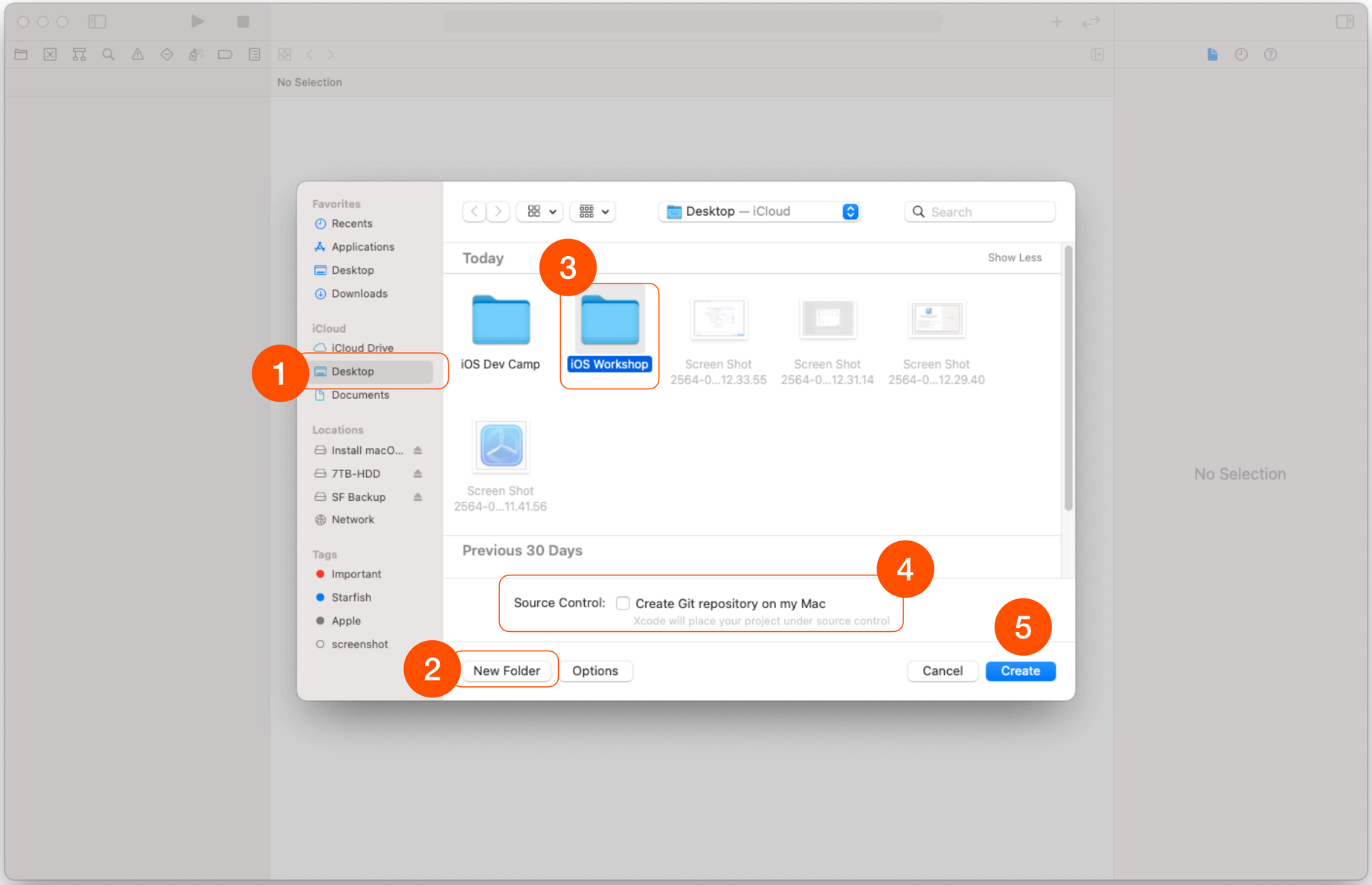
“No more Interface Builder and Auto Layout”

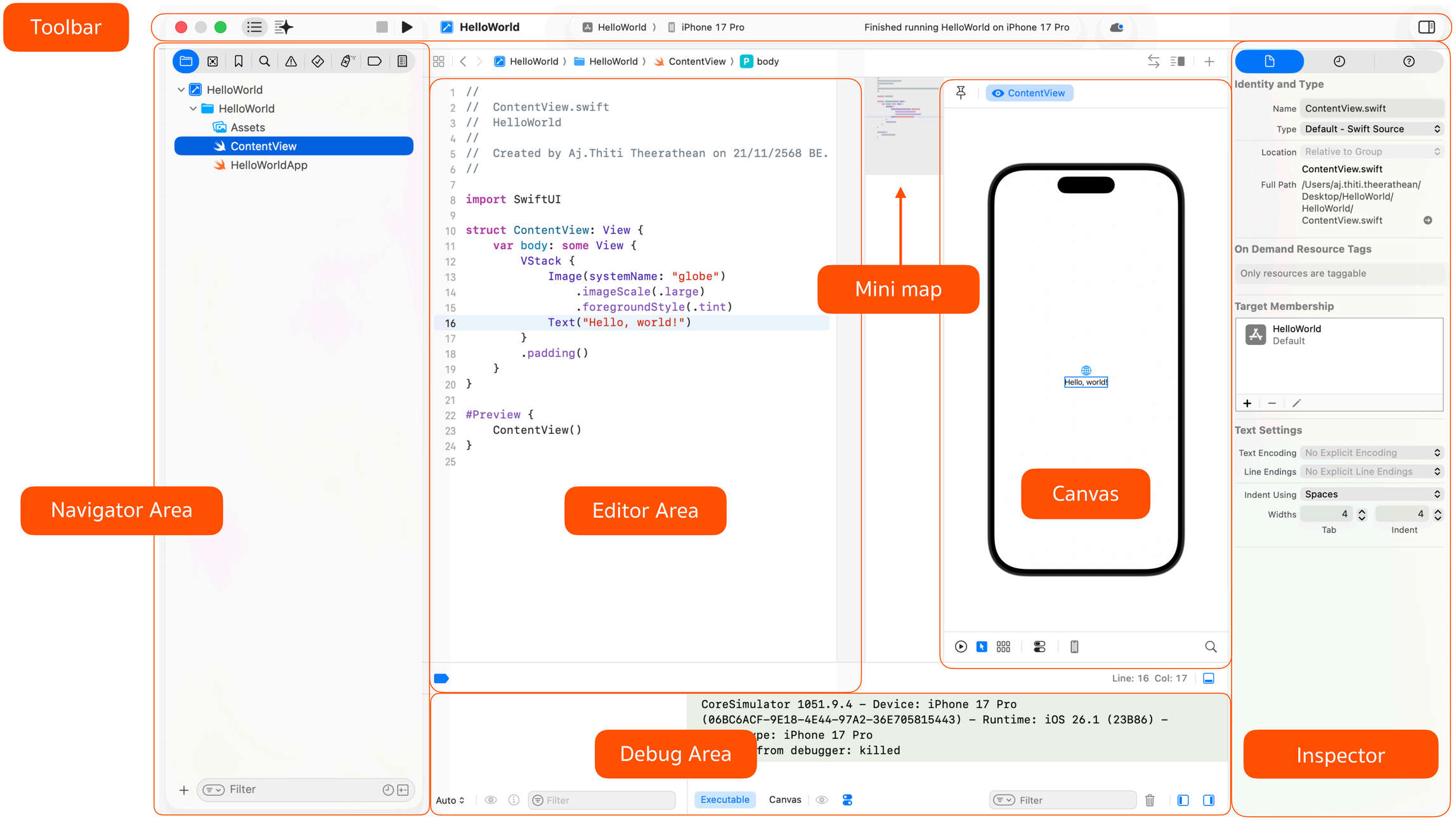
Create your Hello World App for iOS Device with SwiftUI


Xcode
Version 26.1.1
No Recent Projects
Create New Project...
Clone Git Repository...
Open Existing Project...


Choose a te for your new project:
1
Multiplatform
iOS
macOS
watchOS
tvOS
visionOS
DriverKit
Other
Filter
Application
2
App
Document App
Game
Augmented Reality App
App Playground
Sticker Pack App
iMessage App
Safari Extension App
Framework & Library
Framework
Static Library
Metal Library
3
Cancel
Previous
Next

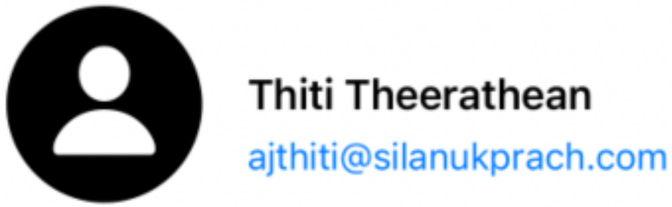

Choose options for your new project:
4
Product Name:
HelloWorld
5
Team:
Add account...
Organization Identifier:
ios.ajthti.com
Bundle Identifier:
ios.ajthti.com.HelloWorld
6
Interface:
SwiftUI
Language:
Swift
Testing System:
None
Storage:
None
Host in CloudKit
7
Cancel
Previous
Next


No Selection
Favorites
Recents
Applications
Desktop
Downloads
iCloud
iCloud Drive
Desktop
Documents
Locations
Install macO...
7TB-HDD
SF Backup
Network
Tags
Important
Starfish
Apple
screenshot
Desktop — iCloud
Search
Today
Show Less
iOS Dev Camp
iOS Workshop
Screen Shot 2564-0...12.33.55
Screen Shot 2564-0...12.31.14
Screen Shot 2564-0...12.29.40
Screen Shot 2564-0...11.41.56
Previous 30 Days
Source Control:
Create Git repository on my Mac
Xcode will place your project under source control
New Folder
Options
Cancel
Create
1
2
3
4
5
No Selection

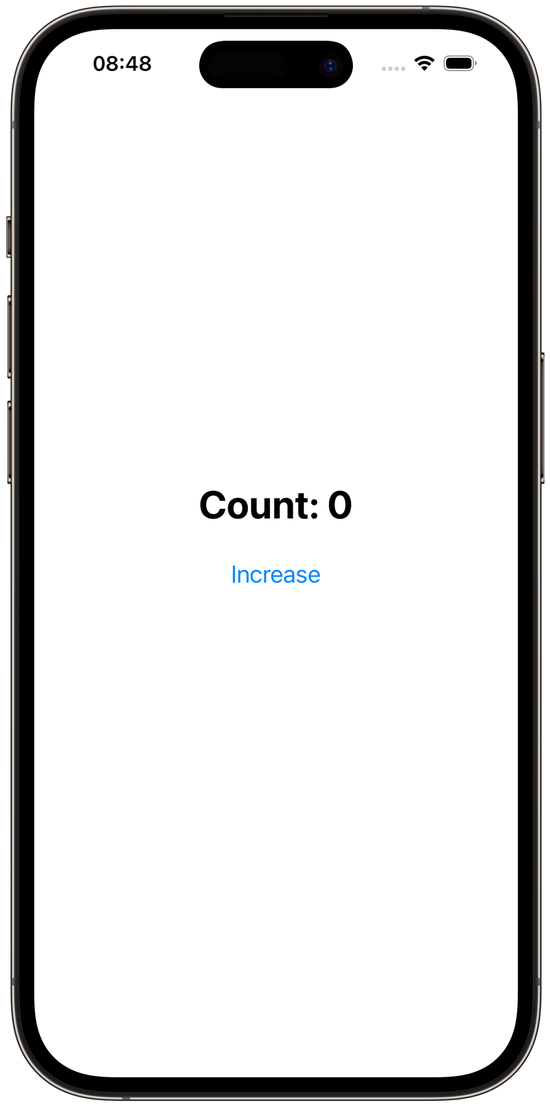

Toolbar
HelloWorld
HelloWorld ⟩ iPhone 17 Pro
Finished running HelloWorld on iPhone 17 Pro
HelloWorld
HelloWorld
Assets
ContentView
HelloWorldApp
HelloWorld ⟩ HelloWorld ⟩ ContentView ⟩ body
1 //
2 // ContentView.swift
3 // HelloWorld
4 //
5 // Created by Aj.Thiti Theerathean on 21/11/2568 BE.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11 var body: some View {
12 VStack {
13 Image(systemName: "globe")
14 .imageScale(.large)
15 .foregroundStyle(.tint)
16 Text("Hello, world!")
17 }
18 .padding()
19 }
20 }
21
22 #Preview {
23 ContentView()
24 }
25
Mini map
ContentView
Hello, world!
Canvas
Identity and Type
Name ContentView.swift
Type Default - Swift Source
Location Relative to Group
ContentView.swift
Full Path /Users/aj.thiti.theerathean/Desktop/HelloWorld/HelloWorld/ContentView.swift
On Demand Resource Tags
Only resources are taggable
Target Membership
HelloWorld
Default
Text Settings
Text Encoding No Explicit Encoding
Line Endings No Explicit Line Endings
Indent Using Spaces
Widths 4 Tab 4 Indent
Navigator Area
Editor Area
Line: 16 Col: 17
CoreSimulator 1051.9.4 - Device: iPhone 17 Pro
(06BC6ACF-9E18-4E44-97A2-36E705815443) - Runtime: iOS 26.1 (23B86) -
pe: iPhone 17 Pro
from debugger: killed
Debug Area
Inspector
Auto
Filter
Executable
Canvas

การสร้าง UI แบบ Imperative vs Declarative

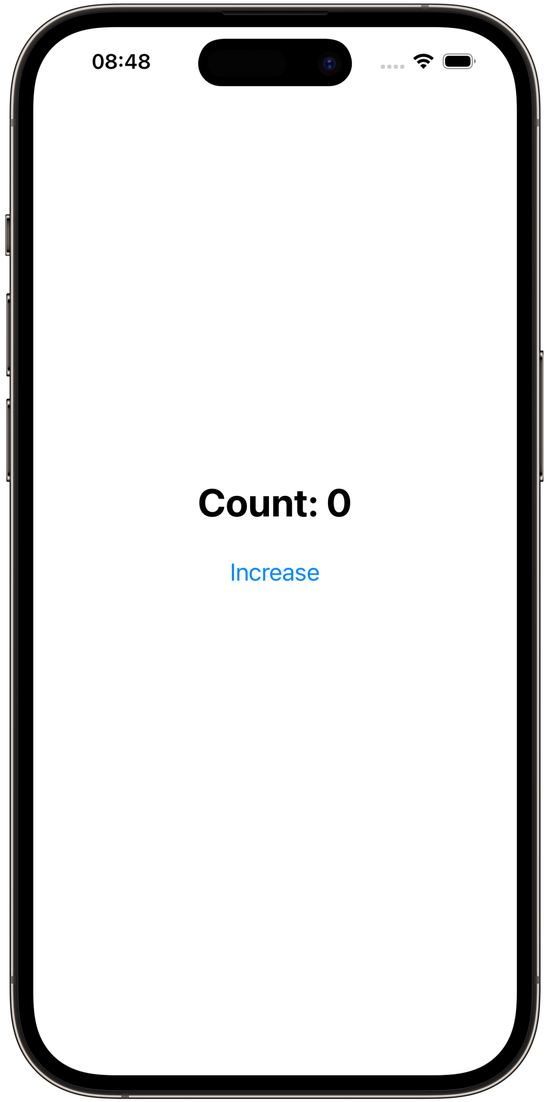
Imperative Programming




```swift
let cardView = UIView()
cardView.frame = CGRect(x: 50, y: 200, width: 300 , height: 100)
cardView.backgroundColor = .white
cardView.layer.borderColor = UIColor.white.cgColor
cardView.layer.borderWidth = 1
self.view.addSubview(cardView)

let profileImageView = UIImageView()
profileImageView.frame = CGRect(x: 5, y: 10, width: 80, height: 80)
profileImageView.image = UIImage(systemName:"person.circle.fill")
profileImageView.layer.borderWidth = 1
profileImageView.layer.masksToBounds = false
profileImageView.layer.borderColor = UIColor.white.cgColor
profileImageView.layer.cornerRadius = profileImageView.frame.height / 2
profileImageView.clipsToBounds = true
cardView.addSubview(profileImageView)

let nameLabel = UILabel()
nameLabel.frame = CGRect(x: 90, y: 30, width: 200, height: 20)
nameLabel.text = "Thiti Theerathean"
cardView.addSubview(nameLabel)

let emailLabel = UILabel()
emailLabel.frame = CGRect(x: 90, y: 50, width: 200, height: 20)
emailLabel.font = UIFont.systemFont(ofSize: 12)
emailLabel.text = "ajthiti@silanukprach.com"
cardView.addSubview(emailLabel)
```

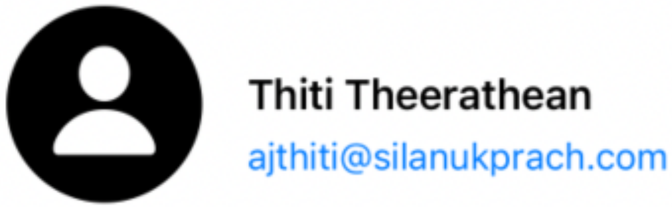



```swift
class ViewController: UIViewController {

        private var count = 0

        // UILabel แสดงตัวเลข
        private let countLabel: UILabel = {
            let label = UILabel()
            label.text = "Count: 0"
            label.font = UIFont.systemFont(ofSize: 32, weight: .bold)
            label.translatesAutoresizingMaskIntoConstraints = false
            return label
        }()

        // ปุ่ม Increase
        private let increaseButton: UIButton = {
            let button = UIButton(type: .system)
            button.setTitle("Increase", for: .normal)
            button.titleLabel?.font = UIFont.systemFont(ofSize: 20)
            button.translatesAutoresizingMaskIntoConstraints = false
            return button
        }()

        override func viewDidLoad() {
            super.viewDidLoad()

            view.backgroundColor = .systemBackground

            //เพิ่ม subviews
            view.addSubview(countLabel)
            view.addSubview(increaseButton)

            //จัดการ Auto Layout
            NSLayoutConstraint.activate([
                countLabel.centerXAnchor.constraint(equalTo: view.centerXAnchor),
                countLabel.centerYAnchor.constraint(equalTo: view.centerYAnchor, constant: -40),

                increaseButton.centerXAnchor.constraint(equalTo: view.centerXAnchor),
                increaseButton.topAnchor.constraint(equalTo: countLabel.bottomAnchor, constant: 20)
            ])

            //กำหนด event ให้ปุ่ม
            increaseButton.addTarget(self, action: #selector(increaseTapped), for: .touchUpInside)
        }

        @objc private func increaseTapped() {
            //ปรับค่าตัวแปร state
            count += 1

            //อัปเดต UI ด้วยตนเอง (Imperative)
            countLabel.text = "Count: \(count)"
        }

}
```

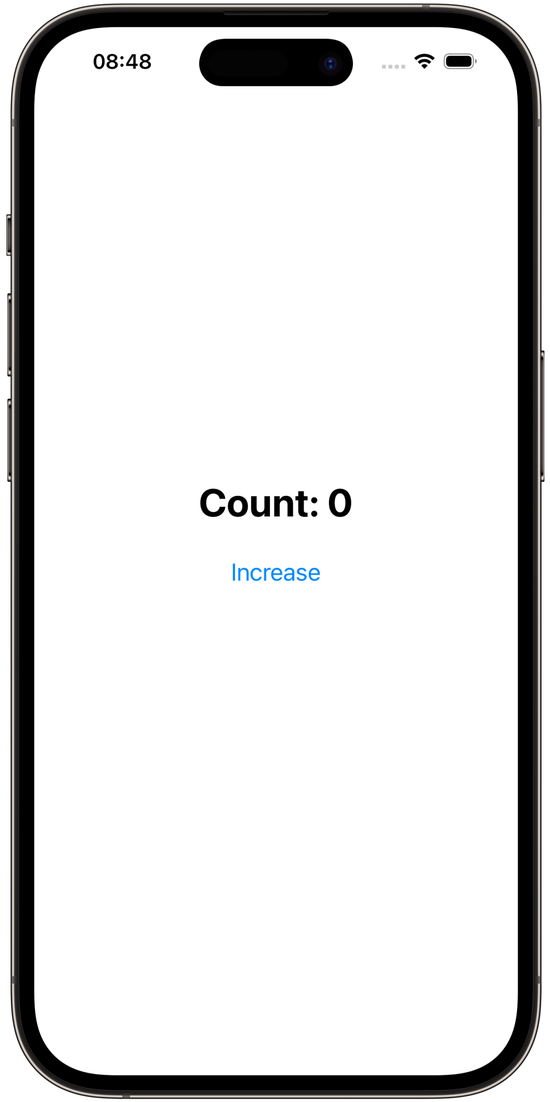



```swift
import UIKit

class ViewController: UIViewController {

    private var count = 0

    @IBOutlet weak var countLabel: UILabel!

    @IBAction func increaseTapped(_ sender: Any) {
        count += 1
        updateUI()
    }


    override func viewDidLoad() {
        super.viewDidLoad()
        view.backgroundColor = .systemBackground
        updateUI()
    }

    // MARK: - Helper Methods
    private func updateUI() {
        countLabel.text = "Count: \(count)"
    }

}
```

Declarative Programming




```
HStack {
    Image(systemName: "person.circle.fill")
         .resizable()
         .frame(width: 80.0 , height: 80.0)
         .padding(EdgeInsets(top: 5, leading: 5, bottom: 5, trailing: 5))

     VStack(alignment: .leading) {
         Text("Thiti Theerathean")
             .font(.headline)
         Text("ajthiti@silanukprach.com")
             .font(.subheadline)

     } .padding(EdgeInsets(top: 20, leading: 5, bottom: 5, trailing: 15))

}
```




```swift
import SwiftUI

struct ContentView: View {
    @State private var count = 0

    var body: some View {
        VStack {
            Text("Count: \(count)")
                .font(.largeTitle)

            Button("Increase") {
                count += 1
            }
        }
        .padding()
    }
}
```

แนวคิด Iterative UI Design กับการใช้ SwiftUI

Welcome Screen

การนำแนวคิด Iterative UI Design มาประยุกต์ใช้ร่วมกับ SwiftUI

โรงเรียนของคุณกำลังจัดทำโครงการ “พัฒนาแอปพลิเคชันของโรงเรียนด้วย SwiftUI” เพื่อให้นักเรียนเข้าถึงข่าวสาร ตารางกิจกรรม และระบบสนับสนุนการเรียนรู้ได้ง่ายขึ้น แอปใหม่นี้มีชื่อว่า **MyJourney**

ผู้ใช้แอป คือ นักเรียนระดับมัธยมศึกษาตอนปลาย ครู และบุคลากร ซึ่งมีระดับทักษะด้านเทคโนโลยีที่หลากหลาย และใช้อุปกรณ์ที่มีขนาดหน้าจอแตกต่างกัน (เช่น iPhone SE – iPhone 15 Pro)

เป้าหมายของ Welcome Screen

- การแสดงข้อความต้อนรับ ไม่ซับซ้อนเกินไป เพราะเป็นหน้าจอแรกของแอป
- มีปุ่มหลัก “Get Started” ที่เห็นง่าย ชัดเจน และกดสะดวก
- สามารถใช้งานได้ทั้งใน Light และ Dark Mode
- รองรับผู้ใช้ที่มีความบกพร่องทางสายตา

เพื่อสร้างความประทับใจแรกที่ดี แอปต้องมี **Welcome Screen** ที่ดึงดูด สื่อสารชัดเจน และใช้งานง่ายตั้งแต่เปิดแอปครั้งแรก

Iteration 1 — สร้าง “โครงสร้างพื้นฐาน” ของหน้าจอ (Core Structure)

```
struct ContentView: View {
    var body: some View {
        VStack() {
            Text("Welcome")
            Text("Start your journey with our app.")
        }
        .padding()
    }
}

#Preview {
    ContentView()
}
```

Iteration 2 — ปรับปรุงด้านการมองเห็น (Visual Enhancement)

```
VStack(spacing: 20) {
    Text("Welcome")
        .font(.largeTitle)
        .bold()

    Text("Start your journey with our app.")
        .font(.subheadline)
        .foregroundStyle(.secondary)
        .multilineTextAlignment(.center)
        .padding(.horizontal, 32)

}
.padding()
```

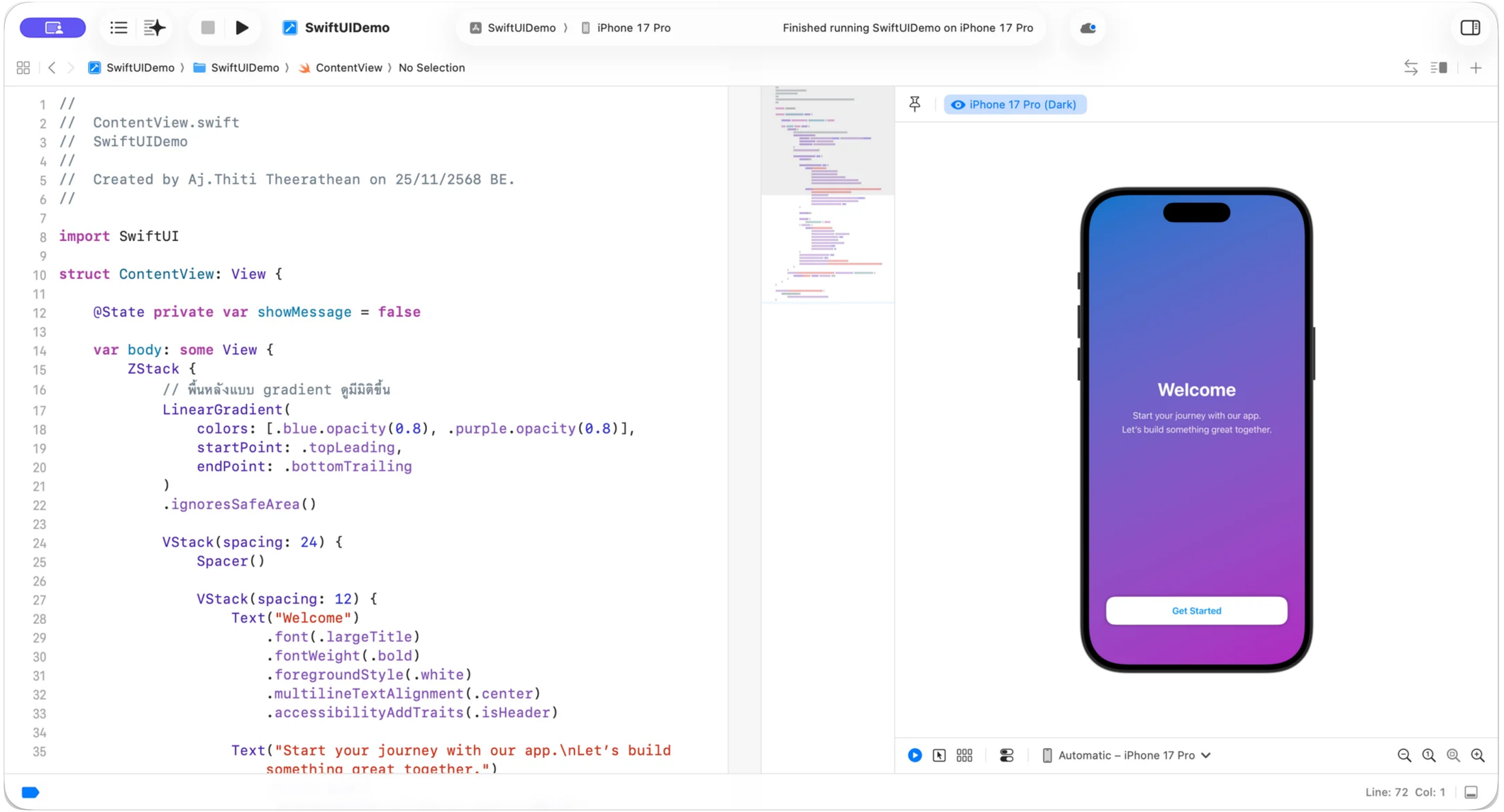
Iteration 3 — การเพิ่มการตอบสนองกับผู้ใช้งาน (Interactivity) ด้วยปุ่มกด

```swift
struct ContentView: View {

    @State private var showMessage = false

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

            Text("Start your journey with our app.")
                .font(.subheadline)
                .foregroundStyle(.secondary)
                .multilineTextAlignment(.center)
                .padding(.horizontal, 32)

            Button {
                showMessage = true      // เปลี่ยน state เมื่อกดปุ่ม
            } label: {
                Text("Get Started")
                    .padding(.vertical, 12)
                    .padding(.horizontal, 40)
                    .background(.blue)
                    .foregroundStyle(.white)
                    .cornerRadius(22)
            }
            .buttonStyle(.plain)
        }
        .padding()
        .alert("You tapped Get Started", isPresented: $showMessage) {
            Button("OK", role: .cancel) { }
        }
    }
}
```

Iteration 4 — ทดสอบการใช้งาน (Usability Testing)

```swift
#Preview("iPhone 17 Pro (Dark)") {
    ContentView()
        .preferredColorScheme(.dark)
}
```

Iteration 5 — การปรับแต่งเพื่อความสมบูรณ์ (Refinement)



คำถามเพื่อการสะท้อนผลการเรียนรู้จากการทำกิจกรรม

1. นักเรียนพบความท้าทายหรือข้อจำกัดใดบ้าง ในกิจกรรมการสร้าง UI ด้วย UIKit แบบ Programmatic ?
2. การใช้ Interface Builder (IB) ร่วมกับ @IBOutlet และ @IBAction ช่วยลดภาระการเขียนโค้ดได้อย่างไร และ ยังมีข้อจำกัดใดอยู่บ้าง เมื่อเทียบกับการใช้ SwiftUI ?
3. การพัฒนาส่วนติดต่อกับผู้ใช้ (UI) ในรูปแบบ Imperative (UIKit) และแบบ Declarative (SwiftUI) มีความแตกต่างกันอย่างไร ?
4. การออกแบบ UI แบบ Iterative แตกต่างจากการออกแบบครั้งเดียวจบ (One-shot Design) อย่างไร และเหตุใดแนวคิดแบบ Iterative จึงให้ผลลัพธ์ต่อการสร้างประสบการณ์ของผู้ใช้ (UX) ที่ดีกว่าในระยะยาว ?
5. การใช้ Live Preview ใน SwiftUI ช่วยให้นักพัฒนาสามารถออกแบบ UI ด้วยแนวคิด Iterative UI Design ได้อย่างไร ?
6. นักเรียนคิดว่า แนวคิด Iterative UI Design มีความเกี่ยวข้องอย่างไร กับแนวคิดเชิงวิศวกรรมซอฟต์แวร์อื่นๆ เช่น Agile Development, Design Thinking หรือ Prototyping ?