<u>Experiment 3</u>

Shashwat Tripathi
D15A   Batch C
Roll No: 64

AIM: To design Flutter UI by including common widgets (using Images).

THEORY:
Widgets as Building Blocks:

Flutter apps are built using widgets, which are reusable components that represent UI elements.
Common widgets like Text, Image, Container, Row, Column, ListView, etc., provide building blocks for your UI.
Widgets can be nested within each other to create complex layouts and hierarchies.
Image Handling:

Flutter provides various ways to display images in your UI:
Asset Images: Images stored within your app's assets folder. Use AssetImage widget.
Network Images: Images loaded from URLs. Use NetworkImage widget.
Memory Images: Images loaded from memory buffers. Use MemoryImage widget.
File Images: Images loaded from local files. Use FileImage widget.
Each widget offers customization options like scaling, fitting, and alignment.
Layout and Composition:

Flutter uses a flexible layout system based on widgets.
Widgets like Container can define padding, margins, and background colors.
Layout widgets like Row and Column arrange children horizontally or vertically.
Stack widget allows layering widgets with positioning control.
Use padding, margins, and alignment properties to fine-tune the visual hierarchy.
Best Practices:

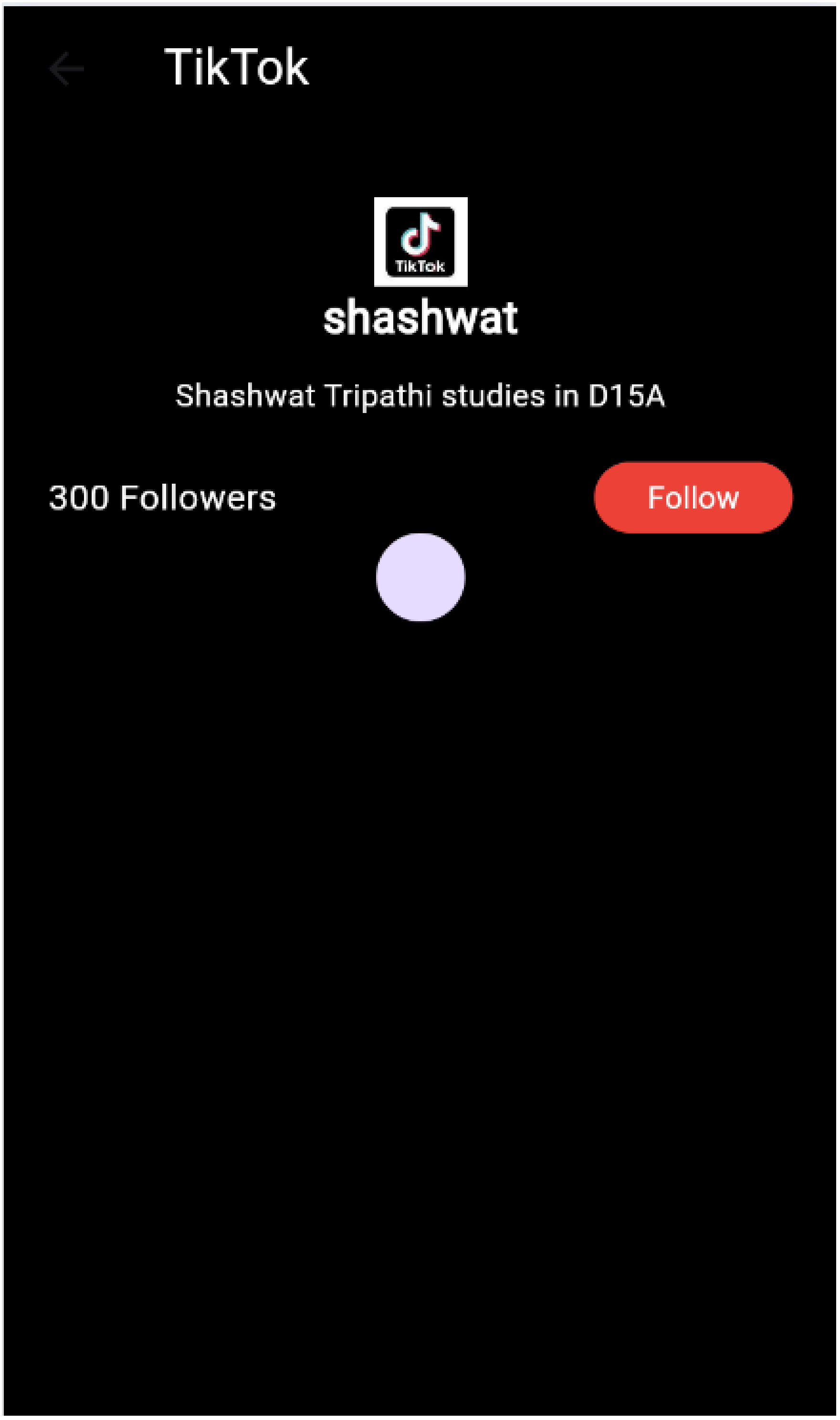
Choose the appropriate image widget based on the source and loading strategy.
Optimize image sizes and compression to improve performance.
Use placeholder widgets while images are loading to enhance user experience.
Consider using image caches to avoid redundant downloads.
Leverage Flutter's built-in animation features for smooth transitions and effects.
Follow accessibility guidelines to ensure your UI is usable for everyone.
Additional Considerations:

Explore advanced image manipulation packages like image and cached_network_image.
Experiment with different layout widgets and techniques to achieve desired UI structures.
Test your UI on different devices and screen sizes for responsiveness.
Consider incorporating image gestures like tapping, zooming, and panning.

Code:
profile.dart
```
import 'package:flutter/material.dart';
import
'package:flutter_tiktok_shashwat/features/user_auth/presentation/page
s/upload.dart';

class ProfilePage extends StatefulWidget {
  // Pass user information as arguments
  final String userName = "shashwat";
  final String bio = "Shashwat Tripathi studies in D15A";
  final String profileImageUrl = "";
  final int followerCount = 300;
  final bool isFollowing = false;

  const ProfilePage({
    Key? key,
    // required this.userName,
    // required this.bio,
    // required this.profileImageUrl,
    // required this.followerCount,
    // required this.isFollowing,
  }) : super(key: key);

  @override
  State<ProfilePage> createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
  bool _isFollowing = false;

  @override
  void initState() {
    super.initState();
    _isFollowing = widget.isFollowing;
  }

  void _toggleFollowing() {
    setState(() {
      _isFollowing = !_isFollowing;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor:
          Colors.black, // Set background color to black for TikTok
theme
      appBar: AppBar(
        backgroundColor: Colors.black, // Apply black background to
AppBar
        title: Text(
          "TikTok",
          style:
              TextStyle(color: Colors.white), // Set title text color
to white
        ),
      ),
      body: SingleChildScrollView(
        child: Padding(
          padding: const EdgeInsets.all(20.0),
          child: Column(
            children: [
              // Profile Image
              // CircleAvatar(
              //   backgroundImage:
NetworkImage(widget.profileImageUrl),
              //   radius: 50.0,
              // ),
              // SizedBox(height: 10.0),
              SizedBox(height: 10), // Adjust the height as needed
              Image.asset(
                'assets/tiktok_logo.jpg', // Replace with the correct
path to your TikTok logo image
                height: 40, // Adjust height as needed
              ),

              // Username
              Text(
                widget.userName,
                style: TextStyle(
                  fontSize: 20.0,
                  fontWeight: FontWeight.bold,
                  color: Colors.white, // Set text color to white
                ),
              ),
              SizedBox(height: 10.0),

              // Bio
              Text(
                widget.bio,
                style:
                    TextStyle(color: Colors.white), // Set text color
to white
              ),
              SizedBox(height: 20.0),

              // Follower Count
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                  Text(
                    "${widget.followerCount.toString()} Followers",
                    style: TextStyle(
                        fontSize: 16.0,
                        color: Colors.white), // Set text color to
white
                  ),
                  // Follow Button
                  ElevatedButton(
                    onPressed: _toggleFollowing,
                    child: Text(
                      _isFollowing ? "Following" : "Follow",
                      style: TextStyle(
                          color: Colors.white), // Set text color to
white
                    ),
                    style: ElevatedButton.styleFrom(
                      primary: Colors.red, // Set button color to red
                    ),
                  ),
                ],
              ),

              // ... other profile content (e.g., posts, etc.)

              Align(
                alignment: Alignment.bottomCenter,
                child: Padding(
                  padding: const EdgeInsets.only(bottom: 20.0),
                  // child: FloatingActionButton(
                  //   onPressed: () => Navigator.pushNamed(context,
                  //       '/upload'), // Replace with your upload
page route name
```

```
                  //   backgroundColor: Colors.red, // Set button
color to red
                  //   child: Icon(Icons.add),
                  // ),
                  child: GestureDetector(
                    onTap: () {
                      Navigator.of(context).push(
                        MaterialPageRoute(
                            builder: (context) =>
VideoUploaderPage()),
                      );
                    },
                    child: CircleAvatar(
                      backgroundImage:
NetworkImage(widget.profileImageUrl),
                      radius: 20.0,
                    ),
                  ),
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

OUTPUT :



CONCLUSION: Thus, we have used some common widgets like Images to create our login page of the application.