[**Project**](https://bbhosted.cuny.edu/webapps/assignment/uploadAssignment?content_id=_80353513_1&course_id=_2286381_1&group_id=&mode=view)

Ivan Chukharev

*Source code:*

**ContentView.swift**

//

// ContentView.swift

// Project

//

// Created by Ivan Chukharev on 12/1/23.

//

import SwiftUI

struct ContentView: View {

@State private var cartItems: [CartItem] = []

var body: some View {

NavigationView {

VStack {

TabView {

HomeView()

.tabItem {

Label("Home", systemImage: "house")

}

ShopView(addToCart: addToCart)

.tabItem {

Label("Shop", systemImage: "cart.badge.plus.fill")

}

AboutView()

.tabItem {

Label("About", systemImage: "info.circle")

}

ContactView()

.tabItem {

Label("Contacts", systemImage: "phone.circle")

}

CartView(cartItems: $cartItems)

.tabItem {

Label("Cart", systemImage: "cart")

}

}

.accentColor(.green)

}

}

}

func addToCart(item: ShopItem) {

if let index = cartItems.firstIndex(where: { $0.name == item.name }) {

cartItems[index].quantity += 1

} else {

cartItems.append(CartItem(name: item.name, price: item.price, imageName: item.imageName))

}

}

}

struct ShopItem: Identifiable {

let id = UUID()

let name: String

let price: Double

let imageName: String

}

struct HomeView: View {

var body: some View {

ZStack {

Image("hero")

.resizable()

.scaledToFill()

.edgesIgnoringSafeArea(.all)

VStack {

Text("Welcome to Pet's Life App!")

.font(.largeTitle)

.foregroundColor(.brown)

.padding()

Text("Explore our shop, learn about us, and get in touch with us.")

.foregroundColor(.brown)

.padding()

}

}

}

}

struct ShopView: View {

var addToCart: (ShopItem) -> Void

let shopItems: [ShopItem] = [

ShopItem(name: "Item 1", price: 9.99, imageName: "p1"),

ShopItem(name: "Item 2", price: 19.99, imageName: "p2"),

ShopItem(name: "Item 3", price: 29.99, imageName: "p3"),

ShopItem(name: "Item 4", price: 39.99, imageName: "p4"),

ShopItem(name: "Item 5", price: 9.99, imageName: "p5"),

ShopItem(name: "Item 6", price: 19.99, imageName: "p6"),

ShopItem(name: "Item 7", price: 29.99, imageName: "p7"),

ShopItem(name: "Item 8", price: 39.99, imageName: "p8"),

]

var body: some View {

VStack {

HStack {

Image("logo")

.resizable()

.aspectRatio(contentMode: .fit)

.frame(width: 50, height: 50)

.cornerRadius(15)

Text("Shop Here!")

.font(.largeTitle)

.foregroundColor(.brown)

.fontWeight(.bold)

}

.padding()

List(shopItems) { item in

ShopItemRow(item: item, addToCart: addToCart)

}

}

.padding()

}

}

struct ShopItemRow: View {

var item: ShopItem

var addToCart: (ShopItem) -> Void

var body: some View {

HStack {

Image(item.imageName)

.resizable()

.aspectRatio(contentMode: .fit)

.frame(width: 80, height: 80)

VStack(alignment: .leading) {

Text(item.name)

.font(.headline)

Text("$\(item.price, specifier: "%.2f")")

.foregroundColor(.gray)

}

Spacer()

Button("+") {

addToCart(item)

}

.padding()

.frame(width: 50, height: 50)

.background(Color.green)

.foregroundColor(.brown)

.cornerRadius(8)

}

.padding(.vertical, 8)

}

}

struct CartItem: Identifiable {

let id = UUID()

let name: String

let price: Double

let imageName: String

var quantity: Int = 1

var totalPrice: Double {

return Double(quantity) \* price

}

}

struct CartView: View {

@Binding var cartItems: [CartItem]

@State private var showAlert = false

var body: some View {

VStack {

HStack {

Image("logo")

.resizable()

.aspectRatio(contentMode: .fit)

.frame(width: 50, height: 50)

.cornerRadius(15)

Text("Choping Cart")

.font(.largeTitle)

.foregroundColor(.brown)

.fontWeight(.bold)

}

.padding()

List {

ForEach(cartItems) { item in

HStack {

Image(item.imageName)

.resizable()

.aspectRatio(contentMode: .fit)

.frame(width: 30, height: 30)

Text("\(item.name) (\(item.quantity)) - $\(item.totalPrice, specifier: "%.2f")")

}

}

.onDelete { indexSet in

cartItems.remove(atOffsets: indexSet)

}

}

HStack {

Spacer()

Text("Total: $\(calculateTotalPrice(), specifier: "%.2f")")

.font(.headline)

.padding(.vertical, 8)

Spacer()

}

Button("Checkout") {

if (cartItems.count != 0) {

cartItems.removeAll()

showAlert = true

}

}

.padding()

.frame(maxWidth: .infinity)

.background(Color.green)

.foregroundColor(.brown)

.cornerRadius(8)

.padding()

.alert(isPresented: $showAlert) {

Alert(

title: Text("Purchase Successful"),

message: Text("Thank you for your purchase!"),

dismissButton: .default(Text("OK"))

)

}

}

.padding()

}

private func calculateTotalPrice() -> Double {

return cartItems.reduce(0) { $0 + ($1.price \* Double($1.quantity)) }

}

}

struct AboutView: View {

var body: some View {

VStack(alignment: .leading, spacing: 20) {

HStack {

Image("logo")

.resizable()

.aspectRatio(contentMode: .fit)

.frame(width: 50, height: 50)

.cornerRadius(15)

Text("Who Are We?")

.font(.largeTitle)

.foregroundColor(.brown)

.fontWeight(.bold)

}

.padding()

Text("Welcome to our online store. We are dedicated to providing quality products and a seamless shopping experience.")

HStack {

Image("a1")

.resizable()

.aspectRatio(contentMode: .fit)

.frame(width: 150, height: 150)

.cornerRadius(10)

VStack(alignment: .leading, spacing: 8) {

Text("Our Mission")

.font(.title2)

.foregroundColor(.brown)

.fontWeight(.bold)

Text("At Pet's Life, our mission is to bring joy to pets and their owners. We curate a wide range of products, from essentials to unique items, to enhance the lives of your furry friends.")

}

}

}

.padding()

}

}

struct ContactView: View {

@State private var name = ""

@State private var email = ""

@State private var message = ""

@State private var showAlert = false

var body: some View {

VStack {

HStack {

Image("logo")

.resizable()

.aspectRatio(contentMode: .fit)

.frame(width: 50, height: 50)

.cornerRadius(15)

Text("Contact Us")

.font(.largeTitle)

.foregroundColor(.brown)

.fontWeight(.bold)

}

.padding()

Form {

Section(header: Text("Your Information")) {

TextField("Name", text: $name)

TextField("Email", text: $email)

}

Section(header: Text("Message")) {

TextEditor(text: $message)

}

Section {

Button("Submit") {

if(name != "" && email != "" && message != ""){

showAlert = true

name = ""

email = ""

message = ""

}

}

.alert(isPresented: $showAlert) {

Alert(

title: Text("Feedback recieved"),

message: Text("Thank you for your feedback!"),

dismissButton: .default(Text("OK"))

)

}

}

}

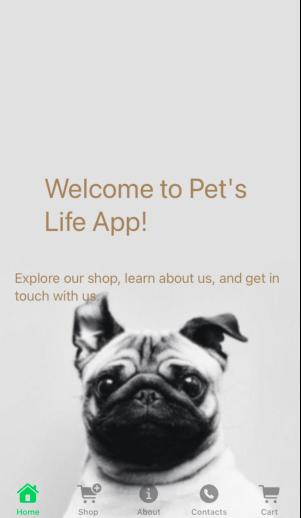
.padding()

}

}

}

*Screenshots:*



A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generatedA hand pushing a shopping cart

Description automatically generatedA screenshot of a phone

Description automatically generatedScreens screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated