**EXPERIMENT NO: - 05**

**Name:-** Laksh Sodhai **Class:-** D15A **Roll:No: -** 57

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

---

**Theory: -**

In Flutter, the screens and pages are known as routes, and these routes are just a widget. In Android, a route is similar to an Activity.

In any mobile app, navigating to different pages defines the workflow of the application, and the way to handle the navigation is known as routing. Flutter provides a basic routing class MaterialPageRoute and two methods Navigator.push() and Navigator.pop() that shows how to navigate between two routes. The following steps are required to start navigation in your application.

Gestures enable the app to respond to user interactions, making the application more dynamic and responsive.

- **Navigation and Routing in Flutter**

  Navigation is the process of moving between different screens or pages in an app. Flutter provides a simple and effective way to handle this through the use of the Navigator widget and routes.

  **1. Using Navigator Widget**

  The Navigator widget manages a stack of routes, allowing for pushing and popping routes on the stack.

  - **Pushing a Route**: To navigate to a new screen, use Navigator.push().
  - **Popping a Route**: To go back to the previous screen, use Navigator.pop().

```
ElevatedButton(
  onPressed: () {
  Navigator.push(
```

```
      context,
      MaterialPageRoute(builder: (context) => SecondScreen()),
    );},
);
```

**2. Named Routes**

Flutter also allows the use of named routes to navigate, which can make the routing process cleaner, especially in larger applications.

```
MaterialApp(
  initialRoute: '/',
  routes: {
    '/': (context) => HomeScreen(),
    '/second': (context) => SecondScreen(),
  },
);
```

Navigate to the route using Navigator.pushNamed()

```
Navigator.pushNamed(context, '/second');
```

**Handling Gestures in Flutter**

Gestures refer to user interactions with the app, such as taps, swipes, pinches, and drags. Flutter provides several widgets and gesture detectors to handle these interactions.

**Tap Gestures**

The most common gesture is the tap, which can be handled using the GestureDetector widget or specific buttons like InkWell or ElevatedButton.

**Long Press Gesture**

For long press gestures, Flutter provides the onLongPress callback in GestureDetector or InkWell.

**Swipe and Drag Gestures**

Flutter also provides swipe and drag gesture handling. The onHorizontalDragUpdate and onVerticalDragUpdate callbacks are used for dragging gestures.

**Code: -**

main.dart

```
import 'package:flutter/material.dart';
import 'screens/login_screen.dart';
import 'screens/register_screen.dart';
import 'screens/main_screen.dart';
import 'screens/home_screen.dart';
import 'screens/servers_screen.dart';
import 'screens/settings_screen.dart';
import 'screens/profile_screen.dart';
import 'screens/speed_test_screen.dart';
import 'screens/vpn_details_screen.dart';
import 'screens/otp_verification_screen.dart';
import 'screens/phone_login_screen.dart';

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

class VPNApp extends StatelessWidget {
  const VPNApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'SecureShield VPN',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        colorScheme: ColorScheme.dark(
          primary: Colors.blue,
          background: const Color(0xFF1A237E),
          secondary: Colors.blue,
          surface: Colors.white.withOpacity(0.1),
        ),
        scaffoldBackgroundColor: const
Color(0xFF1A237E),
        appBarTheme: const AppBarTheme(
          backgroundColor: Colors.transparent,
          elevation: 0,
          iconTheme: IconThemeData(color:
Colors.white),
          titleTextStyle: TextStyle(
            color: Colors.white,
            fontSize: 24,
            fontWeight: FontWeight.bold,
          ),
        ),
        textTheme: const TextTheme(
          headlineLarge: TextStyle(
            color: Colors.white,
            fontSize: 32,
            fontWeight: FontWeight.bold,
          ),
          headlineMedium: TextStyle(
            color: Colors.white,
            fontSize: 24,
            fontWeight: FontWeight.bold,
          ),
          bodyLarge: TextStyle(
            color: Colors.white,
            fontSize: 16,
          ),
          bodyMedium: TextStyle(
            color: Colors.white70,
            fontSize: 14,
          ),
        ),
        elevatedButtonTheme:
ElevatedButtonThemeData(
          style: ElevatedButton.styleFrom(
            backgroundColor: Colors.blue,
            foregroundColor: Colors.white,
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(12),
            ),
            padding: const
EdgeInsets.symmetric(vertical: 16),
          ),
        ),
        inputDecorationTheme:
InputDecorationTheme(
          filled: true,
          fillColor: Colors.white.withOpacity(0.1),
          border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(12),
            borderSide: BorderSide.none,
          ),
          enabledBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(12),
            borderSide: BorderSide(color:
Colors.white.withOpacity(0.1)),
          ),
          focusedBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(12),
            borderSide: const BorderSide(color:
Colors.blue),
          ),
          labelStyle: const TextStyle(color:
Colors.white70),
          hintStyle: const TextStyle(color:
Colors.white60),
        ),
      ),
      initialRoute: '/',
      onGenerateRoute: (settings) {
        switch (settings.name) {
          case '/':
            return MaterialPageRoute(builder: (_) =>
const LoginScreen());
```

```
case '/register':
          return MaterialPageRoute(builder: (_) =>
const RegisterScreen());
        case '/main':
          return MaterialPageRoute(builder: (_) =>
const MainScreen());
        case '/home':
          return MaterialPageRoute(builder: (_) =>
const HomeScreen());
        case '/servers':
          return MaterialPageRoute(builder: (_) =>
const ServersScreen());
        case '/settings':
          return MaterialPageRoute(builder: (_) =>
const SettingsScreen());
        case '/profile':
          return MaterialPageRoute(builder: (_) =>
const ProfileScreen());
        case '/speedtest':
          return MaterialPageRoute(builder: (_) =>
const SpeedTestScreen());
        case '/vpn-details':
          return MaterialPageRoute(builder: (_) =>
const VPNDetailsScreen());
        case '/phone-login':
          return MaterialPageRoute(builder: (_) =>
const PhoneLoginScreen());
        case '/otp':
          if (settings.arguments != null) {
            final args = settings.arguments as
Map<String, String?>;
            return MaterialPageRoute(
              builder: (_) => OTPVerificationScreen(
                email: args['email'],
                phoneNumber: args['phoneNumber'],
              ),
            );
          }
          return MaterialPageRoute(builder: (_) =>
const LoginScreen());
        default:
          return MaterialPageRoute(builder: (_) =>
const LoginScreen());
      }
    },
    builder: (context, child) {
      return MediaQuery(
        data:
MediaQuery.of(context).copyWith(textScaleFactor:
1.0),
        child: child!,
      );
    },
  );
 }
}

// App Constants
class VPNConstants {
  static const appName = 'SecureShield VPN';
  static const appVersion = '1.0.0';

  // Colors
  static const primaryColor = Color(0xFF1A237E);
  static const accentColor = Colors.blue;

  // Animation Durations
  static const animationDuration =
Duration(milliseconds: 300);

  // Padding & Sizes
  static const defaultPadding = 16.0;
  static const borderRadius = 12.0;

  // Strings
  static const strConnected = 'Connected';
  static const strDisconnected = 'Not Connected';
  static const strConnecting = 'Connecting...';

  // Server List
  static const List<Map<String, dynamic>> servers =
[
    {'country': 'United States', 'flag': 'US', 'ping': 45},
    {'country': 'United Kingdom', 'flag': 'GB', 'ping':
65},
    {'country': 'Japan', 'flag': 'JP', 'ping': 85},
    {'country': 'Germany', 'flag': 'DE', 'ping': 55},
    {'country': 'Singapore', 'flag': 'SG', 'ping': 75},
  ];
}
```

**Login_screen.dart**

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

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

  @override
  State<LoginScreen> createState() =>
_LoginScreenState();
}

class _LoginScreenState extends
State<LoginScreen> {
  final _formKey = GlobalKey<FormState>();
  final _emailController = TextEditingController();
  final _passwordController =
TextEditingController();
  bool _isPasswordVisible = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        decoration: const BoxDecoration(
          gradient: LinearGradient(
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
            colors: [
              Color(0xFF1A237E),
              Color(0xFF0D47A1),
              Color(0xFF01579B),
            ],
          ),
        ),
        child: Stack(
          children: [
            Positioned(
              top: -100,
              right: -100,
              child: Container(
                width: 200,
                height: 200,
                decoration: BoxDecoration(
                  shape: BoxShape.circle,
                  color: Colors.white.withOpacity(0.1),
                ),
              ),
            ),
            Positioned(
              bottom: -50,
              left: -50,
              child: Container(
                width: 150,
                height: 150,
                decoration: BoxDecoration(
                  shape: BoxShape.circle,
                  color: Colors.white.withOpacity(0.1),
                ),
              ),
            ),
            SingleChildScrollView(
              child: Padding(
                padding: const EdgeInsets.all(20.0),
                child: Form(
                  key: _formKey,
                  child: Column(
                    mainAxisAlignment:
MainAxisAlignment.center,
                    children: [
                      const SizedBox(height: 80),
                      Container(
                        padding: const EdgeInsets.all(20),
                        decoration: BoxDecoration(
                          borderRadius:
BorderRadius.circular(30),
                          boxShadow: [
                            BoxShadow(
                              color:
Colors.blue.withOpacity(0.2),
                              blurRadius: 20,
                              spreadRadius: 5,
                            ),
                          ],
                        ),
                        child: Column(
                          children: [
                            Container(
                              padding: const EdgeInsets.all(16),
                              decoration: BoxDecoration(
                                color: Colors.white,
                                shape: BoxShape.circle,
                                boxShadow: [
                                  BoxShadow(
                                    color:
Colors.blue.withOpacity(0.5),
                                    blurRadius: 20,
                                    spreadRadius: 5,
                                  ),
                                ],
```

```
),
                    child: const Icon(
                     Icons.shield,
                     size: 80,
                     color: Colors.blue,
                    ),
                   ),
                   const SizedBox(height: 20),
                   ShaderMask(
                    shaderCallback: (bounds) =>
const LinearGradient(
                     colors: [Colors.white,
Colors.lightBlueAccent],
                    ).createShader(bounds),
                    child: const Text(
                     'SecureShield VPN',
                     style: TextStyle(
                      fontSize: 32,
                      fontWeight: FontWeight.bold,
                      color: Colors.white,
                     ),
                    ),
                   ),
                   const Text(
                    'Secure • Fast • Reliable',
                    style: TextStyle(
                     color: Colors.white70,
                     fontSize: 16,
                    ),
                   ),
                  ],
                 ),
                ),
                const SizedBox(height: 40),
                Container(
                 decoration: BoxDecoration(
                  color: Colors.white10,
                  borderRadius:
BorderRadius.circular(15),
                  border: Border.all(color:
Colors.white24),
                 ),
                 child: TextFormField(
                  controller: _emailController,
                  style: const TextStyle(color:
Colors.white),
                  decoration: InputDecoration(
                   labelText: 'Email',
                   labelStyle: const TextStyle(color:
Colors.white70),
                   border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(15),
                    borderSide: BorderSide.none,
                   ),
                   prefixIcon: const Icon(Icons.email,
color: Colors.white70),
                  ),
                  validator: (value) {
                   if (value?.isEmpty ?? true) {
                    return 'Please enter email';
                   }
                   return null;
                  },
                 ),
                ),
                const SizedBox(height: 20),
                Container(
                 decoration: BoxDecoration(
                  color: Colors.white10,
                  borderRadius:
BorderRadius.circular(15),
                  border: Border.all(color:
Colors.white24),
                 ),
                 child: TextFormField(
                  controller: _passwordController,
                  style: const TextStyle(color:
Colors.white),
                  obscureText: !_isPasswordVisible,
                  decoration: InputDecoration(
                   labelText: 'Password',
                   labelStyle: const TextStyle(color:
Colors.white70),
                   border: OutlineInputBorder(
                    borderRadius:
BorderRadius.circular(15),
                    borderSide: BorderSide.none,
                   ),
                   prefixIcon: const Icon(Icons.lock,
color: Colors.white70),
                   suffixIcon: IconButton(
                    icon: Icon(
                     _isPasswordVisible
                       ? Icons.visibility
                       : Icons.visibility_off,
                     color: Colors.white70,
                    ),
                    onPressed: () {
                     setState(() {
                      _isPasswordVisible =
!_isPasswordVisible;
                     });
                    },
```

```
                ),
              ),
              validator: (value) {
                if (value?.isEmpty ?? true) {
                  return 'Please enter password';
                }
                return null;
              },
            ),
          ),
          const SizedBox(height: 10),
          Align(
            alignment: Alignment.centerRight,
            child: TextButton(
              onPressed: () {
                // Add forgot password
functionality
              },
              child: const Text(
                'Forgot Password?',
                style: TextStyle(color:
Colors.white70),
              ),
            ),
          ),
          const SizedBox(height: 20),
          Container(
            decoration: BoxDecoration(
              gradient: const LinearGradient(
                colors: [Colors.blue,
Colors.lightBlue],
              ),
              borderRadius:
BorderRadius.circular(15),
              boxShadow: [
                BoxShadow(
                  color:
Colors.blue.withOpacity(0.3),
                  blurRadius: 10,
                  spreadRadius: 2,
                ),
              ],
            ),
            child: ElevatedButton(
              onPressed: () {
                if
(_formKey.currentState?.validate() ?? false) {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) =>
OTPVerificationScreen(
                        email: _emailController.text,
                      ),
                    ),
                  );
                }
              },
              style: ElevatedButton.styleFrom(
                backgroundColor:
Colors.transparent,
                shadowColor: Colors.transparent,
                minimumSize: const
Size.fromHeight(55),
                shape: RoundedRectangleBorder(
                  borderRadius:
BorderRadius.circular(15),
                ),
              ),
              child: const Text(
                'Login',
                style: TextStyle(
                  fontSize: 18,
                  fontWeight: FontWeight.bold,
                  color: Colors.white,
                ),
              ),
            ),
          ),
          const SizedBox(height: 20),
          Row(
            mainAxisAlignment:
MainAxisAlignment.center,
            children: [
              Container(
                height: 1,
                width: 50,
                color: Colors.white24,
              ),
              const Padding(
                padding:
EdgeInsets.symmetric(horizontal: 10),
                child: Text(
                  'Or continue with',
                  style: TextStyle(color:
Colors.white70),
                ),
              ),
              Container(
                height: 1,
                width: 50,
                color: Colors.white24,
```

```
),
                ],
              ),
              const SizedBox(height: 20),
              Row(
                mainAxisAlignment:
MainAxisAlignment.center,
                children: [
                  _buildSocialButton(
                    icon: Icons.phone,
                    onPressed: () {
                      // Add phone login functionality
                    },
                  ),
                  const SizedBox(width: 20),
                  _buildSocialButton(
                    icon: Icons.g_mobiledata,
                    onPressed: () {
                      // Add Google login functionality
                    },
                  ),
                ],
              ),
              const SizedBox(height: 20),
              Row(
                mainAxisAlignment:
MainAxisAlignment.center,
                children: [
                  const Text(
                    "Don't have an account? ",
                    style: TextStyle(color:
Colors.white70),
                  ),
                  TextButton(
                    onPressed: () {
                      Navigator.push(
                        context,
                        MaterialPageRoute(
                          builder: (context) => const
RegisterScreen(),
                        ),
                      );
                    },
                    child: const Text(
                      'Sign Up',
                      style: TextStyle(
                        color: Colors.white,
                        fontWeight: FontWeight.bold,
                      ),
                    ),
                  ),
                ],
                        ),
                      ],
                    ),
                  ),
                ),
              ),
            ],
          ),
        ),
      );
    }

    Widget _buildSocialButton({
      required IconData icon,
      required VoidCallback onPressed,
    }) {
      return Container(
        decoration: BoxDecoration(
          color: Colors.white10,
          borderRadius: BorderRadius.circular(15),
          border: Border.all(color: Colors.white24),
        ),
        child: IconButton(
          icon: Icon(icon, color: Colors.white),
          onPressed: onPressed,
        ),
      );
    }
}
```

**Speed_test_screen.dart**

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

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

  @override
  State<SpeedTestScreen> createState() =>
_SpeedTestScreenState();
}

class _SpeedTestScreenState extends
State<SpeedTestScreen> with
SingleTickerProviderStateMixin {
  late AnimationController _controller;
  double downloadSpeed = 0;
  double uploadSpeed = 0;
  double currentValue = 0;
  bool isTesting = false;
  Timer? _timer;

  @override
  void initState() {
    super.initState();
    _controller = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 2),
    );
  }

  void startSpeedTest() {
    setState(() {
      isTesting = true;
      downloadSpeed = 0;
      uploadSpeed = 0;
      currentValue = 0;
    });

    _timer = Timer.periodic(const
Duration(milliseconds: 100), (timer) {
      setState(() {
        if (currentValue < 140) {
          currentValue += 2;
          downloadSpeed = 94.999 * (currentValue /
140);
          uploadSpeed = 68.887 * (currentValue / 140);
        } else {
          _timer?.cancel();
          isTesting = false;
        }
      });
    });
  }

  @override
  void dispose() {
    _controller.dispose();
    _timer?.cancel();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.transparent,
      appBar: AppBar(
        title: const Text('Speed Test',
          style: TextStyle(fontSize: 24, fontWeight:
FontWeight.bold),
        ),
        backgroundColor: Colors.transparent,
        elevation: 0,
      ),
      body: Column(
        children: [
          const SizedBox(height: 20),
          _buildServerInfo(),
          const SizedBox(height: 40),
          _buildSpeedometer(),
          const SizedBox(height: 20),
          _buildSpeedResults(),
        ],
      ),
    );
  }

  Widget _buildServerInfo() {
    return Row(
      mainAxisAlignment:
MainAxisAlignment.spaceEvenly,
      children: [
        _buildInfoCard('Server', Icons.router, 'Local
Server', Colors.blue),
        _buildInfoCard('Location', Icons.location_on,
'Current', Colors.purple),
      ],
    );
  }

  Widget _buildInfoCard(String title, IconData icon,
String value, Color color) {
    return Container(
      padding: const EdgeInsets.all(16),
      decoration: BoxDecoration(
        color: Colors.white.withOpacity(0.1),
        borderRadius: BorderRadius.circular(15),
        border: Border.all(color: Colors.white24),
      ),
```

```
child: Column(
      children: [
       Icon(icon, color: color, size: 30),
       const SizedBox(height: 8),
       Text(title,
        style: const TextStyle(color: Colors.white70),
       ),
       Text(value,
        style: const TextStyle(
         color: Colors.white,
         fontWeight: FontWeight.bold,
        ),
       ),
      ],
     ),
    );
  }

  Widget _buildSpeedometer() {
   return SizedBox(
    height: 300,
    child: Stack(
     alignment: Alignment.center,
     children: [
      CustomPaint(
       painter: SpeedometerPainter(
        value: currentValue,
        isTesting: isTesting,
       ),
       size: const Size(300, 300),
      ),
      Column(
       mainAxisSize: MainAxisSize.min,
       children: [
        if (!isTesting && currentValue == 0)
         ElevatedButton(
          onPressed: startSpeedTest,
          style: ElevatedButton.styleFrom(
           backgroundColor: Colors.blue,
           shape: RoundedRectangleBorder(
            borderRadius:
BorderRadius.circular(20),
           ),
           padding: const EdgeInsets.symmetric(
            horizontal: 32,
            vertical: 16,
           ),
          ),
          child: const Text('Start Test'),
         ),
        if (isTesting || currentValue > 0)
         Text(
          '${currentValue.toStringAsFixed(1)}',
          style: const TextStyle(
           fontSize: 32,
           fontWeight: FontWeight.bold,
           color: Colors.white,
          ),
         ),
        ],
       ),
      ],
     ),
    );
   }

   Widget _buildSpeedResults() {
    return Row(
     mainAxisAlignment:
MainAxisAlignment.spaceEvenly,
     children: [
      _buildSpeedCard('Download', Icons.download,
downloadSpeed),
      _buildSpeedCard('Upload', Icons.upload,
uploadSpeed),
     ],
    );
   }

   Widget _buildSpeedCard(String type, IconData
icon, double speed) {
    return Container(
     padding: const EdgeInsets.all(16),
     decoration: BoxDecoration(
      color: Colors.white.withOpacity(0.1),
      borderRadius: BorderRadius.circular(15),
      border: Border.all(color: Colors.white24),
     ),
     child: Column(
      children: [
       Icon(icon, color: Colors.blue, size: 30),
       const SizedBox(height: 8),
       Text('$type Speed',
        style: const TextStyle(color: Colors.white70),
       ),
       Text(
        '${speed.toStringAsFixed(2)} Mbps',
        style: const TextStyle(
         color: Colors.white,
         fontWeight: FontWeight.bold,
         fontSize: 18,
        ),
       ),
      ],
     ),
    );
   }
}

class SpeedometerPainter extends CustomPainter {
```

```
final double value;
  final bool isTesting;

  SpeedometerPainter({required this.value, required
this.isTesting});

  @override
  void paint(Canvas canvas, Size size) {
    final center = Offset(size.width / 2, size.height /
2);
    final radius = min(size.width / 2, size.height / 2) -
20;

    final paint = Paint()
      ..style = PaintingStyle.stroke
      ..strokeWidth = 20;

    paint.color = Colors.grey.withOpacity(0.2);
    canvas.drawArc(
      Rect.fromCircle(center: center, radius: radius),
      pi,
      pi,
      false,
      paint,
    );

    final gradient = SweepGradient(
      startAngle: pi,
      endAngle: 2 * pi,
      colors: const [
        Colors.green,
        Colors.blue,
        Colors.purple,
        Colors.pink,
      ],
    );

    paint.shader = gradient.createShader(
      Rect.fromCircle(center: center, radius: radius),
    );

    final progressAngle = (value / 140) * pi;
    canvas.drawArc(
      Rect.fromCircle(center: center, radius: radius),
      pi,
      progressAngle,
      false,
      paint,
    );
  }

  @override
  bool shouldRepaint(SpeedometerPainter
oldDelegate) =>
      value != oldDelegate.value || isTesting !=
oldDelegate.isTesting;
  }
```

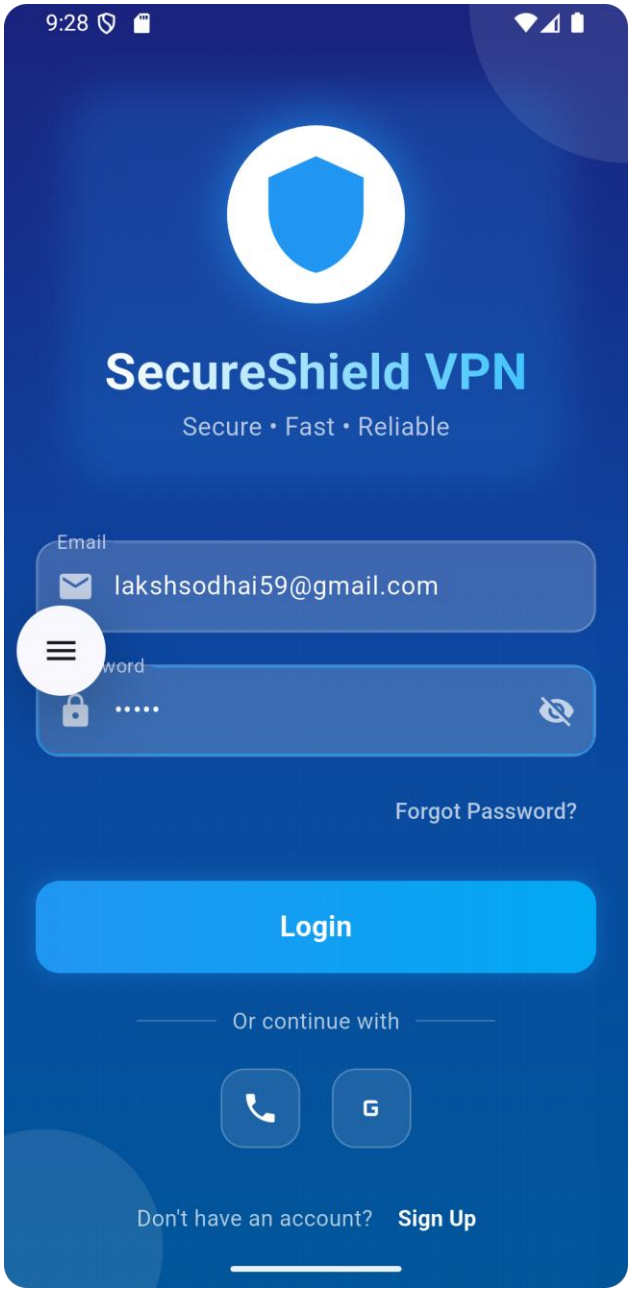
**servers_screen.dart**

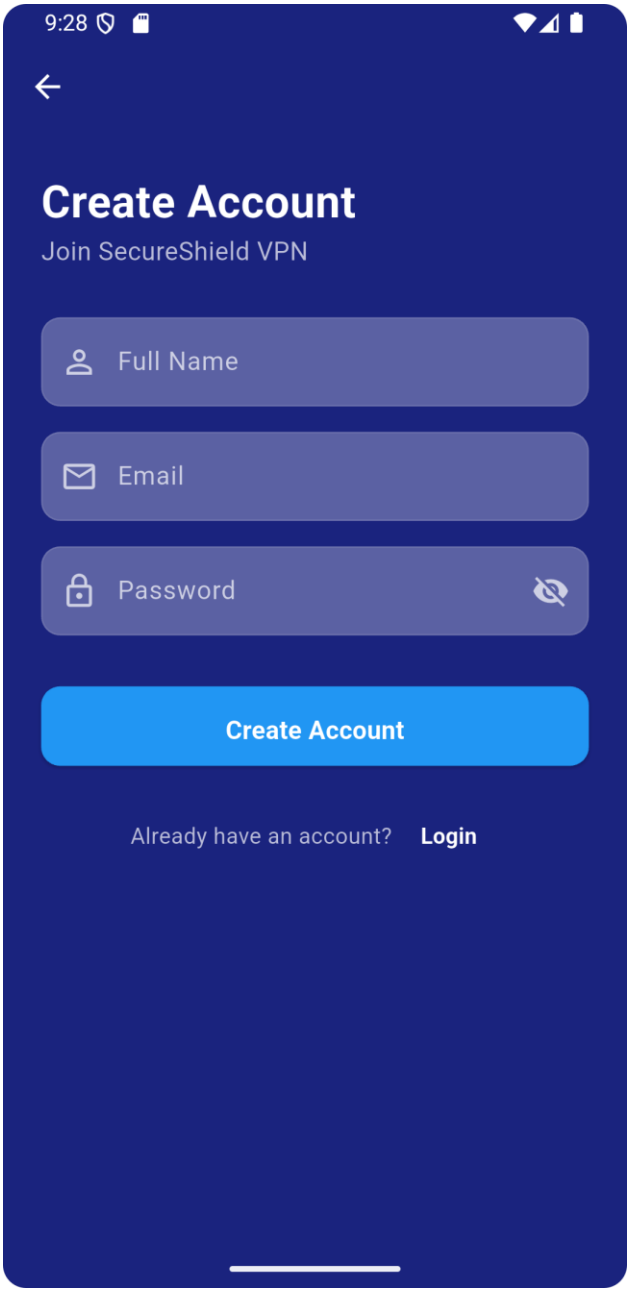
```
import 'package:flutter/material.dart';
import 'dart:math';
import 'dart:async';
class ServersScreen extends StatelessWidget {
  const ServersScreen({super.key});

  @override
  Widget build(BuildContext context) {
    final List<Map<String, dynamic>> servers =
[
      {"country": "United States", "flag": "US",
"ping": 45},
      {"country": "United Kingdom", "flag": "GB",
"ping": 65},
      {"country": "Japan", "flag": "JP", "ping":
85},
      {"country": "Germany", "flag": "DE", "ping":
55},
      {"country": "Singapore", "flag": "SG",
"ping": 75},
      {"country": "Canada", "flag": "CA", "ping":
50},
      {"country": "France", "flag": "FR", "ping":
60},
      {"country": "Australia", "flag": "AU", "ping":
90},
      {"country": "Netherlands", "flag": "NL",
"ping": 58},
      {"country": "India", "flag": "IN", "ping":
95},
    ];

    return Scaffold(
      backgroundColor: Colors.transparent,
      appBar: AppBar(
        title: const Text('Server Locations',
          style: TextStyle(fontSize: 24, fontWeight:
FontWeight.bold),
        ),
        backgroundColor: Colors.transparent,
        elevation: 0,
      ),
      body: ListView.builder(
        padding: const EdgeInsets.all(16),
        itemCount: servers.length,
        itemBuilder: (context, index) {
          final server = servers[index];
          return Container(
            margin: const EdgeInsets.only(bottom:
12),
            decoration: BoxDecoration(
              color: Colors.white.withOpacity(0.1),
              borderRadius:
BorderRadius.circular(15),
              border: Border.all(color:
Colors.white24),
              boxShadow: [
                BoxShadow(
                  color:
Colors.black.withOpacity(0.1),
                  blurRadius: 10,
                ),
              ],
            ),
            child: ListTile(
              contentPadding: const
EdgeInsets.all(16),
              leading: Text(
                server["flag"].toString(),
                style: const TextStyle(fontSize: 30),
              ),
              title: Text(
                server["country"].toString(),
                style: const TextStyle(
                  color: Colors.white,
                  fontWeight: FontWeight.bold,
                ),
              ),
              subtitle: Row(
                children: [
                  Icon(Icons.speed, size: 16, color:
Colors.orange.shade400),
                  const SizedBox(width: 4),
                  Text(
                    '${server["ping"]}ms',
                    style: TextStyle(color:
Colors.orange.shade400),
                  ),
                ],
              ),
              trailing: Container(
                padding: const EdgeInsets.all(8),
                decoration: BoxDecoration(
                  color: Colors.blue.withOpacity(0.2),
                  borderRadius:
BorderRadius.circular(10),
```

```
            ),

 child: const Icon(Icons.power_settings_new,
color: Colors.blue),
            ),
            onTap: () {

ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(
                  content: Text('Connecting to
${server["country"]}...'),
                  duration: const Duration(seconds:
1),
                ),
              );
            },
          ),
        );
      }
    }
```

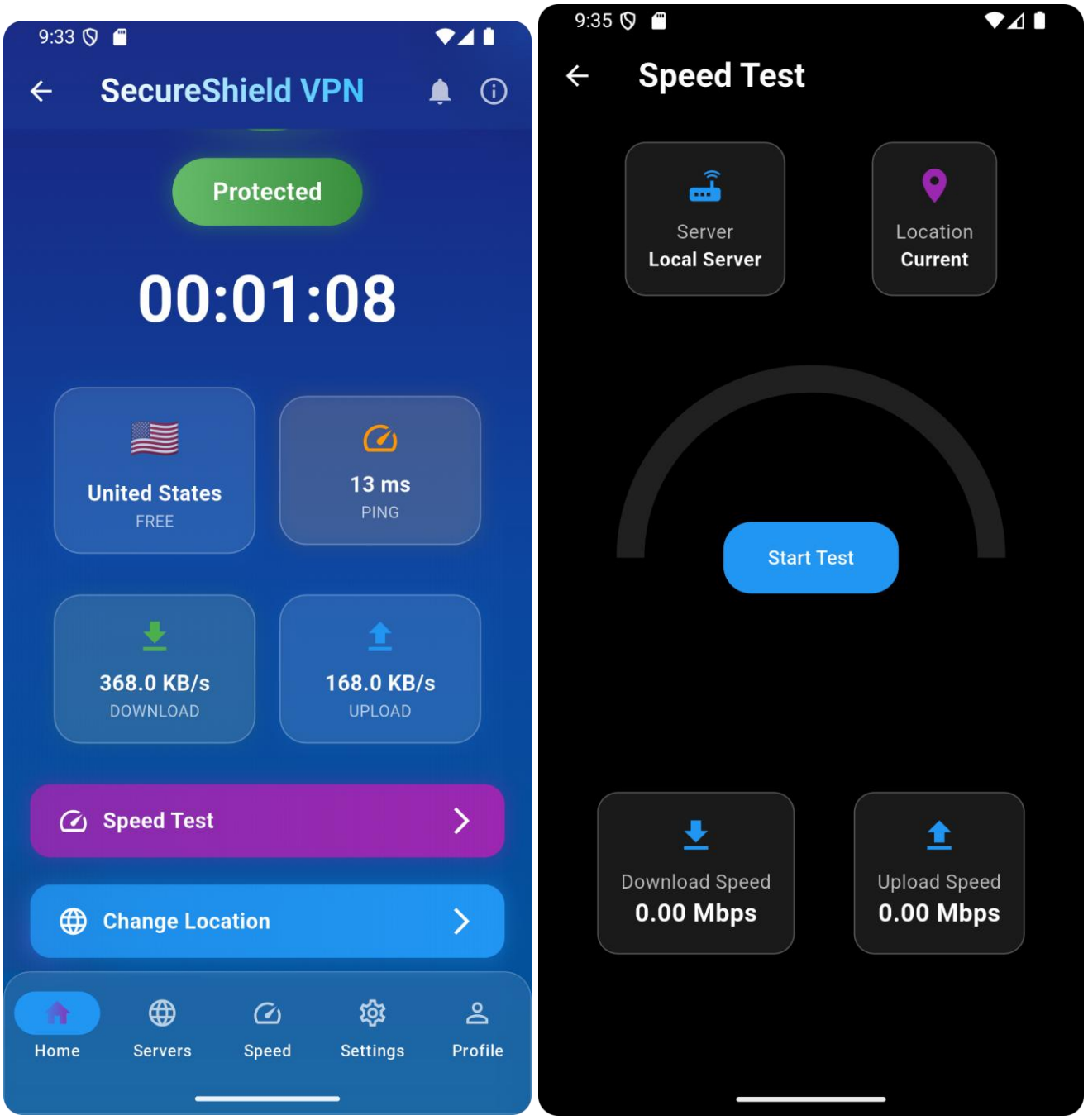
**Output-**

After clicking on Don’t have an account? It navigates to the registration page.



In home page, after clicking on Speed test it navigates to the speed test page.



In home page, after clicking on change location it navigates to the Server locations page.