

# Code Implementation



## User Story 5:

“As a community member, I want to comment on and like posts so that I can engage with others and build relationships.”

## Event View:

The **EventView** is the main screen displaying event details. I included state variables to manage user interaction and appearance:

- **likeCount** tracks the number of likes.
- **isLiked** indicates if the event is liked by the user.
- **customGreen** is a computed property that defines a custom green colour.

The **headerView** is a **HStack** that displays the event location with an icon and a user profile icon.

The body property of **EventCardView** consists of a **VStack** displaying:

- The event image, styled with resizable properties and rounded corners.
- A **HStack** with the event title and a like button that toggles **isLiked** and updates **likeCount**.
- Additional details, such as category, attendees, and date.
- A section with more detailed information about the event, including an "About" section and location.
- An interest button that changes appearance and function based on **buttonState**.

## Event Data Model

The **Event** struct represents the event data model and conforms to **Identifiable** and **Hashable**. It includes:

- **id**: A unique identifier.
- **image**: The event's image name.
- **title**: The event's title.
- **location**: The event's location.
- **date**: The event's date.
- **category**: The event's category.

```
1 //
2 // EventViewModel.swift
3 // EventPage
4 //
5 // Created by Dilyan Mezev on 27/05/2024.
6 //
7
8 import Foundation
9 import Combine
10
11 // Event Model
12 struct Event: Identifiable, Hashable {
13     var id = UUID()
14     var image: String
15     var title: String
16     var location: String
17     var date: String
18     var category: String
19 }
20
21 class EventViewModel: ObservableObject {
22     @Published var popularEvents: [Event] = [
23         Event(image: "event1", title: "Charity Concert", location: "Philips Stadium", date: "20 May", category: "Entertainment"),
24         Event(image: "event2", title: "Glow Festival", location: "Eindhoven", date: "18 May", category: "Entertainment")
25     ]
26
27     @Published var upcomingEvents: [Event] = [
28         Event(image: "event3", title: "Glow Festival", location: "Eindhoven", date: "18 May", category: "Entertainment"),
29         Event(image: "event4", title: "Charity Concert", location: "Philips Stadium", date: "20 May", category: "Entertainment")
30     ]
31 }
32 |
```

## Connection Between Data Model and Event View

The **EventView** and **EventCardView** views use instances of the **Event** struct to display event details. These instances are managed by the **EventViewModel** class, which holds arrays of **Event** objects. By using the **@Published** property wrapper, changes in **EventViewModel**'s data automatically propagate to the views, ensuring the UI remains up-to-date with the underlying data. The connection between the data model and the views is established through the use of bindings and observable objects, allowing for a responsive and dynamic user interface.