



บริษัท **ศิลา** นักปราชญ์ จำกัด

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

# Design an App Interface

โดย ธิติ ธีระเรียม

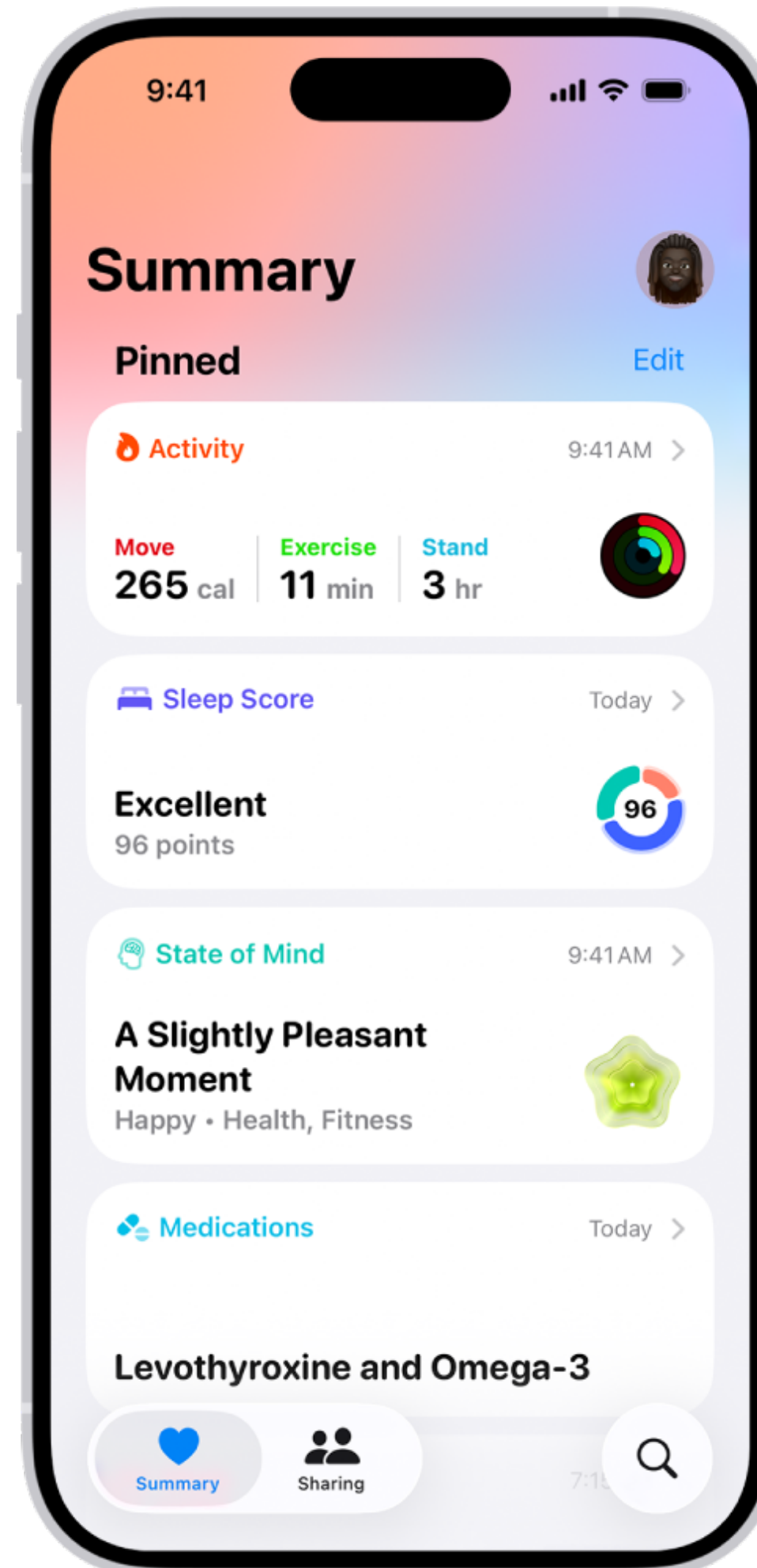
Apple Certified Trainer (App Development with Swift)

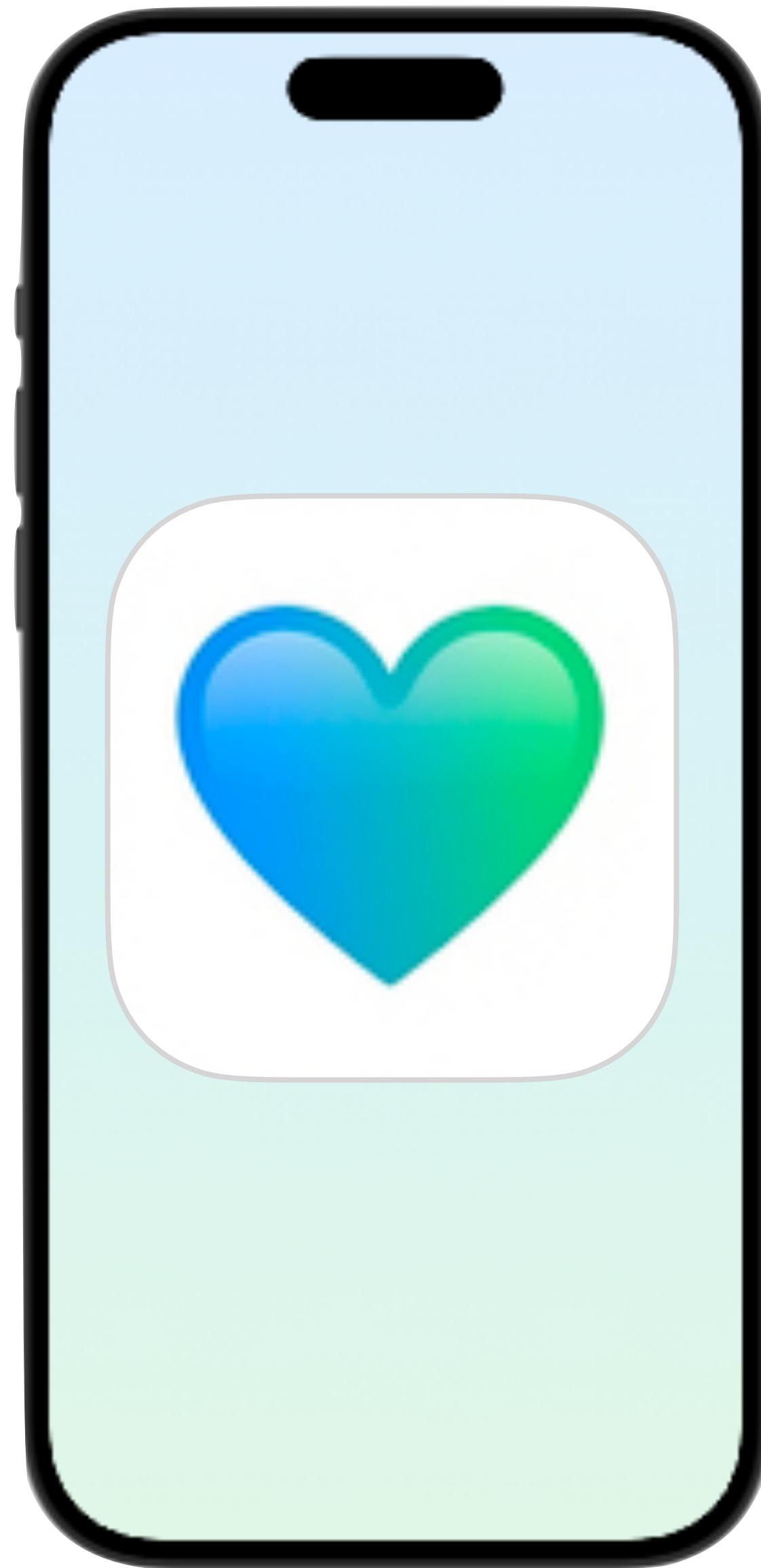
**Issue:** January 2026



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

# แอปนี้ใช้ทำอะไร ?





# Health Tips App

หน่วยงานด้านสาธารณสุขในมหาวิทยาลัยต้องการพัฒนาแอปขนาดเล็กสำหรับเผยแพร่คำแนะนำด้านสุขภาพพื้นฐานแก่บุคลากรและนักศึกษา

- แอปดังกล่าวจะถูกใช้งานบน iPhone เป็นหลัก
- มีวัตถุประสงค์เพื่อให้ผู้ใช้สามารถเปิดอ่านคำแนะนำได้อย่างรวดเร็วในช่วงเวลาสั้นๆ เช่น ระหว่างพักเรียนหรือพักทำงาน
- แอปนี้ไม่ได้มุ่งเน้นการบันทึกข้อมูลหรือการโต้ตอบซับซ้อน ไม่มีระบบสมาชิก ไม่มีการเก็บประวัติการใช้งาน และไม่มีระบบแจ้งเตือน
- จุดประสงค์หลักคือ “การนำเสนอข้อมูลที่อ่านง่าย เข้าใจเร็ว และนำไปใช้ได้ทันที”

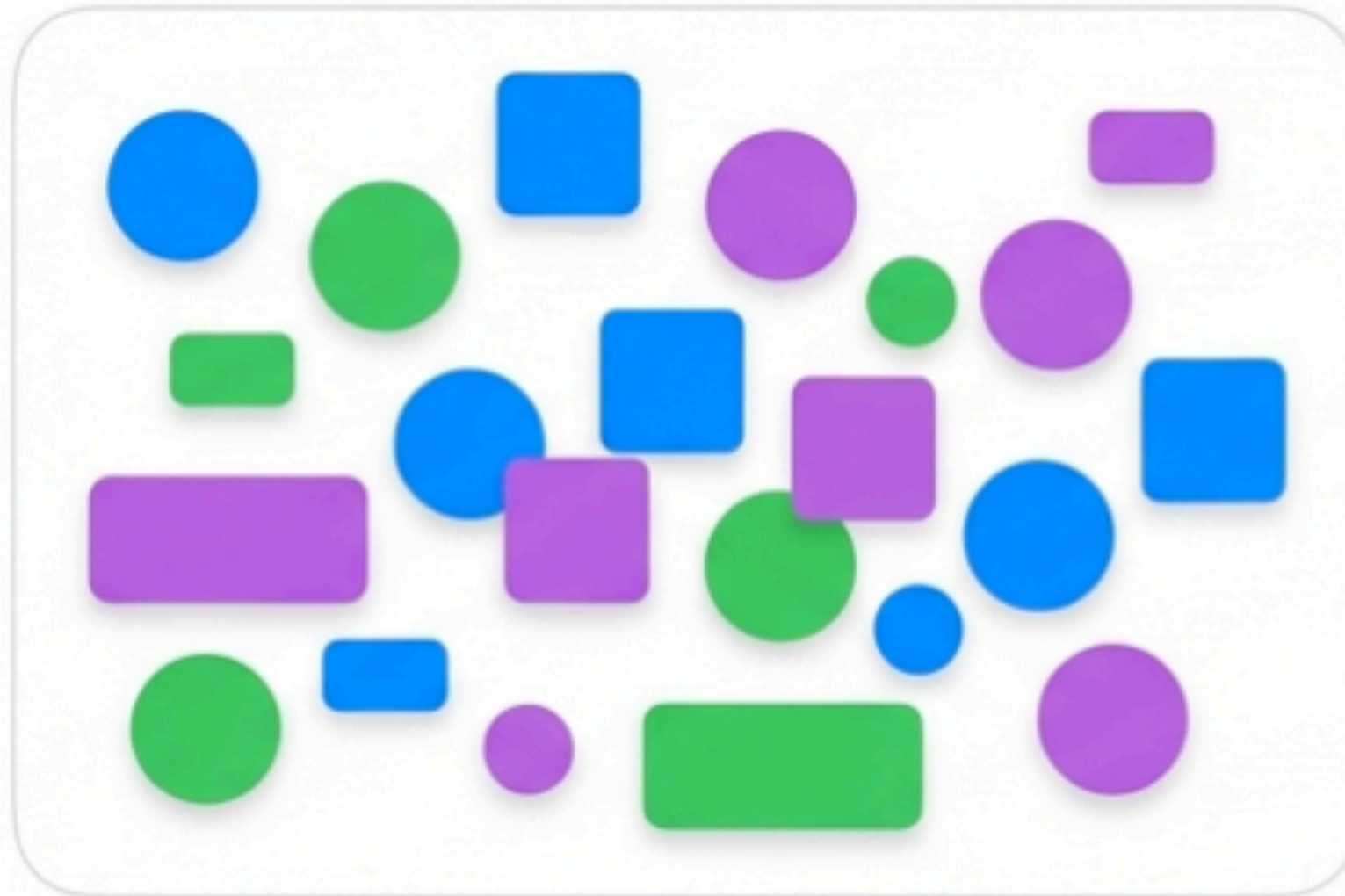
สิ่งที่สำคัญจึงไม่ใช่การจัดวางองค์ประกอบอย่างไรให้สวย แต่คือ  
**“ผู้ใช้ควรเข้าใจอะไรภายในไม่กี่วินาทีแรก”**



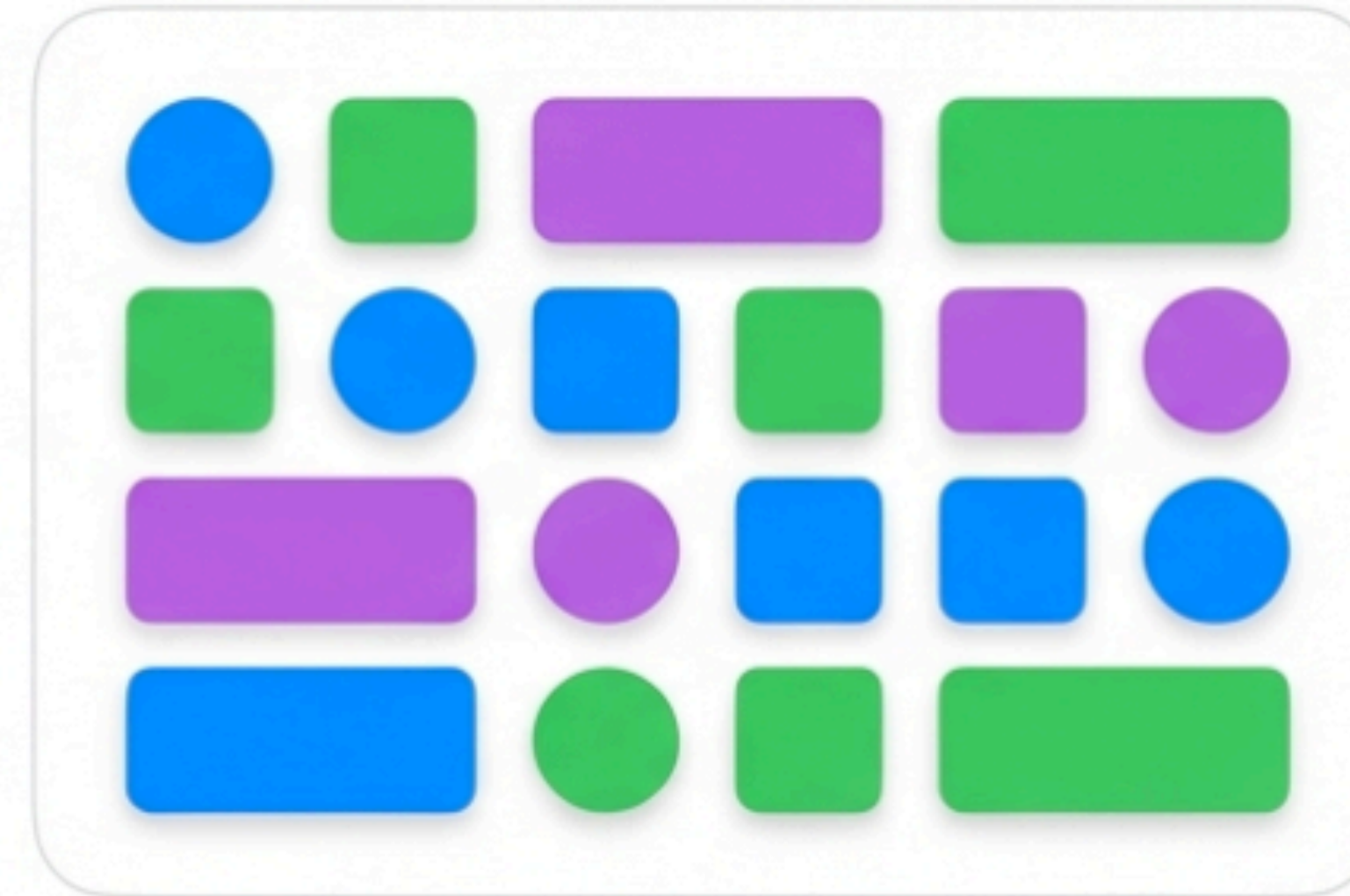
# UI คือโครงสร้างของการสื่อสาร

การออกแบบ UI ไม่ใช่เพียงการจัดวางให้สวยงาม แต่เป็นตัวกลางที่ช่วยให้ผู้ใช้เข้าใจสาระสำคัญได้ทันที

Just Decoration



Structured Communication



- Reduce Cognitive Load  
(ลดภาระทางความคิด)

- Support Perception  
(สนับสนุนการรับรู้)

- Semantic Relationships  
(ความสัมพันธ์ซึ่งความหมาย)



# Human Interface Guidelines



## ความชัดเจน (Clarity)

ผู้ใช้ต้องเข้าใจวัตถุประสงค์ของ  
หน้าจอได้ทันที สื่อสารสาระสำคัญ  
ได้อย่างรวดเร็ว



## ความสม่ำเสมอ (Consistency)

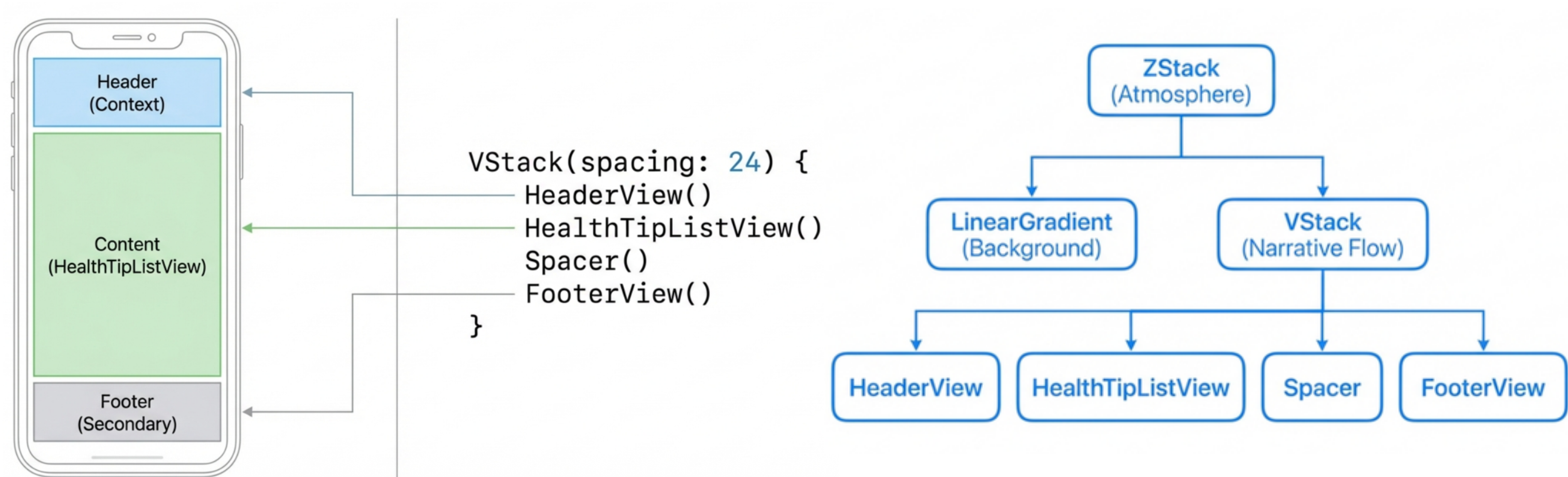
ใช้รูปแบบที่สอดคล้องกับพฤติกรรม  
การใช้งาน เพื่อให้สามารถคาดเดา  
การใช้งานได้



## เน้นเนื้อหา (Deference to Content)

UI ต้องทำหน้าที่สนับสนุนเนื้อหา  
ไม่แย่งความสนใจจากเนื้อหาหลัก

# การกำหนดลำดับความสำคัญของข้อมูล (Visual Hierarchy)



ก่อนลงมือเขียนโค้ดจริง นักออกแบบควรกำหนดโครงสร้างของหน้าจอในเชิงแนวคิดเสียก่อน เช่น ลำดับบนลงล่างของข้อมูล การจัดกลุ่มของไอคอนและข้อความ หรือบทบาทของพื้นหลัง การคิดเชิงโครงสร้างล่วงหน้าจะทำให้โค้ดที่เขียนขึ้นสะท้อนแนวคิดการออกแบบอย่างชัดเจน แทนที่จะเป็นผลจากการทดลองแบบลองผิดลองถูก



# การสร้างลำดับชั้นทางตัวอักษร (Typographic hierarchy)



การใช้ขนาดตัวอักษรที่แตกต่างกันอย่างชัดเจน การใช้สีเข้มกับหัวข้อและการใช้สีรองสำหรับข้อความอธิบาย เป็นการสร้าง **ลำดับชั้นทางตัวอักษร (typographic hierarchy)** ซึ่งช่วยให้สายตาของผู้ใช้เคลื่อนที่อย่างเป็นธรรมชาติจากข้อมูลหลักไปสู่ข้อมูลรอง หากทั้งสองข้อความมีขนาดและน้ำหนักเท่ากัน จะทำให้ผู้ใช้ต้องใช้ความพยายามมากขึ้นในการแยกแยะความสำคัญ ซึ่งขัดกับหลักการลดภาระทางความคิด (cognitive load reduction)

## การจัดกลุ่มข้อมูล (Meaningful Group)



### Drink Water

Stay hydrated by drinking 6–8 glasses daily.

สำหรับแต่ละคำแนะนำด้านสุขภาพ นักออกแบบควรมองว่าเป็น “หน่วยข้อมูล” ที่ต้องจัดกลุ่มอย่างมีความหมาย ไอคอน หัวข้อ และคำอธิบาย ไม่ใช่องค์ประกอบที่แยกจากกัน แต่เป็นข้อมูลชุดเดียวกัน การจัดกลุ่มอย่างเหมาะสมช่วยให้ผู้ใช้รับรู้ความสัมพันธ์ของข้อมูลได้ทันที ลดความสับสน และสนับสนุนหลักความสม่ำเสมอ (Consistency)



การจัดวางองค์ประกอบมิได้เป็นเพียงการควบคุมตำแหน่งเชิงพิกัด แต่เป็นการกำหนดความสัมพันธ์เชิงความหมาย (semantic relationships) ระหว่างองค์ประกอบต่าง ๆ ในหน้าจอ



การจัดกลุ่มข้อมูลที่มีความสัมพันธ์  
ในระดับเดียวกัน

ลำดับความสำคัญและลำดับการอ่าน

องค์ประกอบหลักกับองค์ประกอบรอง



Product Name: HealthTips

Team: Add account...

Organization Identifier: ios.ajthti.com

Bundle Identifier: ios.ajthti.com.HealthTips

Interface: SwiftUI

Language: Swift

Testing System: None

Storage: None

☐ Host in CloudKit





```
struct ContentView: View {  
    var body: some View {  
        ZStack {  
            LinearGradient(  
                colors: [  
                    Color.blue.opacity(0.15),  
                    Color.green.opacity(0.15)  
                ],  
                startPoint: .top,  
                endPoint: .bottom  
            )  
            .ignoresSafeArea()  
  
            VStack(spacing: 24) {  
                HeaderView()  
                HealthTipListView()  
                Spacer()  
                FooterView()  
            }  
            .padding()  
        }  
    }  
}
```

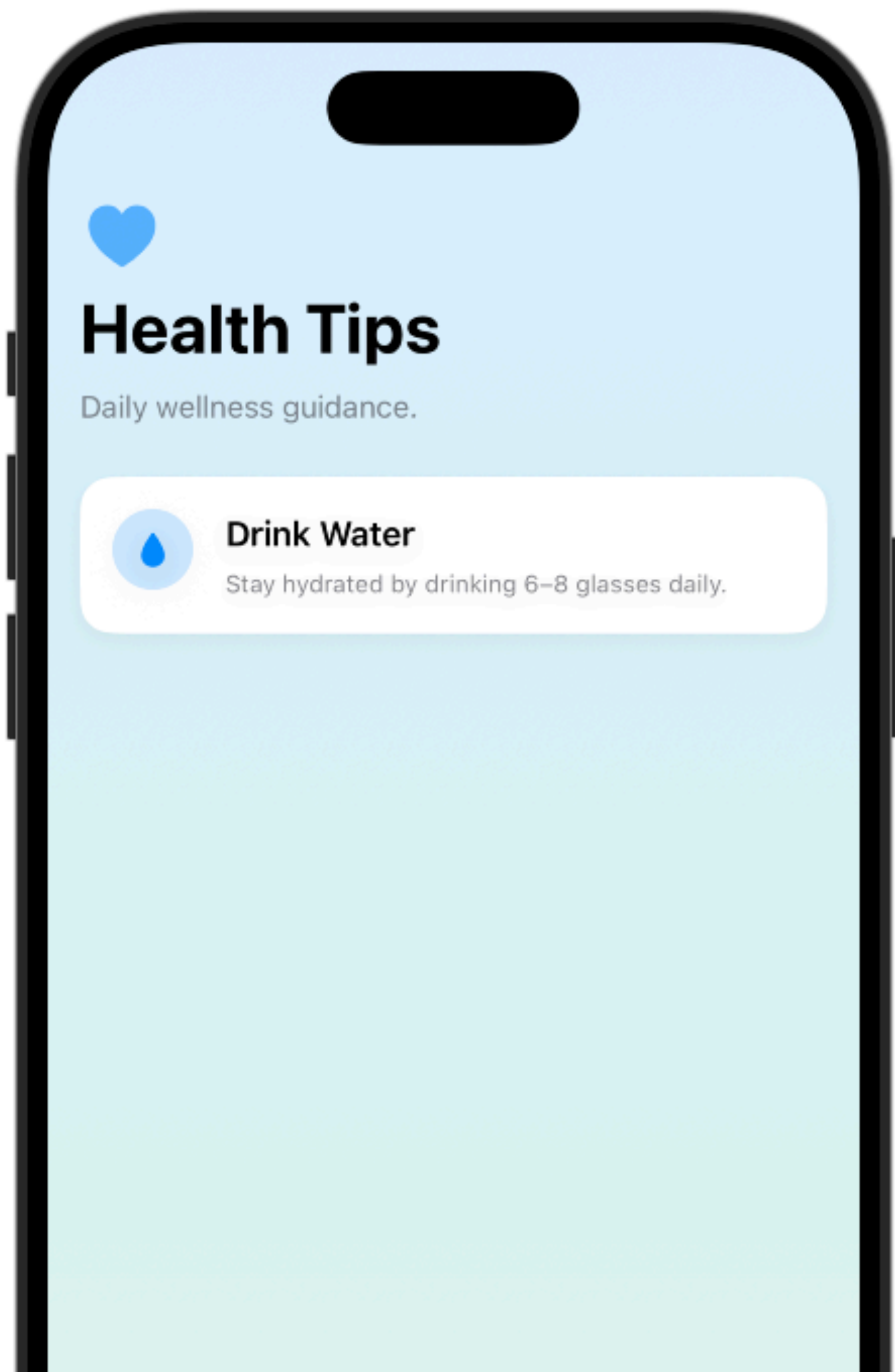


```
struct HeaderView: View {  
    var body: some View {  
        VStack(alignment: .leading, spacing: 8) {  
  
            Image(systemName: "heart.fill")  
                .font(.largeTitle)  
                .foregroundColor(.blue.opacity(0.6))  
  
            Text("Health Tips")  
                .font(.system(size: 34, weight: .bold))  
  
            Text("Daily wellness guidance.")  
                .font(.subheadline)  
                .foregroundColor(.secondary)  
        }  
        .frame(maxWidth: .infinity, alignment: .leading)  
    }  
}
```



```
struct FooterView: View {  
    var body: some View {  
        Text("Stay healthy, feel great!")  
            .font(.footnote)  
            .foregroundColor(.secondary)  
            .multilineTextAlignment(.center)  
            .frame(maxWidth: .infinity)  
    }  
}
```





```

struct HealthTipCardView: View {
    let title: String
    let detail: String
    let systemImage: String
    let iconColor: Color

    var body: some View {
        HStack(alignment: .top, spacing: 16) {

            ZStack {
                Circle()
                    .fill(iconColor.opacity(0.2))
                    .frame(width: 40, height: 40)

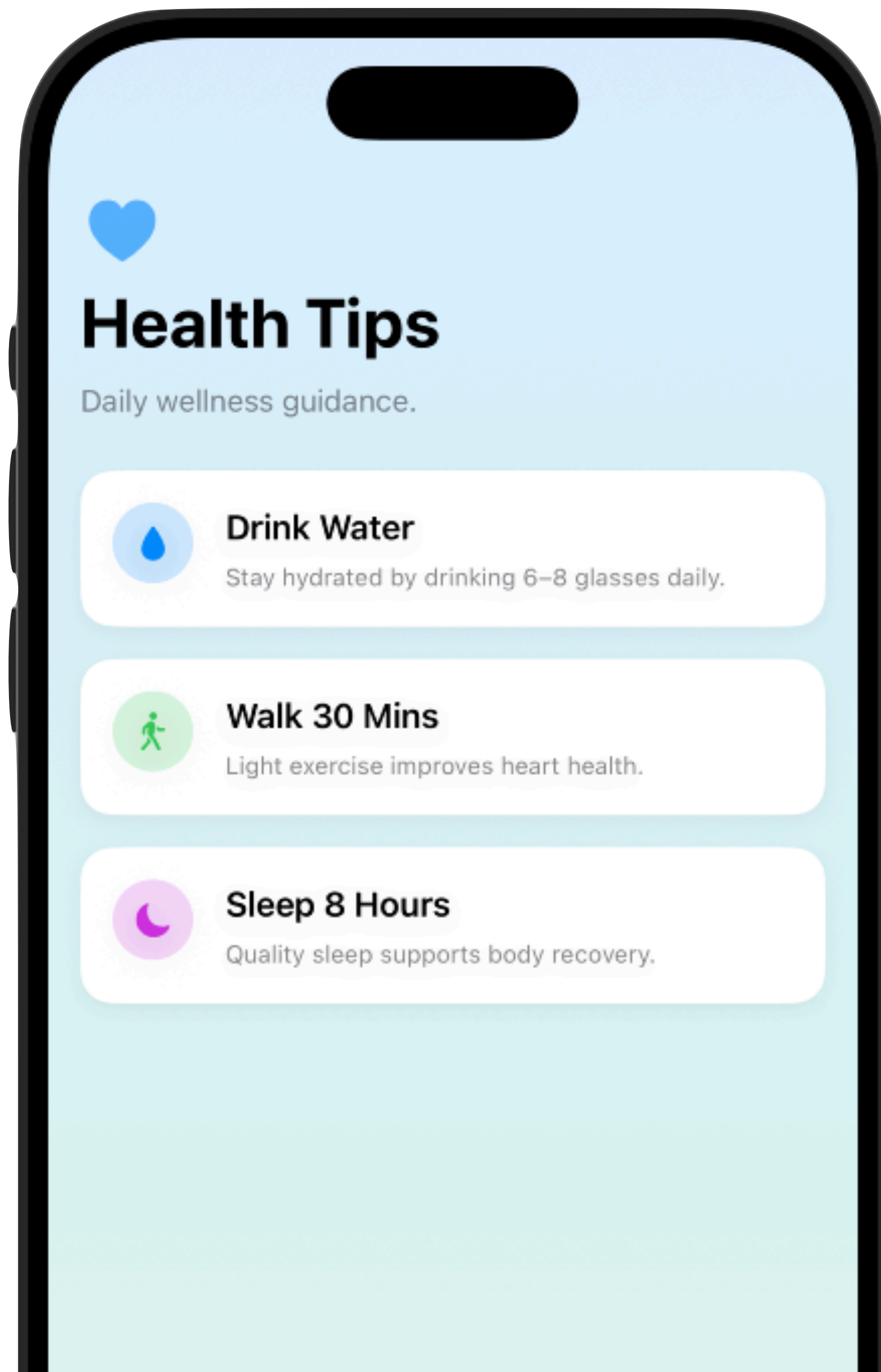
                Image(systemName: systemImage)
                    .foregroundColor(iconColor)
            }

            VStack(alignment: .leading, spacing: 4) {
                Text(title)
                    .font(.headline)

                Text(detail)
                    .font(.caption)
                    .foregroundStyle(.secondary)
            }

            Spacer()
        }
        .padding()
        .background(
            RoundedRectangle(cornerRadius: 16)
                .fill(Color.white)
        )
        .shadow(color: .black.opacity(0.05), radius: 5, x: 0, y: 3)
    }
}

```



```
struct HealthTipListView: View {
    var body: some View {
        VStack(spacing: 16) {
            HealthTipCardView(
                title: "Drink Water",
                detail: "Stay hydrated by drinking 6-8 glasses daily.",
                systemImage: "drop.fill",
                iconColor: .blue
            )

            HealthTipCardView(
                title: "Walk 30 Mins",
                detail: "Light exercise improves heart health.",
                systemImage: "figure.walk",
                iconColor: .green
            )

            HealthTipCardView(
                title: "Sleep 8 Hours",
                detail: "Quality sleep supports body recovery.",
                systemImage: "moon.fill",
                iconColor: .purple
            )
        }
    }
}
```

# คำถามเพื่อการสะท้อนผลการเรียนรู้จากการทำกิจกรรม

1. อธิบายความหมายของการออกแบบส่วนติดต่อผู้ใช้ (UI) ในฐานะ “โครงสร้างของการสื่อสาร” พร้อมยกตัวอย่างมาโดยสังเขป ?
2. อธิบายบทบาทของ VStack, HStack และ ZStack ใน SwiftUI ในฐานะโครงสร้างเชิงความหมาย ?
3. หลักการใดใน Human Interface Guidelines (Clarity, Consistency, Deference to Content) ถูกนำมาใช้ชัดเจนที่สุดในงานของคุณ และสะท้อนออกมาในโครงสร้าง UI อย่างไร ?
4. อธิบายเหตุผลในการแยก UI ออกเป็น View ย่อย (เช่น HeaderView, Content, FooterView) และผลที่เกิดขึ้นต่อความอ่านง่ายของโค้ดและความเข้าใจของผู้ใช้ ?
5. อธิบายว่าโครงสร้าง View hierarchy ใน SwiftUI ของคุณช่วยสนับสนุนการนำสายตา (visual hierarchy) และลดภาระทางความคิดของผู้ใช้อย่างไร ?
6. อธิบายบทบาทของ Card หรือ หน่วยข้อมูล (information unit) ในแอปที่คุณพัฒนา และเหตุใดองค์ประกอบภายใน Card จึงควรถูกจัดกลุ่มร่วมกัน ?
7. จากกิจกรรมทั้งหมด คุณสามารถอธิบายได้หรือไม่ว่า “โค้ด SwiftUI ที่ดี” ควรมีลักษณะอย่างไรในเชิงการออกแบบ พร้อมให้เหตุผลสนับสนุน ?