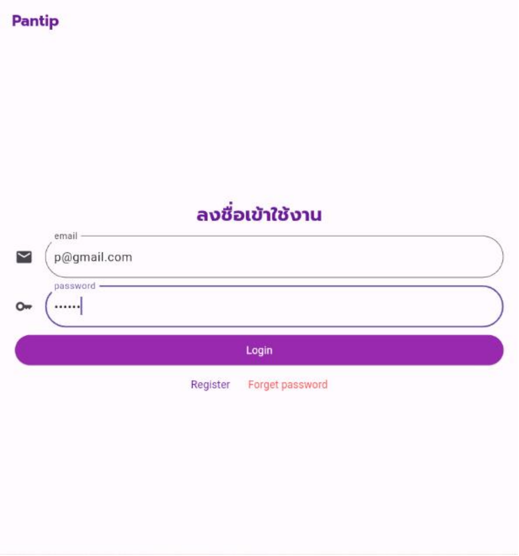
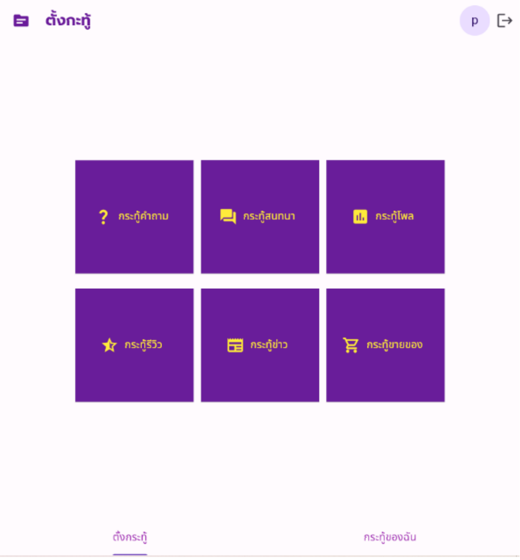
Miniproject 2 Pantip



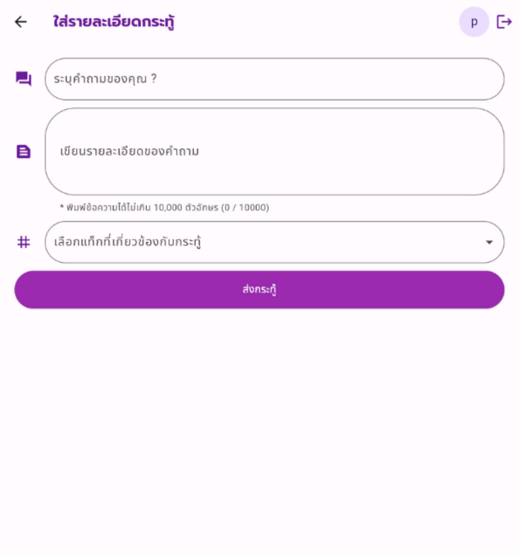
email: p@gmail.com

password: ••••••

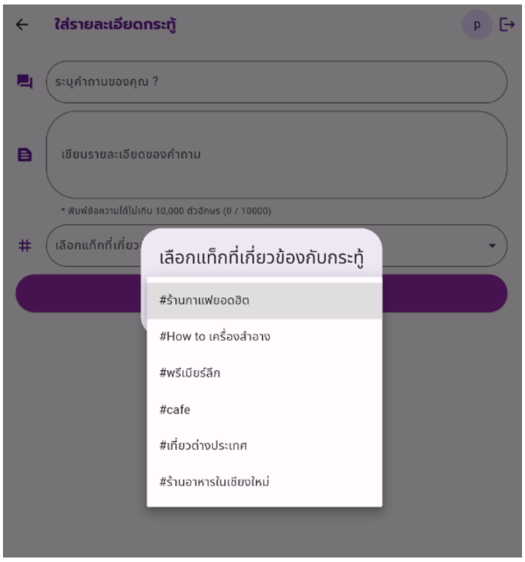


หน้า Login, Regiter, Forget password

หน้าเลือกแบบการสร้างกระทู้

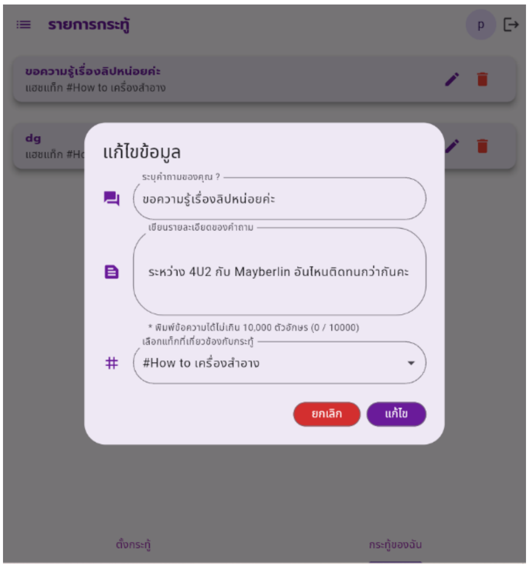


* พิมพ์ข้อความได้ไม่เกิน 10,000 ตัวอักษร (0 / 10000)



หน้ากรอกข้อมูลสร้างกระทู้

ตัวอย่างการเลือกแฮชแท็ก



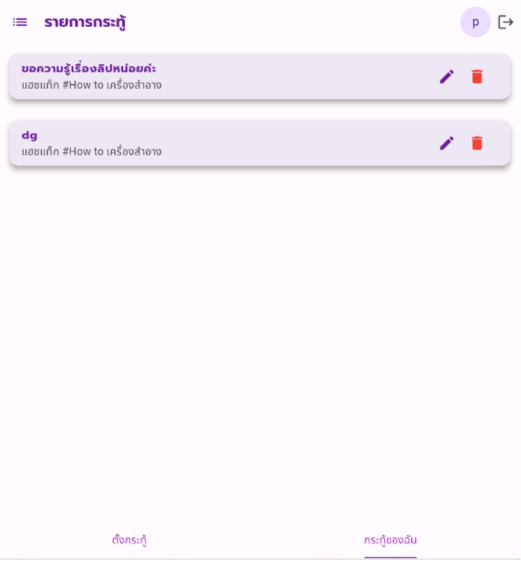
ระบุคำถามของคุณ ?: ขอความรู้เรื่องลิปหน่อยค่ะ

เขียนรายละเอียดของคำถาม: ระหว่าง 4U2 กับ Mayberlin อันไหนติดทนกว่ากันคะ

* พิมพ์ข้อความได้ไม่เกิน 10,000 ตัวอักษร (0 / 10000)

เลือกแท็กที่เกี่ยวข้องกับกระทู้: #How to เครื่องสำอาง

หน้าแก้ไขกระทู้


รายการกระทู้
p
ขอความรู้เรื่องลิปหน่อยค่ะ
แฮชแท็ก #How to เครื่องสำอาง
dg
แฮชแท็ก #How to เครื่องสำอาง
ตั้งกระทู้
กระทู้ของฉัน

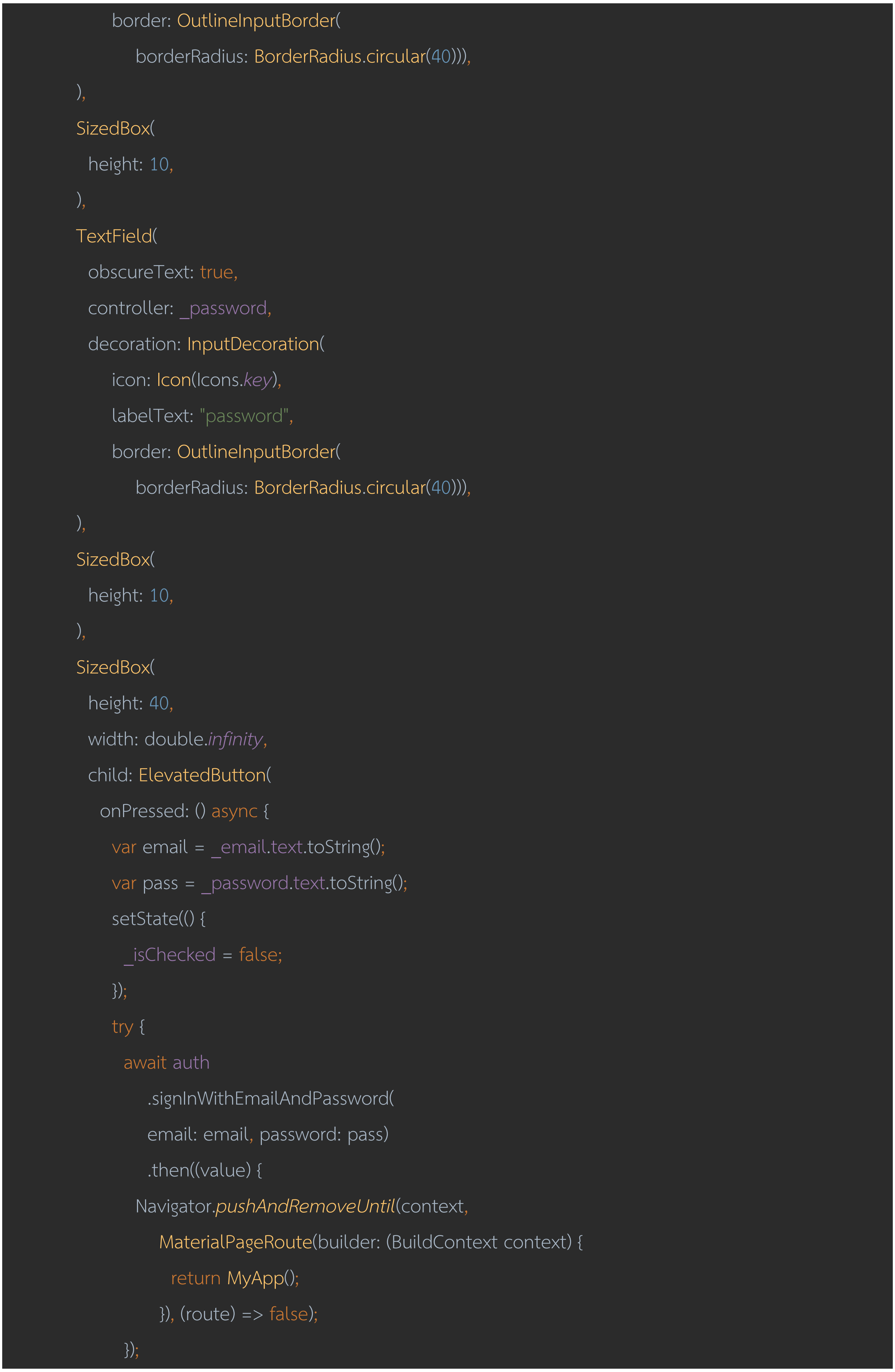
หน้ารายการกระทู้ที่สร้าง

หน้า main.dart

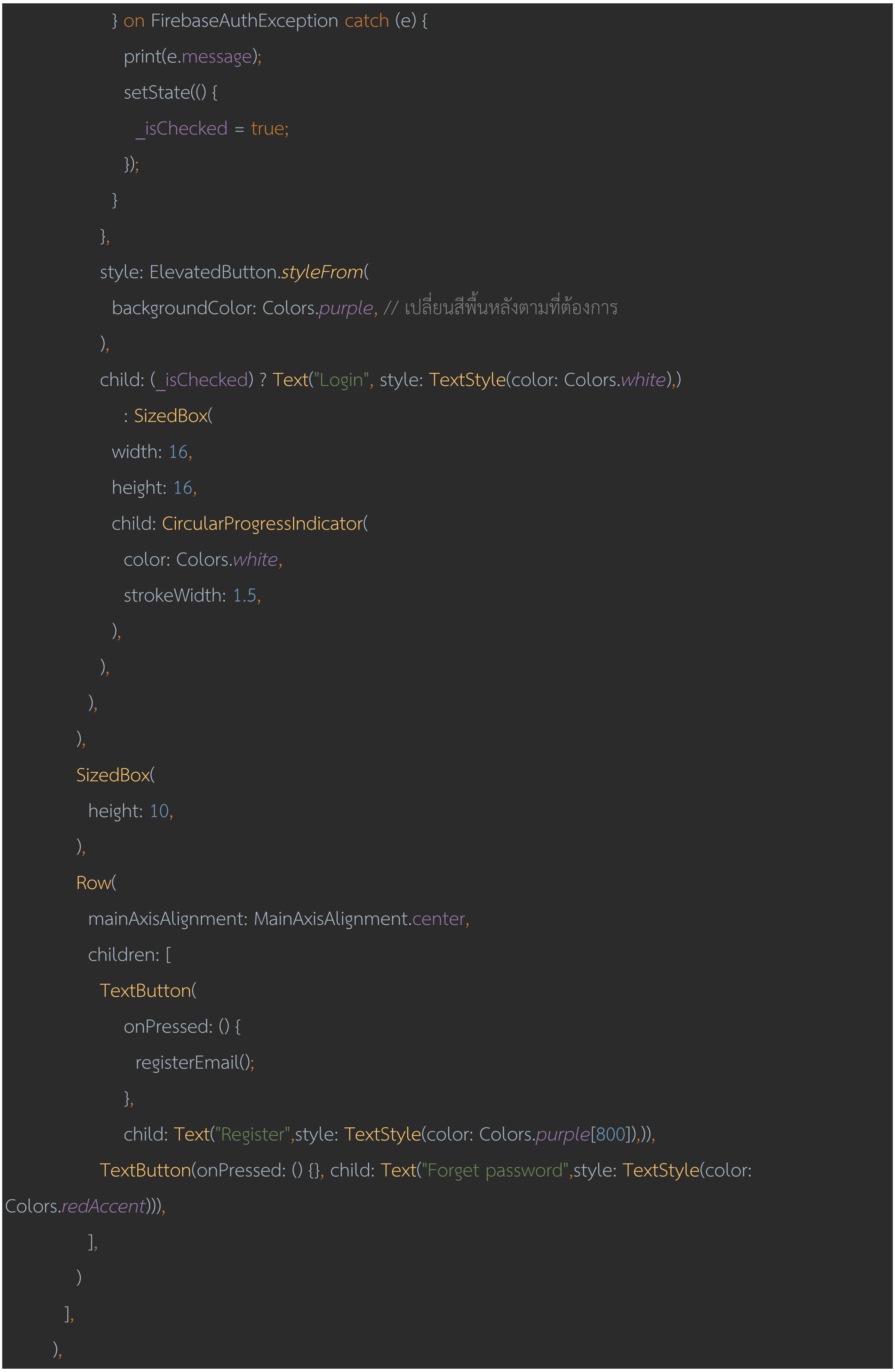
```dart
import 'package:chapter4_cloud_firestore/page/list_topic.dart';
import 'package:chapter4_cloud_firestore/page/topic.dart';
import 'package:chapter4_cloud_firestore/page/sign_in_screen.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
      options: FirebaseOptions(
          apiKey: "AIzaSyAKT5f30ZlZMIdUErn_uaxGaZVV7QNZ1sI",
          appId: "1:994850625306:web:6fb130818d30618530fea3",
          messagingSenderId: "994850625306",
          projectId: "flutter-firebase-cloud-2fbac"));
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false,
    home: SignInScreen(),
  ));
}

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

  @override
  Widget build(BuildContext context) {
    return DefaultTabController(
        length: 2,
        child: Scaffold(
          bottomNavigationBar: TabBar(
            indicatorColor: Colors.deepPurple,
            labelColor: Colors.deepPurple,
            tabs: [
              Tab(
                child: Text("ตั้งกระทู้",style: TextStyle(color: Colors.purple)),
              ),
              Tab(
                child: Text("กระทู้ของฉัน",style: TextStyle(color: Colors.purple)),
              ),
            ],
          ),
          body: TabBarView(
            children: [
              Topic(),
              ListTopic()
            ],
          ),
        ),
    );
  }
}
```

หน้า sign_in_screen.dart

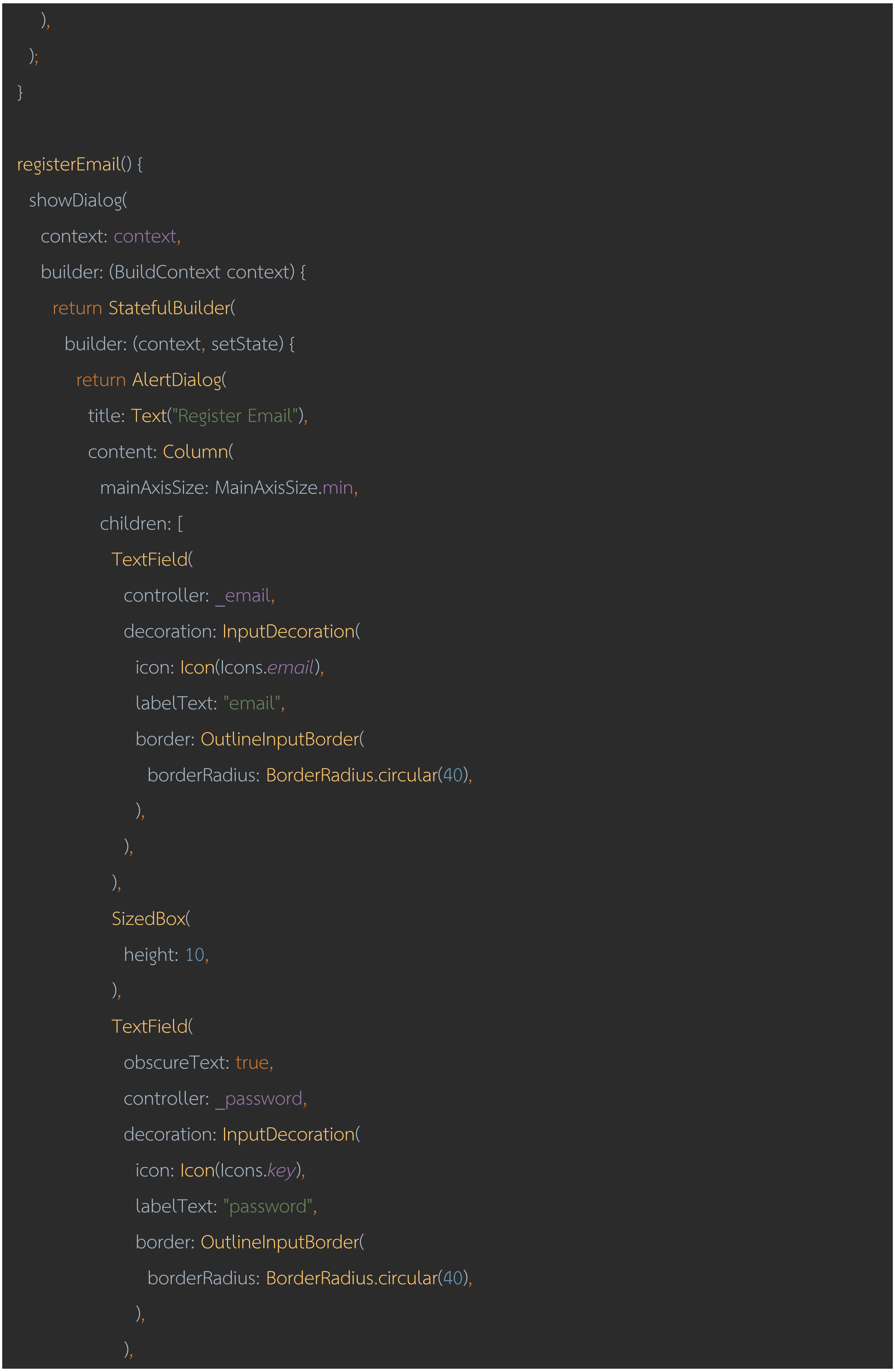
```
import 'package:chapter4_cloud_firestore/main.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class SignInScreen extends StatefulWidget {
  const SignInScreen({super.key});
  @override
  State<SignInScreen> createState() => _SignInScreenState();
}

class _SignInScreenState extends State<SignInScreen> {
  TextEditingController _email = TextEditingController();
  TextEditingController _password = TextEditingController();

  final auth = FirebaseAuth.instance;

  bool _isChecked = true;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Pantip",style: TextStyle(fontSize:20,fontFamily:'Prompt',fontWeight: FontWeight.bold,color:
Colors.purple[800])),
      ),
      body: Container(
        margin: EdgeInsets.all(20),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              "ลงชื่อเข้าใช้งาน",
              style: TextStyle(fontSize:25,fontFamily:'Prompt',fontWeight: FontWeight.bold,color:
Colors.purple[800]),
            ),
            SizedBox(
              height: 10,
            ),
            TextField(
              controller: _email,
              decoration: InputDecoration(
                  icon: Icon(Icons.email),
                  labelText: "email",
```



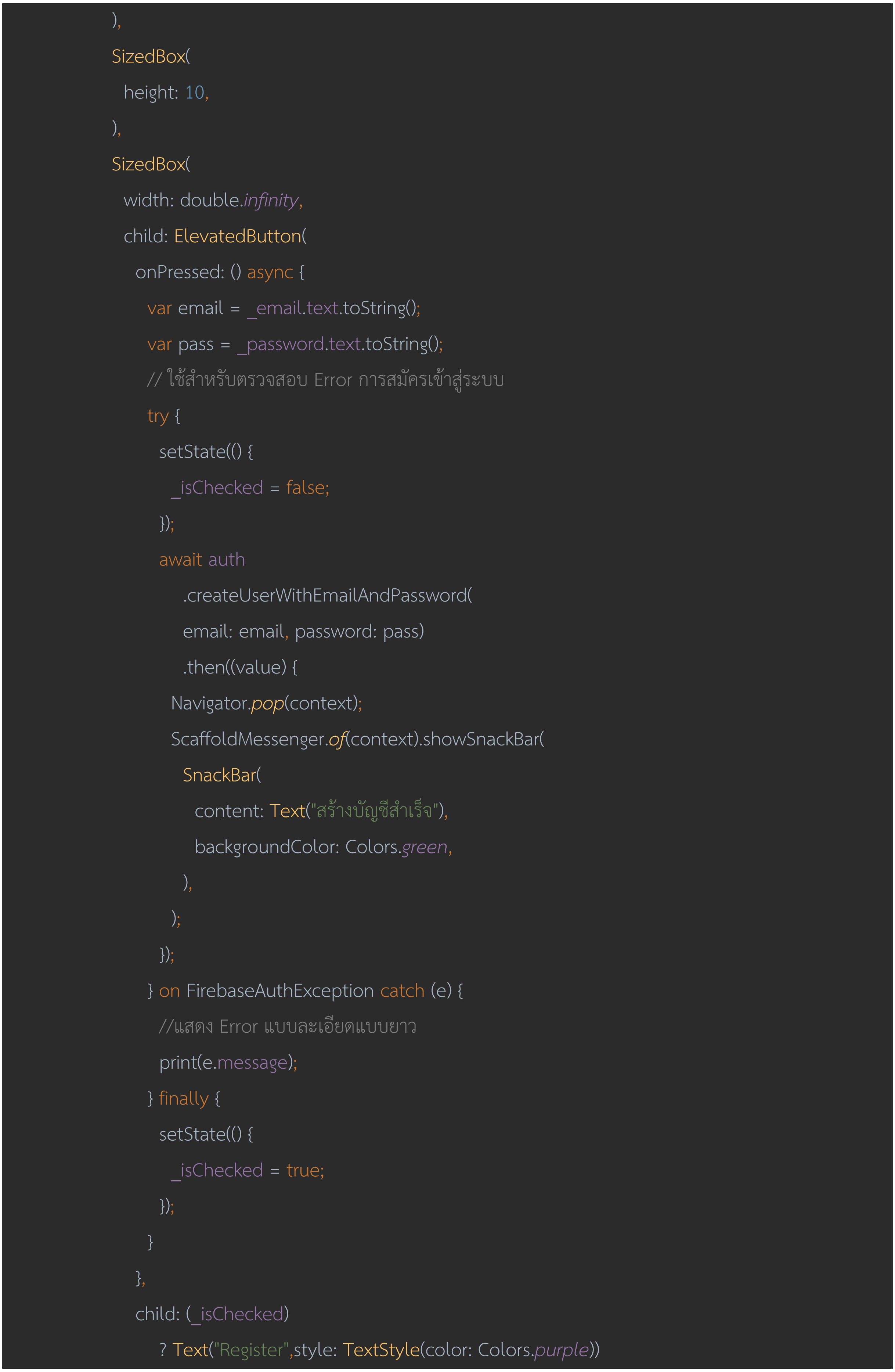
```
      border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(40))),
),
SizedBox(
  height: 10,
),
TextField(
  obscureText: true,
  controller: _password,
  decoration: InputDecoration(
      icon: Icon(Icons.key),
      labelText: "password",
      border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(40))),
),
SizedBox(
  height: 10,
),
SizedBox(
  height: 40,
  width: double.infinity,
  child: ElevatedButton(
    onPressed: () async {
      var email = _email.text.toString();
      var pass = _password.text.toString();
      setState(() {
        _isChecked = false;
      });
      try {
        await auth
            .signInWithEmailAndPassword(
            email: email, password: pass)
            .then((value) {
          Navigator.pushAndRemoveUntil(context,
              MaterialPageRoute(builder: (BuildContext context) {
                return MyApp();
              }), (route) => false);
        });
```



```
         } on FirebaseAuthException catch (e) {
          print(e.message);
          setState(() {
           _isChecked = true;
          });
         }
        },
        style: ElevatedButton.styleFrom(
         backgroundColor: Colors.purple, // เปลี่ยนสีพื้นหลังตามที่ต้องการ
        ),
        child: (_isChecked) ? Text("Login", style: TextStyle(color: Colors.white),)
          : SizedBox(
         width: 16,
         height: 16,
         child: CircularProgressIndicator(
          color: Colors.white,
          strokeWidth: 1.5,
         ),
        ),
       ),
      ),
      SizedBox(
       height: 10,
      ),
      Row(
       mainAxisAlignment: MainAxisAlignment.center,
       children: [
        TextButton(
          onPressed: () {
           registerEmail();
          },
          child: Text("Register",style: TextStyle(color: Colors.purple[800]),)),
        TextButton(onPressed: () {}, child: Text("Forget password",style: TextStyle(color:
Colors.redAccent))),
       ],
      )
     ],
    ),
```



```
    ),
  );
}

registerEmail() {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return StatefulBuilder(
        builder: (context, setState) {
          return AlertDialog(
            title: Text("Register Email"),
            content: Column(
              mainAxisSize: MainAxisSize.min,
              children: [
                TextField(
                  controller: _email,
                  decoration: InputDecoration(
                    icon: Icon(Icons.email),
                    labelText: "email",
                    border: OutlineInputBorder(
                      borderRadius: BorderRadius.circular(40),
                    ),
                  ),
                ),
                SizedBox(
                  height: 10,
                ),
                TextField(
                  obscureText: true,
                  controller: _password,
                  decoration: InputDecoration(
                    icon: Icon(Icons.key),
                    labelText: "password",
                    border: OutlineInputBorder(
                      borderRadius: BorderRadius.circular(40),
                    ),
                  ),
```

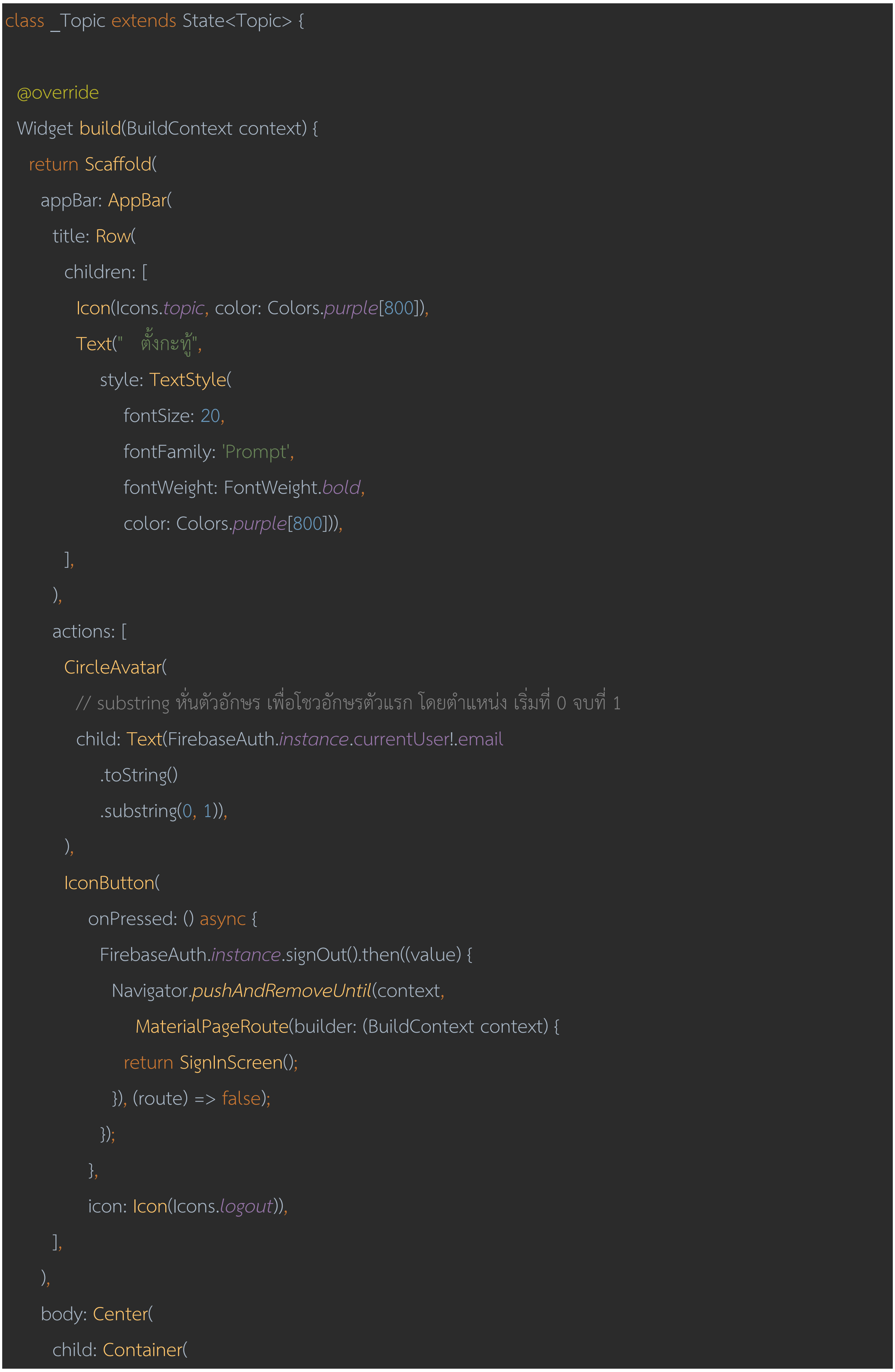


```
),
SizedBox(
  height: 10,
),
SizedBox(
  width: double.infinity,
  child: ElevatedButton(
    onPressed: () async {
      var email = _email.text.toString();
      var pass = _password.text.toString();
      // ใช้สำหรับตรวจสอบ Error การสมัครเข้าสู่ระบบ
      try {
        setState(() {
          _isChecked = false;
        });
        await auth
            .createUserWithEmailAndPassword(
            email: email, password: pass)
            .then((value) {
          Navigator.pop(context);
          ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(
              content: Text("สร้างบัญชีสำเร็จ"),
              backgroundColor: Colors.green,
            ),
          );
        });
      } on FirebaseAuthException catch (e) {
        //แสดง Error แบบละเอียดแบบยาว
        print(e.message);
      } finally {
        setState(() {
          _isChecked = true;
        });
      }
    },
    child: (_isChecked)
        ? Text("Register",style: TextStyle(color: Colors.purple))
```

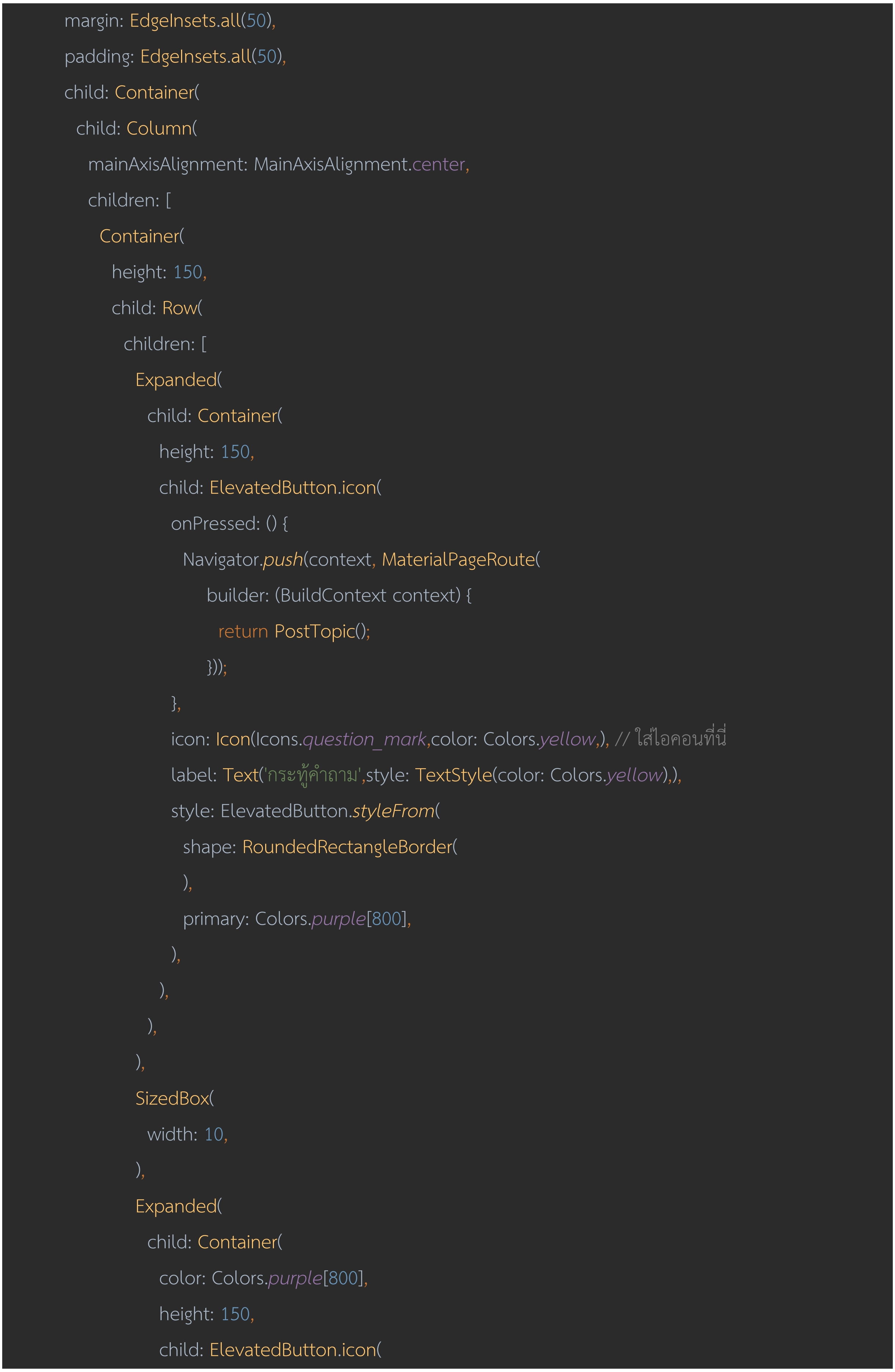
```
                          : SizedBox(
                        width: 16,
                        height: 16,
                        child: CircularProgressIndicator(
                          color: Colors.white,
                          strokeWidth: 1.5,
                        ),
                      ),
                    ),
                  ),
                ],
              ),
            );
          },
        );
      },
    );
  }
}
```

หน้า topic.dart

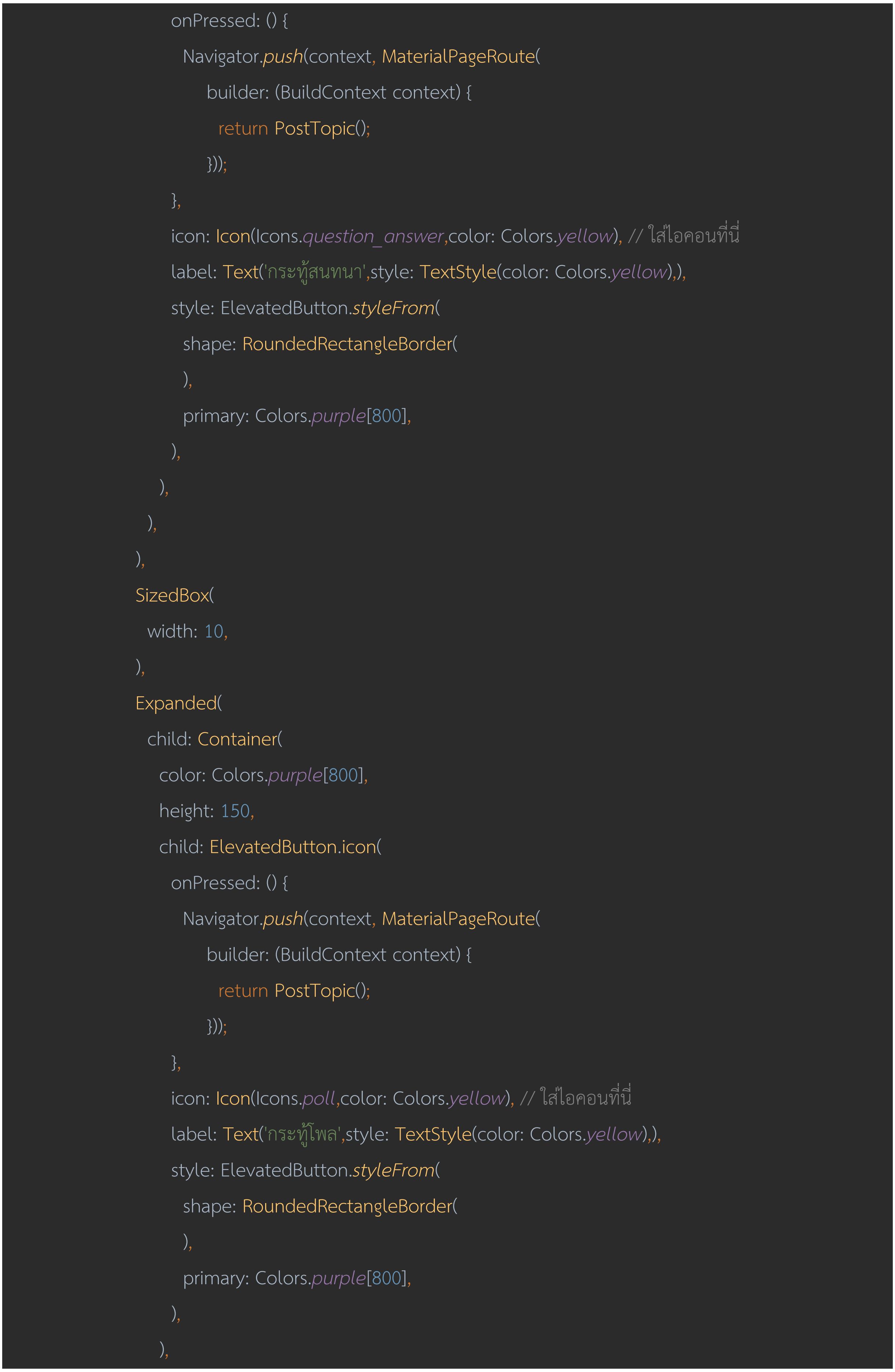
```
import 'package:chapter4_cloud_firestore/page/post_topic.dart';
import 'package:chapter4_cloud_firestore/page/sign_in_screen.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

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

  @override
  State<Topic> createState() => _Topic();
}
```



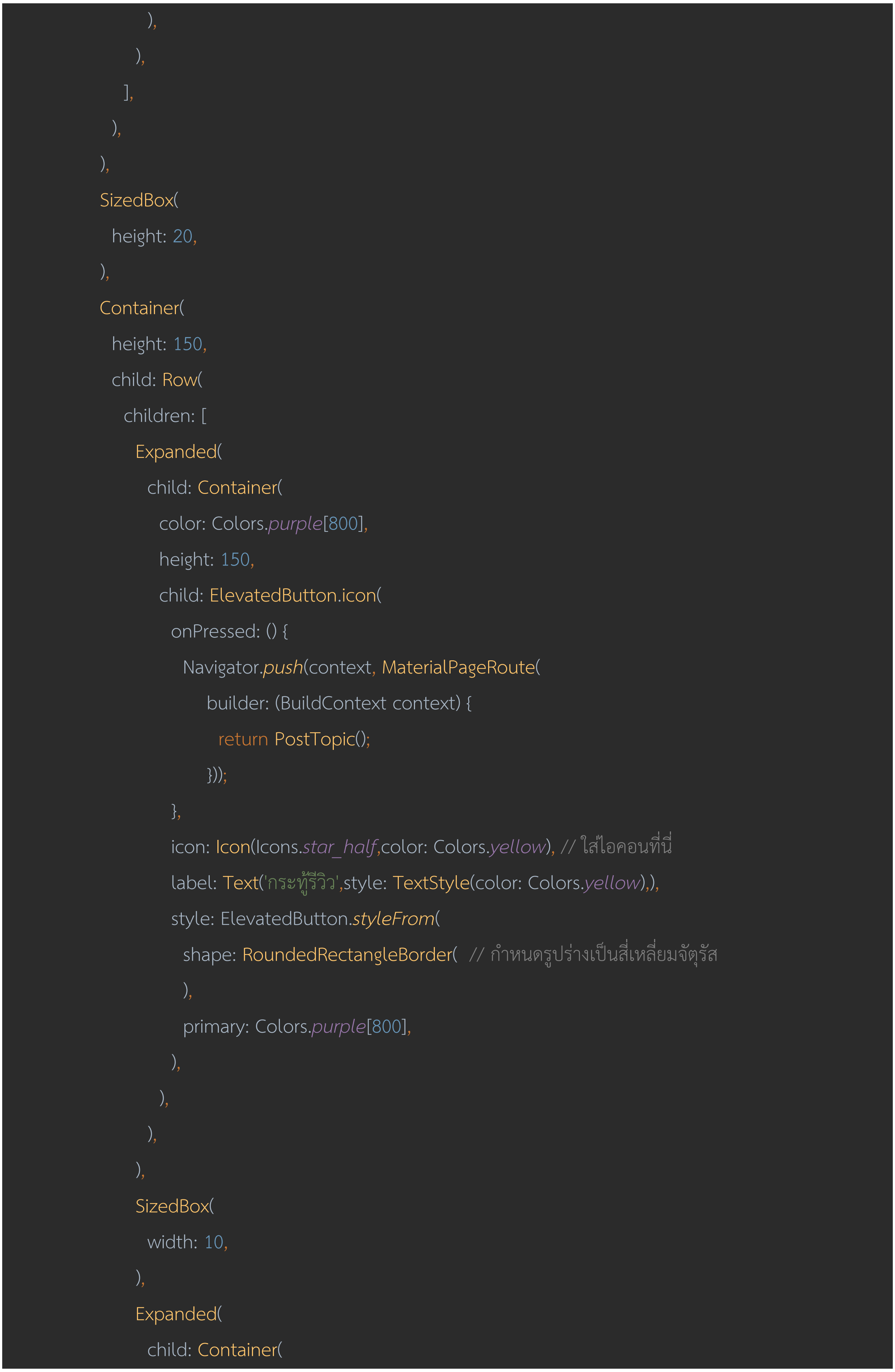
```
class _Topic extends State<Topic> {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Row(
          children: [
            Icon(Icons.topic, color: Colors.purple[800]),
            Text("   ตั้งกะทู้",
                style: TextStyle(
                    fontSize: 20,
                    fontFamily: 'Prompt',
                    fontWeight: FontWeight.bold,
                    color: Colors.purple[800])),
          ],
        ),
        actions: [
          CircleAvatar(
            // substring หั่นตัวอักษร เพื่อโชวอักษรตัวแรก โดยตำแหน่ง เริ่มที่ 0 จบที่ 1
            child: Text(FirebaseAuth.instance.currentUser!.email
                .toString()
                .substring(0, 1)),
          ),
          IconButton(
              onPressed: () async {
                FirebaseAuth.instance.signOut().then((value) {
                  Navigator.pushAndRemoveUntil(context,
                      MaterialPageRoute(builder: (BuildContext context) {
                    return SignInScreen();
                  }), (route) => false);
                });
              },
              icon: Icon(Icons.logout)),
        ],
      ),
      body: Center(
        child: Container(
```



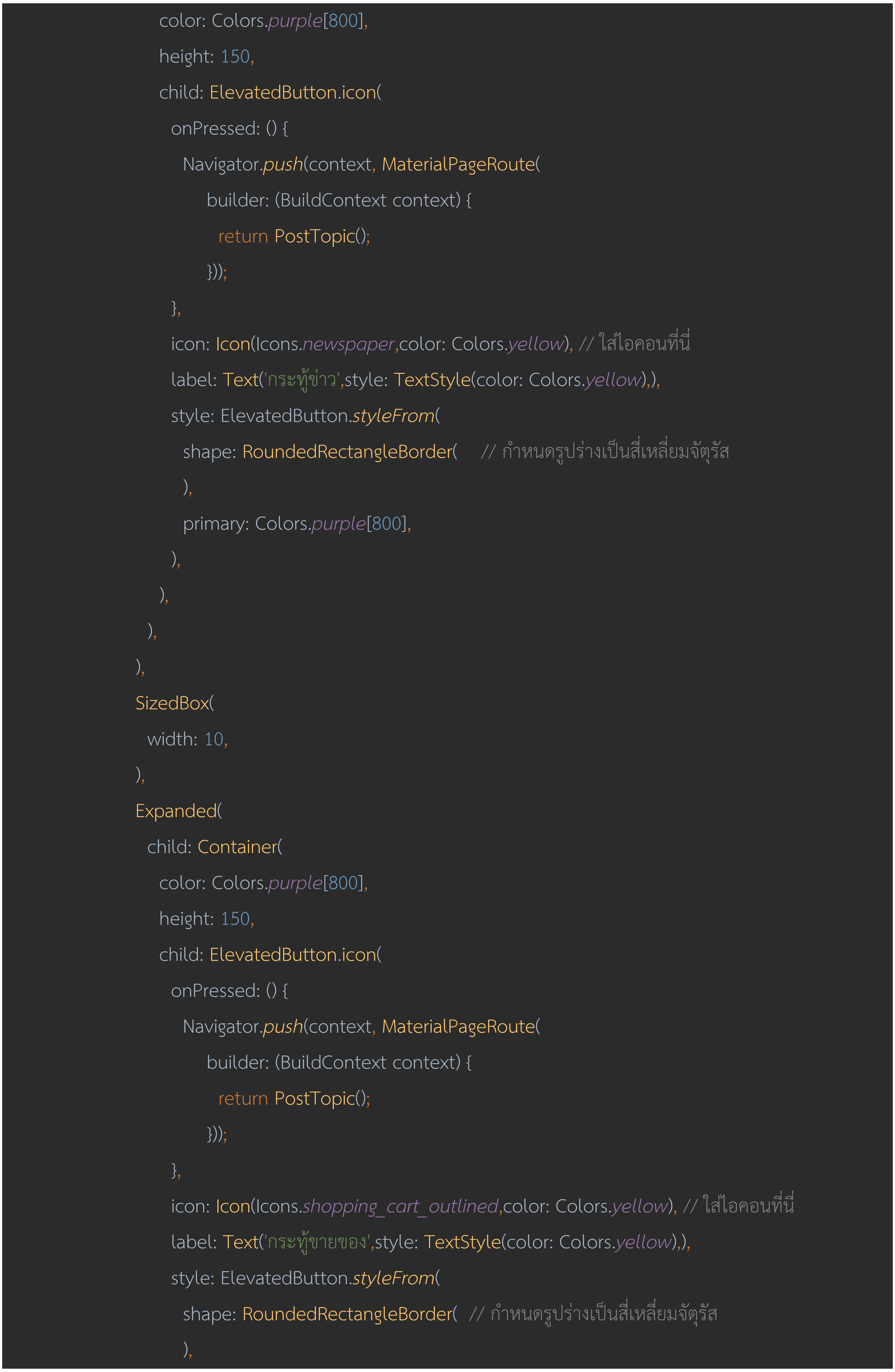
```
margin: EdgeInsets.all(50),
padding: EdgeInsets.all(50),
child: Container(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Container(
        height: 150,
        child: Row(
          children: [
            Expanded(
              child: Container(
                height: 150,
                child: ElevatedButton.icon(
                  onPressed: () {
                    Navigator.push(context, MaterialPageRoute(
                        builder: (BuildContext context) {
                          return PostTopic();
                        }));
                  },
                  icon: Icon(Icons.question_mark,color: Colors.yellow,), // ใส่ไอคอนที่นี่
                  label: Text('กระทู้คำถาม',style: TextStyle(color: Colors.yellow),),
                  style: ElevatedButton.styleFrom(
                    shape: RoundedRectangleBorder(
                    ),
                    primary: Colors.purple[800],
                  ),
                ),
              ),
            ),
            SizedBox(
              width: 10,
            ),
            Expanded(
              child: Container(
                color: Colors.purple[800],
                height: 150,
                child: ElevatedButton.icon(
```



```
      onPressed: () {
        Navigator.push(context, MaterialPageRoute(
            builder: (BuildContext context) {
              return PostTopic();
            }));
      },
      icon: Icon(Icons.question_answer,color: Colors.yellow), // ใส่ไอคอนที่นี่
      label: Text('กระทู้สนทนา',style: TextStyle(color: Colors.yellow),),
      style: ElevatedButton.styleFrom(
        shape: RoundedRectangleBorder(
        ),
        primary: Colors.purple[800],
      ),
    ),
  ),
),
SizedBox(
  width: 10,
),
Expanded(
  child: Container(
    color: Colors.purple[800],
    height: 150,
    child: ElevatedButton.icon(
      onPressed: () {
        Navigator.push(context, MaterialPageRoute(
            builder: (BuildContext context) {
              return PostTopic();
            }));
      },
      icon: Icon(Icons.poll,color: Colors.yellow), // ใส่ไอคอนที่นี่
      label: Text('กระทู้โพล',style: TextStyle(color: Colors.yellow),),
      style: ElevatedButton.styleFrom(
        shape: RoundedRectangleBorder(
        ),
        primary: Colors.purple[800],
      ),
    ),
```



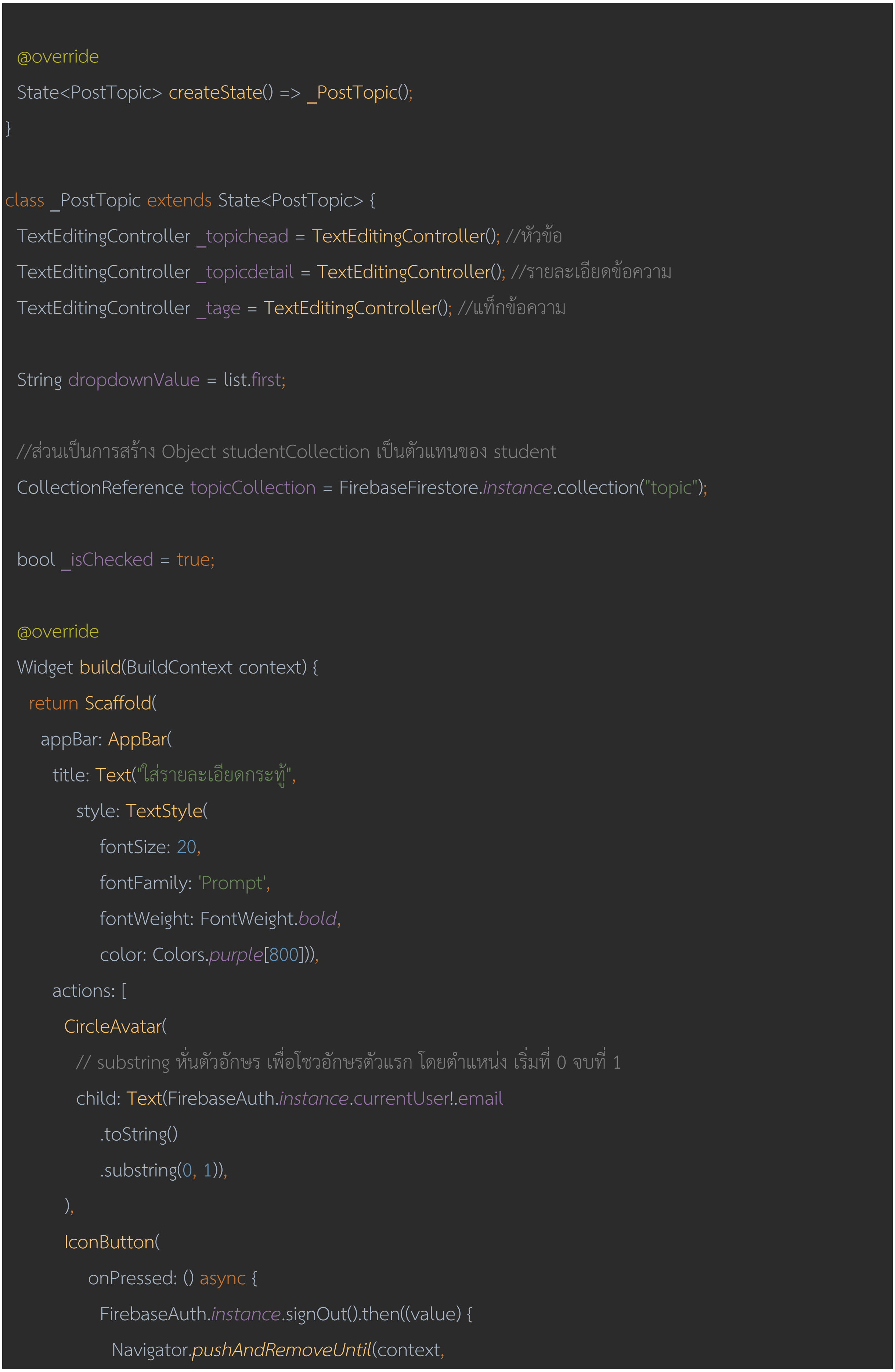
```
        ),
      ),
    ],
  ),
),
SizedBox(
  height: 20,
),
Container(
  height: 150,
  child: Row(
    children: [
      Expanded(
        child: Container(
          color: Colors.purple[800],
          height: 150,
          child: ElevatedButton.icon(
            onPressed: () {
              Navigator.push(context, MaterialPageRoute(
                  builder: (BuildContext context) {
                    return PostTopic();
                  }));
            },
            icon: Icon(Icons.star_half,color: Colors.yellow), // ใส่ไอคอนที่นี่
            label: Text('กระทู้รีวิว',style: TextStyle(color: Colors.yellow),),
            style: ElevatedButton.styleFrom(
              shape: RoundedRectangleBorder(  // กำหนดรูปร่างเป็นสี่เหลี่ยมจัตุรัส
              ),
              primary: Colors.purple[800],
            ),
          ),
        ),
      ),
      SizedBox(
        width: 10,
      ),
      Expanded(
        child: Container(
```



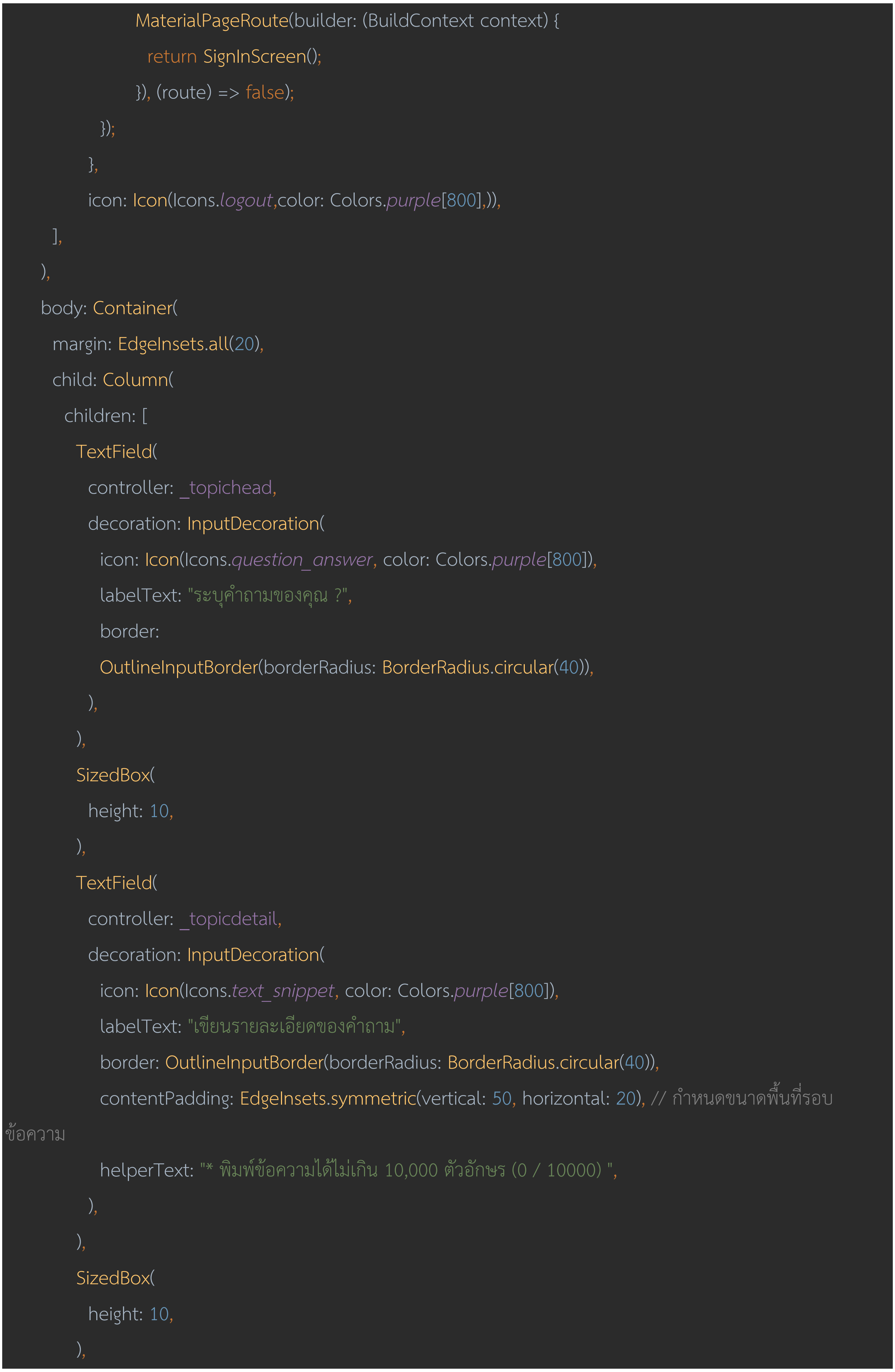
```
      color: Colors.purple[800],
      height: 150,
      child: ElevatedButton.icon(
       onPressed: () {
        Navigator.push(context, MaterialPageRoute(
          builder: (BuildContext context) {
           return PostTopic();
          }));
       },
       icon: Icon(Icons.newspaper,color: Colors.yellow), // ใส่ไอคอนที่นี่
       label: Text('กระทู้ข่าว',style: TextStyle(color: Colors.yellow),),
       style: ElevatedButton.styleFrom(
        shape: RoundedRectangleBorder(   // กำหนดรูปร่างเป็นสี่เหลี่ยมจัตุรัส
        ),
        primary: Colors.purple[800],
       ),
      ),
     ),
    ),
    SizedBox(
     width: 10,
    ),
    Expanded(
     child: Container(
      color: Colors.purple[800],
      height: 150,
      child: ElevatedButton.icon(
       onPressed: () {
        Navigator.push(context, MaterialPageRoute(
          builder: (BuildContext context) {
           return PostTopic();
          }));
       },
       icon: Icon(Icons.shopping_cart_outlined,color: Colors.yellow), // ใส่ไอคอนที่นี่
       label: Text('กระทู้ขายของ',style: TextStyle(color: Colors.yellow),),
       style: ElevatedButton.styleFrom(
        shape: RoundedRectangleBorder(  // กำหนดรูปร่างเป็นสี่เหลี่ยมจัตุรัส
        ),
                              primary: Colors.purple[800],
                            ),
                          ),
                        ),
                      ),
                    ],
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

หน้า post_topic.dart

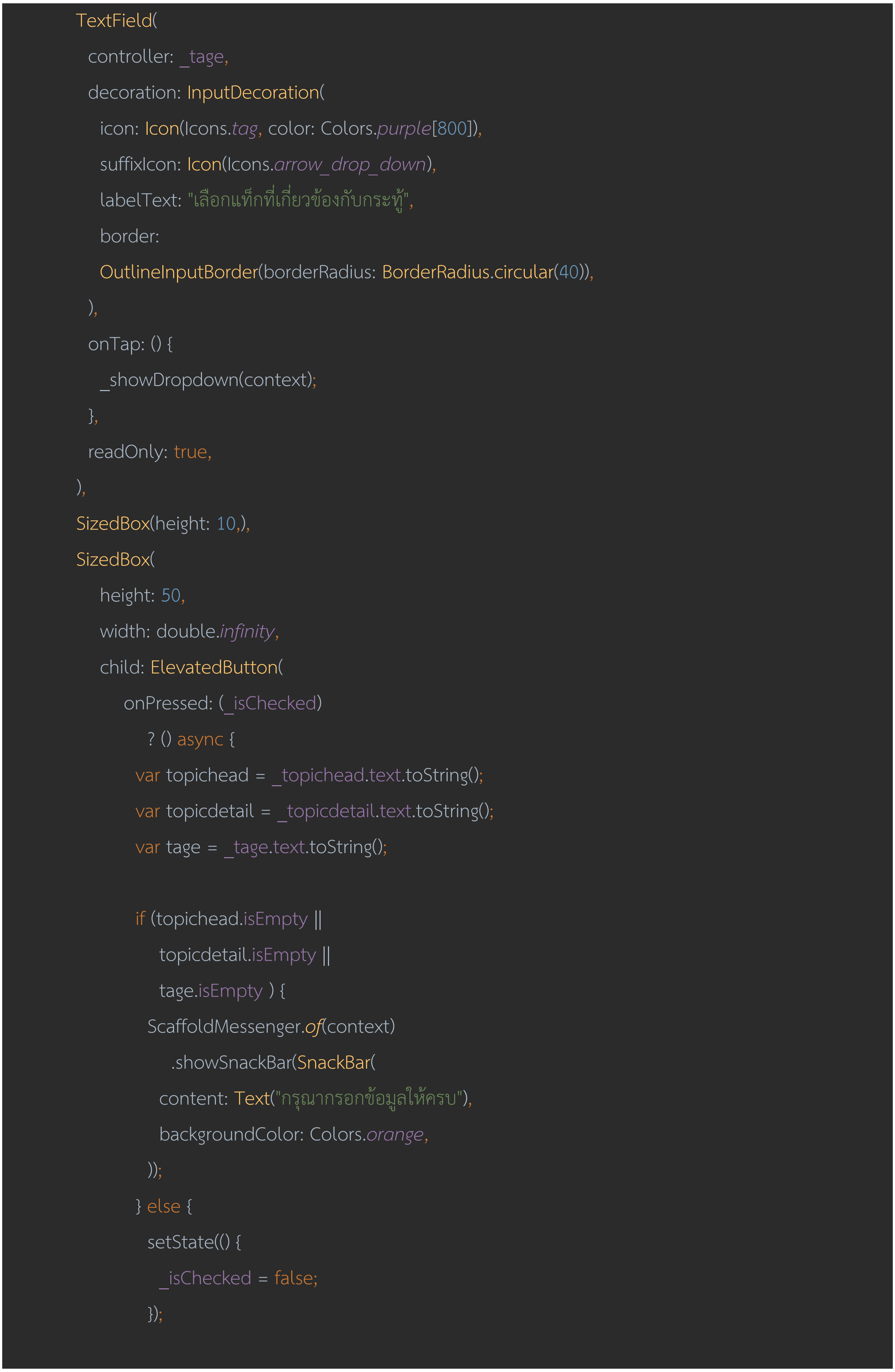
```
import 'package:chapter4_cloud_firestore/page/list_topic.dart';
import 'package:chapter4_cloud_firestore/page/sign_in_screen.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';




List<String> list = <String>[
  '#ร้านกาแฟยอดฮิต',
  '#How to เครื่องสำอาง',
  '#พรีเมียร์ลีก',
  '#cafe',
  '#เที่ยวต่างประเทศ',
  '#ร้านอาหารในเชียงใหม่',
];

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



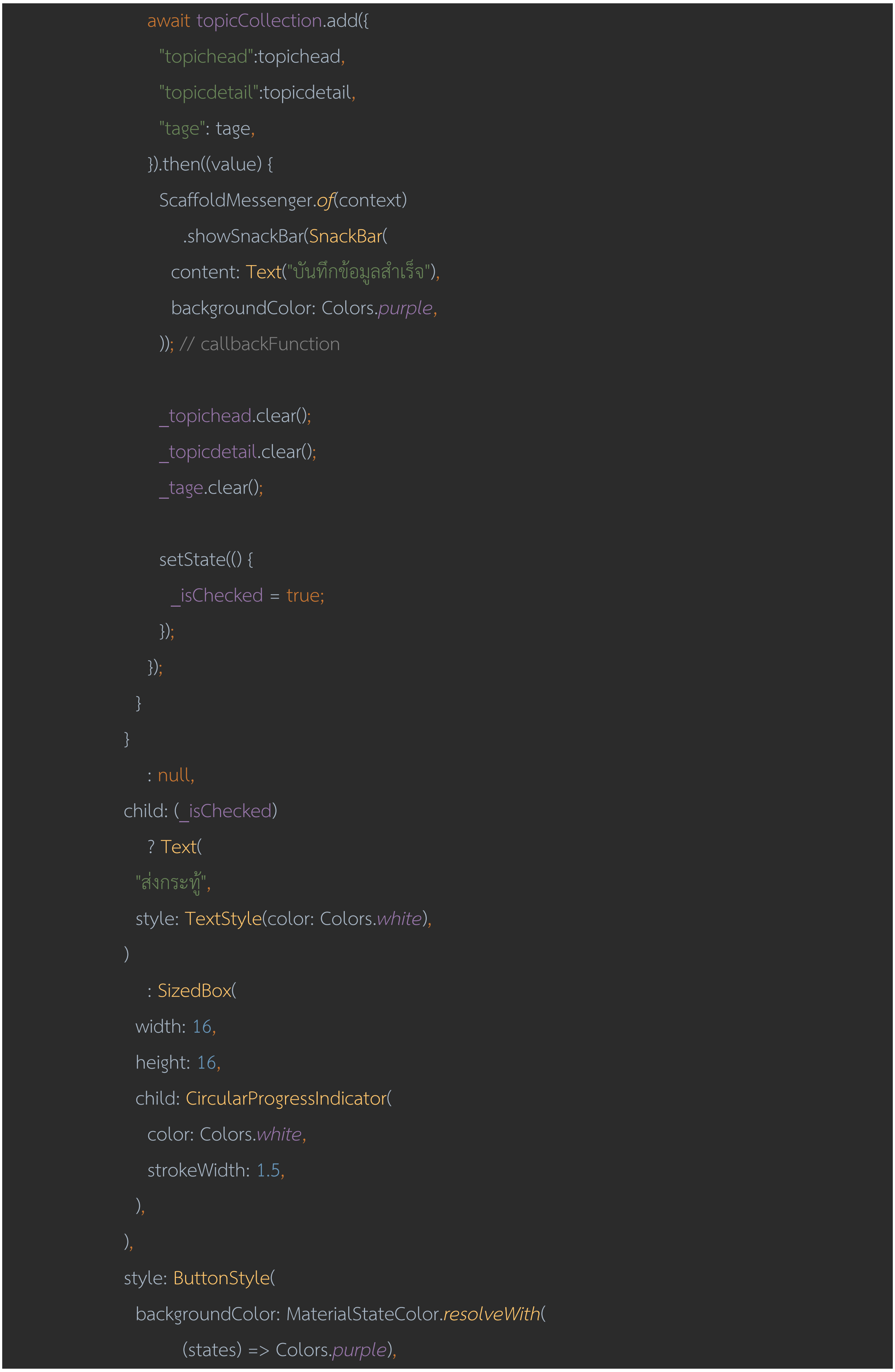
```
  @override
  State<PostTopic> createState() => _PostTopic();
}

class _PostTopic extends State<PostTopic> {
  TextEditingController _topichead = TextEditingController(); //หัวข้อ
  TextEditingController _topicdetail = TextEditingController(); //รายละเอียดข้อความ
  TextEditingController _tage = TextEditingController(); //แท็กข้อความ

  String dropdownValue = list.first;

  //ส่วนเป็นการสร้าง Object studentCollection เป็นตัวแทนของ student
  CollectionReference topicCollection = FirebaseFirestore.instance.collection("topic");

  bool _isChecked = true;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("ใส่รายละเอียดกระทู้",
            style: TextStyle(
                fontSize: 20,
                fontFamily: 'Prompt',
                fontWeight: FontWeight.bold,
                color: Colors.purple[800])),
        actions: [
          CircleAvatar(
            // substring หั่นตัวอักษร เพื่อโชวอักษรตัวแรก โดยตำแหน่ง เริ่มที่ 0 จบที่ 1
            child: Text(FirebaseAuth.instance.currentUser!.email
                .toString()
                .substring(0, 1)),
          ),
          IconButton(
              onPressed: () async {
                FirebaseAuth.instance.signOut().then((value) {
                  Navigator.pushAndRemoveUntil(context,
```



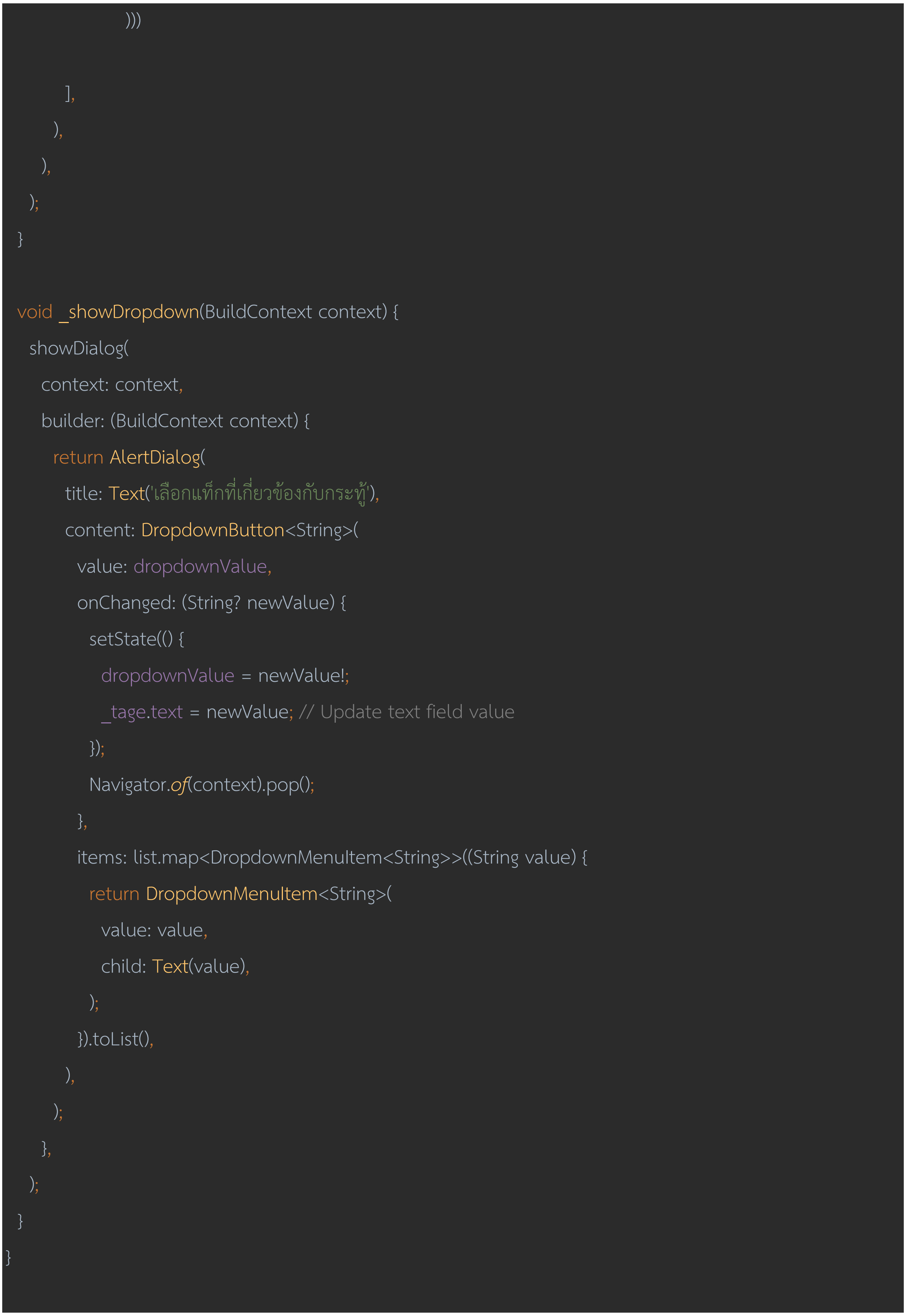
```
                      MaterialPageRoute(builder: (BuildContext context) {
                        return SignInScreen();
                      }), (route) => false);
                }); 
              },
              icon: Icon(Icons.logout,color: Colors.purple[800],)),
        ],
      ),
      body: Container(
        margin: EdgeInsets.all(20),
        child: Column(
          children: [
            TextField(
              controller: _topichead,
              decoration: InputDecoration(
                icon: Icon(Icons.question_answer, color: Colors.purple[800]),
                labelText: "ระบุคำถามของคุณ ?",
                border:
                OutlineInputBorder(borderRadius: BorderRadius.circular(40)),
              ),
            ),
            SizedBox(
              height: 10,
            ),
            TextField(
              controller: _topicdetail,
              decoration: InputDecoration(
                icon: Icon(Icons.text_snippet, color: Colors.purple[800]),
                labelText: "เขียนรายละเอียดของคำถาม",
                border: OutlineInputBorder(borderRadius: BorderRadius.circular(40)),
                contentPadding: EdgeInsets.symmetric(vertical: 50, horizontal: 20), // กำหนดขนาดพื้นที่รอบ
ข้อความ
                helperText: "* พิมพ์ข้อความได้ไม่เกิน 10,000 ตัวอักษร (0 / 10000) ",
              ),
            ),
            SizedBox(
              height: 10,
            ),
```



```
TextField(
  controller: _tage,
  decoration: InputDecoration(
    icon: Icon(Icons.tag, color: Colors.purple[800]),
    suffixIcon: Icon(Icons.arrow_drop_down),
    labelText: "เลือกแท็กที่เกี่ยวข้องกับกระทู้",
    border:
    OutlineInputBorder(borderRadius: BorderRadius.circular(40)),
  ),
  onTap: () {
    _showDropdown(context);
  },
  readOnly: true,
),
SizedBox(height: 10,),
SizedBox(
    height: 50,
    width: double.infinity,
    child: ElevatedButton(
        onPressed: (_isChecked)
            ? () async {
          var topichead = _topichead.text.toString();
          var topicdetail = _topicdetail.text.toString();
          var tage = _tage.text.toString();

          if (topichead.isEmpty ||
              topicdetail.isEmpty ||
              tage.isEmpty ) {
            ScaffoldMessenger.of(context)
                .showSnackBar(SnackBar(
              content: Text("กรุณากรอกข้อมูลให้ครบ"),
              backgroundColor: Colors.orange,
            ));
          } else {
            setState(() {
              _isChecked = false;
            });
```



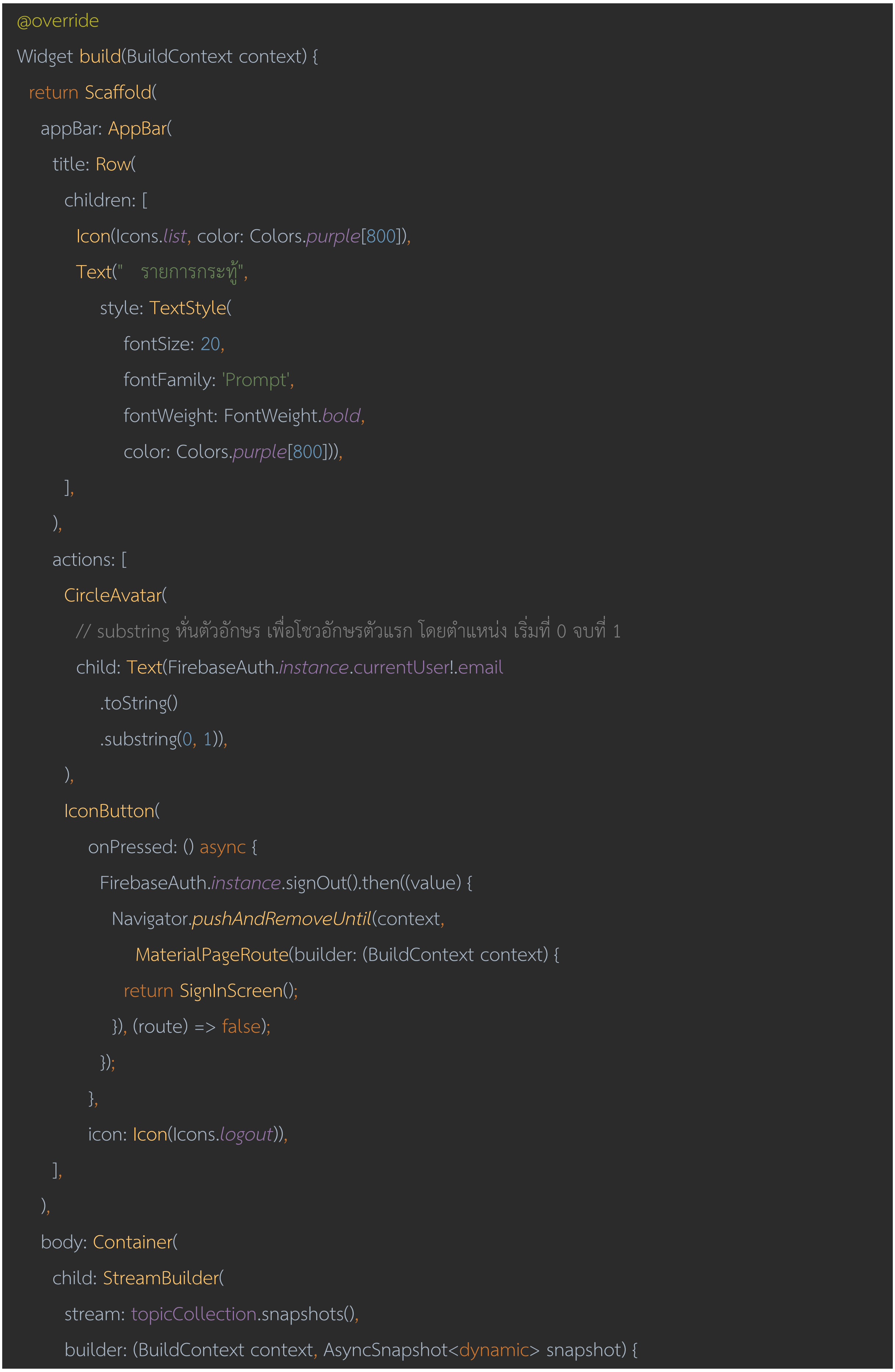
```
    await topicCollection.add({
      "topichead":topichead,
      "topicdetail":topicdetail,
      "tage": tage,
    }).then((value) {
      ScaffoldMessenger.of(context)
          .showSnackBar(SnackBar(
        content: Text("บันทึกข้อมูลสำเร็จ"),
        backgroundColor: Colors.purple,
      )); // callbackFunction

      _topichead.clear();
      _topicdetail.clear();
      _tage.clear();

      setState(() {
        _isChecked = true;
      });
    });
  }
}
    : null,
child: (_isChecked)
    ? Text(
  "ส่งกระทู้",
  style: TextStyle(color: Colors.white),
)
    : SizedBox(
  width: 16,
  height: 16,
  child: CircularProgressIndicator(
    color: Colors.white,
    strokeWidth: 1.5,
  ),
),
style: ButtonStyle(
  backgroundColor: MaterialStateColor.resolveWith(
          (states) => Colors.purple),
```



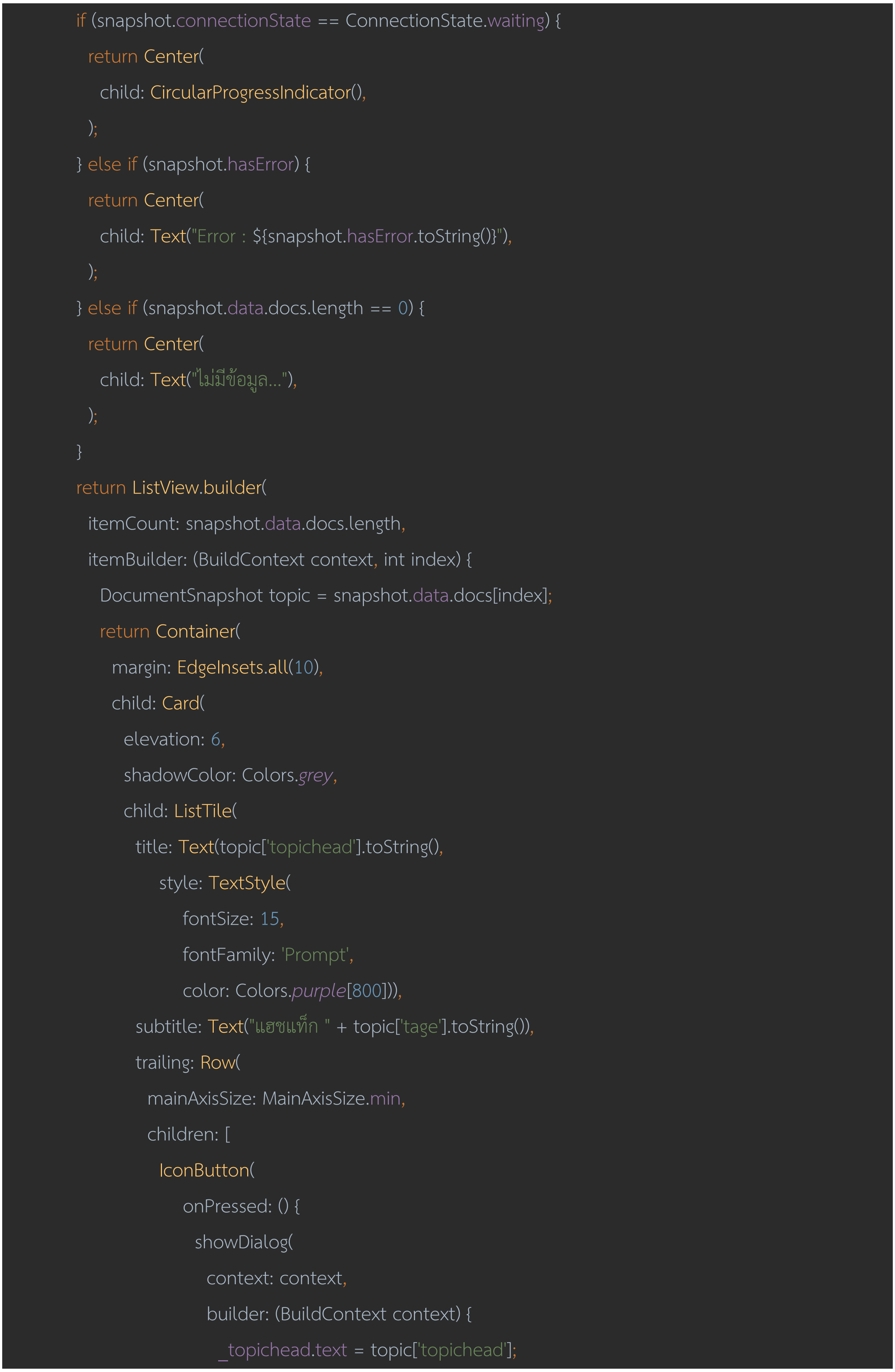
```
                    )))

          ],
        ),
      ),
    );
  }

  void _showDropdown(BuildContext context) {
    showDialog(
      context: context,
      builder: (BuildContext context) {
        return AlertDialog(
          title: Text('เลือกแท็กที่เกี่ยวข้องกับกระทู้'),
          content: DropdownButton<String>(
            value: dropdownValue,
            onChanged: (String? newValue) {
              setState(() {
                dropdownValue = newValue!;
                _tage.text = newValue; // Update text field value
              });
              Navigator.of(context).pop();
            },
            items: list.map<DropdownMenuItem<String>>((String value) {
              return DropdownMenuItem<String>(
                value: value,
                child: Text(value),
              );
            }).toList(),
          ),
        );
      },
    );
  }
}
```

หน้า list_topic.drat

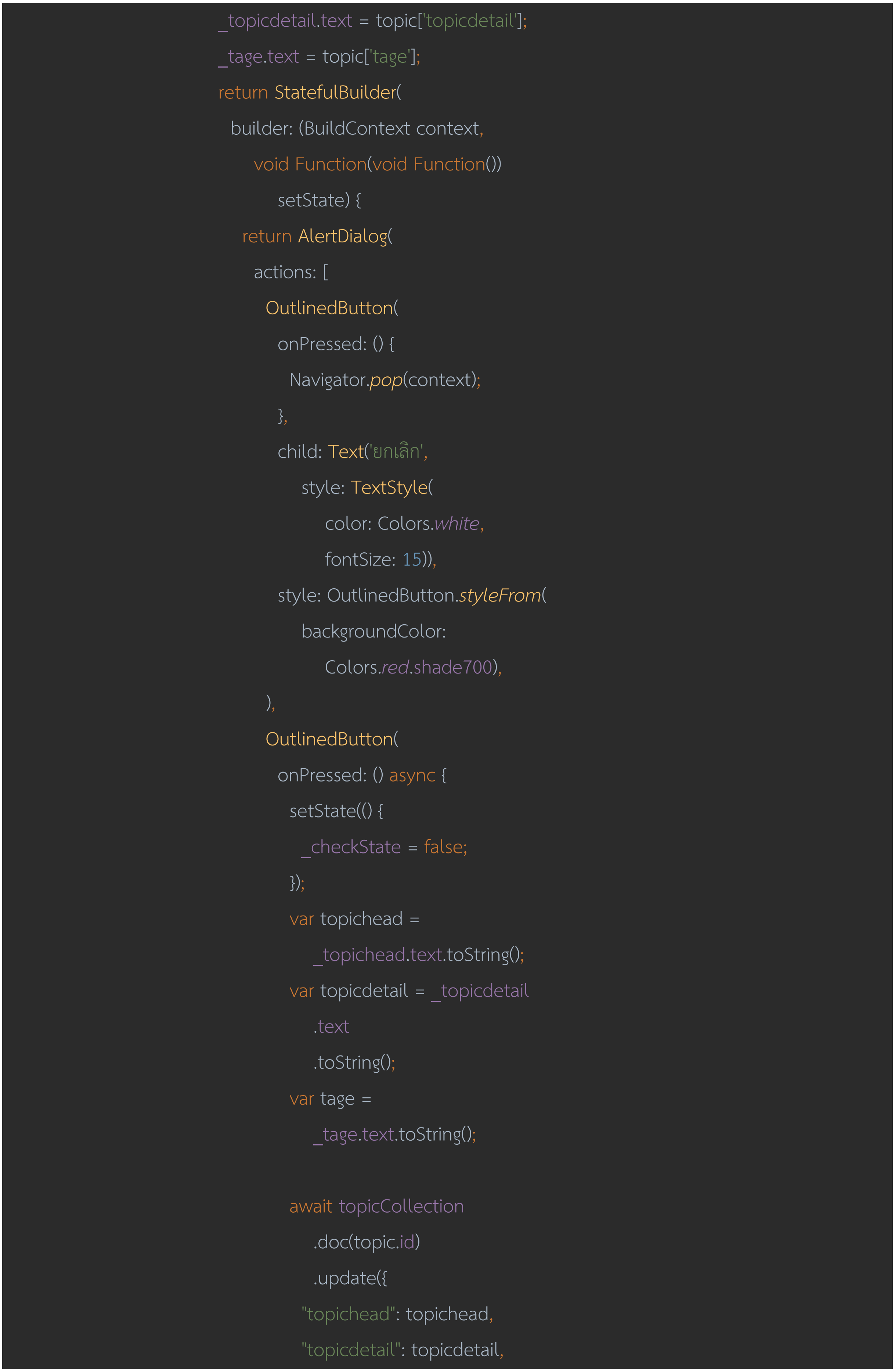
```
import 'package:chapter4_cloud_firestore/page/sign_in_screen.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

List<String> list = <String>[
  '#ร้านกาแฟยอดฮิต',
  '#How to เครื่องสำอาง',
  '#พรีเมียร์ลีก',
  '#cafe',
  '#เที่ยวต่างประเทศ',
  '#ร้านอาหารในเชียงใหม่',
];

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

  @override
  State<ListTopic> createState() => _ListTopic();
}

class _ListTopic extends State<ListTopic> {
  CollectionReference topicCollection =
      FirebaseFirestore.instance.collection("topic");

  bool _checkState = true;

  TextEditingController _topichead = TextEditingController(); //หัวข้อ
  TextEditingController _topicdetail =
      TextEditingController(); //รายละเอียดข้อความ
  TextEditingController _tage = TextEditingController(); //แท็กข้อความ

  String dropdownValue = list.first;

  //จำนวนตัว
```



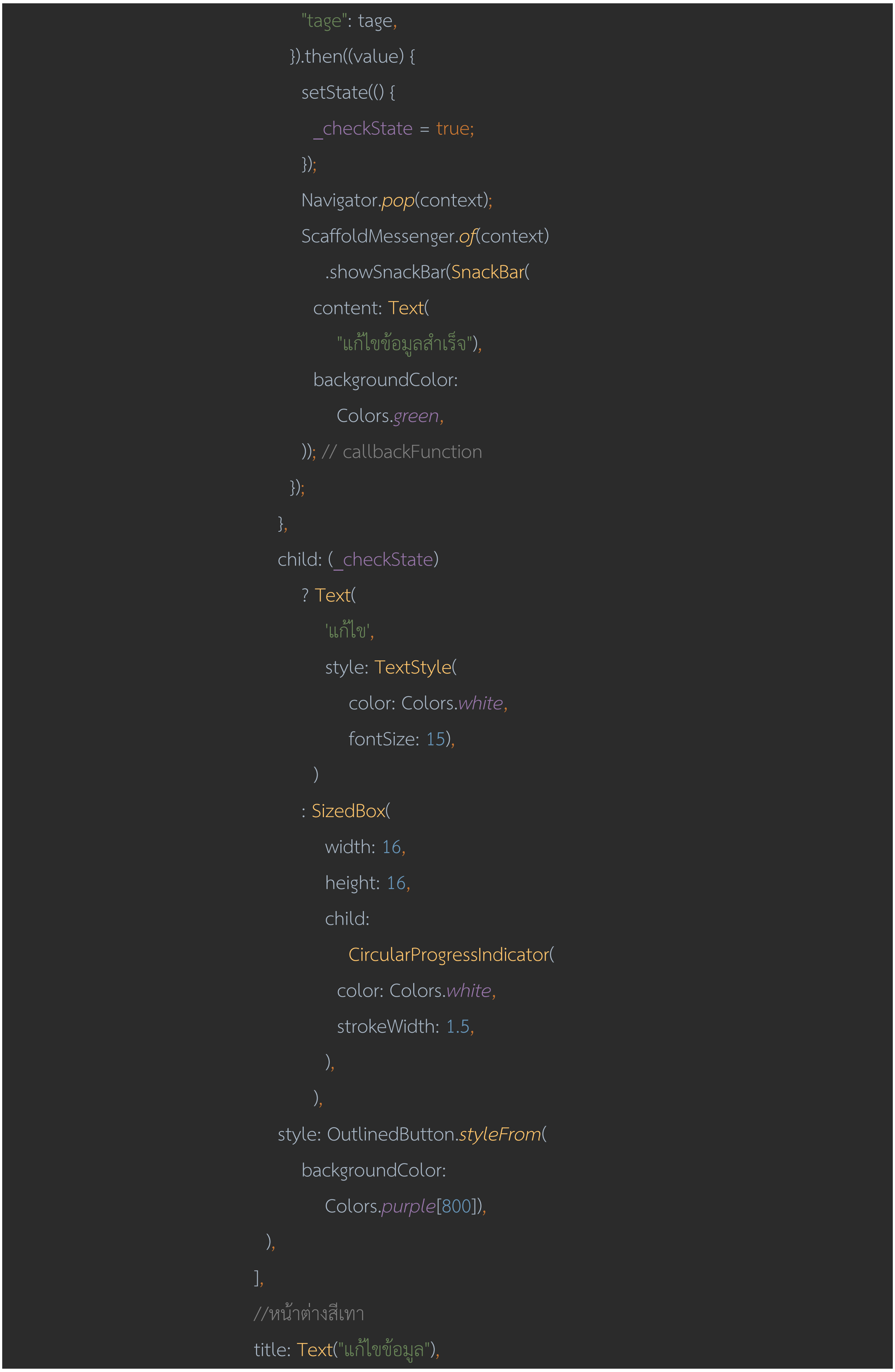
```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Row(
        children: [
          Icon(Icons.list, color: Colors.purple[800]),
          Text("   รายการกระทู้",
              style: TextStyle(
                  fontSize: 20,
                  fontFamily: 'Prompt',
                  fontWeight: FontWeight.bold,
                  color: Colors.purple[800])),
        ],
      ),
      actions: [
        CircleAvatar(
          // substring หั่นตัวอักษร เพื่อโชวอักษรตัวแรก โดยตำแหน่ง เริ่มที่ 0 จบที่ 1
          child: Text(FirebaseAuth.instance.currentUser!.email
              .toString()
              .substring(0, 1)),
        ),
        IconButton(
            onPressed: () async {
              FirebaseAuth.instance.signOut().then((value) {
                Navigator.pushAndRemoveUntil(context,
                    MaterialPageRoute(builder: (BuildContext context) {
                  return SignInScreen();
                }), (route) => false);
              });
            },
            icon: Icon(Icons.logout)),
      ],
    ),
    body: Container(
      child: StreamBuilder(
        stream: topicCollection.snapshots(),
        builder: (BuildContext context, AsyncSnapshot<dynamic> snapshot) {
```



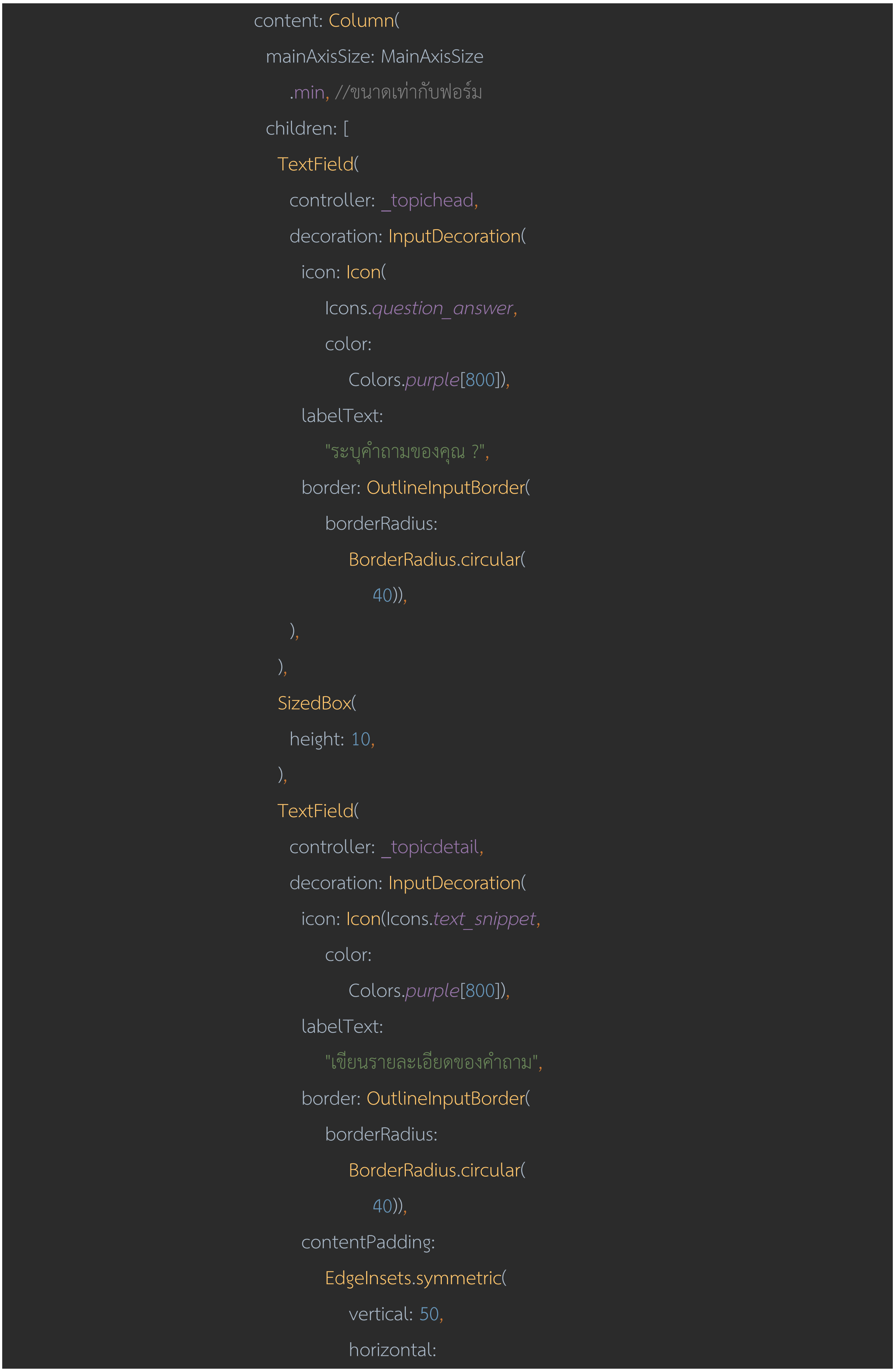
```
if (snapshot.connectionState == ConnectionState.waiting) {
  return Center(
    child: CircularProgressIndicator(),
  );
} else if (snapshot.hasError) {
  return Center(
    child: Text("Error : ${snapshot.hasError.toString()}"),
  );
} else if (snapshot.data.docs.length == 0) {
  return Center(
    child: Text("ไม่มีข้อมูล..."),
  );
}
return ListView.builder(
  itemCount: snapshot.data.docs.length,
  itemBuilder: (BuildContext context, int index) {
    DocumentSnapshot topic = snapshot.data.docs[index];
    return Container(
      margin: EdgeInsets.all(10),
      child: Card(
        elevation: 6,
        shadowColor: Colors.grey,
        child: ListTile(
          title: Text(topic['topichead'].toString(),
              style: TextStyle(
                  fontSize: 15,
                  fontFamily: 'Prompt',
                  color: Colors.purple[800])),
          subtitle: Text("แฮชแท็ก " + topic['tage'].toString()),
          trailing: Row(
            mainAxisSize: MainAxisSize.min,
            children: [
              IconButton(
                  onPressed: () {
                    showDialog(
                      context: context,
                      builder: (BuildContext context) {
                        _topichead.text = topic['topichead'];
```



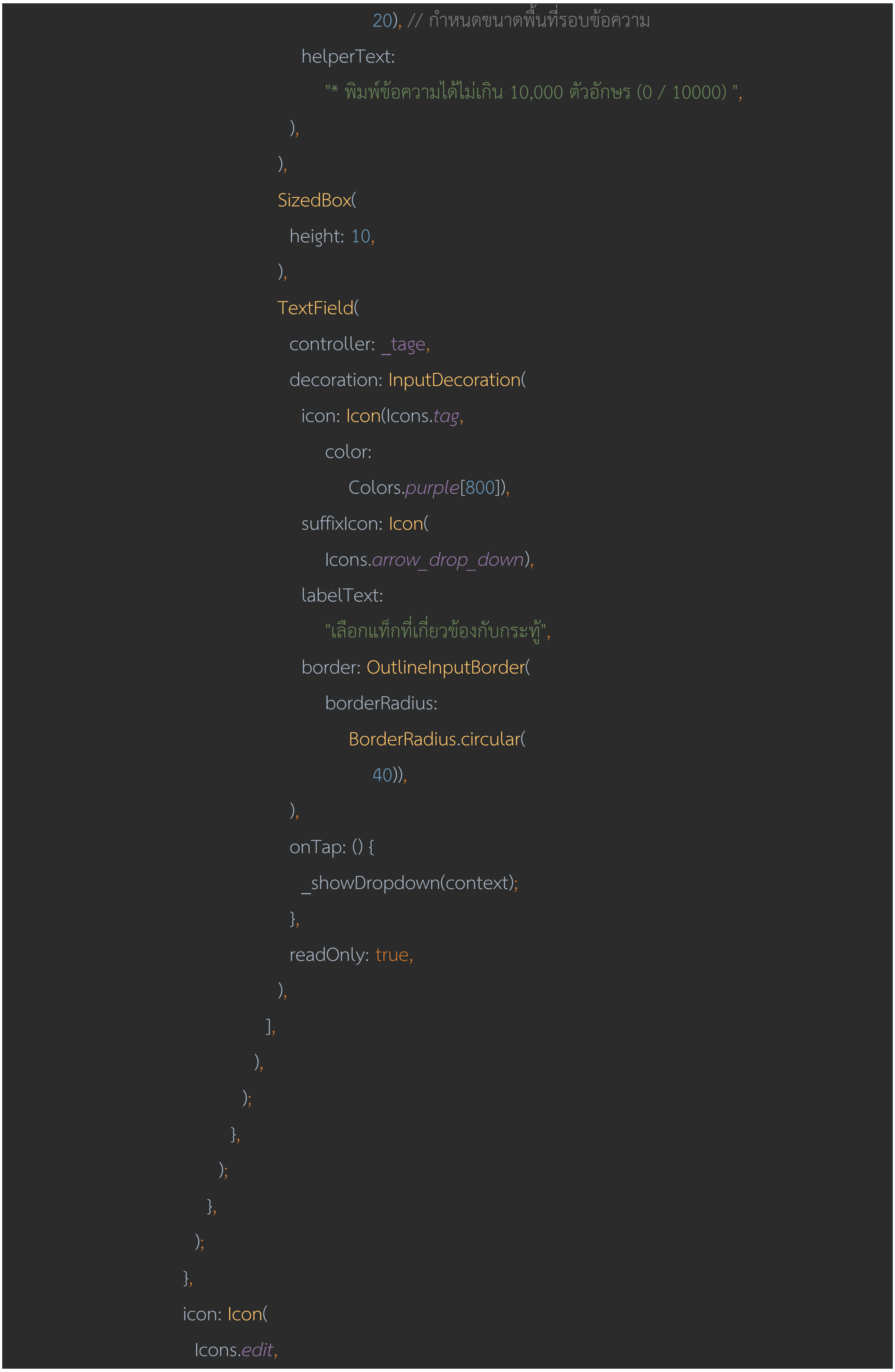
```
_topicdetail.text = topic['topicdetail'];
_tage.text = topic['tage'];
return StatefulBuilder(
 builder: (BuildContext context,
   void Function(void Function())
     setState) {
  return AlertDialog(
   actions: [
    OutlinedButton(
     onPressed: () {
      Navigator.pop(context);
     },
     child: Text('ยกเลิก',
       style: TextStyle(
         color: Colors.white,
         fontSize: 15)),
     style: OutlinedButton.styleFrom(
       backgroundColor:
         Colors.red.shade700),
    ),
    OutlinedButton(
     onPressed: () async {
      setState(() {
       _checkState = false;
      });
      var topichead =
        _topichead.text.toString();
      var topicdetail = _topicdetail
        .text
        .toString();
      var tage =
        _tage.text.toString();

      await topicCollection
        .doc(topic.id)
        .update({
       "topichead": topichead,
       "topicdetail": topicdetail,
```



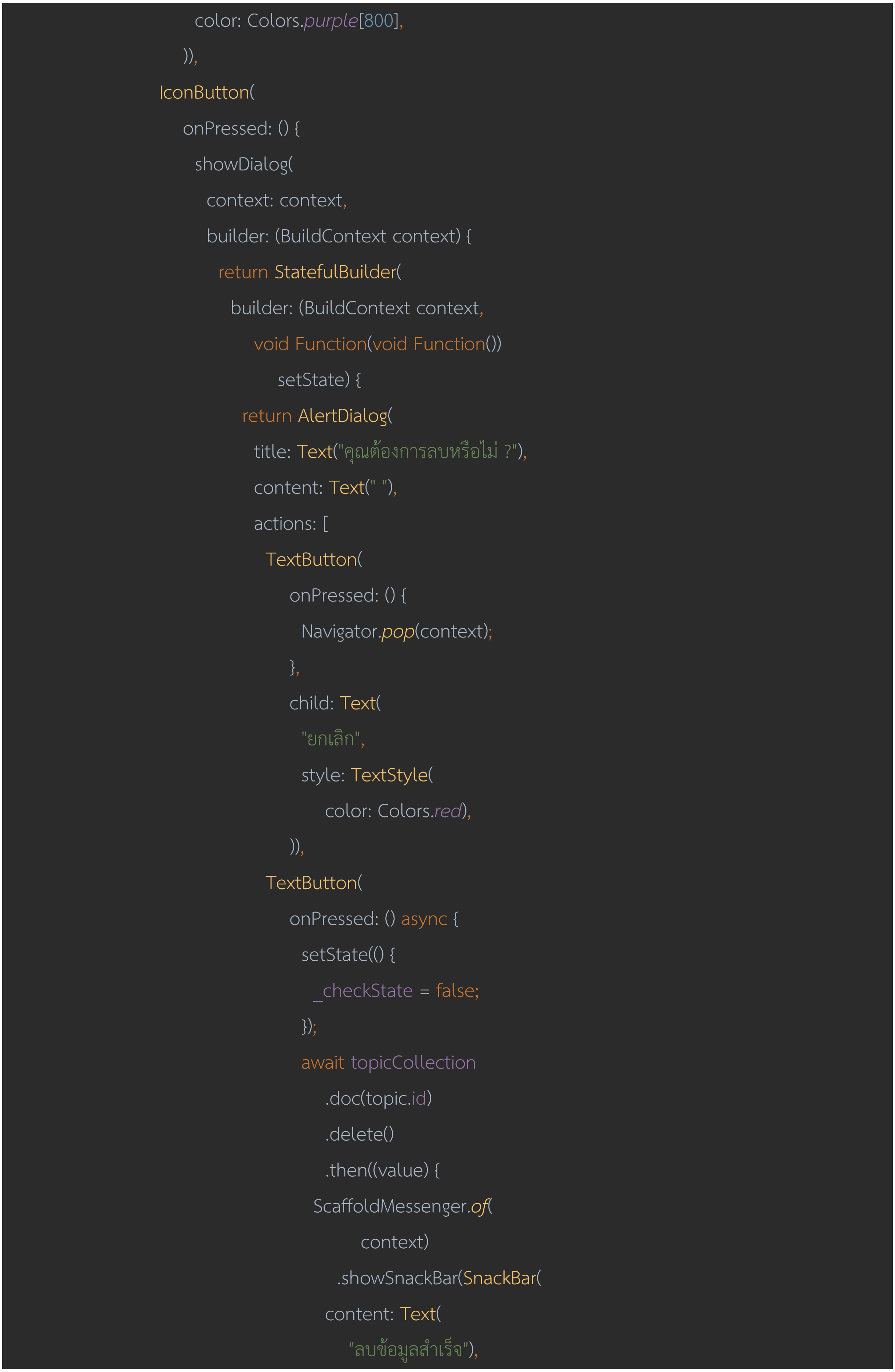
```
    "tage": tage,
   }).then((value) {
    setState(() {
     _checkState = true;
    });
    Navigator.pop(context);
    ScaffoldMessenger.of(context)
      .showSnackBar(SnackBar(
     content: Text(
       "แก้ไขข้อมูลสำเร็จ"),
     backgroundColor:
       Colors.green,
    )); // callbackFunction
   });
  },
  child: (_checkState)
    ? Text(
      'แก้ไข',
      style: TextStyle(
        color: Colors.white,
        fontSize: 15),
     )
    : SizedBox(
      width: 16,
      height: 16,
      child:
        CircularProgressIndicator(
       color: Colors.white,
       strokeWidth: 1.5,
      ),
     ),
  style: OutlinedButton.styleFrom(
    backgroundColor:
      Colors.purple[800]),
 ),
],
//หน้าต่างสีเทา
title: Text("แก้ไขข้อมูล"),
```



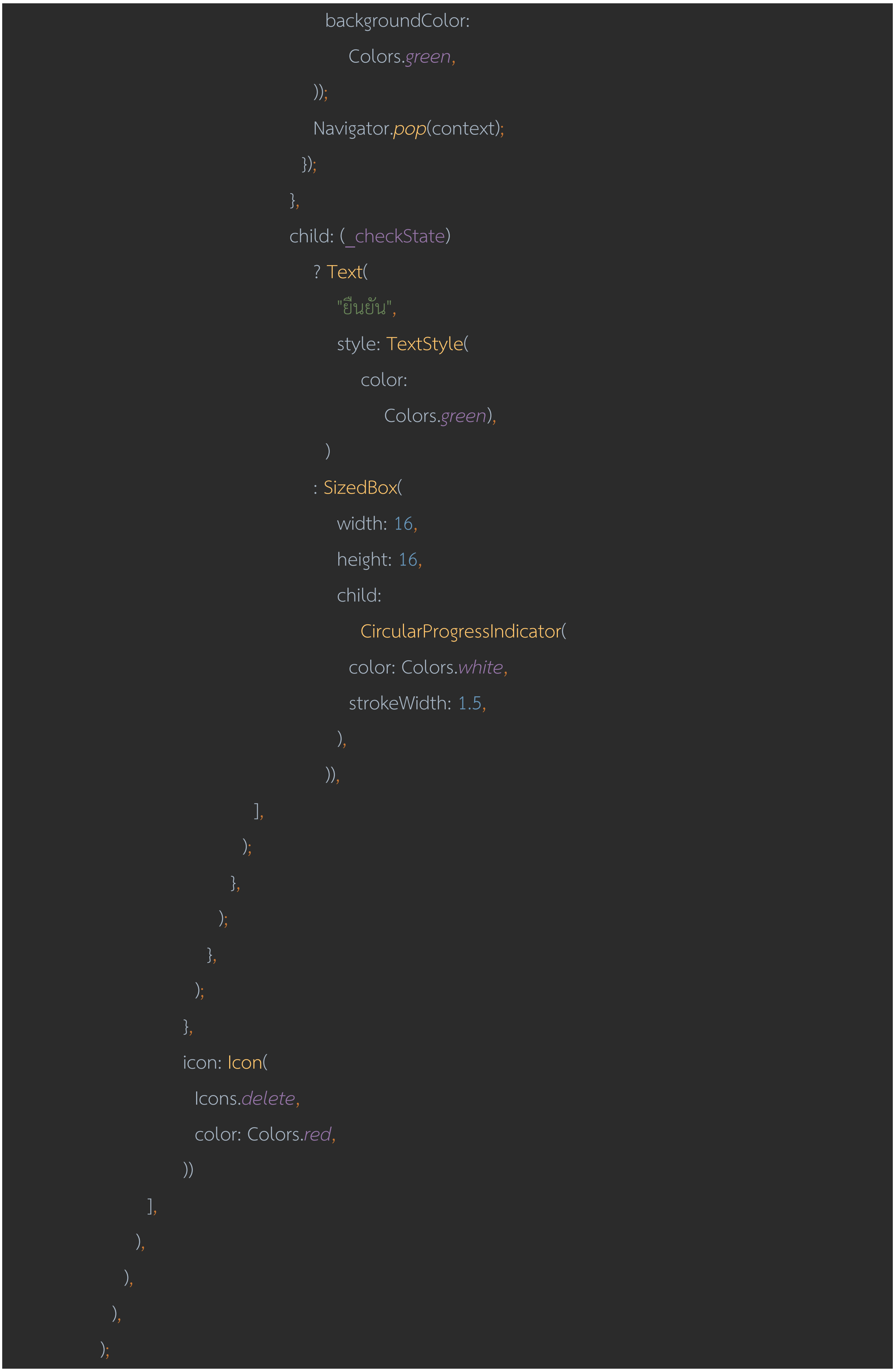
```
content: Column(
  mainAxisSize: MainAxisSize
      .min, //ขนาดเท่ากับฟอร์ม
  children: [
    TextField(
      controller: _topichead,
      decoration: InputDecoration(
        icon: Icon(
            Icons.question_answer,
            color:
                Colors.purple[800]),
        labelText:
            "ระบุคำถามของคุณ ?",
        border: OutlineInputBorder(
            borderRadius:
                BorderRadius.circular(
                    40)),
      ),
    ),
    SizedBox(
      height: 10,
    ),
    TextField(
      controller: _topicdetail,
      decoration: InputDecoration(
        icon: Icon(Icons.text_snippet,
            color:
                Colors.purple[800]),
        labelText:
            "เขียนรายละเอียดของคำถาม",
        border: OutlineInputBorder(
            borderRadius:
                BorderRadius.circular(
                    40)),
        contentPadding:
            EdgeInsets.symmetric(
                vertical: 50,
                horizontal:
```



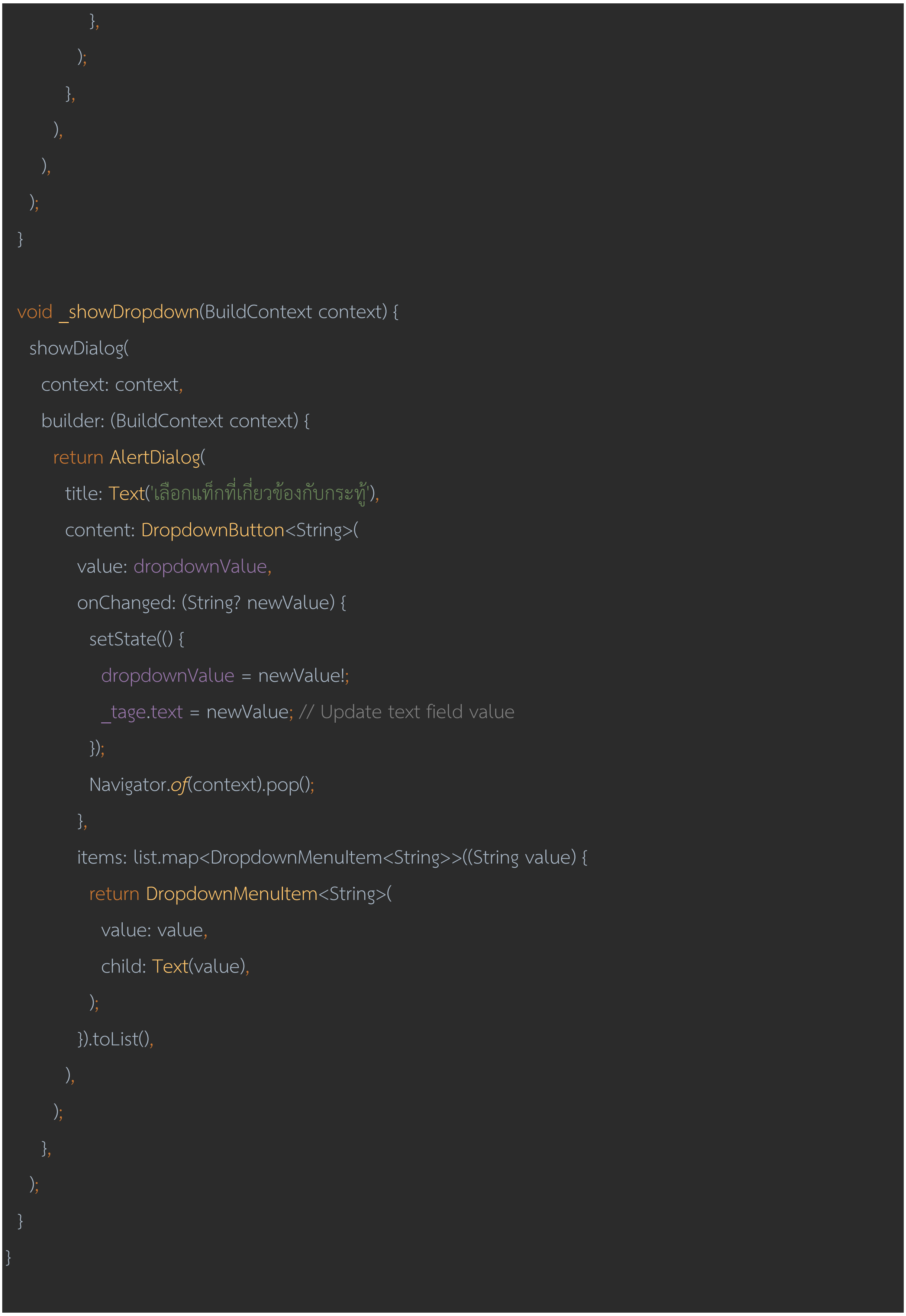
```
                                20), // กำหนดขนาดพื้นที่รอบข้อความ
                    helperText:
                        "* พิมพ์ข้อความได้ไม่เกิน 10,000 ตัวอักษร (0 / 10000) ",
                  ),
                ),
                SizedBox(
                  height: 10,
                ),
                TextField(
                  controller: _tage,
                  decoration: InputDecoration(
                    icon: Icon(Icons.tag,
                        color:
                            Colors.purple[800]),
                    suffixIcon: Icon(
                        Icons.arrow_drop_down),
                    labelText:
                        "เลือกแท็กที่เกี่ยวข้องกับกระทู้",
                    border: OutlineInputBorder(
                        borderRadius:
                            BorderRadius.circular(
                                40)),
                  ),
                  onTap: () {
                    _showDropdown(context);
                  },
                  readOnly: true,
                ),
              ],
            ),
          );
        },
      );
    },
  );
},
icon: Icon(
  Icons.edit,
```



```
   color: Colors.purple[800],
  )),
IconButton(
  onPressed: () {
   showDialog(
    context: context,
    builder: (BuildContext context) {
     return StatefulBuilder(
      builder: (BuildContext context,
        void Function(void Function())
          setState) {
       return AlertDialog(
        title: Text("คุณต้องการลบหรือไม่ ?"),
        content: Text(" "),
        actions: [
         TextButton(
           onPressed: () {
            Navigator.pop(context);
           },
           child: Text(
            "ยกเลิก",
            style: TextStyle(
              color: Colors.red),
           )),
         TextButton(
           onPressed: () async {
            setState(() {
             _checkState = false;
            });
            await topicCollection
              .doc(topic.id)
              .delete()
              .then((value) {
             ScaffoldMessenger.of(
                 context)
               .showSnackBar(SnackBar(
              content: Text(
                "ลบข้อมูลสำเร็จ"),
```



```
                   backgroundColor:
                     Colors.green,
                  ));
                  Navigator.pop(context);
                 });
                },
                child: (_checkState)
                  ? Text(
                    "ยืนยัน",
                    style: TextStyle(
                      color:
                        Colors.green),
                   )
                  : SizedBox(
                    width: 16,
                    height: 16,
                    child:
                      CircularProgressIndicator(
                     color: Colors.white,
                     strokeWidth: 1.5,
                    ),
                   )),
             ],
            );
           },
          );
         },
        );
       },
       icon: Icon(
        Icons.delete,
        color: Colors.red,
       ))
    ],
   ),
  ),
 ),
);
```



```
              },
            );
          },
        ),
      ),
    );
  }

  void _showDropdown(BuildContext context) {
    showDialog(
      context: context,
      builder: (BuildContext context) {
        return AlertDialog(
          title: Text('เลือกแท็กที่เกี่ยวข้องกับกระทู้'),
          content: DropdownButton<String>(
            value: dropdownValue,
            onChanged: (String? newValue) {
              setState(() {
                dropdownValue = newValue!;
                _tage.text = newValue; // Update text field value
              });
              Navigator.of(context).pop();
            },
            items: list.map<DropdownMenuItem<String>>((String value) {
              return DropdownMenuItem<String>(
                value: value,
                child: Text(value),
              );
            }).toList(),
          ),
        );
      },
    );
  }
}
```