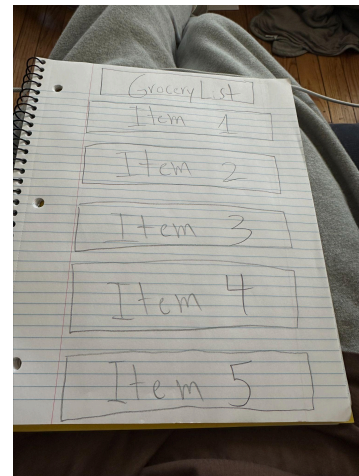
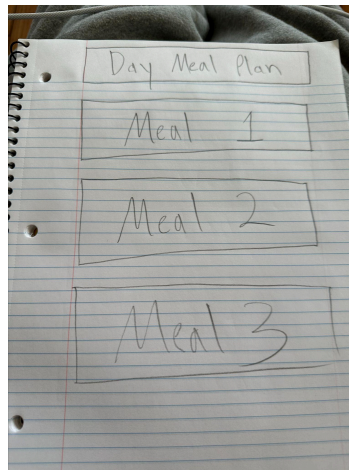
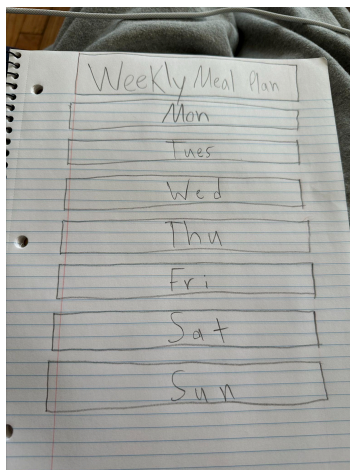


When making the Meal Planner app, there were some things I didn't know how to do. One big challenge was saving meals that users add so they stay in the app after it closes. I used UserDefaults to save favorites, but I don't know how to fully save custom meals or sync them across devices. Another issue is editing and deleting meals, right now, you can only favorite meals, but I'm not sure how to let users change or remove them. I also don't know how to add a search bar to quickly find meals or how to let users drag and drop meals to rearrange them. A better way to show the meal plan would be an interactive calendar instead of a simple list, but I don't know how to do that yet.



Second half - you can turn in POC without the first code/ this should be how your project would look/ on xcode it looks fine but the app may look zoomed out

Coding dates (get the apple sign for date picker)

@binding var allTasks: [duetask]

@State Private var dueDate

Var now date()

```
(
  DatePicker(
    "Start Date"
    Selection: $date,
    displayedCompents: [date.]
  )
  ButtonAction
```

```
import SwiftUI
```

```
struct Meal: Identifiable, Codable {  
    var id = UUID()  
    var name: String  
    var category: String  
    var image: String  
    var description: String  
    var isFavorite: Bool  
}
```

```
func loadMeals() -> [Meal] {  
    if let savedData = UserDefaults.standard.data(forKey: "savedMeals") {  
        if let decoded = try? JSONDecoder().decode([Meal].self, from: savedData) {  
            return decoded  
        }  
    }  
    return [  
        Meal(name: "Grilled Chicken Salad", category: "Lunch", image: "salad", description: "A  
healthy and light grilled chicken salad with fresh greens and vinaigrette.", isFavorite: false),  
        Meal(name: "Spaghetti Bolognese", category: "Dinner", image: "spaghetti", description:  
"Classic Italian pasta dish with rich, meaty tomato sauce.", isFavorite: false),  
        Meal(name: "Omelette", category: "Breakfast", image: "omelette", description: "Fluffy  
omelette with cheese and peppers, perfect for a protein-packed breakfast.", isFavorite: false),  
        Meal(name: "Tacos", category: "Dinner", image: "tacos", description: "Spicy and flavorful  
tacos with seasoned meat and fresh toppings.", isFavorite: false),  
        Meal(name: "Pancakes", category: "Breakfast", image: "pancakes", description: "Soft and  
fluffy pancakes served with syrup and fresh berries.", isFavorite: false),  
        Meal(name: "Grilled Salmon", category: "Dinner", image: "salmon", description: "A delicious  
and healthy grilled salmon fillet served with steamed vegetables.", isFavorite: false),  
        Meal(name: "Caesar Salad", category: "Lunch", image: "caesar_salad", description: "Crispy  
romaine lettuce with creamy caesar dressing, croutons, and parmesan cheese.", isFavorite:  
false)  
    ]  
}
```

```
func saveMeals(mealsToBeSaved: [Meal]) {  
    if let encoded = try? JSONEncoder().encode(mealsToBeSaved) {  
        UserDefaults.standard.set(encoded, forKey: "savedMeals")  
    }  
}
```

```
struct ContentView: View {
```

```

@State var meals = loadMeals()
@State private var newMealName = ""
@State private var newMealCategory = ""
@State private var newMealDescription = ""
@State private var showAddMealView = false
@State private var searchText = ""

var filteredMeals: [Meal] {
    if searchText.isEmpty {
        return meals
    } else {
        return meals.filter { $0.name.localizedCaseInsensitiveContains(searchText) }
    }
}

var body: some View {
    NavigationView {
        VStack {
            Text("Meal Planner")
                .font(.largeTitle)
                .bold()
                .padding()

            TextField("Search Meals", text: $searchText)
                .textFieldStyle(RoundedBorderTextFieldStyle())
                .padding()

            List(filteredMeals.indices, id: \.self) { index in
                NavigationLink(destination: SwiftUIView(selectedMeal: $meals[index])) {
                    HStack {
                        Image(meals[index].image)
                            .resizable()
                            .scaledToFit()
                            .frame(width: 90, height: 100)
                            .cornerRadius(10)

                        VStack(alignment: .leading) {
                            Text(meals[index].name)
                                .font(.title2)
                                .fontWeight(.semibold)

                            Text(meals[index].category)
                                .font(.headline)
                                .fontWeight(.light)
                        }
                    }
                }
            }
        }
    }
}

```

```

        .foregroundColor(Color.gray)
    }
}
}
}
.navigationTitle("Meal Planner")

Button("Add Meal") {
    showAddMealView.toggle()
}
.padding()
.background(Color.blue)
.foregroundColor(.white)
.cornerRadius(10)
.sheet(isPresented: $showAddMealView) {
    AddMealView(meals: $meals)
}
}
}
}
}

```

```

struct AddMealView: View {
    @Binding var meals: [Meal]
    @State private var name = ""
    @State private var category = ""
    @State private var description = ""
    @Environment(\.presentationMode) var presentationMode

    var body: some View {
        NavigationView {
            Form {
                Section(header: Text("Meal Details")) {
                    TextField("Meal Name", text: $name)
                    TextField("Category", text: $category)
                    TextField("Description", text: $description)
                }
            }
            .navigationBarItems(trailing: Button("Save") {
                let newMeal = Meal(name: name, category: category, image: "default", description:
description, isFavorite: false)
                meals.append(newMeal)
                saveMeals(mealsToBeSaved: meals)
                presentationMode.wrappedValue.dismiss()
            })
        }
    }
}

```

```

    })
  }
}

```

```

import SwiftUI
struct SwiftUIView: View {
    @Binding var selectedMeal: Meal

    var body: some View {
        VStack(spacing: 20) {
            Image(selectedMeal.image)
                .resizable()
                .scaledToFit()
                .frame(width: 200, height: 200)
                .cornerRadius(15)

            Text(selectedMeal.name)
                .font(.largeTitle)
                .bold()

            Text(selectedMeal.description)
                .font(.body)
                .padding()
                .multilineTextAlignment(.center)

            Button(action: {
                selectedMeal.isFavorite.toggle()
                saveMeals(mealsToBeSaved: [selectedMeal])
            }) {
                HStack {
                    Image(systemName: selectedMeal.isFavorite ? "heart.fill" : "heart")
                        .foregroundColor(selectedMeal.isFavorite ? .red : .gray)
                    Text(selectedMeal.isFavorite ? "Remove from Favorites" : "Add to Favorites")
                }
                .padding()
                .background(Color.gray.opacity(0.2))
                .cornerRadius(10)
            }
        }
        .padding()
    }
}

```

