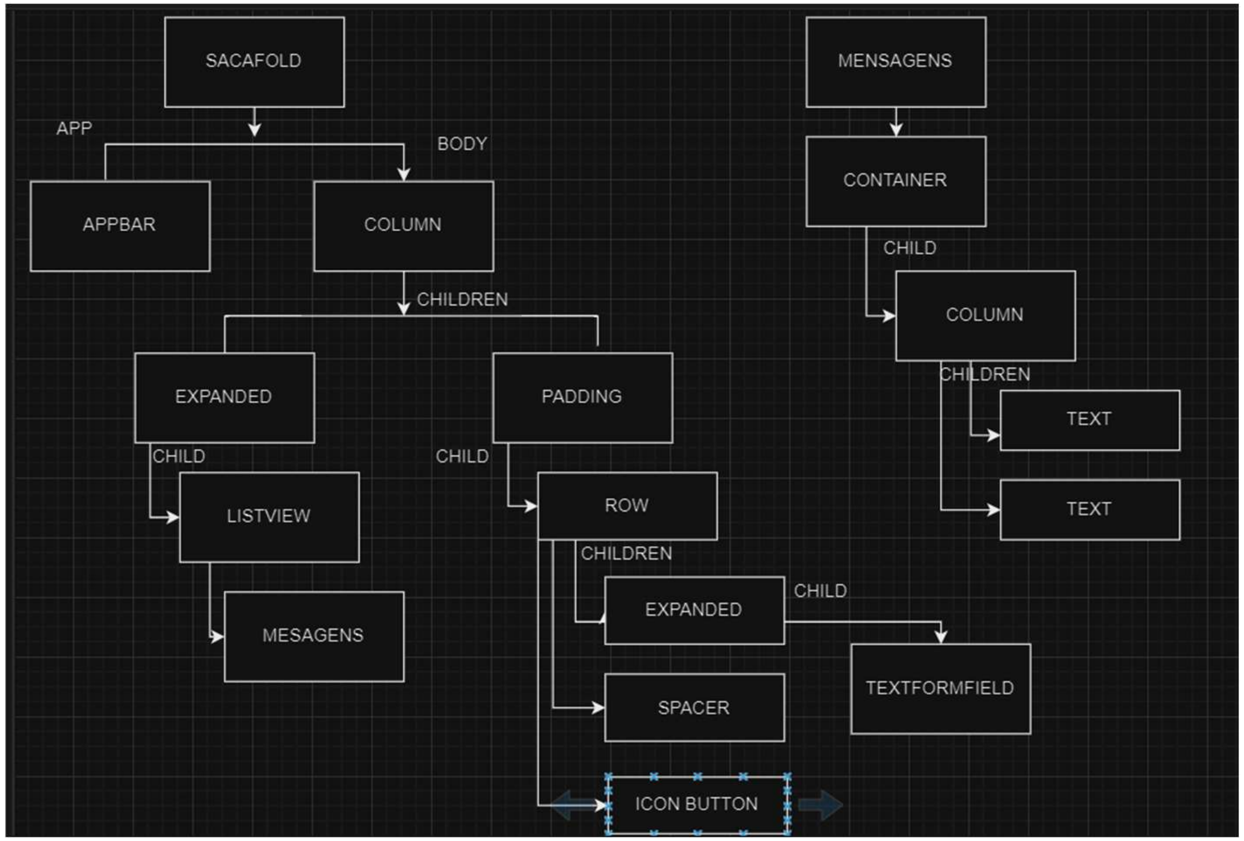
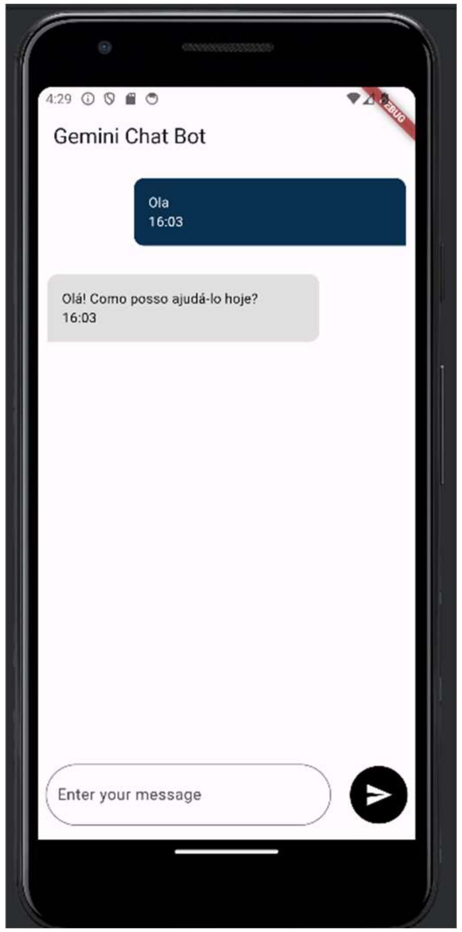
PROJETO Gemini Chat Bot

PROJETO Gemini Chat Bot

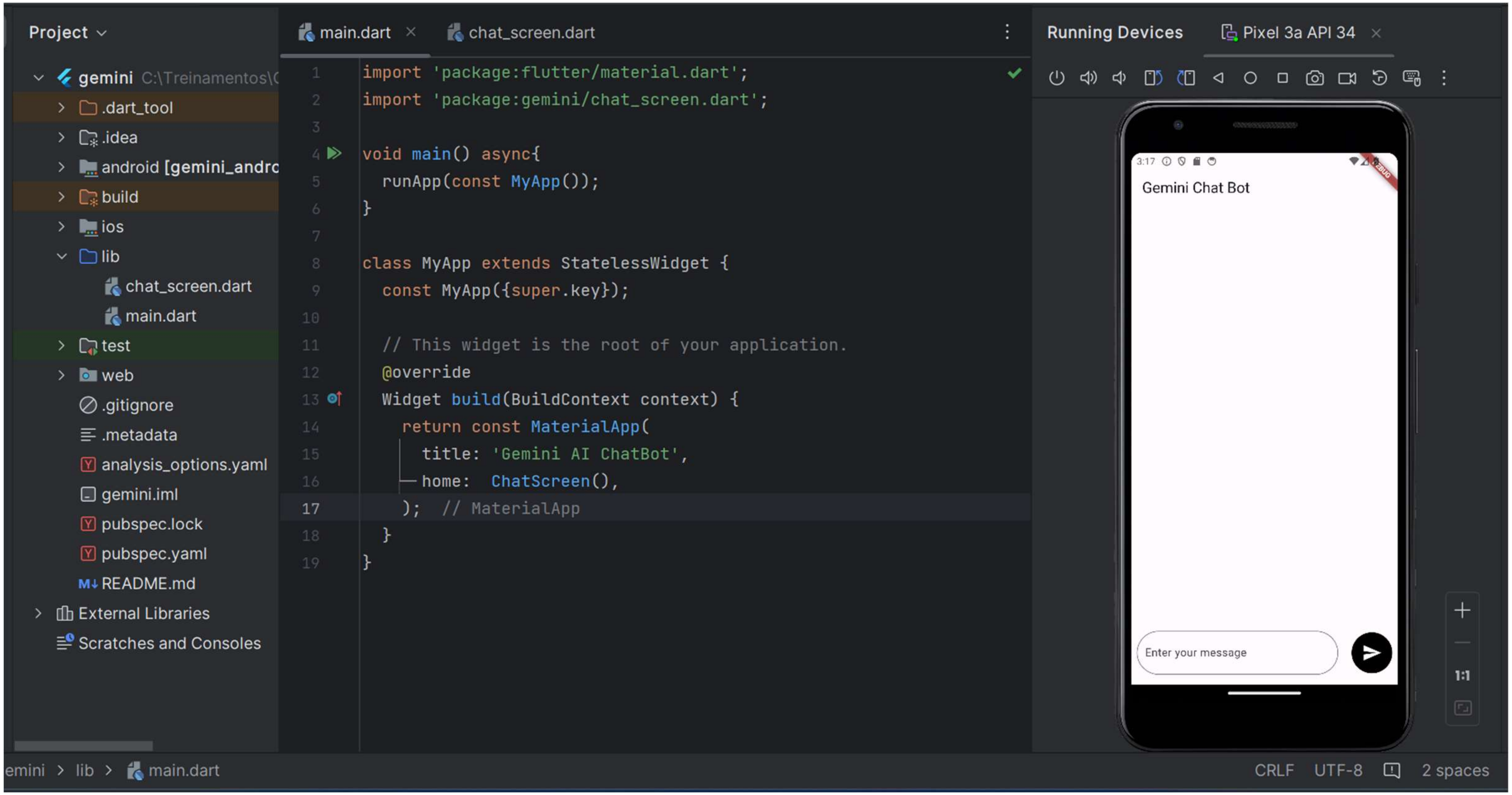




Ola
16:03

Olá! Como posso ajudá-lo hoje?
16:03

Main.dart

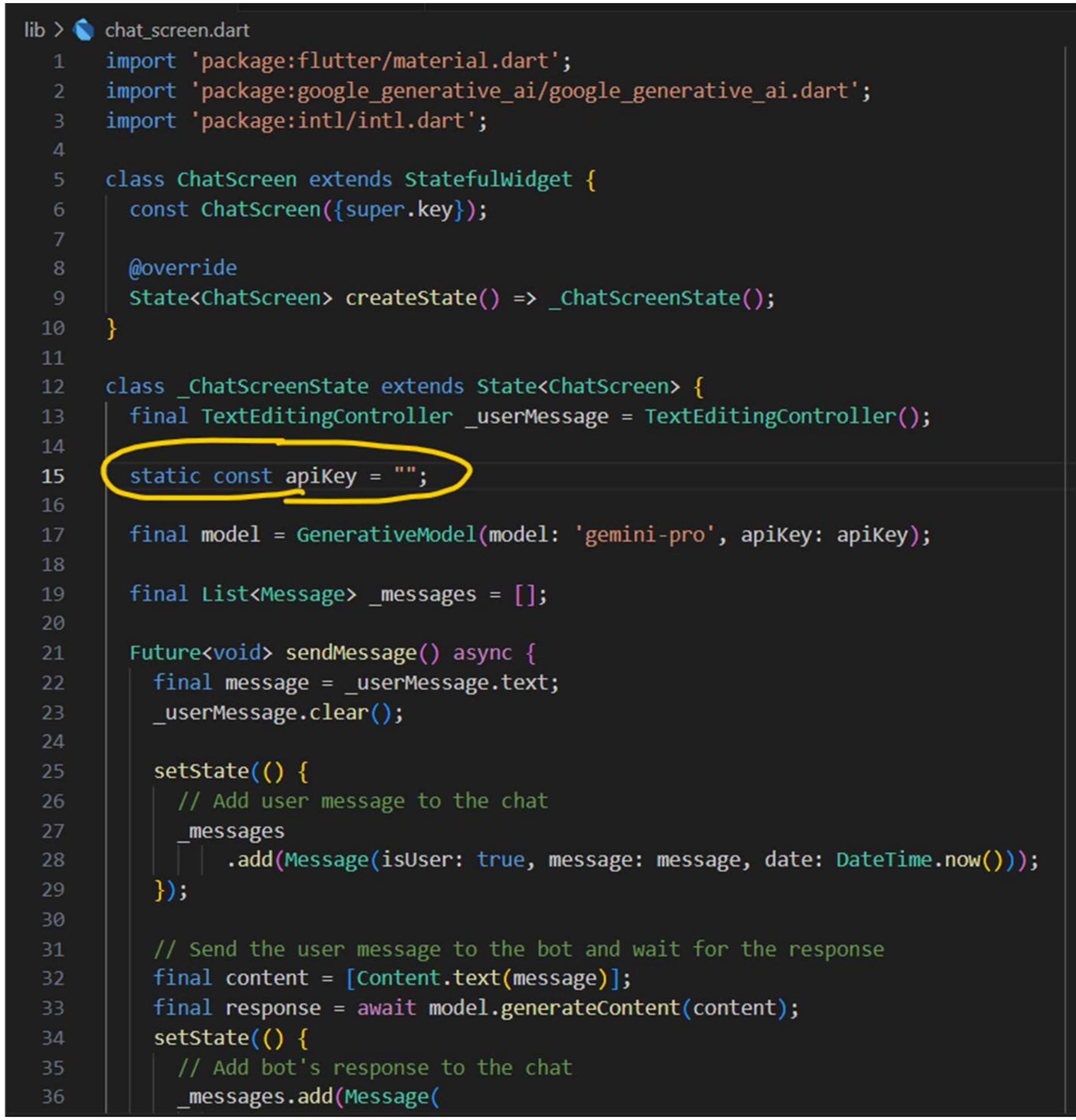


```dart
import 'package:flutter/material.dart';
import 'package:gemini/chat_screen.dart';

void main() async{
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      title: 'Gemini AI ChatBot',
      home:  ChatScreen(),
    );  // MaterialApp
  }
}
```

Chat_screen.dart

Como obter a API key

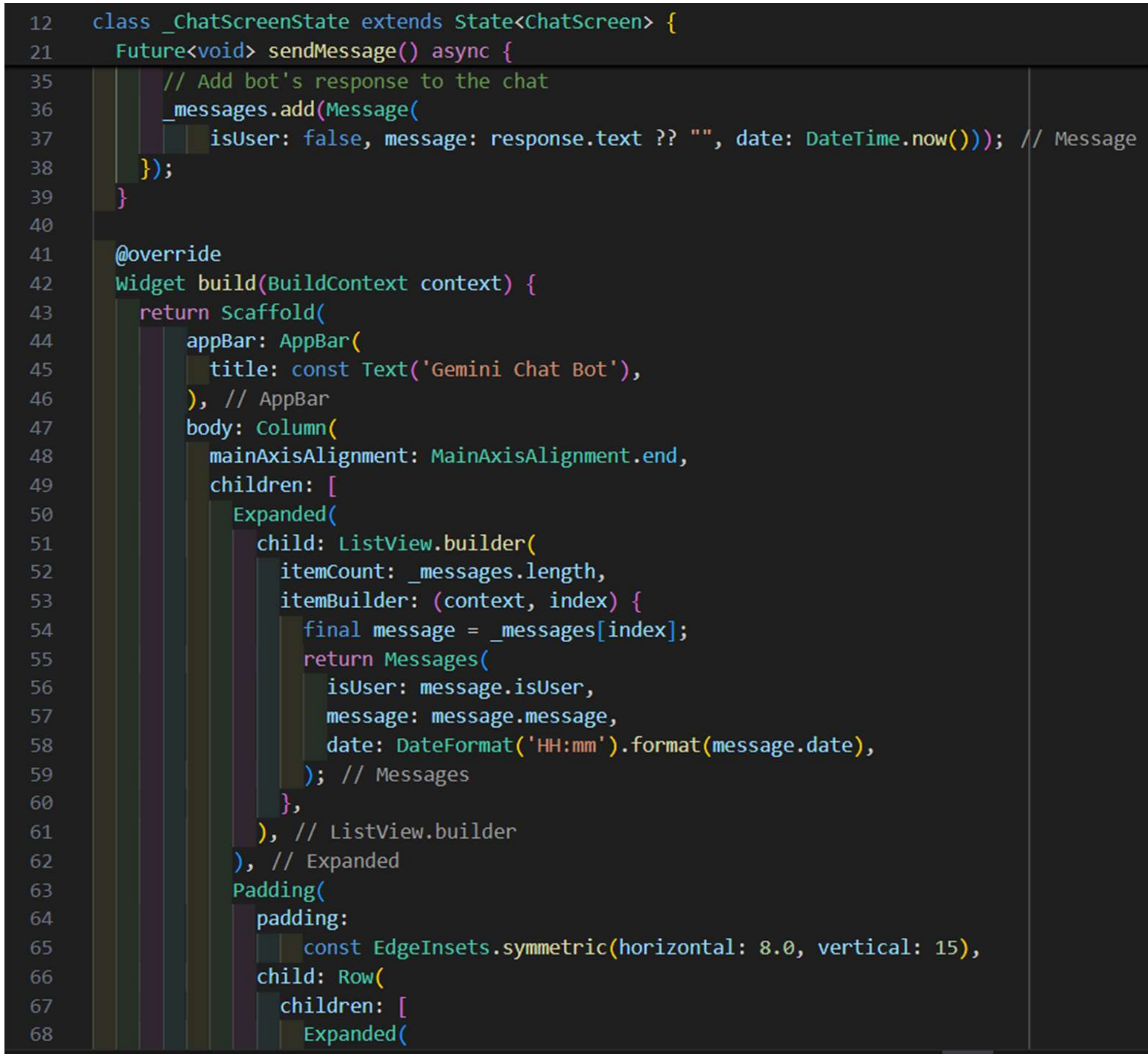
https://aistudio.google.com/app/apikey



lib > chat_screen.dart

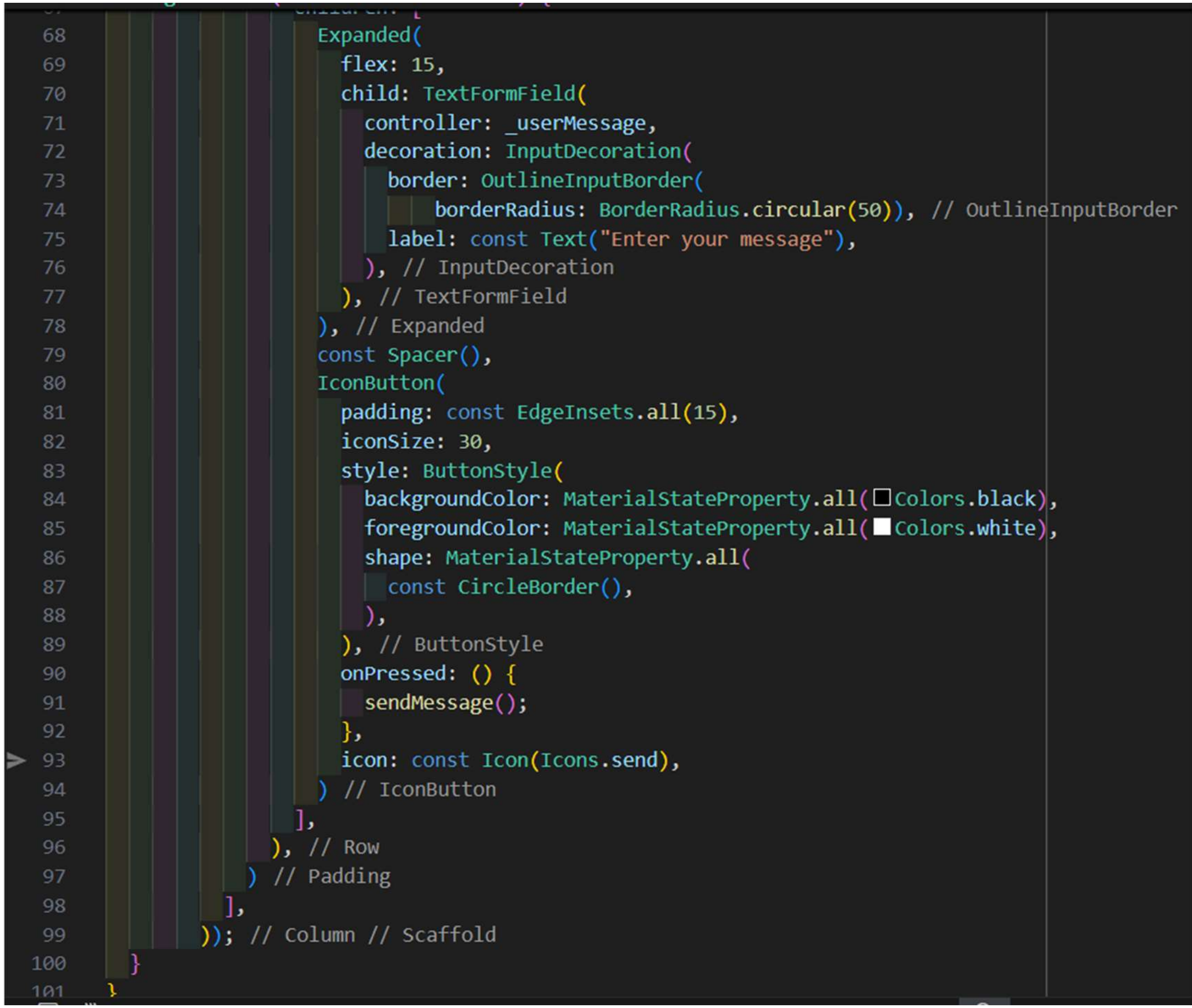
```dart
import 'package:flutter/material.dart';
import 'package:google_generative_ai/google_generative_ai.dart';
import 'package:intl/intl.dart';

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

  @override
  State<ChatScreen> createState() => _ChatScreenState();
}

class _ChatScreenState extends State<ChatScreen> {
  final TextEditingController _userMessage = TextEditingController();

  static const apiKey = "";

  final model = GenerativeModel(model: 'gemini-pro', apiKey: apiKey);

  final List<Message> _messages = [];

  Future<void> sendMessage() async {
    final message = _userMessage.text;
    _userMessage.clear();

    setState(() {
      // Add user message to the chat
      _messages
          .add(Message(isUser: true, message: message, date: DateTime.now()));
    });

    // Send the user message to the bot and wait for the response
    final content = [Content.text(message)];
    final response = await model.generateContent(content);
    setState(() {
      // Add bot's response to the chat
      _messages.add(Message(
```

Chat_screen.dart

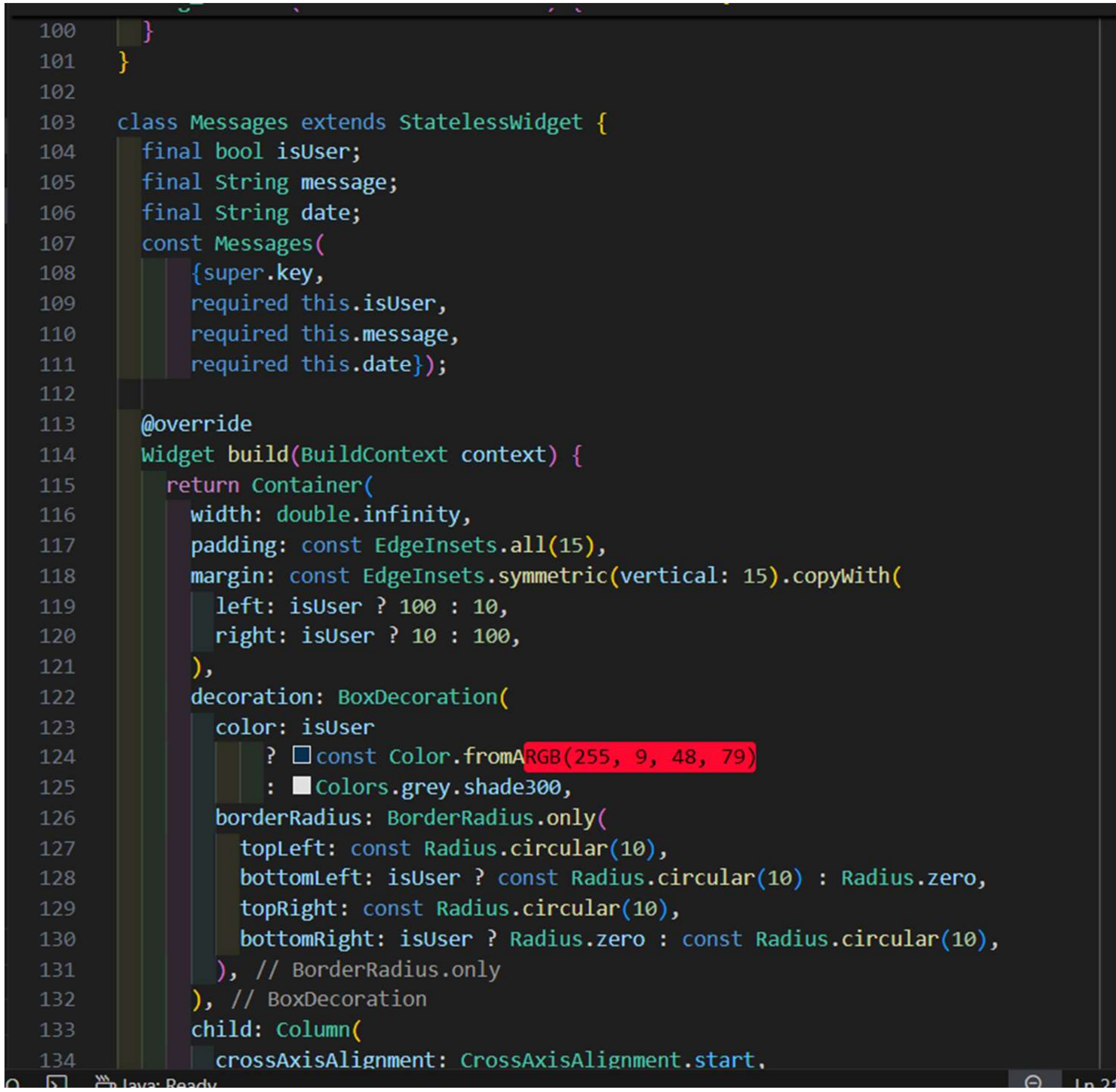


```dart
class _ChatScreenState extends State<ChatScreen> {
  Future<void> sendMessage() async {
      // Add bot's response to the chat
      _messages.add(Message(
          isUser: false, message: response.text ?? "", date: DateTime.now())); // Message
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: const Text('Gemini Chat Bot'),
        ), // AppBar
        body: Column(
          mainAxisAlignment: MainAxisAlignment.end,
          children: [
            Expanded(
              child: ListView.builder(
                itemCount: _messages.length,
                itemBuilder: (context, index) {
                  final message = _messages[index];
                  return Messages(
                    isUser: message.isUser,
                    message: message.message,
                    date: DateFormat('HH:mm').format(message.date),
                  ); // Messages
                },
              ), // ListView.builder
            ), // Expanded
            Padding(
              padding:
                  const EdgeInsets.symmetric(horizontal: 8.0, vertical: 15),
              child: Row(
                children: [
                  Expanded(
```

Chat_screen.dart

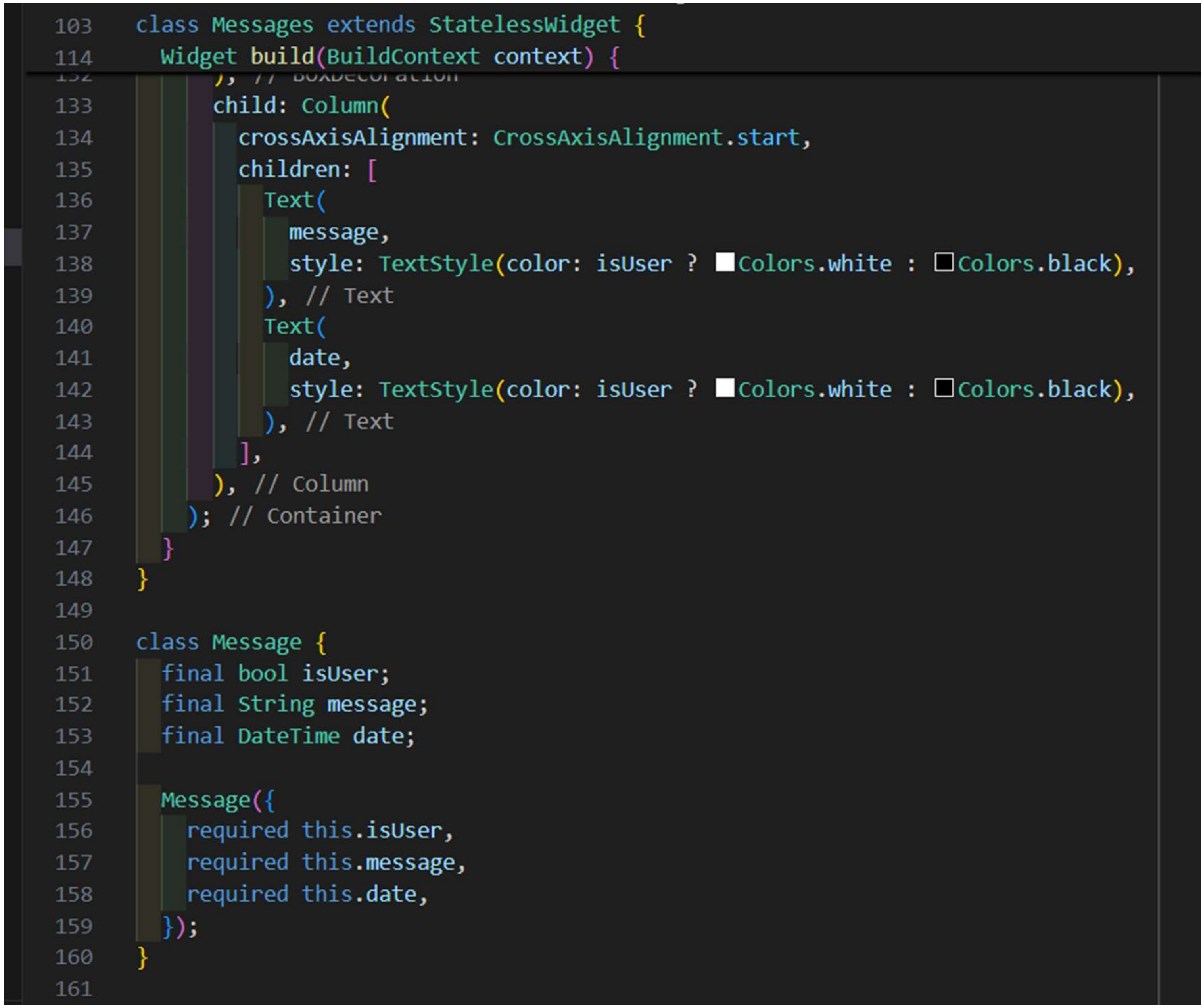

```
                Expanded(
                  flex: 15,
                  child: TextFormField(
                    controller: _userMessage,
                    decoration: InputDecoration(
                      border: OutlineInputBorder(
                          borderRadius: BorderRadius.circular(50)), // OutlineInputBorder
                      label: const Text("Enter your message"),
                    ), // InputDecoration
                  ), // TextFormField
                ), // Expanded
                const Spacer(),
                IconButton(
                  padding: const EdgeInsets.all(15),
                  iconSize: 30,
                  style: ButtonStyle(
                    backgroundColor: MaterialStateProperty.all(■Colors.black),
                    foregroundColor: MaterialStateProperty.all(□Colors.white),
                    shape: MaterialStateProperty.all(
                      const CircleBorder(),
                    ),
                  ), // ButtonStyle
                  onPressed: () {
                    sendMessage();
                  },
                  icon: const Icon(Icons.send),
                ) // IconButton
              ],
            ), // Row
          ) // Padding
        ],
      )); // Column // Scaffold
    }
}
```

Chat_screen.dart



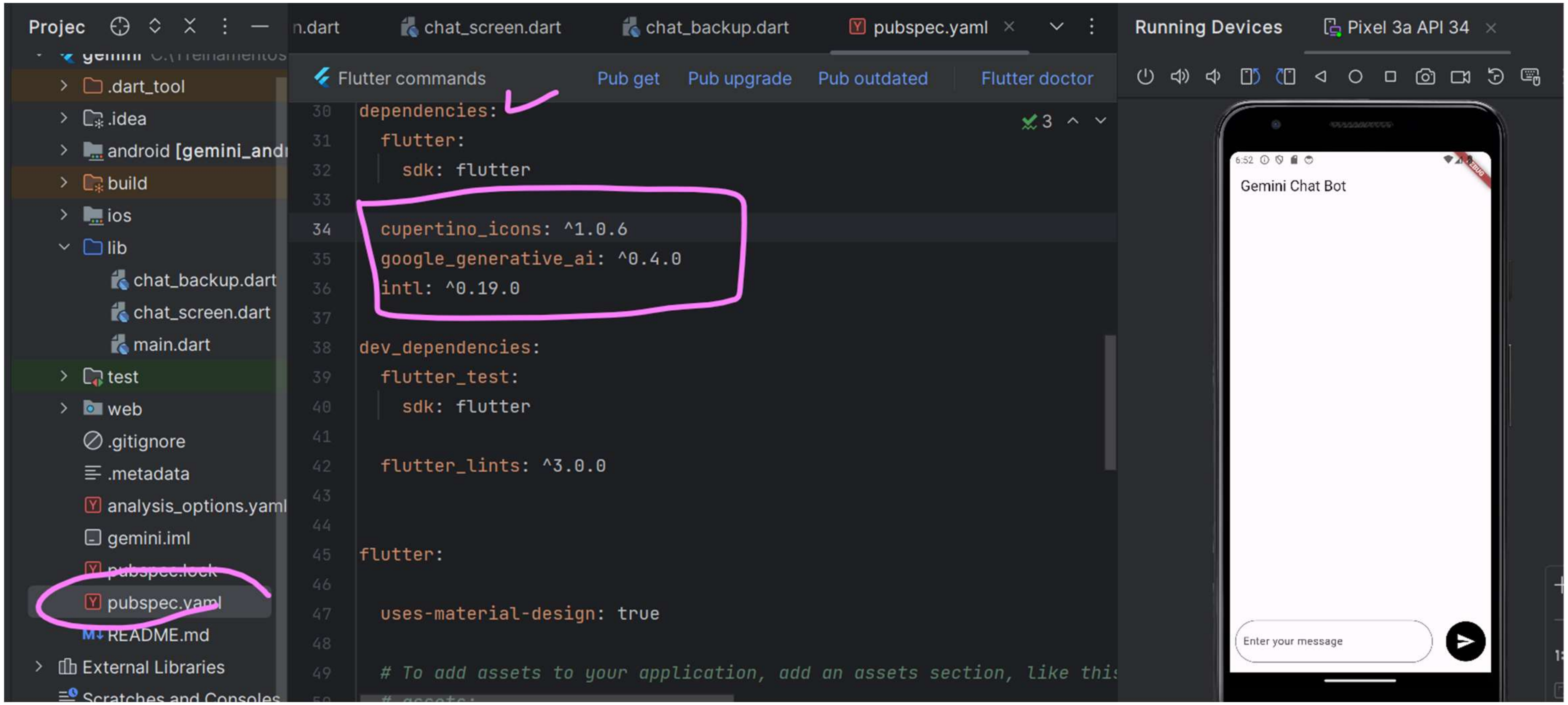
```dart
  }
}

class Messages extends StatelessWidget {
  final bool isUser;
  final String message;
  final String date;
  const Messages(
      {super.key,
      required this.isUser,
      required this.message,
      required this.date});

  @override
  Widget build(BuildContext context) {
    return Container(
      width: double.infinity,
      padding: const EdgeInsets.all(15),
      margin: const EdgeInsets.symmetric(vertical: 15).copyWith(
        left: isUser ? 100 : 10,
        right: isUser ? 10 : 100,
      ),
      decoration: BoxDecoration(
        color: isUser
            ? const Color.fromARGB(255, 9, 48, 79)
            : Colors.grey.shade300,
        borderRadius: BorderRadius.only(
          topLeft: const Radius.circular(10),
          bottomLeft: isUser ? const Radius.circular(10) : Radius.zero,
          topRight: const Radius.circular(10),
          bottomRight: isUser ? Radius.zero : const Radius.circular(10),
        ), // BorderRadius.only
      ), // BoxDecoration
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
```

Chat_screen.dart

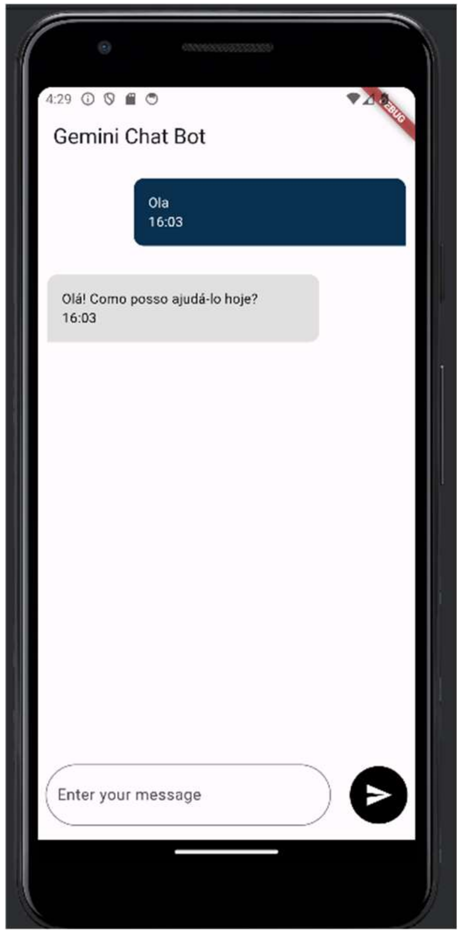


```dart
class Messages extends StatelessWidget {
  Widget build(BuildContext context) {
      ), // BoxDecoration
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            message,
            style: TextStyle(color: isUser ? Colors.white : Colors.black),
          ), // Text
          Text(
            date,
            style: TextStyle(color: isUser ? Colors.white : Colors.black),
          ), // Text
        ],
      ), // Column
    ); // Container
  }
}

class Message {
  final bool isUser;
  final String message;
  final DateTime date;

  Message({
    required this.isUser,
    required this.message,
    required this.date,
  });
}
```

Adicionar as dependências da imagem



```
dependencies:
  flutter:
    sdk: flutter

  cupertino_icons: ^1.0.6
  google_generative_ai: ^0.4.0
  intl: ^0.19.0

dev_dependencies:
  flutter_test:
    sdk: flutter

  flutter_lints: ^3.0.0


flutter:

  uses-material-design: true

  # To add assets to your application, add an assets section, like this
```



Gemini Chat Bot

Ola
16:03

Olá! Como posso ajudá-lo hoje?
16:03