

MPL Experiment-5

Sanket More

D15A 29

AIM: To apply navigation and routing in flutter application.

CODE:

navigation.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_screenutil/flutter_screenutil.dart';
import '../screen/home.dart';
import '../screen/explore.dart';
import '../screen/add_post_screen.dart';

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

  @override
  State<NavigationScreen> createState() => _NavigationScreenState();
}

class _NavigationScreenState extends State<NavigationScreen> {
  int _selectedIndex = 0;

  static const List<Widget> _widgetOptions = <Widget>[
    HomeScreen(),
    ExploreScreen(),
    AddPostScreen(),
  ];

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
```

```

body: Center(
  child: _widgetOptions.elementAt(_selectedIndex),
),
bottomNavigationBar: BottomNavigationBar(
  items: const <BottomNavigationBarItem>[
    BottomNavigationBarItem(
      icon: Icon(Icons.home),
      label: 'Home',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.search),
      label: 'Explore',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.add_box_outlined),
      label: 'Add',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.person),
      label: 'Profile',
    ),
  ],
  currentIndex: _selectedIndex,
  selectedItemColor: Colors.blue, // Highlight color
  unselectedItemColor: Colors.black, // Set unselected item color to
black
  selectedLabelStyle:
    const TextStyle(fontWeight: FontWeight.bold), // Bold selected
label
  onTap: _onItemTapped,
),
);
}
}

```

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...',

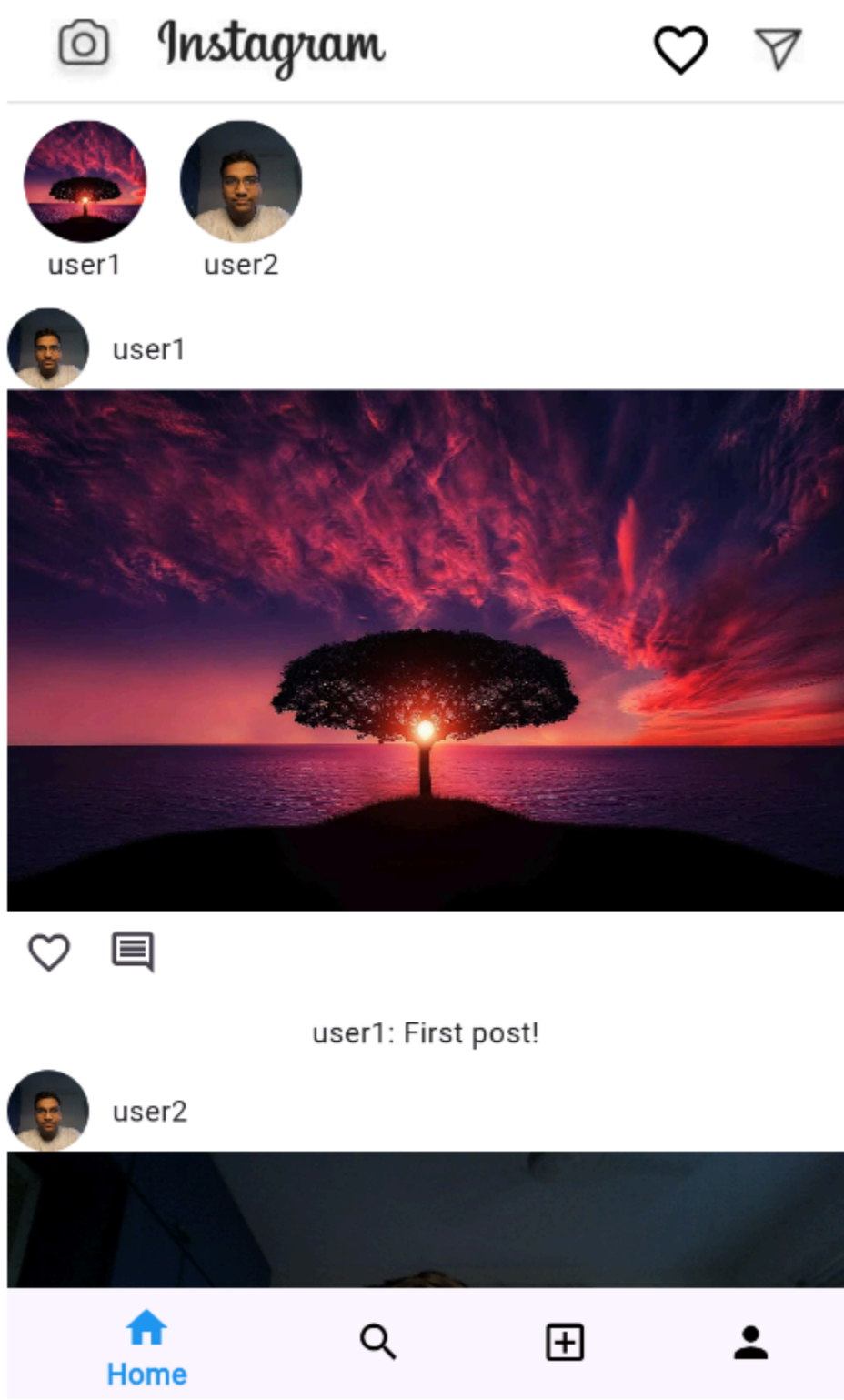
```

```

        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,
        );
      },
    ),
  ],
),
),
),
),
);
}
}

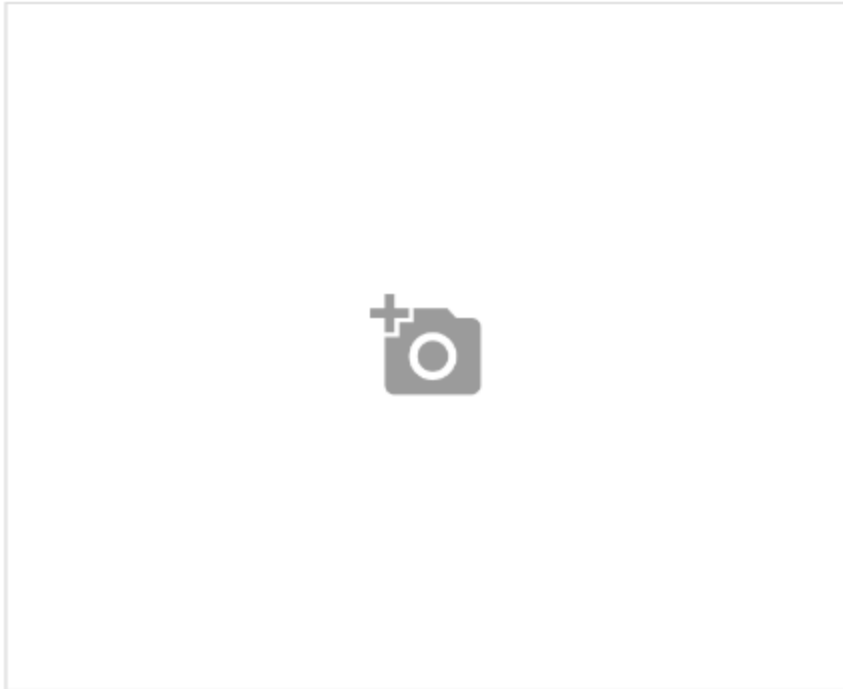
```

OUTPUT:-





New Post



Write a caption...





New Post



Write a caption...



Photo Library



Camera