Name: Atif Ansari Roll no: 04 Class: D15B

MPL Experiment - 5

* Aim: To implement Navigation, Routing, and Gestures in flutter.

* Theory:

Navigation and Routing in Flutter.
In flutter, screens are referred to as routes. Navigation between these routes is managed using the Navigator class. flutter provides multiple ways to navigate between screens:

1. Using Navigator.push() and Navigator.pop()
• Navigator.push (context, MaterialPageRoute (builder: (context) => SecondScreen())) is used to navigate from one screen to another.
• Navigator.pop (context) is used to return to the previous screen.

2. Named Routes
• Named routes allow better management of navigation in larger apps by defining routes in a central location.
• Navigator.pushNamed (context, '/second') is used to navigate ~~a~~ to a named route.
• Navigator.pop (context) is used to return to the previous route.

Gestures in flutter: Gestures allow users to interact with UI elements using touch-based actions like tap, drag, swipe and pinch. The GestureDetector widget in flutter listens for user gestures and responds accordingly.

1. Single Gesture Handling (GestureDetector): Used to detect single gestures like tap, double tap, long press, drag, etc.

2. Multiple Gesture Handling (RawGestureDetector): Used when handling multiple gestures simultaneously. The AllowMultipleGestureRecognizer class enables recognition of multiple gestures within the same widget tree.

* Conclusion: This experiment demonstrated how to implement navigation, routing, and gestures in a flutter application. By using Navigator.push() and Navigator.pop(), we successfully transitioned between screens. Additionally, named routes provided a ~~structed~~ structured approach to navigation management.

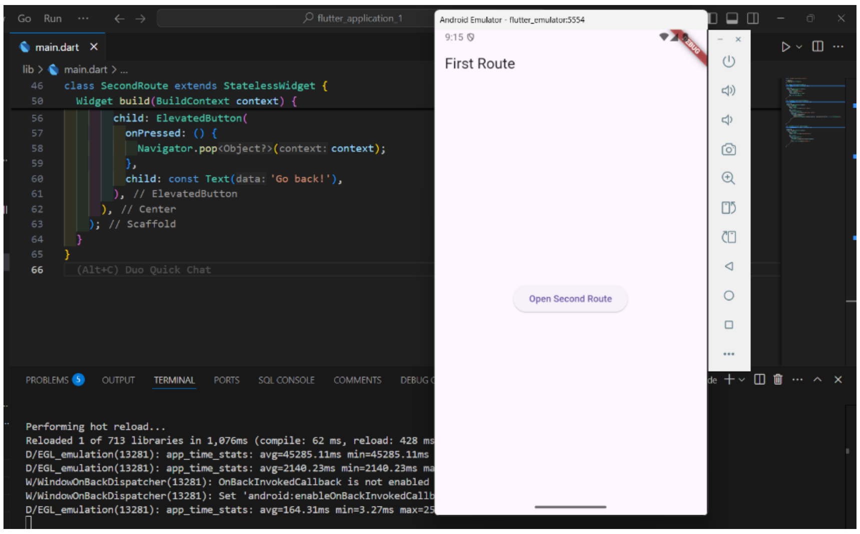
For Gestures, the GestureDetector widget enabled user interaction through touch-based actions. The RawGestureDetector facilitated handling multiple gestures simultaneously. Understanding navigation and gestures in fundamental for building interactive and user-friendly Flutter applications.

1. Basic Navigation Using Navigator.push() and Navigator.pop()

Code:

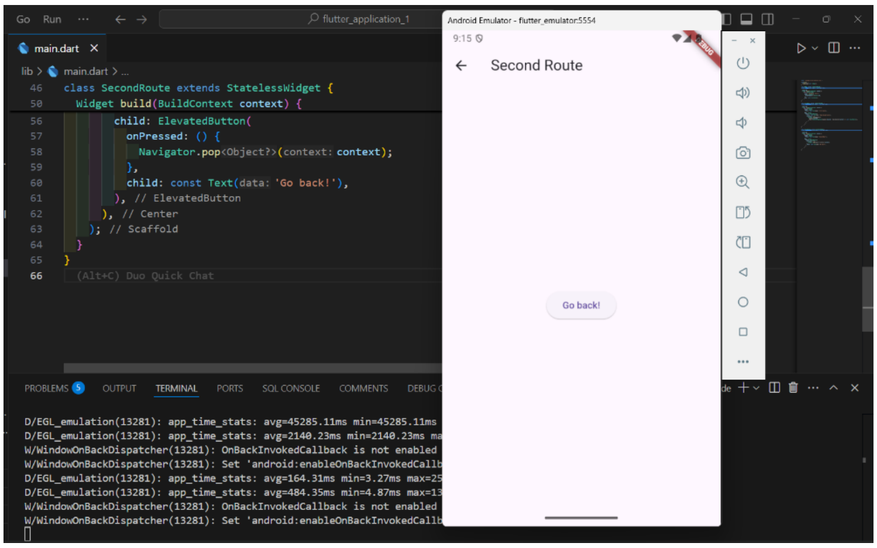
```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Navigation Basics',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const FirstRoute(),
    );
  }
}

class FirstRoute extends StatelessWidget {
  const FirstRoute({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('First Route'),
      ),
      body: Center(
        child: ElevatedButton(
          child: const Text('Open Second Route'),
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => const SecondRoute()),
            );
          },
        ),
      ),
    );
  }
}

class SecondRoute extends StatelessWidget {
  const SecondRoute({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Second Route'),
      ),
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            Navigator.pop(context);
          },
          child: const Text('Go back!'),
        ),
      ),
    );
  }
}
```

Ouput:



```
class SecondRoute extends StatelessWidget {
  Widget build(BuildContext context) {
        child: ElevatedButton(
          onPressed: () {
            Navigator.pop<Object?>(context: context);
          },
          child: const Text(data: 'Go back!'),
        ), // ElevatedButton
      ), // Center
    ); // Scaffold
  }
}
  (Alt+C) Duo Quick Chat
```

```
Performing hot reload...
Reloaded 1 of 713 libraries in 1,076ms (compile: 62 ms, reload: 428 ms
D/EGL_emulation(13281): app_time_stats: avg=45285.11ms min=45285.11ms
D/EGL_emulation(13281): app_time_stats: avg=2140.23ms min=2140.23ms ma
W/WindowOnBackDispatcher(13281): OnBackInvokedCallback is not enabled
W/WindowOnBackDispatcher(13281): Set 'android:enableOnBackInvokedCallb
D/EGL_emulation(13281): app_time_stats: avg=164.31ms min=3.27ms max=25
```



```
class SecondRoute extends StatelessWidget {
  Widget build(BuildContext context) {
          child: ElevatedButton(
            onPressed: () {
              Navigator.pop<Object?>(context: context);
            },
            child: const Text(data: 'Go back!'),
          ), // ElevatedButton
        ), // Center
      ); // Scaffold
    }
}
  (Alt+C) Duo Quick Chat
```

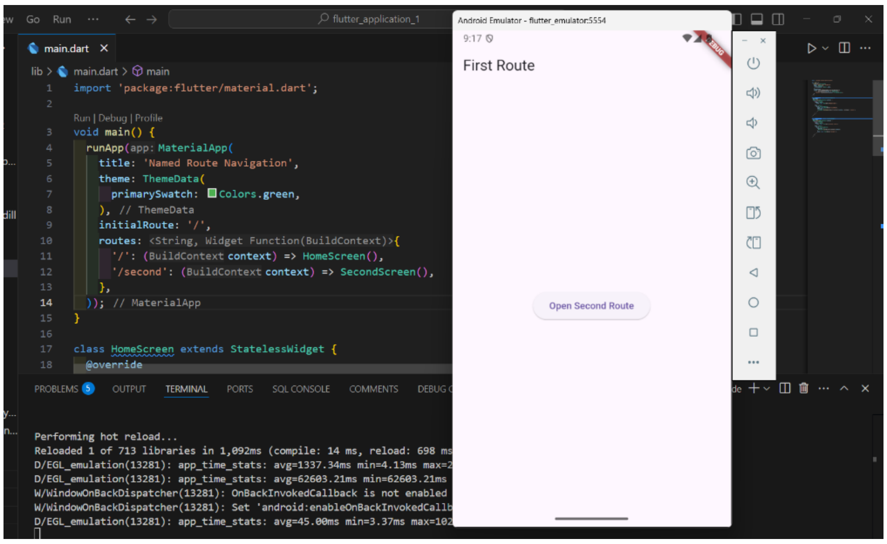
```
D/EGL_emulation(13281): app_time_stats: avg=45285.11ms min=45285.11ms
D/EGL_emulation(13281): app_time_stats: avg=2140.23ms min=2140.23ms ma
W/WindowOnBackDispatcher(13281): OnBackInvokedCallback is not enabled
W/WindowOnBackDispatcher(13281): Set 'android:enableOnBackInvokedCallb
D/EGL_emulation(13281): app_time_stats: avg=164.31ms min=3.27ms max=25
D/EGL_emulation(13281): app_time_stats: avg=484.35ms min=4.87ms max=13
W/WindowOnBackDispatcher(13281): OnBackInvokedCallback is not enabled
W/WindowOnBackDispatcher(13281): Set 'android:enableOnBackInvokedCallb
```

2. Navigation Using Named Routes

Code:

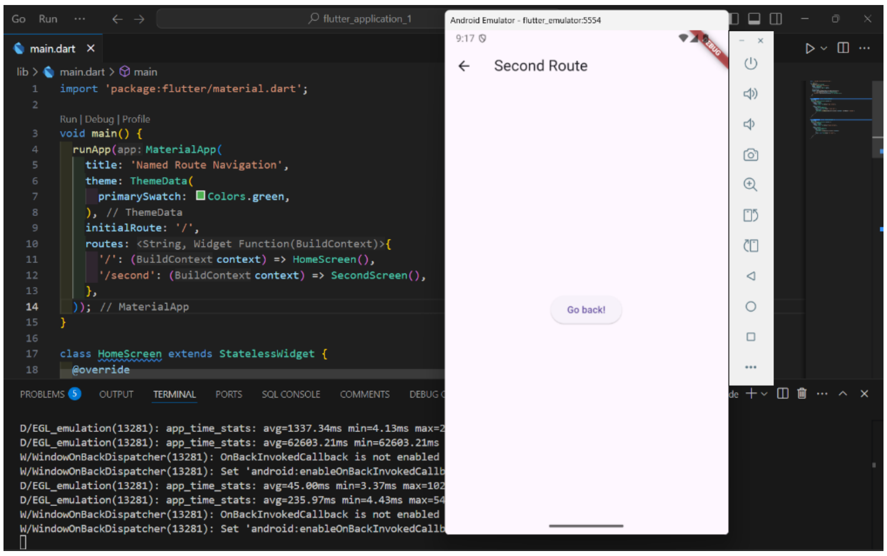
```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Named Route Navigation',
      theme: ThemeData(
        primarySwatch: Colors.green,
      ),
      initialRoute: '/',
      routes: {
        '/': (context) => const HomeScreen(),
        '/second': (context) => const SecondScreen(),
      },
    );
  }
}

class HomeScreen extends StatelessWidget {
 const HomeScreen({Key? key}) : super(key: key);

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(
    title: const Text('Home Screen'),
   ),
   body: Center(
    child: ElevatedButton(
     child: const Text('Click Here'),
     onPressed: () {
      Navigator.pushNamed(context, '/second');
     },
    ),
   ),
  );
 }
}

class SecondScreen extends StatelessWidget {
 const SecondScreen({Key? key}) : super(key: key);

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(
    title: const Text("Second Screen"),
   ),
   body: Center(
    child: ElevatedButton(
     onPressed: () {
      Navigator.pop(context);
     },
     child: const Text('Go back!'),
    ),
   ),
  );
 }
}
```

Ouput:



```
import 'package:flutter/material.dart';

Run | Debug | Profile
void main() {
  runApp(app: MaterialApp(
    title: 'Named Route Navigation',
    theme: ThemeData(
      primarySwatch: 🟩Colors.green,
    ), // ThemeData
    initialRoute: '/',
    routes: <String, Widget Function(BuildContext)>{
      '/': (BuildContext context) => HomeScreen(),
      '/second': (BuildContext context) => SecondScreen(),
    },
  )); // MaterialApp
}

class HomeScreen extends StatelessWidget {
  @override
```

```
Performing hot reload...
Reloaded 1 of 713 libraries in 1,092ms (compile: 14 ms, reload: 698 ms
D/EGL_emulation(13281): app_time_stats: avg=1337.34ms min=4.13ms max=2
D/EGL_emulation(13281): app_time_stats: avg=62603.21ms min=62603.21ms
W/WindowOnBackDispatcher(13281): OnBackInvokedCallback is not enabled
W/WindowOnBackDispatcher(13281): Set 'android:enableOnBackInvokedCallb
D/EGL_emulation(13281): app_time_stats: avg=45.00ms min=3.37ms max=102
```



```
import 'package:flutter/material.dart';

Run | Debug | Profile
void main() {
  runApp(app: MaterialApp(
    title: 'Named Route Navigation',
    theme: ThemeData(
      primarySwatch: 🟩Colors.green,
    ), // ThemeData
    initialRoute: '/',
    routes: <String, Widget Function(BuildContext)>{
      '/': (BuildContext context) => HomeScreen(),
      '/second': (BuildContext context) => SecondScreen(),
    },
  )); // MaterialApp
}

class HomeScreen extends StatelessWidget {
  @override
```

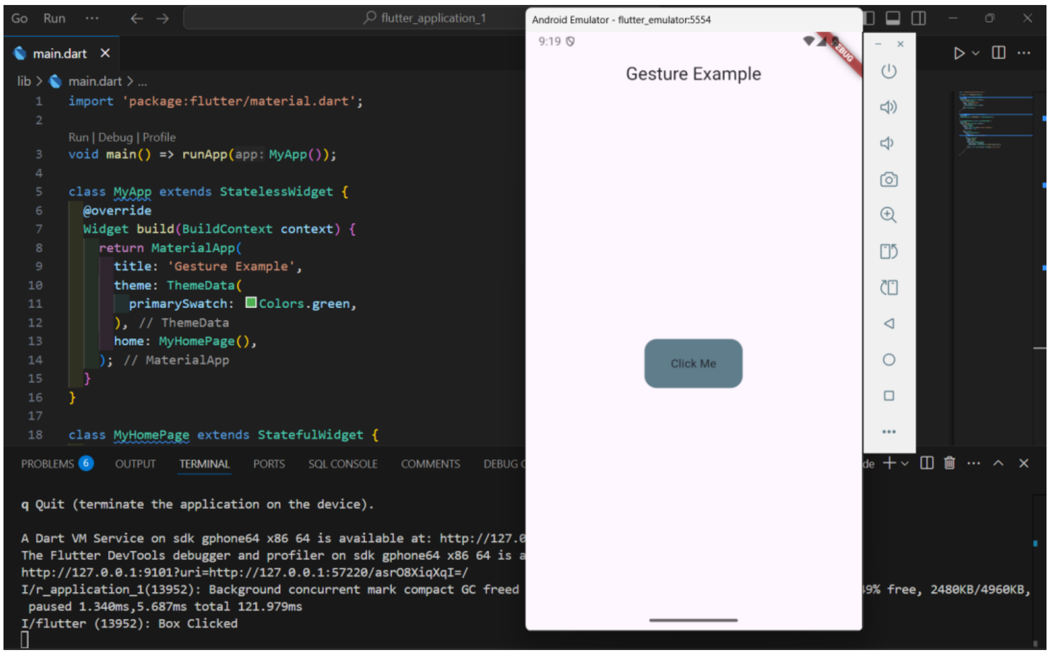
```
D/EGL_emulation(13281): app_time_stats: avg=1337.34ms min=4.13ms max=2
D/EGL_emulation(13281): app_time_stats: avg=62603.21ms min=62603.21ms
W/WindowOnBackDispatcher(13281): OnBackInvokedCallback is not enabled
W/WindowOnBackDispatcher(13281): Set 'android:enableOnBackInvokedCallb
D/EGL_emulation(13281): app_time_stats: avg=45.00ms min=3.37ms max=102
D/EGL_emulation(13281): app_time_stats: avg=235.97ms min=4.43ms max=54
W/WindowOnBackDispatcher(13281): OnBackInvokedCallback is not enabled
W/WindowOnBackDispatcher(13281): Set 'android:enableOnBackInvokedCallb
```

3. Gesture Example Using GestureDetector

Code:

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Gesture Example',
      theme: ThemeData(
        primarySwatch: Colors.green,
      ),
      home: MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  @override
  MyHomePageState createState() => MyHomePageState();
}

class MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Gesture Example'),
        centerTitle: true,
      ),
      body: Center(
        child: GestureDetector(
          onTap: () {
            print('Box Clicked');
          },
          child: Container(
            height: 60.0,
            width: 120.0,
            decoration: BoxDecoration(
              color: Colors.blueGrey,
              borderRadius: BorderRadius.circular(15.0),
            ),
            child: const Center(child: Text('Click Me')),
          ),
        ),
      ),
    );
  }
}
```

Output:



```
import 'package:flutter/material.dart';

Run | Debug | Profile
void main() => runApp(app: MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Gesture Example',
      theme: ThemeData(
        primarySwatch: Colors.green,
      ), // ThemeData
      home: MyHomePage(),
    ); // MaterialApp
  }
}

class MyHomePage extends StatefulWidget {
```

```
q Quit (terminate the application on the device).

A Dart VM Service on sdk gphone64 x86 64 is available at: http://127.0
The Flutter DevTools debugger and profiler on sdk gphone64 x86 64 is a
http://127.0.0.1:9101?uri=http://127.0.0.1:57220/asrO8XiqXqI=/
I/r_application_1(13952): Background concurrent mark compact GC freed                    9% free, 2480KB/4960KB,
 paused 1.340ms,5.687ms total 121.979ms
I/flutter (13952): Box Clicked
```

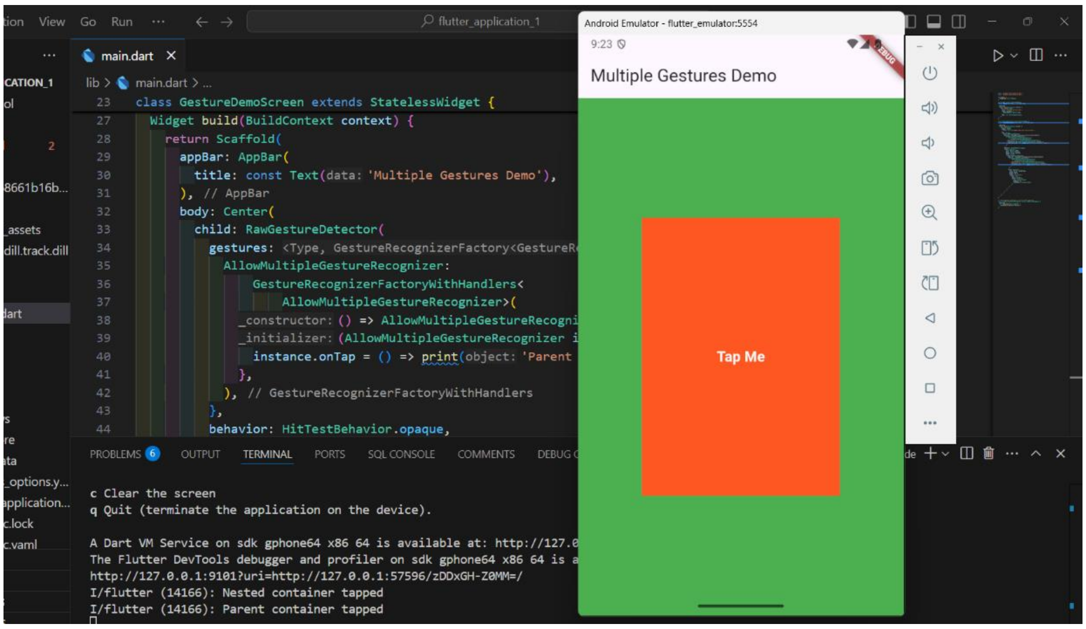
4. Multiple Gesture Handling Using RawGestureDetector

Code:

```
import 'package:flutter/gestures.dart';
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Multiple Gestures Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const GestureDemoScreen(),
    );
  }
}

class GestureDemoScreen extends StatelessWidget {
  const GestureDemoScreen({Key? key}) : super(key: key);
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Multiple Gestures Demo'),
    ),
    body: Center(
      child: RawGestureDetector(
        gestures: {
          AllowMultipleGestureRecognizer:
              GestureRecognizerFactoryWithHandlers<
                  AllowMultipleGestureRecognizer>(
            () => AllowMultipleGestureRecognizer(),
            (AllowMultipleGestureRecognizer instance) {
              instance.onTap = () => print('Parent container tapped');
            },
          ),
        },
        behavior: HitTestBehavior.opaque,
        child: Container(
          color: Colors.green,
          width: double.infinity,
          height: double.infinity,
          child: Center(
            child: RawGestureDetector(
              gestures: {
                AllowMultipleGestureRecognizer:
                    GestureRecognizerFactoryWithHandlers<
                        AllowMultipleGestureRecognizer>(
                  () => AllowMultipleGestureRecognizer(),
                  (AllowMultipleGestureRecognizer instance) {
                    instance.onTap = () => print('Nested container tapped');
                  },
                ),
              },
              child: Container(
                color: Colors.deepOrange,
                width: 250.0,
                height: 350.0,
                child: const Center(
                  child: Text(
                    "Tap Me",
                    style: TextStyle(
                      color: Colors.white,
            fontSize: 18,
            fontWeight: FontWeight.bold,
           ),
          ),
         ),
        ),
       ),
      ),
     ),
    ),
   ),
  );
 }
}

/// Custom Gesture Recognizer to allow multiple gestures
class AllowMultipleGestureRecognizer extends TapGestureRecognizer {
 @override
 void rejectGesture(int pointer) {
  acceptGesture(pointer);
 }
}
```

Output:



```
class GestureDemoScreen extends StatelessWidget {
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(data: 'Multiple Gestures Demo'),
      ), // AppBar
      body: Center(
        child: RawGestureDetector(
          gestures: <Type, GestureRecognizerFactory<GestureR
            AllowMultipleGestureRecognizer:
                GestureRecognizerFactoryWithHandlers<
                    AllowMultipleGestureRecognizer>(
              _constructor: () => AllowMultipleGestureRecogni
              _initializer: (AllowMultipleGestureRecognizer i
                instance.onTap = () => print(object: 'Parent
              },
            ), // GestureRecognizerFactoryWithHandlers
          },
          behavior: HitTestBehavior.opaque,
```

```
c Clear the screen
q Quit (terminate the application on the device).

A Dart VM Service on sdk gphone64 x86 64 is available at: http://127.0
The Flutter DevTools debugger and profiler on sdk gphone64 x86 64 is a
http://127.0.0.1:9101?uri=http://127.0.0.1:57596/zDDxGH-Z0MM=/
I/flutter (14166): Nested container tapped
I/flutter (14166): Parent container tapped
```