Ameya Chaudhary
D15A
09

MAD PWA
EXPT-5

AIM: To apply navigation ,routing and gestures in app

THEORY:

1. Navigator:

Flutter uses a Navigator to manage a stack of routes or pages. Each route represents a screen or page in your application.

2. Routes:

A route in Flutter is a way to navigate to a different screen or page. It is represented by the PageRoute class and typically involves a transition animation.

3. Named Routes:

Flutter supports named routes, where each route is associated with a unique name. Named routes make it easy to reference specific screens throughout your app.

4. Navigator.push:

To navigate to a new screen, you can use Navigator.push method. This adds a new route to the stack and transitions to the specified screen.

5. Navigator.pop:

To go back to the previous screen, you can use Navigator.pop. This removes the current route from the stack and returns to the previous screen.

6. Sending Data to a New Screen:

You can pass data to a new screen by providing arguments to the constructor of the new screen.

7. Named Routes Example:

Define named routes in the MaterialApp

8. Route Arguments with Named Routes:

You can pass arguments with named routes using ModalRoute.of:
dart

```
final String data = ModalRoute.of(context)!.settings.arguments as String;
```

9. Navigator.pushReplacement:

If you want to replace the current screen with a new one, you can use Navigator.pushReplacement. This is useful when, for example, you don't want the user to go back to the login screen after successfully logging in.
dart

These are the fundamental concepts of navigation and routing in Flutter. Understanding these concepts will help you efficiently manage the flow of your app and navigate between different screens.

CODE:

//home_screen.dart

```
import 'package:flutter/material.dart';
import 'search_screen.dart';
import 'song_details_screen.dart';
import 'library_screen.dart'; // Import the LibraryScreen

class HomeScreen extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(
       title: Text(
         'YouTube Music',
         style: TextStyle(color: Colors.red, fontSize: 34),
       ),
       centerTitle: true,
       backgroundColor: Colors.black,
     ),
     body: SingleChildScrollView(
       child: Padding(
         padding: const EdgeInsets.all(16.0),
         child: Column(
           crossAxisAlignment: CrossAxisAlignment.start,
           children: [
Center(
  child: Text(
    'Welcome to YouTube Music!',
    style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold, color: Colors.pink),
  ),
),
SizedBox(height: 10),
Image.asset(
  'assets/welcome_image.png',
  height: 240,
  width: double.infinity,
  fit: BoxFit.cover,
),
SizedBox(height: 20),
Text(
  'Suggested Songs:',
  style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold, color: Colors.white),
),
SizedBox(height: 25),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    _buildSuggestedItem(context, 'Alone', 'assets/song1.png'),
    _buildSuggestedItem(context, 'Kesariya', 'assets/song2.png'),
    _buildSuggestedItem(context, 'We own it', 'assets/song3.png'),
  ],
),
SizedBox(height: 20),
Text(
  'Suggested Artists:',
  style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold, color: Colors.white),
),
SizedBox(height: 25),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    _buildSuggestedItem(context, 'Eminem', 'assets/eminem.png'),
    _buildSuggestedItem(context, 'Arjit Singh', 'assets/arjit.png'),
    _buildSuggestedItem(context, 'Selena Gomez', 'assets/selena.png'),
  ],
),
SizedBox(height: 20),
ElevatedButton(
  onPressed: () {
```

```
            // Navigate to LibraryScreen when the Library button is tapped
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => LibraryScreen()),
            );
          },
          style: ElevatedButton.styleFrom(
            primary: Colors.red,
          ),
          child: Text('Go to Library'),
        ),
      ],
    ),
  ),
),
bottomNavigationBar: BottomNavigationBar(
  items: [
    BottomNavigationBarItem(
      icon: GestureDetector(
        onTap: () {
          // Navigate to HomeScreen (Optional)
        },
        child: Image.asset('assets/home.png', height: 24, width: 24, color: Colors.red),
      ),
      label: 'Home',
    ),
    BottomNavigationBarItem(
      icon: GestureDetector(
        onTap: () {
          Navigator.push(context, MaterialPageRoute(builder: (context) => SearchScreen()));
        },
        child: Image.asset('assets/search.png', height: 24, width: 24, color: Colors.grey),
      ),
      label: 'Search',
    ),
    BottomNavigationBarItem(
      icon: Image.asset('assets/playlist.png', height: 24, width: 24, color: Colors.grey),
      label: 'Library',
    ),
  ],
  backgroundColor: Colors.deepPurple,
  selectedItemColor: Colors.red,
  unselectedItemColor: Colors.grey,
),
    );
  }

  Widget _buildSuggestedItem(BuildContext context, String name, String imagePath) {
    return GestureDetector(
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => SongDetailsScreen(songTitle: name, songImagePath: imagePath),
          ),
        );
      },
      child: Column(
        children: [
          Image.asset(
            imagePath,
            height: 80,
            width: 80,
            fit: BoxFit.cover,
          ),
          SizedBox(height: 5),
          Text(
            name,
            style: TextStyle(color: Colors.white),
          ),
        ],
      ),
    );
  }
}
```

//search_screen.dart

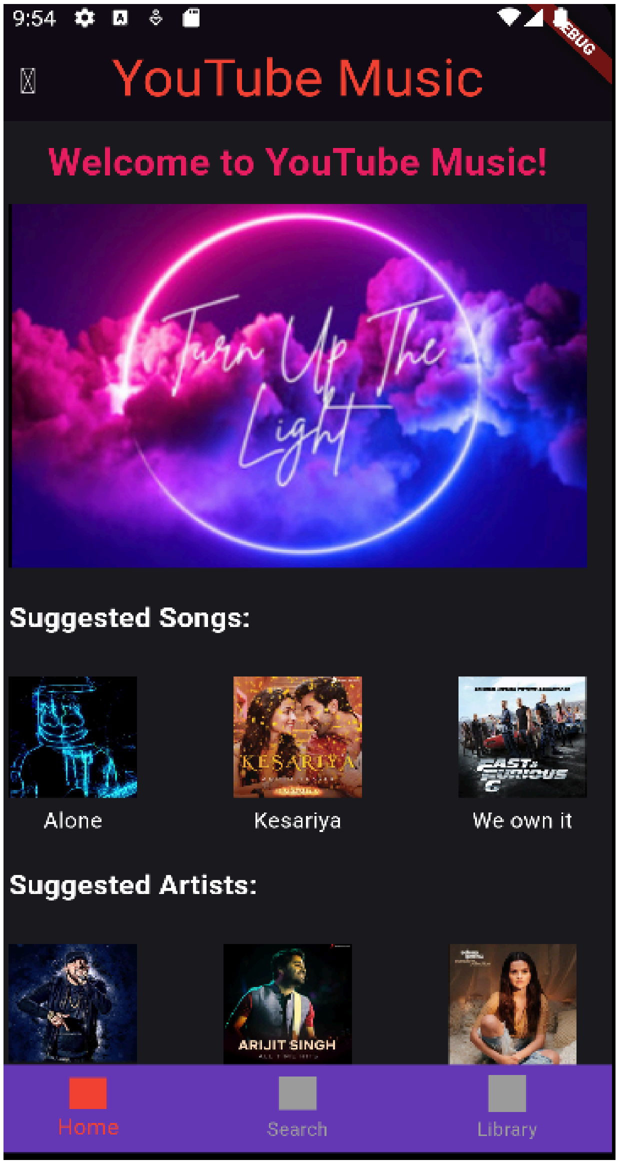
```
import 'package:flutter/material.dart';
import 'home_screen.dart';
import 'library_screen.dart'; // Import the LibraryScreen

class SearchScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          'Search',
    style: TextStyle(color: Colors.white, fontSize: 24),
  ),
  centerTitle: true,
  backgroundColor: Colors.black,
),
body: SingleChildScrollView(
  child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        _buildSearchBar(),

        // Add the top trending songs title
        Text(
          'Top Trending Songs',
          style: TextStyle(fontSize: 26, fontWeight: FontWeight.bold, color: Colors.red),
        ),

        // Add the three images with their song names
        _buildSuggestedItem('Song 1', 'assets/song1.png'),
        _buildSuggestedItem('Song 2', 'assets/song2.png'),
        _buildSuggestedItem('Song 3', 'assets/song3.png'),
      ],
    ),
  ),
),

bottomNavigationBar: BottomNavigationBar(
  items: [
    BottomNavigationBarItem(
      icon: GestureDetector(
        onTap: () {
          Navigator.pushReplacement(
            context,
            MaterialPageRoute(builder: (context) => HomeScreen()),
          );
        },
        child: Image.asset('assets/home.png', height: 24, width: 24, color: Colors.grey),
      ),
      label: 'Home',
    ),
    BottomNavigationBarItem(
      icon: Image.asset('assets/search.png', height: 24, width: 24, color: Colors.red),
          label: 'Search',
        ),
        BottomNavigationBarItem(
          icon: GestureDetector(
            onTap: () {
              Navigator.push(context, MaterialPageRoute(builder: (context) => LibraryScreen()));
            },
            child: Image.asset('assets/playlist.png', height: 24, width: 24, color: Colors.grey),
          ),
          label: 'Library',
        ),
      ],
      backgroundColor: Colors.deepPurple,
      selectedItemColor: Colors.red,
      unselectedItemColor: Colors.grey,
    ),
  );
}

Widget _buildSearchBar() {
  return Container(
    margin: EdgeInsets.symmetric(vertical: 16),
    padding: EdgeInsets.symmetric(horizontal: 16),
    decoration: BoxDecoration(
      color: Colors.grey[900],
      borderRadius: BorderRadius.circular(30),
    ),
    child: TextField(
      style: TextStyle(color: Colors.white),
      decoration: InputDecoration(
        hintText: 'Search for songs...',
        hintStyle: TextStyle(color: Colors.grey),
        suffixIcon: Icon(Icons.search, color: Colors.white),
        border: InputBorder.none,
      ),
    ),
  );
}

Widget _buildSuggestedItem(String name, String imagePath) {
  return Row(
    children: [
      Image.asset(
        imagePath,
         height: 80,
         width: 80,
         fit: BoxFit.cover,
       ),
       SizedBox(width: 16),
       Column(
         crossAxisAlignment: CrossAxisAlignment.start,
         children: [
           Text(
             name,
             style: TextStyle(color: Colors.white),
           ),
         ],
       ),
     ],
   );
 }
}
```

//library_screen.dart

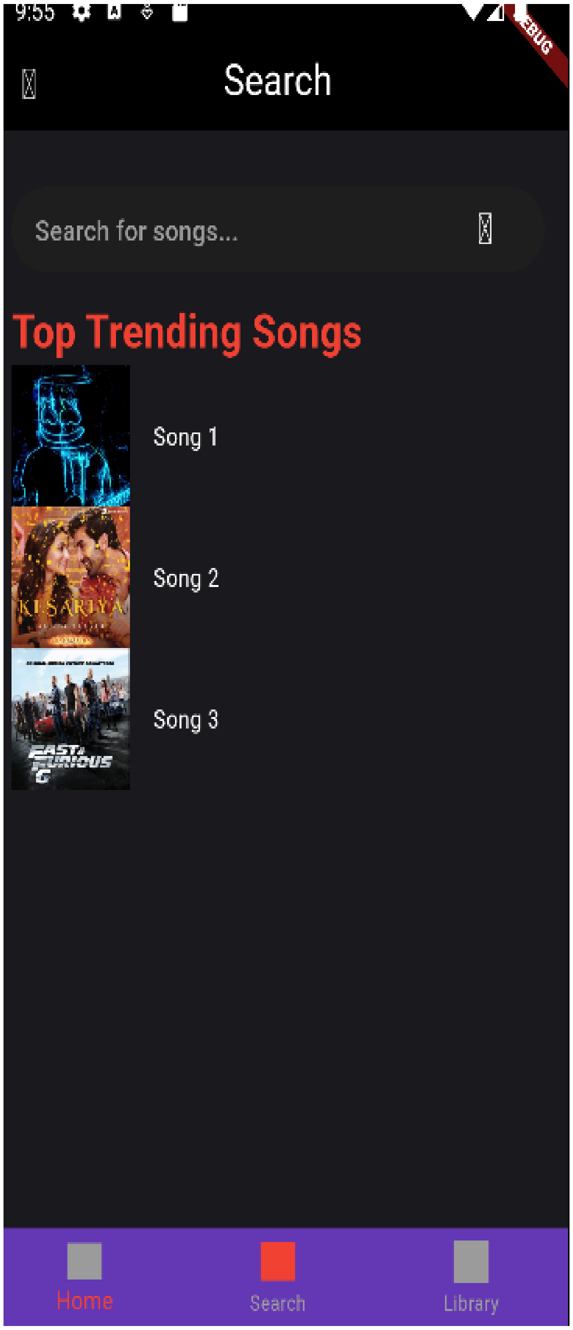
```
import 'package:flutter/material.dart';
import 'home_screen.dart';
import 'search_screen.dart';
import 'song_details_screen.dart';
import 'package:provider/provider.dart';

class LibraryScreen extends StatelessWidget {
 // Map to associate each song title with its image path
 final Map<String, String> songImageMap = {
   'Song 1': 'assets/song1.png',
   'Song 2': 'assets/song2.png',
   'Song 3': 'assets/song3.png',
   // Add more entries as needed
 };

 @override
 Widget build(BuildContext context) {
   List<String> likedSongs = Provider.of<LikedSongs>(context).likedSongs;

   return Scaffold(
     appBar: AppBar(
       title: Text(
         'Library',
         style: TextStyle(color: Colors.red, fontSize: 24),
       ),
       centerTitle: true,
       backgroundColor: Colors.black,
     ),
     body: Padding(
       padding: const EdgeInsets.all(16.0),
       child: Column(
         crossAxisAlignment: CrossAxisAlignment.start,
         children: [
           Text(
             'Liked Songs',
             style: TextStyle(fontSize: 26, fontWeight: FontWeight.bold, color: Colors.red),
           ),
           // Display liked songs here
           Expanded(
             child: ListView.builder(
               itemCount: likedSongs.length,
               itemBuilder: (context, index) {
                 // Extract the song details from the likedSongs list
                 String songTitle = likedSongs[index];
                 String songImagePath = getSongImagePath(songTitle); // Get the image path based
on the song title

                 return ListTile(
                   leading: Image.asset(
                     songImagePath,
                     height: 40,
                     width: 40,
                     fit: BoxFit.cover,
                   ),
                   title: Text(songTitle),
                   // Add more details if needed
                 );
               },
             ),
           ),
         ],
       ),
     ),
     bottomNavigationBar: BottomNavigationBar(
       items: [
         BottomNavigationBarItem(
           icon: GestureDetector(
             onTap: () {
               Navigator.pushReplacement(
                 context,
                 MaterialPageRoute(builder: (context) => HomeScreen()),
               );
             },
             child: Image.asset('assets/home.png', height: 24, width: 24, color: Colors.grey),
           ),
           label: 'Home',
         ),
         BottomNavigationBarItem(
           icon: GestureDetector(
             onTap: () {
               Navigator.push(context, MaterialPageRoute(builder: (context) => SearchScreen()));
             },
             child: Image.asset('assets/search.png', height: 24, width: 24, color: Colors.grey),
           ),
           label: 'Search',
         ),
         BottomNavigationBarItem(
           icon: Image.asset('assets/playlist.png', height: 24, width: 24, color: Colors.grey),
           label: 'Library',
         ),
       ],
       backgroundColor: Colors.deepPurple,
       selectedItemColor: Colors.red,
       unselectedItemColor: Colors.grey,
     ),
   );
 }

 // Helper method to get the image path based on the song title
 String getSongImagePath(String songTitle) {
   // Use the map to get the image path based on the song title
   return songImageMap[songTitle] ?? 'assets/song1.png';
 }
}
```

OUTPUT:

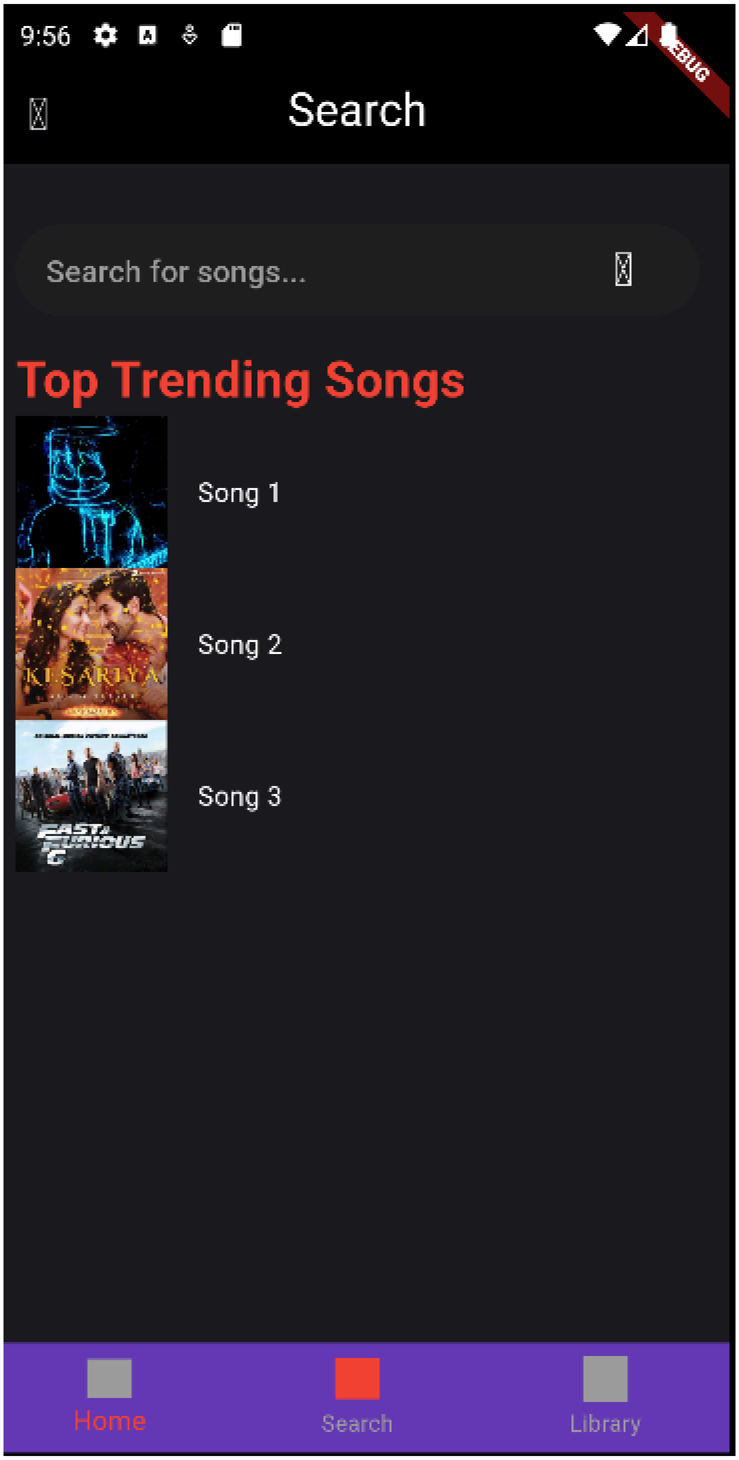
home_screen



//search_screen



//playlist_screen:



CONCLUSION:

In conclusion, mastering navigation and routing in Flutter is essential for building sophisticated and user-centric applications. A well-designed navigation system enhances user engagement, simplifies code maintenance, and contributes to an overall positive user experience. As developers explore and apply these concepts, they gain the ability to create fluid and responsive mobile applications that meet the expectations of modern users.