**import** SwiftUI

**import** FirebaseFirestore

**struct** GuestView: View {

@EnvironmentObject **var** authManager: AuthManager

@State **private** **var** clients: [Client] = []

@State **private** **var** filteredClients: [Client] = []

@State **private** **var** searchText = ""

@State **private** **var** filterPosition: String? = **nil**

@State **private** **var** filterStarkerFuss: String? = **nil**

@State **private** **var** errorMessage: String = ""

**var** body: **some** View {

NavigationStack {

VStack(spacing: 20) {

HStack {

VStack(alignment: .leading) {

Text("Willkommen, Verein")

.font(.title)

.fontWeight(.bold)

**if** **let** email = authManager.userEmail {

Text("Eingeloggt als: \(email)")

.font(.subheadline)

.foregroundColor(.gray)

}

}

Spacer()

Button(action: { authManager.signOut() }) { // Korrigierter Aufruf

Text("Ausloggen")

.foregroundColor(.red)

}

}

.padding()

TextField("Suche Klienten...", text: $searchText)

.textFieldStyle(RoundedBorderTextFieldStyle())

.padding()

.onChange(of: searchText) { applyFilters() }

HStack {

Picker("Position", selection: $filterPosition) {

Text("Alle Positionen").tag(String?.none)

Text("Tor").tag(String?.some("Tor"))

Text("Innenverteidigung").tag(String?.some("Innenverteidigung"))

Text("Aussenverteidiger").tag(String?.some("Aussenverteidiger"))

Text("Mittelfeld").tag(String?.some("Mittelfeld"))

Text("Stürmer").tag(String?.some("Stürmer"))

}

.pickerStyle(MenuPickerStyle())

.onChange(of: filterPosition) { applyFilters() }

Picker("Starker Fuß", selection: $filterStarkerFuss) {

Text("Alle").tag(String?.none)

Text("rechts").tag(String?.some("rechts"))

Text("links").tag(String?.some("links"))

Text("beide").tag(String?.some("beide"))

}

.pickerStyle(MenuPickerStyle())

.onChange(of: filterStarkerFuss) { applyFilters() }

}

.padding(.horizontal)

List {

ForEach(filteredClients) { client **in**

NavigationLink(destination: ClientView(client: client)) {

Text("\(client.vorname) \(client.name)")

}

}

}

}

.navigationTitle("Gast")

.alert(isPresented: .constant(!errorMessage.isEmpty)) {

Alert(title: Text("Fehler"), message: Text(errorMessage), dismissButton: .default(Text("OK")) {

errorMessage = ""

})

}

.onAppear {

loadClients()

}

}

}

**private** **func** loadClients() {

FirestoreManager.shared.getClients { result **in**

**switch** result {

**case** .success(**let** loadedClients):

clients = loadedClients

filteredClients = loadedClients

applyFilters()

**case** .failure(**let** error):

errorMessage = "Fehler beim Laden der Klienten: \(error.localizedDescription)"

}

}

}

**private** **func** applyFilters() {

filteredClients = clients.filter { client **in**

**let** matchesSearch = searchText.isEmpty ||

client.vorname.lowercased().contains(searchText.lowercased()) ||

client.name.lowercased().contains(searchText.lowercased())

**let** matchesPosition = filterPosition == **nil** ||

client.positionFeld?.contains { $0.lowercased().contains(filterPosition!.lowercased()) } ?? **false**

**let** matchesFoot = filterStarkerFuss == **nil** ||

client.starkerFuss?.lowercased() == filterStarkerFuss?.lowercased()

**return** matchesSearch && matchesPosition && matchesFoot

}

}

}

#Preview {

GuestView()

.environmentObject(AuthManager())

}