

MPL Experiment-2

Sanket More

D15A 29

AIM: To design Flutter UI using common widgets.

CODE:-

home.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_screenutil/flutter_screenutil.dart';

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

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

class _HomeScreenState extends State<HomeScreen> {
  final List<Map<String, dynamic>> _posts = [
    {
      'user': 'user1',
      'imageUrl': 'images/post.jpg',
      'caption': 'First post!',
      'likes': 100,
      'comments': 20,
    },
    {
      'user': 'user2',
      'imageUrl': 'images/post2.jpg',
      'caption': 'Another day, another post.',
      'likes': 50,
      'comments': 10,
    },
    // ... more posts
  ];

  final List<Map<String, dynamic>> _stories = [
    {
```

```

        'user': 'user1',
        'imageUrl': 'images/post.jpg', // Replace with your story image URLs
    },
    {
        'user': 'user2',
        'imageUrl': 'images/post2.jpg',
    },
    // ... more stories
];

```

```

@override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.white,
        appBar: AppBar(
            elevation: 0,
            toolbarHeight: 50.h,
            title: Image.asset(
                'images/instagram.jpg',
                height: 30.h,
            ),
        ),
        leading: Padding(
            padding: EdgeInsets.only(left: 16.w),
            child: Image.asset(
                'images/camera.jpg',
                height: 24.h,
            ),
        ),
        actions: [
            Padding(
                padding: EdgeInsets.only(right: 16.w),
                child: Icon(
                    Icons.favorite_border_outlined,
                    color: Colors.black,
                    size: 28.sp,
                ),
            ),
            Padding(
                padding: EdgeInsets.only(right: 16.w),
                child: Image.asset(

```

```

        'images/send.jpg',
        height: 24.h,
      ),
    ),
  ],
  backgroundColor: Colors.white,
  bottom: PreferredSize(
    preferredSize: Size.fromHeight(1.h),
    child: Container(
      height: 1.h,
      color: Colors.grey[300],
    ),
  ),
),
body: SingleChildScrollView(
  // Added SingleChildScrollView
  child: Column(
    children: [
      // Stories Section
      SizedBox(
        height: 100.h, // Adjust height as needed
        child: ListView.builder(
          scrollDirection: Axis.horizontal,
          itemCount: _stories.length,
          itemBuilder: (context, index) {
            return StoryWidget(_stories[index]);
          },
        ),
      ),
      // Posts Section
      ListView.builder(
        shrinkWrap: true, // Important for nested ListViews
        physics:
          const NeverScrollableScrollPhysics(), // Disable
scrolling of inner ListView
        itemCount: _posts.length,
        itemBuilder: (context, index) {
          return PostWidget(_posts[index]);
        },
      ),
    ],
  ),

```

```

        ],
    ),
),
);
}
}

class StoryWidget extends StatelessWidget {
    final Map<String, dynamic> storyData;

    const StoryWidget(this.storyData, {super.key});

    @override
    Widget build(BuildContext context) {
        return Padding(
            padding: const EdgeInsets.all(8.0),
            child: Column(
                children: [
                    CircleAvatar(
                        radius: 30.r,
                        backgroundImage: AssetImage(storyData['imageUrl'] ??
                            'images/default_profile.png'), // Use story image
                    ),
                    Text(storyData['user'] ?? ""),
                ],
            ),
        );
    }
}

class PostWidget extends StatelessWidget {
    final Map<String, dynamic> postData;

    const PostWidget(this.postData, {super.key});

    @override
    Widget build(BuildContext context) {
        return Column(
            children: [
                // User Info
            ],
        );
    }
}

```

```

    Row(
      children: [
        CircleAvatar(
          radius: 20.r,
          backgroundImage: const AssetImage('images/post2.jpg'),
        ),
        SizedBox(width: 10.w),
        Text(postData['user'] ?? ""),
      ],
    ),

    // Post Image
    Image.asset(postData['imageUrl'] ?? ""),

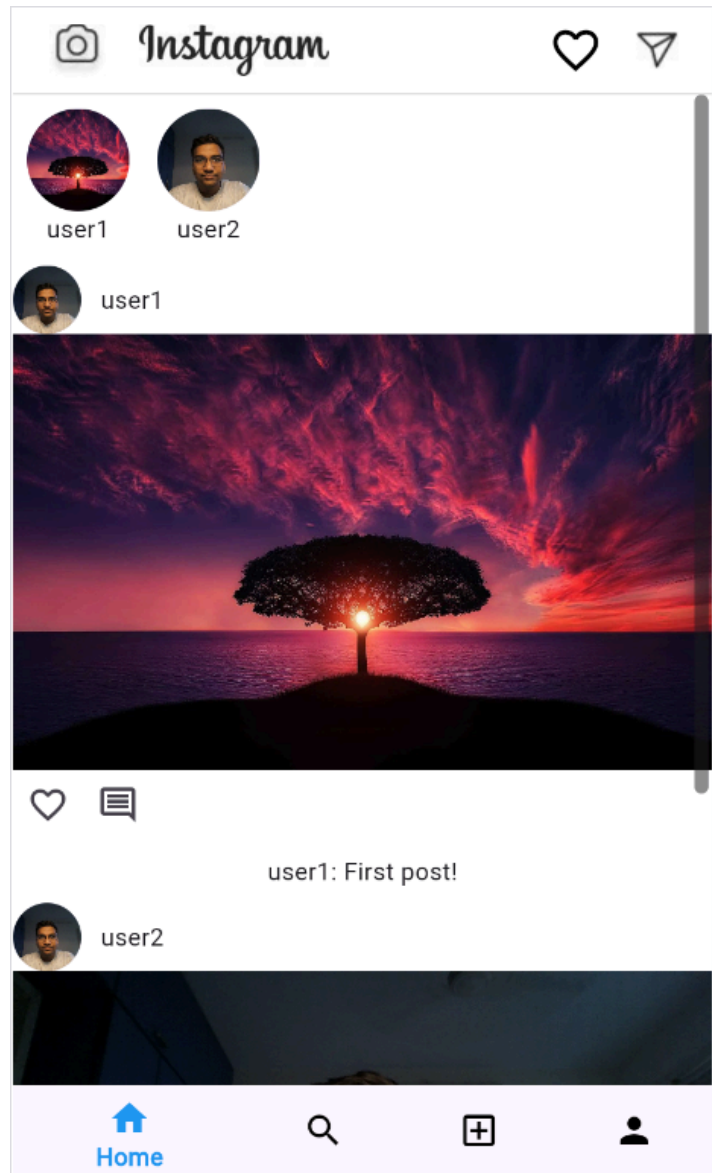
    // Post Actions (Likes, Comments)
    Row(
      children: [
        IconButton(
          icon: const Icon(Icons.favorite_border),
          onPressed: () {
            // Handle like
          },
        ),
        IconButton(
          icon: const Icon(Icons.comment_outlined),
          onPressed: () {
            // Handle comment
          },
        ),
      ],
    ),

    // Caption
    Padding(
      padding: const EdgeInsets.all(8.0),
      child:
        Text('${postData['user'] ?? ""}: ${postData['caption'] ??
""}',
    ),
  ],

```

```
} ;  
}  
}
```

OUTPUT:-



add_post_screen.dart:-

```
// add_post_screen.dart
import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';
import 'dart:io';

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

  @override
  State<AddPostScreen> createState() => _AddPostScreenState();
}

class _AddPostScreenState extends State<AddPostScreen> {
  File? _image;
  final TextEditingController _captionController =
    TextEditingController();
  bool _isLoading = false;
  List<String> _previousImageAssets = [];

  @override
  void initState() {
    super.initState();
    _loadPreviousImages();
  }

  Future<void> _loadPreviousImages() async {
    setState(() {
      _previousImageAssets = [
        'images/person.png', // Replace with your actual image paths
        'images/post.jpg', // Make sure these images are in your assets
        'images/cat.png', // Replace with a valid image in your assets
      ];
    });
  }

  Future<void> _pickImage(ImageSource source) async {
    final pickedFile = await ImagePicker().pickImage(source: source);

    setState(() {
```

```

        if (pickedFile != null) {
            _image = File(pickedFile.path);
        } else {
            print('No image selected.');
```

```

        }
    });
}

void _uploadPost() async {
    setState(() {
        _isLoading = true;
    });

    if (_image != null) {
        String caption = _captionController.text;

        // Simulate upload (replace with your actual logic)
        await Future.delayed(const Duration(seconds: 2));

        print("Uploading image: ${_image!.path}");
        print("Caption: $caption");

        setState(() {
            _image = null;
            _captionController.clear();
            _isLoading = false;
        });

        ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text('Post uploaded successfully!')),
        );
    } else {
        ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text('Please select an image')),
        );
        setState(() {
            _isLoading = false;
        });
    }
}
}

```



```

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.white,
    appBar: AppBar(
      backgroundColor: Colors.white,
      elevation: 1,
      leading: IconButton(
        icon: const Icon(Icons.arrow_back_ios, color: Colors.black),
        onPressed: () {
          Navigator.pop(context);
        },
      ),
      title: const Text('New Post',
        style: TextStyle(color: Colors.black, fontFamily:
'SFProText')),
      actions: [
        IconButton(
          onPressed: _isLoading ? null : _uploadPost,
          icon: _isLoading
            ? const CircularProgressIndicator()
            : const Icon(Icons.upload, color: Colors.blue),
        ),
      ],
    ),
    body: SingleChildScrollView(
      child: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            GestureDetector(
              onTap: () {
                showModalBottomSheet(
                  context: context,
                  builder: (BuildContext context) {
                    return Column(
                      mainAxisAlignment: MainAxisAlignment.min,
                      children: <Widget>[

```

```

        ListTile(
          leading: const Icon(Icons.photo_library),
          title: const Text('Photo Library'),
          onTap: () {
            _pickImage(ImageSource.gallery);
            Navigator.pop(context);
          },
        ),
        ListTile(
          leading: const Icon(Icons.camera_alt),
          title: const Text('Camera'),
          onTap: () {
            _pickImage(ImageSource.camera);
            Navigator.pop(context);
          },
        ),
      ],
    );
  },
);
),
child: Container(
  height: 300,
  width: double.infinity,
  decoration: BoxDecoration(
    border: Border.all(color: Colors.grey.shade300),
  ),
  child: _image != null
    ? Image.file(_image!, fit: BoxFit.cover)
    : const Center(
      child: Icon(Icons.add_a_photo,
        size: 50, color: Colors.grey)),
    ),
),
const SizedBox(height: 16),
TextField(
  controller: _captionController,
  style: const TextStyle(fontFamily: 'SFProText'),
  decoration: const InputDecoration(
    hintText: 'Write a caption...',

```

```


        hintStyle: TextStyle(fontFamily: 'SFProText'),
        border: InputBorder.none,
      ),
      maxLines: null,
    ),
    const SizedBox(height: 16),
    GridView.builder(
      shrinkWrap: true,
      physics: const NeverScrollableScrollPhysics(),
      gridDelegate: const
SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: 3,
        crossAxisSpacing: 8,
        mainAxisSpacing: 8,
      ),
      itemCount: _previousImageAssets.length,
      itemBuilder: (context, index) {
        return Image.asset(
          _previousImageAssets[index],
          fit: BoxFit.cover,
        );
      },
    ),
  ],
),
),
),
),
);
}
}


```

OUTPUT:-

New Post

Write a caption...





Add