

## MPL Lab Exp 3

Name: Anurag Gaiwal

Div: D15A

Roll No. 17

Batch: A

Aim: To include icons, images, fonts in Flutter app.

Theory:

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.

Steps to Add an Image:

Step 1. Create a new folder

- It should be in the root of your flutter project. You can name it whatever you want, but assets are preferred.

- If you want to add other assets to your app, like fonts, it is preferred to make another subfolder named images.

Step 2. Now you can copy your image to images sub-folder. The path should look like assets/images/yourImage. Before adding images also check the above-mentioned supported

image formats.

Step 3. Register the assets folder in pubspec.yaml file and update it.

To add images, write the following code:

flutter:

assets:

- assets/images/yourFirstImage.jpg
- assets/image/yourSecondImage.jpg

If you want to include all the images of the assets folder then add this:

flutter:

assets:

- assets/images/

Step 4. Insert the image code in the file, where you want to add the image.

Step 5. Now you can save all the files and run the app, you will find the output as shown below.

Icon class in Flutter is used to show specific icons in our app. Instead of creating an

image for our icon, we can simply use the Icon class for inserting an icon in our app. For using this class you must ensure that you have set uses-material-design: true in the pubsec.yml file of your object.

Properties:

- color: It is used to set the color of the icon
- size: It is used to resize the Icon
- semanticLabel: It comes in play while using the app in accessibility mode (ie, voice-over)
- textDirection: It is used for rendering Icon

Note: The semanticLabel are not visible in the UI.

### Code in home\_screen.dart:

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:jio_saavn_auth/screens/email_auth/login_screen.dart';
import "package:jio_saavn_auth/screens/library_screen.dart";
import 'package:jio_saavn_auth/screens/navigation_bar.dart';
import 'package:jio_saavn_auth/screens/search_screen.dart';
import 'package:jio_saavn_auth/screens/song_player_screen.dart';
import 'package:jio_saavn_auth/screens/top_artists_songs.dart';
```

```
class Song {
  final String name;
  final String path;
  final String imagePath;

  Song({
    required this.name,
    required this.path,
    required this.imagePath,
  });
}
```

```

List<Song> recommendedSongs = [
  Song(
    name: 'Alone',
    path: 'assets/songs/song1.mp3',
    imagePath: 'assets/song1.png'),
  Song(
    name: 'Kesariya',
    path: 'assets/songs/song2.mp3',
    imagePath: 'assets/song2.png'),
  Song(
    name: 'We own it',
    path: 'assets/songs/song3.mp3',
    imagePath: 'assets/song3.png'),
];

```

// Add this function outside the HomeScreen class

```

Future<List<String>> fetchTopSongsForArtist(String artistName) async {
  // Simulate fetching data, replace this with actual data fetching logic
  await Future.delayed(Duration(seconds: 2)); // Simulating a delay
  // Hardcoded top songs for each artist
  Map<String, List<String>> artistTopSongs = {
    'Without Me': ['song1.mp3', 'song2.mp3', 'song3.mp3'],
    'Heeriye': ['song4.mp3', 'song5.mp3', 'song6.mp3'],
    'Calm Down': ['song7.mp3', 'song8.mp3', 'song9.mp3'],
    // Add more artists and their top songs as needed
  };

  return artistTopSongs[artistName] ?? [];
}

```

```

class HomeScreen extends StatefulWidget {
  const HomeScreen({Key? key}) : super(key: key);
}

```

```

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

class _HomeScreenState extends State<HomeScreen> {
  int _selectedIndex = 0;

  void logout() async {
    await FirebaseAuth.instance.signOut();

    Navigator.popUntil(context, (route) => route.isFirst);
    Navigator.pushReplacement(
      context,
      MaterialPageRoute(builder: (context) => LoginScreen()),
    );
  }

  Color _getColor(int index) {
    return _selectedIndex == index
      ? Colors.white
      : const Color.fromARGB(255, 128, 128, 128);
  }

  Widget _buildSuggestedItem(BuildContext context, Song song) {
    return GestureDetector(
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => SongPlayerScreen(
              songPaths: [song.path],
              songName: song.name,
              imagePath: song.imagePath,
            ),
          ),
        );
      },
    );
  }
}

```

```

    ),
  );
},
child: Container(
  width: 120,
  margin: EdgeInsets.only(right: 5),
  child: Column(
    children: [
      Image.asset(
        song.imagePath,
        height: 100,
        width: 100,
        fit: BoxFit.cover,
      ),
      SizedBox(height: 5),
      Text(
        song.name,
        style: TextStyle(color: Color.fromARGB(255, 255, 255, 255)),
      ),
    ],
  ),
),
);
}

```

```

Widget _buildImageRow(List<Song> songs) {
  return SizedBox(
    height: 150,
    child: ListView.builder(
      scrollDirection: Axis.horizontal,
      itemCount: songs.length,
      itemBuilder: (BuildContext context, int index) {
        return GestureDetector(
          onTap: () async {

```

```

    print('Tapped on ${songs[index].name}');

    // Fetch the top songs for the selected artist
    List<String> topSongs =
        await fetchTopSongsForArtist(songs[index].name);
    print('Top songs for ${songs[index].name}: $topSongs');

    // Navigate to TopArtistSongsScreen with the fetched top songs
    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) => TopArtistSongsScreen(
                artistName: songs[index].name,
                topSongs: topSongs,
            ),
        ),
    );
},
child: _buildSuggestedItem(context, songs[index]),
);
},
),
);
}

```

```

@override
Widget build(BuildContext context) {
    List<Song> topArtists = [
        Song(
            name: 'Without Me',
            path: 'assets/songs/eminem.mp3',
            imagePath: 'assets/travis.png'),
        Song(
            name: 'Heeriye',

```

```
        path: 'assets/songs/arjit.mp3',
        imagePath: 'assets/sonu.png'),
    Song(
        name: 'Calm Down',
        path: 'assets/songs/selena.mp3',
        imagePath: 'assets/drake.png'),
];
```

```
List<Song> newReleases = [
    Song(
        name: 'Supriya sule - TRS hindi',
        path: 'assets/songs/new2.mp3',
        imagePath: 'assets/trsh_supriya_sule.jpg'),
    Song(
        name: 'Satranga',
        path: 'assets/songs/new3.mp3',
        imagePath: 'assets/satranga.png'),
    Song(
        name: 'Duniya jala denge',
        path: 'assets/songs/new3.mp3',
        imagePath: 'assets/sariduniya.png'),
];
```

```
List<Song> jioSaavnPicks = [
    Song(
        name: 'Challeya',
        path: 'assets/songs/pick1.mp3',
        imagePath: 'assets/challeya.png'),
    Song(
        name: 'Zinda Banda',
        path: 'assets/songs/pick2.mp3',
        imagePath: 'assets/zindabanda.png'),
    Song(
        name: 'Jawan Title Track',
```

```

        path: 'assets/songs/pick3.mp3',
        imagePath: 'assets/jawan.png'),
];

return Scaffold(
  appBar: AppBar(
    automaticallyImplyLeading: false,
    backgroundColor: Colors.black,
    actions: [
      IconButton(
        onPressed: () {
          logout();
        },
        icon: Icon(Icons.exit_to_app),
      ),
    ],
  ),
  backgroundColor: Color.fromARGB(255, 30, 30, 30),
  body: SingleChildScrollView(
    child: Padding(
      padding: const EdgeInsets.all(12.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Center(
            child: Align(
              alignment: Alignment.center,
              child: Image.asset(
                'assets/jiosaavn_name.png',
                height: 150,
                width: 200,
              ),
            ),
          ),
        ],
      ),
    ),
  ),
);

```



```
SizedBox(height: 10),
SizedBox(height: 20),
Text(
  'Recommended Songs for you:',
  style: TextStyle(
    fontSize: 20,
    fontWeight: FontWeight.bold,
    color: Color.fromARGB(255, 255, 255, 255),
  ),
),
SizedBox(height: 25),
_buildImageRow(recommendedSongs),
SizedBox(height: 20),
Text(
  'From Top artists for you:',
  style: TextStyle(
    fontSize: 20,
    fontWeight: FontWeight.bold,
    color: Color.fromARGB(255, 255, 255, 255),
  ),
),
SizedBox(height: 25),
_buildImageRow(topArtists),
SizedBox(height: 20),
Text(
  'New Podcasts',
  style: TextStyle(
    fontSize: 20,
    fontWeight: FontWeight.bold,
    color: Color.fromARGB(255, 255, 255, 255),
  ),
),
SizedBox(height: 25),
_buildImageRow(newReleases),
```

```

        SizedBox(height: 20),
        Text(
          'Spotify picks:',
          style: TextStyle(
            fontSize: 20,
            fontWeight: FontWeight.bold,
            color: Color.fromARGB(255, 255, 255, 255),
          ),
        ),
        SizedBox(height: 25),
        _buildImageRow(jioSaavnPicks),
      ],
    ),
  ),
),
bottomNavigationBar: CustomBottomNavigationBar(
  selectedIndex: _selectedIndex,
  onItemTapped: (index) {
    setState(() {
      _selectedIndex = index;
    });
    if (_selectedIndex == 1) {
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => SearchScreen()),
      );
    } else if (_selectedIndex == 2) {
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => MyLibrary()),
      );
    }
  },
),

```

```
);  
}  
}
```

Output:

