NAME:ATHARVA SHINDE **ROLL.NO:54** **CLASS:D15C**

AIM: - To apply navigation, routing, and gestures in Flutter App

Theory:

Flutter is a powerful framework for building cross-platform mobile applications, and it provides efficient mechanisms for:

🔁 1. Navigation & Routing

Navigation allows moving between different screens (also called routes or pages) in a Flutter app. Flutter provides multiple methods to implement navigation:

a) Basic Navigation (Navigator.push and Navigator.pop)

- Navigator.push(context, MaterialPageRoute(builder: (_) => SecondPage())); Pushes a new route onto the stack.
- Navigator.pop(context);
  Pops the top-most route from the stack and returns to the previous screen.

b) Named Routing

- Define routes in MaterialApp's routes property:

c) Navigation Stack

Flutter uses a **stack-based** navigation model where each new screen is "pushed" onto a stack and can be "popped" to return to the previous screen.

✋ 2. Gestures in Flutter

Gestures are used to detect user interaction like taps, swipes, drags, etc.

Flutter uses the GestureDetector widget to handle gestures:

Gesture Type	Widget/Callback Used
Tap	onTap
Double Tap	onDoubleTap
Long Press	onLongPress
Vertical Drag	onVerticalDragUpdate
Horizontal Drag	onHorizontalDragUpdate

GestureDetector is a powerful tool for creating interactive UIs and responding to user inputs like swipe-to-dismiss, tap-to-select, or drag-to-move elements.

InkWell vs GestureDetector:

- GestureDetector: Pure logic-based gesture detection.
- InkWell: Similar, but adds **ripple/touch feedback** when tapped. Ideal for buttons.

Combining Navigation & Gestures

A common real-world example is:

- User **taps a button** → navigates to another screen.
- User **swipes** to dismiss a card or perform an action. These interactions improve the **UX (User Experience)** by making apps feel smooth and intuitive.

Code:

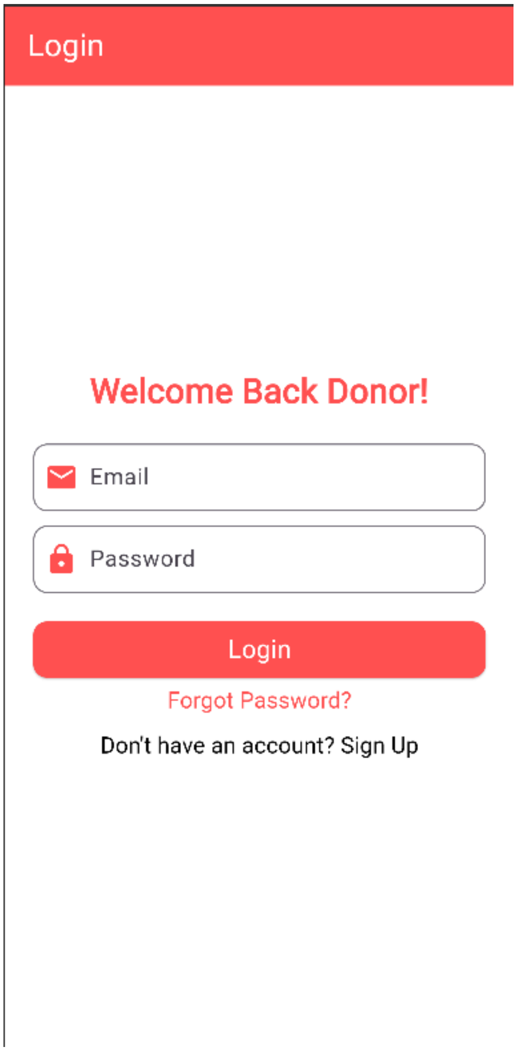
Example: Gesture + Navigation Together

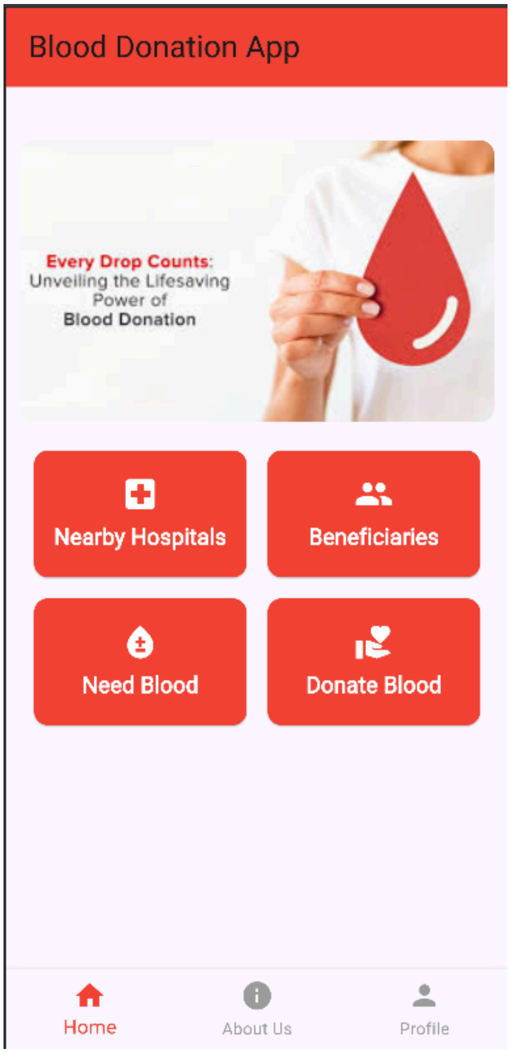
```
GestureDetector(
 onTap: () {
 Navigator.push(
   context,
   MaterialPageRoute(builder: (context) => QRGeneratorScreen()),
  );
 },
 child: Card(
  child: Padding(
   padding: EdgeInsets.all(16),
   child: Text("Tap to Generate QR"),
  ),
 ),
),
```

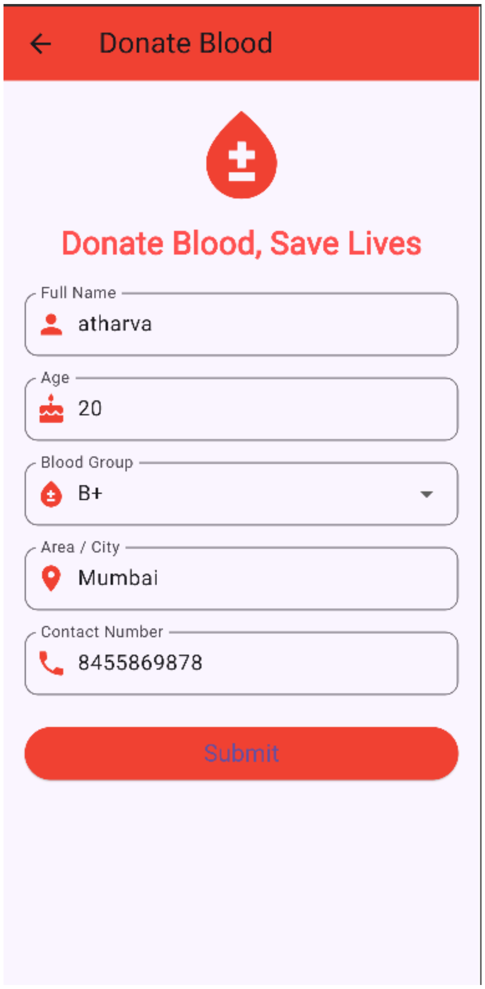
Navigate to Scanner from Your Signup/ Login

```
  ElevatedButton(
          onPressed: signUp,
          child: Text("Sign Up", style: TextStyle(fontSize: 18)),
          style: ElevatedButton.styleFrom(minimumSize: Size(double.infinity, 50)),
         ),
         TextButton(
          onPressed: () {
           Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) =>
  LoginPage()));
          },
          child: Text("Already have an account? Login", style: TextStyle(fontSize: 16)),
         ),
```

Output:


Login
Welcome Back Donor!
Email
Password
Login
Forgot Password?
Don't have an account? Sign Up


Blood Donation App
Every Drop Counts:
Unveiling the Lifesaving
Power of
Blood Donation
Nearby Hospitals
Beneficiaries
Need Blood
Donate Blood
Home
About Us
Profile



Donate Blood, Save Lives

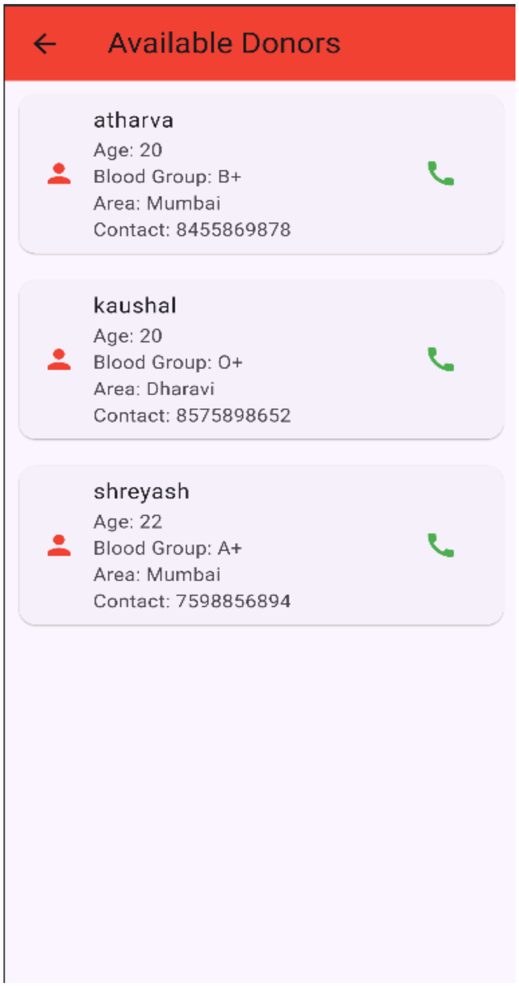
Full Name: atharva

Age: 20

Blood Group: B+

Area / City: Mumbai

Contact Number: 8455869878



atharva
Age: 20
Blood Group: B+
Area: Mumbai
Contact: 8455869878

kaushal
Age: 20
Blood Group: O+
Area: Dharavi
Contact: 8575898652

shreyash
Age: 22
Blood Group: A+
Area: Mumbai
Contact: 7598856894

Conclusion:

In this experiment, we successfully learned and implemented the concepts of **navigation**, **routing**, and **gesture detection** in Flutter. We used the Navigator class to move between screens, understood the difference between **basic and named routing**, and applied GestureDetector to handle various user interactions. This enhances the overall user experience by making the app more dynamic and interactive.