****// Main application file: main.dart****

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

****import 'package:uuid/uuid.dart';****

****void main() {****

****runApp(const ChatApp());****

****}****

****class ChatApp extends StatelessWidget {****

****const ChatApp({Key? key}) : super(key: key);****

****@override****

****Widget build(BuildContext context) {****

****return MaterialApp(****

****title: 'Flutter Chat App',****

****theme: ThemeData(****

****primarySwatch: Colors.blue,****

****visualDensity: VisualDensity.adaptivePlatformDensity,****

****),****

****home: const LoginScreen(),****

****);****

****}****

****}****

****// Models****

****class Message {****

****final String id;****

****final String text;****

****final String senderId;****

****final String senderName;****

****final DateTime timestamp;****

****Message({****

****required this.text,****

****required this.senderId,****

****required this.senderName,****

****required this.timestamp,****

****}) : id = const Uuid().v4();****

****}****

****class ChatUser {****

****final String id;****

****final String name;****

****final String? profileImageUrl;****

****ChatUser({****

****required this.id,****

****required this.name,****

****this.profileImageUrl,****

****});****

****}****

****// Login Screen****

****class LoginScreen extends StatefulWidget {****

****const LoginScreen({Key? key}) : super(key: key);****

****@override****

****\_LoginScreenState createState() => \_LoginScreenState();****

****}****

****class \_LoginScreenState extends State<LoginScreen> {****

****final TextEditingController \_nameController = TextEditingController();****

****@override****

****Widget build(BuildContext context) {****

****return Scaffold(****

****appBar: AppBar(****

****title: const Text('Login to Chat'),****

****),****

****body: Padding(****

****padding: const EdgeInsets.all(16.0),****

****child: Column(****

****mainAxisAlignment: MainAxisAlignment.center,****

****children: [****

****TextField(****

****controller: \_nameController,****

****decoration: const InputDecoration(****

****labelText: 'Enter your name',****

****border: OutlineInputBorder(),****

****),****

****),****

****const SizedBox(height: 20),****

****ElevatedButton(****

****onPressed: () {****

****if (\_nameController.text.trim().isNotEmpty) {****

****final user = ChatUser(****

****id: const Uuid().v4(),****

****name: \_nameController.text.trim(),****

****);****

****Navigator.of(context).pushReplacement(****

****MaterialPageRoute(****

****builder: (context) => ChatListScreen(currentUser: user),****

****),****

****);****

****}****

****},****

****child: const Text('Enter Chat'),****

****),****

****],****

****),****

****),****

****);****

****}****

****@override****

****void dispose() {****

****\_nameController.dispose();****

****super.dispose();****

****}****

****}****

****// Chat List Screen****

****class ChatListScreen extends StatelessWidget {****

****final ChatUser currentUser;****

****const ChatListScreen({Key? key, required this.currentUser}) : super(key: key);****

****@override****

****Widget build(BuildContext context) {****

****// Demo chat rooms****

****final List<Map<String, dynamic>> chatRooms = [****

****{'name': 'General Chat', 'id': '1'},****

****{'name': 'Flutter Developers', 'id': '2'},****

****{'name': 'Dart Discussion', 'id': '3'},****

****];****

****return Scaffold(****

****appBar: AppBar(****

****title: const Text('Chat Rooms'),****

****actions: [****

****Padding(****

****padding: const EdgeInsets.all(8.0),****

****child: CircleAvatar(****

****child: Text(currentUser.name[0].toUpperCase()),****

****),****

****),****

****],****

****),****

****body: ListView.builder(****

****itemCount: chatRooms.length,****

****itemBuilder: (context, index) {****

****return ListTile(****

****leading: const CircleAvatar(****

****child: Icon(Icons.chat),****

****),****

****title: Text(chatRooms[index]['name']),****

****subtitle: const Text('Tap to join this room'),****

****onTap: () {****

****Navigator.of(context).push(****

****MaterialPageRoute(****

****builder: (context) => ChatRoomScreen(****

****roomId: chatRooms[index]['id'],****

****roomName: chatRooms[index]['name'],****

****currentUser: currentUser,****

****),****

****),****

****);****

****},****

****);****

****},****

****),****

****floatingActionButton: FloatingActionButton(****

****onPressed: () {****

****// Show dialog to create new chat room (simplified for demo)****

****showDialog(****

****context: context,****

****builder: (context) => AlertDialog(****

****title: const Text('New chat room feature coming soon!'),****

****actions: [****

****TextButton(****

****onPressed: () => Navigator.pop(context),****

****child: const Text('OK'),****

****),****

****],****

****),****

****);****

****},****

****child: const Icon(Icons.add),****

****),****

****);****

****}****

****}****

****// Chat Room Screen****

****class ChatRoomScreen extends StatefulWidget {****

****final String roomId;****

****final String roomName;****

****final ChatUser currentUser;****

****const ChatRoomScreen({****

****Key? key,****

****required this.roomId,****

****required this.roomName,****

****required this.currentUser,****

****}) : super(key: key);****

****@override****

****\_ChatRoomScreenState createState() => \_ChatRoomScreenState();****

****}****

****class \_ChatRoomScreenState extends State<ChatRoomScreen> {****

****final TextEditingController \_messageController = TextEditingController();****

****final List<Message> \_messages = [];****

****final ScrollController \_scrollController = ScrollController();****

****@override****

****void initState() {****

****super.initState();****

****// Simulate loading messages****

****\_loadMessages();****

****}****

****void \_loadMessages() {****

****// In a real app, you would fetch messages from a database or API****

****// For this demo, we'll add some sample messages****

****final demoMessages = [****

****Message(****

****text: 'Welcome to ${widget.roomName}!',****

****senderId: 'system',****

****senderName: 'System',****

****timestamp: DateTime.now().subtract(const Duration(minutes: 15)),****

****),****

****Message(****

****text: 'How is everyone doing today?',****

****senderId: 'user1',****

****senderName: 'Alice',****

****timestamp: DateTime.now().subtract(const Duration(minutes: 10)),****

****),****

****Message(****

****text: 'I\'m doing great! Working on a Flutter project.',****

****senderId: 'user2',****

****senderName: 'Bob',****

****timestamp: DateTime.now().subtract(const Duration(minutes: 5)),****

****),****

****];****

****setState(() {****

****\_messages.addAll(demoMessages);****

****});****

****}****

****void \_sendMessage() {****

****if (\_messageController.text.trim().isNotEmpty) {****

****final newMessage = Message(****

****text: \_messageController.text.trim(),****

****senderId: widget.currentUser.id,****

****senderName: widget.currentUser.name,****

****timestamp: DateTime.now(),****

****);****

****setState(() {****

****\_messages.add(newMessage);****

****\_messageController.clear();****

****});****

****// Scroll to the bottom after sending a message****

****Future.delayed(const Duration(milliseconds: 100), () {****

****if (\_scrollController.hasClients) {****

****\_scrollController.animateTo(****

****\_scrollController.position.maxScrollExtent,****

****duration: const Duration(milliseconds: 300),****

****curve: Curves.easeOut,****

****);****

****}****

****});****

****}****

****}****

****@override****

****Widget build(BuildContext context) {****

****return Scaffold(****

****appBar: AppBar(****

****title: Text(widget.roomName),****

****actions: [****

****IconButton(****

****icon: const Icon(Icons.info\_outline),****

****onPressed: () {****

****// Show room info****

****showDialog(****

****context: context,****

****builder: (context) => AlertDialog(****

****title: Text(widget.roomName),****

****content: const Text('Room information and members would be displayed here.'),****

****actions: [****

****TextButton(****

****onPressed: () => Navigator.pop(context),****

****child: const Text('Close'),****

****),****

****],****

****),****

****);****

****},****

****),****

****],****

****),****

****body: Column(****

****children: [****

****// Messages list****

****Expanded(****

****child: \_messages.isEmpty****

****? const Center(child: Text('No messages yet'))****

****: ListView.builder(****

****controller: \_scrollController,****

****padding: const EdgeInsets.all(8.0),****

****itemCount: \_messages.length,****

****itemBuilder: (context, index) {****

****final message = \_messages[index];****

****final isMe = message.senderId == widget.currentUser.id;****

****return MessageBubble(****

****message: message,****

****isMe: isMe,****

****);****

****},****

****),****

****),****

****// Message input****

****Container(****

****padding: const EdgeInsets.all(8.0),****

****decoration: BoxDecoration(****

****color: Theme.of(context).cardColor,****

****boxShadow: [****

****BoxShadow(****

****color: Colors.grey.withOpacity(0.2),****

****spreadRadius: 1,****

****blurRadius: 3,****

****offset: const Offset(0, -1),****

****),****

****],****

****),****

****child: Row(****

****children: [****

****IconButton(****

****icon: const Icon(Icons.attach\_file),****

****onPressed: () {****

****// Attachment feature would go here****

****ScaffoldMessenger.of(context).showSnackBar(****

****const SnackBar(content: Text('Attachment feature coming soon!')),****

****);****

****},****

****),****

****Expanded(****

****child: TextField(****

****controller: \_messageController,****

****decoration: const InputDecoration(****

****hintText: 'Type a message',****

****border: InputBorder.none,****

****),****

****textCapitalization: TextCapitalization.sentences,****

****onSubmitted: (\_) => \_sendMessage(),****

****),****

****),****

****IconButton(****

****icon: const Icon(Icons.send),****

****color: Theme.of(context).primaryColor,****

****onPressed: \_sendMessage,****

****),****

****],****

****),****

****),****

****],****

****),****

****);****

****}****

****@override****

****void dispose() {****

****\_messageController.dispose();****

****\_scrollController.dispose();****

****super.dispose();****

****}****

****}****

****// Message Bubble Component****

****class MessageBubble extends StatelessWidget {****

****final Message message;****

****final bool isMe;****

****const MessageBubble({****

****Key? key,****

****required this.message,****

****required this.isMe,****

****}) : super(key: key);****

****@override****

****Widget build(BuildContext context) {****

****return Padding(****

****padding: const EdgeInsets.symmetric(vertical: 5.0, horizontal: 8.0),****

****child: Column(****

****crossAxisAlignment: isMe ? CrossAxisAlignment.end : CrossAxisAlignment.start,****

****children: [****

****if (!isMe)****

****Padding(****

****padding: const EdgeInsets.only(left: 12.0, bottom: 4.0),****

****child: Text(****

****message.senderName,****

****style: TextStyle(****

****fontSize: 12.0,****

****color: Colors.grey[600],****

****),****

****),****

****),****

****Container(****

****padding: const EdgeInsets.symmetric(vertical: 10.0, horizontal: 14.0),****

****decoration: BoxDecoration(****

****color: isMe ? Theme.of(context).primaryColor : Colors.grey[300],****

****borderRadius: BorderRadius.circular(18.0),****

****),****

****child: Text(****

****message.text,****

****style: TextStyle(****

****color: isMe ? Colors.white : Colors.black87,****

****),****

****),****

****),****

****Padding(****

****padding: const EdgeInsets.only(top: 4.0, right: 8.0, left: 8.0),****

****child: Text(****

****'${message.timestamp.hour}:${message.timestamp.minute.toString().padLeft(2, '0')}',****

****style: TextStyle(****

****fontSize: 10.0,****

****color: Colors.grey[600],****

****),****

****),****

****),****

****],****

****),****

****);****

****}****

****}****

****// Add this to your pubspec.yaml:****

****// dependencies:****

****// flutter:****

****// sdk: flutter****

****// uuid: ^3.0.6****