

**MAD & PWA LAB**  
**Experiment 2**

**Name:** Niraj Kothawade

**Class:** D15A

**Roll no:** 24

**Aim:** To design Flutter UI by including common widgets.

**Code:**

**Landing page:**

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

import  
'package:whatsapp_messenger/common/  
routes/routes.dart';

import  
'package:whatsapp_messenger/common/  
utils/coloros.dart';

import  
'package:whatsapp_messenger/common/  
widgets/custom_elevated_button.dart';

import  
'package:whatsapp_messenger/feature/w  
elcome/widgets/language_button.dart';

import  
'package:whatsapp_messenger/feature/w  
elcome/widgets/privacy_and_terms.dart';

class WelcomePage extends  
StatelessWidget {  
  const WelcomePage({super.key});  
  
  navigateToLoginPage(context) {  
    Navigator.of(context)  
      .pushNamedAndRemoveUntil(Routes.  
login, (route) => false);  
  
    @override  
    Widget build(BuildContext context) {  
      return Scaffold(  
        body: Column(  
          children: [  
            Expanded(  
              child: Align(  
                alignment:  
                  Alignment.bottomCenter,  
                child: Padding(  
                  padding: const  
EdgeInsets.symmetric(horizontal: 50,  
vertical: 10),  
                  child:  
                    Image.asset('assets/images/circle.png',  
                      color: Coloors.greenDark),  
                ),  
              )),  
            const SizedBox(height: 40),  
            Expanded(  
              child: Column(  
                children: [  
                  Expanded(  
                    child: Align(  
                      alignment:  
                        Alignment.topCenter,  
                      child: Padding(  
                        padding: const  
EdgeInsets.symmetric(horizontal: 50,  
vertical: 10),  
                        child:  
                          Image.asset('assets/images/circle.png',  
                            color: Coloors.purpleDark),  
                      ),  
                    ),  
                  ),  
                ],  
              ),  
            ),  
          ],  
        ),  
      );  
    }  
  }  
}
```

```

children: [
  const Text(
    'Welcome to WhatsApp',
    style: TextStyle(
      fontSize: 22,
      fontWeight: FontWeight.bold,
    ),
  ),
  const PrivacyAndTerms(),
  CustomElevatedButton(
    onPressed: () =>
    navigateToLoginPage(context),
    text: 'AGREE AND CONTINUE',
  ),
  const SizedBox(height: 50),
  LanguageButton(),
],
),
],
),
],
),
);
}
}
}

Authentification page:

```

```

import 'package:flutter/material.dart';
import
'package:flutter_riverpod/flutter_riverpod
.dart';

```

```

import
'package:whatsapp_messenger/common/
extension/custom_theme_extension.dart';
import
'package:whatsapp_messenger/feature/auth/controller/auth_controller.dart';
import
'package:whatsapp_messenger/feature/auth/widgets/custom_text_field.dart';

class VerificationPage extends ConsumerWidget {
  const VerificationPage({
    super.key,
    required this.smsCodeId,
    required this.phoneNumber,
  });

final String smsCodeId;
final String phoneNumber;

Future<void> verifySmsCode(
  BuildContext context,
  WidgetRef ref,
  String smsCode,
) async {
  await
ref.read(authControllerProvider).verifySms
Code(
  context: context,
  smsCodeId: smsCodeId,
  smsCode: smsCode,
)
}
}
```

```
);

}

@Override
Widget build(BuildContext context,
WidgetRef ref) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor:
        Theme.of(context).scaffoldBackgroundColor,
      elevation: 0,
      centerTitle: true,
      title: Text(
        'Verify your number',
        style: TextStyle(
          color:
            context.theme.authAppbarTextColor,
        ),
      ),
      actions: [
        IconButton(
          onPressed: () {},
          icon: const Icon(Icons.more_vert),
        ),
      ],
    ),
    body: SingleChildScrollView(
      padding: const
        EdgeInsets.symmetric(horizontal: 20),
      child: Column(
        children: [
          Padding(
            padding: const
              EdgeInsets.symmetric(horizontal: 10),
            child: RichText(
              textAlign: TextAlign.center,
              text: TextSpan(
                style: TextStyle(color:
                  context.theme.greyColor),
                children: [
                  const TextSpan(
                    text: "You've tried to register
before requesting an SMS or Call with your
code."),
                ],
              ),
              TextSpan(
                text: " Wrong number?",
                style: TextStyle(
                  color:
                    context.theme.blueColor,
                ),
              ),
            ],
          ),
          const SizedBox(height: 20),
          Container(
            padding: const
              EdgeInsets.symmetric(horizontal: 80),
```

```
        child: CustomTextField(                ),  
        hintText: "-----",                ],  
        fontSize: 30,                ),  
        autoFocus: true,                const SizedBox(height: 10),  
        keyboardType:                Divider(  
        TextInputType.number,                color:  
        onChanged: (value) async {                context.theme.greyColor!.withOpacity(.2),  
            if (value.length == 6) {                ),  
                await verifySmsCode(context,  
                ref, value);                const SizedBox(height: 10),  
            }                Row(  
            },                children: [  
            Icon(Icons.phone, color:  
            context.theme.greyColor),  
            const SizedBox(width: 20),  
            ),                Text(  
            const SizedBox(height: 20),  
            Text(  
            'Enter 6-digit code',  
            style: TextStyle(color:  
            context.theme.greyColor),  
            ),                ),  
            const SizedBox(height: 30),  
            Row(  
            children: [  
            Icon(Icons.message, color:  
            context.theme.greyColor),  
            const SizedBox(width: 20),  
            Text(  
            'Resend SMS',  
            style: TextStyle(  
            color: context.theme.greyColor,  
            ),  
            ),  
            );  
        }  
    ),  
);
```

### Chats page:

```
import  
'package:cached_network_image/cached_  
network_image.dart';  
  
import 'package:flutter/material.dart';  
  
import  
'package:flutter_riverpod/flutter_riverpod  
.dart';  
  
import 'package:intl/intl.dart';  
  
import  
'package:whatsapp_messenger/common/  
extension/custom_theme_extension.dart';  
  
import  
'package:whatsapp_messenger/common/  
models/last_message_model.dart';  
  
import  
'package:whatsapp_messenger/common/  
models/user_model.dart';  
  
import  
'package:whatsapp_messenger/common/  
routes/routes.dart';  
  
import  
'package:whatsapp_messenger/common/  
utils/color.dart';  
  
import  
'package:whatsapp_messenger/feature/ch  
at/controller/chat_controller.dart';  
  
  
class ChatHomePage extends  
ConsumerWidget {  
  
  const ChatHomePage({super.key});  
  
  List<LastMessageModel>  
  getDemoChats() {  
  
    return [
```

```
    LastMessageModel(  
      username: 'Dev Stack',  
      contactId: '1',  
      profileImageUrl:  
        'https://picsum.photos/seed/1/200',  
      lastMessage: 'Hi Everyone 🙌',  
      timeSent:  
        DateTime.now().subtract(const  
          Duration(minutes: 4)),  
    ),  
  
    LastMessageModel(  
      username: 'Kishor',  
      contactId: '2',  
      profileImageUrl:  
        'https://picsum.photos/seed/2/200',  
      lastMessage: 'Hi Kishor',  
      timeSent:  
        DateTime.now().subtract(const  
          Duration(hours: 1)),  
    ),  
  
    LastMessageModel(  
      username: 'Dev Server Chat',  
      contactId: '3',  
      profileImageUrl:  
        'https://picsum.photos/seed/3/200',  
      lastMessage: 'Alice: The new API is  
        ready for testing',  
      timeSent:  
        DateTime.now().subtract(const  
          Duration(hours: 2)),  
    ),  
  
    LastMessageModel(
```

```
username: 'Collins',
contactId: '4',
profileImageUrl:
'https://picsum.photos/seed/4/200',
lastMessage: 'Hey Dev Stack! How\'s
the project going?',
timeSent:
DateTime.now().subtract(const
Duration(hours: 4)),
),
LastMessageModel(
username: 'Friends',
contactId: '5',
profileImageUrl:
'https://picsum.photos/seed/5/200',
lastMessage: 'John: Anyone up for
coffee? ☕',
timeSent:
DateTime.now().subtract(const
Duration(hours: 5)),
),
LastMessageModel(
username: 'Flutter Community',
contactId: '6',
profileImageUrl:
'https://picsum.photos/seed/6/200',
lastMessage: 'New flutter 3.0 features
announced! 🎉',
timeSent:
DateTime.now().subtract(const
Duration(hours: 10)),
),
LastMessageModel(
username: 'Balram Rathore',
contactId: '7',
profileImageUrl:
'https://picsum.photos/seed/7/200',
lastMessage: 'happy coding
🔥 🔥 🔥 ',
timeSent:
DateTime.now().subtract(const
Duration(hours: 10)),
),
];
}

@Override
Widget build(BuildContext context,
WidgetRef ref) {
return Scaffold(
backgroundColor: const
Color(0xFF121B22),
body: ListView.builder(
itemCount: getDemoChats().length,
shrinkWrap: true,
itemBuilder: (context, index) {
final lastMessageData =
getDemoChats()[index];
return Column(
children: [
ListTile(
onTap: () {
Navigator.pushNamed(

```

```
        context,
        ),
      Routes.chat,
      arguments: UserModel(
        username:
      lastMessageData.username,
        uid:
      lastMessageData.contactId,
        profileImageUrl:
      lastMessageData.profileImageUrl,
        active: true,
        lastSeen: 0,
        phoneNumber: '0',
        groupId: [],
      ),
    );
  },
  title: Row(
    mainAxisAlignment:
  MainAxisAlignment.spaceBetween,
    children: [
      Expanded(
        child: Text(
          lastMessageData.username,
          style: const TextStyle(
            fontWeight:
          FontWeight.w500,
          color: Colors.white,
        ),
        overflow:
      TextOverflow.ellipsis,
    ),
    Text(
      DateFormat.Hm().format(last
    MessageData.timeSent),
      style: const TextStyle(
        fontSize: 13,
        color: Colors.grey,
      ),
    ],
  ),
  subtitle: Padding(
    padding: const
  EdgeInsets.only(top: 3),
    child: Row(
      children: [
        const Icon(
          Icons.done_all,
          size: 16,
          color: Coloors.greenDark,
        ),
        const SizedBox(width: 3),
        Expanded(
          child: Text(
            lastMessageData.lastMessa
          ge,
            maxLines: 1,
            overflow:
          TextOverflow.ellipsis,
            style: const TextStyle(
              color: Colors.grey,
            ),
          ),
        ),
      ],
    ),
  ),

```

```
        ),
        child: const Icon(
            Icons.message,
            color: Colors.white,
        ),
    ),
),
);
};

leading: CircleAvatar(
backgroundImage: CachedNetworkImageProvider(
lastMessageData.profileImage
Url,
),
radius: 24,
),
),
Divider(
color: Colors.grey[900],
indent: 80, // Enhanced Status Mode
height: 0,
),
],
);
},
floatingActionButton:
FloatingActionButton(
backgroundColor: Coloors.greenDark,
onPressed: () {
    Navigator.pushNamed(context,
Routes.contact);
},
),
),
),
);
};

Status({
required this.id,
```

## Status Page:

```
import 'dart:html' as html;  
  
import 'package:flutter/material.dart';  
  
import  
'package:image_picker/image_picker.dart'  
;  
  
// Enhanced Status Model  
  
class Status {  
  
    final String id;  
  
    final String name;  
  
    final String imageUrl; // Changed from  
    imagePath to imageUrl  
  
    final DateTime timestamp;  
  
    final bool isViewed;  
  
    final StatusType type;  
  
    final String? caption;  
  
    Status({  
        required this.id,  
        required this.name,  
        required this.imageUrl,  
        required this.timestamp,  
        required this.isViewed,  
        required this.type,  
        this.caption,  
    });  
  
    Map<String, dynamic> toMap() {  
        return {  
            'id': id,  
            'name': name,  
            'imageUrl': imageUrl,  
            'timestamp': timestamp,  
            'isViewed': isViewed,  
            'type': type,  
            'caption': caption,  
        };  
    }  
  
    factory Status.fromMap(Map<String, dynamic> map) {  
        return Status(  
            id: map['id'],  
            name: map['name'],  
            imageUrl: map['imageUrl'],  
            timestamp: map['timestamp'],  
            isViewed: map['isViewed'],  
            type: map['type'],  
            caption: map['caption'],  
        );  
    }  
}
```

```

        required this.name,
        required this.imageUrl,
        required this.timestamp,
        this.isViewed = false,
        this.type = StatusType.image,
        this.caption,
    });
}

enum StatusType { image, text }

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

    @override
    State<StatusHomePage> createState() =>
    _StatusHomePageState();
}

class _StatusHomePageState extends
State<StatusHomePage> {
    final ImagePicker _picker = ImagePicker();
    List<Status> recentUpdates = [];
    List<Status> viewedUpdates = [];
    Status? myCurrentStatus;

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

void _loadDemoStatuses() {
    // Load demo statuses with web-
    // compatible URLs
    recentUpdates = [
        Status(
            id: '1',
            name: 'Balram Rathore',
            imageUrl:
            'https://picsum.photos/400/600', // Using
            placeholder image
            timestamp: DateTime.now(),
        ),
        Status(
            id: '2',
            name: 'Kishor Kumar',
            imageUrl:
            'https://picsum.photos/400/601', // Different
            placeholder
            timestamp: DateTime.now(),
        ),
    ];
    viewedUpdates = [
        Status(
            id: '3',
            name: 'Saket Sinha',
            imageUrl:
            'https://picsum.photos/400/602',
        );
    ];
}

```

```

    timestamp: // Add to recent updates
    DateTime.now().subtract(const
    Duration(hours: 1)),
    isViewed: true,
),
];
}

Future<void> _addImageStatus() async {
try {
    final XFile? image = await
    _picker.pickImage(source:
    ImageSource.gallery);
    if (image == null) return;

    // Create object URL for web
    final bytes = await
    image.readAsBytes();
    final blob = html.Blob([bytes]);
    final url =
    html.Url.createObjectUrlFromBlob(blob);
}

setState(() {
    myCurrentStatus = Status(
        id: DateTime.now().toString(),
        name: 'My Status',
        imageUrl: url,
        timestamp: DateTime.now(),
        type: StatusType.image,
    );
}

// Add to recent updates
recentUpdates.insert(0,
myCurrentStatus!);
});

} catch (e) {
showDialog(
context: context,
builder: (context) => AlertDialog(
title: const Text('Error'),
content: Text('Failed to add status:
$e'),
actions: [
TextButton(
 onPressed: () =>
Navigator.pop(context),
child: const Text('OK'),
),
],
),
);
}
}

Future<void> _addTextStatus() async {
final TextEditingController textController
= TextEditingController();
final result = await showDialog<String>(
context: context,
builder: (context) => AlertDialog(
backgroundColor: const
Color(0xFF121B22),

```

```

title: const Text('Add Text Status',
  style: TextStyle(color:
Colors.white)),
content: TextField(
  controller: textController,
  style: const TextStyle(color:
Colors.white),
  maxLines: 3,
  decoration: const InputDecoration(
    hintText: 'Enter your status...',
    hintStyle: TextStyle(color:
Colors.grey),
    border: OutlineInputBorder(),
  ),
),
actions: [
  TextButton(
    onPressed: () =>
Navigator.pop(context),
    child: const Text('Cancel'),
  ),
  TextButton(
    onPressed: () =>
Navigator.pop(context,
textController.text),
    child: const Text('Post'),
  ),
],
),
);

if (result != null && result.isNotEmpty) {
  setState(() {
    myCurrentStatus = Status(
      id: DateTime.now().toString(),
      name: 'My Status',
      imageUrl: '', // Empty for text status
      timestamp: DateTime.now(),
      type: StatusType.text,
      caption: result,
    );
  });
}

// Add to recent updates
recentUpdates.insert(0,
myCurrentStatus!);
});

}

}

void _viewStatus(Status status) {
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) =>
StatusViewPage(status: status),
    ),
  );
}

// Mark status as viewed
if (!status.isViewed &&
recentUpdates.contains(status)) {

```

```
        setState(() {  
            final index =  
recentUpdates.indexOf(status);  
            final viewedStatus = Status(  
                id: status.id,  
                name: status.name,  
                imageUrl: status.imageUrl,  
                timestamp: status.timestamp,  
                isViewed: true,  
                type: status.type,  
                caption: status.caption,  
            );  
            recentUpdates.removeAt(index);  
            viewedUpdates.insert(0,  
viewedStatus);  
        });  
    }  
}  
  
@override  
Widget build(BuildContext context) {  
    return Scaffold(  
        backgroundColor: const  
Color(0xFF121B22),  
        body: SingleChildScrollView(  
            child: Column(  
                crossAxisAlignment:  
CrossAxisAlignment.start,  
                children: [  
// My Status Section  
        ListTile(  
            leading: Stack(  
                children: [  
                    CircleAvatar(  
                        radius: 30,  
                        backgroundColor: Colors.grey,  
                        backgroundImage:  
myCurrentStatus?.type ==  
StatusType.image  
? NetworkImage(myCurrentStatus!.imageUr  
l)  
: null,  
child: myCurrentStatus == null  
? const Icon(Icons.person,  
color: Colors.white, size:  
35  
: myCurrentStatus?.type ==  
StatusType.text  
? const  
Icon(Icons.text_fields, color: Colors.white)  
: null,  
,  
Positioned(  
bottom: 0,  
right: 0,  
child: Container(  
padding: const  
EdgeInsets.all(4),  
decoration: BoxDecoration(  
color: Colors.green,  

```



```

if (viewedUpdates.isNotEmpty) ...[
  const Padding(
    padding: EdgeInsets.symmetric(horizontal: 15,
    vertical: 8),
    child: Text(
      'Viewed updates',
      style: TextStyle(
        color: Colors.grey,
        fontWeight: FontWeight.bold,
      ),
    ),
  ),
  ListView.builder(
    shrinkWrap: true,
    physics: const
    NeverScrollableScrollPhysics(),
    itemCount:
    viewedUpdates.length,
    itemBuilder: (context, index) =>
    StatusTile(
      status: viewedUpdates[index],
      onTap: () =>
      _viewStatus(viewedUpdates[index]),
    ),
  ],
],
),
floatingActionButton: Column(
  mainAxisSize: MainAxisSize.end,
  children: [
    FloatingActionButton(
      mini: true,
      onPressed: _addTextStatus,
      backgroundColor: Colors.grey[700],
      child: const Icon(Icons.edit),
    ),
    const SizedBox(height: 16),
    FloatingActionButton(
      onPressed: _addImageStatus,
      backgroundColor: Colors.teal[700],
      child: const Icon(Icons.camera_alt),
    ),
  ],
);
}

class StatusTile extends StatelessWidget {
final Status status;
final VoidCallback onTap;

const StatusTile({
super.key,
required this.status,
required this.onTap,
});

}

```

```
        fontWeight: FontWeight.bold,  
        ),  
      ),  
      subtitle: Text(  
        'Today,  
        ${status.timestamp.hour}:${status.timestamp.minute.toString().padLeft(2, '0')}',  
        style: const TextStyle(color:  
        Colors.grey),  
        ),  
        onTap: onTap,  
      );  
    }  
  }  
  
class StatusViewPage extends  
StatelessWidget {  
  final Status status;  
  
  const StatusViewPage({super.key,  
  required this.status});  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      backgroundColor: Colors.black,  
      appBar: AppBar(  
        backgroundColor: Colors.transparent,  
        title: Text(status.name),  
        iconTheme: const  
        IconThemeData(color: Colors.white),  
        ),  
      ),  
      title: Text(  
        status.name,  
        style: const TextStyle(  
        color: Colors.white,  
        ),  
      ),  
    );  
  }  
}
```



```
class CallHomePage extends StatefulWidget {  
  const CallHomePage({super.key});  
  
  @override  
  State<CallHomePage> createState() =>  
  _CallHomePageState();  
}  
  
class _CallHomePageState extends State<CallHomePage> {  
  
  // Expanded demo call data with Picsum  
  // images  
  
  final List<Call> calls = [  
    Call(  
      name: "Mike Johnson",  
      avatarUrl:  
        "https://picsum.photos/seed/mike/200",  
      timestamp: DateTime.now().subtract(const  
        Duration(minutes: 5)),  
      isIncoming: true,  
      isVideo: false,  
    ),  
    Call(  
      name: "Sarah Wilson",  
      avatarUrl:  
        "https://picsum.photos/seed/sarah/200",  
      timestamp: DateTime.now().subtract(const  
        Duration(minutes: 15)),  
      isIncoming: false,  
      isVideo: true,  
    ),  
    Call(  
      name: "Alex Thompson",  
      avatarUrl:  
        "https://picsum.photos/seed/alex/200",  
      timestamp: DateTime.now().subtract(const  
        Duration(hours: 2)),  
      isIncoming: true,  
      isMissed: true,  
      isVideo: false,  
    ),  
    Call(  
      name: "Emma Davis",  
      avatarUrl:  
        "https://picsum.photos/seed/emma/200",  
      timestamp: DateTime.now().subtract(const  
        Duration(hours: 3)),  
      isIncoming: false,  
      isVideo: true,  
    ),  
    Call(  
      name: "James Miller",  
      avatarUrl:  
        "https://picsum.photos/seed/james/200",  
      timestamp: DateTime.now().subtract(const  
        Duration(hours: 5)),  
      isIncoming: true,  
      isMissed: false,  
      isVideo: true,  
    ),  
    Call(  
      name: "Lisa Brown",  
      avatarUrl:  
        "https://picsum.photos/seed/lisa/200",  
      timestamp: DateTime.now().subtract(const  
        Duration(hours: 6)),  
    ),  
  ];
```

```

isIncoming: false,
isMissed: true,
isVideo: false,
),
Call(
  name: "David Clark",
  avatarUrl:
"https://picsum.photos/seed/david/200",
  timestamp: DateTime.now().subtract(const
Duration(hours: 8)),
  isIncoming: true,
  isVideo: false,
),
Call(
  name: "Rachel Green",
  avatarUrl:
"https://picsum.photos/seed/rachel/200",
  timestamp: DateTime.now().subtract(const
Duration(days: 1)),
  isIncoming: false,
  isVideo: true,
),
Call(
  name: "Tom Wilson",
  avatarUrl:
"https://picsum.photos/seed/tom/200",
  timestamp: DateTime.now().subtract(const
Duration(days: 1, hours: 3)),
  isIncoming: true,
  isMissed: true,
  isVideo: false,
),
Call(
  name: "Emily White",
  avatarUrl:
"https://picsum.photos/seed/emily/200",
  timestamp: DateTime.now().subtract(const
Duration(days: 1, hours: 6)),
  isIncoming: false,
  isVideo: true,
);
]

void _showOutgoingCallDialog(
BuildContext context, String name, bool
isVideo) {
  showDialog(
    context: context,
    barrierDismissible: false,
    builder: (context) => Align(
      alignment: Alignment.topCenter,
      child: OutgoingCallDialog(name: name,
      isVideo: isVideo),
    ),
  );
}

String _getTimeString(DateTime timestamp) {
  final now = DateTime.now();
  final difference =
now.difference(timestamp);

  if (difference.inDays > 0) {
    return "Yesterday,
${timestamp.hour.toString().padLeft(2,
'0')}:${timestamp.minute.toString().padLeft(2,
'0')}";
  }
}

```

```

} else if (difference.inHours > 0) {
    return "${difference.inHours}h ago";
} else {
    return
    "${timestamp.hour.toString().padLeft(2,
    '0')}:${timestamp.minute.toString().padLeft(2,
    '0')}";
}
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: const
        Color(0xFF1F2C34),
        body: SafeArea(
            child: Column(
                children: [
                    Expanded(
                        child: ListView.builder(
                            itemCount: calls.length,
                            itemBuilder: (context, index) {
                                final call = calls[index];
                                return ListTile(
                                    leading: CircleAvatar(
                                        radius: 25,
                                        backgroundImage:
                                        NetworkImage(call.avatarUrl),
                                        backgroundColor:
                                        Colors.grey[700],
                                    ),
                                    title: Text(
                                        call.name,
                                    ),
                                    subtitle: Row(
                                        children: [
                                            Icon(
                                                call.isIncoming
                                                ? Icons.call_received
                                                : Icons.call_made,
                                                size: 15,
                                                color: call.isMissed ? Colors.red
                                                : Colors.green,
                                            ),
                                            const SizedBox(width: 4),
                                            Text(
                                                _getTimeString(call.timestamp),
                                                style: TextStyle(
                                                    color: Colors.grey[400],
                                                    fontSize: 13,
                                                ),
                                            ),
                                            ],
                                        ),
                                    trailing: IconButton(
                                        icon: Icon(
                                            call.isVideo ? Icons.videocam :
                                            Icons.call,
                                        ),
                                        color: const Color(0xFF00A884),
                                    ),
                                    onPressed: () {

```



```
        color: Colors.white,                }
        fontSize: 16,                      }
        fontWeight: FontWeight.bold,
    ),
),
Text(
    "Calling${isVideo ? ' video' : "...",
    style: TextStyle(
        color: Colors.grey[400],
        fontSize: 14,
    ),
),
],
),
),
IconButton(
    icon: const Icon(Icons.close, color:
Colors.red),
    onPressed: () =>
Navigator.of(context).pop(),
),
IconButton(
    icon: Icon(
        isVideo ? Icons.videocam : Icons.call,
        color: Colors.green,
    ),
    onPressed: () =>
Navigator.of(context).pop(),
),
],
),
);
```

**Output:**







