MPL Experiment 2

Name: Prathamesh Palve **Class:** D15A **Roll no:**31

Aim: To design Flutter UI by including common widgets.

Theory:

Designing a **ChatGPT-like UI** in Flutter involves utilizing various widgets to create a clean, structured, and interactive chat interface. Layout widgets like **Container, Column, Row, and ListView** help organize chat bubbles, while **GestureDetector, InkWell, and TextField** enhance user interaction.

For displaying chat messages, widgets like **Text, Card, and ListView** are essential. Navigation widgets such as **Drawer, AppBar, and Navigator** ensure smooth transitions between different sections of the app. Managing state using **setState, Provider, or Riverpod** ensures real-time updates to chat messages and responses.

Key UI Components for a ChatGPT App

ListView (reversed) – Displays chat messages in a conversation format from bottom to top.
TextField with a Send Button – Allows users to input messages.
ChatBubble Widget – Custom-designed widget to differentiate user and AI responses.
Bottom Navigation or Drawer – Provides easy navigation to different sections like settings, saved chats, and history.

Steps:

Step 1: Create a new Flutter project or open an existing one.

Set up a Flutter project and add necessary dependencies like `firebase_auth` and `cloud_firestore` for chat storage.

Step 2: Design the layout using Scaffold, incorporating AppBar and Drawer.

- The **AppBar** contains options like settings and premium subscriptions.
- The **Drawer (LeftSlider)** helps navigate between different chat conversations.

Step 3: Implement ListView (with reverse: true) for displaying chat messages.

- Fetch messages from Firestore using a **StreamBuilder**.
- Display chat messages using a **ChatBubble widget**.
- Ensure new messages appear at the bottom (like ChatGPT's UI).

Step 4: Add a TextField and a Send Button.

- Use a `TextField` for message input.
- Implement an **onSend function** that:
  - Saves the user's message in Firestore.
  - Calls an API (e.g., Hugging Face or a local model) for AI-generated responses.
  - Stores the AI's response in Firestore.

Step 5: Use ChatBubble Widgets to Differentiate User and AI Responses.

- User messages appear aligned to the right.

- AI responses appear aligned to the left.

Step 6: Implement a Scroll Controller to Auto-Scroll New Messages.

- Ensure the chat screen scrolls to the bottom when a new message arrives.
- Set `reverse: true` in `ListView` to display messages from bottom to top.

Step 7: Manage State Using setState or Provider.

- Use `setState` for simple state management.
- For complex applications, integrate **Provider or Riverpod** to handle chat state across multiple screens

Code:

```
//home_.dart
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import '../services/api_service.dart';
import '../services/firestore_service.dart';
import '../widgets/chat_bubble.dart';
import '../widgets/empty_state.dart';
import '../widgets/bottom_bar.dart';
import 'setting.dart';
import 'left_slidder.dart';

class HomeScreen extends StatefulWidget {
 final String chatId; // Accepts chatId dynamically

 HomeScreen({required this.chatId});

 @override
 _HomeScreenState createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
 final TextEditingController _messageController = TextEditingController();
 final FirebaseAuth _auth = FirebaseAuth.instance;
 final FirestoreService _firestoreService = FirestoreService();
 final ApiService _apiService = ApiService();
 final ScrollController _scrollController = ScrollController();

 void _sendMessage() async {
   final user = _auth.currentUser;
   if (user == null || _messageController.text.trim().isEmpty) return;

   String userId = user.uid;
   String chatId = widget.chatId;
   String userMessage = _messageController.text.trim();

   try {
     // Store user message in Firestore
     await _firestoreService.addMessage(userId, chatId, userMessage, true);

     setState(() {
       _messageController.clear();
     });

     // Scroll to bottom after sending message
     Future.delayed(Duration(milliseconds: 300), () {
       _scrollToBottom();
     });

     // Get response from Hugging Face API
     String aiResponse = await _apiService.getHuggingFaceResponse(userMessage);

     // Store AI response in Firestore
     await _firestoreService.addMessage(userId, chatId, aiResponse, false);

     print("✅ AI Response stored successfully!");
   } catch (e) {
     print("❌ Error sending message: $e");
   }
```

```
}

void _scrollToBottom() {
  if (_scrollController.hasClients) {
    _scrollController.animateTo(
      0, // Since we reversed ListView, scroll to position 0
      duration: Duration(milliseconds: 300),
      curve: Curves.easeOut,
    );
  }
}

@override
Widget build(BuildContext context) {
  final user = _auth.currentUser;
  if (user == null) {
    return Scaffold(
      body: Center(child: Text("User not logged in")),
    );
  }

  return Scaffold(
    backgroundColor: Colors.black,
    appBar: AppBar(
      backgroundColor: Colors.black,
      elevation: 0,
      leading: Builder(
        builder: (context) => IconButton(
          icon: Icon(Icons.menu, color: Colors.white),
          onPressed: () {
            Scaffold.of(context).openDrawer();
          },
        ),
      ),
      actions: [
        TextButton(
          onPressed: () {},
          child: Text("Get Plus ✨", style: TextStyle(color: Colors.white)),
        ),
        IconButton(
          icon: Icon(Icons.more_vert, color: Colors.white),
          onPressed: () {},
        ),
      ],
    ),
    drawer: LeftSlider(currentChatId: widget.chatId),
    body: Column(
      children: [
        Expanded(
          child: StreamBuilder<QuerySnapshot>(
            stream: _firestoreService.getMessages(user.uid, widget.chatId),
            builder: (context, snapshot) {
              if (snapshot.connectionState == ConnectionState.waiting) {
                return Center(child: CircularProgressIndicator());
              }

              if (!snapshot.hasData || snapshot.data!.docs.isEmpty) {
                return EmptyState(); // Show empty state if no messages
              }

              return ListView(
                 controller: _scrollController,
                 padding: EdgeInsets.all(16),
                 reverse: true, // 🔥 Messages appear from bottom to top
                 children: snapshot.data!.docs.map((doc) {
                   Map<String, dynamic> data = doc.data() as Map<String,
dynamic>;
                   bool isUser = data['isUser'] ?? false;
                   return ChatBubble(message: data['message'] ?? '', isUser:
isUser);
                 }).toList(),
               );
             },
           ),
         ),
         BottomBar(
           messageController: _messageController,
           onSendPressed: _sendMessage,
         ),
       ],
     ),
   );
 }
}
```

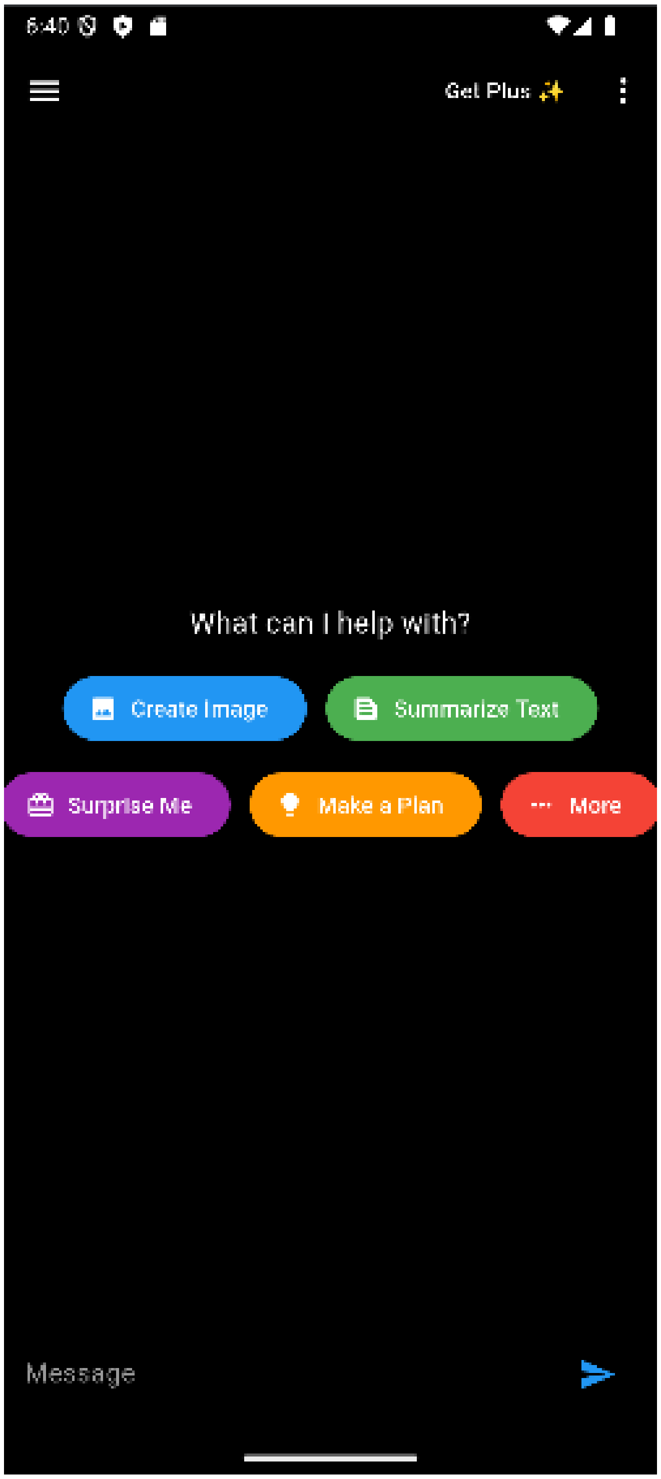
//account.dart

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

class SettingsScreen extends StatelessWidget {
 final FirebaseAuth _auth = FirebaseAuth.instance;

 @override
 Widget build(BuildContext context) {
   final User? user = _auth.currentUser;

   return Scaffold(
     backgroundColor: Colors.black,
     appBar: AppBar(
       backgroundColor: Colors.black,
       elevation: 0,
       leading: IconButton(
         icon: Icon(Icons.arrow_back, color: Colors.white),
         onPressed: () {
           Navigator.pop(context);
```

```
          },
        ),
        title: Text("Settings", style: TextStyle(color: Colors.white)),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: [
            Row(
              children: [
                CircleAvatar(
                  radius: 30,
                  backgroundColor: Colors.blue,
                  child: Text(
                    user?.email != null ? user!.email![0].toUpperCase() : "U",
                    style: TextStyle(color: Colors.white),
                  ),
                ),
                SizedBox(width: 12),
                Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(
                      user?.displayName ?? "User",
                      style: TextStyle(color: Colors.white, fontSize: 18,
fontWeight: FontWeight.bold),
                    ),
                    SizedBox(height: 4),
                    Text(
                      user?.email ?? "No email found",
                      style: TextStyle(color: Colors.white70),
                    ),
                  ],
                ),
              ],
            ),
            SizedBox(height: 20),
            _buildSettingItem(Icons.email, "Email", user?.email ?? "No email
found"),
            _buildSettingItem(Icons.phone, "Phone number", user?.phoneNumber ??
"Not linked"),
            _buildSettingItem(Icons.star, "Upgrade to Plus"),
            _buildSettingItem(Icons.tune, "Customize"),
            _buildSettingItem(Icons.storage, "Data Controls"),
            _buildSettingItem(Icons.mic, "Voice"),
            _buildSettingItem(Icons.info, "About"),
            Divider(color: Colors.grey),
            ListTile(
              leading: Icon(Icons.logout, color: Colors.red),
              title: Text("Sign out", style: TextStyle(color: Colors.red)),
              onTap: () async {
                await _auth.signOut();
                Navigator.popUntil(context, (route) => route.isFirst); // Go back to login
              },
            ),
          ],
        ),
      ),
    );
  }

  Widget _buildSettingItem(IconData icon, String title, [String? subtitle]) {
    return ListTile(
      leading: Icon(icon, color: Colors.white),
      title: Text(title, style: TextStyle(color: Colors.white)),
      subtitle: subtitle != null ? Text(subtitle, style: TextStyle(color: Colors.white70)) : null,
      onTap: () {},
    );
  }
}
```

//sidebar.dart
```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'setting.dart';
import 'home.dart';

class LeftSlider extends StatelessWidget {
 final String currentChatId; // Accepts currentChatId

 LeftSlider({required this.currentChatId});

 final FirebaseAuth _auth = FirebaseAuth.instance;
 final FirebaseFirestore _firestore = FirebaseFirestore.instance;

 void _startNewChat(BuildContext context) async {
   final user = _auth.currentUser;
   if (user == null) return;

   // Generate a new chat ID
   DocumentReference newChatRef = _firestore
       .collection('users')
       .doc(user.uid)
       .collection('chats')
       .doc();

   // Create an empty chat document
   await newChatRef.set({
     'createdAt': FieldValue.serverTimestamp(),
   });

   // Navigate to HomeScreen with the new chat ID
   Navigator.pushReplacement(
     context,
     MaterialPageRoute(
       builder: (context) => HomeScreen(chatId: newChatRef.id),
     ),
   );
 }

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

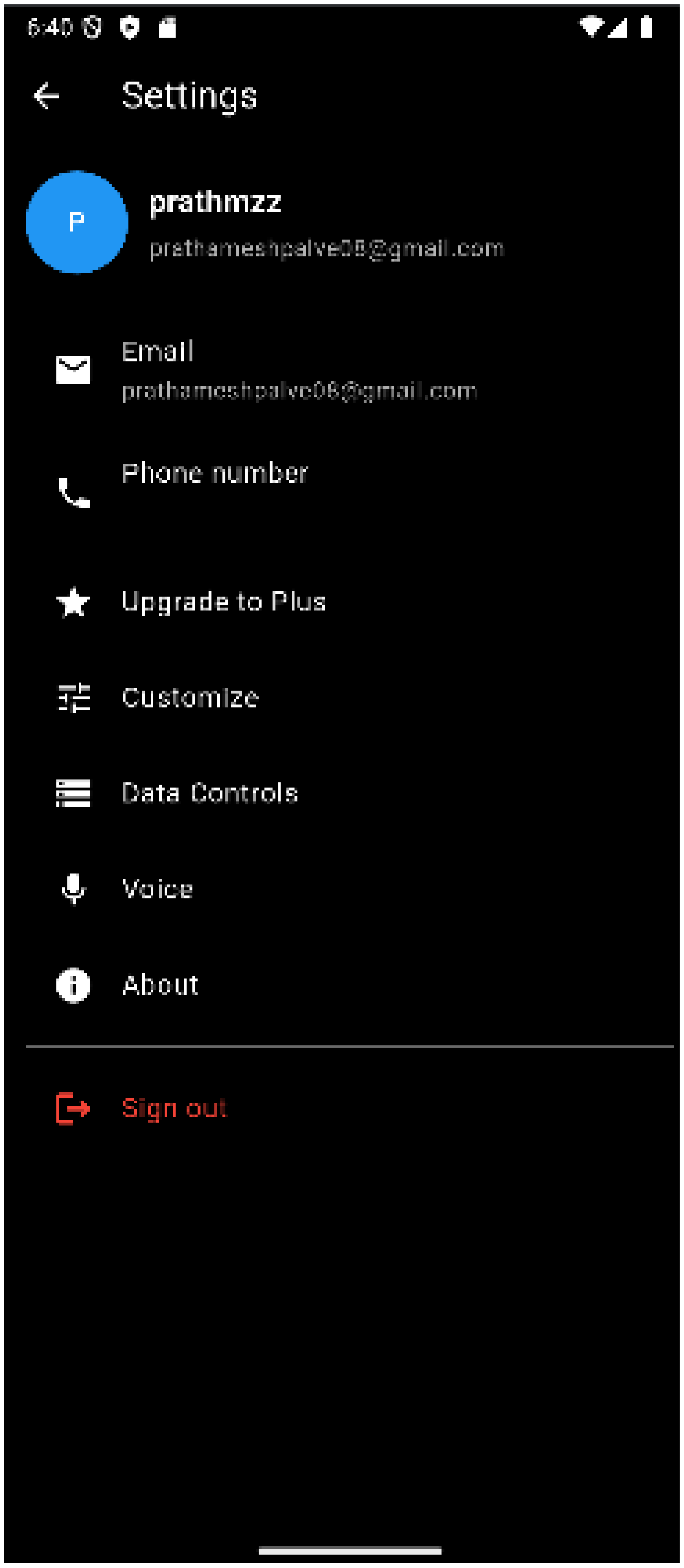
```
return Drawer(
  child: Container(
    color: Colors.black87,
    child: Column(
      children: [
        SizedBox(height: 40),
        // Search Bar
        Padding(
          padding: const EdgeInsets.symmetric(horizontal: 16),
          child: TextField(
            decoration: InputDecoration(
              hintText: "Search",
              filled: true,
              fillColor: Colors.grey[850],
              prefixIcon: Icon(Icons.search, color: Colors.white70),
              border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(8),
                borderSide: BorderSide.none,
              ),
            ),
          ),
        ),
        SizedBox(height: 20),
        // New Chat Button
        ListTile(
          leading: Icon(Icons.add, color: Colors.white),
          title: Text("New Chat", style: TextStyle(color: Colors.white)),
          onTap: () => _startNewChat(context),
        ),
        Divider(color: Colors.grey),
        Expanded(
          child: StreamBuilder<QuerySnapshot>(
            stream: _firestore
                .collection('users')
                .doc(_auth.currentUser?.uid)
                .collection('chats')
                .orderBy('createdAt', descending: true)
                .snapshots(),
            builder: (context, snapshot) {
              if (!snapshot.hasData || snapshot.data!.docs.isEmpty) {
                return Center(
                  child: Text(
                    "No chat history",
                    style: TextStyle(color: Colors.white70),
                     ),
                   );
                 }
                 return ListView(
                   children: snapshot.data!.docs.map((doc) {
                     String chatId = doc.id;
                     return ListTile(
                       title: Text(
                         "Chat ${chatId.substring(0, 8)}",
                         style: TextStyle(color: Colors.white70),
                       ),
                       selected: chatId == currentChatId, // Highlight active
chat

                       onTap: () {
                         Navigator.pushReplacement(
                           context,
                           MaterialPageRoute(
                             builder: (context) => HomeScreen(chatId: chatId),
                           ),
                         );
                       },
                     );
                   }).toList(),
                 );
               },
             ),
           ),
           Divider(color: Colors.grey),
           // Profile Section with Navigation to Settings
           ListTile(
             leading: CircleAvatar(
               backgroundColor: Colors.blue,
               child: Text(_auth.currentUser?.email?.substring(0, 1) ?? "U"),
             ),
             title: Text(
               _auth.currentUser?.email ?? "User",
               style: TextStyle(color: Colors.white),
             ),
             trailing: Icon(Icons.expand_more, color: Colors.white70),
             onTap: () {
               Navigator.push(
                 context,
                 MaterialPageRoute(builder: (context) => SettingsScreen()),
               );
             },
          ),
        ],
      ),
    ),
  );
 }
}
```



Output:

HomePage:

MyAccount:



6:40

Settings

P

prathmzz

prathameshpalve08@gmail.com

Email

prathameshpalve08@gmail.com

Phone number

Upgrade to Plus

Customize

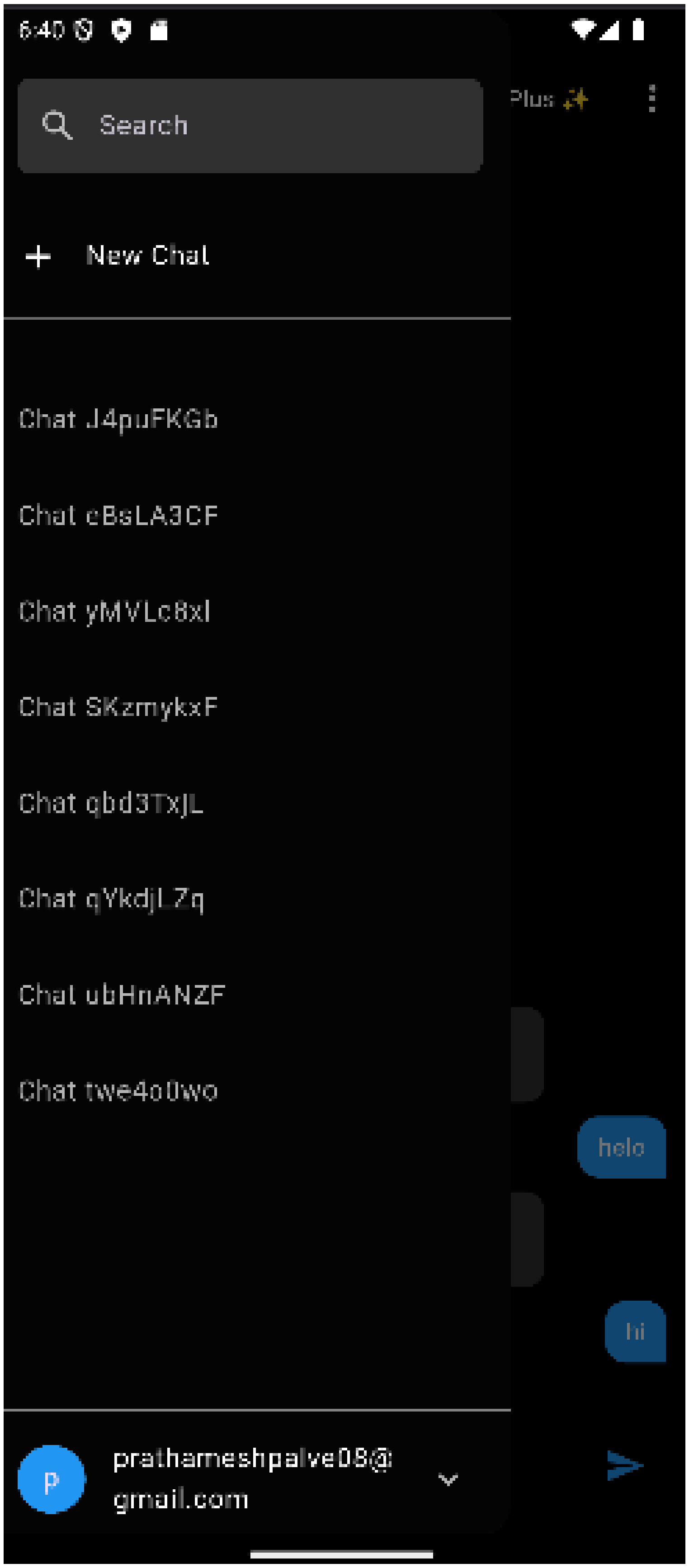
Data Controls

Voice

About

Sign out

Sidebar:



Chat J4puFKGb

Chat cBsLA3CF

Chat yMVLc8xl

Chat SKzmykxF

Chat qbd3TxJL

Chat qYkdjLZq

Chat ubHnANZF

Chat twe4o0wo

helo

hi