Name - Gaurang A Raorane Div - D15A
Roll no - 49 Batch - C

Experiment - 3

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

Theory:-

1) Icons:
   Icons are small, visual representations of actions, objects, or concepts. In Flutter, icons can be added using the Icon widget, which allows developers to incorporate a wide range of pre-built icons from the Material Icons library or other icon packs. To use icons, developers simply specify the icon data or icon name within the Icon widget.
2) Images:
   Images are graphical assets that can be used to convey information, illustrate content, or enhance the visual appeal of an app. Flutter provides support for displaying both local and network images using the Image widget. Local images, stored in the project's assets folder, can be imported and displayed using the AssetImage class. Network images can be loaded asynchronously using the NetworkImage class.
3) Fonts:
   Fonts define the visual appearance of text in an app and are instrumental in maintaining brand consistency and UI design. In Flutter, developers can incorporate custom fonts by adding font files (e.g., TrueType or OpenType) to the project and defining font families in the pubspec.yaml file. Once imported, custom fonts can be applied to text using the TextStyle widget by specifying the desired font family.

Integration Process:

Icons:
- Import the Material Icons package in the pubspec.yaml file.
- Use the Icon widget to add icons to Flutter widgets by specifying the icon data or icon name.

Images:
- Add image assets to the project's assets folder.
- Declare the image assets in the pubspec.yaml file.
- Display images using the Image widget and specifying the image asset path.

Fonts:
- Add custom font files (e.g., .ttf, .otf) to the project directory.
- Define font families in the pubspec.yaml file by specifying the font file paths.
- Apply custom fonts to text using the TextStyle widget and specifying the font family.

Best Practices:
- Optimize image assets to reduce file size and improve app performance.

- Use vector-based icons whenever possible to ensure scalability and resolution independence.
- Maintain consistency in font usage across the app to reinforce brand identity and enhance readability.
- Test the app on different devices and screen sizes to ensure proper rendering of icons, images, and fonts.
-

TVShowScreen.dart

```
import 'package:flutter/services.dart';
import 'package:intl/intl.dart';
import 'package:flutter/material.dart';
import 'package:osiris/Models/TvShow.dart';
import 'package:osiris/Services/API.dart';
import 'package:osiris/Services/consts.dart';
import 'package:osiris/Widgets/CustomLists.dart';
import 'package:osiris/Widgets/LoadingScreen.dart';
import 'package:osiris/Widgets/SeasonsList.dart';
import 'package:unicons/unicons.dart';
import 'package:osiris/Services/extraServices.dart';
import 'package:osiris/Widgets/DetailScreenComponents.dart';

class TVShowScreen extends StatefulWidget {
  TVShowScreen(this.movieId, {super.key});
  String movieId;

  @override
  State<TVShowScreen> createState() => _TVShowScreenState();
}

class _TVShowScreenState extends State<TVShowScreen> {
  bool isLoading = true;
  late List<TvShow> recommendedTvShows;

  Future<void> fetchData() async {
    recommendedTvShows =
        await APIService().getRecommendedTvShows(widget.movieId);
    setState(() {
      isLoading = false;
    });
  }

  @override
  void initState() {
    super.initState();
```

```
  fetchData();
}

var selectedSeason = 0;
@override
Widget build(BuildContext context) {
  var size = MediaQuery.of(context).size;
  return Scaffold(
    backgroundColor: background_primary,
    body: isLoading
        ? const LoadingScreen()
        : FutureBuilder(
            future: APIService().getTvShowDetail(widget.movieId),
            builder: (context, AsyncSnapshot snapshot) {
              if (snapshot.hasData) {
                var status = snapshot.data!.status.toString();
                var releaseDate = snapshot.data!.firstAirDate.toString();
                var seasonCount = snapshot.data!.numberOfSeasons;
                var seasonList = [];
                for (var i = 1; i <= seasonCount; i++) {
                  seasonList.add("Season $i");
                }
                return ListView(
                    scrollDirection: Axis.vertical,
                    physics: const AlwaysScrollableScrollPhysics(
                        parent: BouncingScrollPhysics()),
                    padding: EdgeInsets.zero,
                    shrinkWrap: true,
                    children: [
                      Stack(
                        children: [
                          Container(
                            width: size.width,
                            height: size.height * 0.40 > 300
                                ? size.height * 0.40
                                : 300,
                            decoration: BoxDecoration(
                              image: DecorationImage(
                                image: snapshot.data!.backdropPath == null
                                    ? const AssetImage(
                                            "assets/LoadingImage.png")
                                        as ImageProvider
                                    : NetworkImage(
                                        "https://image.tmdb.org/t/p/original${snapshot.data!.backdropPath}",
```

```
            ),
        fit: BoxFit.cover,
      ),
    ),
  ),
  Container(
    width: size.width,
    height: size.height * 0.40 > 300
        ? size.height * 0.40
        : 300,
    decoration: BoxDecoration(
      gradient: LinearGradient(
          begin: Alignment.topCenter,
          end: Alignment.bottomCenter,
          colors: [
            Colors.transparent,
            Colors.transparent,
            background_primary.withOpacity(0.50),
            background_primary.withOpacity(0.75),
            background_primary.withOpacity(0.85),
            background_primary.withOpacity(0.90),
            background_primary.withOpacity(0.95),
            background_primary.withOpacity(1.00)
          ]),
    ),
  ),
  Container(
    width: size.width,
    height: size.height * 0.35 > 300
        ? size.height * 0.35
        : 300,
    margin: const EdgeInsets.all(8),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      mainAxisAlignment: MainAxisAlignment.end,
      children: [
        Text(
          snapshot.data!.voteAverage
              .toString()
              .substring(0, 3),
          style: const TextStyle(
              fontSize: 40,
              fontWeight: FontWeight.w500,
              color: Colors.white),
```

```
      ),
      Text(
        snapshot.data!.name.toString(),
        overflow: TextOverflow.ellipsis,
        maxLines: 2,
        style: const TextStyle(
            fontSize: 24,
            fontWeight: FontWeight.w500,
            color: Colors.white),
      ),
      Row(
        children: [
          CircularButtons(
            UniconsLine.play,
            onTap: () {
              HapticFeedback.lightImpact();
              APIService()
                  .getTrailerLink(
                      snapshot.data!.id.toString(),
                      "tv")
                  .then(
                      (value) => LaunchUrl(value));
            },
          ),
          CircularButtons(
            UniconsLine.plus,
            onTap: () {
              HapticFeedback.lightImpact();
              pshowDialog(
                  context, widget.movieId, "tv");
            },
          ),
          Visibility(
            visible: snapshot.data!.adult,
            child: CircularButtons(
              UniconsLine.eighteen_plus,
              onTap: () {},
            ),
          ),
        ],
      )
    ],
  ),
)
```

```
  ],
),
FutureBuilder(
  future:
      APIService().getMovieGenres(widget.movieId, "tv"),
  builder: (context, AsyncSnapshot snapshot) {
    if (snapshot.hasData) {
      return Container(
        height: 36,
        width: size.width,
        margin: const EdgeInsets.only(left: 8),
        child: ListView.builder(
          scrollDirection: Axis.horizontal,
          shrinkWrap: true,
          physics: const BouncingScrollPhysics(),
          itemCount: snapshot.data.length,
          itemBuilder: (context, index) {
            return TextContainer(
                snapshot.data![index].name.toString(),
                const EdgeInsets.only(right: 8),
                const Color(0xFF14303B));
          },
        ),
      );
    } else {
      return TextContainer(
          "Loading",
          const EdgeInsets.all(8),
          const Color(0xFF14303B));
    }
  },
),
Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    TitleText("Status"),
    Row(
      children: [
        TextContainer(
            status,
            const EdgeInsets.only(
                left: 8, right: 8, bottom: 8),
            const Color(0xFF382E39)),
        TextContainer(
```

```
                    "Release:
${DateFormat.yMMMMd().format(DateTime.parse(releaseDate))}",
                    const EdgeInsets.only(
                        left: 8, right: 8, bottom: 8),
                    const Color(0xFF545551)),
                ],
              ),
              TitleText("Overview"),
              TextContainer(
                snapshot.data!.overview.toString().isEmpty ||
                        snapshot.data!.overview.toString() ==
                            "null"
                    ? "No overview available"
                    : snapshot.data!.overview.toString(),
                const EdgeInsets.all(8),
                const Color(0xFF0F1D39)),
              TitleText("Seasons"),
              Container(
                height: 36,
                width: size.width,
                margin: const EdgeInsets.only(left: 8),
                child: ListView.builder(
                  scrollDirection: Axis.horizontal,
                  shrinkWrap: true,
                  physics: const BouncingScrollPhysics(),
                  itemCount: seasonCount,
                  itemBuilder: (context, index) {
                    return GestureDetector(
                      onTap: () {
                        HapticFeedback.lightImpact();
                        setState(() {
                          selectedSeason = index;
                        });
                      },
                      child: TextContainer(
                          seasonList[index].toString(),
                          const EdgeInsets.only(right: 8),
                          index == selectedSeason
                              ? const Color(0xFF545551)
                              : const Color(0xFF14303B)),
                    );
                  },
                ),
              ),
```

```
                    SeasonsList(selectedSeason + 1, widget.movieId),
                    TitleText("Recommendations"),
                    CustomListTV(recommendedTvShows),
                  ],
                )
              ]);
          } else {
            return const LoadingScreen();
          }
        },
      ),
    );
  }
}
```

SectionText.dart

```
import 'package:flutter/material.dart';
import 'package:osiris/Services/consts.dart';

Widget SectionText(String ktitle, String ntitle) {
  return Container(
      width: double.infinity,
      margin: const EdgeInsets.all(8),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.start,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
          Padding(
            padding: const EdgeInsets.only(right: 8.0),
            child: Text(
              ktitle.toUpperCase(),
              style: const TextStyle(
                  color: Colors.white,
                  fontSize: 16,
                  fontWeight: FontWeight.w500,
                  letterSpacing: 5),
            ),
          ),
          Text(
            ntitle.toUpperCase(),
            style: TextStyle(
                color: accent_secondary,
                fontSize: 16,
                fontWeight: FontWeight.w500,
```

```
            letterSpacing: 5),
        ),
      ],
    ));
}
```

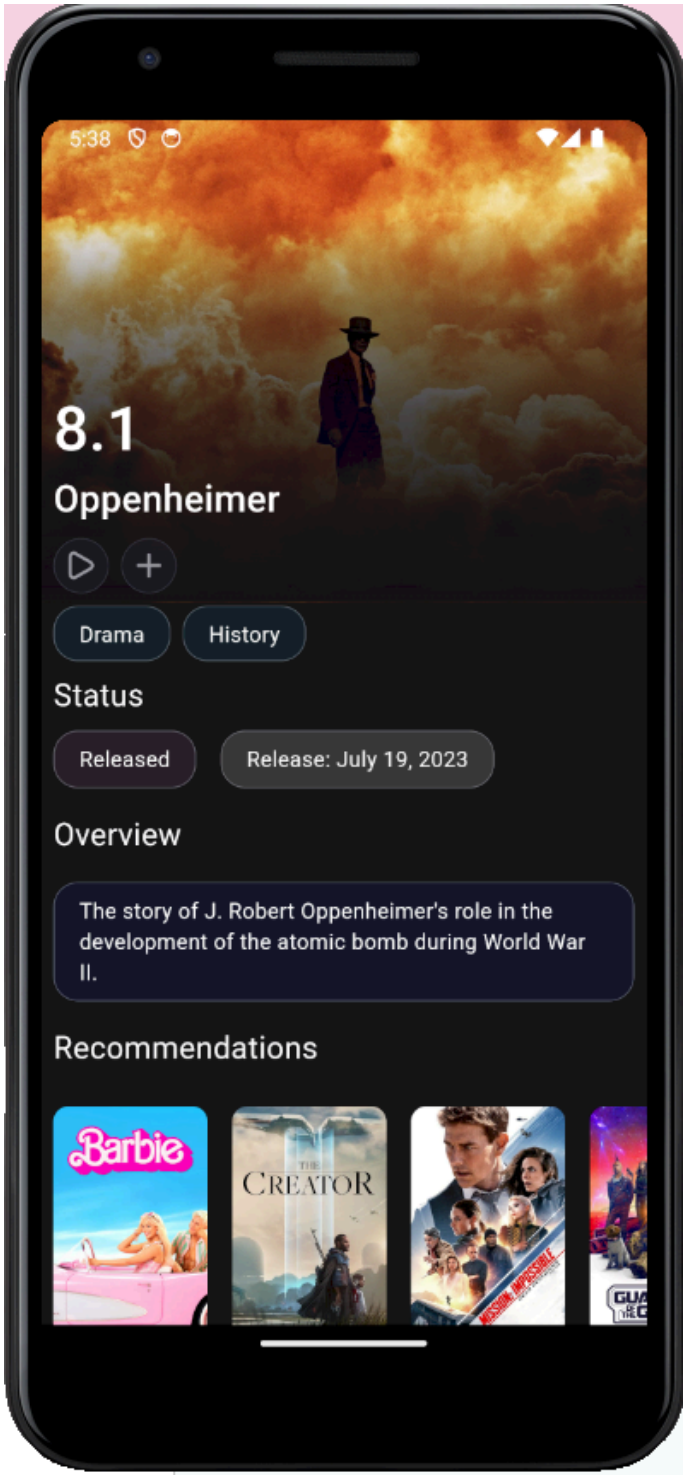
SeasonsList.dart

```
import 'package:flutter/material.dart';
import 'package:osiris/Services/API.dart';

class SeasonsList extends StatefulWidget {
  SeasonsList(this.seasonNumber, this.movieId, {super.key});
  int seasonNumber;
  String movieId;

  @override
  State<SeasonsList> createState() => _SeasonsListState();
}

class _SeasonsListState extends State<SeasonsList> {
  @override
  Widget build(BuildContext context) {
    return SizedBox(
      height: 150,
      child: FutureBuilder(
          future: APIService()
              .getEpisodes(widget.movieId, widget.seasonNumber.toString()),
          builder: (context, AsyncSnapshot snapshot) {
            if (snapshot.hasData) {
              return ListView.builder(
                scrollDirection: Axis.horizontal,
                shrinkWrap: true,
                physics: const AlwaysScrollableScrollPhysics(
                    parent: BouncingScrollPhysics()),
                itemCount: snapshot.data.length,
                itemBuilder: (context, index) {
                  var url = snapshot.data[index].stillPath.toString();
                  return Container(
                    margin: const EdgeInsets.all(10),
                    decoration: BoxDecoration(
                      color: const Color(0xFF14303B).withOpacity(0.25),
                      borderRadius: BorderRadius.circular(10),
                      border: Border.all(
                        color: const Color(0xFFA3A3B0).withOpacity(0.5),
```

```
    width: 0.75,
  ),
),
child: Column(children: [
  Container(
    height: 100,
    width: 220,
    alignment: Alignment.bottomRight,
    decoration: BoxDecoration(
      borderRadius: const BorderRadius.only(
          topLeft: Radius.circular(10),
          topRight: Radius.circular(10)),
      image: DecorationImage(
          image: url == "null"
              ? const AssetImage("assets/LoadingImage.png")
                  as ImageProvider
              : NetworkImage(
                  "https://image.tmdb.org/t/p/w500$url"),
          fit: BoxFit.cover),
    ),
    child: Container(
      margin: const EdgeInsets.all(10),
      child: Text(
        snapshot.data[index].episodeNumber.toString(),
        style: const TextStyle(
            color: Colors.white,
            fontSize: 24,
            shadows: <Shadow>[
              Shadow(
                offset: Offset(0, 0),
                blurRadius: 10,
                color: Colors.black,
              ),
            ]),
      ),
    ),
  ),
  Container(
    margin: const EdgeInsets.only(left: 8, top: 4),
    width: 200,
    child: Text(snapshot.data[index].name.toString(),
        textAlign: TextAlign.left,
        maxLines: 1,
        overflow: TextOverflow.ellipsis,
```

```
                    style: const TextStyle(
                      color: Colors.white,
                    )),
                ),
              ]),
            );
          },
        );
      }
      return const Center(child: CircularProgressIndicator());
    }),
  );
 }
}
```

Output:-


5:38
8.1
Oppenheimer
Drama
History
Status
Released
Release: July 19, 2023
Overview
The story of J. Robert Oppenheimer's role in the development of the atomic bomb during World War II.
Recommendations
Barbie
THE CREATOR