

1. Einleitung

Diese Hausarbeit beschreibt die Entwicklung meiner Reise-App mit Flutter. Ziel der App ist es, Benutzern verschiedene Reisen und Ausflugsziele basierend auf Kategorien anzuzeigen. Die App ermöglicht es den Benutzern, Reisen als Favoriten zu markieren und verschiedene Filter anzuwenden, um die angezeigten Reisen nach Jahreszeit und Familienfreundlichkeit zu filtern.

2. Entwicklungsumgebung

- **Programmiersprache:** Dart
- **Framework:** Flutter
- **IDE:** Visual Studio Code

3. Projektstruktur

Das Projekt ist in verschiedene Dateien und Verzeichnisse unterteilt, um die Übersichtlichkeit und Wartbarkeit des Codes zu gewährleisten.

- **models/**
 - `category.dart`
 - `trip.dart`
- **screens/**
 - `categories_screen.dart`
 - `category_trips_screen.dart`
 - `favorites_screen.dart`
 - `filters_screen.dart`
 - `tabs_screen.dart`
 - `trip_detail_screen.dart`
- **widgets/**
 - `app_drawer.dart`
 - `category_item.dart`
 - `trip_item.dart`
- **app_data.dart**
- **main.dart**

4. Implementierung der Modelle

`category.dart`

- Definiert die Klasse `Category`, die die Kategorien repräsentiert.
- Eigenschaften: `id`, `title`, `imageUrl`.

`trip.dart`

- Definiert die Klasse `Trip`, die die Reisen repräsentiert.
- Eigenschaften: `id`, `categories`, `title`, `activities`, `program`, `duration`, `season`, `tripType`, `isForFamilies`, `isInSummer`, `isInWinter`.
- Aufzählungstypen `Season` und `TripType` zur Definition von Jahreszeiten und Reisearten.

5. Implementierung der Bildschirme

`categories_screen.dart`

- Stellt die Kategorien in einem `GridView` dar.
- Verwendet `CategoryItem`-Widget zur Darstellung jeder Kategorie.

`category_trips_screen.dart`

- Zeigt die Reisen einer ausgewählten Kategorie an.
- Verwendet `ListView.builder` zur dynamischen Erstellung der Reiseelemente.

`favorites_screen.dart`

- Zeigt die vom Benutzer als Favorit markierten Reisen an.
- Verwendet ebenfalls `ListView.builder`.

`filters_screen.dart`

- Ermöglicht dem Benutzer das Anwenden von Filtern (Sommer, Winter, familienfreundlich).
- Speichert die ausgewählten Filter und wendet sie auf die angezeigten Reisen an.

`tabs_screen.dart`

- Zeigt zwei Haupttabs: Kategorien und Favoriten.
- Verwendet `BottomNavigationBar` zur Navigation zwischen den Tabs.

`trip_detail_screen.dart`

- Zeigt die Details einer ausgewählten Reise an, einschließlich Aktivitäten und Tagesprogramm.
- Ermöglicht das Markieren einer Reise als Favorit.

6. Implementierung der Widgets

`app_drawer.dart`

- Definiert ein Navigation Drawer für die App.
- Enthält Links zu den Hauptbildschirmen der App.

`category_item.dart`

- Repräsentiert eine einzelne Kategorie als Widget.
- Enthält eine Methode zur Navigation zur `CategoryTripsScreen` bei Auswahl.

`trip_item.dart`

- Repräsentiert eine einzelne Reise als Widget.
- Enthält Methoden zur Umwandlung von `season` und `tripType` in Text und zur Navigation zur `TripDetailScreen` bei Auswahl.

7. Datenverwaltung

`app_data.dart`

- Enthält die Dummy-Daten für Kategorien und Reisen, die in der App angezeigt werden.

8. Hauptdatei

`main.dart`

- Einstiegspunkt der App.
- Enthält die Hauptklasse `MyApp` und verwaltet den Zustand der Filter und Favoriten.
- Definiert die Routen für die Navigation zwischen den Bildschirmen.

9. Herausforderungen

- **State Management:** Die Verwaltung des Zustands der Filter und Favoriten war eine zentrale Herausforderung. Hier wurde `setState` verwendet, um den Zustand zu aktualisieren und die Änderungen im UI widerzuspiegeln.
- **Datenfilterung:** Die Logik zur Filterung der Reisen basierend auf den ausgewählten Filtern musste sorgfältig implementiert werden, um sicherzustellen, dass nur die relevanten Reisen angezeigt werden.
- **Responsive Design:** Die App wurde so gestaltet, dass sie auf verschiedenen Bildschirmgrößen gut aussieht. Dies erforderte den Einsatz von flexiblen Layouts wie `GridView` und `ListView`.

10. Fazit

Die Entwicklung meiner Reise-App mit Flutter bot eine umfassende Einführung in die Entwicklung von mobilen Anwendungen mit diesem Framework. Die Anwendung von State Management, Routing und benutzerdefinierten Widgets waren Schlüsselthemen, die im Laufe des Projekts behandelt wurden. Trotz einiger Herausforderungen konnte eine funktionale und ansprechende App erstellt werden, die Benutzern eine gute Benutzererfahrung bietet.

11. Zukünftige Erweiterungen

- **Backend-Anbindung:** Integration eines Backends zur Speicherung und Verwaltung der Reise- und Benutzerdaten.
- **Benutzeranmeldung:** Hinzufügen von Benutzeranmeldungen, um personalisierte Erlebnisse zu ermöglichen.
- **Erweiterte Filter:** Hinzufügen weiterer Filteroptionen wie Preis, Bewertung, etc.
- **Kartenintegration:** Integration von Karten zur Anzeige der genauen Lage der Reiseziele.