EXPERIMENT NO: - 05

Name:- Manav Punjabi **Class:-** D15A **Roll:No: -** 44

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';

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Chat App',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: const LoginScreen(),
    );
  }
}
```

Login_page.dart:-

```
import 'package:flutter/material.dart';
import 'home_screen.dart';
import 'signup_screen.dart';  // Added this import

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

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

class _LoginScreenState extends State<LoginScreen>
{
  bool isLogin = true;
  final _emailController = TextEditingController();
  final _passwordController = TextEditingController();

  @override
  void dispose() {
    _emailController.dispose();
    _passwordController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SafeArea(
        child: SingleChildScrollView(
          child: Padding(
            padding: const EdgeInsets.all(16.0),
            child: Column(
              mainAxisAlignment:
MainAxisAlignment.center,
              children: [
                const SizedBox(height: 50),
                // App Logo
                Container(
                  height: 100,
                  width: 100,
                  decoration: BoxDecoration(
                    color: Colors.blue,
                    borderRadius: BorderRadius.circular(20),
                  ),
                  child: const Icon(
                    Icons.chat,
                    size: 60,
                    color: Colors.white,
                  ),
                ),
                const SizedBox(height: 30),

                // Welcome Text
                Text(
                  isLogin ? 'Welcome Back' : 'Create
Account',
                  style: const TextStyle(
                    fontSize: 28,
                    fontWeight: FontWeight.bold,
                  ),
                ),
                const SizedBox(height: 10),
                Text(
                  isLogin ? 'Login to continue' : 'Sign up to
get started',
                  style: TextStyle(
                    fontSize: 16,
                    color: Colors.grey[600],
                  ),
                ),
                const SizedBox(height: 30),

                // Email Field
                TextField(
                  controller: _emailController,
                  decoration: InputDecoration(
                    labelText: 'Email',
                    prefixIcon: const Icon(Icons.email),
                    border: OutlineInputBorder(
                      borderRadius:
BorderRadius.circular(12),
                    ),
                    enabledBorder: OutlineInputBorder(
                      borderRadius:
BorderRadius.circular(12),
                      borderSide: const BorderSide(color:
Colors.grey),
                    ),
                    focusedBorder: OutlineInputBorder(
                      borderRadius:
BorderRadius.circular(12),
                      borderSide: const BorderSide(color:
Colors.blue),
                    ),
                  ),
                  keyboardType:
TextInputType.emailAddress,
                ),
                const SizedBox(height: 16),

                // Password Field
                TextField(
                  controller: _passwordController,
                  decoration: InputDecoration(
                    labelText: 'Password',
                    prefixIcon: const Icon(Icons.lock),
                    border: OutlineInputBorder(
```

```
                borderRadius:
BorderRadius.circular(12),
              ),
              enabledBorder: OutlineInputBorder(
                borderRadius:
BorderRadius.circular(12),
                borderSide: const BorderSide(color:
Colors.grey),
              ),
              focusedBorder: OutlineInputBorder(
                borderRadius:
BorderRadius.circular(12),
                borderSide: const BorderSide(color:
Colors.blue),
              ),
            ),
            obscureText: true,
          ),
          const SizedBox(height: 20),

          // Forgot Password Button
          if (isLogin)
            Align(
              alignment: Alignment.centerRight,
              child: TextButton(
                onPressed: () {
                  // Forgot password logic
                },
                child: const Text('Forgot Password?'),
              ),
            ),
          const SizedBox(height: 20),

          // Login/Signup Button
          SizedBox(
            width: double.infinity,
            height: 50,
            child: ElevatedButton(
              onPressed: () {
                // Login/Signup logic ke baad
                Navigator.pushReplacement(
                  context,
                  MaterialPageRoute(builder: (context)
=> const HomeScreen()),
                );
              },
              style: ElevatedButton.styleFrom(
                shape: RoundedRectangleBorder(
                  borderRadius:
BorderRadius.circular(12),
                ),
              ),
              child: Text(
                isLogin ? 'Login' : 'Sign Up',
                style: const TextStyle(fontSize: 16),
              ),
            ),
          ),
          const SizedBox(height: 20),

          // Toggle Login/Signup
          Row(
            mainAxisAlignment:
MainAxisAlignment.center,
            children: [
              Text(
                isLogin ? 'Don\'t have an account?' :
'Already have an account?',
                style: TextStyle(color:
Colors.grey[600]),
              ),
              TextButton(
                onPressed: () {
                  if (isLogin) {
                    // If on login, go to signup screen
                    Navigator.push(
                      context,
                      MaterialPageRoute(builder:
(context) => const SignupScreen()),
                    );
                  } else {
                    // If seeing signup text, toggle back to
login
                    setState(() {
                      isLogin = !isLogin;
                    });
                  }
                },
                child: Text(
                  isLogin ? 'Sign Up' : 'Login',
                  style: const TextStyle(fontWeight:
FontWeight.bold),
                ),
              ),
            ],
          ),
        ],
      ),
    ),
  );
 }
}
```

profile_screen.dart

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

import 'home_screen.dart';

import 'signup_screen.dart';  // Added this import


class LoginScreen extends StatefulWidget {

 const LoginScreen({super.key});


 @override

  State<LoginScreen> createState() => _LoginScreenState();

}


class _LoginScreenState extends State<LoginScreen> {

 bool isLogin = true;

        final _emailController = TextEditingController();

       final _passwordController = TextEditingController();


 @override

 void dispose() {

  _emailController.dispose();

  _passwordController.dispose();

  super.dispose();

 }


 @override

 Widget build(BuildContext context) {

  return Scaffold(

    body: SafeArea(

     child: SingleChildScrollView(

      child: Padding(

                    padding: const EdgeInsets.all(16.0),

        child: Column(

                  mainAxisAlignment: MainAxisAlignment.center,

          children: [

           const SizedBox(height: 50),

           // App Logo

           Container(

            height: 100,

            width: 100,

            decoration: BoxDecoration(

             color: Colors.blue,

                        borderRadius: BorderRadius.circular(20),

            ),

            child: const Icon(

             Icons.chat,

             size: 60,

             color: Colors.white,

            ),

           ),

           const SizedBox(height: 30),


           // Welcome Text

           Text(

             isLogin ? 'Welcome Back' : 'Create Account',

            style: const TextStyle(

             fontSize: 28,
```

```
                fontWeight:
FontWeight.bold,
              ),
            ),
            const SizedBox(height: 10),
            Text(
                isLogin ? 'Login to
continue' : 'Sign up to get started',
              style: TextStyle(
                fontSize: 16,
                color: Colors.grey[600],
              ),
            ),
            const SizedBox(height: 30),

            // Email Field
            TextField(
                controller:
_emailController,
                decoration:
InputDecoration(
              labelText: 'Email',
                prefixIcon: const
Icon(Icons.email),
                border:
OutlineInputBorder(
                borderRadius:
BorderRadius.circular(12),
              ),
                enabledBorder:
OutlineInputBorder(
                borderRadius:
BorderRadius.circular(12),
                borderSide: const
BorderSide(color: Colors.grey),
              ),
                focusedBorder:
OutlineInputBorder(
                borderRadius:
BorderRadius.circular(12),
                borderSide: const
BorderSide(color: Colors.blue),
              ),
            ),
                keyboardType:
TextInputType.emailAddress,
            ),
            const SizedBox(height: 16),

            // Password Field
            TextField(
                controller:
_passwordController,
                decoration:
InputDecoration(
              labelText: 'Password',
                prefixIcon: const
Icon(Icons.lock),
                border:
OutlineInputBorder(
                borderRadius:
BorderRadius.circular(12),
              ),
                enabledBorder:
OutlineInputBorder(
                borderRadius:
BorderRadius.circular(12),
                borderSide: const
BorderSide(color: Colors.grey),
              ),
                focusedBorder:
```

```
OutlineInputBorder(
                        borderRadius:
BorderRadius.circular(12),
                      borderSide: const
BorderSide(color: Colors.blue),
                ),
              ),
              obscureText: true,
            ),
            const SizedBox(height: 20),

            // Forgot Password Button
            if (isLogin)
              Align(
                          alignment:
Alignment.centerRight,
                child: TextButton(
                  onPressed: () {
                    // Forgot password logic
                  },
                    child: const Text('Forgot
Password?'),
                ),
              ),
            const SizedBox(height: 20),

            // Login/Signup Button
            SizedBox(
              width: double.infinity,
              height: 50,
              child: ElevatedButton(
                onPressed: () {
                  // Login/Signup logic ke
baad

                  Navigator.pushReplacement(
                    context,

MaterialPageRoute(builder:  (context)
=> const HomeScreen()),
                  );
                },
                          style:
ElevatedButton.styleFrom(
                          shape:
RoundedRectangleBorder(
                      borderRadius:
BorderRadius.circular(12),
                    ),
                  ),
                  child: Text(
                    isLogin ? 'Login' : 'Sign
Up',
                          style:  const
TextStyle(fontSize: 16),
                  ),
                ),
              ),
            const SizedBox(height: 20),

            // Toggle Login/Signup
            Row(
                    mainAxisAlignment:
MainAxisAlignment.center,
              children: [
                Text(
                  isLogin ? 'Don\'t have an
account?' : 'Already have an account?',
```

```
                    style: TextStyle(color:
Colors.grey[600]),
                ),
                TextButton(
                  onPressed: () {
                    if (isLogin) {
                          // If on login, go to
signup screen
                      Navigator.push(
                        context,

MaterialPageRoute(builder:    (context)
=> const SignupScreen()),
                      );
                    } else {
                      // If seeing signup text,
toggle back to login
                      setState(() {
                        isLogin = !isLogin;
                      });
                    }
                  },
                  child: Text(
                        isLogin ? 'Sign Up' :
'Login',
                                  style:  const
TextStyle(fontWeight:
FontWeight.bold),
                  ),
                ),
              ],
            ),
          ],
        ),
                ),
              ),
            ),
          );
        }
      }
```

search_screen.dart

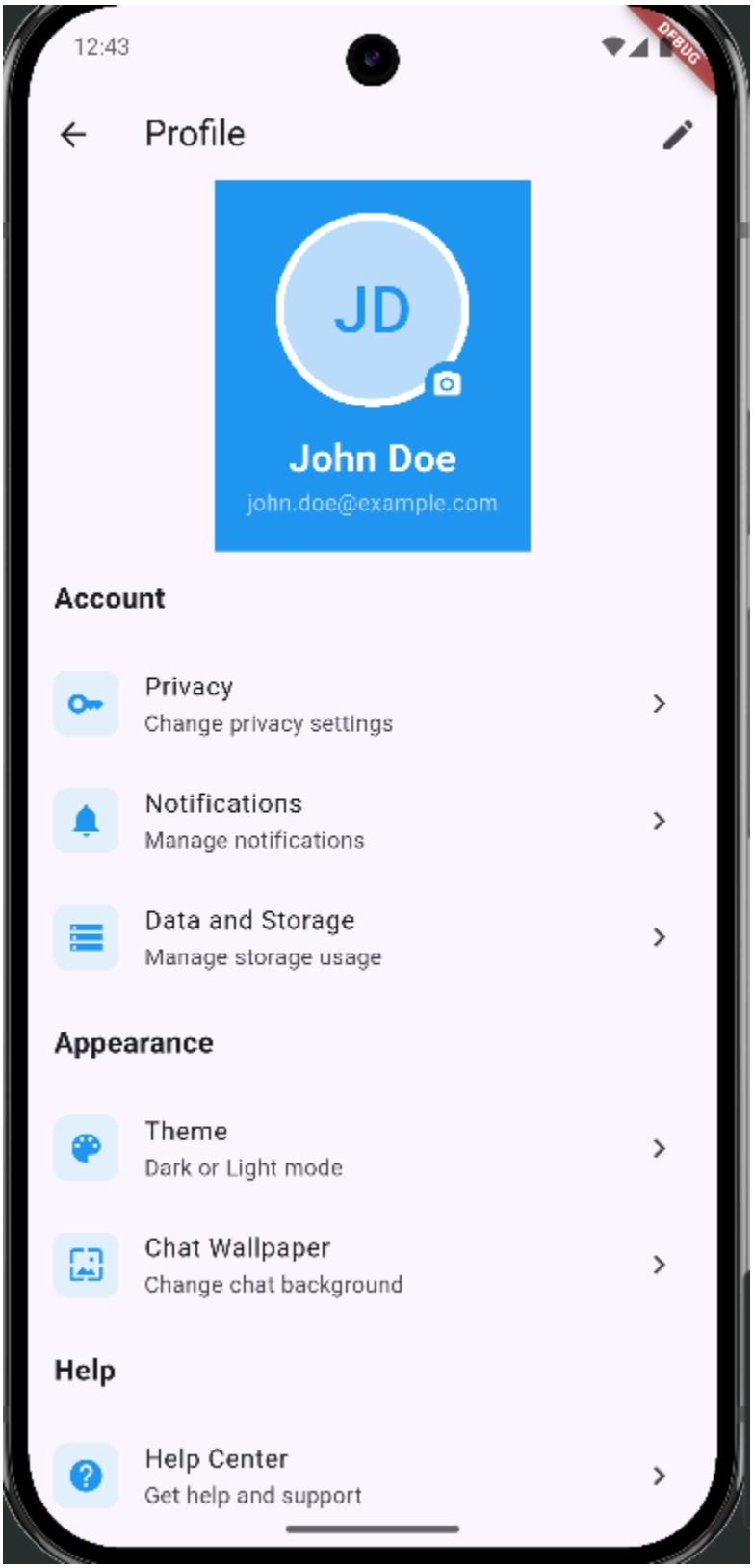
```
import 'package:flutter/material.dart';

import 'chat_screen.dart';


class SearchScreen extends StatefulWidget {

 const SearchScreen({super.key});


 @override

   State<SearchScreen> createState() => _SearchScreenState();

}


class _SearchScreenState extends State<SearchScreen> {

          final TextEditingController _searchController = TextEditingController();

 bool _isSearching = false;


 // Dummy data for demonstration

      final List<Map<String, String>> _recentSearches = [

   {'name': 'John Doe', 'status': 'Online'},

    {'name': 'Jane Smith', 'status': 'Last seen 2h ago'},

       {'name': 'Mike Johnson', 'status': 'Online'},

 ];


      final List<Map<String, String>> _suggestedUsers = [

   {'name': 'Alice Brown', 'status': 'Online'},

    {'name': 'Bob Wilson', 'status': 'Last seen yesterday'},

   {'name': 'Carol White', 'status': 'Online'},

   {'name': 'David Black', 'status': 'Last seen 30m ago'},

 ];


 @override

 Widget build(BuildContext context) {

  return Scaffold(

   appBar: AppBar(

    title: _isSearching

       ? _buildSearchField()

       : const Text('Search'),

    actions: [

     IconButton(

                icon: Icon(_isSearching ? Icons.close : Icons.search),

       onPressed: () {

        setState(() {

         _isSearching = !_isSearching;

         if (!_isSearching) {

          _searchController.clear();

         }

        });

       },

      ),

    ],

   ),

   body: Column(

                crossAxisAlignment: CrossAxisAlignment.start,

    children: [

     if (!_isSearching) ...[
```

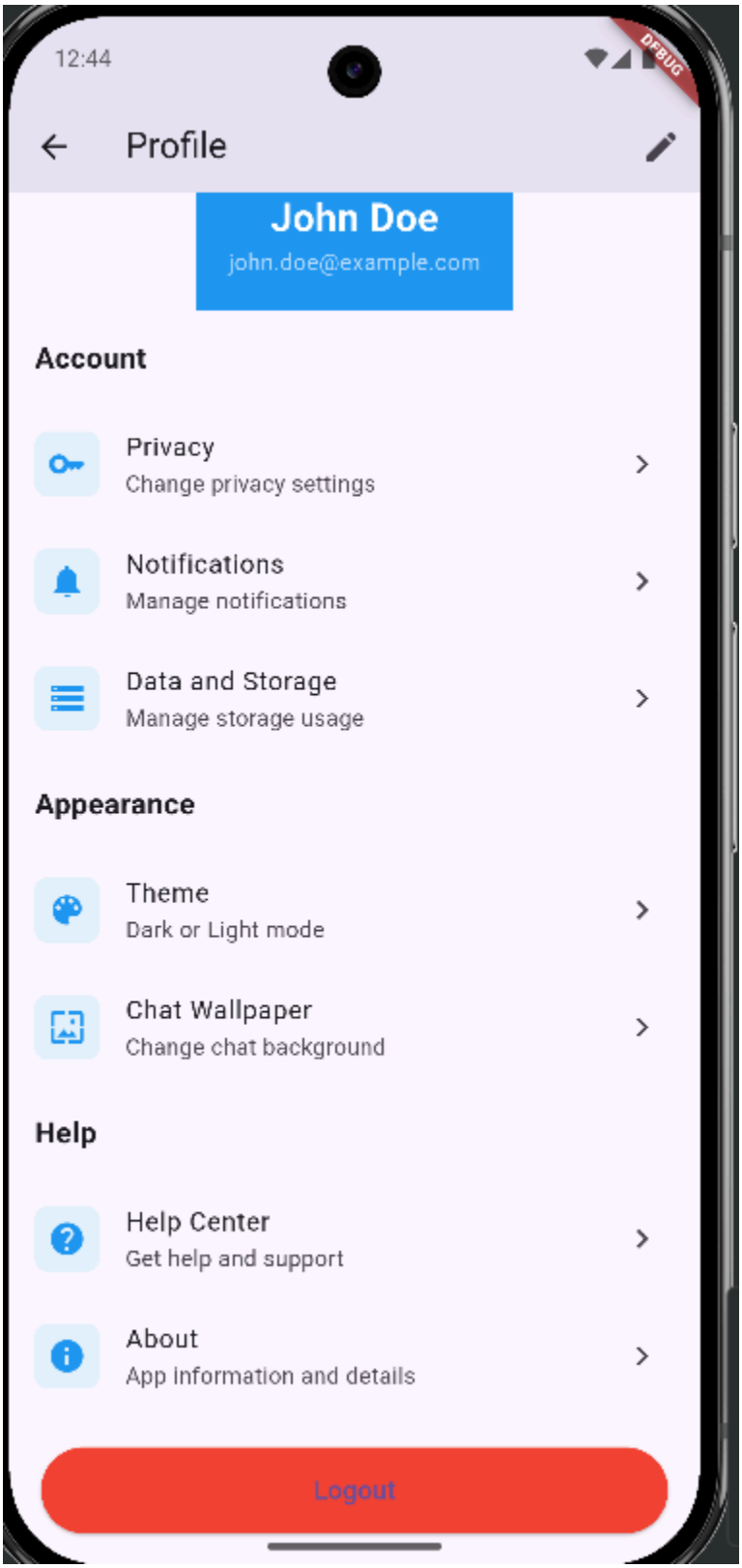
```
        // Recent Searches
        const Padding(
          padding: EdgeInsets.all(16),
          child: Text(
            'Recent Searches',
            style: TextStyle(
              fontSize: 18,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
        ListView.builder(
          shrinkWrap: true,
                                    itemCount:
_recentSearches.length,
          itemBuilder: (context, index) {
            return _buildUserTile(

_recentSearches[index]['name']!,

_recentSearches[index]['status']!,
              isRecent: true,
            );
          },
        ),

        // Suggested Users
        const Padding(
          padding: EdgeInsets.all(16),
          child: Text(
            'Suggested Users',
            style: TextStyle(
              fontSize: 18,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
        Expanded(
          child: ListView.builder(
                                    itemCount:
_suggestedUsers.length,
            itemBuilder: (context, index) {
              return _buildUserTile(

_suggestedUsers[index]['name']!,

_suggestedUsers[index]['status']!,
              );
            },
          ),
        ),
      ],
      if (_isSearching)
        Expanded(
          child: _buildSearchResults(),
        ),
    ],
  ),
                          floatingActionButton:
FloatingActionButton(
    onPressed: () {
      _showAddContactDialog();
    },
    child: const Icon(Icons.person_add),
  ),
```

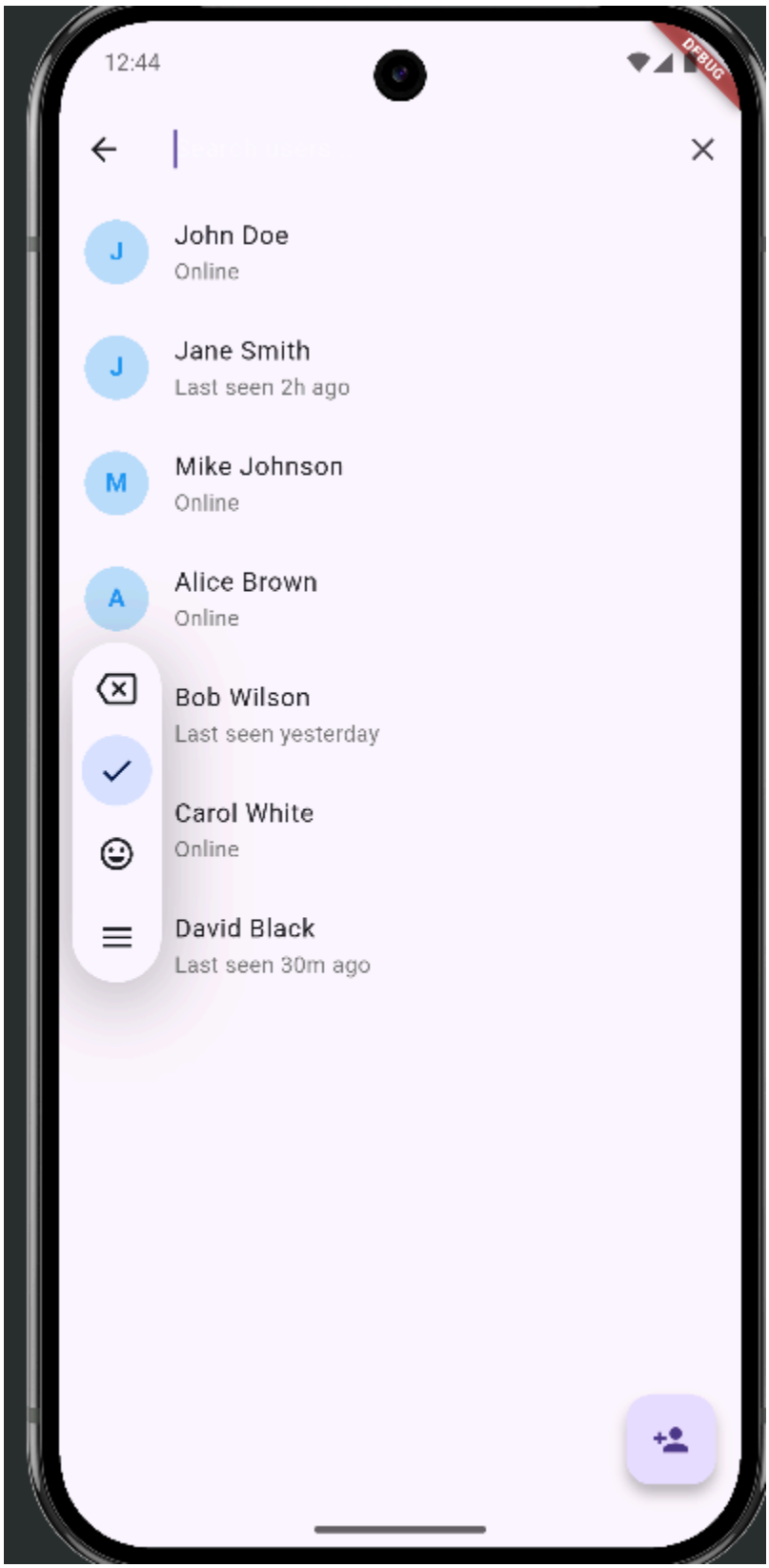
```
    );
  }

  Widget _buildSearchField() {
    return TextField(
      controller: _searchController,
      autofocus: true,
      decoration: const InputDecoration(
        hintText: 'Search users...',
        border: InputBorder.none,
        hintStyle: TextStyle(color: Colors.white70),
      ),
      style: const TextStyle(color: Colors.white),
      onChanged: (value) {
        setState(() {});
      },
    );
  }

  Widget _buildSearchResults() {
    // Filter users based on search query
    final searchQuery = _searchController.text.toLowerCase();
    final searchResults = [..._recentSearches, ..._suggestedUsers]
        .where((user) => user['name']!.toLowerCase().contains(searchQuery))
        .toList();

    return ListView.builder(
      itemCount: searchResults.length,
      itemBuilder: (context, index) {
        return _buildUserTile(
          searchResults[index]['name']!,
          searchResults[index]['status']!,
        );
      },
    );
  }

  Widget _buildUserTile(String name, String status, {bool isRecent = false}) {
    return ListTile(
      leading: CircleAvatar(
        backgroundColor: Colors.blue[100],
        child: Text(
          name[0],
          style: const TextStyle(
            color: Colors.blue,
            fontWeight: FontWeight.bold,
          ),
        ),
      ),
      title: Text(name),
      subtitle: Text(
        status,
        style: TextStyle(color: Colors.grey[600]),
      ),
      trailing: isRecent
          ? IconButton(
        icon: const Icon(Icons.close),
```

```
      onPressed: () {
        // Remove from recent searches
      },
    )
        : null,
    onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context)
=> const ChatScreen()),
      );
    },
  );
 }


 void _showAddContactDialog() {
  showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: const Text('Add New Contact'),
      content: Column(
        mainAxisSize: MainAxisSize.min,
        children: [
          TextField(
            decoration: const InputDecoration(
              labelText: 'Name',
              prefixIcon: Icon(Icons.person),
            ),
          ),
          const SizedBox(height: 16),
          TextField(
            decoration: const InputDecoration(
              labelText: 'Email or Phone',
              prefixIcon: Icon(Icons.email),
            ),
          ),
        ],
      ),
      actions: [
        TextButton(
          onPressed: () {
            Navigator.pop(context);
          },
          child: const Text('Cancel'),
        ),
        ElevatedButton(
          onPressed: () {
            // Add contact logic
            Navigator.pop(context);
          },
          child: const Text('Add'),
        ),
      ],
    ),
  );
 }
}
```

OUTPUT: -


12:43
DEBUG
Profile
JD
John Doe
john.doe@example.com
Account
Privacy
Change privacy settings
Notifications
Manage notifications
Data and Storage
Manage storage usage
Appearance
Theme
Dark or Light mode
Chat Wallpaper
Change chat background
Help
Help Center
Get help and support


12:44
DEBUG
Profile
John Doe
john.doe@example.com
Account
Privacy
Change privacy settings
Notifications
Manage notifications
Data and Storage
Manage storage usage
Appearance
Theme
Dark or Light mode
Chat Wallpaper
Change chat background
Help
Help Center
Get help and support
About
App information and details
Logout


12:44
DEBUG
Search users
J
John Doe
Online
J
Jane Smith
Last seen 2h ago
M
Mike Johnson
Online
A
Alice Brown
Online
Bob Wilson
Last seen yesterday
Carol White
Online
David Black
Last seen 30m ago