Name: Anshi Tiwari
Class: D15C
Roll No: 56

Experiment 05 :
To apply navigation, routing and gestures in Flutter App

Theory:

Navigation, routing, and gestures are essential for building dynamic and user-friendly mobile applications in Flutter. **Navigation** involves moving between different screens or pages using the Navigator widget. **Routing** is the mechanism of defining paths to navigate within an app. Flutter supports two types of routing: **Named Routing** and **Anonymous Routing**. **Gestures** enable user interaction through touch events like taps, swipes, and long presses using GestureDetector and InkWell widgets. Together, these features enhance the app's interactivity and provide a seamless user experience.

Code:

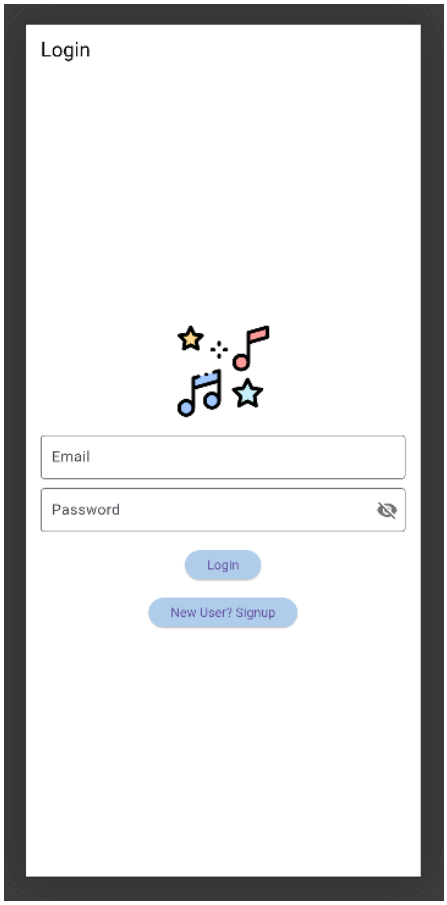
```
class MusicRecommendationPage extends StatefulWidget {
  const MusicRecommendationPage({super.key});

  @override
  _MusicRecommendationPageState createState() =>
      _MusicRecommendationPageState();
}

class _MusicRecommendationPageState extends State<MusicRecommendationPage> {
  final TextEditingController _searchController = TextEditingController();
  List<Map<String, String>> _allSongs = [];
  List<Map<String, String>> _filteredSongs = [];

  @override
  void initState() {
    super.initState();
    _loadSongs();
    _searchController.addListener(_filterSongs);
  }

  Future<void> _loadSongs() async {
    try {
```

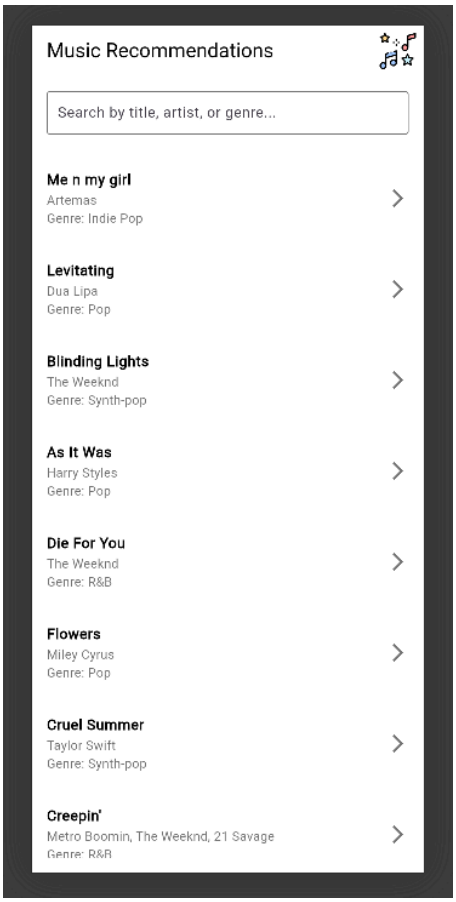
```
    final String response = await rootBundle.loadString('assets/songs.json');
    final List<dynamic> data = json.decode(response);

    // Properly convert dynamic map to Map<String, String>
    setState(() {
      _allSongs = data.map<Map<String, String>>((song) {
        return {
          "title": song["title"].toString(),
          "artist": song["artist"].toString(),
          "genre": song["genre"].toString(),
          "url": song["url"].toString(),
        };
      }).toList();
      _filteredSongs = _allSongs;
    });
    print('Loaded Songs: $_allSongs');
  } catch (e) {
    print('Error loading JSON: $e');
    // If there's an error, clear the lists
    setState(() {
      _allSongs = [];
      _filteredSongs = [];
    });    }  }

void _filterSongs() {
  String query = _searchController.text.toLowerCase();
  setState(() {
    _filteredSongs = _allSongs.where((song) {
      return (song['title']?.toLowerCase().contains(query) ?? false) ||
          (song['artist']?.toLowerCase().contains(query) ?? false) ||
          (song['genre']?.toLowerCase().contains(query) ?? false);
    }).toList();    });  }

Future<void> _launchURL(String url) async {
  final Uri uri = Uri.parse(url);
  if (await canLaunchUrl(uri)) {
    await launchUrl(uri, mode: LaunchMode.externalApplication);
  } else {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text('Could not launch $url')),      );    }  }

void _logout() {
  Navigator.pushAndRemoveUntil(
    context,
```

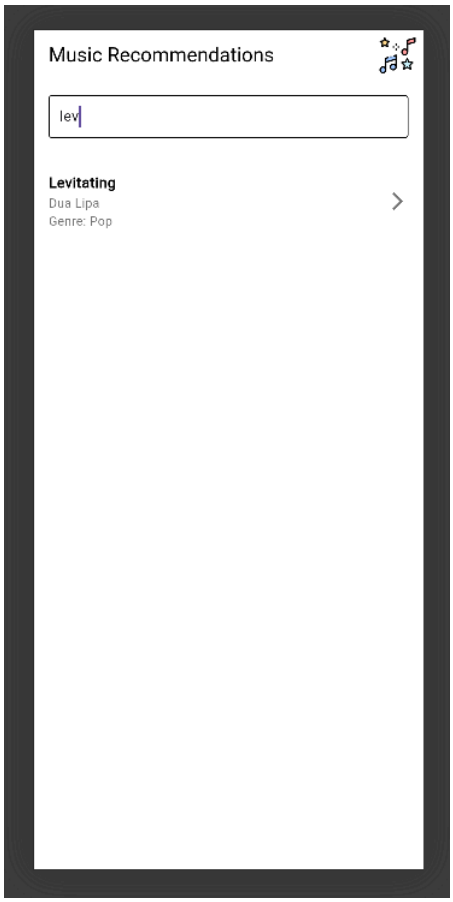
```
    MaterialPageRoute(builder: (context) => const LoginPage()),
    (Route<dynamic> route) => false,
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Music Recommendations'),
      actions: [
        IconButton(
          icon: const Icon(Icons.logout),
          onPressed: _logout,
          tooltip: 'Logout',
          color: Colors.white,
        ),
        Padding(
          padding: const EdgeInsets.all(8.0),
          child: Image.network(
            'https://cdn-icons-png.flaticon.com/512/2907/2907253.png',
            errorBuilder: (context, error, stackTrace) {
              return const Icon(Icons.error, color: Colors.red);
            },
          ),
        ),
      ],
    ),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        children: [
          TextField(
            controller: _searchController,
            decoration: const InputDecoration(
              hintText: 'Search by title, artist, or genre...',
              enabledBorder: OutlineInputBorder(
                borderSide: BorderSide(color: Colors.black54),
              ),
              focusedBorder: OutlineInputBorder(
                borderSide: BorderSide(color: Colors.black),
              ),
            ),
          ),
```

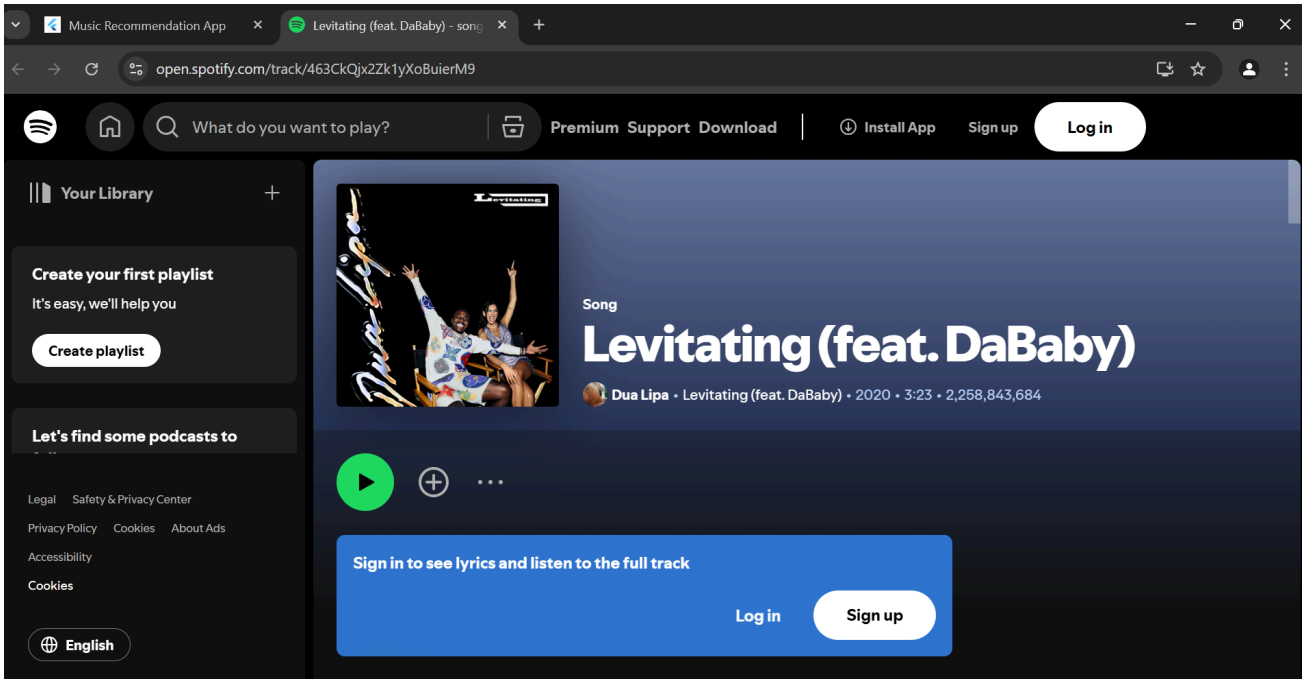
```
const SizedBox(height: 20),
Expanded(
  child: _filteredSongs.isEmpty
      ? const Center(
          child: Text(
            'No songs found',
            style: TextStyle(color: Colors.black54, fontSize: 18),
          ),
        )
      : ListView.builder(
          itemCount: _filteredSongs.length,
          itemBuilder: (context, index) {
            final song = _filteredSongs[index];
            return ListTile(
              contentPadding:
                  const EdgeInsets.symmetric(vertical: 10.0),
              title: Text(
                song['title'] ?? 'Unknown Title',
                style: const TextStyle(
                    color: Colors.black,
                    fontWeight: FontWeight.bold),
              ),
              subtitle: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text(
                    song['artist'] ?? 'Unknown Artist',
                    style: const TextStyle(color: Colors.black54),
                  ),
                  Text(
                    "Genre: ${song['genre'] ?? 'Unknown Genre'}",
                    style: const TextStyle(color: Colors.black54),
                  ),
                ],
              ),
              onTap: () => _launchURL(song['url']!),
              trailing: const Icon(Icons.arrow_forward_ios,
                  color: Colors.black54),
            );                    },                  ),          ),         ],       ),     ),    );  }}
```

Output:


Login
Email
Password
Login
New User? Signup


Music Recommendations
Search by title, artist, or genre...
Me n my girl
Artemas
Genre: Indie Pop
Levitating
Dua Lipa
Genre: Pop
Blinding Lights
The Weeknd
Genre: Synth-pop
As It Was
Harry Styles
Genre: Pop
Die For You
The Weeknd
Genre: R&B
Flowers
Miley Cyrus
Genre: Pop
Cruel Summer
Taylor Swift
Genre: Synth-pop
Creepin'
Metro Boomin, The Weeknd, 21 Savage
Genre: R&B


Music Recommendations
lev
Levitating
Dua Lipa
Genre: Pop


Music Recommendation App
Levitating (feat. DaBaby) - song
open.spotify.com/track/463CkQjx2Zk1yXoBuierM9
What do you want to play?
Premium
Support
Download
Install App
Sign up
Log in
Your Library
Create your first playlist
It's easy, we'll help you
Create playlist
Let's find some podcasts to
Legal
Safety & Privacy Center
Privacy Policy
Cookies
About Ads
Accessibility
Cookies
English
Song
Levitating (feat. DaBaby)
Dua Lipa • Levitating (feat. DaBaby) • 2020 • 3:23 • 2,258,843,684
Sign in to see lyrics and listen to the full track
Log in
Sign up

Explanation:

Navigation and Routing:

Login to Music Recommendation Page: On successful login, the app navigates from the LoginPage to MusicRecommendationPage using:

```
Navigator.pushReplacement(
  context,
  MaterialPageRoute(builder: (context) => const MusicRecommendationPage()),
);
```

Logout Navigation: From MusicRecommendationPage, the user can log out and return to the LoginPage using:

```
Navigator.pushAndRemoveUntil(
  context,
  MaterialPageRoute(builder: (context) => const LoginPage()),
  (Route<dynamic> route) => false,
);
```

Gestures: The app uses onTap gestures on the list of music recommendations. When a user taps on a song, it opens the corresponding URL in an external browser using the url_launcher package:

```
onTap: () => _launchURL(song['url']!),
```

UI and Styling: The app is styled with Material Design principles, using ThemeData for a consistent look.

The LoginPage includes: Email and password fields with validation, Password visibility toggle using an IconButton gesture.

The MusicRecommendationPage includes: A search bar for filtering music, A ListView to display a list of songs with a gesture for URL navigation.

Conclusion:

This Flutter code builds a simple login screen for a music recommendation app. It verifies user credentials and navigates to the next page if they are correct. The design is clean, with input fields and password visibility toggle.