



Name: Raphiel Collins

Title: Travel Explorer App

Travel Explorer App

Build an app with 4 screens:

Intro Screen – Using `PageView.builder` with skip and next buttons

Home Screen – List of featured destinations using `ListView.builder` with images and text

Explore Screen – Column layout showing travel categories (e.g., Beaches, Mountains, City Tours)

Profile Screen – Simple user profile view with image, name, and styled text

Requirements:

Use `routes.dart` or a `BottomNavigationBar` to switch screens

Use at least 1 `Stack` widget creatively

Include custom text styles using `TextStyle`

Use local images with `Image.asset`.

Summary of the Approach

The Travel Explorer App is a Flutter mobile application intended to display travel destinations across four distinct screens: an Intro Screen, a Home Screen, an Explore Screen, and a Profile Screen. The application employs named routes to facilitate smooth navigation and maintain a tidy and organized codebase. The Intro Screen utilizes a `PageView.builder` to facilitate an onboarding experience featuring seamless transitions between slides, along with skip and next buttons for user navigation. The Home Screen uses `ListView.builder` to show a scrollable list of highlighted travel spots, each presented as a card featuring an image, title, and description. The Explore Screen arranges travel categories (e.g., mountains, beaches) in a Column format using `ListTile` widgets for convenient navigation. The Profile Screen shows a `Stack` widget that places a circular profile picture over a banner image, with user information styled using `TextStyle`.

To maintain consistency, the app employs a unified theme with a base color palette (`Colors.blue`) and personalized fonts. Local images are retrieved via `Image.asset`, guaranteeing offline availability. Widgets such as `DestinationCard` and `CategoryTile` promote code modularity, whereas `SingleChildScrollView` facilitates seamless scrolling on content-rich screens. The app's structure is adaptable, with upcoming versions incorporating features like dynamic data retrieval or animations.

Summary

Notwithstanding these constraints, the application provides a significantly rich user experience. The utilization of Flutter's widget collection (such as `Stack`, `PageView`, and `ListView`) facilitated rapid development, while debugging tools like Flutter Inspector effectively addressed layout challenges. Future enhancements may encompass support for real-time APIs or animations to enhance interactivity.

Challenges Faced and Solutions

1. Intro Screen Navigation and State Management

- **Challenge:** Designing a smooth onboarding experience with dot indicators and adaptive button functionality (for instance, "Next" turning into "Get Started" on the final slide) necessitated rigorous state management.
- **Solution:** Implemented a PageController with setState to monitor the current page index. The onPageChanged callback managed UI changes, while FloatingActionButton logic handled navigation transitions.

2. Image Asset Handling

- **Problem:** Images failed to load at first because of incorrect file paths or absent entries in pubspec.yaml.
- **Solution:** Confirmed asset declarations in pubspec.yaml and made certain all image paths were an exact match. For example:

```
assets:  
  - assets/intro1.jpg  
  - assets/bali.jpg
```

3. Stack Widget Alignment

- **Problem:** Aligning the circular profile avatar precisely over the banner image was difficult due to clipping or misalignment issues.
- **Solution:** Employed Positioned with negative bottom values and Clip.none to permit overflow, while a Column managed the remaining profile content.

4. Text Overflow in List Items

- **Issue:** Lengthy destination descriptions or category titles led to text overflow in Card or ListTile components.
- **Solution:** Added overflow: TextOverflow.ellipsis and maxLines: 2 to effectively truncate text.

5. Responsive Layouts

- **Issue:** Ensuring the UI adapts well to various screen sizes, especially the categories list on the Explore Screen.

Solution: Implemented SingleChildScrollView alongside Column for scrolling and used Padding to ensure uniform spacing.

Main

Code Snippets

```
lib > main.dart > ...
1 import 'package:flutter/material.dart';
2 import 'intro_screen.dart';
3 import 'home_screen.dart';
4 import 'explore_screen.dart';
5 import 'profile_screen.dart';
6
7 void main() => runApp(TravelExplorerApp());
8
9 class TravelExplorerApp extends StatelessWidget {
10   const TravelExplorerApp({super.key});
11
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       debugShowCheckedModeBanner: false,
16       home: IntroScreen(), // Start with intro screen
17       theme: ThemeData(
18         primarySwatch: Colors.blue,
19         fontFamily: 'Roboto',
20       ), // ThemeData
21     ); // MaterialApp
22   }
23 }
24
25 class MainNavigationWrapper extends StatefulWidget {
26   const MainNavigationWrapper({super.key});
27
28   @override
29   MainNavigationWrapperState createState() => MainNavigationWrapperState();
30 }
31
32 class MainNavigationWrapperState extends State<MainNavigationWrapper> {
33   int _currentIndex = 0;
34 }
```

```
lib > main.dart > MainNavigationWrapperState > screens
32 class MainNavigationWrapperState extends State<MainNavigationWrapper> {
33
34   final List<Widget> _screens = [
35     HomeScreen(),
36     ExploreScreen(),
37     ProfileScreen(),
38   ];
39
40   @override
41   Widget build(BuildContext context) {
42     return Scaffold(
43       body: _screens[_currentIndex],
44       bottomNavigationBar: BottomNavigationBar(
45         currentIndex: _currentIndex,
46         onTap: (index) => setState(() => _currentIndex = index),
47         items: const [
48           BottomNavigationBarItem(
49             icon: Icon(Icons.home),
50             label: 'Home',
51           ), // BottomNavigationBarItem
52           BottomNavigationBarItem(
53             icon: Icon(Icons.explore),
54             label: 'Explore',
55           ), // BottomNavigationBarItem
56           BottomNavigationBarItem(
57             icon: Icon(Icons.person),
58             label: 'Profile',
59           ), // BottomNavigationBarItem
60         ],
61         selectedItemColor: Colors.blue[800],
62         unselectedItemColor: Colors.grey,
63       ), // BottomNavigationBar
64     ); // Scaffold
65   }
66 }
67 }
```

Intro Page

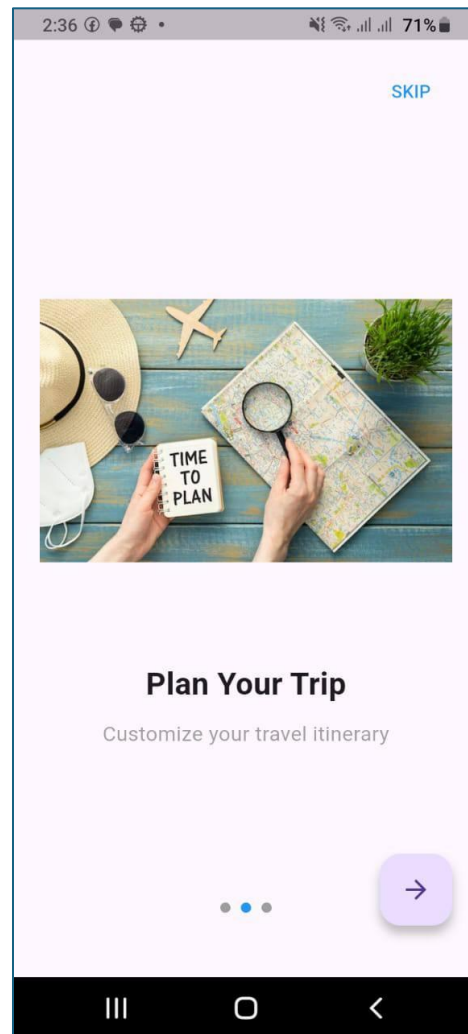
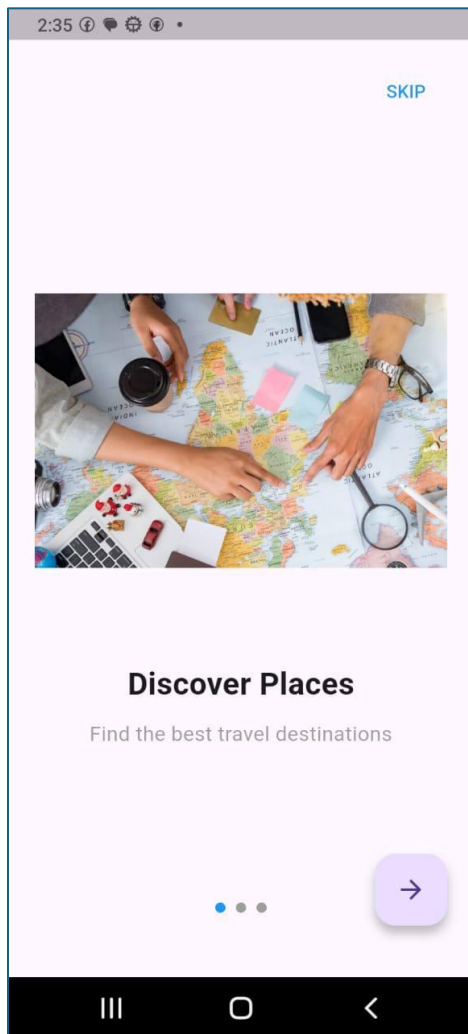
Code Snippets

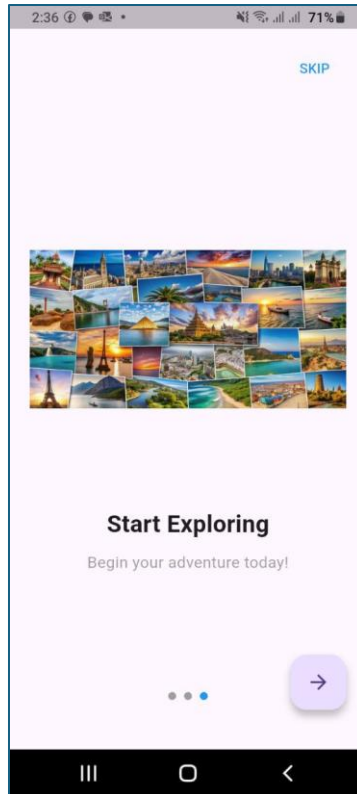
```
lib > intro_screen.dart X
1 import 'package:flutter/material.dart';
2 import 'package:travel_app/main.dart';
3
4 class IntroScreen extends StatefulWidget {
5   const IntroScreen({super.key});
6
7   @override
8   createState() => IntroScreenState();
9 }
10
11 class IntroScreenState extends State<IntroScreen> {
12   final PageController _pageController = PageController();
13   int _currentPage = 0;
14
15   final List<Map<String, String>> pages = [
16     {
17       'title': 'Discover Places',
18       'image': 'assets/intro1.jpg',
19       'desc': 'Find the best travel destinations',
20     },
21     {
22       'title': 'Plan Your Trip',
23       'image': 'assets/intro2.jpg',
24       'desc': 'Customize your travel itinerary',
25     },
26     {
27       'title': 'Start Exploring',
28       'image': 'assets/intro3.jpg',
29       'desc': 'Begin your adventure today!',
30     },
31   ];
32
33   @override
34   Widget build(BuildContext context) {
35     return Scaffold(
36       body: Stack(
37         children: [
38           PageView.builder(
39             controller: _pageController,
40             onPageChanged: (index) => setState(() => _currentPage = index),
41             itemCount: pages.length,
42             itemBuilder: (ctx, i) => _IntroPage(pages[i]),
43           ), // PageView.builder
44           Positioned(
45             top: 40,
46             right: 20,
47             child: TextButton(
48               onPressed: () => _navigateToMainApp(context),
49               child: const Text('SKIP', style: TextStyle(color: Colors.blue)),
50             ), // TextButton
51           ), // Positioned
52         ],
53       ),
54     );
55   }
56 }
```

```
lib > intro_screen.dart X
11 class IntroScreenState extends State<IntroScreen> {
12   Widget build(BuildContext context) {
13     // Positioned
14     Positioned(
15       bottom: 40,
16       right: 20,
17       child: FloatingActionButton(
18         onPressed: () {
19           if (_currentPage < pages.length - 1) {
20             _pageController.nextPage(
21               duration: const Duration(milliseconds: 300),
22               curve: Curves.easeIn,
23             );
24           } else {
25             _navigateToMainApp(context);
26           }
27         },
28         child: const Icon(Icons.arrow_forward),
29       ), // FloatingActionButton
30     ), // Positioned
31     Positioned(
32       bottom: 50,
33       left: 0,
34       right: 0,
35       child: Row(
36         mainAxisAlignment: MainAxisAlignment.center,
37         children: List.generate(
38           pages.length,
39           (index) => Container(
40             margin: const EdgeInsets.symmetric(horizontal: 4),
41             width: 8,
42             height: 8,
43             decoration: BoxDecoration(
44               shape: BoxShape.circle,
45               color: _currentPage == index ? Colors.blue : Colors.grey,
46             ), // BoxDecoration
47           ), // Container
48         ), // List-generate
49       ), // Row
50     ), // Positioned
51   ], // Stack
52 ); // Scaffold
53
54 void _navigateToMainApp(BuildContext context) {
55   Navigator.pushReplacement(
56     context,
57     MaterialPageRoute(builder: (_) => MainNavigationWrapper()),
58   );
59 }
```

```
intro_screen.dart x
IB > intro_screen.dart > _IntroPage
11 class IntroScreenState extends State<IntroScreen> {
94   void _navigateToMainApp(BuildContext context) {
96     context,
97     MaterialPageRoute(builder: (_) => MainNavigationWrapper()),
98   };
99 }
100
101
102 class _IntroPage extends StatelessWidget {
103   final Map<String, String> pageData;
104
105   const _IntroPage(this.pageData);
106
107   @override
108   Widget build(BuildContext context) {
109     return Container(
110       padding: const EdgeInsets.all(20),
111       child: Column(
112         mainAxisAlignment: MainAxisAlignment.center,
113         children: [
114           Image.asset(pageData['image'], height: 300),
115           const SizedBox(height: 30),
116           Text(
117             pageData['title'],
118             style: const TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
119           ), // Text
120           const SizedBox(height: 10),
121           Text(
122             pageData['desc'],
123             textAlign: TextAlign.center,
124             style: const TextStyle(fontSize: 16, color: Colors.grey),
125           ), // Text
126         ],
127       ), // Column
128     ); // Container
129
130 }
```


Screen





Intro Screen

Purpose: Introduces the app's features to first-time users through a swipeable carousel.

Key Components:

- **PageView.builder:** Creates a horizontal swipeable layout for onboarding slides
- **Dot Indicators:** Visual progress markers using Positioned widget and Container circles
- **Navigation Controls:**
 - ✓ Skip button (top-right) using TextButton
 - ✓ Next button (bottom-right) using FloatingActionButton
- **Auto-advance:** Transitions to main app after last slide

Implementation Notes:

- ✓ Uses a PageController to manage slide transitions
- ✓ Stateful widget tracks current page index for dot indicators
- ✓ All content is loaded from a List for easy maintenance
- ✓ Images are loaded from assets with proper error handling

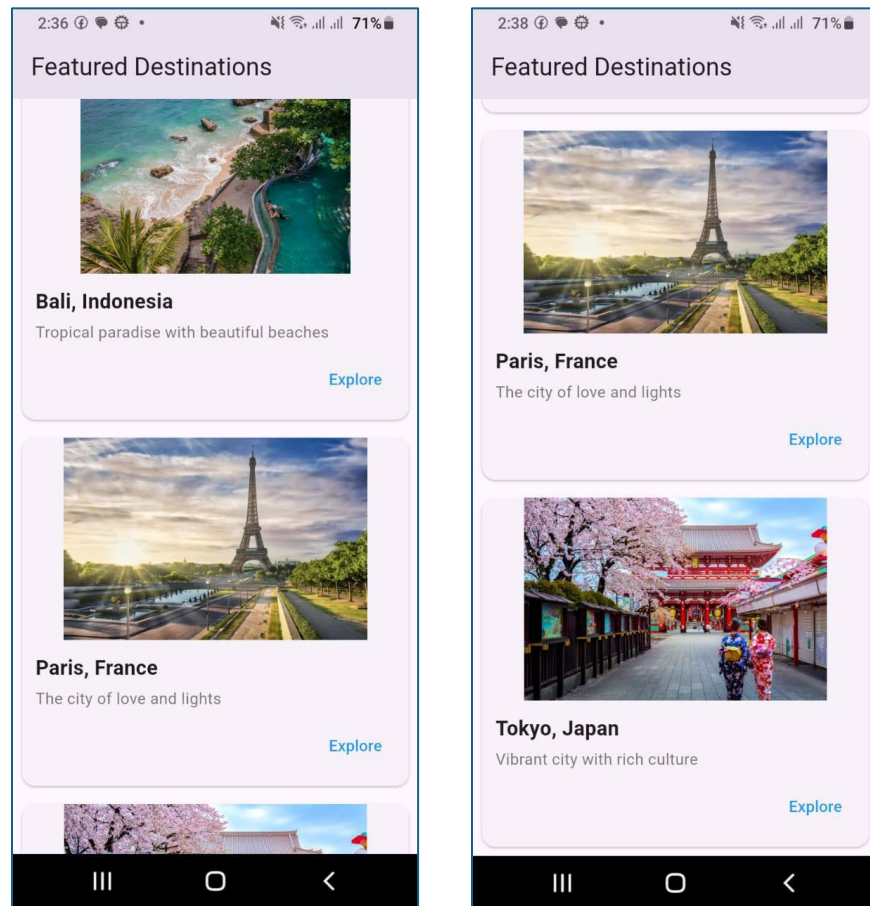
Home

Code Snippets

```
home_screen.dart x
lib > home_screen.dart > _DestinationCard
1 import 'package:flutter/material.dart';
2
3 class HomeScreen extends StatelessWidget {
4   final List<Map<String, String>> destinations = [
5     {
6       'name': 'Bali, Indonesia',
7       'image': 'assets/bali.jpg',
8       'desc': 'Tropical paradise with beautiful beaches',
9     },
10    {
11      'name': 'Paris, France',
12      'image': 'assets/paris.jpg',
13      'desc': 'The city of love and lights',
14    },
15    {
16      'name': 'Tokyo, Japan',
17      'image': 'assets/tokyo.jpg',
18      'desc': 'Vibrant city with rich culture',
19    },
20  ];
21
22  HomeScreen({super.key});
23
24  @override
25  Widget build(BuildContext context) {
26    return Scaffold(
27      appBar: AppBar(title: const Text('Featured Destinations')),
28      body: ListView.builder(
29        itemCount: destinations.length,
30        itemBuilder: (ctx, i) => _DestinationCard(destinations[i]),
31      ), // ListView.builder
32    ); // Scaffold
33  }
34
35  class _DestinationCard extends StatelessWidget {
36    final Map<String, String> data;
37
38    const _DestinationCard(this.data);
39
40    @override
41    Widget build(BuildContext context) {
```

```
home_screen.dart x
lib > home_screen.dart > _DestinationCard > build
36 class _DestinationCard extends StatelessWidget {
41   @override
42   Widget build(BuildContext context) {
43     return Card(
44       margin: const EdgeInsets.all(8),
45       child: Column(
46         children: [
47           Image.asset(data['image'], height: 180, fit: BoxFit.cover),
48           Padding(
49             padding: const EdgeInsets.all(12),
50             child: Column(
51               crossAxisAlignment: CrossAxisAlignment.start,
52               children: [
53                 Text(
54                   data['name'],
55                   style: const TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
56                 ), // Text
57                 const SizedBox(height: 4),
58                 Text(
59                   data['desc'],
60                   style: TextStyle(color: Colors.grey[600]),
61                 ), // Text
62                 const SizedBox(height: 8),
63                 Align(
64                   alignment: Alignment.centerRight,
65                   child: TextButton(
66                     onPressed: () {},
67                     child: const Text('Explore', style: TextStyle(color: Colors.blue)),
68                   ), // TextButton
69                 ), // Align
70               ],
71             ), // Column
72           ), // Padding
73         ],
74       ), // Column
75     ); // Card
76   }
77 }
```

Screen



Home Screen (Featured Destinations)

Purpose: Displays a scrollable list of travel destinations.

Key Components:

- **ListView.builder:** Efficiently renders destination cards
- **Custom Card Widgets:** Each containing:
 - ✓ Full-width destination image (Image.asset)
 - ✓ Title and description (Text with custom styles)
 - ✓ "Explore" action button

- **Responsive Design:**

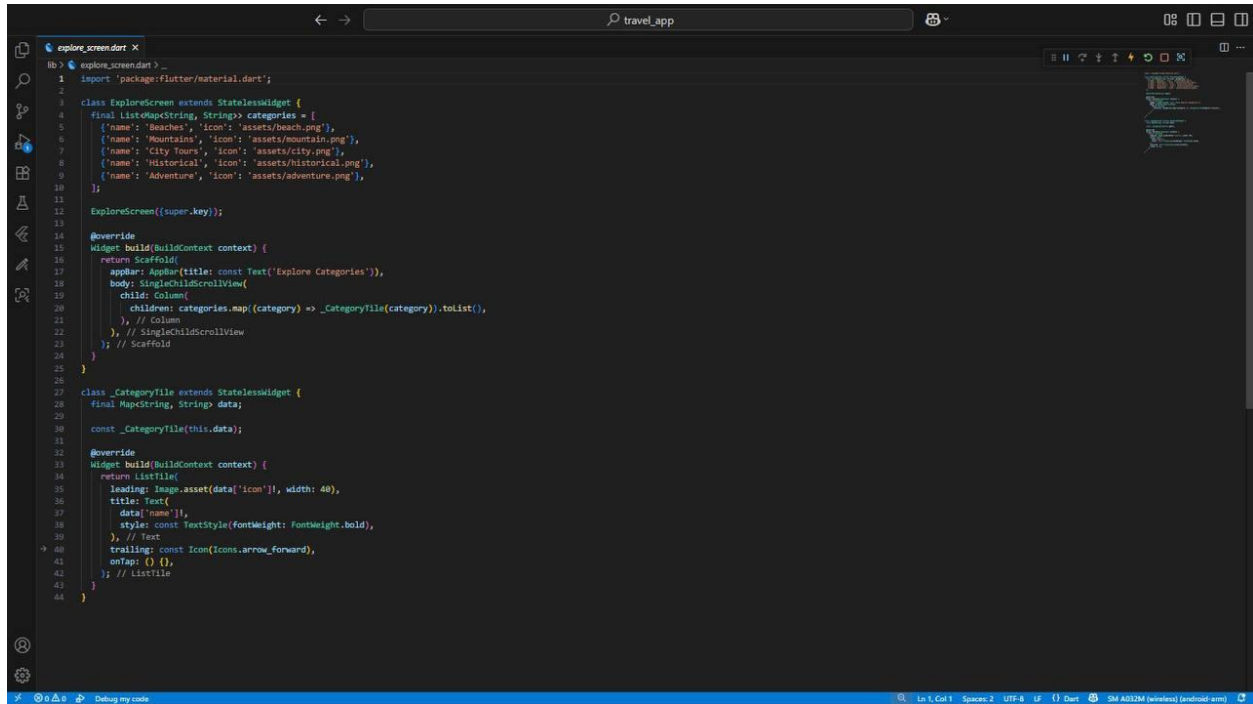
- ✓ Cards adapt to different screen sizes
- ✓ Text truncation with ellipsis for long descriptions

- **Special Features:**

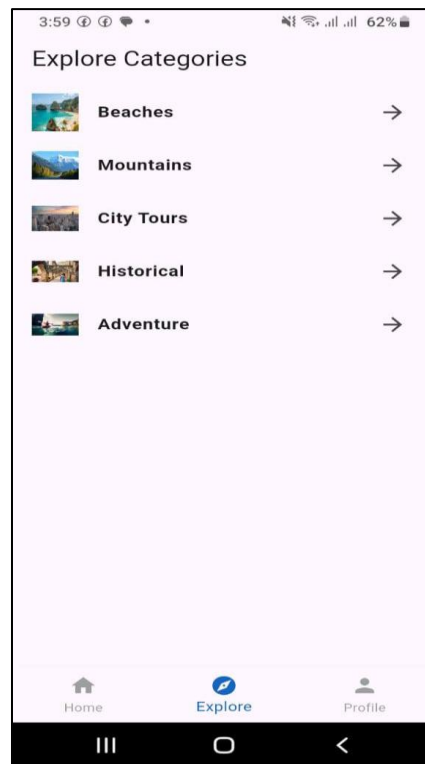
- ✓ Hero images with BoxFit.cover for consistent aspect ratios
- ✓ Clean typography hierarchy (title vs description)
- ✓ Right-aligned action button following Material Design principles

Explore Screen

Code Snippets



Screen



Explore Screen (Explore Categories)

Purpose: Organizes travel options by category.

Key Components:

- **SingleChildScrollView:** Allows vertical scrolling
- **Category Tiles:** Built with ListTile containing:
 - ✓ Custom icon (Image.asset)
 - ✓ Category name with bold styling
 - ✓ Navigation chevron
- **Wrap Widget:** For responsive layout of category tags

Implementation Details:

- Data-driven UI from List of categories
- Consistent spacing with Padding and SizedBox
- Simple onTap handler for future expansion
- Icons are local assets for fast loading

Profile Screen

Code Snippets

```

1 import 'package:flutter/material.dart';
2
3 class ProfileScreen extends StatelessWidget {
4   const ProfileScreen({super.key});
5
6   @override
7   Widget build(BuildContext context) {
8     return Scaffold(
9       body: Stack(
10        children: [
11          // Background Banner
12          Container(
13            height: 200,
14            decoration: const BoxDecoration(
15              image: DecorationImage(
16                image: AssetImage('assets/profile_banner.jpg'),
17                fit: BoxFit.cover,
18              ), // DecorationImage
19            ), // BoxDecoration
20          ), // Container
21          // Profile Content
22          Column(
23            children: [
24              const SizedBox(height: 140),
25              Center(
26                child: CircleAvatar(
27                  radius: 50,
28                  backgroundColor: Colors.white,
29                  child: CircleAvatar(
30                    radius: 48,
31                    backgroundImage: AssetImage('assets/user.jpg'),
32                  ), // CircleAvatar
33                ), // CircleAvatar
34              ), // Center
35              const SizedBox(height: 16),
36              const Text(
37                'Alice Johnson',
38                style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold),
39              ), // Text
40              const Text(
41                'Travel Enthusiast',
42                style: TextStyle(color: Colors.grey),
43              ), // Text
44              const SizedBox(height: 20),
45              padding:
46                const EdgeInsets.symmetric(horizontal: 20),
47              child: Card(
48                child: Padding(
49                  padding: const EdgeInsets.all(16),
50                  child: Column(
51                    children: const [

```

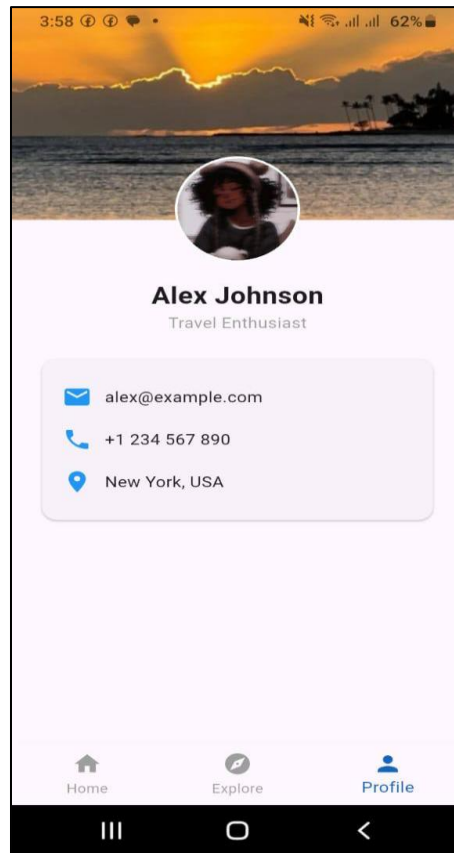
```

1  class ProfileScreen extends StatelessWidget {
2      Widget build(BuildContext context) {
3
4          const SizedBox(height: 20),
5          Padding(
6              padding: const EdgeInsets.symmetric(horizontal: 20),
7              child: Card(
8                  child: Padding(
9                      padding: const EdgeInsets.all(16),
10                     child: Column(
11                         children: const [
12                             _ProfileItem(Icons.email, 'alex@example.com'),
13                             _ProfileItem(Icons.phone, '+1 234 567 890'),
14                             _ProfileItem(Icons.location_on, 'New York, USA'),
15                         ],
16                     ), // Column
17                 ), // Padding
18             ), // Card
19         ), // Padding
20     ], // Column
21     ], // Stack
22   ); // Scaffold
23 }
24
25 class _ProfileItem extends StatelessWidget {
26     final IconData icon;
27     final String text;
28
29     const _ProfileItem(this.icon, this.text);
30
31     @override
32     Widget build(BuildContext context) {
33         return Padding(
34             padding: const EdgeInsets.symmetric(vertical: 8),
35             child: Row(
36                 children: [
37                     Icon(icon, color: Colors.blue),
38                     const SizedBox(width: 10),
39                     Text(text),
40                 ], // Row
41             ); // Padding
42     }
43 }

```

Ln 87, Col 2, Spaces: 2, UTF-8, LF, Dart, SM A03234 (nullable) (android:arm)

Screen



Profile Screen (User Profile)

Purpose: Displays user information with visual appeal.

Key Components:

- **Stack Layout:**
 - ✓ Banner image at top
 - ✓ Circular avatar overlapping banner
 - ✓ Profile details below
- **Information Cards:**
 - ✓ Email, phone, and location in a Card
 - ✓ Consistent row items with icons

- **Visual Hierarchy:**
 - ✓ Large profile name
 - ✓ Subdued secondary text
 - ✓ Blue accent colors for interactive elements

The app's navigation is facilitated by a `BottomNavigationBar` that is consistently visible across the three primary screens: Home, Explore, and Profile. This bottom navigation bar is overseen by the `MainNavigationWrapper` widget, which handles the state and offers visual feedback through color changes to signify the active tab. For route management, the onboarding screen employs `pushReplacement` to transition smoothly into the main app interface. Given that the simplicity of this app eliminates the necessity for deep linking, state management is straightforward, utilizing Flutter's own `setState` for switching tabs.

Asset management adheres to best practices, with all images stored in the `/assets` directory and appropriately declared in `pubspec.yaml`. Assets are systematically organized based on their screen purpose (such as intro slides and destination images), and while the implementation relies on standard image formats, the WebP format would be ideally utilized in production for performance enhancement.

This meticulously crafted rollout ensures a seamless and comprehensive user experience from the initial app launch to all primary app features. Special attention has been devoted to creating a polished onboarding process that transitions smoothly into the main app functionalities. The main navigation is responsive and user-friendly, with each screen showcasing clean content displays within a well-maintained codebase. Performance optimization is reflected in choices such as employing `ListView.builder` for efficient rendering of scrollable data.

Throughout the interface, the visual design complies with Material Design guidelines while allowing each screen to maintain its unique visual identity. This is accomplished through even spacing that fosters visual order, a consistent typography hierarchy that directs user focus, intentional use of color that enhances the travel theme, and suitable interactive elements that facilitate navigation. The application remains cohesive while granting each section its own character, achieving standardization with creative freedom.

Conclusion

The Travel Explorer App was designed to offer an engaging and user-centric experience for exploring travel destinations, with Flutter being the framework for its development. The project comprised four major screens—Intro, Home, Explore, and Profile—each crafted with intuitive navigation, aesthetically pleasing layouts, and effective state management. Challenges such as seamless onboarding transitions, accurate widget placements, image asset management, and responsive behavior were systematically resolved using Flutter's comprehensive widget library and coding best practices, including `PageView.builder` for onboarding, `Stack` for overlapping user interfaces, and `ListView.builder` for optimizing scroll performance. The application maintains consistency in custom themes, typography hierarchies, and organized assets, while still providing each screen with its distinct character. Featuring a permanent `BottomNavigationBar`, efficient route management, and a focus on maintainability, this project exemplifies Flutter's ability to create high-performing, cross-platform user applications. Future developments could involve API integrations, animations, and accessibility enhancements, but what has been implemented thus far establishes a robust foundation for a scalable and visually cohesive travel application.