## EXPERIMENT NO: - 03

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

**AIM: -** To include icons, images, fonts in Flutter app.

---

**Theory: -**

Flutter is a versatile open-source UI framework , which allows developers to build natively compiled applications for mobile, web, and desktop platforms from a single codebase. One of the key strengths of Flutter is its flexibility in creating highly customizable UIs. This practical focuses on incorporating essential visual elements—icons, images, and custom fonts—into a Flutter application. These elements enhance the visual appeal and usability of the app, providing an engaging experience for users.

A flutter app when built has both assets (resources) and code. Assets are available and deployed during runtime. The asset is a file that can include static data, configuration files, icons, and images. The Flutter app supports many image formats, such as JPEG, WebP, PNG, GIF, animated WebP/GIF, BMP, and WBMP.

Visual elements play a significant role in app development.

- **Enhanced User Experience**: Images and icons make your app visually appealing and user-friendly.
- **Information Conveyance**: They convey information quickly and intuitively. A well-chosen icon can replace lengthy text.
- **Branding**: Custom icons and images reinforce your app's branding, making it memorable.

### ➢ Adding Icons in Flutter

Flutter provides built-in material design icons through the Icons class. Custom icons can also be added using third-party packages such as flutter_launcher_icons and font_awesome_flutter.

```
Icon(
  Icons.home,
  size: 40,
);
```

## ➢ Adding Images in Flutter

Flutter supports images from three sources:

1. **Assets** (Stored locally in the project)

   - Place the image inside the assets/images folder in the project.
   - Declare the image in pubspec.yaml

     ```
     flutter:
       assets:
         - assets/images/sample.png
     ```

   - Display the image in the app

     ```
     Image.asset('assets/images/sample.png');
     ```

2. **Network** (Fetched from the internet)

   Displaying images from the internet or network is very simple. Flutter provides a built-in method Image.network to work with images from a URL. The Image.network method also allows you to use some optional properties, such as height, width, color, fit, and many more.

   ```
   Image.network('https://example.com/sample.jpg');
   ```

3. **Memory or File** (Stored on the device)

## ➢ Adding Custom Fonts in Flutter

By default, Flutter uses the Roboto font, but custom fonts can be added for a unique UI.

- Download the font and place it in the assets/fonts/ folder.
- Declare the font in pubspec.yaml
- Use the font in the app

  ```
  Text(
    'Custom Font Example',
    style: TextStyle(fontFamily: 'CustomFont', fontSize: 24),
  );
  ```

Code: -

**Home_screen.dart**

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

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

  @override
  State<HomeScreen> createState() =>
_HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen>
{
  bool isConnected = false;
  String timerText = "00:00:00";
  Timer? _timer;
  int _seconds = 0;
  double downloadSpeed = 0;
  double uploadSpeed = 0;
  int ping = 0;

  void startTimer() {
    _timer = Timer.periodic(const Duration(seconds: 1),
(timer) {
      setState(() {
        _seconds++;
        int hours = _seconds ~/ 3600;
        int minutes = (_seconds % 3600) ~/ 60;
        int seconds = _seconds % 60;
        timerText = '${hours.toString().padLeft(2,
'0')}:${minutes.toString().padLeft(2,
'0')}:${seconds.toString().padLeft(2, '0')}';
        downloadSpeed = (300 + (_seconds %
100)).toDouble();
        uploadSpeed = (150 + (_seconds %
50)).toDouble();
        ping = 5 + (_seconds % 10);
      });
    });
  }

  void stopTimer() {
    _timer?.cancel();
    setState(() {
      _seconds = 0;
      timerText = "00:00:00";
      downloadSpeed = 0;
      uploadSpeed = 0;
      ping = 0;
    });
  }

  void toggleConnection() {
    setState(() {
      isConnected = !isConnected;
      if (isConnected) {
        startTimer();
      } else {
        stopTimer();
      }
    });
  }

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.transparent,
      appBar: _buildAppBar(),
      body: SingleChildScrollView(
        child: Padding(
          padding: const
EdgeInsets.symmetric(horizontal: 20),
          child: Column(
            children: [
              const SizedBox(height: 30),
              _buildConnectionButton(),
              const SizedBox(height: 20),
              _buildConnectionStatus(),
              const SizedBox(height: 20),
              _buildTimer(),
              const SizedBox(height: 30),
              _buildLocationAndPing(),
              const SizedBox(height: 30),
              _buildSpeedMetrics(),
              const SizedBox(height: 30),
              _buildActionButtons(),
            ],
          ),
        ),
      ),
    );
  }

  PreferredSizeWidget _buildAppBar() {
    return AppBar(
      title: ShaderMask(
        shaderCallback: (bounds) => const
LinearGradient(
          colors: [Colors.white, Colors.lightBlueAccent],
        ).createShader(bounds),
```

```
  child: const Text(
        'SecureShield VPN',
        style: TextStyle(
          fontSize: 24,
          fontWeight: FontWeight.bold,
          color: Colors.white,
        ),
      ),
    ),
    backgroundColor: Colors.transparent,
    elevation: 0,
    actions: [
      Icon(Icons.notifications, color:
Colors.white.withOpacity(0.7)),
      const SizedBox(width: 16),
      Icon(Icons.info_outline, color:
Colors.white.withOpacity(0.7)),
      const SizedBox(width: 16),
    ],
  );
 }

 Widget _buildConnectionButton() {
  return GestureDetector(
    onTap: toggleConnection,
    child: Container(
      width: 200,
      height: 200,
      decoration: BoxDecoration(
        shape: BoxShape.circle,
        gradient: LinearGradient(
          begin: Alignment.topLeft,
          end: Alignment.bottomRight,
          colors: isConnected
              ? [Colors.green.shade400,
Colors.green.shade700]
              : [Colors.blue.shade400,
Colors.blue.shade700],
        ),
        boxShadow: [
          BoxShadow(
            color: (isConnected ? Colors.green :
Colors.blue).withOpacity(0.3),
            blurRadius: 20,
            spreadRadius: 5,
          ),
        ],
      ),
      child: Column(
        mainAxisAlignment:
MainAxisAlignment.center,
        children: [
          Icon(
            isConnected ? Icons.power_settings_new :
Icons.power_off,
size: 50,
            color: Colors.white,
          ),
          const SizedBox(height: 10),
          Text(
            isConnected ? 'Connected' : 'Tap to Connect',
            style: const TextStyle(
              color: Colors.white,
              fontSize: 16,
              fontWeight: FontWeight.bold,
            ),
          ),
        ],
      ),
    ),
  );
 }

 Widget _buildConnectionStatus() {
  return Container(
    padding: const EdgeInsets.symmetric(horizontal:
30, vertical: 12),
    decoration: BoxDecoration(
      gradient: LinearGradient(
        colors: isConnected
            ? [Colors.green.shade400,
Colors.green.shade700]
            : [Colors.blue.shade400,
Colors.blue.shade700],
      ),
      borderRadius: BorderRadius.circular(30),
      boxShadow: [
        BoxShadow(
          color: (isConnected ? Colors.green :
Colors.blue).withOpacity(0.3),
          blurRadius: 10,
          spreadRadius: 2,
        ),
      ],
    ),
    child: Text(
      isConnected ? 'Protected' : 'Not Protected',
      style: const TextStyle(
        color: Colors.white,
        fontSize: 18,
        fontWeight: FontWeight.bold,
      ),
    ),
  );
 }

 Widget _buildTimer() {
  return Text(
    timerText,
    style: const TextStyle(
```

```
 fontSize: 48,
       fontWeight: FontWeight.bold,
       color: Colors.white,
     ),
   );
 }

 Widget _buildLocationAndPing() {
   return Row(
     mainAxisAlignment:
MainAxisAlignment.spaceEvenly,
     children: [
       _buildMetricCard(
         'United States',
         'US',
         'FREE',
         Colors.blue,
       ),
       _buildMetricCard(
         '$ping ms',
         Icons.speed,
         'PING',
         Colors.orange,
       ),
     ],
   );
 }

 Widget _buildSpeedMetrics() {
   return Row(
     mainAxisAlignment:
MainAxisAlignment.spaceEvenly,
     children: [
       _buildMetricCard(
         '${downloadSpeed.toStringAsFixed(1)} KB/s',
         Icons.download,
         'DOWNLOAD',
         Colors.green,
       ),
       _buildMetricCard(
         '${uploadSpeed.toStringAsFixed(1)} KB/s',
         Icons.upload,
         'UPLOAD',
         Colors.blue,
       ),
     ],
   );
 }

 Widget _buildMetricCard(String value, dynamic
icon, String label, Color color) {
   return Container(
     width: 150,
     padding: const EdgeInsets.all(15),
     decoration: BoxDecoration(
       color: Colors.white.withOpacity(0.1),
       borderRadius: BorderRadius.circular(20),
       border: Border.all(color: Colors.white24),
       boxShadow: [
         BoxShadow(
           color: color.withOpacity(0.1),
           blurRadius: 10,
           spreadRadius: 2,
         ),
       ],
     ),
     child: Column(
       children: [
         icon is IconData
             ? Icon(icon, color: color, size: 30)
             : Text(icon, style: const TextStyle(fontSize:
30)),
         const SizedBox(height: 8),
         Text(
           value,
           style: const TextStyle(
             color: Colors.white,
             fontSize: 16,
             fontWeight: FontWeight.bold,
           ),
         ),
         Text(
           label,
           style: TextStyle(
             color: Colors.white.withOpacity(0.7),
             fontSize: 12,
           ),
         ),
       ],
     ),
   );
 }

 Widget _buildActionButtons() {
   return Column(
     children: [
       _buildGradientButton(
         'Speed Test',
         Icons.speed,
         Colors.purple,
             () => Navigator.pushNamed(context,
'/speedtest'),
       ),
       const SizedBox(height: 10),
       _buildGradientButton(
         'Change Location',
         Icons.language,
         Colors.blue,
             () => Navigator.pushNamed(context,
'/servers'),
```

```
),
    ],
  );
 }

 Widget _buildGradientButton(
    String text,
    IconData icon,
    Color color,
    VoidCallback onTap,
    ) {
  return Container(
    margin: const EdgeInsets.only(bottom: 10),
    decoration: BoxDecoration(
      gradient: LinearGradient(
        colors: [color.withOpacity(0.8), color],
      ),
      borderRadius: BorderRadius.circular(15),
      boxShadow: [
        BoxShadow(
          color: color.withOpacity(0.3),
          blurRadius: 10,
          spreadRadius: 2,
        ),
      ],
    ),
    child: Material(
      color: Colors.transparent,
      child: InkWell(
        onTap: onTap,
        borderRadius: BorderRadius.circular(15),
```

**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),
            child: Padding(
              padding: const
EdgeInsets.symmetric(horizontal: 20, vertical: 15),
              child: Row(
                mainAxisAlignment:
MainAxisAlignment.spaceBetween,
                children: [
                  Row(
                    children: [
                      Icon(icon, color: Colors.white),
                      const SizedBox(width: 10),
                      Text(
                        text,
                        style: const TextStyle(
                          color: Colors.white,
                          fontSize: 16,
                          fontWeight: FontWeight.bold,
                        ),
                      ),
                    ],
                  ),
                  const Icon(Icons.arrow_forward_ios, color:
Colors.white),
                ],
              ),
            ),
          ),
        );
      }
```

```
            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),
                ),
              );
            },
          ),
        );
```

**Vpn_details_screen.dart**

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

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const Color(0xFF1A237E),
      appBar: AppBar(
        backgroundColor: Colors.transparent,
        elevation: 0,
        title: const Text('VPN Details'),
        leading: IconButton(
          icon: const Icon(Icons.arrow_back),
          onPressed: () => Navigator.pop(context),
        ),
      ),
      body: SingleChildScrollView(
        padding: const EdgeInsets.all(16),
        child: Column(
          children: [
            // Connection Details Card
            Container(
              padding: const EdgeInsets.all(16),
              decoration: BoxDecoration(
                color: Colors.white.withOpacity(0.1),
                borderRadius: BorderRadius.circular(12),
              ),
              child: Column(
                crossAxisAlignment:
CrossAxisAlignment.start,
                children: [
                  const Text(
                    'Connection Details',
                    style: TextStyle(
                      color: Colors.white,
                      fontSize: 20,
                      fontWeight: FontWeight.bold,
                    ),
                  ),
                  const SizedBox(height: 16),
                  _buildDetailRow('IP Address',
'192.168.1.1'),
                  _buildDetailRow('Protocol', 'UDP'),
                  _buildDetailRow('Port', '1194'),
                  _buildDetailRow('Encryption', 'AES-256-
GCM'),
                ],
              ),
            ),
            const SizedBox(height: 16),

            // Server Stats Card
            Container(
              padding: const EdgeInsets.all(16),
              decoration: BoxDecoration(
                color: Colors.white.withOpacity(0.1),
                borderRadius: BorderRadius.circular(12),
              ),
              child: Column(
                crossAxisAlignment:
CrossAxisAlignment.start,
                children: [
                  const Text(
                    'Server Stats',
                    style: TextStyle(
                      color: Colors.white,
                      fontSize: 20,
                      fontWeight: FontWeight.bold,
                    ),
                  ),
                  const SizedBox(height: 16),
                  _buildStatRow(
                    icon: Icons.speed,
                    title: 'Server Load',
                    value: '45%',
                    color: Colors.orange,
                  ),
                  _buildStatRow(
                    icon: Icons.people,
                    title: 'Active Users',
                    value: '1,234',
                    color: Colors.blue,
                  ),
                  _buildStatRow(
                    icon: Icons.network_check,
                    title: 'Uptime',
                    value: '99.9%',
                    color: Colors.green,
                  ),
                ],
              ),
            ),
            const SizedBox(height: 16),

            // Network Info Card
            Container(
              padding: const EdgeInsets.all(16),
              decoration: BoxDecoration(
                color: Colors.white.withOpacity(0.1),
                borderRadius: BorderRadius.circular(12),
              ),
              child: Column(
                crossAxisAlignment:
CrossAxisAlignment.start,
                children: [
                  const Text(
```

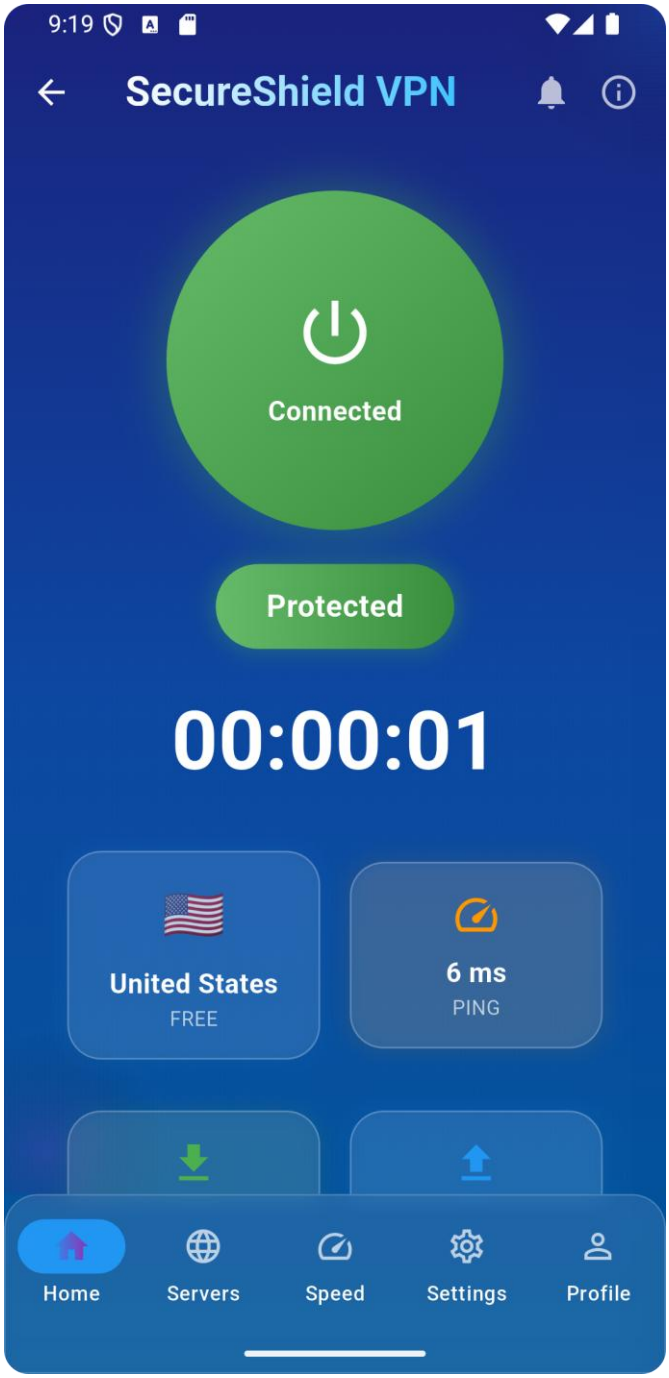
```
'Network Information',
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 20,
                  fontWeight: FontWeight.bold,
                ),
              ),
              const SizedBox(height: 16),
              _buildNetworkInfo(
                'Download Speed',
                '120 Mbps',
                Icons.download,
                Colors.green,
              ),
              _buildNetworkInfo(
                'Upload Speed',
                '85 Mbps',
                Icons.upload,
                Colors.blue,
              ),
              _buildNetworkInfo(
                'Latency',
                '45 ms',
                Icons.timer,
                Colors.orange,
              ),
            ],
          ),
        ),
        const SizedBox(height: 16),

        // Usage Stats Card
        Container(
          padding: const EdgeInsets.all(16),
          decoration: BoxDecoration(
            color: Colors.white.withOpacity(0.1),
            borderRadius: BorderRadius.circular(12),
          ),
          child: Column(
            crossAxisAlignment:
CrossAxisAlignment.start,
            children: [
              const Text(
                'Usage Statistics',
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 20,
                  fontWeight: FontWeight.bold,
                ),
              ),
              const SizedBox(height: 16),
              _buildUsageStats('Data Used', '1.5 GB'),
              _buildUsageStats('Time Connected', '2h
30m'),
              _buildUsageStats('Sessions Today', '3'),
            ],
          ),
        ),
      ],
    ),
  );
}

Widget _buildDetailRow(String label, String value) {
  return Padding(
    padding: const EdgeInsets.symmetric(vertical: 8),
    child: Row(
      mainAxisAlignment:
MainAxisAlignment.spaceBetween,
      children: [
        Text(
          label,
          style: const TextStyle(color: Colors.white70),
        ),
        Text(
          value,
          style: const TextStyle(color: Colors.white),
        ),
      ],
    ),
  );
}

Widget _buildStatRow({
  required IconData icon,
  required String title,
  required String value,
  required Color color,
}) {
  return Padding(
    padding: const EdgeInsets.symmetric(vertical: 8),
    child: Row(
      children: [
        Container(
          padding: const EdgeInsets.all(8),
          decoration: BoxDecoration(
            color: color.withOpacity(0.2),
            borderRadius: BorderRadius.circular(8),
          ),
          child: Icon(icon, color: color, size: 20),
        ),
        const SizedBox(width: 12),
        Text(
          title,
          style: const TextStyle(color: Colors.white70),
        ),
        const Spacer(),
        Text(
```

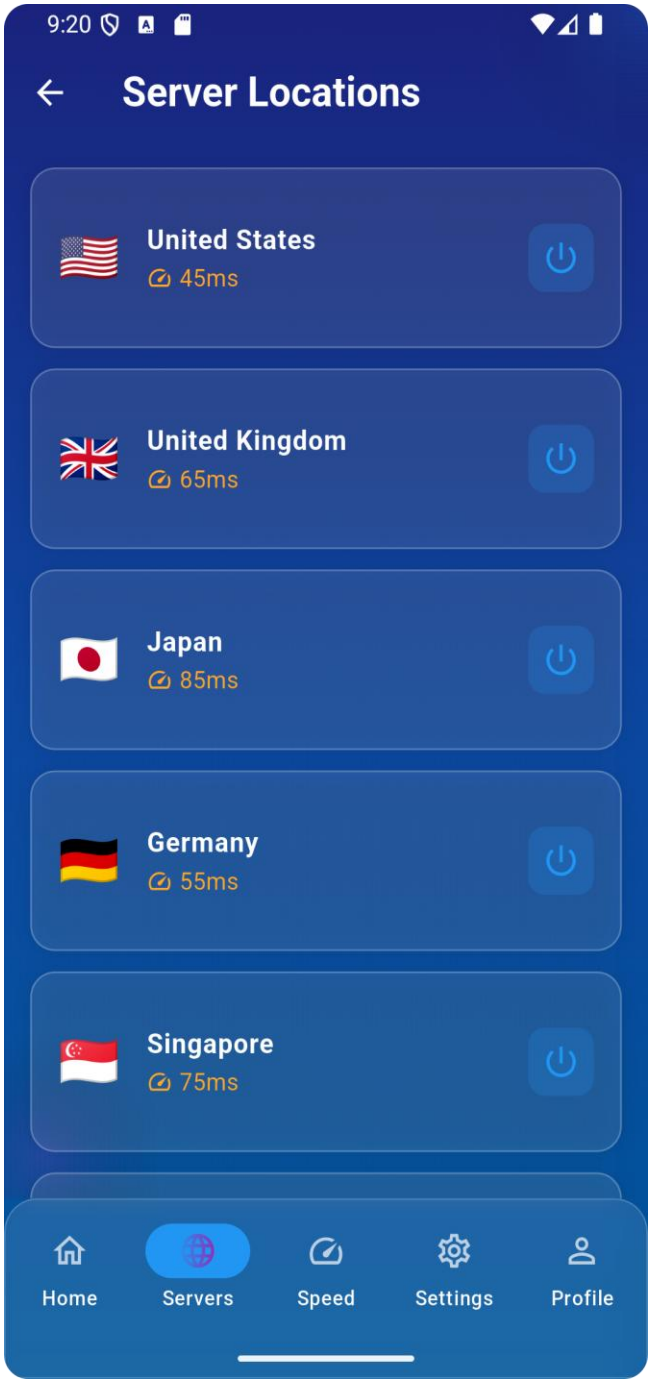
```
 value,
          style: const TextStyle(

 color: Colors.white,
            fontWeight: FontWeight.bold,
          ),
        ),
      ],
    ),
  );
 }

  Widget _buildNetworkInfo(String title, String value,
IconData icon, Color color) {
   return Padding(
     padding: const EdgeInsets.symmetric(vertical: 8),
     child: Row(
       children: [
         Icon(icon, color: color),
         const SizedBox(width: 12),
         Column(
           crossAxisAlignment:
CrossAxisAlignment.start,
           children: [
             Text(
               title,
               style: const TextStyle(color:
Colors.white70),
             ),
             Text(
               value,
               style: const TextStyle(
                 color: Colors.white,
                 fontWeight: FontWeight.bold,
               ),
             ),
           ],
         ),
       ],
     ),
   );
 }

  Widget _buildUsageStats(String label, String value) {
   return Padding(
     padding: const EdgeInsets.symmetric(vertical: 8),
     child: Row(
       mainAxisAlignment:
MainAxisAlignment.spaceBetween,
       children: [
         Text(
           label,
           style: const TextStyle(color: Colors.white70),
         ),
         Text(
```

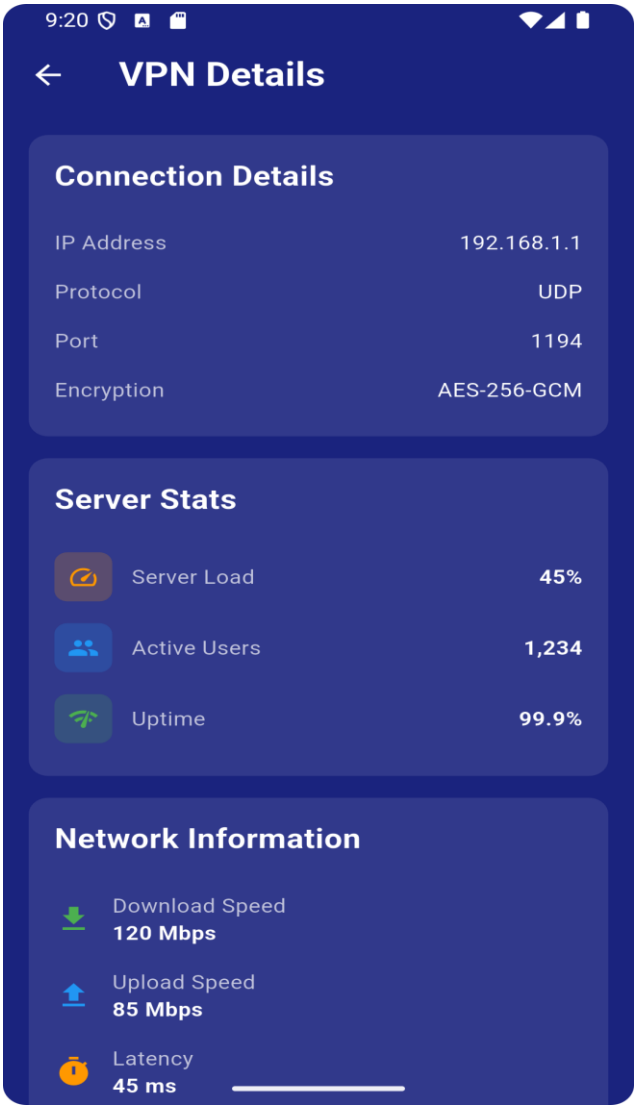
```
 value,
          style: const TextStyle(
            color: Colors.white,
            fontWeight: FontWeight.bold,
          ),

),
      ],
    ),
  );
 }
}
```

**OUTPUT: -**

v


9:19
SecureShield VPN
Connected
Protected
00:00:01
United States
FREE
6 ms
PING
Home
Servers
Speed
Settings
Profile


9:20
Server Locations
United States
45ms
United Kingdom
65ms
Japan
85ms
Germany
55ms
Singapore
75ms
Home
Servers
Speed
Settings
Profile


9:20
VPN Details
Connection Details
IP Address
192.168.1.1
Protocol
UDP
Port
1194
Encryption
AES-256-GCM
Server Stats
Server Load
45%
Active Users
1,234
Uptime
99.9%
Network Information
Download Speed
120 Mbps
Upload Speed
85 Mbps
Latency
45 ms