```
// ===== File: DelegationApp/App/AppContainer.swift =====
//
// AppContainer.swift
// iCuno test
//
// Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation
import SwiftUI

/// ??????? DI-?????????: ??????????? ???? ????????.
final class AppContainer: ObservableObject {
    let taskService: TaskService
    let chatService: ChatService
    let profileService: ProfileService

    init(
        taskService: TaskService,
        chatService: ChatService,
        profileService: ProfileService,
    ) {
        self.taskService = taskService
        self.chatService = chatService
        self.profileService = profileService
    }
}

extension AppContainer {
    /// ????????? ??? ??????/??????? ?????? ? ?????? ????????.
    static let preview = AppContainer(
        taskService: MockTaskService(),
        chatService: MockChatService(),
        profileService: MockProfileService()
    )
}


// ===== File: DelegationApp/App/AppRouter.swift =====
////
////  RouteView.swift
////  iCuno test
////
////  RootView ? ????????.
////
//
//import SwiftUI
//
//struct RootView: View {
//    @EnvironmentObject var container: AppContainer
//    @State private var selectedTab = 0
//
//    var body: some View {
//        TabView(selection: $selectedTab) {
//
//            // ??????? "?????"
//            NavigationStack {
//                MapScreen(vm: .init(service: container.taskService))
//            }
//            .tabItem {
//                Label("?????", systemImage: "map")
//            }
//            .tag(0)
//
//            // ??????? "???????"
//            NavigationStack {
//                RouteScreen(vm: .init(service: container.taskService))
//            }
//            .tabItem {
//                Label("???????", systemImage: "point.topleft.down.curvedto.point.bottomright.up")
//            }
//            .tag(1)
//
//            // ????? ??????? "??????????"
//            NavigationStack {
//                MyAdsScreen()
//            }
//            .tabItem {
//                Label("??????????", systemImage: "rectangle.stack.badge.plus")
//            }
//            .tag(2)
//
//            // ??????? "????"
//            NavigationStack {
//                ChatsScreen(vm: .init(service: container.chatService))
//            }
//            .tabItem {
//                Label("????", systemImage: "bubble.left.and.bubble.right")
//            }
//            .tag(3)
//
//            // ??????? "???????"
//            NavigationStack {
//                ProfileScreen(vm: .init(service: container.profileService))
//            }
//            .tabItem {
//                Label("???????", systemImage: "person.circle")
//            }
//            .tag(4)
//        }
//        .tint(Theme.ColorToken.turquoise)
//        .background(Color.black)
//    }
//}


// ===== File: DelegationApp/App/DelegationApp.swift =====
import SwiftUI
import YandexMapsMobile
//
//@main
//struct DelegationApp: App {
//    @StateObject private var container = AppContainer.preview
//
//    init() {
//        // ? ????????????? Yandex MapKit
//        // ???? ??????? ???? ????, ??? ??, ??? ?? ??? ???????????? ? ???????? ???????
//        YMKMapKit.setApiKey("df3f9145-2080-42b7-9b91-b879c34236bb")
//        YMKMapKit.sharedInstance()
//    }
//
////    var body: some Scene {
////        WindowGroup {
////            RootView()
////                .environmentObject(container)
////        }
////    }
//    let service = MockTaskService()
//    let searchService = AddressSearchService()
//    let vm = MapViewModel(service: service, searchService: searchService)
//
//    var body: some Scene {
//        WindowGroup {
//            MapScreen(vm: vm)
//        }
//    }
//}
//
////#Preview {
////    let service = MockTaskService()
////    let searchService = AddressSearchService()
////    let vm = MapViewModel(service: service, searchService: searchService)
////    MapScreen(vm: vm)
////}

//@main
//struct DelegationApp: App {
//    @StateObject private var container = AppContainer.preview
//    @StateObject private var mapVM: MapViewModel
//
//    init() {
//        // ????????????? Yandex MapKit
//        YMKMapKit.setApiKey("df3f9145-2080-42b7-9b91-b879c34236bb")
//        YMKMapKit.sharedInstance()
//
//        let service = MockTaskService()
//        let searchService = AddressSearchService()
//        _mapVM = StateObject(wrappedValue: MapViewModel(service: service,
//                                                         searchService: searchService))
//    }
//
//    var body: some Scene {
//        WindowGroup {
////            MapScreen(vm: mapVM)
////            ChatsScreen(
////                vm: ChatsViewModel(service: MockChatService())
////            )
//
//
////            NavigationStack {
////                RouteScreen(vm: PreviewData.routeVM)
////            }
////            .preferredColorScheme(.light)
//
//            let service = MockTaskService()
//            let searchService = AddressSearchService()
//            let vm = MapViewModel(service: service, searchService: searchService)
//
//            return NavigationStack {
//                MapScreen(vm: vm)
//            }
//        }
//    }
//}

@main
struct DelegationApp: App {
    /// ????? ????????? ????????????.
    @StateObject private var container = AppContainer.preview

    init() {
        // ??? ??????? ????? ????????????? ??????? ???????????????? SDK ????:
        // YandexMapConfigurator.configureIfNeeded()
        //
        // ?? ?????? ??? ?????? ??? ??? ????? ? ??????? (AddressSearchService / YandexMapView),
        // ??? ??? ????? ????? ?? ??????? ?? ??????????? Map SDK.
    }

    var body: some Scene {
        WindowGroup {
            RootView()
```

```
                .environmentObject(container)
                .ignoresSafeArea()
        }
    }
}

/// ???????, ????? ????????, ??? ??? ??????????? ? SwiftUI Preview.
enum RuntimeEnvironment {
    static var isPreview: Bool {
        #if DEBUG
        if ProcessInfo.processInfo.environment["XCODE_RUNNING_FOR_PREVIEWS"] == "1" {
            return true
        }
        #endif
        return false
    }
}


// ===== File: DelegationApp/App/YandexMapView.swift =====
import SwiftUI
import YandexMapsMobile

/// ??????? ??? YMKMapView ??? ????????????? ? SwiftUI.
///
/// ?????: ? ?????? ?? ?? ??????? ???????? ????? ??????,
/// ????? ?? ????? SwiftUI Preview.
struct YandexMapView: UIViewRepresentable {

    /// ?????????? ?????? ?????.
    @Binding var centerPoint: YMKPoint?

    final class Coordinator {
        var mapView: YMKMapView?
        var placemark: YMKPlacemarkMapObject?
    }

    func makeCoordinator() -> Coordinator {
        Coordinator()
    }

    func makeUIView(context: Context) -> UIView {
        // ?????????, ? ??????? ??? ??????? ??????? ??????? YMKMapView.
        let container = UIView()
        container.backgroundColor = .clear
//        // ? ?????? ? ?????? ?? ?????????, ?????? ?????? UIView.
//        guard !RuntimeEnvironment.isPreview else {
//            return container
//        }

        // ? ??????? ??????? ?????????????? SDK ? ?????.
        YandexMapConfigurator.configureIfNeeded()

        let mapView = YMKMapView(frame: .zero)
        mapView!.translatesAutoresizingMaskIntoConstraints = false
        container.addSubview(mapView!)

        NSLayoutConstraint.activate([
            mapView!.topAnchor.constraint(equalTo: container.topAnchor),
            mapView!.bottomAnchor.constraint(equalTo: container.bottomAnchor),
            mapView!.leadingAnchor.constraint(equalTo: container.leadingAnchor),
            mapView!.trailingAnchor.constraint(equalTo: container.trailingAnchor)
        ])

        context.coordinator.mapView = mapView

        // ????????? ?????.
        let startPoint = centerPoint ?? YMKPoint(
            latitude: 55.751244,
            longitude: 37.618423
        )
        updateMap(on: mapView!, coordinator: context.coordinator, to: startPoint)

        return container
    }

    func updateUIView(_ uiView: UIView, context: Context) {
        guard
//            !RuntimeEnvironment.isPreview,
            let mapView = context.coordinator.mapView,
            let point = centerPoint
        else { return }

        updateMap(on: mapView, coordinator: context.coordinator, to: point)
    }

    // MARK: - Internal helpers

    private func updateMap(
        on mapView: YMKMapView,
        coordinator: Coordinator,
        to point: YMKPoint
    ) {
        let map = mapView.mapWindow.map
        let position = YMKCameraPosition(
            target: point,
            zoom: 15,
            azimuth: 0,
            tilt: 0
        )
        let animation = YMKAnimation(type: .smooth, duration: 1.0)
        map.move(with: position, animation: animation, cameraCallback: nil)

        let mapObjects = map.mapObjects
        if let oldPlacemark = coordinator.placemark {
            mapObjects.remove(with: oldPlacemark)
        }
        let placemark = mapObjects.addPlacemark(with: point)
        coordinator.placemark = placemark
    }
}


// ===== File: DelegationApp/Core/Components/FilterChip.swift =====
//
//  FilterChip.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import SwiftUI

struct FilterChip: View {
    let title: String
    @Binding var isSelected: Bool

    var body: some View {
        Button {
            isSelected.toggle()
        } label: {
            HStack(spacing: 8) {
                if isSelected { Image(systemName: "checkmark") }
                Text(title)
```

```
                    .font(.system(size: 15, weight: .semibold))
            }
            .padding(.vertical, 10)
            .padding(.horizontal, 14)
            .background(
                RoundedRectangle(cornerRadius: Theme.Radius.l, style: .continuous)
                    .fill(isSelected ? Theme.ColorToken.turquoise : Theme.ColorToken.milk)
            )
            .foregroundStyle(isSelected ? Color.white : Theme.ColorToken.textPrimary)
            .softCardShadow()
        }
        .buttonStyle(.plain)
    }
}


// ===== File: DelegationApp/Core/Components/FloatingPlusButton.swift =====
//
//  FloatingPlusButton.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import SwiftUI

struct FloatingPlusButton: View {
    var action: () -> Void
    var body: some View {
        Button(action: action) {
            Image(systemName: "plus")
                .font(.system(size: 24, weight: .bold))
                .foregroundStyle(Color.white)
                .frame(width: 64, height: 64)
                .background(Circle().fill(Theme.ColorToken.turquoise))
                .softCardShadow()
        }
        .buttonStyle(.plain)
        .accessibilityLabel("???????")
    }
}


// ===== File: DelegationApp/Core/Components/PriceTag.swift =====
//
//  PriceTag.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import SwiftUI

struct PriceTag: View {
    let price: Int
    let eta: Int
    var isHighlighted: Bool = false

    var body: some View {
        VStack(spacing: 4) {
            Text("\(price) ?")
                .font(.system(size: 16, weight: .semibold))
            Text("\(eta) ???")
                .font(.system(size: 12, weight: .regular))
                .foregroundStyle(Theme.ColorToken.textSecondary)
        }
        .padding(.horizontal, 16)
        .padding(.vertical, 10)
        .background(
            RoundedRectangle(cornerRadius: Theme.Radius.l, style: .continuous)
                .fill(Theme.ColorToken.white)
                .overlay(
                    RoundedRectangle(cornerRadius: Theme.Radius.l)
                        .stroke(isHighlighted ? Theme.ColorToken.turquoise : Color.clear, lineWidth: 2)
                )
        )
        .softCardShadow()
    }
}


// ===== File: DelegationApp/Core/Components/StarsView.swift =====
//
//  StarsView.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import SwiftUI

struct StarsView: View {
    let rating: Double
    let max: Int = 5

    var body: some View {
        HStack(spacing: 4) {
            ForEach(0..<max, id: \.self) { idx in
                let filled = rating >= Double(idx + 1) - 0.001
                Image(systemName: filled ? "star.fill" : "star")
                    .foregroundStyle(filled ? Theme.ColorToken.peach : Theme.ColorToken.textSecondary)
            }
        }
    }
}


// ===== File: DelegationApp/Core/Models/AdModels.swift =====
//
//  AdModels.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 24.11.2025.
//

//
//  AdModels.swift
//  iCuno test
//
//  ??????? ??? ?????? ??????????.
//

import Foundation

/// ?????? ??????????. ???? ???????????? ?????? ??? ???-??????
/// ?? ?????? "??? ??????????".
struct AdItem: Identifiable {
    let id: UUID = .init()
    let title: String
    let priceDescription: String
    let isExpired: Bool
    let views: Int
    let responses: Int
    let favorites: Int
```

```
}


// ===== File: DelegationApp/Core/Models/ChatModels.swift =====
import Foundation

struct ChatPreview: Identifiable {
    let id: UUID = .init()
    let initials: String
    let name: String
    let lastMessage: String
    let time: String
    let unreadCount: Int
}


// ===== File: DelegationApp/Core/Models/ProfileModels.swift =====
//
//  ProfileModels.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

struct Profile {
    let name: String
    let phone: String
    let rating: Double
    let completed: Int
    let cancelled: Int
}

struct Review: Identifiable {
    let id: UUID = .init()
    let authorInitial: String
    let authorName: String
    let text: String
    let ago: String
    let stars: Int
}


// ===== File: DelegationApp/Core/Models/TaskModels.swift =====
//
//  TaskModels.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

struct TaskItem: Identifiable {
    let id: UUID = .init()
    let title: String
    let price: Int        // ?
    let etaMinutes: Int   // ???
    let distanceKm: Double
}


// ===== File: DelegationApp/Core/Services/ChatService.swift =====
//
//  ChatService.swift
//  iCuno test
```

```
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

protocol ChatService {
    func loadChats() -> [ChatPreview]
}


// ===== File: DelegationApp/Core/Services/Mock/MockChatService.swift =====
import Foundation

final class MockChatService: ChatService {
    func loadChats() -> [ChatPreview] {
        [
            .init(initials: "?", name: "?????? ?.", lastMessage: "???????, ???!", time: "14:30", unreadCount: 2),
            .init(initials: "?", name: "???? ?.", lastMessage: "???? ????? ??? ???????", time: "?????", unreadCount: 0
        ]
    }
}


// ===== File: DelegationApp/Core/Services/Mock/MockProfileService.swift =====
//
//  MockProfileService.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

final class MockProfileService: ProfileService {
    func loadProfile() -> Profile {
        .init(name: "??????? ??????",
              phone: "+7 999 123-45-67",
              rating: 4.9,
              completed: 127,
              cancelled: 3)
    }
    func loadReviews() -> [Review] {
        [
            .init(authorInitial: "?", authorName: "????? ?.",
                  text: "???????? ???????????! ??? ?????? ?????? ? ???????????. ??????????!",
                  ago: "2 ??? ?????", stars: 5),
            .init(authorInitial: "?", authorName: "??????? ?.",
                  text: "????? ???????! ??????? ?????? ?????, ??? ?????????.",
                  ago: "?????? ?????", stars: 5)
        ]
    }
}


// ===== File: DelegationApp/Core/Services/Mock/MockTaskService.swift =====
//
//  MockTaskService.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

final class MockTaskService: TaskService {
    func loadNearbyTasks() -> [TaskItem] {
        [
            .init(title: "?????? ??????", price: 200, etaMinutes: 14, distanceKm: 1.1),
            .init(title: "??????? ???????", price: 400, etaMinutes: 10, distanceKm: 2.0),
            .init(title: "????????? ?????", price: 500, etaMinutes: 18, distanceKm: 3.5),
            .init(title: "?????? ???????", price: 250, etaMinutes: 7, distanceKm: 0.6)
        ]
    }
    func loadRouteTasks() -> [TaskItem] {
        [
            .init(title: "?????????? ??????", price: 350, etaMinutes: 8, distanceKm: 0.9),
            .init(title: "?????? ????", price: 150, etaMinutes: 12, distanceKm: 0.5)
        ]
    }
}


// ===== File: DelegationApp/Core/Services/Networking/APIClient.swift =====
//
//  APIClient.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//




// ===== File: DelegationApp/Core/Services/Networking/AddressSearchService.swift =====
// AddressSearchService.swift
// iCuno test / DelegationApp

import Foundation
import YandexMapsMobile

final class AddressSearchService {

    private let searchManager: YMKSearchManager?
    private var searchSession: YMKSearchSession?
    private let isEnabled: Bool

    init() {
//        // ? ?????? ????????? ??????.
//        if RuntimeEnvironment.isPreview {
//            self.searchManager = nil
//            self.isEnabled = false
//            return
//        }

        YandexMapConfigurator.configureIfNeeded()

        let managerType: YMKSearchManagerType = .combined
        let search = YMKSearch.sharedInstance()
        self.searchManager = search?.createSearchManager(with: managerType)
        self.isEnabled = (self.searchManager != nil)
    }

    func searchAddress(
        _ text: String,
        completion: @escaping (YMKPoint?) -> Void
    ) {
        let trimmed = text.trimmingCharacters(in: .whitespacesAndNewlines)
        guard !trimmed.isEmpty else {
            completion(nil)
            return
        }

        // ? ?????? ?????? ?????? ?? ????.
        guard isEnabled, let searchManager else {
            completion(nil)
            return
        }

        let bbox = YMKBoundingBox(
            southWest: YMKPoint(latitude: -85.0, longitude: -180.0),
            northEast: YMKPoint(latitude: 85.0, longitude: 180.0)
        )
        let geometry = YMKGeometry(boundingBox: bbox)

        let options = YMKSearchOptions()
        options.geometry = true

        searchSession = searchManager.submit(
            withText: trimmed,
            geometry: geometry,
            searchOptions: options
        ) { [weak self] response, error in
            defer { self?.searchSession = nil }

            if let error {
                print("Search error: \(error)")
                completion(nil)
                return
            }

            guard
                let collection = response?.collection,
                let firstItem = collection.children.first,
                let obj = firstItem.obj,
                let point = obj.geometry.first?.point
            else {
                completion(nil)
                return
            }

            completion(point)
        }
    }
}


// ===== File: DelegationApp/Core/Services/Networking/Endpoints.swift =====
//
//  Endpoints.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//



// ===== File: DelegationApp/Core/Services/ProfileService.swift =====
//
//  ProfileService.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

protocol ProfileService {
    func loadProfile() -> Profile
    func loadReviews() -> [Review]
}
```

```
protocol AddAnnouncementService {

}


// ===== File: DelegationApp/Core/Services/TaskService.swift =====
//
//  TaskService.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

protocol TaskService {
    func loadNearbyTasks() -> [TaskItem]
    func loadRouteTasks() -> [TaskItem]
}


// ===== File: DelegationApp/Core/Theme/Theme.swift =====
//
//  Theme.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import SwiftUI

enum Theme {
    enum ColorToken {
        static let turquoise = Color.hex("#3CC8C4")
        static let white     = Color.hex("#FFFFFF")
        static let milk      = Color.hex("#F7F3E9")
        static let peach     = Color.hex("#FFC9A6")
        static let textPrimary   = Color.black.opacity(0.9)
        static let textSecondary = Color.black.opacity(0.6)
        static let shadow = Color.black.opacity(0.08)
    }

    enum Radius {
        static let s: CGFloat = 10
        static let m: CGFloat = 16
        static let l: CGFloat = 24
        static let xl: CGFloat = 28
    }

    enum Spacing {
        static let xs: CGFloat = 6
        static let s: CGFloat = 8
        static let m: CGFloat = 12
        static let l: CGFloat = 16
        static let xl: CGFloat = 20
        static let xxl: CGFloat = 24
    }

    enum Shadow {
        static let soft = ShadowStyle(radius: 16, y: 8, opacity: 0.10)
        struct ShadowStyle {
            let radius: CGFloat
            let y: CGFloat
            let opacity: Double
        }
    }
}
```

```
extension View {
    /// ?????? ????????? ???? ??? iOS
    func softCardShadow() -> some View {
        shadow(color: Theme.ColorToken.shadow, radius: Theme.Shadow.soft.radius, x: 0, y: Theme.Shadow.soft.y)
    }
}


// ===== File: DelegationApp/Core/Utils/Extentions/Color+Hex.swift =====
//
//  Color+Hex.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 08.11.2025.
//

import SwiftUI

extension Color {
    static func hex(_ hex: String) -> Color {
        let hex = hex.trimmingCharacters(in: CharacterSet.alphanumerics.inverted)
        var int: UInt64 = 0; Scanner(string: hex).scanHexInt64(&int)
        let a, r, g, b: UInt64
        switch hex.count {
        case 3: (a,r,g,b) = (255, (int >> 8) * 17, (int >> 4 & 0xF) * 17, (int & 0xF) * 17)
        case 6: (a,r,g,b) = (255, int >> 16, int >> 8 & 0xFF, int & 0xFF)
        case 8: (a,r,g,b) = (int >> 24, int >> 16 & 0xFF, int >> 8 & 0xFF, int & 0xFF)
        default:(a,r,g,b) = (255,0,0,0)
        }
        return Color(.sRGB,
                     red: Double(r)/255, green: Double(g)/255,
                     blue: Double(b)/255, opacity: Double(a)/255)
    }
}


// ===== File: DelegationApp/Features/Ads/AdsView/AdsScreen.swift =====
//
//  AdsScreen.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 24.11.2025.
//

//
//  MyAdsScreen.swift
//  iCuno test
//
//  ????? "??? ??????????" ?? ??????? ?????.
//

import SwiftUI

/// ????? "??? ??????????".
struct MyAdsScreen: View {
    @State private var selectedFilter: AdsFilter = .waiting
    @State private var showNewAdSheet = false

    // ???? ??? ???????. ????? ????? ???????? ??????? ???????.
    private let ads: [AdItem] = [
        .init(
            title: "?????? ? ??????????? ????????",
            priceDescription: "?? 300 ? ?? ??????",
            isExpired: true,
            views: 58,
            responses: 1,
            favorites: 3
        )
    ]

    var body: some View {
        ZStack(alignment: .bottom) {
            ScrollView {
                VStack(alignment: .leading, spacing: Theme.Spacing.l) {
                    summarySection
                    filtersSection
//                    promoSection
                    expiredSection
                }
                .padding(.horizontal, Theme.Spacing.l)
                .padding(.top, Theme.Spacing.m)
                // ????? ????? ??? ?????? ??????
            }


            newAdButton
        }
        .sheet(isPresented: $showNewAdSheet) {
            NewAdCategoryScreen()
        }
        .navigationTitle("??? ??????????")
        .navigationBarTitleDisplayMode(.inline)
    }

    // MARK: - ?????????

    /// ??????? ? ????? ???????? ?????? ??????.
    private var summarySection: some View {
        HStack(spacing: Theme.Spacing.m) {
            SmallSummaryCard(
                title: "?????? ? ?????",
                subtitle: "????????? ??? ????????????",
                gradient: LinearGradient(
                    colors: [Color.hex("#B6FAC3"), Color.hex("#84A4FA")],
                    startPoint: .topLeading,
                    endPoint: .bottomTrailing
                )
            )

            SmallSummaryCard(
                title: "29 990 ?",
                subtitle: "??????????",
                gradient: LinearGradient(
                    colors: [Color.hex("#0D47A1"), Color.hex("#1976D2")],
                    startPoint: .topLeading,
                    endPoint: .bottomTrailing
                )
            )
        }
    }

    /// ??????? "???? ???????? / ???????? / ?????????".
    private var filtersSection: some View {
        HStack(spacing: Theme.Spacing.l) {
            ForEach(AdsFilter.allCases) { filter in
                VStack(spacing: 4) {
                    Text(filter.titleWithCount)
                        .font(.system(size: 15, weight: .semibold))
                        .foregroundStyle(filter == selectedFilter ? Theme.ColorToken.turquoise : Color.gray)

                    Rectangle()
                        .fill(filter == selectedFilter ? Theme.ColorToken.turquoise : Color.clear)
                        .frame(height: 3)
                        .cornerRadius(1.5)
                }
                .onTapGesture {
                    selectedFilter = filter
                }
            }
        }
        .padding(.top, Theme.Spacing.l)
    }

//    /// ????? ?????-???????? "?? 25% ?????? ??????".
//    private var promoSection: some View {
//        RoundedRectangle(cornerRadius: Theme.Radius.l, style: .continuous)
//            .fill(
//                LinearGradient(
//                    colors: [Color.hex("#0046A5"), Color.hex("#0059D6")],
//                    startPoint: .topLeading,
//                    endPoint: .bottomTrailing
//                )
//            )
//            .overlay(alignment: .leading) {
//                VStack(alignment: .leading, spacing: 4) {
//                    Text("?? 25% ?????? ??????")
//                        .font(.system(size: 16, weight: .semibold))
//                    Text("?????????? ??????????? ?????? ?? ??????? ?????????? ???????")
//                        .font(.system(size: 13))
//                        .fixedSize(horizontal: false, vertical: true)
//                }
//                .foregroundStyle(Color.gray)
//                .padding(16)
//            }
//            .frame(maxWidth: .infinity)
//    }

    /// ?????? ? ?????????? "????? ???? ??????????" ? ?????????? ??????????.
    private var expiredSection: some View {
        VStack(alignment: .leading, spacing: Theme.Spacing.m) {
//            Text("????? ???? ??????????")
//                .font(.system(size: 15, weight: .semibold))
//                .foregroundStyle(Color.white)

            ForEach(ads.filter { $0.isExpired }) { ad in
                AdCardView(ad: ad)
            }
        }
        .padding(.top, Theme.Spacing.l)
    }

    /// ?????? ??????? ?????? "?????????? ??????????".
    private var newAdButton: some View {
        Button {
            showNewAdSheet = true
        } label: {
            Text("?????????? ??????????")
                .font(.system(size: 17, weight: .semibold))
                .foregroundStyle(Color.white)
                .frame(maxWidth: .infinity)
                .padding(.vertical, 14)
                .background(
                    RoundedRectangle(cornerRadius: 18, style: .continuous)
                        .fill(Color.black.opacity(0.7))
                )
                .padding(.horizontal, Theme.Spacing.l)

        }
        .buttonStyle(.plain)
    }
```

```
}

// MARK: - ??????????????? ???? ? ??????

/// ??? ??????? ? ??????? ????????.
private enum AdsFilter: CaseIterable, Identifiable {
    case waiting
    case active
    case drafts

    var id: Self { self }

    var title: String {
        switch self {
        case .waiting: return "???? ????????"
        case .active:  return "????????"
        case .drafts:  return "?????????"
        }
    }

    /// ??? ??????? ???????????? ?? ?? ?????, ??? ? ?? ??????.
    var count: Int {
        switch self {
        case .waiting: return 1
        case .active:  return 1
        case .drafts:  return 0
        }
    }

    var titleWithCount: String {
        "\(title) \(count)"
    }
}

/// ????????? ???????? ?????? ("?????? ? ?????" / "29 990 ?...").
private struct SmallSummaryCard: View {
    let title: String
    let subtitle: String
    let gradient: LinearGradient

    var body: some View {
        ZStack(alignment: .leading) {
            RoundedRectangle(cornerRadius: Theme.Radius.l, style: .continuous)
                .fill(gradient)

            VStack(alignment: .leading, spacing: 4) {
                Text(title)
                    .font(.system(size: 15, weight: .semibold))
                Text(subtitle)
                    .font(.system(size: 12))
                    .foregroundStyle(Color.black.opacity(0.8))
            }
            .foregroundStyle(Color.black)
            .padding(12)
        }
        .frame(maxWidth: .infinity, minHeight: 72)
    }
}

/// ???????? ?????? ??????????.
private struct AdCardView: View {
    let ad: AdItem

    var body: some View {
        VStack(alignment: .leading) {
            HStack(alignment: .top, spacing: 12) {
                RoundedRectangle(cornerRadius: 12, style: .continuous)
```

```
                    .fill(Color.gray.opacity(0.4))
                    .frame(width: 96, height: 72)
                    .overlay(
                        Image(systemName: "photo")
                            .font(.system(size: 24))
                            .foregroundStyle(Color.white.opacity(0.7))
                    )

                VStack(alignment: .leading, spacing: 4) {
                    Text(ad.title)
                        .font(.system(size: 16, weight: .semibold))
                        .foregroundStyle(Color.gray)
                        .lineLimit(2)

                    Text(ad.priceDescription)
                        .font(.system(size: 15, weight: .semibold))
                        .foregroundStyle(Color.gray)

                    Text("????? ???? ??????????")
                        .font(.system(size: 13))
                        .foregroundStyle(Color.gray)
                }

                Spacer()

                Image(systemName: "pencil")
                    .font(.system(size: 16, weight: .semibold))
                    .foregroundStyle(Color.gray.opacity(0.8))
            }

            HStack(spacing: 12) {
                IconCounterView(systemName: "eye", text: "\(ad.views)")
                IconCounterView(systemName: "person", text: "\(ad.responses)")
                IconCounterView(systemName: "heart", text: "\(ad.favorites)")
            }
            .font(.system(size: 13))
            .foregroundStyle(Color.gray)
            .padding(.vertical, 10)
        }
        .padding(12)
        .background(Color.secondary.opacity(0.1))
        .cornerRadius(15)
//        .background(
//            RoundedRectangle(cornerRadius: Theme.Radius.l, style: .continuous)
//                .fill(Color.white.opacity(0.5))
//        )
    }
}

/// ?????????? "?????? + ?????", "????????? + ?????" ? ?.?.
private struct IconCounterView: View {
    let systemName: String
    let text: String

    var body: some View {
        HStack(spacing: 4) {
            Image(systemName: systemName)
            Text(text)
        }
    }
}

//#Preview("MyAdsScreen") {
//    NavigationStack {
//        MyAdsScreen()
//    }
//    .preferredColorScheme(.dark)
```

```
//}


// ===== File: DelegationApp/Features/Ads/AdsView/NewAdCategoryScreen.swift =====
//
//  NewAdCategoryScreen.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 24.11.2025.
//

//
//  NewAdCategoryScreen.swift
//  iCuno test
//
//  ????? "????? ??????????" (????? ?????????).
//

import SwiftUI

/// ????? ?????? ????????? ??? ?????? ??????????.
struct NewAdCategoryScreen: View {
    @Environment(\.dismiss) private var dismiss

    private let categories: [AdCategory] = [
        .init(title: "???????? ?? ????", systemImage: "car.fill"),
        .init(title: "?????? ?????????", systemImage: "building.2.fill"),
        .init(title: "?????? ???????????", systemImage: "briefcase.fill"),
        .init(title: "?????? ??????", systemImage: "scissors"),
        .init(title: "???????????????? ??????", systemImage: "swift")
    ]

    var body: some View {
        NavigationStack {
            VStack(alignment: .leading, spacing: 0) {
                header
                categoriesList
                Spacer()
            }
        }
    }

    // MARK: - ???????

    private var header: some View {
        VStack(alignment: .leading, spacing: 16) {
            HStack {
                Button {
                    dismiss()
                } label: {
                    Image(systemName: "xmark")
                        .font(.system(size: 18, weight: .semibold))
                        .foregroundStyle(Color.gray)
                        .padding(8)
                }

                Spacer()
            }

            Text("????? ??????????")
                .font(.system(size: 24, weight: .bold))
                .foregroundStyle(Color.white)
        }
        .padding(.horizontal, Theme.Spacing.l)
        .padding(.top, Theme.Spacing.m)
        .padding(.bottom, Theme.Spacing.l)
    }
    private var categoriesList: some View {
        VStack(spacing: 0) {
            ForEach(categories) { category in
                Button {
                    // ???? ?????? ????????? ?????.
                    // ????? ????? ????? ????? ????????? ????? ???????? ??????????.
                    dismiss()
                } label: {
                    HStack(spacing: 12) {
                        Image(systemName: category.systemImage)
                            .font(.system(size: 20))
                            .frame(width: 28, height: 28)
                            .foregroundStyle(Color.white)

                        Text(category.title)
                            .font(.system(size: 17))
                            .foregroundStyle(Color.gray)

                        Spacer()

                        Image(systemName: "chevron.right")
                            .font(.system(size: 15, weight: .semibold))
                            .foregroundStyle(Color.gray)
                    }
                    .padding(.horizontal, Theme.Spacing.l)
                    .padding(.vertical, 14)
                }
                .buttonStyle(.plain)

                Divider()
                    .background(Color.gray.opacity(0.6))
                    .padding(.leading, Theme.Spacing.l + 28 + 12)
            }
        }
    }
}

private struct AdCategory: Identifiable {
    let id: UUID = .init()
    let title: String
    let systemImage: String
}

//#Preview("NewAdCategoryScreen") {
//    NewAdCategoryScreen()
//        .preferredColorScheme(.dark)
//}


// ===== File: DelegationApp/Features/Chats/View/ChatsScreen.swift =====
import SwiftUI

struct ChatsScreen: View {
    @StateObject var vm: ChatsViewModel
    init(vm: ChatService) { _vm = StateObject(wrappedValue: .init(service: vm)) }
    init(vm: ChatsViewModel) { _vm = StateObject(wrappedValue: vm) }

    var body: some View {
        List {
            ForEach(vm.chats) { chat in
                HStack(spacing: 12) {
                    Circle()
                        .fill(LinearGradient(colors: [Theme.ColorToken.turquoise, Theme.ColorToken.peach],
                                             startPoint: .topLeading, endPoint: .bottomTrailing))
                        .frame(width: 44, height: 44)
                        .overlay(Text(chat.initials).foregroundStyle(.white).font(.system(size: 17, weight: .bold)))
                    VStack(alignment: .leading, spacing: 4) {
                        HStack {
                            Text(chat.name).font(.system(size: 16, weight: .semibold))
                            Spacer()
                            Text(chat.time).foregroundStyle(Theme.ColorToken.textSecondary).font(.system(size: 13))
                        }
                        Text(chat.lastMessage)
                            .foregroundStyle(Theme.ColorToken.textSecondary)
                            .lineLimit(1)
                            .font(.system(size: 14))
                    }
                    if chat.unreadCount > 0 {
                        Text("\(chat.unreadCount)")
                            .font(.system(size: 12, weight: .bold))
                            .padding(.vertical, 4).padding(.horizontal, 8)
                            .background(Capsule().fill(Theme.ColorToken.turquoise))
                            .foregroundStyle(.white)
                    }
                }
                .listRowBackground(Theme.ColorToken.white)
            }
        }
        .scrollContentBackground(.hidden)
        .background(Theme.ColorToken.milk)
        .navigationTitle("?????????")
    }
}

#Preview {
    ChatsScreen(
        vm: ChatsViewModel(service: MockChatService())
    )
}


// ===== File: DelegationApp/Features/Chats/ViewModel/ChatsViewModel.swift =====
import Foundation

final class ChatsViewModel: ObservableObject {
    @Published var chats: [ChatPreview] = []

    private let service: ChatService
    init(service: ChatService) {
        self.service = service
        self.chats = service.loadChats()
    }
}


// ===== File: DelegationApp/Features/Chats.swift =====
//
//  Chats.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//



// ===== File: DelegationApp/Features/Map/MapScreen/MapScreen.swift =====
//import SwiftUI
//
//// MARK: - ????? "?????"
//
///// ???????? ????? "?????": ?????, ???????, ?????, ????-??????.
//struct MapScreen: View {
```

```
//    @StateObject private var vm: MapViewModel
//    @State private var showCreate = false
//
//    /// ??????? ????? ??????????? ????? (???????? ????? / ????????).
//    private let mapMode: MapDisplayMode
//
//    init(
//        vm: MapViewModel,
//        mapMode: MapDisplayMode = MapDisplayConfig.defaultMode()
//    ) {
//        _vm = StateObject(wrappedValue: vm)
//        self.mapMode = mapMode
//    }
//
//    var body: some View {
//        VStack() {
//            searchBar
//            errorLabel
//            chipsRow
//            mapArea
//        }
//        .background(Theme.ColorToken.milk)
////        .navigationTitle("?????")
//
//    }
//
//    // MARK: - ??????
//
//    /// ????????? ??????.
//    private var searchBar: some View {
//        HStack(spacing: 8) {
//            Image(systemName: "magnifyingglass")
////                .foregroundColor(Theme.ColorToken.textSecondary)
//                .foregroundColor(Color.red)
//
//            TextField(
//                "??????? ?????",
//                text: $vm.searchText,
//                onCommit: vm.performSearch
//            )
//            .textFieldStyle(.plain)
//
//            if !vm.searchText.isEmpty {
//                Button {
//                    vm.searchText = ""
//                } label: {
//                    Image(systemName: "xmark.circle.fill")
//                        .foregroundColor(Theme.ColorToken.textSecondary)
//                        .imageScale(.medium)
//                }
//            }
//
//            Button(action: vm.performSearch) {
//                Text("?????")
//                    .font(.system(size: 15, weight: .semibold))
//            }
//        }
////        .padding(.horizontal, 16)
//        .background(Color.blue)
//        .softCardShadow()
////        .ignoresSafeArea()
//    }
//
//    /// ????????? ?? ?????? ?????? (???? ????).
//    private var errorLabel: some View {
//        Group {
//            if let message = vm.errorMessage {
```

```
//                Text(message)
//                    .font(.caption)
//                    .foregroundColor(.red)
//                    .frame(maxWidth: .infinity, alignment: .leading)
//                    .padding(.horizontal, 16)
//                    .padding(.top, 4)
//            }
//        }
//    }
//
//    /// ?????????????? ?????? ?????-????????.
//    private var chipsRow: some View {
//        ScrollView(.horizontal, showsIndicators: false) {
//            HStack(spacing: Theme.Spacing.m) {
//                ForEach(vm.chips, id: \.self) { chip in
//                    FilterChip(
//                        title: chip,
//                        isSelected: Binding(
//                            get: { vm.selected.contains(chip) },
//                            set: { isOn in
//                                if isOn {
//                                    vm.selected.insert(chip)
//                                } else {
//                                    vm.selected.remove(chip)
//                                }
//                            }
//                        )
//                    )
//                }
//            }
//            .padding(.horizontal)
//            .padding(.vertical, 8)
//        }
//        .background(Color.green)
//    }
//
//    /// ??????? ????? + ????????? ??????.
//    private var mapArea: some View {
//        ZStack(alignment: .bottom) {
//            MapCanvasView(centerPoint: $vm.centerPoint, mode: mapMode)
//        }
//    }
//}
//
//// MARK: - Preview
////#Preview {
////    let service = MockTaskService()
////    let vm = MapViewModel(service: service)
////    MapScreen(vm: vm, mapMode: .placeholder)
////}
//
//
//// MapCanvasView.swift
//// iCuno test
//

//
//  MapScreen.swift
//  iCuno test
//
//  ????? "?????": ?????, ???????-???? ? ???? ?????.
//

import SwiftUI

// MARK: - ????? "?????"
```

```
struct MapScreen: View {
    @StateObject private var vm: MapViewModel
    @State private var showCreate = false

    /// ????? ??????????? ????? (????????? ????? / ???????????).
    private let mapMode: MapDisplayMode

    init(
        vm: MapViewModel,
        mapMode: MapDisplayMode = MapDisplayConfig.defaultMode()
    ) {
        _vm = StateObject(wrappedValue: vm)
        self.mapMode = mapMode
    }

    var body: some View {
        ZStack(alignment: .top) {
            // ?????? ???? ? ??????-????? ?? ???? ?????
            mapArea
//                .background(Color.green)
//                .cornerRadius(15)

            // ??????? ???? ? ????? + ?????? + ?????
            VStack(spacing: 5) {
                // ????????? ?????? ?? ??????-????
                Spacer().frame(height: 50)

                searchBar
//                    .background(Color.red)
                    .background(Color.clear)
                    .cornerRadius(15)
                errorLabel
                    .background(Color.clear)
                    .cornerRadius(15)
                chipsRow
                    .background(Color.clear)
                    .cornerRadius(15)

                Spacer()
            }
            .padding(.horizontal, 16)
            .padding(.top, 8)
            .ignoresSafeArea()

        }
    }

    // MARK: - ??????

    /// ????????? ??????.
    private var searchBar: some View {
        HStack(spacing: 8) {
            Image(systemName: "magnifyingglass")
                .foregroundColor(Theme.ColorToken.textSecondary)

            TextField(
                "??????? ?????",
                text: $vm.searchText,
                onCommit: vm.performSearch
            )
            .textFieldStyle(.plain)

            if !vm.searchText.isEmpty {
                Button {
                    vm.searchText = ""
                } label: {
                    Image(systemName: "xmark.circle.fill")
```

```
                        .foregroundColor(Theme.ColorToken.textSecondary)
                        .imageScale(.medium)
                }
            }

            Button(action: vm.performSearch) {
                Text("?????")
                    .font(.system(size: 15, weight: .semibold))
            }
        }
        .padding(.horizontal, 12)
        .padding(.vertical, 10)
        // ???????? ?? «??????????» ???? ?????? ?????
        .background(.ultraThinMaterial)
        .clipShape(RoundedRectangle(cornerRadius: 16, style: .continuous))
        .softCardShadow()
    }

    /// ????????? ?? ?????? (???? ????).
    private var errorLabel: some View {
        Group {
            if let message = vm.errorMessage {
                Text(message)
                    .font(.caption)
                    .foregroundColor(.red)
                    .frame(maxWidth: .infinity, alignment: .leading)
            }
        }
    }

    /// ?????????????? ?????? ????????-??????.
    private var chipsRow: some View {
        ScrollView(.horizontal, showsIndicators: false) {
            HStack(spacing: Theme.Spacing.m) {
                ForEach(vm.chips, id: \.self) { chip in
                    FilterChip(
                        title: chip,
                        isSelected: Binding(
                            get: { vm.selected.contains(chip) },
                            set: { isOn in
                                if isOn {
                                    vm.selected.insert(chip)
                                } else {
                                    vm.selected.remove(chip)
                                }
                            }
                        )
                    )
                }
            }
            .padding(0)
        }
        // ?????? ?????????: ??? .background(Color.green)
        // ??? ?????????? ? ????? "?????" ??? ??????
    }

    /// ???? ? ??????.
    private var mapArea: some View {
        MapCanvasView(centerPoint: $vm.centerPoint, mode: mapMode)
            .ignoresSafeArea(edges: .top) // ????? ??? ???? ???????? ? ??? ?????????? ????
    }
}


// ===== File: DelegationApp/Features/Map/MapViewModel/MapViewModel.swift =====
//
// MapScreen.swift
// iCuno test
//
// Created by maftuna murtaza??? on 07.11.2025.
//

import SwiftUI
import YandexMapsMobile
import Foundation

// MARK: - ?????? ??????????? ?????

/// ????? ??????????? ?????.
/// - `.real` ? ????????? ????? ???????.
/// - `.placeholder` ? ???????? (??? ?????? / ????? ????? ????????? ??????).
enum MapDisplayMode {
    case real
    case placeholder
}

/// ???????????????? ???????????? ?????.
///
/// ????? ???? ?????, ??? ?? ???????, ??? ?????? ????????????:
/// ????????? ????? ??? ????????.
enum MapDisplayConfig {

    /// ???????? ???????, ??????? ??????, ? ????? ?????? ?????????? ?????.
    ///
    /// ????? ???????? ??????????, ???????? ????? DEBUG/RELEASE,
    /// ????????? ??????? ? ?.?.
    static func defaultMode() -> MapDisplayMode {
        // ??????:
        // ? DEBUG ????? ??????? ????????, ????? ????? ?? ?????? ???????? UI.
        // ? RELEASE ? ???????? ?????.
        #if DEBUG
        return .placeholder
        #else
        return .real
        #endif
    }
}

// MARK: - ????? ?????

/// ???, ??????? ???????? ?????? ?? "????? ?????":
/// ??? ???????? ? ???????? ???????? ????? ??? ????????.
///
/// ?????: ?????? ??????/?????????/??????? ???????? ?????????? ?? ????? ??????,
/// ?????? ??? ??? ????? ? `MapViewModel` ? ????????.
/// ??????? ??? ??????/?????????.
/// ??? ?? ????? ?? ??? ???????, ?? ??? ????? ? ?????? ??? ??, ??? ????????.
struct MapCanvasView: View {

    /// ?????, ?? ??????? ???????????? ?????.
    @Binding var centerPoint: YMKPoint?

    /// ??????? ????? ??????????? (???????? ????? / ????????).
    let mode: MapDisplayMode

    var body: some View {
        Group {
            switch mode {
            case .real:
                // ????? ????? ???????.
                YandexMapView(centerPoint: $centerPoint)


            case .placeholder:
                // ???????? ??? ?????? / ?????? ??? UI.
                Rectangle()
                    .fill(Theme.ColorToken.milk)
                    .overlay(
                        VStack(spacing: 8) {
                            Image(systemName: "map")
                                .font(.system(size: 32))
                                .foregroundColor(Theme.ColorToken.textSecondary)

                            Text("Map placeholder")
                                .font(.system(size: 14, weight: .medium))
                                .foregroundColor(Theme.ColorToken.textSecondary)
                        }
                    )

            }
        }
    }
}

/// ??????? ???????? ?????? ?????.
struct MapPlaceholderView: View {
    var body: some View {
        Rectangle()
            .fill(Theme.ColorToken.milk)
            .overlay(
                VStack(spacing: 8) {
                    Image(systemName: "map")
                        .font(.system(size: 32))
                        .foregroundColor(Theme.ColorToken.textSecondary)
                    Text("Map placeholder")
                        .font(.system(size: 14, weight: .medium))
                        .foregroundColor(Theme.ColorToken.textSecondary)
                }
            )
    }
}

// MARK: - ViewModel ?????

/// ViewModel ??? ?????? ?????: ???????, ??????, ????? ??????, ????? ?????.
final class MapViewModel: ObservableObject {

    // MARK: - ??????? (????)

    @Published var chips: [String] = [
        "??????", "?????????", "???????",
        "??????", "?????????", "??????"
    ]

    @Published var selected: Set<String> = []

    // MARK: - ?????? ?????

    @Published var tasks: [TaskItem] = []

    // MARK: - ????? ? ?????

    /// ????? ? ???? ?????? ??????.
    @Published var searchText: String = ""

    /// ??????? ?????, ?? ??????? ???????????? ?????.
    @Published var centerPoint: YMKPoint?

    /// ????????? ?? ?????? (????????, "?????? ?? ???????").
    @Published var errorMessage: String?
    private let service: TaskService
    private let searchService: AddressSearchService

    init(
        service: TaskService,
        searchService: AddressSearchService = AddressSearchService()
    ) {
        self.service = service
        self.searchService = searchService

        // ????????? ?????? ?????????? (??? ? ??????).
        self.tasks = service.loadNearbyTasks()

        // ????????? ????? ????? ? ?????? (?????? ???????? ?? ?????????).
        self.centerPoint = YMKPoint(
            latitude: 55.751244,
            longitude: 37.618423
        )
    }

    // MARK: - ?????? ????????

    func toggle(_ chip: String) {
        if selected.contains(chip) {
            selected.remove(chip)
        } else {
            selected.insert(chip)
        }
    }

    // MARK: - ????? ??????

    /// ????????? ????? ?? ?????? ? ???????? ?????.
    ///
    /// ?????: ???? ??? ???????? ???? ?????, ????? ?? UI ???????????? ????????.
    /// ?????? ?? ????? ??????????? ???? ?????, ?? `centerPoint` ?????????,
    /// ? ??? ????????? ????????? ????? ?? ????? ??????? ?????????? ?????.
    func performSearch() {
        let query = searchText.trimmingCharacters(in: .whitespacesAndNewlines)
        guard !query.isEmpty else {
            // ?????? ?????? ? ?????? ?????????? ??????.
            errorMessage = nil
            return
        }

        searchService.searchAddress(query) { [weak self] point in
            DispatchQueue.main.async {
                guard let self else { return }

                if let point {
                    // ?????: ?????????? ????? ? ???? ?????.
                    self.centerPoint = point
                    self.errorMessage = nil
                } else {
                    // ?????? ?? ?????.
                    self.errorMessage = "?????? ?? ???????"
                }
            }
        }
    }
}


// ===== File: DelegationApp/Features/Profile/View/ProfileScreen.swift =====
import SwiftUI

struct ProfileScreen: View {
    @StateObject var vm: ProfileViewModel
    init(vm: ProfileViewModel) { _vm = StateObject(wrappedValue: vm) }

    var body: some View {
        ScrollView {
            VStack(spacing: Theme.Spacing.l) {
                header
                settings
                support
                reviews
            }
            .padding(.bottom, 32)
        }
        .background(Theme.ColorToken.milk)
        .navigationTitle("???????")
        .toolbar(.hidden, for: .navigationBar)
    }

    private var header: some View {
        VStack(alignment: .leading, spacing: 12) {
            HStack(alignment: .center, spacing: 14) {
                Circle().fill(Theme.ColorToken.milk).frame(width: 56, height: 56)
                    .overlay(Image(systemName: "person.fill").font(.system(size: 26)).foregroundStyle(Theme.ColorToken

                VStack(alignment: .leading, spacing: 6) {
                    Text(vm.profile.name).font(.system(size: 20, weight: .semibold))
                    Text(vm.profile.phone).foregroundStyle(Theme.ColorToken.textSecondary)
                        .font(.system(size: 14))
                }
                Spacer()
                Text("ID")
                    .font(.system(size: 13, weight: .bold))
                    .padding(.vertical, 6).padding(.horizontal, 10)
                    .background(RoundedRectangle(cornerRadius: 10).fill(Theme.ColorToken.peach.opacity(0.3)))
            }

            HStack(spacing: 28) {
                VStack(alignment: .leading) {
                    HStack(spacing: 6) {
                        Image(systemName: "star.fill").foregroundStyle(Theme.ColorToken.peach)
                        Text("\(vm.profile.rating, specifier: "%.1f")")
                            .font(.system(size: 16, weight: .semibold))
                    }
                    Text("???????").foregroundStyle(Theme.ColorToken.textSecondary).font(.system(size: 12))
                }
                VStack(alignment: .leading) {
                    Text("\(vm.profile.completed)").font(.system(size: 16, weight: .semibold))
                    Text("?????????").foregroundStyle(Theme.ColorToken.textSecondary).font(.system(size: 12))
                }
                VStack(alignment: .leading) {
                    Text("\(vm.profile.cancelled)").font(.system(size: 16, weight: .semibold))
                    Text("????????").foregroundStyle(Theme.ColorToken.textSecondary).font(.system(size: 12))
                }
                Spacer()
            }
        }
        .padding()
        .background(LinearGradient(colors: [Theme.ColorToken.turquoise.opacity(0.85), Theme.ColorToken.turquoise],
                                   startPoint: .topLeading, endPoint: .bottomTrailing))
        .foregroundStyle(.white)
        .clipShape(RoundedRectangle(cornerRadius: Theme.Radius.xl, style: .continuous))
        .padding(.horizontal)
        .padding(.top, 12)
        .softCardShadow()
    }

    private var settings: some View {
        SectionBox(title: "?????????") {
            ToggleRow(title: "?????? ????", isOn: $vm.darkMode)
            NavRow(title: "???????????")
            NavRow(title: "??????? ? ???????")
        }
    }

    private var support: some View {
        SectionBox(title: "?????????") {
            NavRow(title: "??????")
            NavRow(title: "??????? ? ???????")
        }
    }

    private var reviews: some View {
        SectionBox(title: "??????") {
            ForEach(vm.reviews) { r in
                HStack(alignment: .top, spacing: 12) {
                    Circle().fill(Theme.ColorToken.milk).frame(width: 40, height: 40)
                        .overlay(Text(r.authorInitial).font(.system(size: 16, weight: .bold)))
                    VStack(alignment: .leading, spacing: 6) {
                        HStack {
                            Text(r.authorName).font(.system(size: 15, weight: .semibold))
                            StarsView(rating: Double(r.stars))
                            Spacer()
                        }
                        Text(r.text).font(.system(size: 14)).fixedSize(horizontal: false, vertical: true)
                        Text(r.ago).font(.system(size: 12)).foregroundStyle(Theme.ColorToken.textSecondary)
                    }
                    Spacer(minLength: 0)
                }
                .padding(.vertical, 8)
                .padding(.horizontal, 8)
            }
            Button("?????????? ??? ??????") { }
                .font(.system(size: 15, weight: .semibold))
                .frame(maxWidth: .infinity, alignment: .leading)
                .padding(.top, 6)
                .tint(Theme.ColorToken.turquoise)
                .padding()
        }
    }
}

private struct SectionBox<Content: View>: View {
    let title: String
    @ViewBuilder var content: Content

    var body: some View {
        VStack(alignment: .leading, spacing: 8) {
            Text(title).font(.system(size: 12, weight: .bold))
                .foregroundStyle(Theme.ColorToken.textSecondary)
                .padding(.horizontal)
            VStack(spacing: 0) { content }
                .background(RoundedRectangle(cornerRadius: Theme.Radius.l).fill(Theme.ColorToken.white))
                .softCardShadow()
                .padding(.horizontal)
        }
        .padding(.top, 4)
    }
}

private struct ToggleRow: View {
    let title: String
    @Binding var isOn: Bool
    var body: some View {
        HStack {
            Label(title, systemImage: "moon.fill")
                .labelStyle(.titleAndIcon)
            Spacer()
            Toggle("", isOn: $isOn).labelsHidden()
        }
        .padding()
        .background(Color.clear)
    }
}

private struct NavRow: View {
    let title: String
    var body: some View {
        HStack {
            Text(title)
            Spacer()
            Image(systemName: "chevron.right").foregroundStyle(Theme.ColorToken.textSecondary)
        }
        .padding()
    }
}



#Preview {
    let service = MockProfileService()
    let vm = ProfileViewModel(service: service)
    ProfileScreen(vm: vm)
}

//@StateObject var vm: ProfileViewModel
//init(vm: ProfileViewModel) { _vm = StateObject(wrappedValue: vm) }


// ===== File: DelegationApp/Features/Profile/ViewModel/ProfileViewModel.swift =====
import Foundation

final class ProfileViewModel: ObservableObject {
    @Published var profile: Profile
    @Published var reviews: [Review]
    @Published var darkMode: Bool = false

    private let service: ProfileService
    init(service: ProfileService) {
        self.service = service
        self.profile = service.loadProfile()
        self.reviews = service.loadReviews()
    }
}


// ===== File: DelegationApp/Features/Route/View/RouteScreen.swift =====
import SwiftUI

enum PreviewData {
    static let container = AppContainer.preview

    static let chatsVM  = ChatsViewModel(service: MockChatService())
    static let mapVM    = MapViewModel(service: MockTaskService())
    static let routeVM  = RouteViewModel(service: MockTaskService())
    static let profileVM = ProfileViewModel(service: MockProfileService())
}

struct RouteScreen: View {
    @StateObject var vm: RouteViewModel
    init(vm: RouteViewModel) { _vm = StateObject(wrappedValue: vm) }
    var body: some View {
        ScrollView {
            VStack(spacing: Theme.Spacing.l) {
                VStack(spacing: Theme.Spacing.m) {
                    RouteRow(symbol: "a.circle.fill", text: vm.pointA)
                    RouteRow(symbol: "b.circle.fill", text: vm.pointB)
                    RouteRow(symbol: "clock.fill", text: vm.time)
                }
                .padding()
                .background(RoundedRectangle(cornerRadius: Theme.Radius.l)
                    .fill(Theme.ColorToken.white))
                .softCardShadow()
                .padding(.horizontal)

                HStack {
                    Image(systemName: "arrow.forward.circle")
                    Text("45 ??? · 12.5 ??")
                        .font(.system(size: 16, weight: .semibold))
                    Spacer()
                    Capsule()
                        .fill(Theme.ColorToken.milk)
                        .frame(width: 36, height: 28)
                        .overlay(Text("\(vm.tasks.count)").font(.system(size: 15, weight: .semibold)))
                }
                .padding()
                .background(RoundedRectangle(cornerRadius: Theme.Radius.l)
                    .fill(Theme.ColorToken.white))
                .softCardShadow()
                .padding(.horizontal)

                // ????? ????????
                RoundedRectangle(cornerRadius: Theme.Radius.l)
                    .fill(Theme.ColorToken.milk)
                    .frame(height: 220)
                    .overlay(Text("????? ? ?????????").foregroundStyle(Theme.ColorToken.textSecondary))
                    .padding(.horizontal)

                VStack(alignment: .leading, spacing: Theme.Spacing.m) {
                    Text("??????? ?? ????")
                        .font(.system(size: 18, weight: .semibold))
                    ForEach(vm.tasks) { t in
                        HStack {
                            VStack(alignment: .leading, spacing: 4) {
                                Text(t.title).font(.system(size: 16, weight: .semibold))
                                Text("~\(t.distanceKm, specifier: "%.1f") ?? ? \(t.etaMinutes) ???")
                                    .foregroundStyle(Theme.ColorToken.textSecondary)
                                    .font(.system(size: 13))
                            }
                            Spacer()
                            PriceTag(price: t.price, eta: t.etaMinutes)
                        }
                        .padding()
                        .background(RoundedRectangle(cornerRadius: Theme.Radius.m).fill(Theme.ColorToken.white))
                        .softCardShadow()
                    }
                }
                .padding(.horizontal)
                .padding(.bottom, 24)
            }
        }
        .navigationTitle("???????")
    }
}

private struct RouteRow: View {
    let symbol: String
    let text: String
    var body: some View {
        HStack(spacing: 12) {
            Image(systemName: symbol)
                .foregroundStyle(Theme.ColorToken.turquoise)
            Text(text)
            Spacer()
        }
        .font(.system(size: 16))
    }
}


#Preview("RouteScreen") {
    NavigationStack {
        RouteScreen(vm: PreviewData.routeVM)
    }
    .preferredColorScheme(.light)
}


// ===== File: DelegationApp/Features/Route/View/RouteView.swift =====
//
//  RouteView.swift
//  iCuno test
//
//  RootView ? ????????.
//

import SwiftUI

struct RootView: View {
    @EnvironmentObject var container: AppContainer
    @State private var selectedTab = 0

    var body: some View {
        TabView(selection: $selectedTab) {

            // ??????? "?????"
            NavigationStack {
//                MapScreen(vm: .init(service: container.taskService), mapMode: .placeholder)
            }
            .tabItem {
                Label("?????", systemImage: "map")
            }
            .tag(0)

            // ??????? "???????"
            NavigationStack {
//                RouteScreen(vm: .init(service: container.taskService))
            }
            .tabItem {
                Label("???????", systemImage: "point.topleft.down.curvedto.point.bottomright.up")
            }
            .tag(1)

            // ????? ??????? "??????????"
            NavigationStack {
//                MyAdsScreen()
            }
            .tabItem {
                Label("??????????", systemImage: "rectangle.stack.badge.plus")
            }
            .tag(2)

            // ??????? "????"
            NavigationStack {
//                ChatsScreen(vm: .init(service: container.chatService))
```

```
            }
            .tabItem {
                Label("????", systemImage: "bubble.left.and.bubble.right")
            }
            .tag(3)

            // ??????? "???????"
            NavigationStack {
//                ProfileScreen(vm: .init(service: container.profileService))
            }
            .tabItem {
                Label("???????", systemImage: "person.circle")
            }
            .tag(4)
        }
        .background(Color.black)
        .ignoresSafeArea()
//        .tint(Theme.ColorToken.turquoise)
        .cornerRadius(20)

//        .background(.ultraThinMaterial)
//        .clipShape(RoundedRectangle(cornerRadius: 16, style: .continuous))
        .softCardShadow()

    }
}


// ===== File: DelegationApp/Features/Route/ViewModel/RouteViewModel.swift =====
import Foundation

final class RouteViewModel: ObservableObject {
    @Published var pointA: String = "?????????? ???????"
    @Published var pointB: String = "??????? ??? ??????? ????????"
    @Published var time: String = "17:00"
    @Published var tasks: [TaskItem] = []

    private let service: TaskService
    init(service: TaskService) {
        self.service = service
        self.tasks = service.loadRouteTasks()
    }
}


// ===== File: DelegationApp/Features/Untitled.swift =====
//import SwiftUI
//
//extension Color {
//    static func hex(_ hex: String) -> Color {
//        let hex = hex.trimmingCharacters(in: CharacterSet.alphanumerics.inverted)
//        var int: UInt64 = 0; Scanner(string: hex).scanHexInt64(&int)
//        let a, r, g, b: UInt64
//        switch hex.count {
//        case 3: (a,r,g,b) = (255, (int >> 8) * 17, (int >> 4 & 0xF) * 17, (int & 0xF) * 17)
//        case 6: (a,r,g,b) = (255, int >> 16, int >> 8 & 0xFF, int & 0xFF)
//        case 8: (a,r,g,b) = (int >> 24, int >> 16 & 0xFF, int >> 8 & 0xFF, int & 0xFF)
//        default:(a,r,g,b) = (255,0,0,0)
//        }
//        return Color(.sRGB,
//                     red: Double(r)/255, green: Double(g)/255,
//                     blue: Double(b)/255, opacity: Double(a)/255)
//    }
//}


// ===== File: DelegationApp/Others/YandexMapConfigurator.swift =====
import YandexMapsMobile

/// ???????????????? ????????? Yandex MapKit.
enum YandexMapConfigurator {
    private static var isConfigured = false

    static func configureIfNeeded() {
        // ? SwiftUI Preview ?????? ?? ?????????????? SDK.
//        if RuntimeEnvironment.isPreview { return }
        guard !isConfigured else { return }

        // ? ???? ???? ???????? ????
        YMKMapKit.setApiKey("df3f9145-2080-42b7-9b91-b879c34236bb")
        YMKMapKit.sharedInstance()
        isConfigured = true
    }
}


// ===== File: DelegationApp/RootView.swift =====
//import SwiftUI
//
//struct RootView: View {
//    /// DI-????????? ? ?????????
//    @EnvironmentObject var container: AppContainer
//    @State private var selected = 0
//
//    var body: some View {
//        TabView(selection: $selected) {
//
//            // ? ??????? ?????
//            NavigationStack {
//                MapScreen(vm: MapViewModel(service: container.taskService))
//            }
//            .tabItem {
//                Label("?????", systemImage: "map")
//            }
//            .tag(0)
//
//            // ? ??????? ???????
//            NavigationStack {
//                RouteScreen(vm: RouteViewModel(service: container.taskService))
//            }
//            .tabItem {
//                Label("???????", systemImage: "point.topleft.down.curvedto.point.bottomright.up")
//            }
//            .tag(1)
//
//            // ? ??????? ????
//            NavigationStack {
//                ChatsScreen(vm: container.chatService)
//            }
//            .tabItem {
//                Label("????", systemImage: "bubble.left.and.bubble.right")
//            }
//            .tag(2)
//
//            // ? ??????? ???????
//            NavigationStack {
//                ProfileScreen(vm: ProfileViewModel(service: container.profileService))
//            }
//            .tabItem {
//                Label("???????", systemImage: "person")
//            }
//            .tag(3)
//        }
//        .tint(Theme.ColorToken.turquoise)
//        .background(Theme.ColorToken.milk)
//    }
//}
////
////#Preview {
////    RootView()
////        .environmentObject(AppContainer.preview)
////}


// ===== File: iCuno test/ContentView.swift =====
//
//  ContentView.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//







// ===== File: iCuno test/iCuno_testApp.swift =====
//
//  iCuno_testApp.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

//import SwiftUI
//
//@main
//struct iCuno_testApp: App {
//    var body: some Scene {
//        WindowGroup {
//            ContentView()
//        }
//    }
//}


// ===== File: iCuno testTests/iCuno_testTests.swift =====
//
//  iCuno_testTests.swift
//  iCuno testTests
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import XCTest
@testable import iCuno_test

final class iCuno_testTests: XCTestCase {

    override func setUpWithError() throws {
        // Put setup code here. This method is called before the invocation of each test method in the class.
    }

    override func tearDownWithError() throws {
        // Put teardown code here. This method is called after the invocation of each test method in the class.
    }

    func testExample() throws {
        // This is an example of a functional test case.
        // Use XCTAssert and related functions to verify your tests produce the correct results.
        // Any test you write for XCTest can be annotated as throws and async.
        // Mark your test throws to produce an unexpected failure when your test encounters an uncaught error.
        // Mark your test async to allow awaiting for asynchronous code to complete. Check the results with assertions
    }

    func testPerformanceExample() throws {
        // This is an example of a performance test case.
        self.measure {
            // Put the code you want to measure the time of here.
        }
    }

}


// ===== File: iCuno testUITests/iCuno_testUITests.swift =====
//
//  iCuno_testUITests.swift
//  iCuno testUITests
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import XCTest

final class iCuno_testUITests: XCTestCase {

    override func setUpWithError() throws {
        // Put setup code here. This method is called before the invocation of each test method in the class.

        // In UI tests it is usually best to stop immediately when a failure occurs.
        continueAfterFailure = false

        // In UI tests it?s important to set the initial state - such as interface orientation - required for your tes
    }

    override func tearDownWithError() throws {
        // Put teardown code here. This method is called after the invocation of each test method in the class.
    }

    @MainActor
    func testExample() throws {
        // UI tests must launch the application that they test.
        let app = XCUIApplication()
        app.launch()

        // Use XCTAssert and related functions to verify your tests produce the correct results.
    }

    @MainActor
    func testLaunchPerformance() throws {
        // This measures how long it takes to launch your application.
        measure(metrics: [XCTApplicationLaunchMetric()]) {
            XCUIApplication().launch()
        }
    }
}


// ===== File: iCuno testUITests/iCuno_testUITestsLaunchTests.swift =====
//
//  iCuno_testUITestsLaunchTests.swift
//  iCuno testUITests
//
//  Created by maftuna murtazaeva on 07.11.2025.
//
```

```
import XCTest

final class iCuno_testUITestsLaunchTests: XCTestCase {

    override class var runsForEachTargetApplicationUIConfiguration: Bool {
        true
    }

    override func setUpWithError() throws {
        continueAfterFailure = false
    }

    @MainActor
    func testLaunch() throws {
        let app = XCUIApplication()
        app.launch()

        // Insert steps here to perform after app launch but before taking a screenshot,
        // such as logging into a test account or navigating somewhere in the app

        let attachment = XCTAttachment(screenshot: app.screenshot())
        attachment.name = "Launch Screen"
        attachment.lifetime = .keepAlways
        add(attachment)
    }
}


// ===== File: temp_code_export/CombinedCode.swift =====
// ===== File: DelegationApp/App/AppContainer.swift =====
//
// AppContainer.swift
// iCuno test
//
// Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation
import SwiftUI

/// ??????? DI-?????????: ??????????? ???? ????????.
final class AppContainer: ObservableObject {
    let taskService: TaskService
    let chatService: ChatService
    let profileService: ProfileService

    init(
        taskService: TaskService,
        chatService: ChatService,
        profileService: ProfileService,
    ) {
        self.taskService = taskService
        self.chatService = chatService
        self.profileService = profileService
    }
}

extension AppContainer {
    /// ????????? ??? ??????/??????? ?????? ? ?????? ????????.
    static let preview = AppContainer(
        taskService: MockTaskService(),
        chatService: MockChatService(),
        profileService: MockProfileService()
    )
}
```

```
// ===== File: DelegationApp/App/AppRouter.swift =====
////
////  RouteView.swift
////  iCuno test
////
////  RootView ? ????????.
////
//
//import SwiftUI
//
//struct RootView: View {
//    @EnvironmentObject var container: AppContainer
//    @State private var selectedTab = 0
//
//    var body: some View {
//        TabView(selection: $selectedTab) {
//
//            // ??????? "?????"
//            NavigationStack {
//                MapScreen(vm: .init(service: container.taskService))
//            }
//            .tabItem {
//                Label("?????", systemImage: "map")
//            }
//            .tag(0)
//
//            // ??????? "???????"
//            NavigationStack {
//                RouteScreen(vm: .init(service: container.taskService))
//            }
//            .tabItem {
//                Label("???????", systemImage: "point.topleft.down.curvedto.point.bottomright.up")
//            }
//            .tag(1)
//
//            // ????? ??????? "??????????"
//            NavigationStack {
//                MyAdsScreen()
//            }
//            .tabItem {
//                Label("??????????", systemImage: "rectangle.stack.badge.plus")
//            }
//            .tag(2)
//
//            // ??????? "????"
//            NavigationStack {
//                ChatsScreen(vm: .init(service: container.chatService))
//            }
//            .tabItem {
//                Label("????", systemImage: "bubble.left.and.bubble.right")
//            }
//            .tag(3)
//
//            // ??????? "???????"
//            NavigationStack {
//                ProfileScreen(vm: .init(service: container.profileService))
//            }
//            .tabItem {
//                Label("???????", systemImage: "person.circle")
//            }
//            .tag(4)
//        }
//        .tint(Theme.ColorToken.turquoise)
//        .background(Color.black)
//    }
//}
```

```
// ===== File: DelegationApp/App/DelegationApp.swift =====
import SwiftUI
import YandexMapsMobile
//
//@main
//struct DelegationApp: App {
//    @StateObject private var container = AppContainer.preview
//
//    init() {
//        // ? ????????????? Yandex MapKit
//        // ???? ??????? ???? ????, ??? ??, ??? ?? ??? ???????????? ? ???????? ???????
//        YMKMapKit.setApiKey("df3f9145-2080-42b7-9b91-b879c34236bb")
//        YMKMapKit.sharedInstance()
//    }
//
////    var body: some Scene {
////        WindowGroup {
////            RootView()
////                .environmentObject(container)
////        }
////    }
//    let service = MockTaskService()
//    let searchService = AddressSearchService()
//    let vm = MapViewModel(service: service, searchService: searchService)
//
//    var body: some Scene {
//        WindowGroup {
//            MapScreen(vm: vm)
//        }
//    }
//}
//
////#Preview {
////    let service = MockTaskService()
////    let searchService = AddressSearchService()
////    let vm = MapViewModel(service: service, searchService: searchService)
////    MapScreen(vm: vm)
////}

//@main
//struct DelegationApp: App {
//    @StateObject private var container = AppContainer.preview
//    @StateObject private var mapVM: MapViewModel
//
//    init() {
//        // ????????????? Yandex MapKit
//        YMKMapKit.setApiKey("df3f9145-2080-42b7-9b91-b879c34236bb")
//        YMKMapKit.sharedInstance()
//
//        let service = MockTaskService()
//        let searchService = AddressSearchService()
//        _mapVM = StateObject(wrappedValue: MapViewModel(service: service,
//                                                        searchService: searchService))
//    }
//
//    var body: some Scene {
//        WindowGroup {
////            MapScreen(vm: mapVM)
////            ChatsScreen(
////                vm: ChatsViewModel(service: MockChatService())
////            )
//
//
////            NavigationStack {
////                RouteScreen(vm: PreviewData.routeVM)
////            }
////            .preferredColorScheme(.light)
//
//            let service = MockTaskService()
//            let searchService = AddressSearchService()
//            let vm = MapViewModel(service: service, searchService: searchService)
//
//            return NavigationStack {
//                MapScreen(vm: vm)
//            }
//        }
//    }
//}

@main
struct DelegationApp: App {
    /// ????? ????????? ????????????.
    @StateObject private var container = AppContainer.preview

    init() {
        // ??? ??????? ????? ????????????? ??????? ???????????????? SDK ????:
        // YandexMapConfigurator.configureIfNeeded()
        //
        // ?? ?????? ??? ?????? ??? ??? ????? ? ??????? (AddressSearchService / YandexMapView),
        // ??? ??? ????? ????? ?? ??????? ?? ??????????? Map SDK.
    }

    var body: some Scene {
        WindowGroup {
            RootView()
                .environmentObject(container)
                .ignoresSafeArea()
        }
    }
}

/// ???????, ????? ????????, ??? ??? ??????????? ? SwiftUI Preview.
enum RuntimeEnvironment {
    static var isPreview: Bool {
        #if DEBUG
        if ProcessInfo.processInfo.environment["XCODE_RUNNING_FOR_PREVIEWS"] == "1" {
            return true
        }
        #endif
        return false
    }
}


// ===== File: DelegationApp/App/YandexMapView.swift =====
import SwiftUI
import YandexMapsMobile

/// ??????? ??? YMKMapView ??? ????????????? ? SwiftUI.
///
/// ?????: ? ?????? ?? ?? ??????? ???????? ????? ??????,
/// ????? ?? ????? SwiftUI Preview.
struct YandexMapView: UIViewRepresentable {

    /// ?????????? ?????? ?????.
    @Binding var centerPoint: YMKPoint?

    final class Coordinator {
        var mapView: YMKMapView?
        var placemark: YMKPlacemarkMapObject?
    }

    func makeCoordinator() -> Coordinator {
        Coordinator()
    }

    func makeUIView(context: Context) -> UIView {
        // ?????????, ? ??????? ??? ??????? ??????? ??????? YMKMapView.
        let container = UIView()
        container.backgroundColor = .clear
//        // ? ?????? ? ?????? ?? ?????????, ?????? ?????? UIView.
//        guard !RuntimeEnvironment.isPreview else {
//            return container
//        }

        // ? ??????? ??????? ?????????????? SDK ? ?????.
        YandexMapConfigurator.configureIfNeeded()

        let mapView = YMKMapView(frame: .zero)
        mapView!.translatesAutoresizingMaskIntoConstraints = false
        container.addSubview(mapView!)

        NSLayoutConstraint.activate([
            mapView!.topAnchor.constraint(equalTo: container.topAnchor),
            mapView!.bottomAnchor.constraint(equalTo: container.bottomAnchor),
            mapView!.leadingAnchor.constraint(equalTo: container.leadingAnchor),
            mapView!.trailingAnchor.constraint(equalTo: container.trailingAnchor)
        ])

        context.coordinator.mapView = mapView

        // ????????? ?????.
        let startPoint = centerPoint ?? YMKPoint(
            latitude: 55.751244,
            longitude: 37.618423
        )
        updateMap(on: mapView!, coordinator: context.coordinator, to: startPoint)

        return container
    }

    func updateUIView(_ uiView: UIView, context: Context) {
        guard
//            !RuntimeEnvironment.isPreview,
            let mapView = context.coordinator.mapView,
            let point = centerPoint
        else { return }

        updateMap(on: mapView, coordinator: context.coordinator, to: point)
    }

    // MARK: - Internal helpers

    private func updateMap(
        on mapView: YMKMapView,
        coordinator: Coordinator,
        to point: YMKPoint
    ) {
        let map = mapView.mapWindow.map
        let position = YMKCameraPosition(
            target: point,
            zoom: 15,
            azimuth: 0,
            tilt: 0
        )
        let animation = YMKAnimation(type: .smooth, duration: 1.0)
        map.move(with: position, animation: animation, cameraCallback: nil)

        let mapObjects = map.mapObjects
        if let oldPlacemark = coordinator.placemark {
            mapObjects.remove(with: oldPlacemark)
        }
        let placemark = mapObjects.addPlacemark(with: point)
        coordinator.placemark = placemark
    }
}


// ===== File: DelegationApp/Core/Components/FilterChip.swift =====
//
//  FilterChip.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import SwiftUI

struct FilterChip: View {
    let title: String
    @Binding var isSelected: Bool

    var body: some View {
        Button {
            isSelected.toggle()
        } label: {
            HStack(spacing: 8) {
                if isSelected { Image(systemName: "checkmark") }
                Text(title)
                    .font(.system(size: 15, weight: .semibold))
            }
            .padding(.vertical, 10)
            .padding(.horizontal, 14)
            .background(
                RoundedRectangle(cornerRadius: Theme.Radius.l, style: .continuous)
                    .fill(isSelected ? Theme.ColorToken.turquoise : Theme.ColorToken.milk)
            )
            .foregroundStyle(isSelected ? Color.white : Theme.ColorToken.textPrimary)
            .softCardShadow()
        }
        .buttonStyle(.plain)
    }
}


// ===== File: DelegationApp/Core/Components/FloatingPlusButton.swift =====
//
//  FloatingPlusButton.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import SwiftUI

struct FloatingPlusButton: View {
    var action: () -> Void
    var body: some View {
        Button(action: action) {
            Image(systemName: "plus")
                .font(.system(size: 24, weight: .bold))
                .foregroundStyle(Color.white)
                .frame(width: 64, height: 64)
                .background(Circle().fill(Theme.ColorToken.turquoise))
                .softCardShadow()
        }
        .buttonStyle(.plain)
        .accessibilityLabel("???????")
    }
}


// ===== File: DelegationApp/Core/Components/PriceTag.swift =====
//
//  PriceTag.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import SwiftUI

struct PriceTag: View {
    let price: Int
    let eta: Int
    var isHighlighted: Bool = false

    var body: some View {
        VStack(spacing: 4) {
            Text("\(price) ?")
                .font(.system(size: 16, weight: .semibold))
            Text("\(eta) ???")
                .font(.system(size: 12, weight: .regular))
                .foregroundStyle(Theme.ColorToken.textSecondary)
        }
        .padding(.horizontal, 16)
        .padding(.vertical, 10)
        .background(
            RoundedRectangle(cornerRadius: Theme.Radius.l, style: .continuous)
                .fill(Theme.ColorToken.white)
                .overlay(
                    RoundedRectangle(cornerRadius: Theme.Radius.l)
                        .stroke(isHighlighted ? Theme.ColorToken.turquoise : Color.clear, lineWidth: 2)
                )
        )
        .softCardShadow()
    }
}


// ===== File: DelegationApp/Core/Components/StarsView.swift =====
//
//  StarsView.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import SwiftUI

struct StarsView: View {
    let rating: Double
    let max: Int = 5

    var body: some View {
        HStack(spacing: 4) {
            ForEach(0..<max, id: \.self) { idx in
                let filled = rating >= Double(idx + 1) - 0.001
                Image(systemName: filled ? "star.fill" : "star")
                    .foregroundStyle(filled ? Theme.ColorToken.peach : Theme.ColorToken.textSecondary)
            }
        }
    }
}
```

```
// ===== File: DelegationApp/Core/Models/AdModels.swift =====
//
//  AdModels.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 24.11.2025.
//

//
//  AdModels.swift
//  iCuno test
//
//  ??????? ??? ?????? ??????????.
//

import Foundation

/// ?????? ??????????. ???? ???????????? ?????? ??? ???-??????
/// ?? ?????? "??? ??????????".
struct AdItem: Identifiable {
    let id: UUID = .init()
    let title: String
    let priceDescription: String
    let isExpired: Bool
    let views: Int
    let responses: Int
    let favorites: Int
}


// ===== File: DelegationApp/Core/Models/ChatModels.swift =====
import Foundation

struct ChatPreview: Identifiable {
    let id: UUID = .init()
    let initials: String
    let name: String
    let lastMessage: String
    let time: String
    let unreadCount: Int
}


// ===== File: DelegationApp/Core/Models/ProfileModels.swift =====
//
//  ProfileModels.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

struct Profile {
    let name: String
    let phone: String
    let rating: Double
    let completed: Int
    let cancelled: Int
}

struct Review: Identifiable {
    let id: UUID = .init()
    let authorInitial: String
    let authorName: String
    let text: String
    let ago: String
    let stars: Int
}


// ===== File: DelegationApp/Core/Models/TaskModels.swift =====
//
//  TaskModels.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

struct TaskItem: Identifiable {
    let id: UUID = .init()
    let title: String
    let price: Int        // ?
    let etaMinutes: Int   // ???
    let distanceKm: Double
}


// ===== File: DelegationApp/Core/Services/ChatService.swift =====
//
//  ChatService.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

protocol ChatService {
    func loadChats() -> [ChatPreview]
}


// ===== File: DelegationApp/Core/Services/Mock/MockChatService.swift =====
import Foundation

final class MockChatService: ChatService {
    func loadChats() -> [ChatPreview] {
        [
            .init(initials: "?", name: "?????? ?.", lastMessage: "???????, ???!", time: "14:30", unreadCount: 2),
            .init(initials: "?", name: "???? ?.", lastMessage: "???? ????? ??? ???????", time: "?????", unreadCount: 0
        ]
    }
}


// ===== File: DelegationApp/Core/Services/Mock/MockProfileService.swift =====
//
//  MockProfileService.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

final class MockProfileService: ProfileService {
    func loadProfile() -> Profile {
        .init(name: "??????? ??????",
              phone: "+7 999 123-45-67",
              rating: 4.9,
              completed: 127,
              cancelled: 3)
    }
    func loadReviews() -> [Review] {
        [
            .init(authorInitial: "?", authorName: "????? ?.",
                  text: "???????? ???????????! ??? ?????? ?????? ? ???????????. ??????????!",
                  ago: "2 ??? ?????", stars: 5),
            .init(authorInitial: "?", authorName: "??????? ?.",
                  text: "????? ???????! ??????? ?????? ?????, ??? ?????????.",
                  ago: "?????? ?????", stars: 5)
        ]
    }
}


// ===== File: DelegationApp/Core/Services/Mock/MockTaskService.swift =====
//
//  MockTaskService.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

final class MockTaskService: TaskService {
    func loadNearbyTasks() -> [TaskItem] {
        [
            .init(title: "?????? ??????", price: 200, etaMinutes: 14, distanceKm: 1.1),
            .init(title: "??????? ???????", price: 400, etaMinutes: 10, distanceKm: 2.0),
            .init(title: "????????? ?????", price: 500, etaMinutes: 18, distanceKm: 3.5),
            .init(title: "?????? ???????", price: 250, etaMinutes: 7, distanceKm: 0.6)
        ]
    }
    func loadRouteTasks() -> [TaskItem] {
        [
            .init(title: "?????????? ??????", price: 350, etaMinutes: 8, distanceKm: 0.9),
            .init(title: "?????? ????", price: 150, etaMinutes: 12, distanceKm: 0.5)
        ]
    }
}


// ===== File: DelegationApp/Core/Services/Networking/APIClient.swift =====
//
//  APIClient.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//



// ===== File: DelegationApp/Core/Services/Networking/AddressSearchService.swift =====
// AddressSearchService.swift
// iCuno test / DelegationApp

import Foundation
import YandexMapsMobile

final class AddressSearchService {

    private let searchManager: YMKSearchManager?
    private var searchSession: YMKSearchSession?
    private let isEnabled: Bool
    init() {
//        // ? ?????? ????????? ??????.
//        if RuntimeEnvironment.isPreview {
//            self.searchManager = nil
//            self.isEnabled = false
//            return
//        }

        YandexMapConfigurator.configureIfNeeded()

        let managerType: YMKSearchManagerType = .combined
        let search = YMKSearch.sharedInstance()
        self.searchManager = search?.createSearchManager(with: managerType)
        self.isEnabled = (self.searchManager != nil)
    }

    func searchAddress(
        _ text: String,
        completion: @escaping (YMKPoint?) -> Void
    ) {
        let trimmed = text.trimmingCharacters(in: .whitespacesAndNewlines)
        guard !trimmed.isEmpty else {
            completion(nil)
            return
        }

        // ? ?????? ?????? ?????? ?? ????.
        guard isEnabled, let searchManager else {
            completion(nil)
            return
        }

        let bbox = YMKBoundingBox(
            southWest: YMKPoint(latitude: -85.0, longitude: -180.0),
            northEast: YMKPoint(latitude: 85.0, longitude: 180.0)
        )
        let geometry = YMKGeometry(boundingBox: bbox)

        let options = YMKSearchOptions()
        options.geometry = true

        searchSession = searchManager.submit(
            withText: trimmed,
            geometry: geometry,
            searchOptions: options
        ) { [weak self] response, error in
            defer { self?.searchSession = nil }

            if let error {
                print("Search error: \(error)")
                completion(nil)
                return
            }

            guard
                let collection = response?.collection,
                let firstItem = collection.children.first,
                let obj = firstItem.obj,
                let point = obj.geometry.first?.point
            else {
                completion(nil)
                return
            }

            completion(point)
        }
    }
}


// ===== File: DelegationApp/Core/Services/Networking/Endpoints.swift =====
//
//  Endpoints.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//



// ===== File: DelegationApp/Core/Services/ProfileService.swift =====
//
//  ProfileService.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

protocol ProfileService {
    func loadProfile() -> Profile
    func loadReviews() -> [Review]
}

protocol AddAnnouncementService {

}


// ===== File: DelegationApp/Core/Services/TaskService.swift =====
//
//  TaskService.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import Foundation

protocol TaskService {
    func loadNearbyTasks() -> [TaskItem]
    func loadRouteTasks() -> [TaskItem]
}


// ===== File: DelegationApp/Core/Theme/Theme.swift =====
//
//  Theme.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//

import SwiftUI

enum Theme {
    enum ColorToken {
        static let turquoise = Color.hex("#3CC8C4")
        static let white     = Color.hex("#FFFFFF")
        static let milk      = Color.hex("#F7F3E9")
        static let peach     = Color.hex("#FFC9A6")
        static let textPrimary   = Color.black.opacity(0.9)
        static let textSecondary = Color.black.opacity(0.6)
        static let shadow = Color.black.opacity(0.08)
    }

    enum Radius {
        static let s: CGFloat = 10
        static let m: CGFloat = 16
        static let l: CGFloat = 24
        static let xl: CGFloat = 28
    }

    enum Spacing {
        static let xs: CGFloat = 6
        static let s: CGFloat = 8
        static let m: CGFloat = 12
        static let l: CGFloat = 16
        static let xl: CGFloat = 20
        static let xxl: CGFloat = 24
    }

    enum Shadow {
        static let soft = ShadowStyle(radius: 16, y: 8, opacity: 0.10)
        struct ShadowStyle {
            let radius: CGFloat
            let y: CGFloat
            let opacity: Double
        }
    }
}

extension View {
    /// ?????? ????????? ???? ??? iOS
    func softCardShadow() -> some View {
        shadow(color: Theme.ColorToken.shadow, radius: Theme.Shadow.soft.radius, x: 0, y: Theme.Shadow.soft.y)
    }
}


// ===== File: DelegationApp/Core/Utils/Extentions/Color+Hex.swift =====
//
//  Color+Hex.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 08.11.2025.
//

import SwiftUI

extension Color {
    static func hex(_ hex: String) -> Color {
        let hex = hex.trimmingCharacters(in: CharacterSet.alphanumerics.inverted)
        var int: UInt64 = 0; Scanner(string: hex).scanHexInt64(&int)
        let a, r, g, b: UInt64
        switch hex.count {
        case 3: (a,r,g,b) = (255, (int >> 8) * 17, (int >> 4 & 0xF) * 17, (int & 0xF) * 17)
        case 6: (a,r,g,b) = (255, int >> 16, int >> 8 & 0xFF, int & 0xFF)
        case 8: (a,r,g,b) = (int >> 24, int >> 16 & 0xFF, int >> 8 & 0xFF, int & 0xFF)
        default:(a,r,g,b) = (255,0,0,0)
        }
        return Color(.sRGB,
                     red: Double(r)/255, green: Double(g)/255,
                     blue: Double(b)/255, opacity: Double(a)/255)
    }
}


// ===== File: DelegationApp/Features/Ads/AdsView/AdsScreen.swift =====
//
//  AdsScreen.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 24.11.2025.
//

//
//  MyAdsScreen.swift
//  iCuno test
//
//  ????? "??? ??????????" ?? ??????? ?????.
//

import SwiftUI

/// ????? "??? ??????????".
struct MyAdsScreen: View {
    @State private var selectedFilter: AdsFilter = .waiting
    @State private var showNewAdSheet = false

    // ???? ??? ???????. ????? ????? ???????? ??????? ???????.
    private let ads: [AdItem] = [
        .init(
            title: "?????? ? ??????????? ????????",
            priceDescription: "?? 300 ? ?? ??????",
            isExpired: true,
            views: 58,
            responses: 1,
            favorites: 3
        )
    ]

    var body: some View {
        ZStack(alignment: .bottom) {
            ScrollView {
                VStack(alignment: .leading, spacing: Theme.Spacing.l) {
                    summarySection
                    filtersSection
//                    promoSection
                    expiredSection
                }
                .padding(.horizontal, Theme.Spacing.l)
                .padding(.top, Theme.Spacing.m)
                // ????? ????? ??? ?????? ??????
            }


            newAdButton
        }
        .sheet(isPresented: $showNewAdSheet) {
            NewAdCategoryScreen()
        }
        .navigationTitle("??? ??????????")
        .navigationBarTitleDisplayMode(.inline)
    }

    // MARK: - ?????????

    /// ??????? ? ????? ???????? ?????? ??????.
    private var summarySection: some View {
        HStack(spacing: Theme.Spacing.m) {
            SmallSummaryCard(
                title: "?????? ? ?????",
                subtitle: "????????? ??? ????????????",
                gradient: LinearGradient(
                    colors: [Color.hex("#B6FAC3"), Color.hex("#84A4FA")],
                    startPoint: .topLeading,
                    endPoint: .bottomTrailing
                )
            )

            SmallSummaryCard(
                title: "29 990 ?",
                subtitle: "??????????",
                gradient: LinearGradient(
                    colors: [Color.hex("#0D47A1"), Color.hex("#1976D2")],
                    startPoint: .topLeading,
                    endPoint: .bottomTrailing
                )
            )
        }
    }

    /// ??????? "???? ???????? / ???????? / ?????????".
    private var filtersSection: some View {
        HStack(spacing: Theme.Spacing.l) {
            ForEach(AdsFilter.allCases) { filter in
                VStack(spacing: 4) {
                    Text(filter.titleWithCount)
                        .font(.system(size: 15, weight: .semibold))
                        .foregroundStyle(filter == selectedFilter ? Theme.ColorToken.turquoise : Color.gray)

                    Rectangle()
                        .fill(filter == selectedFilter ? Theme.ColorToken.turquoise : Color.clear)
                        .frame(height: 3)
                        .cornerRadius(1.5)
                }
                .onTapGesture {
                    selectedFilter = filter
                }
            }
        }
        .padding(.top, Theme.Spacing.l)
    }

//    /// ????? ?????-???????? "?? 25% ?????? ??????".
//    private var promoSection: some View {
//        RoundedRectangle(cornerRadius: Theme.Radius.l, style: .continuous)
//            .fill(
//                LinearGradient(
//                    colors: [Color.hex("#0046A5"), Color.hex("#0059D6")],
//                    startPoint: .topLeading,
//                    endPoint: .bottomTrailing
//                )
//            )
//            .overlay(alignment: .leading) {
//                VStack(alignment: .leading, spacing: 4) {
//                    Text("?? 25% ?????? ??????")
//                        .font(.system(size: 16, weight: .semibold))
//                    Text("?????????? ??????????? ?????? ?? ??????? ?????????? ???????")
//                        .font(.system(size: 13))
//                        .fixedSize(horizontal: false, vertical: true)
//                }
//                .foregroundStyle(Color.gray)
//                .padding(16)
//            }
//            .frame(maxWidth: .infinity)
//    }

    /// ?????? ? ?????????? "????? ???? ??????????" ? ?????????? ??????????.
    private var expiredSection: some View {
        VStack(alignment: .leading, spacing: Theme.Spacing.m) {
//            Text("????? ???? ??????????")
//                .font(.system(size: 15, weight: .semibold))
//                .foregroundStyle(Color.white)

            ForEach(ads.filter { $0.isExpired }) { ad in
                AdCardView(ad: ad)
            }
        }
        .padding(.top, Theme.Spacing.l)
    }

    /// ?????? ??????? ?????? "?????????? ??????????".
    private var newAdButton: some View {
        Button {
            showNewAdSheet = true
        } label: {
            Text("?????????? ??????????")
                .font(.system(size: 17, weight: .semibold))
                .foregroundStyle(Color.white)
                .frame(maxWidth: .infinity)
                .padding(.vertical, 14)
                .background(
                    RoundedRectangle(cornerRadius: 18, style: .continuous)
                        .fill(Color.black.opacity(0.7))
                )
                .padding(.horizontal, Theme.Spacing.l)

        }
        .buttonStyle(.plain)
    }
}

// MARK: - ??????????????? ???? ? ??????

/// ??? ??????? ? ??????? ????????.
private enum AdsFilter: CaseIterable, Identifiable {
    case waiting
    case active
    case drafts

    var id: Self { self }

    var title: String {
        switch self {
        case .waiting: return "???? ????????"
        case .active:  return "????????"
        case .drafts:  return "?????????"
        }
    }

    /// ??? ??????? ???????????? ?? ?? ?????, ??? ? ?? ??????.
    var count: Int {
        switch self {
        case .waiting: return 1
        case .active:  return 1
        case .drafts:  return 0
        }
    }

    var titleWithCount: String {
        "\(title) \(count)"
    }
}

/// ????????? ???????? ?????? ("?????? ? ?????" / "29 990 ?...").
private struct SmallSummaryCard: View {
    let title: String
    let subtitle: String
    let gradient: LinearGradient

    var body: some View {
        ZStack(alignment: .leading) {
            RoundedRectangle(cornerRadius: Theme.Radius.l, style: .continuous)
                .fill(gradient)

            VStack(alignment: .leading, spacing: 4) {
                Text(title)
                    .font(.system(size: 15, weight: .semibold))
                Text(subtitle)
                    .font(.system(size: 12))
                    .foregroundStyle(Color.black.opacity(0.8))
            }
            .foregroundStyle(Color.black)
            .padding(12)
        }
        .frame(maxWidth: .infinity, minHeight: 72)
    }
}

/// ???????? ?????? ??????????.
private struct AdCardView: View {
    let ad: AdItem

    var body: some View {
        VStack(alignment: .leading) {
            HStack(alignment: .top, spacing: 12) {
                RoundedRectangle(cornerRadius: 12, style: .continuous)
                    .fill(Color.gray.opacity(0.4))
                    .frame(width: 96, height: 72)
                    .overlay(
                        Image(systemName: "photo")
                            .font(.system(size: 24))
                            .foregroundStyle(Color.white.opacity(0.7))
                    )

                VStack(alignment: .leading, spacing: 4) {
                    Text(ad.title)
                        .font(.system(size: 16, weight: .semibold))
                        .foregroundStyle(Color.gray)
                        .lineLimit(2)

                    Text(ad.priceDescription)
                        .font(.system(size: 15, weight: .semibold))
                        .foregroundStyle(Color.gray)

                    Text("????? ???? ??????????")
                        .font(.system(size: 13))
                        .foregroundStyle(Color.gray)
                }

                Spacer()

                Image(systemName: "pencil")
                    .font(.system(size: 16, weight: .semibold))
                    .foregroundStyle(Color.gray.opacity(0.8))
            }

            HStack(spacing: 12) {
                IconCounterView(systemName: "eye", text: "\(ad.views)")
                IconCounterView(systemName: "person", text: "\(ad.responses)")
                IconCounterView(systemName: "heart", text: "\(ad.favorites)")
            }
            .font(.system(size: 13))
            .foregroundStyle(Color.gray)
            .padding(.vertical, 10)
        }
        .padding(12)
        .background(Color.secondary.opacity(0.1))
        .cornerRadius(15)
//        .background(
//            RoundedRectangle(cornerRadius: Theme.Radius.l, style: .continuous)
//                .fill(Color.white.opacity(0.5))
//        )
    }
}

/// ?????????? "?????? + ?????", "????????? + ?????" ? ?.?.
private struct IconCounterView: View {
    let systemName: String
    let text: String

    var body: some View {
        HStack(spacing: 4) {
            Image(systemName: systemName)
            Text(text)
        }
    }
}

//#Preview("MyAdsScreen") {
//    NavigationStack {
//        MyAdsScreen()
//    }
//    .preferredColorScheme(.dark)
//}


// ===== File: DelegationApp/Features/Ads/AdsView/NewAdCategoryScreen.swift =====
//
//  NewAdCategoryScreen.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 24.11.2025.
//

//
//  NewAdCategoryScreen.swift
//  iCuno test
//
//  ????? "????? ??????????" (????? ?????????).
//

import SwiftUI

/// ????? ?????? ????????? ??? ?????? ??????????.
struct NewAdCategoryScreen: View {
    @Environment(\.dismiss) private var dismiss

    private let categories: [AdCategory] = [
        .init(title: "???????? ?? ????", systemImage: "car.fill"),
        .init(title: "?????? ?????????", systemImage: "building.2.fill"),
        .init(title: "?????? ???????????", systemImage: "briefcase.fill"),
        .init(title: "?????? ??????", systemImage: "scissors"),
        .init(title: "???????????????? ??????", systemImage: "swift")
    ]

    var body: some View {
        NavigationStack {
            VStack(alignment: .leading, spacing: 0) {
                header
                categoriesList
                Spacer()
            }
        }
    }

    // MARK: - ???????

    private var header: some View {
        VStack(alignment: .leading, spacing: 16) {
            HStack {
                Button {
                    dismiss()
                } label: {
                    Image(systemName: "xmark")
                        .font(.system(size: 18, weight: .semibold))
                        .foregroundStyle(Color.gray)
                        .padding(8)
                }

                Spacer()
            }

            Text("????? ??????????")
                .font(.system(size: 24, weight: .bold))
                .foregroundStyle(Color.white)
        }
        .padding(.horizontal, Theme.Spacing.l)
        .padding(.top, Theme.Spacing.m)
        .padding(.bottom, Theme.Spacing.l)
    }

    private var categoriesList: some View {
        VStack(spacing: 0) {
            ForEach(categories) { category in
                Button {
                    // ???? ?????? ????????? ?????.
                    // ????? ????? ????? ????? ????????? ????? ???????? ??????????.
                    dismiss()
                } label: {
                    HStack(spacing: 12) {
                        Image(systemName: category.systemImage)
                            .font(.system(size: 20))
                            .frame(width: 28, height: 28)
                            .foregroundStyle(Color.white)

                        Text(category.title)
                            .font(.system(size: 17))
                            .foregroundStyle(Color.gray)

                        Spacer()

                        Image(systemName: "chevron.right")
                            .font(.system(size: 15, weight: .semibold))
                            .foregroundStyle(Color.gray)
                    }
                    .padding(.horizontal, Theme.Spacing.l)
                    .padding(.vertical, 14)
                }
                .buttonStyle(.plain)

                Divider()
                    .background(Color.gray.opacity(0.6))
                    .padding(.leading, Theme.Spacing.l + 28 + 12)
            }
        }
    }
}
```

```
private struct AdCategory: Identifiable {
    let id: UUID = .init()
    let title: String
    let systemImage: String
}

//#Preview("NewAdCategoryScreen") {
//    NewAdCategoryScreen()
//        .preferredColorScheme(.dark)
//}


// ===== File: DelegationApp/Features/Chats/View/ChatsScreen.swift =====
import SwiftUI

struct ChatsScreen: View {
    @StateObject var vm: ChatsViewModel
    init(vm: ChatService) { _vm = StateObject(wrappedValue: .init(service: vm)) }
    init(vm: ChatsViewModel) { _vm = StateObject(wrappedValue: vm) }

    var body: some View {
        List {
            ForEach(vm.chats) { chat in
                HStack(spacing: 12) {
                    Circle()
                        .fill(LinearGradient(colors: [Theme.ColorToken.turquoise, Theme.ColorToken.peach],
                                             startPoint: .topLeading, endPoint: .bottomTrailing))
                        .frame(width: 44, height: 44)
                        .overlay(Text(chat.initials).foregroundStyle(.white).font(.system(size: 17, weight: .bold)))

                    VStack(alignment: .leading, spacing: 4) {
                        HStack {
                            Text(chat.name).font(.system(size: 16, weight: .semibold))
                            Spacer()
                            Text(chat.time).foregroundStyle(Theme.ColorToken.textSecondary).font(.system(size: 13))
                        }
                        Text(chat.lastMessage)
                            .foregroundStyle(Theme.ColorToken.textSecondary)
                            .lineLimit(1)
                            .font(.system(size: 14))
                    }
                    if chat.unreadCount > 0 {
                        Text("\(chat.unreadCount)")
                            .font(.system(size: 12, weight: .bold))
                            .padding(.vertical, 4).padding(.horizontal, 8)
                            .background(Capsule().fill(Theme.ColorToken.turquoise))
                            .foregroundStyle(.white)
                    }
                }
                .listRowBackground(Theme.ColorToken.white)
            }
        }
        .scrollContentBackground(.hidden)
        .background(Theme.ColorToken.milk)
        .navigationTitle("?????????")
    }
}

#Preview {
    ChatsScreen(
        vm: ChatsViewModel(service: MockChatService())
    )
}


// ===== File: DelegationApp/Features/Chats/ViewModel/ChatsViewModel.swift =====
import Foundation
```

```
final class ChatsViewModel: ObservableObject {
    @Published var chats: [ChatPreview] = []

    private let service: ChatService
    init(service: ChatService) {
        self.service = service
        self.chats = service.loadChats()
    }
}


// ===== File: DelegationApp/Features/Chats.swift =====
//
//  Chats.swift
//  iCuno test
//
//  Created by maftuna murtazaeva on 07.11.2025.
//



// ===== File: DelegationApp/Features/Map/MapScreen/MapScreen.swift =====
//import SwiftUI
//
//// MARK: - ????? "?????"
//
///// ???????? ????? "?????": ?????, ???????, ?????, ????-??????.
//struct MapScreen: View {
//    @StateObject private var vm: MapViewModel
//    @State private var showCreate = false
//
//    /// ??????? ????? ??????????? ????? (???????? ????? / ????????).
//    private let mapMode: MapDisplayMode
//
//    init(
//        vm: MapViewModel,
//        mapMode: MapDisplayMode = MapDisplayConfig.defaultMode()
//    ) {
//        _vm = StateObject(wrappedValue: vm)
//        self.mapMode = mapMode
//    }
//
//    var body: some View {
//        VStack() {
//            searchBar
//            errorLabel
//            chipsRow
//            mapArea
//        }
//        .background(Theme.ColorToken.milk)
////        .navigationTitle("?????")
//
//    }
//
//    // MARK: - ??????
//
//    /// ????????? ??????.
//    private var searchBar: some View {
//        HStack(spacing: 8) {
//            Image(systemName: "magnifyingglass")
////                .foregroundColor(Theme.ColorToken.textSecondary)
//                .foregroundColor(Color.red)
//
//            TextField(
//                "??????? ?????",
//                text: $vm.searchText,
//                onCommit: vm.performSearch
//            )
//            .textFieldStyle(.plain)
//
//            if !vm.searchText.isEmpty {
//                Button {
//                    vm.searchText = ""
//                } label: {
//                    Image(systemName: "xmark.circle.fill")
//                        .foregroundColor(Theme.ColorToken.textSecondary)
//                        .imageScale(.medium)
//                }
//            }
//
//            Button(action: vm.performSearch) {
//                Text("?????")
//                    .font(.system(size: 15, weight: .semibold))
//            }
//        }
////        .padding(.horizontal, 16)
//        .background(Color.blue)
//        .softCardShadow()
////        .ignoresSafeArea()
//    }
//
//    /// ????????? ?? ?????? ?????? (???? ????).
//    private var errorLabel: some View {
//        Group {
//            if let message = vm.errorMessage {
//                Text(message)
//                    .font(.caption)
//                    .foregroundColor(.red)
//                    .frame(maxWidth: .infinity, alignment: .leading)
//                    .padding(.horizontal, 16)
//                    .padding(.top, 4)
//            }
//        }
//    }
//
//    /// ?????????????? ?????? ?????-????????.
//    private var chipsRow: some View {
//        ScrollView(.horizontal, showsIndicators: false) {
//            HStack(spacing: Theme.Spacing.m) {
//                ForEach(vm.chips, id: \.self) { chip in
//                    FilterChip(
//                        title: chip,
//                        isSelected: Binding(
//                            get: { vm.selected.contains(chip) },
//                            set: { isOn in
//                                if isOn {
//                                    vm.selected.insert(chip)
//                                } else {
//                                    vm.selected.remove(chip)
//                                }
//                            }
//                        )
//                    )
//                }
//            }
//            .padding(.horizontal)
//            .padding(.vertical, 8)
//        }
//        .background(Color.green)
//    }
//
//    /// ??????? ????? + ????????? ??????.
//    private var mapArea: some View {
```

```
//        ZStack(alignment: .bottom) {
//            MapCanvasView(centerPoint: $vm.centerPoint, mode: mapMode)
//        }
//    }
//}
//
//// MARK: - Preview
////#Preview {
////    let service = MockTaskService()
////    let vm = MapViewModel(service: service)
////    MapScreen(vm: vm, mapMode: .placeholder)
////}
//
//
//// MapCanvasView.swift
//// iCuno test
//

//
//  MapScreen.swift
//  iCuno test
//
//  ????? "?????": ?????, ???????-???? ? ???? ?????.
//

import SwiftUI

// MARK: - ????? "?????"

struct MapScreen: View {
    @StateObject private var vm: MapViewModel
    @State private var showCreate = false

    /// ????? ??????????? ????? (????????? ????? / ???????????).
    private let mapMode: MapDisplayMode

    init(
        vm: MapViewModel,
        mapMode: MapDisplayMode = MapDisplayConfig.defaultMode()
    ) {
        _vm = StateObject(wrappedValue: vm)
        self.mapMode = mapMode
    }

    var body: some View {
        ZStack(alignment: .top) {
            // ?????? ???? ? ??????-????? ?? ???? ?????
            mapArea
//                .background(Color.green)
//                .cornerRadius(15)

            // ??????? ???? ? ????? + ?????? + ?????
            VStack(spacing: 5) {
                // ????????? ?????? ?? ??????-????
                Spacer().frame(height: 50)

                searchBar
//                    .background(Color.red)
                    .background(Color.clear)
                    .cornerRadius(15)
                errorLabel
                    .background(Color.clear)
                    .cornerRadius(15)
                chipsRow
                    .background(Color.clear)
                    .cornerRadius(15)
                Spacer()
            }
            .padding(.horizontal, 16)
            .padding(.top, 8)
            .ignoresSafeArea()

        }
    }

    // MARK: - ??????

    /// ????????? ??????.
    private var searchBar: some View {
        HStack(spacing: 8) {
            Image(systemName: "magnifyingglass")
                .foregroundColor(Theme.ColorToken.textSecondary)

            TextField(
                "??????? ?????",
                text: $vm.searchText,
                onCommit: vm.performSearch
            )
            .textFieldStyle(.plain)

            if !vm.searchText.isEmpty {
                Button {
                    vm.searchText = ""
                } label: {
                    Image(systemName: "xmark.circle.fill")
                        .foregroundColor(Theme.ColorToken.textSecondary)
                        .imageScale(.medium)
                }
            }

            Button(action: vm.performSearch) {
                Text("?????")
                    .font(.system(size: 15, weight: .semibold))
            }
        }
        .padding(.horizontal, 12)
        .padding(.vertical, 10)
        // ???????? ?? «??????????» ???? ?????? ?????
        .background(.ultraThinMaterial)
        .clipShape(RoundedRectangle(cornerRadius: 16, style: .continuous))
        .softCardShadow()
    }

    /// ????????? ?? ?????? (???? ????).
    private var errorLabel: some View {
        Group {
            if let message = vm.errorMessage {
                Text(message)
                    .font(.caption)
                    .foregroundColor(.red)
                    .frame(maxWidth: .infinity, alignment: .leading)
            }
        }
    }

    /// ?????????????? ?????? ????????-??????.
    private var chipsRow: some View {
        ScrollView(.horizontal, showsIndicators: false) {
            HStack(spacing: Theme.Spacing.m) {
                ForEach(vm.chips, id: \.self) { chip in
                    FilterChip(
                        title: chip,
                        isSelected: Binding(
                            get: { vm.selected.contains(chip) },
                            set: { isOn in
                                if isOn {
                                    vm.selected.insert(chip)
                                } else {
                                    vm.selected.remove(chip)
                                }
                            }
                        )
                    )
                }
            }
            .padding(0)
        }
        // ?????? ?????????: ??? .background(Color.green)
        // ??? ?????????? ? ????? "?????" ??? ??????
    }

    /// ???? ? ??????.
    private var mapArea: some View {
        MapCanvasView(centerPoint: $vm.centerPoint, mode: mapMode)
            .ignoresSafeArea(edges: .top) // ????? ??? ???? ???????? ? ??? ?????????? ????
    }
}


// ===== File: DelegationApp/Features/Map/MapViewModel/MapViewModel.swift =====
//
// MapScreen.swift
// iCuno test
//
// Created by maftuna murtaza??? on 07.11.2025.
//

import SwiftUI
import YandexMapsMobile
import Foundation

// MARK: - ?????? ??????????? ?????

/// ????? ??????????? ?????.
/// - `.real` ? ????????? ????? ???????.
/// - `.placeholder` ? ???????? (??? ?????? / ????? ????? ????????? ??????).
enum MapDisplayMode {
    case real
    case placeholder
}

/// ???????????????? ???????????? ?????.
///
/// ????? ???? ?????, ??? ?? ???????, ??? ?????? ????????????:
/// ????????? ????? ??? ????????.
enum MapDisplayConfig {

    /// ???????? ???????, ??????? ??????, ? ????? ?????? ?????????? ?????.
    ///
    /// ????? ???????? ??????????, ???????? ????? DEBUG/RELEASE,
    /// ????????? ??????? ? ?.?.
    static func defaultMode() -> MapDisplayMode {
        // ??????:
        // ? DEBUG ????? ??????? ????????, ????? ????? ?? ?????? ???????? UI.
        // ? RELEASE ? ???????? ?????.
        #if DEBUG
        return .placeholder
        #else
        return .real
        #endif
    }
}

// MARK: - ????? ?????

/// ???, ??????? ???????? ?????? ?? "????? ?????":
/// ??? ???????? ? ???????? ???????? ????? ??? ????????.
///
/// ?????: ?????? ??????/?????????/??????? ???????? ?????????? ?? ????? ??????,
/// ?????? ??? ??? ????? ? `MapViewModel` ? ????????.
/// ??????? ??? ??????/?????????.
/// ??? ?? ????? ?? ??? ???????, ?? ??? ????? ? ?????? ??? ??, ??? ????????.
struct MapCanvasView: View {

    /// ?????, ?? ??????? ???????????? ?????.
    @Binding var centerPoint: YMKPoint?

    /// ??????? ????? ??????????? (???????? ????? / ????????).
    let mode: MapDisplayMode

    var body: some View {
        Group {
            switch mode {
            case .real:
                // ????? ????? ???????.
                YandexMapView(centerPoint: $centerPoint)


            case .placeholder:
                // ???????? ??? ?????? / ?????? ??? UI.
                Rectangle()
                    .fill(Theme.ColorToken.milk)
                    .overlay(
                        VStack(spacing: 8) {
                            Image(systemName: "map")
                                .font(.system(size: 32))
                                .foregroundColor(Theme.ColorToken.textSecondary)

                            Text("Map placeholder")
                                .font(.system(size: 14, weight: .medium))
                                .foregroundColor(Theme.ColorToken.textSecondary)
                        }
                    )

            }
        }
    }
}

/// ??????? ???????? ?????? ?????.
struct MapPlaceholderView: View {
    var body: some View {
        Rectangle()
            .fill(Theme.ColorToken.milk)
            .overlay(
                VStack(spacing: 8) {
                    Image(systemName: "map")
                        .font(.system(size: 32))
                        .foregroundColor(Theme.ColorToken.textSecondary)
                    Text("Map placeholder")
                        .font(.system(size: 14, weight: .medium))
                        .foregroundColor(Theme.ColorToken.textSecondary)
                }
            )
    }
}
```

```
// MARK: - ViewModel ?????

/// ViewModel ??? ?????? ?????: ???????, ??????, ????? ??????, ????? ?????.
final class MapViewModel: ObservableObject {

    // MARK: - ??????? (????)

    @Published var chips: [String] = [
        "??????", "?????????", "???????",
        "??????", "?????????", "??????"
    ]

    @Published var selected: Set<String> = []

    // MARK: - ?????? ?????

    @Published var tasks: [TaskItem] = []

    // MARK: - ????? ? ?????

    /// ????? ? ???? ?????? ??????.
    @Published var searchText: String = ""

    /// ??????? ?????, ?? ??????? ???????????? ?????.
    @Published var centerPoint: YMKPoint?

    /// ????????? ?? ?????? (????????, "?????? ?? ???????").
    @Published var errorMessage: String?

    private let service: TaskService
    private let searchService: AddressSearchService

    init(
        service: TaskService,
        searchService: AddressSearchService = AddressSearchService()
    ) {
        self.service = service
        self.searchService = searchService

        // ????????? ?????? ?????????? (??? ? ??????).
        self.tasks = service.loadNearbyTasks()

        // ????????? ????? ????? ? ?????? (?????? ???????? ?? ?????????).
        self.centerPoint = YMKPoint(
            latitude: 55.751244,
            longitude: 37.618423
        )
    }

    // MARK: - ?????? ????????

    func toggle(_ chip: String) {
        if selected.contains(chip) {
            selected.remove(chip)
        } else {
            selected.insert(chip)
        }
    }

    // MARK: - ????? ??????

    /// ????????? ????? ?? ?????? ? ???????? ?????.
    ///
    /// ?????: ???? ??? ???????? ???? ?????, ????? ?? UI ???????????? ????????.
    /// ?????? ?? ????? ??????????? ???? ?????, ?? `centerPoint` ?????????,
    /// ? ??? ????????? ????????? ????? ?? ????? ??????? ?????????? ?????.
    func performSearch() {
        let query = searchText.trimmingCharacters(in: .whitespacesAndNewlines)
        guard !query.isEmpty else {
            // ?????? ?????? ? ?????? ?????????? ??????.
            errorMessage = nil
            return
        }

        searchService.searchAddress(query) { [weak self] point in
            DispatchQueue.main.async {
                guard let self else { return }

                if let point {
                    // ?????: ?????????? ????? ? ???? ?????.
                    self.centerPoint = point
                    self.errorMessage = nil
                } else {
                    // ?????? ?? ?????.
                    self.errorMessage = "?????? ?? ???????"
                }
            }
        }
    }
}


// ===== File: DelegationApp/Features/Profile/View/ProfileScreen.swift =====
import SwiftUI

struct ProfileScreen: View {
    @StateObject var vm: ProfileViewModel
    init(vm: ProfileViewModel) { _vm = StateObject(wrappedValue: vm) }

    var body: some View {
        ScrollView {
            VStack(spacing: Theme.Spacing.l) {
                header
                settings
                support
                reviews
            }
            .padding(.bottom, 32)
        }
        .background(Theme.ColorToken.milk)
        .navigationTitle("???????")
        .toolbar(.hidden, for: .navigationBar)
    }

    private var header: some View {
        VStack(alignment: .leading, spacing: 12) {
            HStack(alignment: .center, spacing: 14) {
                Circle().fill(Theme.ColorToken.milk).frame(width: 56, height: 56)
                    .overlay(Image(systemName: "person.fill").font(.system(size: 26)).foregroundStyle(Theme.ColorToken

                VStack(alignment: .leading, spacing: 6) {
                    Text(vm.profile.name).font(.system(size: 20, weight: .semibold))
                    Text(vm.profile.phone).foregroundStyle(Theme.ColorToken.textSecondary)
                        .font(.system(size: 14))
                }
                Spacer()
                Text("ID")
                    .font(.system(size: 13, weight: .bold))
                    .padding(.vertical, 6).padding(.horizontal, 10)
                    .background(RoundedRectangle(cornerRadius: 10).fill(Theme.ColorToken.peach.opacity(0.3)))
            }

            HStack(spacing: 28) {
                VStack(alignment: .leading) {
                    HStack(spacing: 6) {
                        Image(systemName: "star.fill").foregroundStyle(Theme.ColorToken.peach)
                        Text("\(vm.profile.rating, specifier: "%.1f")")
                            .font(.system(size: 16, weight: .semibold))
                    }
                    Text("???????").foregroundStyle(Theme.ColorToken.textSecondary).font(.system(size: 12))
                }
                VStack(alignment: .leading) {
                    Text("\(vm.profile.completed)").font(.system(size: 16, weight: .semibold))
                    Text("?????????").foregroundStyle(Theme.ColorToken.textSecondary).font(.system(size: 12))
                }
                VStack(alignment: .leading) {
                    Text("\(vm.profile.cancelled)").font(.system(size: 16, weight: .semibold))
                    Text("????????").foregroundStyle(Theme.ColorToken.textSecondary).font(.system(size: 12))
                }
                Spacer()
            }
        }
        .padding()
        .background(LinearGradient(colors: [Theme.ColorToken.turquoise.opacity(0.85), Theme.ColorToken.turquoise],
                                   startPoint: .topLeading, endPoint: .bottomTrailing))
        .foregroundStyle(.white)
        .clipShape(RoundedRectangle(cornerRadius: Theme.Radius.xl, style: .continuous))
        .padding(.horizontal)
        .padding(.top, 12)
        .softCardShadow()
    }

    private var settings: some View {
        SectionBox(title: "?????????") {
            ToggleRow(title: "?????? ????", isOn: $vm.darkMode)
            NavRow(title: "???????????")
            NavRow(title: "??????? ? ???????")
        }
    }

    private var support: some View {
        SectionBox(title: "?????????") {
            NavRow(title: "??????")
            NavRow(title: "??????? ? ???????")
        }
    }

    private var reviews: some View {
        SectionBox(title: "??????") {
            ForEach(vm.reviews) { r in
                HStack(alignment: .top, spacing: 12) {
                    Circle().fill(Theme.ColorToken.milk).frame(width: 40, height: 40)
                        .overlay(Text(r.authorInitial).font(.system(size: 16, weight: .bold)))
                    VStack(alignment: .leading, spacing: 6) {
                        HStack {
                            Text(r.authorName).font(.system(size: 15, weight: .semibold))
                            StarsView(rating: Double(r.stars))
                            Spacer()
                        }
                        Text(r.text).font(.system(size: 14)).fixedSize(horizontal: false, vertical: true)
                        Text(r.ago).font(.system(size: 12)).foregroundStyle(Theme.ColorToken.textSecondary)
                    }
                    Spacer(minLength: 0)
                }
                .padding(.vertical, 8)
                .padding(.horizontal, 8)
            }
            Button("?????????? ??? ??????") { }
                .font(.system(size: 15, weight: .semibold))
                .frame(maxWidth: .infinity, alignment: .leading)
                .padding(.top, 6)
                .tint(Theme.ColorToken.turquoise)
                .padding()
        }
    }
}

private struct SectionBox<Content: View>: View {
    let title: String
    @ViewBuilder var content: Content

    var body: some View {
        VStack(alignment: .leading, spacing: 8) {
            Text(title).font(.system(size: 12, weight: .bold))
                .foregroundStyle(Theme.ColorToken.textSecondary)
                .padding(.horizontal)
            VStack(spacing: 0) { content }
                .background(RoundedRectangle(cornerRadius: Theme.Radius.l).fill(Theme.ColorToken.white))
                .softCardShadow()
                .padding(.horizontal)
        }
        .padding(.top, 4)
    }
}

private struct ToggleRow: View {
    let title: String
    @Binding var isOn: Bool
    var body: some View {
        HStack {
            Label(title, systemImage: "moon.fill")
                .labelStyle(.titleAndIcon)
            Spacer()
            Toggle("", isOn: $isOn).labelsHidden()
        }
        .padding()
        .background(Color.clear)
    }
}

private struct NavRow: View {
    let title: String
    var body: some View {
        HStack {
            Text(title)
            Spacer()
            Image(systemName: "chevron.right").foregroundStyle(Theme.ColorToken.textSecondary)
        }
        .padding()
    }
}



#Preview {
    let service = MockProfileService()
    let vm = ProfileViewModel(service: service)
    ProfileScreen(vm: vm)
}

//@StateObject var vm: ProfileViewModel
//init(vm: ProfileViewModel) { _vm = StateObject(wrappedValue: vm) }


// ===== File: DelegationApp/Features/Profile/ViewModel/ProfileViewModel.swift =====
import Foundation

final class ProfileViewModel: ObservableObject {
    @Published var profile: Profile
    @Published var reviews: [Review]
    @Published var darkMode: Bool = false

    private let service: ProfileService
    init(service: ProfileService) {
        self.service = service
        self.profile = service.loadProfile()
        self.reviews = service.loadReviews()
    }
}


// ===== File: DelegationApp/Features/Route/View/RouteScreen.swift =====
import SwiftUI

enum PreviewData {
    static let container = AppContainer.preview

    static let chatsVM  = ChatsViewModel(service: MockChatService())
    static let mapVM    = MapViewModel(service: MockTaskService())
    static let routeVM  = RouteViewModel(service: MockTaskService())
    static let profileVM = ProfileViewModel(service: MockProfileService())
}

struct RouteScreen: View {
    @StateObject var vm: RouteViewModel
    init(vm: RouteViewModel) { _vm = StateObject(wrappedValue: vm) }

    var body: some View {
        ScrollView {
            VStack(spacing: Theme.Spacing.l) {
                VStack(spacing: Theme.Spacing.m) {
                    RouteRow(symbol: "a.circle.fill", text: vm.pointA)
                    RouteRow(symbol: "b.circle.fill", text: vm.pointB)
                    RouteRow(symbol: "clock.fill", text: vm.time)
                }
                .padding()
                .background(RoundedRectangle(cornerRadius: Theme.Radius.l)
                    .fill(Theme.ColorToken.white))
                .softCardShadow()
                .padding(.horizontal)

                HStack {
                    Image(systemName: "arrow.forward.circle")
                    Text("45 ??? · 12.5 ??")
                        .font(.system(size: 16, weight: .semibold))
                    Spacer()
                    Capsule()
                        .fill(Theme.ColorToken.milk)
                        .frame(width: 36, height: 28)
                        .overlay(Text("\(vm.tasks.count)").font(.system(size: 15, weight: .semibold)))
                }
                .padding()
                .background(RoundedRectangle(cornerRadius: Theme.Radius.l)
                    .fill(Theme.ColorToken.white))
                .softCardShadow()
                .padding(.horizontal)

                // ????? ????????
                RoundedRectangle(cornerRadius: Theme.Radius.l)
                    .fill(Theme.ColorToken.milk)
                    .frame(height: 220)
                    .overlay(Text("????? ? ?????????").foregroundStyle(Theme.ColorToken.textSecondary))
                    .padding(.horizontal)

                VStack(alignment: .leading, spacing: Theme.Spacing.m) {
                    Text("??????? ?? ????")
                        .font(.system(size: 18, weight: .semibold))
                    ForEach(vm.tasks) { t in
                        HStack {
                            VStack(alignment: .leading, spacing: 4) {
                                Text(t.title).font(.system(size: 16, weight: .semibold))
                                Text("~\(t.distanceKm, specifier: "%.1f") ?? ? \(t.etaMinutes) ???")
                                    .foregroundStyle(Theme.ColorToken.textSecondary)
                                    .font(.system(size: 13))
                            }
                            Spacer()
                            PriceTag(price: t.price, eta: t.etaMinutes)
                        }
                        .padding()
                        .background(RoundedRectangle(cornerRadius: Theme.Radius.m).fill(Theme.ColorToken.white))
                        .softCardShadow()
                    }
                }
                .padding(.horizontal)
                .padding(.bottom, 24)
            }
        }
        .navigationTitle("???????")
    }
}

private struct RouteRow: View {
    let symbol: String
    let text: String
    var body: some View {
        HStack(spacing: 12) {
            Image(systemName: symbol)
                .foregroundStyle(Theme.ColorToken.turquoise)
            Text(text)
            Spacer()
        }
        .font(.system(size: 16))
    }
}


#Preview("RouteScreen") {
    NavigationStack {
        RouteScreen(vm: PreviewData.routeVM)
    }
    .preferredColorScheme(.light)
}


// ===== File: DelegationApp/Features/Route/View/RouteView.swift =====
//
//  RouteView.swift
//  iCuno test
//
//  RootView ? ????????.
//

import SwiftUI

struct RootView: View {
    @EnvironmentObject var container: AppContainer
    @State private var selectedTab = 0

    var body: some View {
        TabView(selection: $selectedTab) {

            // ??????? "?????"
            NavigationStack {
//                MapScreen(vm: .init(service: container.taskService), mapMode: .placeholder)
            }
            .tabItem {
                Label("?????", systemImage: "map")
            }
            .tag(0)

            // ??????? "???????"
            NavigationStack {
//                RouteScreen(vm: .init(service: container.taskService))
            }
            .tabItem {
                Label("???????", systemImage: "point.topleft.down.curvedto.point.bottomright.up")
            }
            .tag(1)

            // ????? ??????? "??????????"
            NavigationStack {
//                MyAdsScreen()
            }
            .tabItem {
                Label("??????????", systemImage: "rectangle.stack.badge.plus")
            }
            .tag(2)

            // ??????? "????"
            NavigationStack {
//                ChatsScreen(vm: .init(service: container.chatService))
            }
            .tabItem {
                Label("????", systemImage: "bubble.left.and.bubble.right")
            }
            .tag(3)

            // ??????? "???????"
            NavigationStack {
//                ProfileScreen(vm: .init(service: container.profileService))
            }
            .tabItem {
                Label("???????", systemImage: "person.circle")
            }
            .tag(4)
        }
        .background(Color.black)
        .ignoresSafeArea()
//        .tint(Theme.ColorToken.turquoise)
        .cornerRadius(20)

//        .background(.ultraThinMaterial)
//        .clipShape(RoundedRectangle(cornerRadius: 16, style: .continuous))
        .softCardShadow()

    }
}


// ===== File: DelegationApp/Features/Route/ViewModel/RouteViewModel.swift =====
import Foundation

final class RouteViewModel: ObservableObject {
    @Published var pointA: String = "?????????? ???????"
    @Published var pointB: String = "??????? ??? ??????? ????????"
    @Published var time: String = "17:00"
    @Published var tasks: [TaskItem] = []

    private let service: TaskService
    init(service: TaskService) {
        self.service = service
        self.tasks = service.loadRouteTasks()
    }
}


// ===== File: DelegationApp/Features/Untitled.swift =====
//import SwiftUI
//
//extension Color {
//    static func hex(_ hex: String) -> Color {
//        let hex = hex.trimmingCharacters(in: CharacterSet.alphanumerics.inverted)
//        var int: UInt64 = 0; Scanner(string: hex).scanHexInt64(&int)
//        let a, r, g, b: UInt64
//        switch hex.count {
//        case 3: (a,r,g,b) = (255, (int >> 8) * 17, (int >> 4 & 0xF) * 17, (int & 0xF) * 17)
//        case 6: (a,r,g,b) = (255, int >> 16, int >> 8 & 0xFF, int & 0xFF)
//        case 8: (a,r,g,b) = (int >> 24, int >> 16 & 0xFF, int >> 8 & 0xFF, int & 0xFF)
//        default:(a,r,g,b) = (255,0,0,0)
//        }
//        return Color(.sRGB,
//                     red: Double(r)/255, green: Double(g)/255,
//                     blue: Double(b)/255, opacity: Double(a)/255)
//    }
//}


// ===== File: DelegationApp/Others/YandexMapConfigurator.swift =====
import YandexMapsMobile

/// ???????????????? ????????? Yandex MapKit.
enum YandexMapConfigurator {
    private static var isConfigured = false

    static func configureIfNeeded() {
        // ? SwiftUI Preview ?????? ?? ?????????????? SDK.
//        if RuntimeEnvironment.isPreview { return }
        guard !isConfigured else { return }

        // ? ???? ???? ???????? ????
        YMKMapKit.setApiKey("df3f9145-2080-42b7-9b91-b879c34236bb")
        YMKMapKit.sharedInstance()
        isConfigured = true
    }
}
```