

Name: Lisa Sharma

Roll No: 53

MAD and PWA Lab Experiment - 5

Aim: To apply navigation, routing and gestures in Flutter App

Theory:

Navigation:

Navigation is a fundamental aspect of mobile app development that involves transitioning between different screens or pages. In Flutter, the Navigator class plays a central role in managing the navigation stack, allowing developers to push and pop routes as users move through the app.

Navigator Class:

The Navigator class handles the navigation stack, maintaining a history of routes. The push method adds a new route to the stack, typically triggered by user actions. The pop method removes the current route from the stack, enabling backward navigation.

Routing:

Routing in Flutter involves defining and organizing the paths or routes within the application. Routes represent different screens or pages, and their effective use is crucial for structuring the app's architecture.

MaterialPageRoute and CupertinoPageRoute:

MaterialPageRoute is employed in apps following Material Design principles, providing a standard Android-style transition between screens.

CupertinoPageRoute is used for iOS-style designs, ensuring a consistent and native user experience.

Named Routes:

Named routes offer a more organized and readable approach to navigation.

By assigning names to routes, such as '/details', developers can easily navigate to

specific screens using `Navigator.pushNamed`.

Gestures in Flutter:

Gestures enhance user interaction by allowing the app to respond to touch or mouse inputs. In Flutter, the `GestureDetector` widget is instrumental in recognizing and handling various gestures.

`GestureDetector` Widget:

The `GestureDetector` widget is used to detect a variety of gestures, including taps, drags, and long presses.

By wrapping UI components with `GestureDetector`, developers can specify callback functions to execute when specific gestures are detected.

Gesture Recognizers:

Flutter provides gesture recognizers for more complex gestures, such as panning, pinching, and swiping.

These recognizers, when combined with a `GestureDetector`, enable the app to respond to a broader range of user inputs.

`InkWell` and `InkResponse`:

The `InkWell` and `InkResponse` widgets bring a material ripple effect to touchable UI components.

Integrating these widgets enhances the visual feedback during user interactions, contributing to a more polished and intuitive user experience.

Code:

`Booking_details.dart`

```
import 'dart:ui';

import 'package:flutter/material.dart';
import 'package:go_router/go_router.dart';
import 'package:models/models.dart';

import '../shared/theme/colors.dart';
import '../shared/widgets/select_date_widget.dart';
import '../shared/widgets/select_destination_widget.dart';
import '../shared/widgets/select_guests_widget.dart';

class BookingDetailsScreen extends StatefulWidget {
  const BookingDetailsScreen({super.key});
```

```

@override
State<BookingDetailsScreen> createState() => _BookingDetailsScreenState();
}

class _BookingDetailsScreenState extends State<BookingDetailsScreen> {
  var step = BookingStep.selectDate;

  @override
  Widget build(BuildContext context) {
    final textTheme = Theme.of(context).textTheme;

    return BackdropFilter(
      filter: ImageFilter.blur(sigmaX: 8.0, sigmaY: 8.0),
      child: Scaffold(
        backgroundColor: appWhite.withOpacity(0.5),
        appBar: AppBar(
          backgroundColor: Colors.transparent,
          automaticallyImplyLeading: false,
          leading: IconButton(
            onPressed: () => context.pop(),
            icon: const Icon(Icons.close),
          ),
          title: Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              TextButton(
                onPressed: () {},
                child: Text(
                  'Stays',
                  style: textTheme.titleMedium!.copyWith(
                    fontWeight: FontWeight.bold,
                  ),
                ),
              TextButton(
                onPressed: () {},
                child: Text(
                  'Experiences',
                  style: textTheme.titleMedium,
                ),
              ),
            ],
          ),
          actions: const [SizedBox(width: 48.0)],
        ),
        body: SingleChildScrollView(
          child: Padding(
            padding: const EdgeInsets.only(left: 16.0, right: 16.0, top: 16.0),
            child: Column(
              children: [
                GestureDetector(
                  onTap: () {
                    setState(() {
                      step = BookingStep.selectDestination;
                    });
                  },
                  child: Hero(
                    tag: 'search',
                    child: SelectDestinationWidget(step: step),
                  ),
                ),
                GestureDetector(
                  onTap: () {

```

```

        setState(() {
          step = BookingStep.selectDate;
        });
      },
      child: SelectDateWidget(step: step),
    ),
    (step == BookingStep.selectDate)
      ? const SizedBox()
      : GestureDetector(
          onTap: () {
            setState(() {
              step = BookingStep.selectGuests;
            });
          },
          child: SelectGuestsWidget(step: step),
        ),
  ],
),
),
),
bottomNavigationBar: (step == BookingStep.selectDate)
  ? null
  : BottomAppBar(
      padding: const EdgeInsets.symmetric(horizontal: 16.0),
      notchMargin: 0,
      color: Colors.white,
      surfaceTintColor: Colors.white,
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          TextButton(
            onPressed: () {
              // if (step == BookingStep.selectDestination) {
              //   setState(() {
              //     step = BookingStep.selectGuests;
              //   });
              // } else {
              //   setState(() {
              //     step = BookingStep.selectDestination;
              //   });
              // }
            },
            child: Text(
              'Clear all',
              style: Theme.of(context).textTheme.bodyLarge!.copyWith(
                fontWeight: FontWeight.bold,
                decoration: TextDecoration.underline,
              ),
            ),
          ),
          FilledButton.icon(
            onPressed: () {},
            style: FilledButton.styleFrom(
              backgroundColor: appRed,
              minimumSize: const Size(100, 56.0),
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(8.0),
              ),
            ),
            icon: const Icon(Icons.search),
            label: const Text('Search'),
          ),
        ],
      ),
    ),
  ),
),

```

```

    ),
  ),
);
}
}

```

select_guests_widget.dart

```

import 'package:flutter/material.dart';
import 'package:flutter_animate/flutter_animate.dart';
import 'package:models/models.dart';

class SelectGuestsWidget extends StatelessWidget {
  const SelectGuestsWidget({
    super.key,
    required this.step,
  });

  final BookingStep step;

  @override
  Widget build(BuildContext context) {
    final textTheme = Theme.of(context).textTheme;

    return Card(
      elevation: 0.0,
      clipBehavior: Clip.antiAlias,
      child: AnimatedContainer(
        height: step == BookingStep.selectGuests ? 274 : 60,
        width: double.infinity,
        padding: const EdgeInsets.symmetric(
          vertical: 16.0,
          horizontal: 16.0,
        ),
        duration: const Duration(milliseconds: 300),
        child: step == BookingStep.selectGuests
          ? Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  'Who\'s coming?',
                  style: textTheme.headlineSmall!.copyWith(
                    fontWeight: FontWeight.bold,
                  ),
                ),
                const SizedBox(height: 16.0),
                SizedBox(
                  height: 190,
                  child: ListView(
                    physics: const NeverScrollableScrollPhysics(),
                    padding: EdgeInsets.zero,
                    children: [
                      _buildGuestsQuantitySelector(
                        context,
                        'Adults',
                        'Ages 13 or above',
                        () {},
                        () {},
                        '0',
                      ),
                      const Divider(),
                      _buildGuestsQuantitySelector(
                        context,

```

```

        'Children',
        'Ages 2-12',
        () {},
        () {},
        '0',
    ),
    const Divider(),
    _buildGuestsQuantitySelector(
        context,
        'Infants',
        'Under 2',
        () {},
        () {},
        '0',
    ),
],
),
),
],
)
    .animate(delay: const Duration(milliseconds: 300))
    .fadeIn(duration: const Duration(milliseconds: 300))
: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
        Text(
            'Who',
            style: textTheme.bodyMedium,
        ),
        Text(
            'Add guests',
            style: Theme.of(context)
                .textTheme
                .bodyMedium!
                .copyWith(fontWeight: FontWeight.bold),
        ),
    ],
),
);
}

```

```

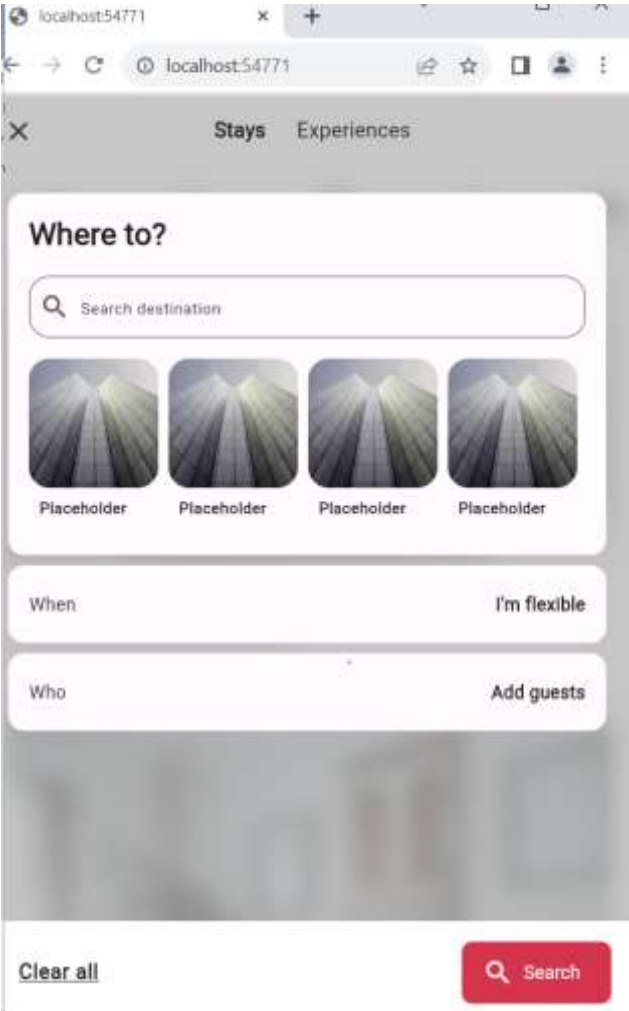
Container _buildGuestsQuantitySelector(
    BuildContext context,
    String title,
    String subtitle,
    VoidCallback onDecrement,
    VoidCallback onIncrement,
    String value,
) {
    final textTheme = Theme.of(context).textTheme;

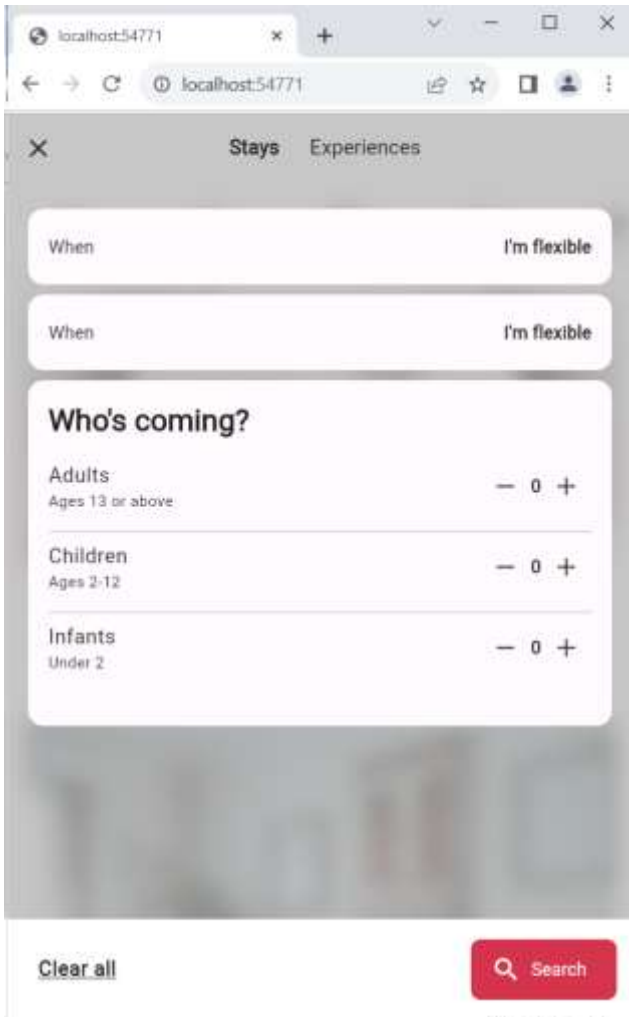
    return Container(
        margin: const EdgeInsets.only(bottom: 8.0),
        child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
                Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                        Text(title, style: textTheme.bodyLarge),
                        Text(subtitle, style: textTheme.bodySmall),
                    ],
                ),
            ],
        ),
    );
}

```

```
Row(
  children: [
    IconButton(
      onPressed: onDecrement,
      icon: const Icon(Icons.remove),
    ),
    Text(
      value,
      style:
        textTheme.bodyMedium!.copyWith(fontWeight: FontWeight.bold),
    ),
    IconButton(
      onPressed: onIncrement,
      icon: const Icon(Icons.add),
    ),
  ],
),
],
),
);
}
```

Output:





Conclusion:

In this experiment, we have successfully created routing in the bottom navigation bar and connected all pages successfully using Navigator class and implemented it successfully.