

## GROUP MEMBERS

Name	Registration Number
Athapaththu A.H.M.C.P. (Leader)	IT20252090
Kumarasiri R.D.S.R.	IT19003160
Herath A.M.T.N.	IT20015558
Silva H.C.M.	IT20263058

Link to the Repo: → <https://github.com/priyankara96/Kamatha>

Video Link: → [https://mysliit-my.sharepoint.com/:v/g/personal/it20252090\\_my\\_sliit\\_lk/ESxhEycvN3xEiLKrYaPCHiQBYEXbsdjQLLs9WJtZDw10w?e=XNgzJt](https://mysliit-my.sharepoint.com/:v/g/personal/it20252090_my_sliit_lk/ESxhEycvN3xEiLKrYaPCHiQBYEXbsdjQLLs9WJtZDw10w?e=XNgzJt)

### Crop Advisory (User)

In this component, User can log in to the system and add their Questions about agriculture. Other users can see that questions and give their answers and likes that questions. There is a tab called Questionnaire in the Drawer view. User can edit / delete their questions and answers.

#### CODE SNIPPETS

##### Screens

###### AddQuestion.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/widgets/add_item_form.dart';
import 'package:Kamataha/src/widgets/add_question_form.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class AddQuestions extends StatefulWidget {
  const AddQuestions({Key? key, required User user})
      : _user = user,
        super(key: key);

  final User _user;

  @override
  State<AddQuestions> createState() => _AddQuestionsState();
}

class _AddQuestionsState extends State<AddQuestions> {
  final FocusNode _titleFocusNode = FocusNode();
  final FocusNode _questionFocusNode = FocusNode();

  late User _user;

  @override
  void initState() {
```

```
_user = widget._user;

super.initState();
}

@Override
Widget build(BuildContext context) {
    return GestureDetector(
        onTap: () {
            _titleFocusNode.unfocus();
            _questionFocusNode.unfocus();
        },
        child: Scaffold(
            backgroundColor: CustomColors.firebaseioNavy,
            appBar: AppBar(
                elevation: 0,
                backgroundColor: CustomColors.firebaseioNavy,
                title: AppBarTitle(),
                centerTitle: true,
            ),
            body: SafeArea(
                child: ListView(children: [
                    Padding(
                        padding: const EdgeInsets.only(
                            left: 16.0,
                            right: 16.0,
                            bottom: 20.0,
                        ),
                        child: AddQuestionForm(
                            titleFocusNode: _titleFocusNode,
                            questionFocusNode: _questionFocusNode,
                            user: _user,
                        ),
                    ),
                    ],
                ]),
            );
    }
}
```

## AnswerScreen.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/widgets/answerList.dart';
import 'package:Kamataha/src/widgets/list_comments.dart';
import 'package:flutter/material.dart';

class AnswerScreen extends StatefulWidget {
    const AnswerScreen({Key? key, required this.answer}) : super(key: key);

    final String answer;

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

class _AnswerScreenState extends State<AnswerScreen> {
    final String time = DateTime.now().toString().substring(0, 4);

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            backgroundColor: CustomColors.firebaseioNavy,
            appBar: AppBar(
                elevation: 0,
                backgroundColor: CustomColors.firebaseioNavy,
                title: Row(
                    mainAxisAlignment: MainAxisAlignment.start,
                    mainAxisSize: MainAxisSize.min,
                    children: [
                        const SizedBox(width: 8),
                        Text(
                            'Answers for : ',
                            style: TextStyle(
                                color: CustomColors.firebaseioYellow,
                                fontSize: 18,
                            ),
                        ),
                        Text(
                            widget.answer,
                            style: TextStyle(
                                color: CustomColors.firebaseioOrange,
                                fontSize: 18,
                            ),
                        ),
                    ],
                ),
            ),
        );
    }
}
```

```

        ],
    ),
    centerTitle: true,
    // leading: IconButton(
    //   icon: const Icon(Icons.menu),
    //   onPressed: () => {_buildDrawer(context)},
    // ),
),
body: SafeArea(
    child: Padding(
        padding: const EdgeInsets.only(
            left: 16.0,
            right: 16.0,
            bottom: 20.0,
        ),
        child: AnswerList(
            title: widget.answer,
        ),
    ),
),
bottomNavigationBar: Padding(
    padding: const EdgeInsets.only(left: 90, bottom: 5),
    child: Text(
        "Copyright " + time + " @□" + "Kamatha",
        style: TextStyle(color: CustomColors.firebaseioOrange),
    ),
),
);
}
}
}

```

## Dashboard.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/posts.dart';
import 'package:Kamataha/src/screens/questions.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:Kamataha/src/widgets/item_list.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'add_screen.dart';

```

```
class Dashboard extends StatefulWidget {
  const Dashboard({Key? key, required User user})
      : _user = user,
        super(key: key);

  final User _user;

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

class _DashboardState extends State<Dashboard> {
  late User _user;

  @override
  void initState() {
    _user = widget._user;
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return DefaultTabController(
      length: 2,
      child: Scaffold(
        backgroundColor: CustomColors.firebaseioNavy,
        appBar: AppBar(
          title: Row(
            mainAxisAlignment: MainAxisAlignment.start,
            mainAxisSize: MainAxisSize.min,
          ),
          backgroundColor: CustomColors.firebaseioNavy,
          elevation: 0,
          centerTitle: true,
          bottom: const TabBar(
            tabs: [
              Tab(
                text: "Posts",
                icon: Icon(Icons.favorite),
              ),
              Tab(
                text: "Questions",
                icon: Icon(Icons.question_answer),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

```

        )),
    body: TabBarView(
        children: [
            Posts(user: _user),
            Questions(user: _user)
        ],
    ),
),
);
}
}

```

### DisplayAllQuestion.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/answer_screen.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:Kamataha/src/widgets/text_input_answer_widget.dart';
import 'package:Kamataha/src/widgets/text_input_widget.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';

class DisplayQuestions extends StatefulWidget {
    final String currentTitle;
    final String currentQuestion;
    final String name;
    final String documentId;
    final String time;
    final int likes;
    final User _user;

    const DisplayQuestions(
        {Key? key,
        required this.currentTitle,
        required this.currentQuestion,
        required this.name,
        required this.documentId,
        required this.time,
        required this.likes,
        required User user})

```

```
: _user = user,
super(key: key);

@Override
_DisplayQuestionsState createState() => _DisplayQuestionsState();
}

class _DisplayQuestionsState extends State<DisplayQuestions> {
late User _user;

var isPressed = false;

@Override
void initState() {
    _user = widget._user;
    super.initState();
}

void showToast(isPressed) {
    if (isPressed) {
        Fluttertoast.showToast(
            msg: 'You Liked This',
            toastLength: Toast.LENGTH_SHORT,
            gravity: ToastGravity.BOTTOM,
            timeInSecForIosWeb: 1,
            backgroundColor: Colors.red,
            textColor: Colors.yellow);
    } else {
        Fluttertoast.showToast(
            msg: 'Like Removed',
            toastLength: Toast.LENGTH_SHORT,
            gravity: ToastGravity.BOTTOM,
            timeInSecForIosWeb: 1,
            backgroundColor: Colors.red,
            textColor: Colors.yellow);
    }
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: CustomColors.firebaseioNavy,
        appBar: AppBar(
            elevation: 0,
            backgroundColor: CustomColors.firebaseioNavy,
```

```
title: AppBarTitle(),
centerTitle: true,
),
floatingActionButton: Padding(
padding: const EdgeInsets.only(bottom: 50),
child: FloatingActionButton(
backgroundColor: onPressed ? Colors.red : Colors.green,
onPressed: () {
setState(() => onPressed = !onPressed);
showToast(onPressed);
if (onPressed) {
Database.updateLikesQ(
likes: widget.likes, docId: widget.documentId);
} else {
if (widget.likes == 0) {
return;
}
Database.decrementLikesQ(
likes: widget.likes, docId: widget.documentId);
}
},
child: const Icon(Icons.favorite),
),
),
body: SafeArea(
child: ListView(
children: [
Column(
children: [
Center(
child: Padding(
padding: const EdgeInsets.all(8.0),
child: Text(
"Title ↗ " + widget.currentTitle,
style: const TextStyle(
color: Colors.white,
fontSize: 20,
),
),
),
),
),
),
],
),
Center(
child: Padding(
padding: const EdgeInsets.all(8.0),
child: Text(

```

```
"Posted by " + widget.name,
style: const TextStyle(
    color: Colors.white,
    fontSize: 10,
),
),
),
),
),
),
Center(
child: Padding(
padding: const EdgeInsets.all(8.0),
child: Text(
widget.time + " ❤️ " + widget.likes.toString(),
style: const TextStyle(
color: Colors.white,
fontSize: 10,
),
),
),
),
),
),
),
Center(
child: Padding(
padding: const EdgeInsets.all(16.0),
child: SelectableText(
widget.currentQuestion,
style: const TextStyle(
color: Colors.white,
fontSize: 20,
letterSpacing: 1,
wordSpacing: 1),
),
),
),
),
),
ElevatedButton(
onPressed: () {
Navigator.of(context).push(MaterialPageRoute(
builder: (context) => AnswerScreen(
answer: widget.currentTitle,
)));
},
),
child: const Text("See Answers"),
style: ButtonStyle(
backgroundColor: MaterialStateProperty.all(
CustomColors.firebaseioOrange)),
),
```

```
        TextInputAnswerWidget(user: _user, title: widget.currentTitle)
    ],
),
],
),
),
);
}
}
```

## EditAnswerScreen.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/widgets/edit_answer_form.dart';
import 'package:Kamataha/src/widgets/edit_comment_form.dart';
import 'package:flutter/material.dart';

class EditAnswerScreen extends StatefulWidget {
    final String currentTitle;
    final String answer;
    final String documentId;
    final String user;

    // ignore: use_key_in_widget_constructors
    const EditAnswerScreen(
        {required this.currentTitle,
        required this.answer,
        required this.documentId,
        required this.user});

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

class _EditAnswerScreenState extends State<EditAnswerScreen> {
    final FocusNode _titleFocusNode = FocusNode();

    final FocusNode _descriptionFocusNode = FocusNode();

    bool _isDeleting = false;
```

```
@override
Widget build(BuildContext context) {
  return GestureDetector(
    onTap: () {
      _titleFocusNode.unfocus();
      _descriptionFocusNode.unfocus();
    },
    child: Scaffold(
      backgroundColor: CustomColors.firebaseioNavy,
      appBar: AppBar(
        elevation: 0,
        backgroundColor: CustomColors.firebaseioNavy,
        title: Row(
          mainAxisAlignment: MainAxisAlignment.start,
          mainAxisSize: MainAxisSize.min,
          children: [
            const SizedBox(width: 8),
            Text(
              widget.user.toString().substring(0, 10) + "...'s",
              style: TextStyle(
                color: CustomColors.firebaseioYellow,
                fontSize: 18,
              ),
            ),
            Text(
              ' answer ',
              style: TextStyle(
                color: CustomColors.firebaseioOrange,
                fontSize: 18,
              ),
            ),
          ],
        ),
        centerTitle: true,
        actions: [
          _isDeleting
            ? const Padding(
                padding: EdgeInsets.only(
                  top: 10.0,
                  bottom: 10.0,
                  right: 16.0,
                ),
            ),
            child: CircularProgressIndicator(
              valueColor: AlwaysStoppedAnimation<Color>(
                Colors.redAccent,
              ),
            ),
        ],
      ),
    ),
  );
}
```

```
        ),
        strokeWidth: 3,
    ),
)
: IconButton(
    icon: const Icon(
        Icons.delete,
        color: Colors.redAccent,
        size: 32,
    ),
onPressed: () async {
    setState(() {
        _isDeleting = true;
    });
    await Database.deleteAnswer(
        docId: widget.documentId,
    );
    setState(() {
        _isDeleting = false;
    });
    Navigator.of(context).pop();
},
),
],
),
body: SafeArea(
    child: Padding(
        padding: const EdgeInsets.only(
            left: 16.0,
            right: 16.0,
            bottom: 20.0,
        ),
        child: EditAnswerForm(
            documentId: widget.documentId,
            titleFocusNode: _titleFocusNode,
            descriptionFocusNode: _descriptionFocusNode,
            currentTitle: widget.currentTitle,
            answer: widget.answer,
        ),
    ),
),
),
),
```

```
    );
}
}
```

## EditQuestionScreen.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/widgets/edit_item_form.dart';
import 'package:Kamataha/src/widgets/edit_question_form.dart';
import 'package:flutter/material.dart';

class EditQuestionScreen extends StatefulWidget {
    final String currentTitle;
    final String currentQuestions;
    final String documentId;
    final String user;

    // ignore: use_key_in_widget_constructors
    const EditQuestionScreen(
        {required this.currentTitle,
        required this.currentQuestions,
        required this.documentId,
        required this.user});

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

class _EditQuestionScreenState extends State<EditQuestionScreen> {
    final FocusNode _titleFocusNode = FocusNode();

    final FocusNode _descriptionFocusNode = FocusNode();

    bool _isDeleting = false;

    @override
    Widget build(BuildContext context) {
        return GestureDetector(
            onTap: () {
                _titleFocusNode.unfocus();
                _descriptionFocusNode.unfocus();
            }
        );
    }
}
```

```
},
child: Scaffold(
  backgroundColor: CustomColors.firebaseioNavy,
  appBar: AppBar(
    elevation: 0,
    backgroundColor: CustomColors.firebaseioNavy,
    title: Row(
      mainAxisAlignment: MainAxisAlignment.start,
      mainAxisSize: MainAxisSize.min,
      children: [
        const SizedBox(width: 8),
        Text(
          widget.user.toString().substring(0, 10) + "...'s",
          style: TextStyle(
            color: CustomColors.firebaseioYellow,
            fontSize: 18,
          ),
        ),
        Text(
          ' Question ',
          style: TextStyle(
            color: CustomColors.firebaseioOrange,
            fontSize: 18,
          ),
        ),
      ],
    ),
    centerTitle: true,
    actions: [
      _isDeleting
        ? const Padding(
            padding: EdgeInsets.only(
              top: 10.0,
              bottom: 10.0,
              right: 16.0,
            ),
            child: CircularProgressIndicator(
              valueColor: AlwaysStoppedAnimation<Color>(
                Colors.redAccent,
              ),
              strokeWidth: 3,
            ),
          )
        : IconButton(
            icon: const Icon(
```

```
        Icons.delete,
        color: Colors.redAccent,
        size: 32,
    ),
    onPressed: () async {
        setState(() {
            _isDeleting = true;
        });

        await Database.deleteQuestion(
            docId: widget.documentId,
        );

        setState(() {
            _isDeleting = false;
        });

        Navigator.of(context).pop();
    },
),
],
),
),
body: SafeArea(
    child: Padding(
        padding: const EdgeInsets.only(
            left: 16.0,
            right: 16.0,
            bottom: 20.0,
        ),
        child: EditQuestionForm(
            documentId: widget.documentId,
            titleFocusNode: _titleFocusNode,
            descriptionFocusNode: _descriptionFocusNode,
            currentTitle: widget.currentTitle,
            currentQuestion: widget.currentQuestions,
        ),
        ),
        ),
        ),
        ),
    );
}
}
```

### PrivateAnswer.dart

```
import 'package:Kamataha/src/widgets/private_answer_list.dart';
import 'package:Kamataha/src/widgets/private_comment_list.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class PrivateAnswers extends StatefulWidget {
    const PrivateAnswers({Key? key, required User user})
        : _user = user,
        super(key: key);

    final User _user;

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

class _PrivateAnswersState extends State<PrivateAnswers> {
    late User _user;

    @override
    void initState() {
        _user = widget._user;
        super.initState();
    }

    @override
    Widget build(BuildContext context) {
        return SafeArea(
            child: Padding(
                padding: const EdgeInsets.only(
                    left: 16.0,
                    right: 16.0,
                    bottom: 20.0,
                ),
                child: PrivateAnswerList(
                    user: _user,
                ),
            ),
        );
    }
}
```

## PrivateQuestionAnswer.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/add_question.dart';
import 'package:Kamataha/src/screens/private_answer.dart';
import 'package:Kamataha/src/screens/private_comments.dart';
import 'package:Kamataha/src/screens/private_feelings.dart';
import 'package:Kamataha/src/screens/private_questions.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'add_screen.dart';

class PrivateQuestionScreen extends StatefulWidget {
  const PrivateQuestionScreen({Key? key, required User user})
      : _user = user,
        super(key: key);

  final User _user;

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

class _PrivateQuestionScreenState extends State<PrivateQuestionScreen> {
  late User _user;

  @override
  void initState() {
    _user = widget._user;
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return DefaultTabController(
      length: 2,
      child: Scaffold(
        backgroundColor: CustomColors.firebaseioNavy,
        appBar: AppBar(
          title: Row(
            mainAxisAlignment: MainAxisAlignment.start,
            mainAxisSize: MainAxisSize.min,
            children: [
              const SizedBox(width: 8),
              Text(

```

```
        _user.displayName.toString().substring(0, 10) + "...'s",
        style: TextStyle(
            color: CustomColors.firebaseioYellow,
            fontSize: 18,
        ),
    ),
    Text(
        ' Answer ',
        style: TextStyle(
            color: CustomColors.firebaseioOrange,
            fontSize: 18,
        ),
    ),
),
],
),
backgroundColor: CustomColors.firebaseioNavy,
elevation: 0,
centerTitle: true,
bottom: const TabBar(
    tabs: [
        Tab(
            text: "Questions",
            icon: Icon(Icons.favorite),
        ),
        Tab(
            text: "Answers",
            icon: Icon(Icons.comment),
        ),
    ],
)),
floatingActionButton: FloatingActionButton(
    child: const Icon(Icons.add),
    backgroundColor: CustomColors.firebaseioOrange,
    onPressed: () {
        Navigator.of(context).push(
            MaterialPageRoute(
                builder: (context) => AddQuestions(user: _user),
            ),
        );
    },
),
body: TabBarView(
    children: [
        PrivateQuestion(user: _user),
        PrivateAnswers(user: _user),
    ],
)
```

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

### PrivateQuestions.dart

```
import 'package:Kamataha/src/widgets/private_item_list.dart';
import 'package:Kamataha/src/widgets/private_question_list.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class PrivateQuestion extends StatefulWidget {
  const PrivateQuestion({Key? key, required User user})
      : _user = user,
        super(key: key);

  final User _user;

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

class _PrivateQuestionState extends State<PrivateQuestion> {
  late User _user;

  @override
  void initState() {
    _user = widget._user;
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Padding(
        padding: const EdgeInsets.only(
          left: 16.0,
          right: 16.0,
          bottom: 20.0,
        ),
      ),
    );
  }
}
```

```
        ),
        child: PrivateQuestionList(
            user: _user,
        ),
    );
}
}
```

## Widgets

### AddQuestionForm.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'custom_form_field.dart';

// ignore: must_be_immutable
class AddQuestionForm extends StatefulWidget {
  final FocusNode titleFocusNode;
  final FocusNode questionFocusNode;

  const AddQuestionForm({
    Key? key,
    required this.titleFocusNode,
    required this.questionFocusNode,
    required User user,
  }) : _user = user,
        super(key: key);

  final User _user;

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

class _AddQuestionFormState extends State<AddQuestionForm> {
  final _addQuestionFormKey = GlobalKey<FormState>();

  late User _user;

  bool _isProcessing = false;

  final TextEditingController _titleController = TextEditingController();
  final TextEditingController _questionController = TextEditingController();

  @override
  void initState() {
    _user = widget._user;
    super.initState();
  }
}
```

```
@override
Widget build(BuildContext context) {
  return Form(
    key: _addQuestionFormKey,
    child: Column(
      children: [
        Padding(
          padding: const EdgeInsets.only(
            left: 8.0,
            right: 8.0,
            bottom: 24.0,
          ),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              const SizedBox(height: 24.0),
              Text(
                'Title',
                style: TextStyle(
                  color: CustomColors.firebaseioGrey,
                  fontSize: 22.0,
                  letterSpacing: 1,
                  fontWeight: FontWeight.bold,
                ),
              ),
              const SizedBox(height: 8.0),
              CustomFormField(
                isLabelEnabled: false,
                controller: _titleController,
                focusNode: widget.titleFocusNode,
                keyboardType: TextInputType.text,
                inputAction: TextInputAction.next,
                validator: (value) => Validator.validateField(
                  value: value,
                ),
                label: 'Title',
                hint: 'Enter your title',
              ),
              const SizedBox(height: 24.0),
              Text(
                'Your Questions',
                style: TextStyle(
                  color: CustomColors.firebaseioGrey,
                  fontSize: 22.0,
                  letterSpacing: 1,
```

```
        fontWeight: FontWeight.bold,  
    ),  
),  
const SizedBox(height: 8.0),  
CustomFormField(  
    maxLines: 10,  
    isLabelEnabled: false,  
    controller: _questionController,  
    focusNode: widget.questionFocusNode,  
    keyboardType: TextInputType.text,  
    inputAction: TextInputAction.done,  
    validator: (value) => Validator.validateField(  
        value: value,  
    ),  
    label: 'Your Questions',  
    hint: 'Enter your Questions',  
),  
],  
),  
),  
_isProcessing  
? Padding(  
    padding: const EdgeInsets.all(16.0),  
    child: CircularProgressIndicator(  
        valueColor: AlwaysStoppedAnimation<Color>(  
            CustomColors.firebaseioOrange,  
        ),  
    ),  
),  
:  
ElevatedButton(  
    style: ButtonStyle(  
        backgroundColor: MaterialStateProperty.all(  
            CustomColors.firebaseioOrange,  
        ),  
        shape: MaterialStateProperty.all(  
            RoundedRectangleBorder(  
                borderRadius: BorderRadius.circular(45),  
            ),  
        ),  
    ),  
    onPressed: () async {  
        widget.titleFocusNode.unfocus();  
        widget.questionFocusNode.unfocus();  
        if ( addQuestionFormKey.currentState!.validate() ) {  
            addQuestionFormKey.currentState!.submitForm();  
            setState(() {  
                _isProcessing = true;  
            });  
            FirebaseFirestore.instance.collection('questions').add(  
                {  
                    'question': questionController.text,  
                    'date': DateTime.now().toString(),  
                }  
            );  
            questionController.clear();  
            titleController.clear();  
            Navigator.pop(context);  
        }  
    },  
),
```

```

        setState(() {
            _isProcessing = true;
        });

        await Database.addQuestion(
            title: _titleController.text,
            question: _questionController.text,
            user: _user.displayName.toString());

        setState(() {
            _isProcessing = false;
        });

        Navigator.of(context).pop();
    }
},
child: Padding(
    padding: const EdgeInsets.only(top: 16.0, bottom: 16.0),
    child: Text(
        'Send',
        style: TextStyle(
            fontSize: 24,
            fontWeight: FontWeight.bold,
            color: CustomColors.firebaseioGrey,
            letterSpacing: 2,
        ),
        ),
        ),
        ),
        ),
        ],
),
);
}
}

```

### **AnswerList.dart**

```

// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

```

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

class AnswerList extends StatefulWidget {
  const AnswerList({required this.title});
  final String title;
  @override
  State<AnswerList> createState() => _AnswerListState();
}

class _AnswerListState extends State<AnswerList> {
  //ADDED
  var colors = [
    Colors.red,
    Colors.blue,
    Colors.cyan,
    Colors.green,
    Colors.yellow,
  ];
  @override
  Widget build(BuildContext context) {
    return StreamBuilder<QuerySnapshot>(
      stream: Database.readAnswers(widget.title),
      builder: (context, snapshot) {
        if (snapshot.hasError) {
          return const Text('Something went wrong');
        } else if (snapshot.hasData || snapshot.data != null) {
          if (snapshot.data!.docs.isEmpty) {
            return Center(
              child: Text(
                "No answers yet",
                style: TextStyle(color: CustomColors.firebaseioGrey),
              ),
            );
          }
        }
        return ListView.separated(
          separatorBuilder: (context, index) => const SizedBox(height: 16.0),
          itemCount: snapshot.data!.docs.length,
          itemBuilder: (context, index) {
            var noteInfo = snapshot.data!.docs[index].data() as dynamic;
            String user = noteInfo["user"];
            String answer = noteInfo['answer'];
            String time = noteInfo['time'];

            return Ink(
              decoration: BoxDecoration(

```

```
        color: colors[index],
        borderRadius: BorderRadius.circular(8.0),
    ),
    child: ListTile(
        shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(8.0),
        ),
        title: Text(
            user + " says :",
            maxLines: 1,
            overflow: TextOverflow.ellipsis,
            style: TextStyle(color: CustomColors.firebaseioGrey),
        ),
        subtitle: Column(
            children: [
                Text(
                    answer,
                    maxLines: 2,
                    overflow: TextOverflow.ellipsis,
                    style: const TextStyle(
                        color: Colors.white,
                        fontSize: 20,
                    ),
                ),
                const Text(""),
                Text(
                    time.substring(0, 20),
                    maxLines: 1,
                    overflow: TextOverflow.ellipsis,
                    style: const TextStyle(fontSize: 10),
                ),
            ],
        ),
    );
},
);
};

}

return Center(
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
    ),
),
```

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

### EditAnswerForm.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:flutter/material.dart';

import 'custom_form_field.dart';

class EditAnswerForm extends StatefulWidget {
    final FocusNode titleFocusNode;
    final FocusNode descriptionFocusNode;
    final String currentTitle;
    final String answer;
    final String documentId;

    const EditAnswerForm({
        required this.titleFocusNode,
        required this.descriptionFocusNode,
        required this.currentTitle,
        required this.answer,
        required this.documentId,
    });

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

class _EditAnswerFormState extends State<EditAnswerForm> {
    final _EditAnswerFormKey = GlobalKey<FormState>();

    bool _isProcessing = false;

    late TextEditingController _titleController;
    late TextEditingController _descriptionController;
```



```
        color: Colors.white,
    ),
),
const SizedBox(height: 24.0),
Text(
    'answer',
    style: TextStyle(
        color: CustomColors.firebaseioGrey,
        fontSize: 22.0,
        letterSpacing: 1,
        fontWeight: FontWeight.bold,
    ),
),
const SizedBox(height: 8.0),
CustomFormField(
    maxLines: 10,
    isLabelEnabled: false,
    controller: _descriptionController,
    focusNode: widget.descriptionFocusNode,
    keyboardType: TextInputType.text,
    inputAction: TextInputAction.done,
    validator: (value) => Validator.validateField(
        value: value,
    ),
    label: 'answer',
    hint: 'Enter your note description',
),
],
),
),
),
_isProcessing
? Padding(
    padding: const EdgeInsets.all(16.0),
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
    ),
)
: Padding(
    padding: const EdgeInsets.only(
        left: 30,
        right: 30,
    ),
    child: ElevatedButton(

```

```
style: ButtonStyle(
    backgroundColor: MaterialStateProperty.all(
        CustomColors.firebaseioOrange,
    ),
    minimumSize: MaterialStateProperty.all(Size.zero),
    shape: MaterialStateProperty.all(
        RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(45),
        ),
    ),
),
),
),
),
onPressed: () async {
    widget.titleFocusNode.unfocus();
    widget.descriptionFocusNode.unfocus();

    if (_EditAnswerFormKey.currentState!.validate()) {
        setState(() {
            _isProcessing = true;
        });

        await Database.updateAnsrs(
            docId: widget.documentId,
            answer: _descriptionController.text,
        );

        setState(() {
            _isProcessing = false;
        });

        Navigator.of(context).pop();
    }
},
),
child: Padding(
    padding: const EdgeInsets.only(top: 16.0, bottom: 16.0),
    child: Text(
        'Update answer',
        style: TextStyle(
            fontSize: 24,
            fontWeight: FontWeight.bold,
            color: CustomColors.firebaseioGrey,
            letterSpacing: 2,
        ),
    ),
),
),
),
),
```

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

### EditQuestionForm.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:flutter/material.dart';
import 'custom_form_field.dart';

class EditQuestionForm extends StatefulWidget {
  final FocusNode titleFocusNode;
  final FocusNode descriptionFocusNode;
  final String currentTitle;
  final String currentQuestion;
  final String documentId;

  const EditQuestionForm({
    required this.titleFocusNode,
    required this.descriptionFocusNode,
    required this.currentTitle,
    required this.currentQuestion,
    required this.documentId,
  });

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

class _EditQuestionFormState extends State<EditQuestionForm> {
  final _EditQuestionFormKey = GlobalKey<FormState>();

  bool _isProcessing = false;

  late TextEditingController _titleController;
  late TextEditingController _descriptionController;
```

```
@override
void initState() {
    _titleController = TextEditingController(
        text: widget.currentTitle,
    );

    _descriptionController = TextEditingController(
        text: widget.currentQuestion,
    );
    super.initState();
}

@Override
Widget build(BuildContext context) {
    return Form(
        key: _EditQuestionFormKey,
        child: ListView(
            children: [
                Padding(
                    padding: const EdgeInsets.only(
                        left: 8.0,
                        right: 8.0,
                        bottom: 24.0,
                    ),
                    child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                            const SizedBox(height: 24.0),
                            Text(
                                'Title',
                                style: TextStyle(
                                    color: CustomColors.firebaseioGrey,
                                    fontSize: 22.0,
                                    letterSpacing: 1,
                                    fontWeight: FontWeight.bold,
                                ),
                            ),
                            const SizedBox(height: 8.0),
                            CustomFormField(
                                isLabelEnabled: false,
                                controller: _titleController,
                                focusNode: widget.titleFocusNode,
                                keyboardType: TextInputType.text,
                                inputAction: TextInputAction.next,
                                validator: (value) => Validator.validateField(

```

```
        value: value,
    ),
    label: 'Title',
    hint: 'Enter your note title',
),
const SizedBox(height: 24.0),
Text(
    'Question',
    style: TextStyle(
        color: CustomColors.firebaseioGrey,
        fontSize: 22.0,
        letterSpacing: 1,
        fontWeight: FontWeight.bold,
    ),
),
const SizedBox(height: 8.0),
CustomFormField(
    maxLines: 10,
    isLabelEnabled: false,
    controller: _descriptionController,
    focusNode: widget.descriptionFocusNode,
    keyboardType: TextInputType.text,
    inputAction: TextInputAction.done,
    validator: (value) => Validator.validateField(
        value: value,
    ),
    label: 'Question',
    hint: 'Enter your Question',
),
],
),
),
),
_isProcessing
? Padding(
    padding: const EdgeInsets.all(16.0),
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
    ),
)
: Padding(
    padding: const EdgeInsets.only(
        left: 30,
        right: 30,
```

```
        ),
        child: ElevatedButton(
            style: ButtonStyle(
                backgroundColor: MaterialStateProperty.all(
                    CustomColors.firebaseioOrange,
                ),
                minimumSize: MaterialStateProperty.all(Size.zero),
                shape: MaterialStateProperty.all(
                    RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(45),
                    ),
                ),
            ),
            onPressed: () async {
                widget.titleFocusNode.unfocus();
                widget.descriptionFocusNode.unfocus();

                if (_EditQuestionFormKey.currentState!.validate()) {
                    setState(() {
                        _isProcessing = true;
                    });

                    await Database.updateQuestion(
                        docId: widget.documentId,
                        title: _titleController.text,
                        question: _descriptionController.text,
                    );

                    setState(() {
                        _isProcessing = false;
                    });

                    Navigator.of(context).pop();
                }
            },
        ),
        child: Padding(
            padding: const EdgeInsets.only(top: 16.0, bottom: 16.0),
            child: Text(
                'Update Question',
                style: TextStyle(
                    fontSize: 24,
                    fontWeight: FontWeight.bold,
                    color: CustomColors.firebaseioGrey,
                    letterSpacing: 2,
                ),
            ),
        ),
    ),
);
```

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

### PrivateAnswerList.dart

```
// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/edit_answer_screen.dart';
import 'package:Kamataha/src/screens/edit_comment_screen.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class PrivateAnswerList extends StatefulWidget {
    const PrivateAnswerList({required User user}) : _user = user;
    final User _user;
    @override
    State<PrivateAnswerList> createState() => _PrivateAnswerListState();
}

class _PrivateAnswerListState extends State<PrivateAnswerList> {
    late User _user;

    @override
    void initState() {
        _user = widget._user;
        super.initState();
    }

    @override
    Widget build(BuildContext context) {
        return StreamBuilder<QuerySnapshot>(
            stream: Database.readPrivateAnswers(_user.displayName),
```

```
builder: (context, snapshot) {
  if (snapshot.hasError) {
    return const Text('Something went wrong');
  } else if (snapshot.hasData || snapshot.data != null) {
    if (snapshot.data!.docs.isEmpty) {
      return Center(
        child: Text(
          "No answers yet",
          style: TextStyle(color: CustomColors.firebaseioGrey),
        ),
      );
    }
    return ListView.separated(
      separatorBuilder: (context, index) => const SizedBox(height: 16.0),
      itemCount: snapshot.data!.docs.length,
      itemBuilder: (context, index) {
        var noteInfo = snapshot.data!.docs[index].data() as dynamic;
        String docID = snapshot.data!.docs[index].id;
        String title = noteInfo["title"];
        String answer = noteInfo['answer'];
        String user = noteInfo['user'];
        String time = noteInfo['time'];

        return Padding(
          padding: const EdgeInsets.all(8.0),
          child: Ink(
            decoration: BoxDecoration(
              color: Color(0xFF9CF29A),
              borderRadius: BorderRadius.circular(8.0),
            ),
            child: ListTile(
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(8.0),
              ),
              onTap: () => Navigator.of(context).push(
                MaterialPageRoute(
                  builder: (context) => EditAnswerScreen(
                    currentTitle: title,
                    answer: answer,
                    documentId: docID,
                    user: user,
                  ),
                ),
              ),
            ),
            title: Text(

```

```
        title,
        maxLines: 1,
        overflow: TextOverflow.ellipsis,
    ),
    subtitle: Column(
        children: [
            Text(
                answer,
                maxLines: 2,
                overflow: TextOverflow.ellipsis,
            ),
            const Text(""),
            Text(
                user + " : " + time.substring(0, 20),
                maxLines: 1,
                overflow: TextOverflow.ellipsis,
                style: const TextStyle(fontSize: 10),
            ),
        ],
    ),
    trailing: const Padding(
        padding: EdgeInsets.only(top: 5),
        child: Icon(Icons.edit),
    ),
),
),
),
),
);
},
),
),
),
);
},
);
}
}

return Center(
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
        ),
        );
);
},
);
},
);
});
```

## PrivateQuestionList.dart

```
// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/edit_question_screen.dart';
import 'package:Kamataha/src/screens/edit_screen.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class PrivateQuestionList extends StatefulWidget {
  const PrivateQuestionList({required User user}) : _user = user;
  final User _user;
  @override
  State<PrivateQuestionList> createState() => _PrivateQuestionListState();
}

class _PrivateQuestionListState extends State<PrivateQuestionList> {
  late User _user;

  @override
  void initState() {
    _user = widget._user;
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return StreamBuilder<QuerySnapshot>(
      stream: Database.readPrivateQuestions(_user.displayName),
      builder: (context, snapshot) {
        if (snapshot.hasError) {
          return const Text('Something went wrong');
        } else if (snapshot.hasData || snapshot.data != null) {
          if (snapshot.data!.docs.isEmpty) {
            return Center(
              child: Text(
                "No questions yet",
                style: TextStyle(color: CustomColors.firebaseioGrey),
              ),
            );
          }
        }
        return ListView.separated(

```

```
separatorBuilder: (context, index) => const SizedBox(height: 16.0),
itemCount: snapshot.data!.docs.length,
itemBuilder: (context, index) {
  var noteInfo = snapshot.data!.docs[index].data() as dynamic;
  String docID = snapshot.data!.docs[index].id;
  String title = noteInfo["title"];
  String question = noteInfo['question'];
  String user = noteInfo['user'];
  String time = noteInfo['time'];
  int likes = noteInfo['likes'];

  return Padding(
    padding: const EdgeInsets.all(8.0),
    child: Ink(
      decoration: BoxDecoration(
        color: Color(0xFF9CF29A),
        borderRadius: BorderRadius.circular(8.0),
      ),
      child: ListTile(
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(8.0),
        ),
        onTap: () => Navigator.of(context).push(
          MaterialPageRoute(
            builder: (context) => EditQuestionScreen(
              currentTitle: title,
              currentQuestions: question,
              documentId: docID,
              user: user,
            ),
          ),
        ),
        title: Text(
          title,
          maxLines: 1,
          overflow: TextOverflow.ellipsis,
        ),
        subtitle: Column(
          children: [
            Text(
              question,
              maxLines: 2,
              overflow: TextOverflow.ellipsis,
            ),
            const Text(""),
          ],
        ),
      ),
    ),
  );
}
```

```
        Text(
            user +
                " : " +
                time.substring(0, 20) +
                " ❤️" +
                likes.toString(),
            maxLines: 1,
            overflow: TextOverflow.ellipsis,
            style: const TextStyle(fontSize: 10),
        ),
    ],
),
),
trailing: const Padding(
    padding: EdgeInsets.only(top: 5),
    child: Icon(Icons.edit),
),
),
),
),
),
);
},
);
}
}

return Center(
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
        ),
),
);
},
);
});
```

## QuestionList.dart

```
// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/display_all_question.dart';
import 'package:Kamataha/src/screens/display_all.dart';
```

```
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class QuestionList extends StatefulWidget {
    const QuestionList({required User user}) : _user = user;
    final User _user;
    @override
    State<QuestionList> createState() => _QuestionListState();
}

class _QuestionListState extends State<QuestionList> {
    late User _user;

    @override
    void initState() {
        _user = widget._user;
        super.initState();
    }

    @override
    Widget build(BuildContext context) {
        return StreamBuilder<QuerySnapshot>(
            stream: Database.readQuestions(),
            builder: (context, snapshot) {
                if (snapshot.hasError) {
                    return const Text('Something went wrong');
                } else if (snapshot.hasData || snapshot.data != null) {
                    if (snapshot.data!.docs.isEmpty) {
                        return Center(
                            child: Text(
                                "No Questions yet",
                                style: TextStyle(color: CustomColors.firebaseioGrey),
                            ),
                        );
                    }
                }
                return ListView.separated(
                    separatorBuilder: (context, index) => const SizedBox(height: 16.0),
                    itemCount: snapshot.data!.docs.length,
                    itemBuilder: (context, index) {
                        var noteInfo = snapshot.data!.docs[index].data() as dynamic;
                        String docID = snapshot.data!.docs[index].id;
                        String title = noteInfo["title"];
                        String question = noteInfo['question'];

```

```
String user = noteInfo['user'];
String time = noteInfo['time'];
int likes = noteInfo['likes'];

return Padding(
    padding: const EdgeInsets.all(8.0),
    child: Ink(
        decoration: BoxDecoration(
            color: Color(0xFF9CF29A),
            borderRadius: BorderRadius.circular(8.0),
        ),
        child: ListTile(
            shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(8.0),
            ),
            onTap: () => Navigator.of(context).push(
                MaterialPageRoute(
                    builder: (context) => DisplayQuestions(
                        currentTitle: title,
                        currentQuestion: question,
                        documentId: docID,
                        name: user,
                        time: time,
                        user: _user,
                        likes: likes,
                    ),
                ),
            ),
        ),
        title: Text(
            title,
            maxLines: 1,
            overflow: TextOverflow.ellipsis,
        ),
        subtitle: Column(
            children: [
                Text(
                    question,
                    maxLines: 2,
                    overflow: TextOverflow.ellipsis,
                ),
                const Text(""),
                Text(
                    user +
                    " : " +
                    time.substring(0, 20) +

```

```
        " ❤️ □ " +
        likes.toString(),
      maxLines: 1,
      overflow: TextOverflow.ellipsis,
      style: const TextStyle(fontSize: 10),
    ),
  ],
),
),
trailing: const Padding(
  padding: EdgeInsets.only(top: 5),
  child: Icon(Icons.arrow_forward_ios_sharp),
),
),
));
},
);
}
}

return Center(
  child: CircularProgressIndicator(
    valueColor: AlwaysStoppedAnimation<Color>(
      CustomColors.firebaseioOrange,
    ),
  ),
);
},
);
});
```

## TextInputAnswerWidget.dart

```
// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/answer_screen.dart';
import 'package:Kamataha/src/screens/comment_screen.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
```

```
// ignore: duplicate_ignore
class TextInputAnswerWidget extends StatefulWidget {
    final _user;
    final String title;

    const TextInputAnswerWidget({required User user, required this.title})
        : _user = user;

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

class _TextInputAnswerWidgetState extends State<TextInputAnswerWidget> {
    late User _user;

    @override
    void initState() {
        _user = widget._user;
        super.initState();
    }

    final _addItemFormKey = GlobalKey<FormState>();

    final TextEditingController _answerController = TextEditingController();

    @override
    Widget build(BuildContext context) {
        return Form(
            key: _addItemFormKey,
            child: TextFormField(
                controller: _answerController,
                style: TextStyle(
                    color: CustomColors.firebaseioYellow,
                ),
                validator: (value) => Validator.validateField(
                    value: _answerController.text,
                ),
                // controller: controller,
                decoration: InputDecoration(
                    prefixIcon: const Icon(Icons.message),
                    labelText: "Type a answer: ✉",
                    labelStyle: TextStyle(
                        color: CustomColors.firebaseioGrey,
                    ),
                    suffixIcon: IconButton(

```

```
        icon: const Icon(Icons.send),
        splashColor: Colors.blue,
        tooltip: "Post answer",
        onPressed: () async {
            if (_addItemFormKey.currentState!.validate()) {
                await Database.addAnswer(
                    user: _user.displayName.toString(),
                    answer: _answerController.text,
                    title: widget.title);
                _answerController.clear();
                Navigator.of(context).push(MaterialPageRoute(
                    builder: (context) =>
                        AnswerScreen(answer: widget.title),
                )));
            }
        },
    )));
}
```

## CRUD OPERATIONS

### AddQuestion

```
static Future<void> addQuestion({  
    required String title,  
    required String question,  
    required String user,  
) async {  
    DocumentReference documentReferencer =  
        _firestore.collection('question').doc();  
  
    Map<String, dynamic> data = <String, dynamic>{  
        "title": title,  
        "question": question,  
        "user": user,  
        "time": time,  
        "likes": 0  
    };  
  
    await documentReferencer  
        .set(data)  
        .whenComplete(() => print("Note question added to the database"))  
        .catchError((e) => print(e));  
}
```

### ReadQuestion

```
static Stream<QuerySnapshot> readQuestions() {  
    Query<Map<String, dynamic>> noteQuestionCollection =  
        _firestore.collection('question').orderBy("time", descending: true);  
    print('fetched');  
  
    return noteQuestionCollection.snapshots();  
}
```

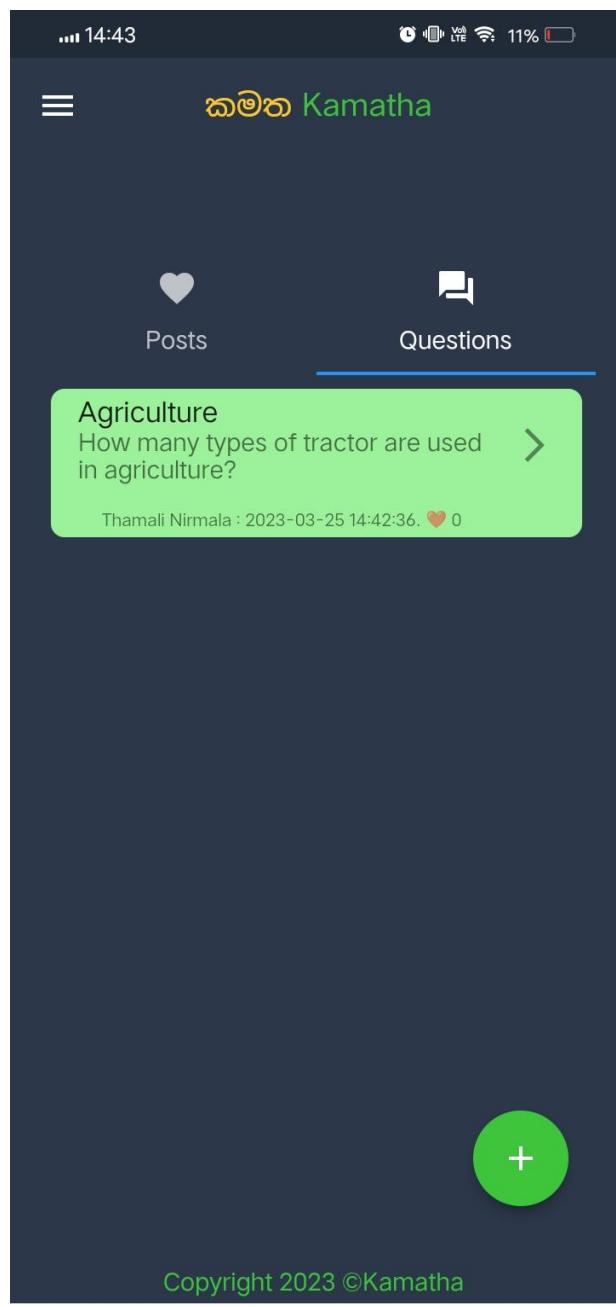
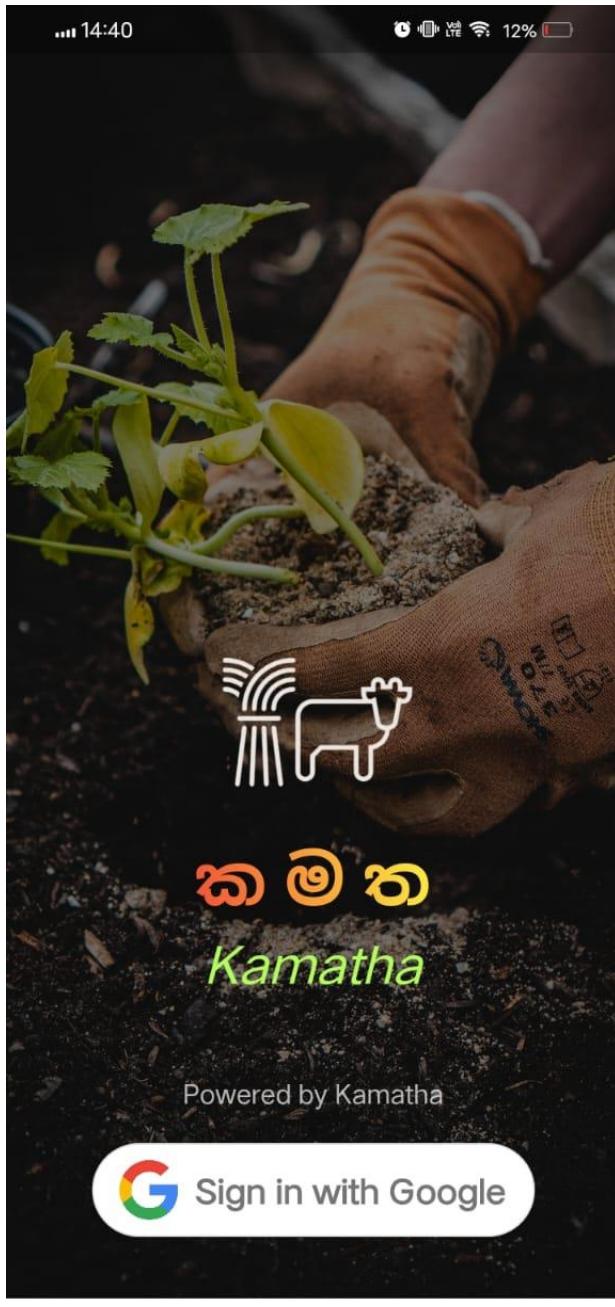
### **UpdateQuestion**

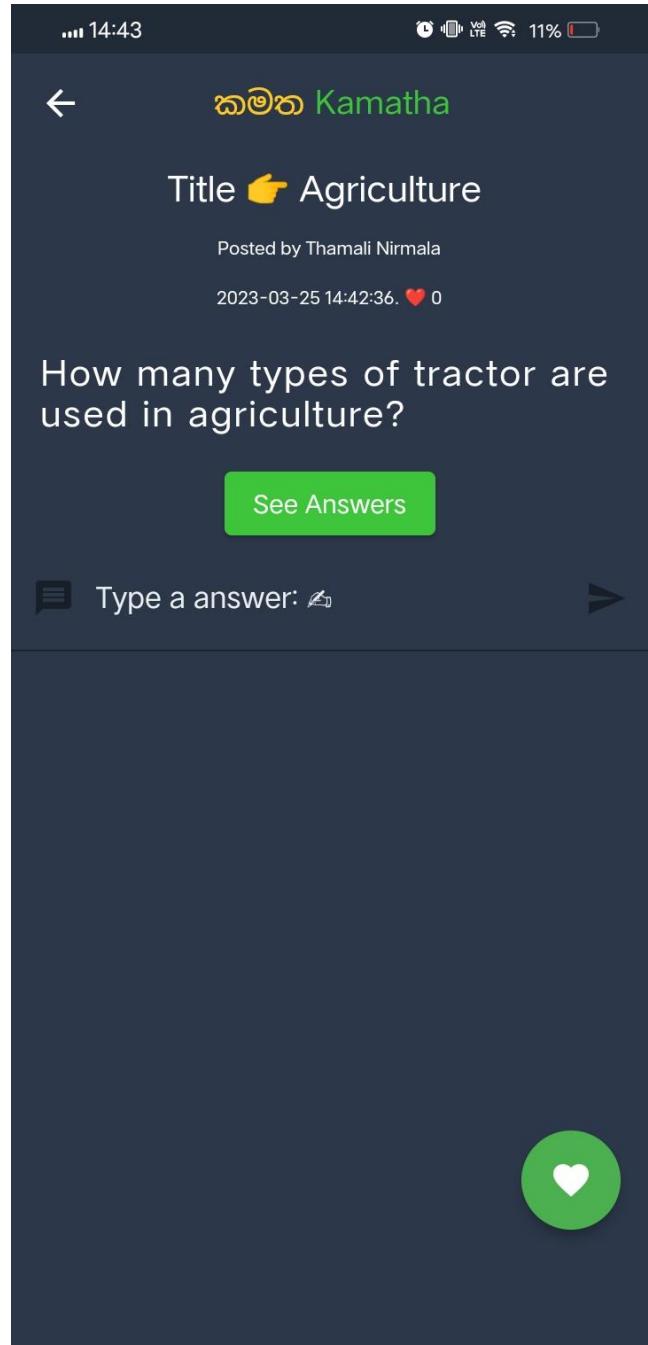
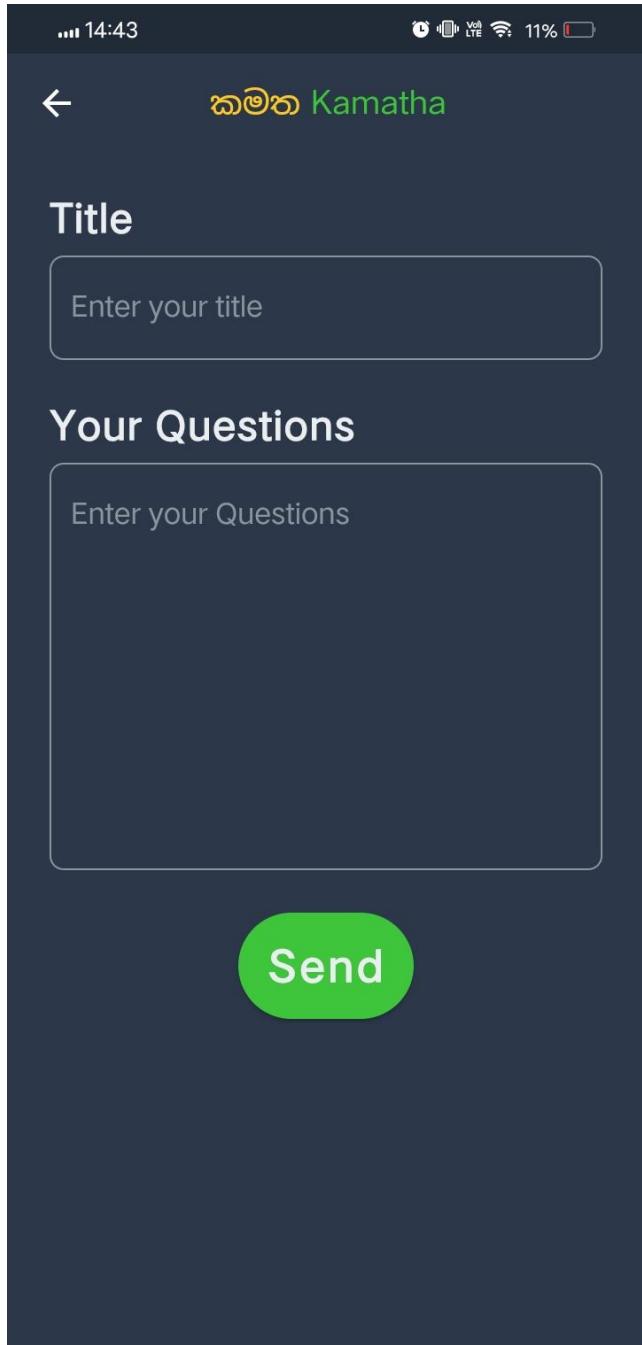
```
static Future<void> updateQuestion({  
    required String title,  
    required String question,  
    required String docId,  
) async {  
    DocumentReference documentReferencer =  
        _firestore.collection('question').doc(docId);  
  
    Map<String, dynamic> data = <String, dynamic>{  
        "title": title,  
        "question": question,  
    };  
  
    await documentReferencer  
        .update(data)  
        .whenComplete(() => print("Note question updated in the database"))  
        .catchError((e) => print(e));  
}
```

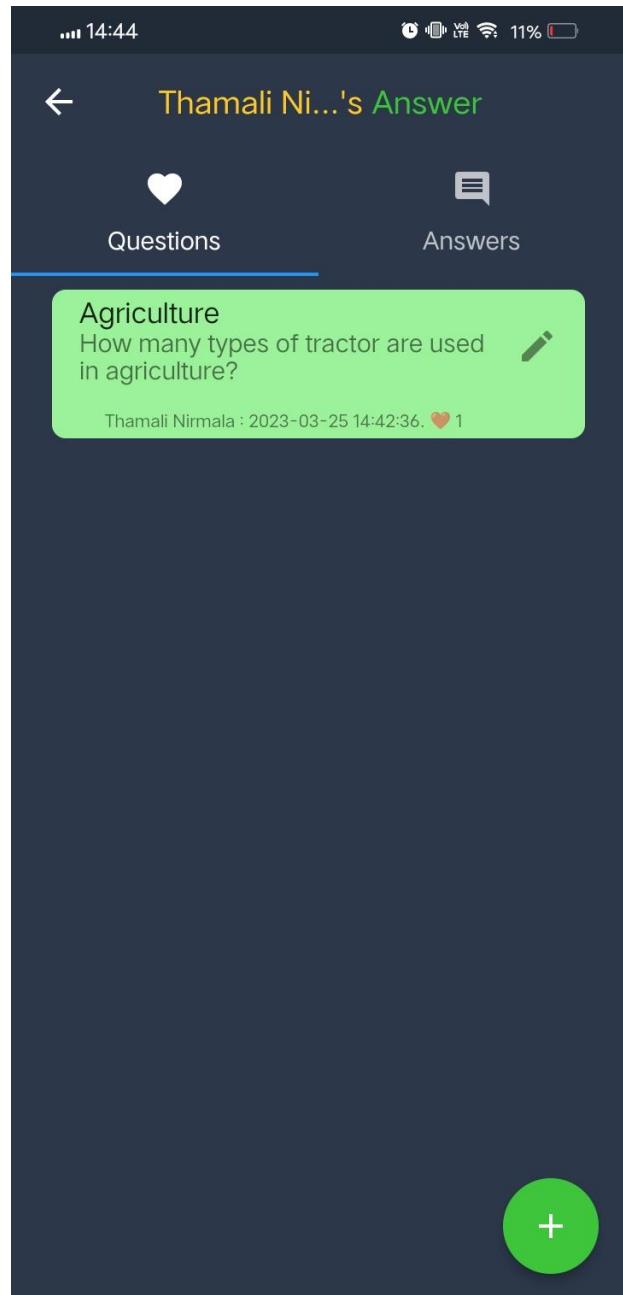
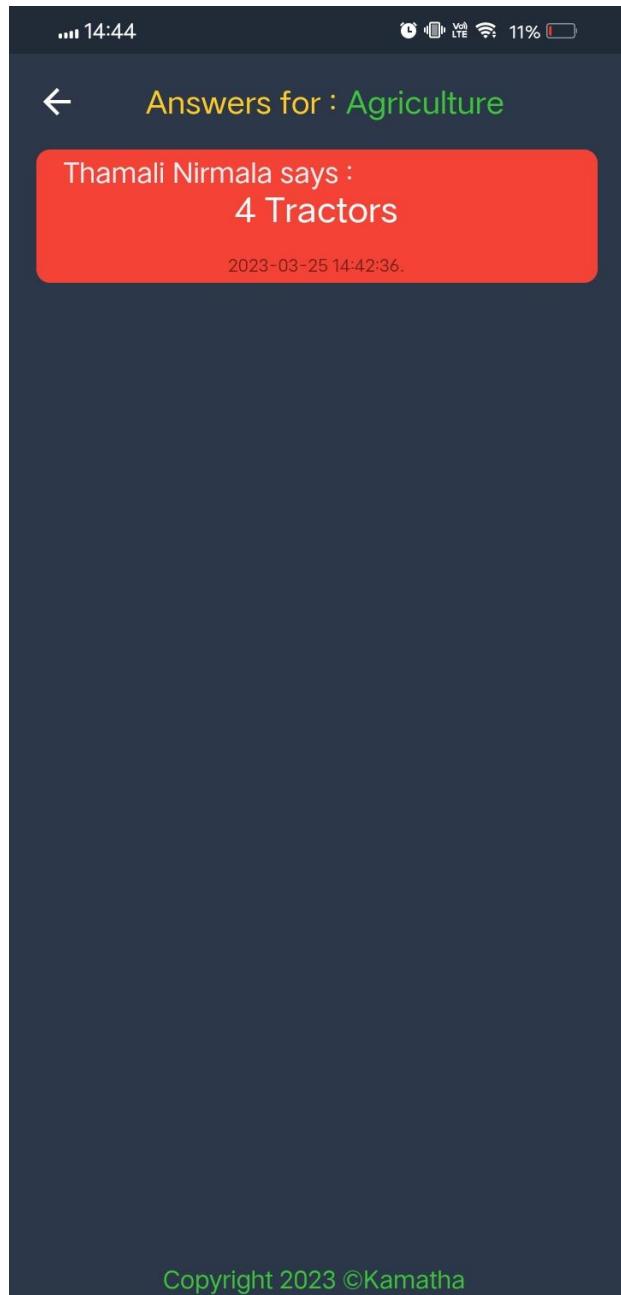
### **DeleteQuestion**

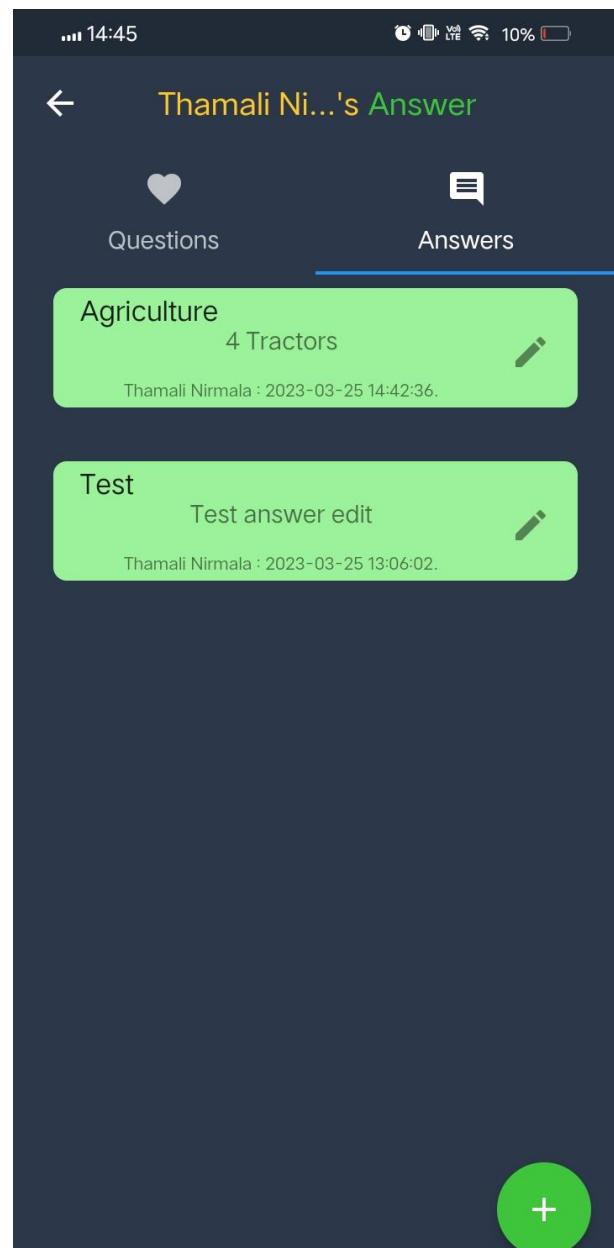
```
static Future<void> deleteQuestion({  
    required String docId,  
) async {  
    DocumentReference documentReferencer =  
        _firestore.collection('question').doc(docId);  
  
    await documentReferencer  
        .delete()  
        .whenComplete(() => print('Note question deleted from the database'))  
        .catchError((e) => print(e));  
}
```

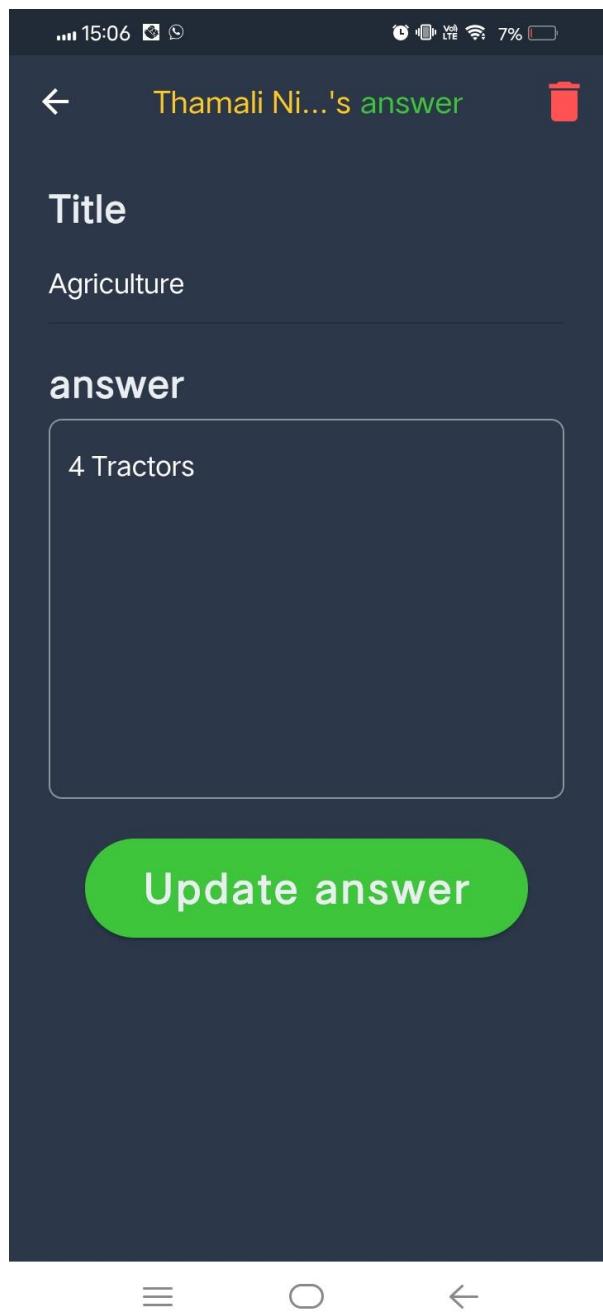
## UI Screenshots









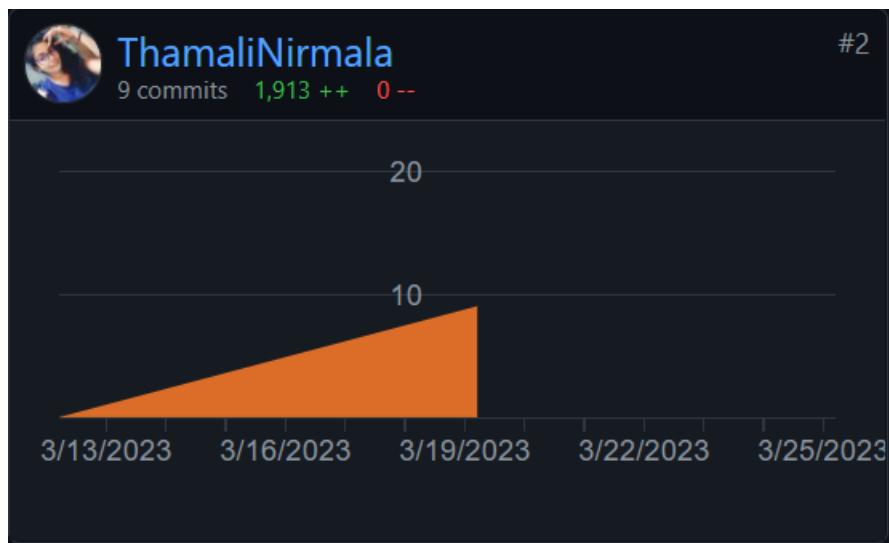


≡

○

←

## Screenshots of Contributions



## **References**

<https://firebase.google.com/docs>

<https://flutter.dev/learn>

<https://dart.dev/guides>

## **Personalized Crop**

In this component, users can log and post their items to sell, and there are two sections, my ads and all ads. User can view own ads in my ads section, and he can update ads and delete ads. And, he can view complaints regarding his ads. And also, the user can view all ads section in other users' ads and if there are any complaints regarding ads he can add complaint in add complaint section.

## **CODE SNIPPETS**

### **Screens**

addMyads.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/widgets/add_item_form.dart';
import 'package:Kamataha/src/widgets/add_myadd_form.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class AddMyAdd extends StatefulWidget {
  const AddMyAdd({Key? key, required User user})
      : _user = user,
        super(key: key);

  final User _user;

  @override
  State<AddMyAdd> createState() => _AddMyAddState();
}

class _AddMyAddState extends State<AddMyAdd> {
  final FocusNode _titleFocusNode = FocusNode();
  final FocusNode _priceFocusNode = FocusNode();
  final FocusNode _locationFocusNode = FocusNode();
  final FocusNode _quantityFocusNode = FocusNode();
  final FocusNode _phoneNumberFocusNode = FocusNode();
```

```
final FocusNode _descriptionFocusNode = FocusNode();  
  
late User _user;  
  
@override  
void initState() {  
    _user = widget._user;  
  
    super.initState();  
}  
  
@override  
Widget build(BuildContext context) {  
    return GestureDetector(  
        onTap: () {  
            _titleFocusNode.unfocus();  
            _descriptionFocusNode.unfocus();  
        },  
        child: Scaffold(  
            backgroundColor: CustomColors.firebaseioNavy,  
            appBar: AppBar(  
                elevation: 0,  
                backgroundColor: CustomColors.firebaseioNavy,  
                title: AppBarTitle(),  
                centerTitle: true,  
            ),  
            body: SafeArea(  
                child: ListView(children: [  
                    Padding(  
                        padding: const EdgeInsets.only(  
                            left: 16.0,  
                            right: 16.0,  
                            bottom: 20.0,  
                        ),  
                    child: AddMyAddForm(  
                        titleFocusNode: _titleFocusNode,  
                        priceFocusNode: _priceFocusNode,  
                        locationFocusNode: _locationFocusNode,  
                        quantityFocusNode: _quantityFocusNode,  
                        phoneNumberFocusNode: _phoneNumberFocusNode,  
                        descriptionFocusNode: _descriptionFocusNode,  
                        user: _user,  
                    ),  
                ]),  
            ),  
        ),  
    );  
}
```

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

### myAdd.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/add_myads.dart';
import 'package:Kamataha/src/screens/add_screen.dart';
import 'package:Kamataha/src/widgets/item_list.dart';
import 'package:Kamataha/src/widgets/my_add.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class MyAdd extends StatefulWidget {
  const MyAdd({Key? key, required User user})
      : _user = user,
        super(key: key);

  final User _user;

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

class _MyAddState extends State<MyAdd> {
  late User _user;

  @override
  void initState() {
    _user = widget._user;
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: CustomColors.firebaseioNavy,
      body: MyAddList(user: _user),
    );
  }
}
```

```
floatingActionButton: FloatingActionButton(
    child: const Icon(Icons.add),
    backgroundColor: CustomColors.firebaseioOrange,
    onPressed: () {
        Navigator.of(context).push(
            MaterialPageRoute(
                builder: (context) => AddMyAdd(user: _user),
            ),
        );
    },
),
);
}
```

## allAdd.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/add_myads.dart';
import 'package:Kamataha/src/screens/add_screen.dart';
import 'package:Kamataha/src/widgets/all_add_list.dart';
import 'package:Kamataha/src/widgets/item_list.dart';
import 'package:Kamataha/src/widgets/my_add.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class AllAdd extends StatefulWidget {
    const AllAdd({Key? key, required User user})
        : _user = user,
        super(key: key);

    final User _user;

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

class _AllAddState extends State<AllAdd> {
    late User _user;
```

```

@Override
void initState() {
    _user = widget._user;
    super.initState();
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: CustomColors.firebaseioNavy,
        body: AllAddList(user: _user),
        floatingActionButton: FloatingActionButton(
            child: const Icon(Icons.add),
            backgroundColor: CustomColors.firebaseioOrange,
            onPressed: () {
                Navigator.of(context).push(
                    MaterialPageRoute(
                        builder: (context) => AddMyAdd(user: _user),
                    ),
                );
            },
        ),
    );
}

```

## displayAllAdds.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/add_complaint.dart';
import 'package:Kamataha/src/screens/comment_screen.dart';
import 'package:Kamataha/src/screens/complaints.dart';
import 'package:Kamataha/src/screens/edit_item_screen.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:Kamataha/src/widgets/text_input_widget.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';

class DisplayAllAds extends StatefulWidget {
    final String currentTitle;

```

```
final String currentPrice;
final String currentLocation;
final String currentQuantity;
final String currentPhoneNumber;
final String currentDescription;
final String name;
final String documentId;
final String time;
final User _user;
final String type;
final String signedUrl;

const DisplayAllAds(
    {Key? key,
     required this.currentTitle,
     required this.currentPrice,
     required this.currentLocation,
     required this.currentQuantity,
     required this.currentPhoneNumber,
     required this.currentDescription,
     required this.name,
     required this.documentId,
     required this.time,
     required User user,
     this.type = "",
     this.signedUrl = ""})
    : _user = user,
    super(key: key);

@Override
_DisplayAllAdsState createState() => _DisplayAllAdsState();
}

class _DisplayAllAdsState extends State<DisplayAllAds> {
late User _user;

var isPressed = false;

@Override
void initState() {
    _user = widget._user;
    super.initState();
}

void showToast(isPressed) {
```

```
    if (isPressed) {
        Fluttertoast.showToast(
            msg: 'You Liked This',
            toastLength: Toast.LENGTH_SHORT,
            gravity: ToastGravity.BOTTOM,
            timeInSecForIosWeb: 1,
            backgroundColor: Colors.red,
            textColor: Colors.yellow);
    } else {
        Fluttertoast.showToast(
            msg: 'Like Removed',
            toastLength: Toast.LENGTH_SHORT,
            gravity: ToastGravity.BOTTOM,
            timeInSecForIosWeb: 1,
            backgroundColor: Colors.red,
            textColor: Colors.yellow);
    }
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: CustomColors.firebaseioNavy,
        appBar: AppBar(
            elevation: 0,
            backgroundColor: CustomColors.firebaseioNavy,
            title: AppBarTitle(),
            centerTitle: true,
        ),
        body: SafeArea(
            child: ListView(
                children: [
                    Center(
                        child: Container(
                            child: Image.network(
                                widget.signedUrl,
                                width: 200,
                                height: 200,
                            ),
                    )));
                ],
            ),
            Column(
                children: [
                    Center(
                        child: Padding(

```





```
        letterSpacing: 1,
        wordSpacing: 1),
    ),
),
),
),
if (widget.type == "all")
ElevatedButton(
 onPressed: () {
// Navigator.of(context).push(MaterialPageRoute(
//     builder: (context) => CommentScreen(
//         comment: widget.currentTitle,
//         )));
},
child: const Text("Call"),
style: ButtonStyle(
    backgroundColor: MaterialStateProperty.all(
        CustomColors.firebaseioOrange)),
),
if (widget.type == "all")
ElevatedButton(
 onPressed: () {
Navigator.of(context).push(MaterialPageRoute(
    builder: (context) => AddComplaint(
        title: widget.currentTitle,
        user: _user,
        )));
},
child: const Text("Add complaint"),
style: ButtonStyle(
    backgroundColor: MaterialStateProperty.all(
        CustomColors.firebaseioOrange)),
),
if (widget.type == "my")
ElevatedButton(
 onPressed: () {
Navigator.of(context).push(MaterialPageRoute(
    builder: (context) => EditItemNewScreen(
        title: widget.currentTitle,
        price: widget.currentPrice,
        location: widget.currentLocation,
        description: widget.currentDescription,
        phoneNumber: widget.currentPhoneNumber,
        quantity: widget.currentQuantity,
        signedUrl: widget.signedUrl,
        documentId: widget.documentId,
        )));
}
);
```

```
        user: widget._user.displayName.toString(),
    )));
},
child: const Text("EDIT"),
style: ButtonStyle(
    backgroundColor: MaterialStateProperty.all(
        CustomColors.firebaseioOrange)),
),
if (widget.type == "my")
ElevatedButton(
    onPressed: () async {
        await Database.deleteAd(
            docId: widget.documentId,
        );
        Navigator.of(context).pop();
    },
    child: const Text("DELETE"),
    style: ButtonStyle(
        backgroundColor: MaterialStateProperty.all(
            CustomColors.firebaseioOrange)),
),
if (widget.type == "my")
ElevatedButton(
    onPressed: () {
        Navigator.of(context).push(MaterialPageRoute(
            builder: (context) => Complaints(
                title: widget.currentTitle,
                user: _user,
            )));
    },
    child: const Text("VIEW COMPLAINT"),
    style: ButtonStyle(
        backgroundColor: MaterialStateProperty.all(
            CustomColors.firebaseioOrange)),
),
],
),
],
),
),
);
}
}
```

## editItemScreen.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/widgets/edit_comment_form.dart';
import 'package:Kamataha/src/widgets/edit_sell_item_form.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class EditItemNewScreen extends StatefulWidget {
    final String title;
    final String price;
    final String location;
    final String description;
    final String phoneNumber;
    final String quantity;
    final String signedUrl;
    final String documentId;
    final String user;

    // ignore: use_key_in_widget_constructors
    const EditItemNewScreen({
        required this.title,
        required this.price,
        required this.location,
        required this.description,
        required this.phoneNumber,
        required this.quantity,
        required this.signedUrl,
        required this.documentId,
        required this.user
    });

    @override
    _EditItemNewScreenState createState() => _EditItemNewScreenState();
}
```

```
}

class _EditItemNewScreenState extends State<EditItemNewScreen> {
    final FocusNode _titleFocusNode = FocusNode();

    final FocusNode _descriptionFocusNode = FocusNode();

    bool _isDeleting = false;

    @override
    Widget build(BuildContext context) {
        return GestureDetector(
            onTap: () {
                _titleFocusNode.unfocus();
                _descriptionFocusNode.unfocus();
            },
            child: Scaffold(
                backgroundColor: CustomColors.firebaseioNavy,
                appBar: AppBar(
                    elevation: 0,
                    backgroundColor: CustomColors.firebaseioNavy,
                    title: Row(
                        mainAxisAlignment: MainAxisAlignment.start,
                        mainAxisSize: MainAxisSize.min,
                        children: [
                            const SizedBox(width: 8),
                            Text(
                                "Edit Item",
                                style: TextStyle(
                                    color: CustomColors.firebaseioYellow,
                                    fontSize: 18,
                                ),
                            ),
                        ],
                    ),
                    centerTitle: true,
                    // actions: [
                    //     _isDeleting
                    //         ? const Padding(
                    //             padding: EdgeInsets.only(
                    //                 top: 10.0,
                    //                 bottom: 10.0,
                    //                 right: 16.0,
                    //             ),
                    //             child: CircularProgressIndicator(

```

```
//           valueColor: AlwaysStoppedAnimation<Color>(
//             Colors.redAccent,
//           ),
//           strokeWidth: 3,
//         ),
//       )
//     : IconButton(
//       icon: const Icon(
//         Icons.delete,
//         color: Colors.redAccent,
//         size: 32,
//       ),
//       onPressed: () async {
//         setState(() {
//           _isDeleting = true;
//         });
//
//         await Database.deleteComment(
//           docId: widget.documentId,
//         );
//
//         setState(() {
//           _isDeleting = false;
//         });
//
//         Navigator.of(context).pop();
//       },
//     ),
//   ],
// ),
body: SafeArea(
  child: Padding(
    padding: const EdgeInsets.only(
      left: 16.0,
      right: 16.0,
      bottom: 20.0,
    ),
    child: EditSellItemForm(
      title: widget.title,
      price: widget.price,
      location: widget.location,
      description: widget.description,
      phoneNumber: widget.phoneNumber,
      quantity: widget.quantity,
      signedUrl: widget.signedUrl,
```

```

        documentId: widget.documentId,
        user: widget.user
    ),
),
),
),
),
);
}
}

```

### sellerDashboard.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/all_add.dart';
import 'package:Kamataha/src/screens/my_add.dart';
import 'package:Kamataha/src/screens/posts.dart';
import 'package:Kamataha/src/screens/questions.dart';
import 'package:Kamataha/src/widgets/all_add_list.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:Kamataha/src/widgets/item_list.dart';
import 'package:Kamataha/src/widgets/my_add.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'add_screen.dart';

class SellerDashboard extends StatefulWidget {
    const SellerDashboard({Key? key, required User user})
        : _user = user,
        super(key: key);

    final User _user;

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

class _DashboardState extends State<SellerDashboard> {
    late User _user;

    @override

```

```
void initState() {
    _user = widget._user;
    super.initState();
}

@Override
Widget build(BuildContext context) {
    return DefaultTabController(
        length: 2,
        child: Scaffold(
            backgroundColor: CustomColors.firebaseioNavy,
            appBar: AppBar(
                title: Text(
                    "Items",
                    style: TextStyle(color: Colors.green),
                ),
                backgroundColor: CustomColors.firebaseioNavy,
                elevation: 0,
                centerTitle: true,
                bottom: const TabBar(
                    tabs: [
                        Tab(
                            text: "All Ads",
                            //icon: Icon(Icons.),
                        ),
                        Tab(
                            text: "My Ads",
                            //icon: Icon(Icons.question_answer),
                        ),
                    ],
                )),
                // floatingActionButton: FloatingActionButton(
                //     child: const Icon(Icons.add),
                //     backgroundColor: CustomColors.firebaseioOrange,
                //     onPressed: () {
                //         Navigator.of(context).push(
                //             MaterialPageRoute(
                //                 builder: (context) => AddScreen(user: _user),
                //             ),
                //         );
                //     },
                // ),
                body: TabBarView(
                    children: [AllAdd(user: _user), MyAdd(user: _user)],
                ),
            ),
        ),
    );
}
```

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

### addComplaint.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/widgets/add_complaint_form.dart';
import 'package:Kamataha/src/widgets/add_item_form.dart';
import 'package:Kamataha/src/widgets/add_question_form.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class AddComplaint extends StatefulWidget {
    const AddComplaint({Key? key, required this.title, required User user})
        : _user = user,
        super(key: key);

    final User _user;
    final String title;

    @override
    State<AddComplaint> createState() => _AddComplaintState();
}

class _AddComplaintState extends State<AddComplaint> {
    final FocusNode _titleFocusNode = FocusNode();
    final FocusNode _questionFocusNode = FocusNode();

    late User _user;

    @override
    void initState() {
        _user = widget._user;

        super.initState();
    }
}
```

```
@override
Widget build(BuildContext context) {
    return GestureDetector(
        onTap: () {
            _titleFocusNode.unfocus();
            _questionFocusNode.unfocus();
        },
        child: Scaffold(
            backgroundColor: CustomColors.firebaseioNavy,
            appBar: AppBar(
                elevation: 0,
                backgroundColor: CustomColors.firebaseioNavy,
                title: AppBarTitle(),
                centerTitle: true,
            ),
            body: SafeArea(
                child: ListView(children: [
                    Padding(
                        padding: const EdgeInsets.only(
                            left: 16.0,
                            right: 16.0,
                            bottom: 20.0,
                        ),
                        child: AddComplaintForm(
                            titleFocusNode: _titleFocusNode,
                            questionFocusNode: _questionFocusNode,
                            user: _user,
                            title: widget.title),
                    ),
                ],
            ),
        );
    }
}
```

## Complaints.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/add_screen.dart';
```

```
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:Kamataha/src/widgets/complaints_list.dart';
import 'package:Kamataha/src/widgets/item_list.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class Complaints extends StatefulWidget {
    final String title;
    const Complaints({Key? key, required this.title, required User user})
        : _user = user,
        super(key: key);

    final User _user;

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

class _ComplaintsState extends State<Complaints> {
    late User _user;

    @override
    void initState() {
        _user = widget._user;
        super.initState();
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                backgroundColor: CustomColors.firebaseioNavy,
                title: AppBarTitle(),
            ),
            backgroundColor: CustomColors.firebaseioNavy,
            body: ComplaintsList(title: widget.title),
        );
    }
}
```

## WIDGETS

addComplaintForm.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'custom_form_field.dart';

// ignore: must_be_immutable
class AddComplaintForm extends StatefulWidget {
  final FocusNode titleFocusNode;
  final FocusNode questionFocusNode;

  final String title;

  const AddComplaintForm(
    {Key? key,
    required this.titleFocusNode,
    required this.questionFocusNode,
    required User user,
    required this.title})
    : _user = user,
    super(key: key);

  final User _user;

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

class _AddComplaintFormState extends State<AddComplaintForm> {
  final _addQuestionFormKey = GlobalKey<FormState>();

  late User _user;

  bool _isProcessing = false;

  final TextEditingController _titleController = TextEditingController();
  final TextEditingController _questionController = TextEditingController();

  @override
```

```
void initState() {
    _user = widget._user;
    super.initState();
}

@Override
Widget build(BuildContext context) {
    return Form(
        key: _addQuestionFormKey,
        child: Column(
            children: [
                Padding(
                    padding: const EdgeInsets.only(
                        left: 8.0,
                        right: 8.0,
                        bottom: 24.0,
                    ),
                    child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                            const SizedBox(height: 24.0),
                            Text(
                                'Title',
                                style: TextStyle(
                                    color: CustomColors.firebaseioGrey,
                                    fontSize: 22.0,
                                    letterSpacing: 1,
                                    fontWeight: FontWeight.bold,
                                ),
                            ),
                            const SizedBox(height: 8.0),
                            CustomFormField(
                                isLabelEnabled: false,
                                controller: _titleController,
                                focusNode: widget.titleFocusNode,
                                keyboardType: TextInputType.text,
                                inputAction: TextInputAction.next,
                                validator: (value) => Validator.validateField(
                                    value: value,
                                ),
                                label: 'Title',
                                hint: 'Enter your title',
                            ),
                            const SizedBox(height: 24.0),
                            Text(

```

```
'Description',
style: TextStyle(
    color: CustomColors.firebaseioGrey,
    fontSize: 22.0,
    letterSpacing: 1,
    fontWeight: FontWeight.bold,
),
),
const SizedBox(height: 8.0),
CustomFormField(
    maxLines: 10,
    isLabelEnabled: false,
    controller: _questionController,
    focusNode: widget.questionFocusNode,
    keyboardType: TextInputType.text,
    inputAction: TextInputAction.done,
    validator: (value) => Validator.validateField(
        value: value,
    ),
    label: 'Description',
    hint: 'Enter your description',
),
],
),
),
),
_isProcessing
? Padding(
    padding: const EdgeInsets.all(16.0),
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
    ),
)
: ElevatedButton(
    style: ButtonStyle(
        backgroundColor: MaterialStateProperty.all(
            CustomColors.firebaseioOrange,
        ),
        shape: MaterialStateProperty.all(
            RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(45),
            ),
        ),
),
),
```

```
        onPressed: () async {
            widget.titleFocusNode.unfocus();
            widget.questionFocusNode.unfocus();

            if (_addQuestionFormKey.currentState!.validate()) {
                setState(() {
                    _isProcessing = true;
                });

                await Database.addComplaint(
                    title: _titleController.text,
                    question: _questionController.text,
                    user: _user.displayName.toString(),
                    ad: widget.title);

                setState(() {
                    _isProcessing = false;
                });

                Navigator.of(context).pop();
            }
        },
    child: Padding(
        padding: const EdgeInsets.only(top: 16.0, bottom: 16.0),
        child: Text(
            'Send',
            style: TextStyle(
                fontSize: 24,
                fontWeight: FontWeight.bold,
                color: CustomColors.firebaseioGrey,
                letterSpacing: 2,
            ),
            ),
            ),
            ),
            ],
        );
    );
}
```

## addMyAddForm.dart

```
import 'dart:io';
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_storage/firebase_storage.dart';
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:image_picker/image_picker.dart';
import 'custom_form_field.dart';

// ignore: must_be_immutable
class AddMyAddForm extends StatefulWidget {
    final FocusNode titleFocusNode;
    final FocusNode priceFocusNode;
    final FocusNode locationFocusNode;
    final FocusNode quantityFocusNode;
    final FocusNode phoneNumberFocusNode;
    final FocusNode descriptionFocusNode;

    const AddMyAddForm({
        Key? key,
        required this.titleFocusNode,
        required this.priceFocusNode,
        required this.locationFocusNode,
        required this.quantityFocusNode,
        required this.phoneNumberFocusNode,
        required this.descriptionFocusNode,
        required User user,
    }) : _user = user,
        super(key: key);

    final User _user;

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

class _AddMyAddFormState extends State<AddMyAddForm> {
    final _addMyAddFormKey = GlobalKey<FormState>();

    late User _user;
```

```
bool _isProcessing = false;

final TextEditingController _titleController = TextEditingController();
final TextEditingController _priceController = TextEditingController();
final TextEditingController _locationController = TextEditingController();
final TextEditingController _quantityController = TextEditingController();
final TextEditingController _phoneNumberController = TextEditingController();
final TextEditingController _descriptionController = TextEditingController();

String signedUrl = "";

@Override
void initState() {
    _user = widget._user;
    super.initState();
}

uploadPic() async {
    final XFile? image =
        await ImagePicker().pickImage(source: ImageSource.gallery);
    FirebaseStorage storage = FirebaseStorage.instance;
    Reference ref = storage.ref().child("images/" + DateTime.now().toString());
    UploadTask uploadTask = ref.putFile(File(image!.path));
    uploadTask.then((res) async {
        signedUrl = await res.ref.getDownloadURL().then((value) {
            Fluttertoast.showToast(
                msg: 'File Uploaded Successfully',
                toastLength: Toast.LENGTH_SHORT,
                gravity: ToastGravity.BOTTOM,
                timeInSecForIosWeb: 1,
                backgroundColor: Colors.red,
                textColor: Colors.yellow);
            return value;
        });
    });
}

@Override
Widget build(BuildContext context) {
    return Form(
        key: _addMyAddFormKey,
        child: Column(
            children: [
                Padding(
```

```
padding: const EdgeInsets.only(
    left: 8.0,
    right: 8.0,
    bottom: 24.0,
),
child: Column(
    mainAxisAlignment: MainAxisAlignment.start,
    children: [
        const SizedBox(height: 24.0),
        Text(
            'Title',
            style: TextStyle(
                color: CustomColors.firebaseioGrey,
                fontSize: 22.0,
                letterSpacing: 1,
                fontWeight: FontWeight.bold,
            ),
        ),
        const SizedBox(height: 8.0),
        const SizedBox(height: 8.0),
        CustomFormField(
            isLabelEnabled: false,
            controller: _titleController,
            focusNode: widget.titleFocusNode,
            keyboardType: TextInputType.text,
            inputAction: TextInputAction.next,
            validator: (value) => Validator.validateField(
                value: value,
            ),
            label: 'Titlt',
            hint: 'Enter your title',
        ),
        const SizedBox(height: 24.0),
        Text(
            'Price',
            style: TextStyle(
                color: CustomColors.firebaseioGrey,
                fontSize: 22.0,
                letterSpacing: 1,
                fontWeight: FontWeight.bold,
            ),
        ),
        const SizedBox(height: 8.0),
        CustomFormField(
            isLabelEnabled: false,
```

```
        controller: _priceController,
        focusNode: widget.priceFocusNode,
        keyboardType: TextInputType.number,
        inputAction: TextInputAction.next,
        validator: (value) => Validator.validateField(
            value: value,
        ),
        label: 'Price',
        hint: 'Enter your price',
    ),
    const SizedBox(height: 24.0),
    Text(
        'Location',
        style: TextStyle(
            color: CustomColors.firebaseioGrey,
            fontSize: 22.0,
            letterSpacing: 1,
            fontWeight: FontWeight.bold,
        ),
    ),
    const SizedBox(height: 8.0),
    CustomFormField(
        isLabelEnabled: false,
        controller: _locationController,
        focusNode: widget.locationFocusNode,
        keyboardType: TextInputType.text,
        inputAction: TextInputAction.next,
        validator: (value) => Validator.validateField(
            value: value,
        ),
        label: 'Location',
        hint: 'Enter your location',
    ),
    const SizedBox(height: 24.0),
    Text(
        'Quantity',
        style: TextStyle(
            color: CustomColors.firebaseioGrey,
            fontSize: 22.0,
            letterSpacing: 1,
            fontWeight: FontWeight.bold,
        ),
    ),
    const SizedBox(height: 8.0),
    CustomFormField(
```

```
        isLabelEnabled: false,
        controller: _quantityController,
        focusNode: widget.quantityFocusNode,
        keyboardType: TextInputType.number,
        inputAction: TextInputAction.next,
        validator: (value) => Validator.validateField(
            value: value,
        ),
        label: 'Quantity',
        hint: 'Enter your quantity',
    ),
    const SizedBox(height: 24.0),
    Text(
        'Phone Number',
        style: TextStyle(
            color: CustomColors.firebaseioGrey,
            fontSize: 22.0,
            letterSpacing: 1,
            fontWeight: FontWeight.bold,
        ),
    ),
    const SizedBox(height: 8.0),
    CustomFormField(
        isLabelEnabled: false,
        controller: _phoneNumberController,
        focusNode: widget.phoneNumberFocusNode,
        keyboardType: TextInputType.number,
        inputAction: TextInputAction.next,
        validator: (value) => Validator.validateField(
            value: value,
        ),
        label: 'Phone Number',
        hint: 'Enter your item phone number',
    ),
    const SizedBox(height: 24.0),
    Text(
        'Description',
        style: TextStyle(
            color: CustomColors.firebaseioGrey,
            fontSize: 22.0,
            letterSpacing: 1,
            fontWeight: FontWeight.bold,
        ),
    ),
    const SizedBox(height: 8.0),
```

```
CustomFormField(
    maxLines: 10,
    isLabelEnabled: false,
    controller: _descriptionController,
    focusNode: widget.descriptionFocusNode,
    keyboardType: TextInputType.text,
    inputAction: TextInputAction.done,
    validator: (value) => Validator.validateField(
        value: value,
    ),
    label: 'Description',
    hint: 'Enter your item description',
),
ElevatedButton(
    onPressed: () {
        uploadPic();
    },
    child: const Text("+ UPLOAD IMAGE"),
    style: ButtonStyle(
        backgroundColor: MaterialStateProperty.all(
            CustomColors.firebaseioOrange)),
),
],
),
),
),
_isProcessing
? Padding(
    padding: const EdgeInsets.all(16.0),
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
    ),
)
: ElevatedButton(
    style: ButtonStyle(
        backgroundColor: MaterialStateProperty.all(
            CustomColors.firebaseioOrange,
        ),
        shape: MaterialStateProperty.all(
            RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(45),
            ),
        ),
    ),
),
```

```
        onPressed: () async {
            widget.titleFocusNode.unfocus();
            widget.descriptionFocusNode.unfocus();

            if (_addMyAddFormKey.currentState!.validate()) {
                setState(() {
                    _isProcessing = true;
                });

                await Database.addMyAddsItem(
                    title: _titleController.text,
                    price: _priceController.text,
                    location: _locationController.text,
                    quantity: _quantityController.text,
                    phoneNumber: _phoneNumberController.text,
                    description: _descriptionController.text,
                    user: _user.displayName.toString(),
                    signedUrl: signedUrl);

                setState(() {
                    _isProcessing = false;
                });

                Navigator.of(context).pop();
            }
        },
        child: Padding(
            padding: const EdgeInsets.only(top: 16.0, bottom: 16.0),
            child: Text(
                'Post',
                style: TextStyle(
                    fontSize: 24,
                    fontWeight: FontWeight.bold,
                    color: CustomColors.firebaseioGrey,
                    letterSpacing: 2,
                ),
            ),
            ),
            ),
            ),
            ],
        ),
    );
}
}
```

## allAddList.dart

```
// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/display_all.dart';
import 'package:Kamataha/src/screens/display_all_ads.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class AllAddList extends StatefulWidget {
  const AllAddList({required User user}) : _user = user;
  final User _user;
  @override
  State<AllAddList> createState() => _AllAddListState();
}

class _AllAddListState extends State<AllAddList> {
  late User _user;

  @override
  void initState() {
    _user = widget._user;
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return StreamBuilder<QuerySnapshot>(
      stream: Database.readALLAdds(),
      builder: (context, snapshot) {
        if (snapshot.hasError) {
          return const Text('Something went wrong');
        } else if (snapshot.hasData || snapshot.data != null) {
          if (snapshot.data!.docs.isEmpty) {
            return Center(
              child: Text(
                'No items found',
                style: TextStyle(
                  color: Colors.red,
                  fontSize: 18,
                ),
              ),
            );
          }
          List<Map<String, dynamic>> data = snapshot.data!.docs.map((doc) {
            Map<String, dynamic> docData = doc.data();
            docData['id'] = doc.id;
            return docData;
          }).toList();
          return ListView.builder(
            itemCount: data.length,
            itemBuilder: (context, index) {
              Map<String, dynamic> item = data[index];
              return ListTile(
                title: Text(item['name']),
                subtitle: Text(item['description']),
                trailing: IconButton(
                  icon: Icon(Icons.delete),
                  onPressed: () {
                    Database.deleteAdd(item['id']);
                    setState(() {});
                  },
                ),
              );
            },
          );
        }
      },
    );
  }
}
```

```
        "No adds yet",
        style: TextStyle(color: CustomColors.firebaseioGrey),
    ),
);
}
return ListView.separated(
    separatorBuilder: (context, index) => const SizedBox(height: 16.0),
    itemCount: snapshot.data!.docs.length,
    itemBuilder: (context, index) {
        var noteInfo = snapshot.data!.docs[index].data() as dynamic;
        String docID = snapshot.data!.docs[index].id;
        String title = noteInfo["title"];
        String price = noteInfo["price"];
        String location = noteInfo["location"];
        String quantity = noteInfo["quantity"];
        String phoneNumber = noteInfo["phoneNumber"];
        String description = noteInfo['description'];
        String user = noteInfo['user'];
        String time = noteInfo['time'];
        int likes = noteInfo['likes'];
        String signedUrl = noteInfo['signedUrl'];

        return Padding(
            padding: const EdgeInsets.all(8.0),
            child: Ink(
                decoration: BoxDecoration(
                    color: Color(0xFF9CF29A),
                    borderRadius: BorderRadius.circular(8.0),
                ),
                child: ListTile(
                    shape: RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(8.0),
                    ),
                    onTap: () => Navigator.of(context).push(
                        MaterialPageRoute(
                            builder: (context) => DisplayAllAds(
                                currentTitle: title,
                                currentPrice: price,
                                currentLocation: location,
                                currentQuantity: quantity,
                                currentPhoneNumber: phoneNumber,
                                currentDescription: description,
                                documentId: docID,
                                name: user,
                                time: time,
                            )
                        )
                    ),
                ),
            ),
        );
    }
);
```

```
        user: _user,
        type: "all",
        signedUrl: signedUrl),
    ),
),
title: Text(
    title,
    maxLines: 1,
    overflow: TextOverflow.ellipsis,
),
subtitle: Column(
    children: [
        Text(
            description,
            maxLines: 2,
            overflow: TextOverflow.ellipsis,
),
        const Text(""),
        // Text(
        //     user + " : " + time.substring(0, 20),
        //     maxLines: 1,
        //     overflow: TextOverflow.ellipsis,
        //     style: const TextStyle(fontSize: 10),
        // ),
    ],
),
trailing: const Padding(
    padding: EdgeInsets.only(top: 5),
    child: Icon(Icons.arrow_forward_ios_sharp),
),
),
));
}),
),
);
}

return Center(
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
),
),
),
);
},
);
});
```

```
    }
}
```

complaintsList.dart

```
// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';

class ComplaintsList extends StatefulWidget {
    final String title;
    const ComplaintsList({required this.title});
    @override
    State<ComplaintsList> createState() => _ComplaintsListState();
}

class _ComplaintsListState extends State<ComplaintsList> {
    //ADDED
    var colors = [
        Colors.red,
        Colors.blue,
        Colors.cyan,
        Colors.green,
        Colors.yellow,
    ];
    @override
    Widget build(BuildContext context) {
        return StreamBuilder<QuerySnapshot>(
            stream: Database.readComplaints(widget.title),
            builder: (context, snapshot) {
                if (snapshot.hasError) {
                    return const Text('Something went wrong');
                } else if (snapshot.hasData || snapshot.data != null) {
                    if (snapshot.data!.docs.isEmpty) {
                        return Center(
                            child: Text(
                                'No complaints found'
                            )
                        );
                    } else {
                        return ListView.builder(
                            itemCount: snapshot.data!.docs.length,
                            itemBuilder: (context, index) {
                                DocumentSnapshot document = snapshot.data!.docs[index];
                                Map<String, dynamic> data = document.data();
                                String title = data['title'];
                                String description = data['description'];
                                String date = data['date'];
                                String status = data['status'];
                                String colorIndex = data['color'];
                                Color color = colors[int.parse(colorIndex)];
                                return Card(
                                    color: color,
                                    elevation: 2,
                                    shape: RoundedRectangleBorder(
                                        borderRadius: BorderRadius.circular(10)
                                    ),
                                    child: Padding(
                                        padding: EdgeInsets.all(10),
                                        child: Column(
                                            crossAxisAlignment: CrossAxisAlignment.start,
                                            children: [
                                                Text(
                                                    title,
                                                    style: TextStyle(
                                                        color: Colors.black,
                                                        fontWeight: FontWeight.bold
                                                    )
                                                ),
                                                Text(
                                                    description,
                                                    style: TextStyle(
                                                        color: Colors.black,
                                                        fontSize: 14
                                                    )
                                                ),
                                                Text(
                                                    date,
                                                    style: TextStyle(
                                                        color: Colors.grey,
                                                        fontSize: 12
                                                    )
                                                ),
                                                Text(
                                                    status,
                                                    style: TextStyle(
                                                        color: Colors.grey,
                                                        fontSize: 12
                                                    )
                                                )
                                            ]
                                        )
                                    )
                                );
                            }
                        );
                    }
                }
            }
        );
    }
}
```

```
        "No complaints yet",
        style: TextStyle(color: CustomColors.firebaseioGrey),
    ),
);
}
return ListView.separated(
    separatorBuilder: (context, index) => const SizedBox(height: 16.0),
    itemCount: snapshot.data!.docs.length,
    itemBuilder: (context, index) {
        var noteInfo = snapshot.data!.docs[index].data() as dynamic;
        String user = noteInfo["user"];
        String complaint = noteInfo['complaint'];
        String title = noteInfo['title'];
        String time = noteInfo['time'];

        return Ink(
            decoration: BoxDecoration(
                color: colors[index],
                borderRadius: BorderRadius.circular(8.0),
            ),
            child: ListTile(
                shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(8.0),
                ),
                title: Text(
                    title,
                    maxLines: 1,
                    overflow: TextOverflow.ellipsis,
                    style: TextStyle(color: CustomColors.firebaseioGrey),
                ),
                subtitle: Column(
                    children: [
                        Text(
                            complaint,
                            maxLines: 2,
                            overflow: TextOverflow.ellipsis,
                            style: const TextStyle(
                                color: Colors.white,
                                fontSize: 20,
                            ),
                        ),
                        const Text(""),
                    ],
                ),
            ),
        );
    },
);
```

```
        );
    },
);
}

return Center(
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
    ),
);
},
);
}
}
}
```

### editSellItemForm.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'custom_form_field.dart';

// ignore: must_be_immutable
class EditSellItemForm extends StatefulWidget {
    final String title;
    final String price;
    final String location;
    final String description;
    final String phoneNumber;
    final String quantity;
    final String signedUrl;
    final String documentId;
    final String user;

    // final FocusNode _titleFocusNode = FocusNode();
```

```
// final FocusNode _priceFocusNode = FocusNode();
// final FocusNode _locationFocusNode = FocusNode();
// final FocusNode _quantityFocusNode = FocusNode();
// final FocusNode _phoneNumberFocusNode = FocusNode();
// final FocusNode _descriptionFocusNode = FocusNode();

EditSellItemForm(
    {required this.title,
    required this.price,
    required this.location,
    required this.description,
    required this.phoneNumber,
    required this.quantity,
    required this.signedUrl,
    required this.documentId,
    required this.user});

@Override
_EditSellItemFormState createState() => _EditSellItemFormState();
}

class _EditSellItemFormState extends State<EditSellItemForm> {
final _addItemFormKey = GlobalKey<FormState>();

late User _user;

bool _isProcessing = false;

TextEditingController _titleController = TextEditingController();
TextEditingController _priceController = TextEditingController();
TextEditingController _locationController = TextEditingController();
TextEditingController _quantityController = TextEditingController();
TextEditingController _phoneNumberController = TextEditingController();
TextEditingController _descriptionController = TextEditingController();

@Override
void initState() {
    _titleController = TextEditingController(
        text: widget.title,
    );

    _descriptionController = TextEditingController(
        text: widget.description,
    );
}
```

```
_priceController = TextEditingController(  
    text: widget.price,  
)  
_locationController = TextEditingController(  
    text: widget.location,  
)  
_locationController = TextEditingController(  
    text: widget.location,  
)  
_quantityController = TextEditingController(  
    text: widget.quantity,  
)  
_phoneNumberController = TextEditingController(  
    text: widget.phoneNumber,  
)  
super.initState();  
}  
  
@override  
Widget build(BuildContext context) {  
    return Form(  
        key: _addItemFormKey,  
        child: ListView(  
            children: [  
                Padding(  
                    padding: const EdgeInsets.only(  
                        left: 8.0,  
                        right: 8.0,  
                        bottom: 24.0,  
                    ),  
                child: Column(  
                    crossAxisAlignment: CrossAxisAlignment.start,  
                    children: [  
                        const SizedBox(height: 24.0),  
                        Text(  
                            'Title',  
                            style: TextStyle(  
                                color: CustomColors.firebaseioGrey,  
                                fontSize: 22.0,  
                                letterSpacing: 1,  
                                fontWeight: FontWeight.bold,  
                            ),  
                        ),  
                        const SizedBox(height: 8.0),  
                        const SizedBox(height: 8.0),  
                    ],  
                ),  
            ],  
        ),  
    );  
}
```

```
CustomFormField(  
    isEnabled: false,  
    controller: _titleController,  
    focusNode: FocusNode(),  
    keyboardType: TextInputType.text,  
    inputAction: TextInputAction.next,  
    validator: (value) => Validator.validateField(  
        value: value,  
    ),  
    label: 'Title',  
    hint: 'Enter your title',  
>,  
    const SizedBox(height: 24.0),  
    Text(  
        'Price',  
        style: TextStyle(  
            color: CustomColors.firebaseioGrey,  
            fontSize: 22.0,  
            letterSpacing: 1,  
            fontWeight: FontWeight.bold,  
        ),  
    ),  
>,  
    const SizedBox(height: 8.0),  
    CustomFormField(  
        isEnabled: false,  
        controller: _priceController,  
        focusNode: FocusNode(),  
        keyboardType: TextInputType.number,  
        inputAction: TextInputAction.next,  
        validator: (value) => Validator.validateField(  
            value: value,  
        ),  
        label: 'Price',  
        hint: 'Enter your price',  
>,  
    const SizedBox(height: 24.0),  
    Text(  
        'Location',  
        style: TextStyle(  
            color: CustomColors.firebaseioGrey,  
            fontSize: 22.0,  
            letterSpacing: 1,  
            fontWeight: FontWeight.bold,  
        ),  
    ),  
>,
```

```
        const SizedBox(height: 8.0),
        CustomFormField(
            isLabelEnabled: false,
            controller: _locationController,
            focusNode: FocusNode(),
            keyboardType: TextInputType.text,
            inputAction: TextInputAction.next,
            validator: (value) => Validator.validateField(
                value: value,
            ),
            label: 'Location',
            hint: 'Enter your location',
        ),
        const SizedBox(height: 24.0),
        Text(
            'Quantity',
            style: TextStyle(
                color: CustomColors.firebaseioGrey,
                fontSize: 22.0,
                letterSpacing: 1,
                fontWeight: FontWeight.bold,
            ),
        ),
        const SizedBox(height: 8.0),
        CustomFormField(
            isLabelEnabled: false,
            controller: _quantityController,
            focusNode: FocusNode(),
            keyboardType: TextInputType.number,
            inputAction: TextInputAction.next,
            validator: (value) => Validator.validateField(
                value: value,
            ),
            label: 'Quantity',
            hint: 'Enter your quantity',
        ),
        const SizedBox(height: 24.0),
        Text(
            'Phone Number',
            style: TextStyle(
                color: CustomColors.firebaseioGrey,
                fontSize: 22.0,
                letterSpacing: 1,
                fontWeight: FontWeight.bold,
            ),
        ),
```

```
),
const SizedBox(height: 8.0),
CustomFormField(
    isLabelEnabled: false,
    controller: _phoneNumberController,
    focusNode: FocusNode(),
    keyboardType: TextInputType.number,
    inputAction: TextInputAction.next,
    validator: (value) => Validator.validateField(
        value: value,
    ),
    label: 'Phone Number',
    hint: 'Enter your item phone number',
),
const SizedBox(height: 24.0),
Text(
    'Description',
    style: TextStyle(
        color: CustomColors.firebaseioGrey,
        fontSize: 22.0,
        letterSpacing: 1,
        fontWeight: FontWeight.bold,
    ),
),
const SizedBox(height: 8.0),
CustomFormField(
    maxLines: 10,
    isLabelEnabled: false,
    controller: _descriptionController,
    focusNode: FocusNode(),
    keyboardType: TextInputType.text,
    inputAction: TextInputAction.done,
    validator: (value) => Validator.validateField(
        value: value,
    ),
    label: 'Description',
    hint: 'Enter your item description',
),
],
),
),
_isProcessing
? Padding(
    padding: const EdgeInsets.all(16.0),
    child: CircularProgressIndicator(

```

```
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
    ),
)
: ElevatedButton(
    style: ButtonStyle(
        backgroundColor: MaterialStateProperty.all(
            CustomColors.firebaseioOrange,
        ),
        shape: MaterialStateProperty.all(
            RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(45),
            ),
        ),
    ),
    onPressed: () async {
        if (_addItemFormKey.currentState!.validate()) {
            setState(() {
                _isProcessing = true;
            });
        }

        await Database.editSellItem(
            title: _titleController.text,
            price: _priceController.text,
            location: _locationController.text,
            quantity: _quantityController.text,
            phoneNumber: _phoneNumberController.text,
            description: _descriptionController.text,
            signedUrl: widget.signedUrl,
            documentId: widget.documentId,
            user: widget.user);
    }

    setState(() {
        _isProcessing = false;
    });
}

Navigator.of(context).pop();
}
),
child: Padding(
    padding: const EdgeInsets.only(top: 16.0, bottom: 16.0),
    child: Text(
        'Update',
        style: TextStyle(

```

```
        fontSize: 24,
        fontWeight: FontWeight.bold,
        color: CustomColors.firebaseioGrey,
        letterSpacing: 2,
    ),
),
),
),
),
),
],
),
);
}
}
```

## myAdd.dart

```
// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/display_all.dart';
import 'package:Kamataha/src/screens/display_all_ads.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class MyAddList extends StatefulWidget {
    const MyAddList({required User user}) : _user = user;
    final User _user;
    @override
    State<MyAddList> createState() => _MyAddListState();
}

class _MyAddListState extends State<MyAddList> {
```



```
        color: Color(0xFF9CF29A),
        borderRadius: BorderRadius.circular(8.0),
    ),
    child: ListTile(
        shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(8.0),
        ),
        onTap: () => Navigator.of(context).push(
            MaterialPageRoute(
                builder: (context) => DisplayAllAds(
                    currentTitle: title,
                    currentPrice: price,
                    currentLocation: location,
                    currentQuantity: quantity,
                    currentPhoneNumber: phoneNumber,
                    currentDescription: description,
                    documentId: docID,
                    name: user,
                    time: time,
                    user: _user,
                    type: "my",
                    signedUrl: signedUrl),
            ),
        ),
        title: Text(
            title,
            maxLines: 1,
            overflow: TextOverflow.ellipsis,
        ),
        subtitle: Column(
            children: [
                Text(
                    description,
                    maxLines: 2,
                    overflow: TextOverflow.ellipsis,
                ),
                const Text(""),
            ],
        ),
        trailing: const Padding(
            padding: EdgeInsets.only(top: 5),
            child: Icon(Icons.arrow_forward_ios_sharp),
        ),
    ),
),
```

```

        )),
    },
);
}

return Center(
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
    ),
),
);
},
);
}
}
}

```

## CRUD OPEATIONS

```

static Future<void> addMyAddsItem(
    required String title,
    required String price,
    required String location,
    required String quantity,
    required String phoneNumber,
    required String description,
    required String user,
    required String signedUrl}) async {
    DocumentReference documentReferencer = _firestore.collection('ads').doc();

    Map<String, dynamic> data = <String, dynamic>{
        "title": title,
        "price": price,
        "location": location,
        "quantity": quantity,
        "phoneNumber": phoneNumber,
        "description": description,
        "user": user,
        "time": time,
        "signedUrl": signedUrl
    }
}

```

```
};

await documentReferencer
    .set(data)
    .whenComplete(() => print("Note ads added to the database"))
    .catchError((e) => print(e));
}

static Future<void> editSellItem(
    required String title,
    required String price,
    required String location,
    required String quantity,
    required String phoneNumber,
    required String description,
    required String signedUrl,
    required String documentId,
    required String user}) async {
    DocumentReference documentReferencer =
        _firestore.collection('ads').doc(documentId);

    Map<String, dynamic> data = <String, dynamic>{
        "title": title,
        "price": price,
        "location": location,
        "quantity": quantity,
        "phoneNumber": phoneNumber,
        "description": description,
        "signedUrl": signedUrl,
        "user": user
    };

    await documentReferencer
        .set(data)
        .whenComplete(() => print("Note ads added to the database"))
        .catchError((e) => print(e));
}
```

```
static Future<void> addComplaint(
    {required String title,
     required String question,
     required String user,
     required String ad}) async {
  DocumentReference documentReferencer =
      _firebase.collection('complaint').doc();

  Map<String, dynamic> data = <String, dynamic>{
    "title": title,
    "complaint": question,
    "user": user,
    "time": time,
    "ad": ad
  };

  await documentReferencer
      .set(data)
      .whenComplete(() => print("Note question added to the database"))
      .catchError((e) => print(e));
}
```

```
static Future<void> editSellItem(
    {required String title,
     required String price,
     required String location,
     required String quantity,
     required String phoneNumber,
     required String description,
     required String signedUrl,
     required String documentId,
     required String user}) async {
  DocumentReference documentReferencer =
      _firebase.collection('ads').doc(documentId);

  Map<String, dynamic> data = <String, dynamic>{
    "title": title,
    "price": price,
```

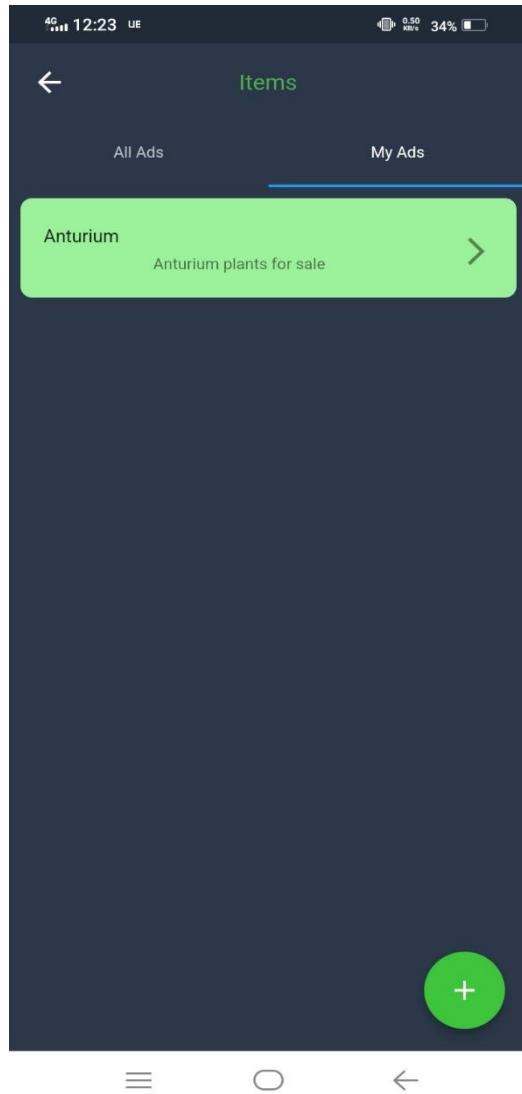
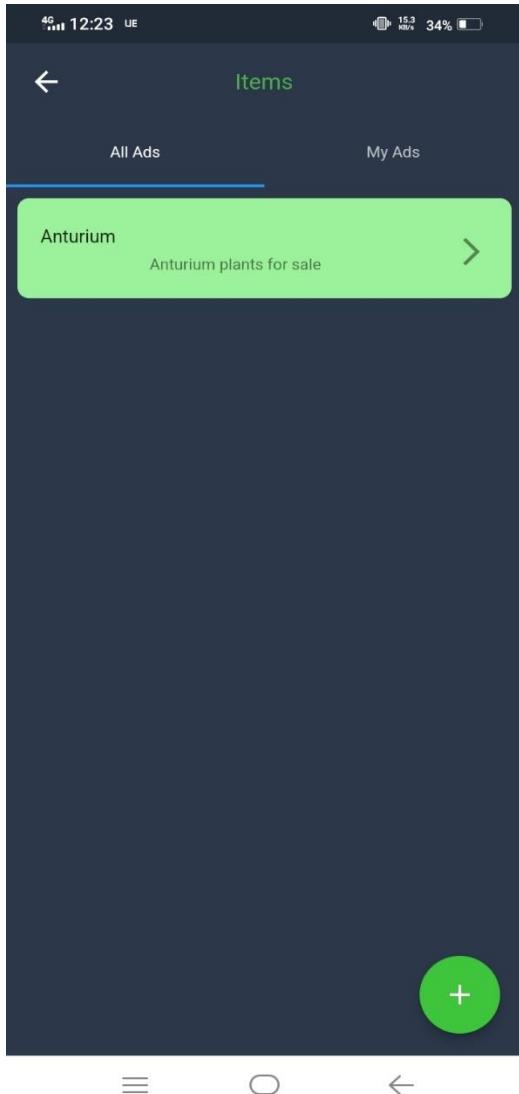
```
"location": location,
"quantity": quantity,
"phoneNumber": phoneNumber,
"description": description,
"signedUrl": signedUrl,
"user": user
};

await documentReferencer
.set(data)
.whenComplete(() => print("Note ads added to the database"))
.catchError((e) => print(e));
}
```

```
static Future<void> deleteAd({
    required String docId,
}) async {
    DocumentReference documentReferencer =
        _firestore.collection('ads').doc(docId);

    await documentReferencer
        .delete()
        .whenComplete(() => print('Note ad deleted from the database'))
        .catchError((e) => print(e));
}
```

## UI SCREENSHOTS





4G 09:52 15.3 kb/s 40% 🔋

4G 12:23 UE 0.00 kb/s 34% 🔋

← കമ ത Kamatha

Enter your location

Quantity

Enter your quantity

Phone Number

Enter your item phone number

Description

Enter your item description

+ UPLOAD IMAGE

Post



4G 12:23 UE

Edit Item

Location

Kurunegala

Quantity

2

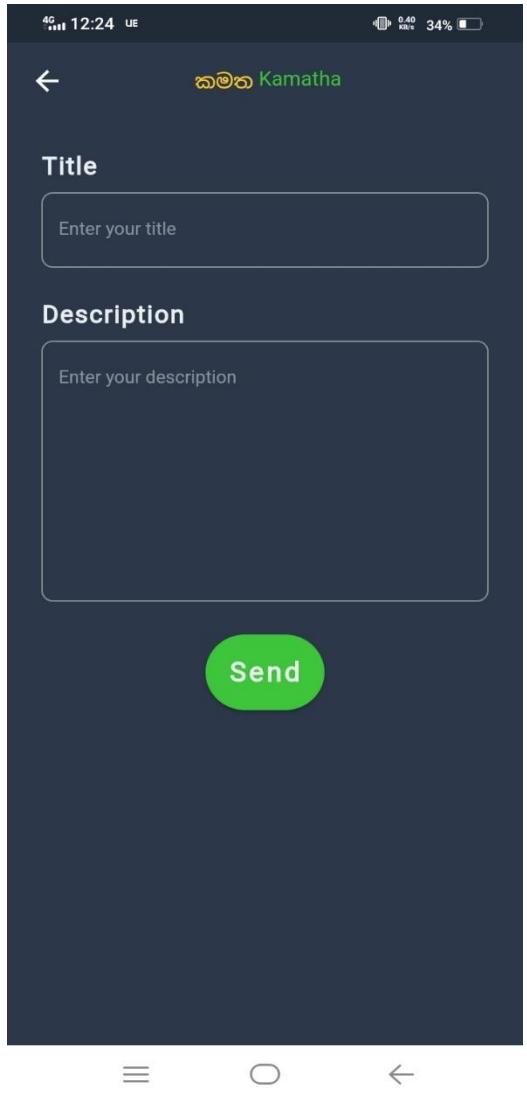
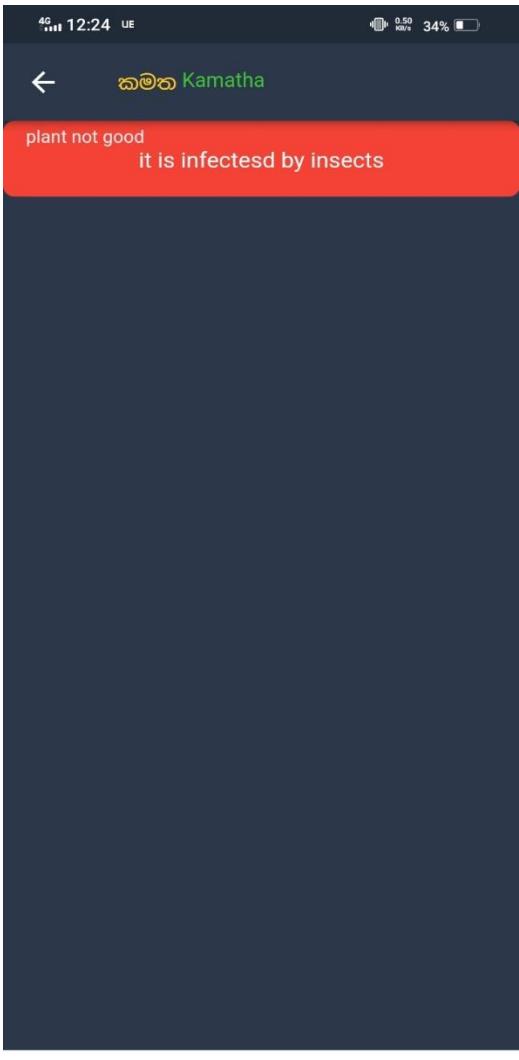
Phone Number

0777111753

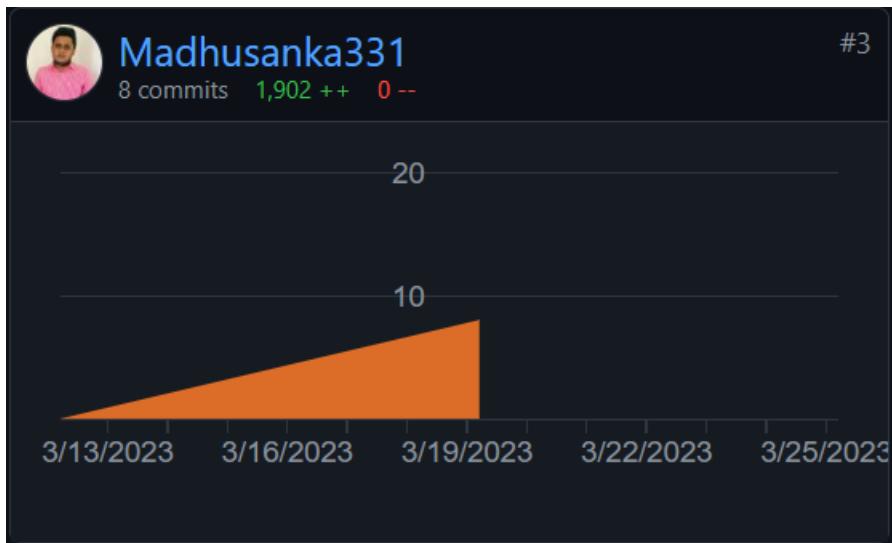
Description

Anturium plants for sale

Update



## SCREENSHOTS OF CONTRIBUTIONS



## Publish Agricultural Posts

In this component, farmer can log in to the system and post their experiences. Other farmers can see the posts and give their comments and like. There is a tab called account in the Drawer view. Farmer can edit / delete their posts and comments.

### CODE SNIPPETS

#### Screens

##### AddScreen.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/widgets/add_item_form.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class AddScreen extends StatefulWidget {
  const AddScreen({Key? key, required User user})
      : _user = user,
        super(key: key);

  final User _user;

  @override
  State<AddScreen> createState() => _AddScreenState();
}

class _AddScreenState extends State<AddScreen> {
  final FocusNode _titleFocusNode = FocusNode();
  final FocusNode _descriptionFocusNode = FocusNode();

  late User _user;

  @override
  void initState() {
    _user = widget._user;
```

```
super.initState();
}

@Override
Widget build(BuildContext context) {
    return GestureDetector(
        onTap: () {
            _titleFocusNode.unfocus();
            _descriptionFocusNode.unfocus();
        },
        child: Scaffold(
            backgroundColor: CustomColors.firebaseioNavy,
            appBar: AppBar(
                elevation: 0,
                backgroundColor: CustomColors.firebaseioNavy,
                title: AppBarTitle(),
                centerTitle: true,
            ),
            body: SafeArea(
                child: ListView(children: [
                    Padding(
                        padding: const EdgeInsets.only(
                            left: 16.0,
                            right: 16.0,
                            bottom: 20.0,
                        ),
                        child: AddItemForm(
                            titleFocusNode: _titleFocusNode,
                            descriptionFocusNode: _descriptionFocusNode,
                            user: _user,
                        ),
                    ),
                    ],
                )));
}
}
```

## CommentScreen.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/widgets/list_comments.dart';
```

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

class CommentScreen extends StatefulWidget {
  const CommentScreen({Key? key, required this.comment}) : super(key: key);

  final String comment;

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

class _CommentScreenState extends State<CommentScreen> {
  final String time = DateTime.now().toString().substring(0, 4);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: CustomColors.firebaseioNavy,
      appBar: AppBar(
        elevation: 0,
        backgroundColor: CustomColors.firebaseioNavy,
        title: Row(
          mainAxisAlignment: MainAxisAlignment.start,
          mainAxisSize: MainAxisSize.min,
          children: [
            const SizedBox(width: 8),
            Text(
              'Comments for : ',
              style: TextStyle(
                color: CustomColors.firebaseioYellow,
                fontSize: 18,
              ),
            ),
            Text(
              widget.comment,
              style: TextStyle(
                color: CustomColors.firebaseioOrange,
                fontSize: 18,
              ),
            ),
          ],
        ),
        centerTitle: true,
        // leading: IconButton(
        //   icon: const Icon(Icons.menu),

```

```

        // onPressed: () => {_buildDrawer(context)},
        // ),
    ),
    body: SafeArea(
        child: Padding(
            padding: const EdgeInsets.only(
                left: 16.0,
                right: 16.0,
                bottom: 20.0,
            ),
            child: ItemListComment(
                title: widget.comment,
            ),
        ),
    ),
    bottomNavigationBar: Padding(
        padding: const EdgeInsets.only(left: 90, bottom: 5),
        child: Text(
            "Copyright " + time + " @" + "Kamatha",
            style: TextStyle(color: CustomColors.firebaseioOrange),
        ),
    ),
);
}
}

```

## DisplayAll.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/comment_screen.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:Kamataha/src/widgets/text_input_widget.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';

class DisplayAllScreen extends StatefulWidget {
    final String currentTitle;
    final String currentDescription;
    final String name;
    final String documentId;
    final String time;
    final int likes;

```

```
final User _user;

const DisplayAllScreen(
    {Key? key,
    required this.currentTitle,
    required this.currentDescription,
    required this.name,
    required this.documentId,
    required this.time,
    required this.likes,
    required User user})
    : _user = user,
    super(key: key);

@Override
_DisplayAllScreenState createState() => _DisplayAllScreenState();
}

class _DisplayAllScreenState extends State<DisplayAllScreen> {
    late User _user;

    var isPressed = false;

    @override
    void initState() {
        _user = widget._user;
        super.initState();
    }

    void showToast(isPressed) {
        if (isPressed) {
            Fluttertoast.showToast(
                msg: 'You Liked This',
                toastLength: Toast.LENGTH_SHORT,
                gravity: ToastGravity.BOTTOM,
                timeInSecForIosWeb: 1,
                backgroundColor: Colors.red,
                textColor: Colors.yellow);
        } else {
            Fluttertoast.showToast(
                msg: 'Like Removed',
                toastLength: Toast.LENGTH_SHORT,
                gravity: ToastGravity.BOTTOM,
                timeInSecForIosWeb: 1,
                backgroundColor: Colors.red,
                textColor: Colors.white);
        }
    }
}
```

```
        textColor: Colors.yellow);
    }
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: CustomColors.firebaseioNavy,
        appBar: AppBar(
            elevation: 0,
            backgroundColor: CustomColors.firebaseioNavy,
            title: AppBarTitle(),
            centerTitle: true,
        ),
        floatingActionButton: Padding(
            padding: const EdgeInsets.only(bottom: 50),
            child: FloatingActionButton(
                backgroundColor: isPressed ? Colors.red : Colors.green,
                onPressed: () {
                    setState(() => isPressed = !isPressed);
                    showToast(isPressed);
                    if (isPressed) {
                        Database.updateLikes(
                            likes: widget.likes, docId: widget.documentId);
                    } else {
                        if (widget.likes == 0) {
                            return;
                        }
                        Database.decrementLikes(
                            likes: widget.likes, docId: widget.documentId);
                    }
                },
                child: const Icon(Icons.favorite),
            ),
        ),
        body: SafeArea(
            child: ListView(
                children: [
                    Column(
                        children: [
                            Center(
                                child: Padding(
                                    padding: const EdgeInsets.all(8.0),
                                    child: Text(
                                        "Title ↗ " + widget.currentTitle,
                                    ),
                                ),
                            ),
                        ],
                    ),
                ],
            ),
        ),
    );
}
```

```
        style: const TextStyle(
            color: Colors.white,
            fontSize: 20,
        ),
    ),
),
),
),
),
Center(
    child: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Text(
            "Posted by " + widget.name,
            style: const TextStyle(
                color: Colors.white,
                fontSize: 10,
            ),
        ),
    ),
),
),
),
),
Center(
    child: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Text(
            widget.time + " ❤️ " + widget.likes.toString(),
            style: const TextStyle(
                color: Colors.white,
                fontSize: 10,
            ),
        ),
    ),
),
),
),
),
),
Center(
    child: Padding(
        padding: const EdgeInsets.all(16.0),
        child: SelectableText(
            widget.currentDescription,
            style: const TextStyle(
                color: Colors.white,
                fontSize: 20,
                letterSpacing: 1,
                wordSpacing: 1,
            ),
        ),
    ),
),
),
),
),
ElevatedButton(
```

```
        onPressed: () {
            Navigator.of(context).push(MaterialPageRoute(
                builder: (context) => CommentScreen(
                    comment: widget.currentTitle,
                )));
        },
        child: const Text("See Comments"),
        style: ButtonStyle(
            backgroundColor: MaterialStateProperty.all(
                CustomColors.firebaseioOrange)),
        ),
        TextInputWidget(user: _user, title: widget.currentTitle)
    ],
),
],
],
),
),
);
}
}
```

### EditCommentScreen.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/widgets/edit_comment_form.dart';
import 'package:flutter/material.dart';

class EditCommentScreen extends StatefulWidget {
    final String currentTitle;
    final String comment;
    final String documentId;
    final String user;

    // ignore: use_key_in_widget_constructors
    const EditCommentScreen(
        {required this.currentTitle,
        required this.comment,
        required this.documentId,
        required this.user});

    @override
    _EditCommentScreenState createState() => _EditCommentScreenState();
}
```

```
class _EditCommentScreenState extends State<EditCommentScreen> {
    final FocusNode _titleFocusNode = FocusNode();

    final FocusNode _descriptionFocusNode = FocusNode();

    bool _isDeleting = false;

    @override
    Widget build(BuildContext context) {
        return GestureDetector(
            onTap: () {
                _titleFocusNode.unfocus();
                _descriptionFocusNode.unfocus();
            },
            child: Scaffold(
                backgroundColor: CustomColors.firebaseioNavy,
                appBar: AppBar(
                    elevation: 0,
                    backgroundColor: CustomColors.firebaseioNavy,
                    title: Row(
                        mainAxisAlignment: MainAxisAlignment.start,
                        mainAxisSize: MainAxisSize.min,
                        children: [
                            const SizedBox(width: 8),
                            Text(
                                widget.user.toString().substring(0, 10) + "...'s",
                                style: TextStyle(
                                    color: CustomColors.firebaseioYellow,
                                    fontSize: 18,
                                ),
                            ),
                            Text(
                                ' Comments ',
                                style: TextStyle(
                                    color: CustomColors.firebaseioOrange,
                                    fontSize: 18,
                                ),
                            ),
                        ],
                ),
                centerTitle: true,
                actions: [
                    _isDeleting
                        ? const Padding(

```

```
padding: EdgeInsets.only(
    top: 10.0,
    bottom: 10.0,
    right: 16.0,
),
child: CircularProgressIndicator(
    valueColor: AlwaysStoppedAnimation<Color>(
        Colors.redAccent,
    ),
    strokeWidth: 3,
),
)
: IconButton(
    icon: const Icon(
        Icons.delete,
        color: Colors.redAccent,
        size: 32,
    ),
    onPressed: () async {
        setState(() {
            _isDeleting = true;
        });
        await Database.deleteComment(
            docId: widget.documentId,
        );
        setState(() {
            _isDeleting = false;
        });
        Navigator.of(context).pop();
    },
),
],
),
body: SafeArea(
    child: Padding(
        padding: const EdgeInsets.only(
            left: 16.0,
            right: 16.0,
            bottom: 20.0,
        ),
        child: EditCommentForm(
            documentId: widget.documentId,
```

```

        titleFocusNode: _titleFocusNode,
        descriptionFocusNode: _descriptionFocusNode,
        currentTitle: widget.currentTitle,
        comment: widget.comment,
        ),
        ),
        ),
        ),
        );
    }
}

```

### EditScreen.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/widgets/edit_item_form.dart';
import 'package:flutter/material.dart';

class EditScreen extends StatefulWidget {
    final String currentTitle;
    final String currentDescription;
    final String documentId;
    final String user;

    // ignore: use_key_in_widget_constructors
    const EditScreen(
        {required this.currentTitle,
        required this.currentDescription,
        required this.documentId,
        required this.user});

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

class _EditScreenState extends State<EditScreen> {
    final FocusNode _titleFocusNode = FocusNode();

    final FocusNode _descriptionFocusNode = FocusNode();

    bool _isDeleting = false;

    @override

```

```
Widget build(BuildContext context) {
  return GestureDetector(
    onTap: () {
      _titleFocusNode.unfocus();
      _descriptionFocusNode.unfocus();
    },
    child: Scaffold(
      backgroundColor: CustomColors.firebaseioNavy,
      appBar: AppBar(
        elevation: 0,
        backgroundColor: CustomColors.firebaseioNavy,
        title: Row(
          mainAxisAlignment: MainAxisAlignment.start,
          mainAxisSize: MainAxisSize.min,
          children: [
            const SizedBox(width: 8),
            Text(
              widget.user.toString().substring(0, 10) + "...'s",
              style: TextStyle(
                color: CustomColors.firebaseioYellow,
                fontSize: 18,
              ),
            ),
            Text(
              ' Feelings ',
              style: TextStyle(
                color: CustomColors.firebaseioOrange,
                fontSize: 18,
              ),
            ),
          ],
        ),
        centerTitle: true,
        actions: [
          _isDeleting
            ? const Padding(
                padding: EdgeInsets.only(
                  top: 10.0,
                  bottom: 10.0,
                  right: 16.0,
                ),
            ),
            child: CircularProgressIndicator(
              valueColor: AlwaysStoppedAnimation<Color>(
                Colors.redAccent,
              ),
            ),
        ],
      ),
    ),
  );
}
```

```
        strokeWidth: 3,
    ),
)
: IconButton(
    icon: const Icon(
        Icons.delete,
        color: Colors.redAccent,
        size: 32,
    ),
    onPressed: () async {
        setState(() {
            _isDeleting = true;
        });
        await Database.deleteItem(
            docId: widget.documentId,
        );
        setState(() {
            _isDeleting = false;
        });
        Navigator.of(context).pop();
    },
),
],
),
body: SafeArea(
    child: Padding(
        padding: const EdgeInsets.only(
            left: 16.0,
            right: 16.0,
            bottom: 20.0,
        ),
        child: EditItemForm(
            documentId: widget.documentId,
            titleFocusNode: _titleFocusNode,
            descriptionFocusNode: _descriptionFocusNode,
            currentTitle: widget.currentTitle,
            currentDescription: widget.currentDescription,
        ),
    ),
),
),
),
),
);
```

```
    }
}
```

## Posts.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/add_screen.dart';
import 'package:Kamataha/src/widgets/item_list.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class Posts extends StatefulWidget {
  const Posts({Key? key, required User user})
      : _user = user,
        super(key: key);

  final User _user;

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

class _PostsState extends State<Posts> {
  late User _user;

  @override
  void initState() {
    _user = widget._user;
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: CustomColors.firebaseioNavy,
      body: ItemList(user: _user),
      floatingActionButton: FloatingActionButton(
        child: const Icon(Icons.add),
        backgroundColor: CustomColors.firebaseioOrange,
        onPressed: () {
          Navigator.of(context).push(
            MaterialPageRoute(
              builder: (context) => AddScreen(user: _user),
            ),
          );
        },
      ),
    );
  }
}
```

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

### PrivateComments.dart

```
import 'package:Kamataha/src/widgets/private_comment_list.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class PrivateComments extends StatefulWidget {
  const PrivateComments({Key? key, required User user})
      : _user = user,
        super(key: key);

  final User _user;

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

class _PrivateCommentsState extends State<PrivateComments> {
  late User _user;

  @override
  void initState() {
    _user = widget._user;
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Padding(
        padding: const EdgeInsets.only(
          left: 16.0,
          right: 16.0,
          bottom: 20.0,
        ),
        child: PrivateCommentList(
          user: _user,
        ),
      ),
    );
  }
}
```

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

## PrivateScreen.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/private_comments.dart';
import 'package:Kamataha/src/screens/privateFeelings.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'add_screen.dart';

class PrivateScreen extends StatefulWidget {
    const PrivateScreen({Key? key, required User user})
        : _user = user,
        super(key: key);

    final User _user;

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

class _PrivateScreenState extends State<PrivateScreen> {
    late User _user;

    @override
    void initState() {
        _user = widget._user;
        super.initState();
    }

    @override
    Widget build(BuildContext context) {
        return DefaultTabController(
            length: 2,
            child: Scaffold(
                backgroundColor: CustomColors.firebaseioNavy,
                appBar: AppBar(
                    title: Row(
                        crossAxisAlignment: CrossAxisAlignment.start,
```

```
mainAxisSize: MainAxisSize.min,
children: [
    const SizedBox(width: 8),
    Text(
        _user.displayName.toString().substring(0, 10) + "...'s",
        style: TextStyle(
            color: CustomColors.firebaseioYellow,
            fontSize: 18,
        ),
    ),
    Text(
        ' Profile ',
        style: TextStyle(
            color: CustomColors.firebaseioOrange,
            fontSize: 18,
        ),
    ),
],
),
backgroundColor: CustomColors.firebaseioNavy,
elevation: 0,
centerTitle: true,
bottom: const TabBar(
    tabs: [
        Tab(
            text: "Posts",
            icon: Icon(Icons.favorite),
        ),
        Tab(
            text: "Comments",
            icon: Icon(Icons.comment),
        ),
    ],
)),
floatingActionButton: FloatingActionButton(
    child: const Icon(Icons.add),
    backgroundColor: CustomColors.firebaseioOrange,
    onPressed: () {
        Navigator.of(context).push(
            MaterialPageRoute(
                builder: (context) => AddScreen(user: _user),
            ),
        );
    },
),
```

```
        body: TabBarView(
            children: [
                PrivateFeelings(user: _user),
                PrivateComments(user: _user),
            ],
        ),
    ),
);
}
}
```

## Sign\_in\_screen.dart

```
// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/auth.dart';
import 'package:Kamataha/src/widgets/google_sign_in_button.dart';
import 'package:flutter/material.dart';
import 'dart:ui' as ui;

class SignInScreen extends StatefulWidget {
    @override
    _SignInScreenState createState() => _SignInScreenState();
}

class _SignInScreenState extends State<SignInScreen> {
    @override
    Widget build(BuildContext context) {
        return Container(
            decoration: const BoxDecoration(
                image: DecorationImage(
                    image: AssetImage("assets/background.jpg"),
                    fit: BoxFit.cover,
                ),
            ),
            child: Scaffold(
                backgroundColor: Colors.black.withOpacity(0.5),
                body: SafeArea(
                    child: Padding(
                        padding: const EdgeInsets.only(
                            left: 16.0,
                            right: 16.0,
                            bottom: 20.0,
                        ),

```

```
),
child: Column(
  mainAxisAlignment: MainAxisAlignment.max,
  children: [
    Row(),
    Expanded(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.min,
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          const SizedBox(height: 320),
          Flexible(
            flex: 1,
            child: Image.asset(
              'assets/logo.png',
              height: 100,
            ),
          ),
        ],
        const SizedBox(height: 20),
        Text(
          "କମାତ୍ର",
          style: TextStyle(
            fontWeight: FontWeight.bold,
            fontSize: 40,
            foreground: Paint()
              ..shader = ui.Gradient.linear(
                const Offset(0, 50),
                const Offset(150, 20),
                <Color>[
                  Colors.red,
                  Colors.yellow,
                ],
              )),
        ),
        const Padding(
          padding: EdgeInsets.only(top: 5),
          child: Text(
            "Kamatha",
            style: TextStyle(
              color: Colors.lightGreenAccent,
              fontSize: 30,
              fontStyle: FontStyle.italic,
            ),
          ),
        ),
      ),
    ),
  ],
),
```

```
        const Padding(
            padding: EdgeInsets.only(top: 50),
            child: Text(
                'Powered by Kamatha',
                style: TextStyle(
                    color: Colors.white70,
                    fontSize: 15,
                ),
            ),
        ),
    ],
),
),
),
),
FutureBuilder(
    future: Authentication.initializeApp(context: context),
    builder: (context, snapshot) {
        if (snapshot.hasError) {
            return const Text('Error initializing Firebase');
        } else if (snapshot.connectionState ==
            ConnectionState.done) {
            return GoogleSignInButton();
        }
        return CircularProgressIndicator(
            valueColor: AlwaysStoppedAnimation<Color>(
                CustomColors.firebaseioOrange,
            ),
        );
    },
),
],
),
),
));
}
}
```

## Widgets

### AddItemForm.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'custom_form_field.dart';

// ignore: must_be_immutable
class AddItemForm extends StatefulWidget {
    final FocusNode titleFocusNode;
    final FocusNode descriptionFocusNode;

    const AddItemForm({
        Key? key,
        required this.titleFocusNode,
        required this.descriptionFocusNode,
        required User user,
    }) : _user = user,
        super(key: key);

    final User _user;

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

class _AddItemFormState extends State<AddItemForm> {
    final _addItemFormKey = GlobalKey<FormState>();

    late User _user;

    bool _isProcessing = false;

    final TextEditingController _titleController = TextEditingController();
    final TextEditingController _descriptionController = TextEditingController();

    @override
    void initState() {
        _user = widget._user;
        super.initState();
    }
}
```

```
@override
Widget build(BuildContext context) {
  return Form(
    key: _addItemFormKey,
    child: Column(
      children: [
        Padding(
          padding: const EdgeInsets.only(
            left: 8.0,
            right: 8.0,
            bottom: 24.0,
          ),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              const SizedBox(height: 24.0),
              Text(
                'Title',
                style: TextStyle(
                  color: CustomColors.firebaseioGrey,
                  fontSize: 22.0,
                  letterSpacing: 1,
                  fontWeight: FontWeight.bold,
                ),
              ),
              const SizedBox(height: 8.0),
              CustomFormField(
                isLabelEnabled: false,
                controller: _titleController,
                focusNode: widget.titleFocusNode,
                keyboardType: TextInputType.text,
                inputAction: TextInputAction.next,
                validator: (value) => Validator.validateField(
                  value: value,
                ),
                label: 'Title',
                hint: 'Enter your title',
              ),
              const SizedBox(height: 24.0),
              Text(
                'Experience / Research',
                style: TextStyle(
                  color: CustomColors.firebaseioGrey,
                  fontSize: 22.0,
                ),
              ),
            ],
          ),
        ),
      ],
    ),
  );
}
```

```
        letterSpacing: 1,
        fontWeight: FontWeight.bold,
    ),
),
const SizedBox(height: 8.0),
CustomFormField(
    maxLines: 10,
    isLabelEnabled: false,
    controller: _descriptionController,
    focusNode: widget.descriptionFocusNode,
    keyboardType: TextInputType.text,
    inputAction: TextInputAction.done,
    validator: (value) => Validator.validateField(
        value: value,
    ),
    label: 'Experience / Research',
    hint: 'Enter your feeling',
),
],
),
),
),
_isProcessing
? Padding(
    padding: const EdgeInsets.all(16.0),
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
    ),
)
: ElevatedButton(
    style: ButtonStyle(
        backgroundColor: MaterialStateProperty.all(
            CustomColors.firebaseioOrange,
        ),
        shape: MaterialStateProperty.all(
            RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(45),
            ),
        ),
    ),
    onPressed: () async {
        widget.titleFocusNode.unfocus();
        widget.descriptionFocusNode.unfocus();
    },
);
```

```

        if (_addItemFormKey.currentState!.validate()) {
            setState(() {
                _isProcessing = true;
            });

            await Database.addItem(
                title: _titleController.text,
                description: _descriptionController.text,
                user: _user.displayName.toString());

            setState(() {
                _isProcessing = false;
            });

            Navigator.of(context).pop();
        }
    },
    child: Padding(
        padding: const EdgeInsets.only(top: 16.0, bottom: 16.0),
        child: Text(
            'Post Your Experience',
            style: TextStyle(
                fontSize: 24,
                fontWeight: FontWeight.bold,
                color: CustomColors.firebaseioGrey,
                letterSpacing: 2,
            ),
            ),
            ),
            ),
            ),
            ),
            ],
            ),
        );
    );
}
}

```

## AppBarTitle.dart

```

// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:flutter/material.dart';

```

```
class AppBarTitle extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Row(
      mainAxisAlignment: MainAxisAlignment.start,
      mainAxisSize: MainAxisSize.min,
      children: [
        const SizedBox(width: 8),
        Text(
          'କମତା',
          style: TextStyle(
            color: CustomColors.firebaseioYellow,
            fontSize: 18,
          ),
        ),
        Text(
          ' Kamatha ',
          style: TextStyle(
            color: CustomColors.firebaseioOrange,
            fontSize: 18,
          ),
        ),
      ],
    );
  }
}
```

## CustomFormField.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:flutter/material.dart';

class CustomFormField extends StatelessWidget {
  const CustomFormField({
    Key? key,
    required TextEditingController controller,
    required FocusNode focusNode,
    required TextInputType keyboardType,
    required TextInputAction inputAction,
    required String label,
    required String hint,
    required Function(String value) validator,
    this.isObscure = false,
```

```
this.isCapitalized = false,
this.maxLines = 1,
this.isLabelEnabled = true,
}) : _emailController = controller,
_emailFocusNode = focusNode,
_keyboardtype = keyboardType,
_inputAction = inputAction,
_label = label,
_hint = hint,
_validator = validator,
super(key: key);

final TextEditingController _emailController;
final FocusNode _emailFocusNode;
final TextInputType _keyboardtype;
final TextInputAction _inputAction;
final String _label;
final String _hint;
final bool isObscure;
final bool isCapitalized;
final int maxLines;
final bool isLabelEnabled;
final Function(String) _validator;

@Override
Widget build(BuildContext context) {
    return TextFormField(
        maxLines: maxLines,
        controller: _emailController,
        focusNode: _emailFocusNode,
        keyboardType: _keyboardtype,
        obscureText: isObscure,
        style: const TextStyle(
            color: Colors.white,
        ),
        textCapitalization:
            isCapitalized ? TextCapitalization.words : TextCapitalization.none,
       textInputAction: _inputAction,
        cursorColor: CustomColors.firebaseioYellow,
        validator: (value) {
            if (value == null || value.isEmpty) {
                return 'Please fill this field.';
            }
        },
        decoration: InputDecoration(
```

```
labelText: isLabelEnabled ? _label : null,
fillColor: Colors.white,
labelStyle: TextStyle(color: CustomColors.firebaseioYellow),
hintText: _hint,
hintStyle: TextStyle(
    color: CustomColors.firebaseioGrey.withOpacity(0.5),
),
errorStyle: const TextStyle(
    color: Colors.redAccent,
    fontWeight: FontWeight.bold,
),
focusedBorder: OutlineInputBorder(
    borderRadius: BorderRadius.circular(8.0),
    borderSide: BorderSide(
        color: Color(0xFF9CF29A),
        width: 2,
    ),
),
enabledBorder: OutlineInputBorder(
    borderRadius: BorderRadius.circular(8.0),
    borderSide: BorderSide(
        color: CustomColors.firebaseioGrey.withOpacity(0.5),
    ),
),
errorBorder: OutlineInputBorder(
    borderRadius: BorderRadius.circular(8.0),
    borderSide: const BorderSide(
        color: Colors.redAccent,
        width: 2,
    ),
),
focusedErrorBorder: OutlineInputBorder(
    borderRadius: BorderRadius.circular(8.0),
    borderSide: const BorderSide(
        color: Colors.redAccent,
        width: 2,
    ),
),
),
),
);
}
}
```

## Drawer.dart

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

class Drawer extends StatefulWidget {
  const Drawer({Key? key, required ListView child}) : super(key: key);

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

class _DrawerState extends State<Drawer> {
  @override
  Widget build(BuildContext context) {
    return Drawer(
      child: ListView(
        padding: EdgeInsets.zero,
        children: [
          const UserAccountsDrawerHeader(
            currentAccountPicture: CircleAvatar(
              backgroundImage: NetworkImage(
                'https://images.unsplash.com/photo-1485290334039-a3c69043e517?crop=entropy&cs=tinysrgb&fit=max&fm=jpg&ixid=MnwxMD8MXxyYw5kb218MHx8fHx8fHx8MTYyOTU3NDE0MQ&ixlib=rb-1.2.1&q=80&utm_campaign=api-credit&utm_medium=referral&utm_source=unsplash_source&w=300'),
            ),
            accountEmail: Text('jane.doe@example.com'),
            accountName: Text(
              'Jane Doe',
              style: TextStyle(fontSize: 24.0),
            ),
            decoration: BoxDecoration(
              color: Colors.black87,
            ),
          ),
          ListTile(
            leading: const Icon(Icons.house),
            title: const Text(
              'Houses',
              style: TextStyle(fontSize: 24.0),
            ),
            onTap: () {},
          ),
          ListTile(
            leading: const Icon(Icons.apartment),
```

```
        title: const Text(
            'Apartments',
            style: TextStyle(fontSize: 24.0),
        ),
        onTap: () {
            // Navigator.pushReplacement(
            //     context,
            //     MaterialPageRoute<void>(
            //         builder: (BuildContext context) => const MyHomePage(
            //             title: 'Apartments',
            //         ),
            //     ),
            // );
        },
    ),
    ListTile(
        leading: const Icon(Icons.house_outlined),
        title: const Text(
            'Townhomes',
            style: TextStyle(fontSize: 24.0),
        ),
        onTap: () {
            // Navigator.pushReplacement(
            //     context,
            //     MaterialPageRoute<void>(
            //         builder: (BuildContext context) => const MyHomePage(
            //             title: 'Townhomes',
            //         ),
            //     ),
            // );
        },
    ),
    const Divider(
        height: 10,
        thickness: 1,
    ),
    ListTile(
        leading: const Icon(Icons.favorite),
        title: const Text(
            'Favorites',
            style: TextStyle(fontSize: 24.0),
        ),
        onTap: () {
            // Navigator.pushReplacement(
            //     context,
```

```

        //    MaterialPageRoute<void>(
        //      builder: (BuildContext context) => const MyHomePage(
        //        title: 'Favorites',
        //      ),
        //    ),
        //  );
        //),
      ],
    ),
  );
}
}

```

## EditCommentForm.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:flutter/material.dart';

import 'custom_form_field.dart';

class EditCommentForm extends StatefulWidget {
  final FocusNode titleFocusNode;
  final FocusNode descriptionFocusNode;
  final String currentTitle;
  final String comment;
  final String documentId;

  const EditCommentForm({
    required this.titleFocusNode,
    required this.descriptionFocusNode,
    required this.currentTitle,
    required this.comment,
    required this.documentId,
  });

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

class _EditCommentFormState extends State<EditCommentForm> {

```

```
final _EditCommentFormKey = GlobalKey<FormState>();  
  
bool _isProcessing = false;  
  
late TextEditingController _titleController;  
late TextEditingController _descriptionController;  
  
@override  
void initState() {  
    _titleController = TextEditingController(  
        text: widget.currentTitle,  
    );  
  
    _descriptionController = TextEditingController(  
        text: widget.comment,  
    );  
    super.initState();  
}  
  
@override  
Widget build(BuildContext context) {  
    return Form(  
        key: _EditCommentFormKey,  
        child: ListView(  
            children: [  
                Padding(  
                    padding: const EdgeInsets.only(  
                        left: 8.0,  
                        right: 8.0,  
                        bottom: 24.0,  
                    ),  
                child: Column(  
                    crossAxisAlignment: CrossAxisAlignment.start,  
                    children: [  
                        const SizedBox(height: 24.0),  
                        Text(  
                            'Title',  
                            style: TextStyle(  
                                color: CustomColors.firebaseioGrey,  
                                fontSize: 22.0,  
                                letterSpacing: 1,  
                                fontWeight: FontWeight.bold,  
                            ),  
                        ),  
                        const SizedBox(height: 8.0),  
                    ],  
                ),  
            ],  
        ),  
    );  
}
```

```
    TextFormField(
      controller: _titleController,
      focusNode: widget.titleFocusNode,
      keyboardType: TextInputType.text,
      enabled: false,
      readOnly: true,
      style: const TextStyle(
        color: Colors.white,
      ),
    ),
    const SizedBox(height: 24.0),
    Text(
      'Comment',
      style: TextStyle(
        color: CustomColors.firebaseioGrey,
        fontSize: 22.0,
        letterSpacing: 1,
        fontWeight: FontWeight.bold,
      ),
    ),
    const SizedBox(height: 8.0),
    CustomFormField(
      maxLines: 10,
      isLabelEnabled: false,
      controller: _descriptionController,
      focusNode: widget.descriptionFocusNode,
      keyboardType: TextInputType.text,
      inputAction: TextInputAction.done,
      validator: (value) => Validator.validateField(
        value: value,
      ),
      label: 'Comment',
      hint: 'Enter your note description',
    ),
  ],
),
),
),
_isProcessing
? Padding(
  padding: const EdgeInsets.all(16.0),
  child: CircularProgressIndicator(
    valueColor: AlwaysStoppedAnimation<Color>(
      CustomColors.firebaseioOrange,
    ),
  ),
),
```

```
)  
: Padding(  
    padding: const EdgeInsets.only(  
        left: 30,  
        right: 30,  
    ),  
    child: ElevatedButton(  
        style: ButtonStyle(  
            backgroundColor: MaterialStateProperty.all(  
                CustomColors.firebaseioOrange,  
            ),  
            minimumSize: MaterialStateProperty.all(Size.zero),  
            shape: MaterialStateProperty.all(  
                RoundedRectangleBorder(  
                    borderRadius: BorderRadius.circular(45),  
                ),  
            ),  
        ),  
        onPressed: () async {  
            widget.titleFocusNode.unfocus();  
            widget.descriptionFocusNode.unfocus();  
  
            if (_EditCommentFormKey.currentState!.validate()) {  
                setState(() {  
                    _isProcessing = true;  
                });  
  
                await Database.updateComments(  
                    docId: widget.documentId,  
                    comment: _descriptionController.text,  
                );  
  
                setState(() {  
                    _isProcessing = false;  
                });  
  
                Navigator.of(context).pop();  
            }  
        },  
        child: Padding(  
            padding: const EdgeInsets.only(top: 16.0, bottom: 16.0),  
            child: Text(  
                'Update Comment',  
                style: TextStyle(  
                    fontSize: 24,
```

```

        fontWeight: FontWeight.bold,
        color: CustomColors.firebaseioGrey,
        letterSpacing: 2,
    ),
),
),
),
),
),
),
],
),
);
}
}

```

## EditItemForm.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:flutter/material.dart';
import 'custom_form_field.dart';

class EditItemForm extends StatefulWidget {
    final FocusNode titleFocusNode;
    final FocusNode descriptionFocusNode;
    final String currentTitle;
    final String currentDescription;
    final String documentId;

    const EditItemForm({
        required this.titleFocusNode,
        required this.descriptionFocusNode,
        required this.currentTitle,
        required this.currentDescription,
        required this.documentId,
    });

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

class _EditItemFormState extends State<EditItemForm> {
    final _editItemFormKey = GlobalKey<FormState>();

```

```
bool _isProcessing = false;

late TextEditingController _titleController;
late TextEditingController _descriptionController;

@Override
void initState() {
    _titleController = TextEditingController(
        text: widget.currentTitle,
    );

    _descriptionController = TextEditingController(
        text: widget.currentDescription,
    );
    super.initState();
}

@Override
Widget build(BuildContext context) {
    return Form(
        key: _editItemFormKey,
        child: ListView(
            children: [
                Padding(
                    padding: const EdgeInsets.only(
                        left: 8.0,
                        right: 8.0,
                        bottom: 24.0,
                    ),
                    child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                            const SizedBox(height: 24.0),
                            Text(
                                'Title',
                                style: TextStyle(
                                    color: CustomColors.firebaseioGrey,
                                    fontSize: 22.0,
                                    letterSpacing: 1,
                                    fontWeight: FontWeight.bold,
                                ),
                            ),
                            const SizedBox(height: 8.0),
                            CustomFormField(

```

```
        isLabelEnabled: false,
        controller: _titleController,
        focusNode: widget.titleFocusNode,
        keyboardType: TextInputType.text,
        inputAction: TextInputAction.next,
        validator: (value) => Validator.validateField(
            value: value,
        ),
        label: 'Title',
        hint: 'Enter your note title',
    ),
    const SizedBox(height: 24.0),
    Text(
        'Description',
        style: TextStyle(
            color: CustomColors.firebaseioGrey,
            fontSize: 22.0,
            letterSpacing: 1,
            fontWeight: FontWeight.bold,
        ),
    ),
    const SizedBox(height: 8.0),
    CustomFormField(
        maxLines: 10,
        isLabelEnabled: false,
        controller: _descriptionController,
        focusNode: widget.descriptionFocusNode,
        keyboardType: TextInputType.text,
        inputAction: TextInputAction.done,
        validator: (value) => Validator.validateField(
            value: value,
        ),
        label: 'Description',
        hint: 'Enter your note description',
    ),
],
),
),
),
_isProcessing
? Padding(
    padding: const EdgeInsets.all(16.0),
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
    ),

```

```
        ),
    )
: Padding(
    padding: const EdgeInsets.only(
        left: 30,
        right: 30,
    ),
    child: ElevatedButton(
        style: ButtonStyle(
            backgroundColor: MaterialStateProperty.all(
                CustomColors.firebaseioOrange,
            ),
            minimumSize: MaterialStateProperty.all(Size.zero),
            shape: MaterialStateProperty.all(
                RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(45),
                ),
            ),
        ),
    ),
    onPressed: () async {
        widget.titleFocusNode.unfocus();
        widget.descriptionFocusNode.unfocus();

        if (_editItemFormKey.currentState!.validate()) {
            setState(() {
                _isProcessing = true;
            });

            await Database.updateItem(
                docId: widget.documentId,
                title: _titleController.text,
                description: _descriptionController.text,
            );

            setState(() {
                _isProcessing = false;
            });

            Navigator.of(context).pop();
        }
    },
    child: Padding(
        padding: const EdgeInsets.only(top: 16.0, bottom: 16.0),
        child: Text(
            'Update Experience',

```

```

        style: TextStyle(
            fontSize: 24,
            fontWeight: FontWeight.bold,
            color: CustomColors.firebaseioGrey,
            letterSpacing: 2,
        ),
    ),
),
),
),
),
),
),
),
],
),
);
}
}

```

## ItemList.dart

```

// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/display_all.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class ItemList extends StatefulWidget {
    const ItemList({required User user}) : _user = user;
    final User _user;
    @override
    State<ItemList> createState() => _ItemListState();
}

class _ItemListState extends State<ItemList> {
    late User _user;

    @override
    void initState() {
        _user = widget._user;
        super.initState();
    }
}

```

```
@override
Widget build(BuildContext context) {
  return StreamBuilder<QuerySnapshot>(
    stream: Database.readItems(),
    builder: (context, snapshot) {
      if (snapshot.hasError) {
        return const Text('Something went wrong');
      } else if (snapshot.hasData || snapshot.data != null) {
        if (snapshot.data!.docs.isEmpty) {
          return Center(
            child: Text(
              "No posts yet",
              style: TextStyle(color: CustomColors.firebaseioGrey),
            ),
          );
        }
      }
      return ListView.separated(
        separatorBuilder: (context, index) => const SizedBox(height: 16.0),
        itemCount: snapshot.data!.docs.length,
        itemBuilder: (context, index) {
          var noteInfo = snapshot.data!.docs[index].data() as dynamic;
          String docID = snapshot.data!.docs[index].id;
          String title = noteInfo["title"];
          String description = noteInfo['description'];
          String user = noteInfo['user'];
          String time = noteInfo['time'];
          int likes = noteInfo['likes'];

          return Padding(
            padding: const EdgeInsets.all(8.0),
            child: Ink(
              decoration: BoxDecoration(
                color: Color(0xFF9CF29A),
                borderRadius: BorderRadius.circular(8.0),
              ),
              child: ListTile(
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(8.0),
                ),
                onTap: () => Navigator.of(context).push(
                  MaterialPageRoute(
                    builder: (context) => DisplayAllScreen(
                      currentTitle: title,
                      currentDescription: description,
                      documentId: docID,

```

```

        name: user,
        time: time,
        user: _user,
        likes: likes,
    ),
),
),
),
title: Text(
    title,
    maxLines: 1,
    overflow: TextOverflow.ellipsis,
),
subtitle: Column(
    children: [
        Text(
            description,
            maxLines: 2,
            overflow: TextOverflow.ellipsis,
        ),
        const Text(""),
        Text(
            user +
            " : " +
            time.substring(0, 20) +
            " ❤️ " +
            likes.toString(),
            maxLines: 1,
            overflow: TextOverflow.ellipsis,
            style: const TextStyle(fontSize: 10),
        ),
    ],
),
trailing: const Padding(
    padding: EdgeInsets.only(top: 5),
    child: Icon(Icons.arrow_forward_ios_sharp),
),
),
));
},
);
}
}

return Center(
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(

```

```
        CustomColors.firebaseioOrange,  
        ),  
        ),  
    );  
},  
);  
}  
}
```

## ListComments.dart

```
// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';

class ItemListComment extends StatefulWidget {
  const ItemListComment({required this.title});
  final String title;
  @override
  State<ItemListComment> createState() => _ItemListCommentState();
}

class _ItemListCommentState extends State<ItemListComment> {
  //ADDED
  var colors = [
    Colors.red,
    Colors.blue,
    Colors.cyan,
    Colors.green,
    Colors.yellow,
  ];
  @override
  Widget build(BuildContext context) {
    return StreamBuilder<QuerySnapshot>(
      stream: Database.readComments(widget.title),
      builder: (context, snapshot) {
        if (snapshot.hasError) {
          return const Text('Something went wrong');
        } else if (snapshot.hasData || snapshot.data != null) {
          if (snapshot.data!.docs.isEmpty) {
            return const Text('No comments available');
          } else {
            return ListView.builder(
              itemCount: snapshot.data!.docs.length,
              itemBuilder: (context, index) {
                DocumentSnapshot document = snapshot.data!.docs[index];
                Map<String, dynamic> data = document.data();
                String comment = data['comment'];
                String timestamp = data['timestamp'].toDate().toString();
                return ListTile(
                  title: Text(comment),
                  subtitle: Text(timestamp),
                );
              },
            );
          }
        }
      },
    );
  }
}
```

```
return Center(
    child: Text(
        "No comments yet",
        style: TextStyle(color: CustomColors.firebaseioGrey),
    ),
);
}
return ListView.separated(
    separatorBuilder: (context, index) => const SizedBox(height: 16.0),
    itemCount: snapshot.data!.docs.length,
    itemBuilder: (context, index) {
        var noteInfo = snapshot.data!.docs[index].data() as dynamic;
        String user = noteInfo["user"];
        String comment = noteInfo['comment'];
        String time = noteInfo['time'];

        return Ink(
            decoration: BoxDecoration(
                color: colors[index],
                borderRadius: BorderRadius.circular(8.0),
            ),
            child: ListTile(
                shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(8.0),
                ),
                title: Text(
                    user + " says :",
                    maxLines: 1,
                    overflow: TextOverflow.ellipsis,
                    style: TextStyle(color: CustomColors.firebaseioGrey),
                ),
                subtitle: Column(
                    children: [
                        Text(
                            comment,
                            maxLines: 2,
                            overflow: TextOverflow.ellipsis,
                            style: const TextStyle(
                                color: Colors.white,
                                fontSize: 20,
                            ),
                        ),
                    ],
                ),
                const Text(""),
                Text(
                    time.substring(0, 20),
                ),
            ),
        );
    },
);
```

```

        maxLines: 1,
        overflow: TextOverflow.ellipsis,
        style: const TextStyle(fontSize: 10),
    ),
],
),
),
),
);
},
);
}
}

return Center(
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
),
);
},
);
},
);
}
}
}

```

## PrivateCommentList.dart

```

// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/edit_comment_screen.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class PrivateCommentList extends StatefulWidget {
    const PrivateCommentList({required User user}) : _user = user;
    final User _user;
    @override
    State<PrivateCommentList> createState() => _PrivateCommentListState();
}

class _PrivateCommentListState extends State<PrivateCommentList> {

```

```
late User _user;

@Override
void initState() {
    _user = widget._user;
    super.initState();
}

@Override
Widget build(BuildContext context) {
    return StreamBuilder<QuerySnapshot>(
        stream: Database.readPrivateComments(_user.displayName),
        builder: (context, snapshot) {
            if (snapshot.hasError) {
                return const Text('Something went wrong');
            } else if (snapshot.hasData || snapshot.data != null) {
                if (snapshot.data!.docs.isEmpty) {
                    return Center(
                        child: Text(
                            "No posts yet",
                            style: TextStyle(color: CustomColors.firebaseioGrey),
                        ),
                    );
                }
                return ListView.separated(
                    separatorBuilder: (context, index) => const SizedBox(height: 16.0),
                    itemCount: snapshot.data!.docs.length,
                    itemBuilder: (context, index) {
                        var noteInfo = snapshot.data!.docs[index].data() as dynamic;
                        String docID = snapshot.data!.docs[index].id;
                        String title = noteInfo["title"];
                        String comment = noteInfo['comment'];
                        String user = noteInfo['user'];
                        String time = noteInfo['time'];

                        return Padding(
                            padding: const EdgeInsets.all(8.0),
                            child: Ink(
                                decoration: BoxDecoration(
                                    color: Color(0xFF9CF29A),
                                    borderRadius: BorderRadius.circular(8.0),
                                ),
                                child: ListTile(
                                    shape: RoundedRectangleBorder(
                                        borderRadius: BorderRadius.circular(8.0),
                                    ),
                                ),
                            ),
                        );
                    },
                );
            }
        },
    );
}
```



```
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
    ),
);
},
);
}
}
```

## PrivateItemList.dart

```
// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/edit_screen.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class PrivateItemList extends StatefulWidget {
  const PrivateItemList({required User user}) : _user = user;
  final User _user;
  @override
  State<PrivateItemList> createState() => _PrivateItemListState();
}

class _PrivateItemListState extends State<PrivateItemList> {
  late User _user;

  @override
  void initState() {
    _user = widget._user;
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return StreamBuilder<QuerySnapshot>(
      stream: Database.readPrivateItems(_user.displayName),
      builder: (context, snapshot) {
        if (snapshot.hasError) {
```

```
        return const Text('Something went wrong');
    } else if (snapshot.hasData || snapshot.data != null) {
        if (snapshot.data!.docs.isEmpty) {
            return Center(
                child: Text(
                    "No posts yet",
                    style: TextStyle(color: CustomColors.firebaseioGrey),
                ),
            );
        }
        return ListView.separated(
            separatorBuilder: (context, index) => const SizedBox(height: 16.0),
            itemCount: snapshot.data!.docs.length,
            itemBuilder: (context, index) {
                var noteInfo = snapshot.data!.docs[index].data() as dynamic;
                String docID = snapshot.data!.docs[index].id;
                String title = noteInfo["title"];
                String description = noteInfo['description'];
                String user = noteInfo['user'];
                String time = noteInfo['time'];
                int likes = noteInfo['likes'];

                return Padding(
                    padding: const EdgeInsets.all(8.0),
                    child: Ink(
                        decoration: BoxDecoration(
                            color: Color(0xFF9CF29A),
                            borderRadius: BorderRadius.circular(8.0),
                        ),
                        child: ListTile(
                            shape: RoundedRectangleBorder(
                                borderRadius: BorderRadius.circular(8.0),
                            ),
                            onTap: () => Navigator.of(context).push(
                                MaterialPageRoute(
                                    builder: (context) => EditScreen(
                                        currentTitle: title,
                                        currentDescription: description,
                                        documentId: docID,
                                        user: user,
                                    ),
                                ),
                            ),
                        ),
                    title: Text(
                        title,
```

```
        maxLines: 1,
        overflow: TextOverflow.ellipsis,
    ),
    subtitle: Column(
        children: [
            Text(
                description,
                maxLines: 2,
                overflow: TextOverflow.ellipsis,
            ),
            const Text(""),
            Text(
                user +
                    " : " +
                    time.substring(0, 20) +
                    " ❤️ " +
                    likes.toString(),
                maxLines: 1,
                overflow: TextOverflow.ellipsis,
                style: const TextStyle(fontSize: 10),
            ),
        ],
    ),
    trailing: const Padding(
        padding: EdgeInsets.only(top: 5),
        child: Icon(Icons.edit),
    ),
),
),
),
);
},
),
),
);
},
);
}
}

return Center(
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
        ),
        );
);
},
);
},
);
});
```

## SearchItemList.dart

```
// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/display_all.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class SearchItem extends StatefulWidget {
    const SearchItem({required this.input, required User user}) : _user = user;
    final String input;
    final User _user;

    @override
    State<SearchItem> createState() => _SearchItemState();
}

class _SearchItemState extends State<SearchItem> {
    late User _user;

    @override
    void initState() {
        _user = widget._user;
        super.initState();
    }

    @override
    Widget build(BuildContext context) {
        return StreamBuilder<QuerySnapshot>(
            stream: Database.searchItem(widget.input.toString()),
            builder: (context, snapshot) {
                if (snapshot.hasError) {
                    return const Text('Something went wrong');
                } else if (snapshot.hasData || snapshot.data != null) {
                    if (snapshot.data!.docs.isEmpty) {
                        return Padding(
                            padding: const EdgeInsets.only(top: 300),
                            child: Column(
                                children: [

```

```

        Center(
            child: Text(
                "No results found",
                style: TextStyle(color: CustomColors.firebaseioGrey),
            ),
        ),
        Center(
            child: Text(
                "For best results please read below instructions",
                style: TextStyle(color: CustomColors.firebaseioYellow),
            ),
        ),
        const Padding(
            padding: EdgeInsets.only(top: 20),
            child: Text(
                "1. Use your name as it is. (Case Sensitive).",
                style: TextStyle(color: Colors.lightGreen),
            ),
        ),
        const Padding(
            padding: EdgeInsets.only(left: 5),
            child: Text(
                "2. Avoid first & final whitespace.",
                style: TextStyle(color: Colors.lightGreen),
            ),
        ),
        const Padding(
            padding: EdgeInsets.only(top: 20),
            child: Text(
                "1. නම එලෙසම හාවතා කරන්න.",
                style: TextStyle(color: Colors.lightGreen),
            ),
        ),
        const Padding(
            padding: EdgeInsets.only(left: 5),
            child: Text(
                "2. මූලට සහ අගට හිස්තුන් තැබීමෙන් වළකින්න.",
                style: TextStyle(color: Colors.lightGreen),
            ),
        ),
    ],
),
);
}
return ListView.separated(

```

```
separatorBuilder: (context, index) => const SizedBox(height: 16.0),
itemCount: snapshot.data!.docs.length,
itemBuilder: (context, index) {
  var noteInfo = snapshot.data!.docs[index].data() as dynamic;
  String docID = snapshot.data!.docs[index].id;
  String title = noteInfo["title"];
  String description = noteInfo['description'];
  String name = noteInfo['user'];
  String time = noteInfo['time'];
  int likes = noteInfo['likes'];

  return Padding(
    padding: const EdgeInsets.all(8.0),
    child: Ink(
      decoration: BoxDecoration(
        color: Color(0xFF9CF29A),
        borderRadius: BorderRadius.circular(8.0),
      ),
      child: ListTile(
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(8.0),
        ),
        onTap: () => Navigator.of(context).push(
          MaterialPageRoute(
            builder: (context) => DisplayAllScreen(
              currentTitle: title,
              currentDescription: description,
              documentId: docID,
              name: name,
              likes: likes,
              time: time,
              user: _user,
            ),
          ),
        ),
        title: Text(
          title,
          maxLines: 1,
          overflow: TextOverflow.ellipsis,
        ),
        subtitle: Column(
          children: [
            Text(
              description,
              maxLines: 2,
```

```
        overflow: TextOverflow.ellipsis,
    ),
    const Text(""),
    Text(
        name +
            " : " +
            time.substring(0, 20) +
            " ❤️ " +
            likes.toString(),
        maxLines: 1,
        overflow: TextOverflow.ellipsis,
        style: const TextStyle(fontSize: 10),
    ),
),
],
),
),
trailing: const Padding(
    padding: EdgeInsets.only(top: 5),
    child: Icon(Icons.arrow_forward),
),
),
),
),
),
);
},
);
}
}

return Center(
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
        ),
),
);
},
);
}
}
```

# CRUD OPERATIONS

```
static Future<void> addItem({  
    required String title,  
    required String description,
```

```
    required String user,
}) async {
  DocumentReference documentReferencer = _firestore.collection('items').doc();

  Map<String, dynamic> data = <String, dynamic>{
    "title": title,
    "description": description,
    "user": user,
    "time": time,
    "likes": 0
  };

  await documentReferencer
    .set(data)
    .whenComplete(() => print("Note item added to the database"))
    .catchError((e) => print(e));
}
```

```
static Future<void> updateItem({
  required String title,
  required String description,
  required String docId,
}) async {
  DocumentReference documentReferencer =
    _firestore.collection('items').doc(docId);

  Map<String, dynamic> data = <String, dynamic>{
    "title": title,
    "description": description,
  };

  await documentReferencer
    .update(data)
    .whenComplete(() => print("Note item updated in the database"))
    .catchError((e) => print(e));
}
```

```
static Future<void> updateLikes({
  required int likes,
  required String docId,
}) async {
  int increment = 1 + likes;
```

```
DocumentReference documentReferencer =
    _firestore.collection('items').doc(docId);

Map<String, dynamic> data = <String, dynamic>{
    "likes": increment,
};

await documentReferencer
    .update(data)
    .whenComplete(() => print("Note item updated in the database"))
    .catchError((e) => print(e));
}
```

```
static Future<void> decrementLikes({
    required int likes,
    required String docId,
}) async {
    int increment = likes - 1;

    DocumentReference documentReferencer =
        _firestore.collection('items').doc(docId);

    Map<String, dynamic> data = <String, dynamic>{
        "likes": increment,
    };

    await documentReferencer
        .update(data)
        .whenComplete(() => print("Note item updated in the database"))
        .catchError((e) => print(e));
}
```

```
static Stream<QuerySnapshot> readItems() {
    Query<Map<String, dynamic>> notesItemCollection =
        _firestore.collection('items').orderBy("time", descending: true);
    print('fetched');

    return notesItemCollection.snapshots();
}
```

```
static Future<void> addComment(
```

```
{required String comment,
required String user,
required String title}) async {
DocumentReference documentReferencer =
_firestore.collection('comments').doc();

Map<String, dynamic> data = <String, dynamic>{
  "user": user,
  "comment": comment,
  "time": time,
  "title": title
};

await documentReferencer
.set(data)
.whenComplete(() => print("Note item added to the database"))
.catchError((e) => print(e));
}
```

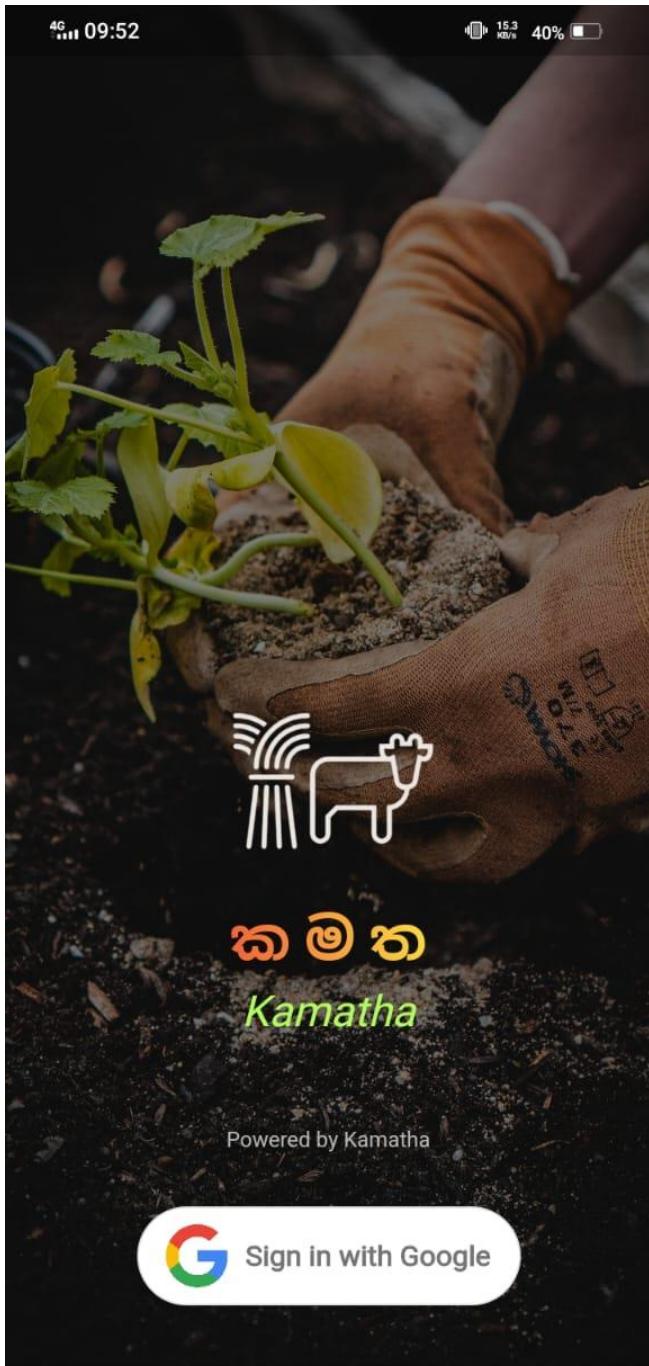
```
static Future<void> deleteComment({
  required String docId,
}) async {
  DocumentReference documentReferencer =
_firestore.collection('comments').doc(docId);

  await documentReferencer
.delete()
.whenComplete(() => print('Note comment deleted from the database'))
.catchError((e) => print(e));
}
```

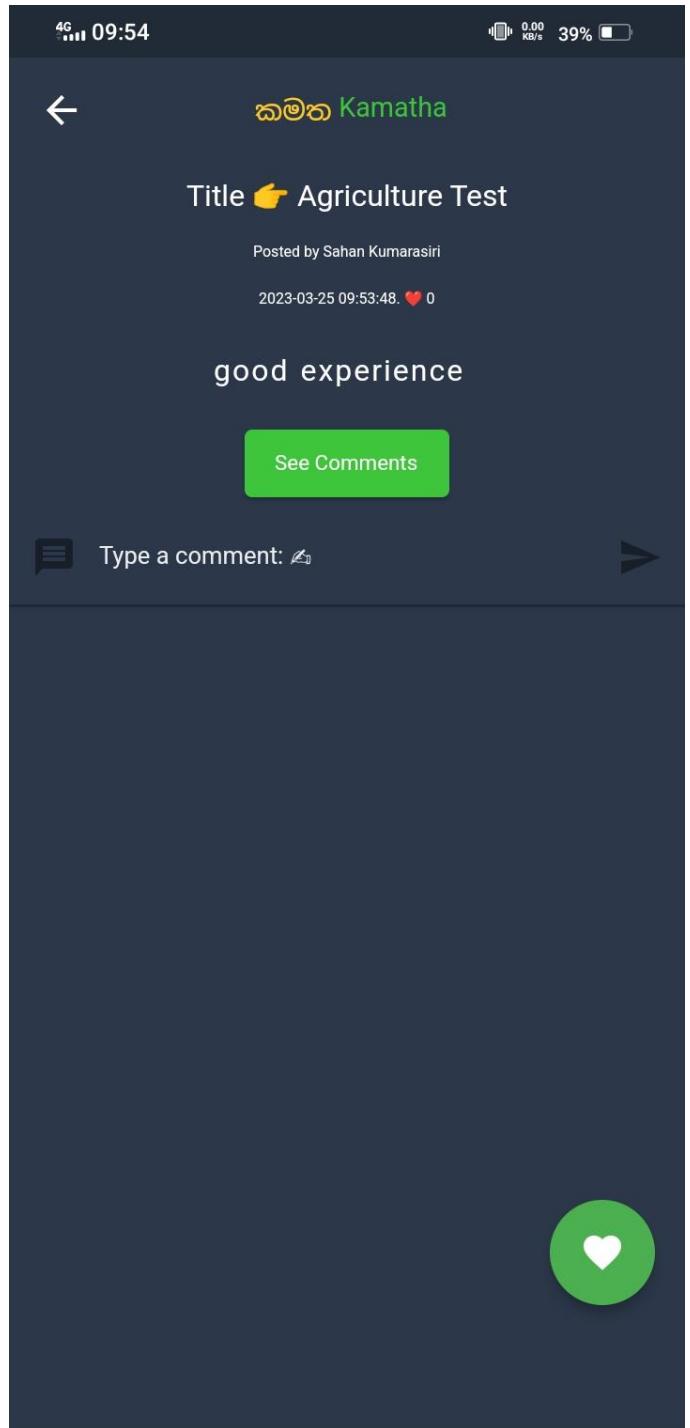
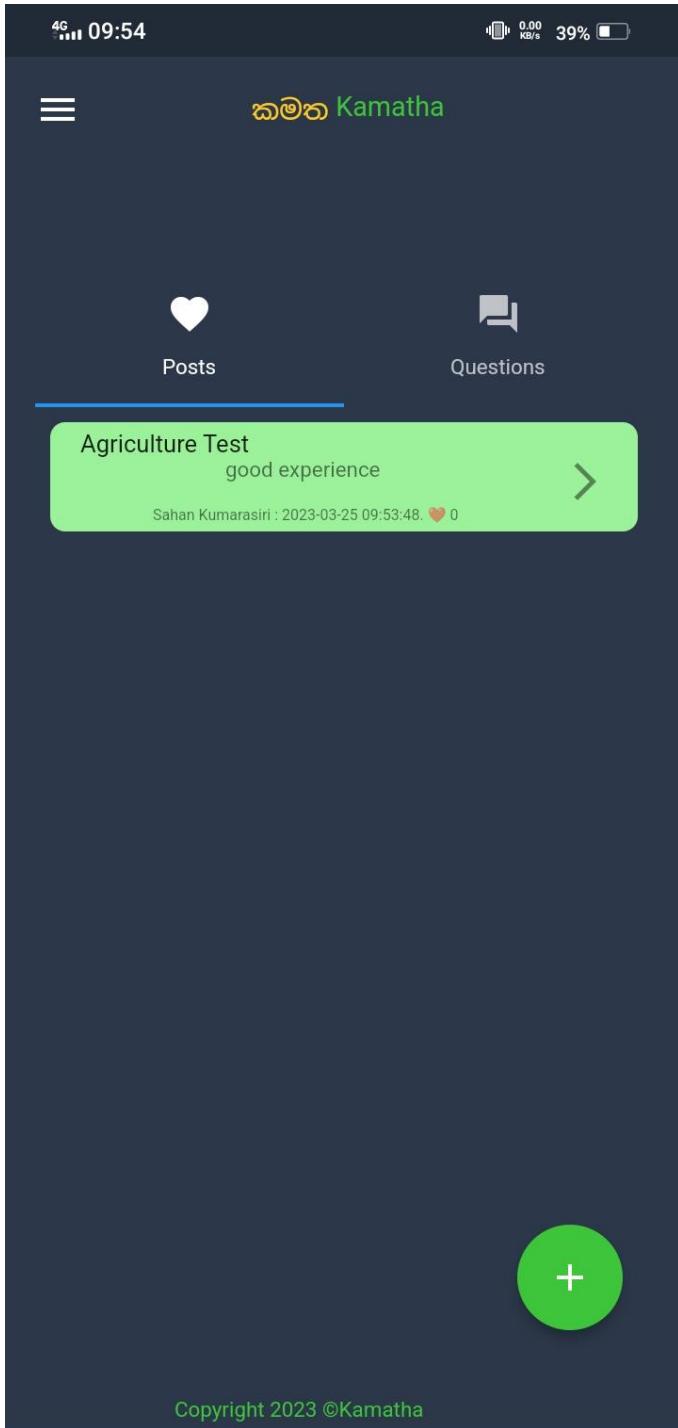
```
static Future<void> deleteItem({
  required String docId,
}) async {
  DocumentReference documentReferencer =
_firestore.collection('items').doc(docId);

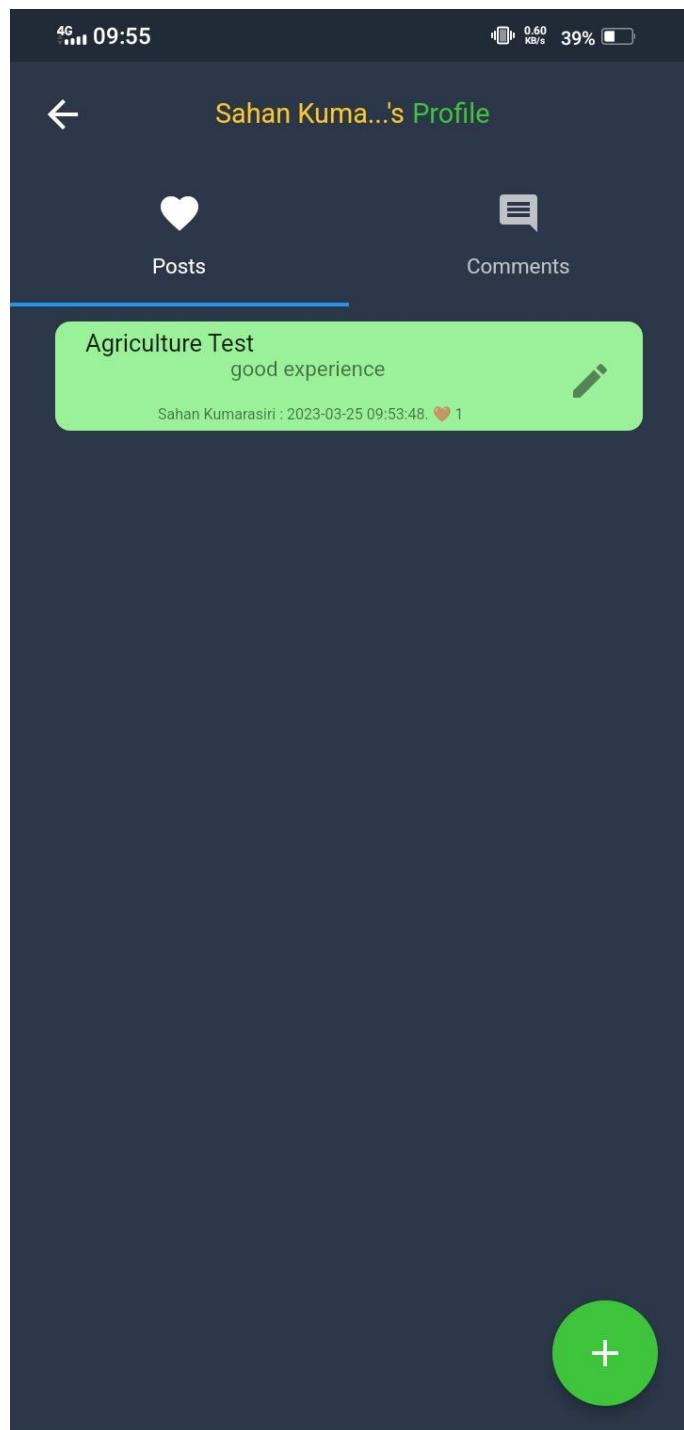
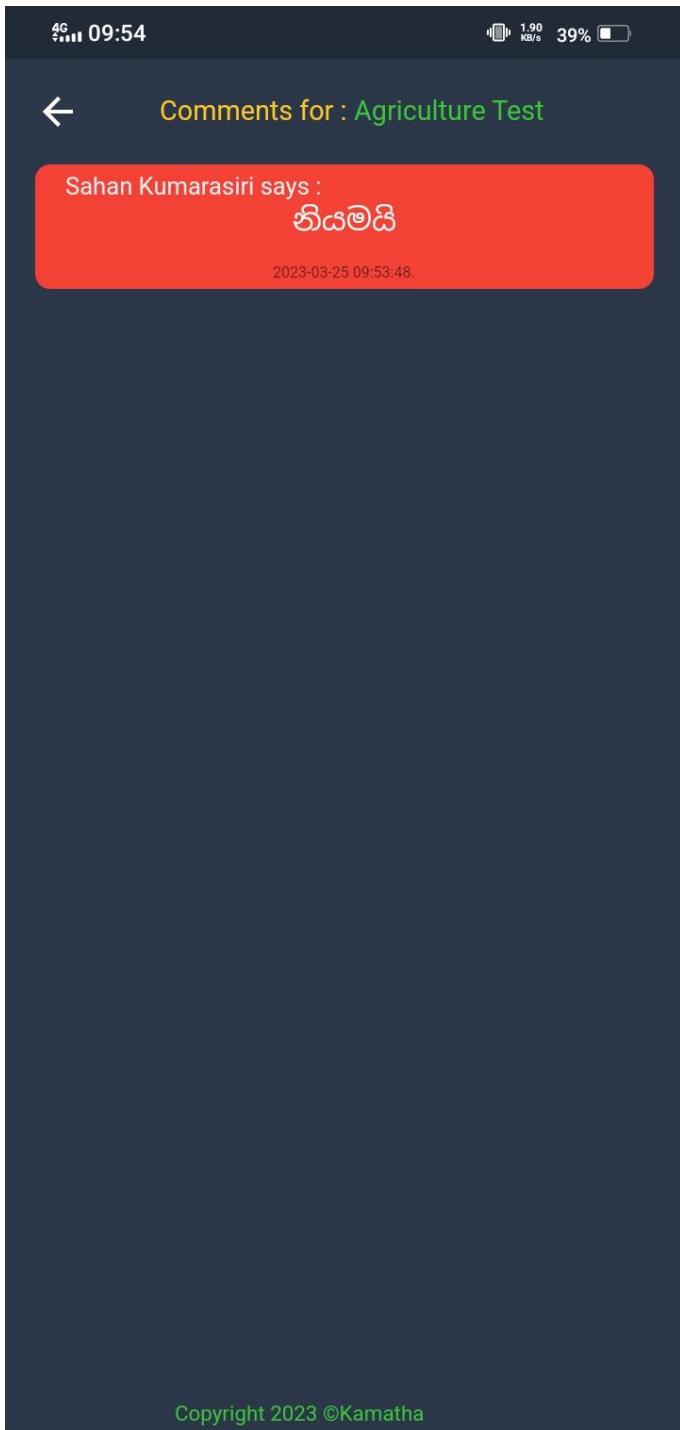
  await documentReferencer
.delete()
.whenComplete(() => print('Note item deleted from the database'))
.catchError((e) => print(e));
}
```

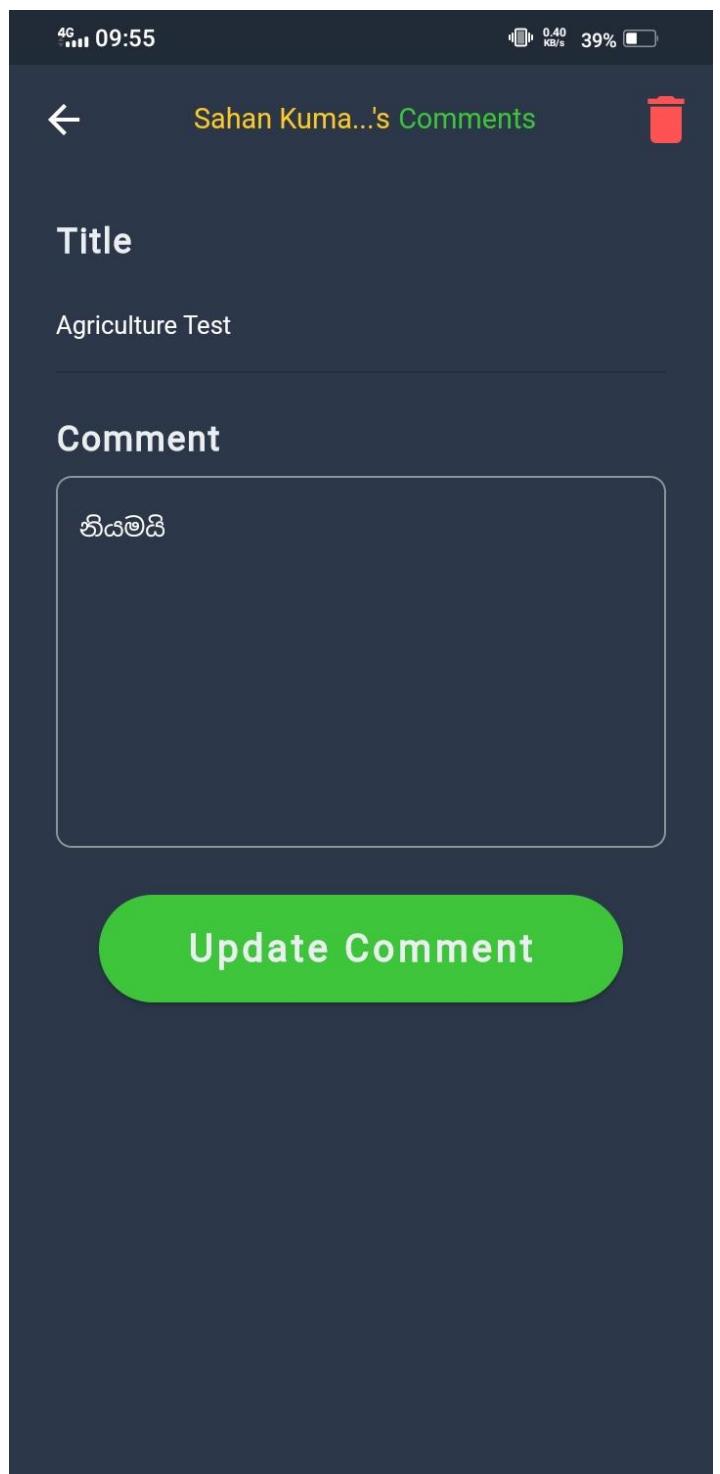
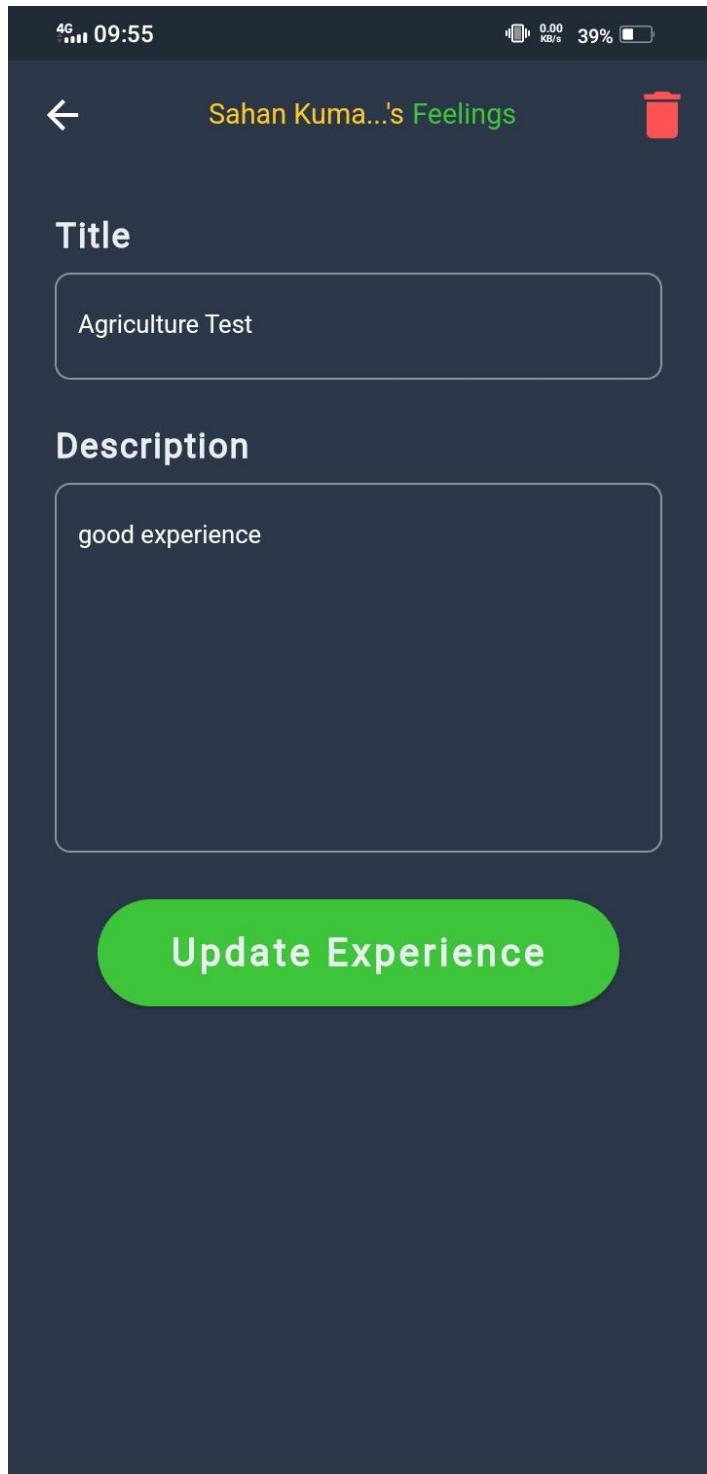
## UI Screenshots



This screenshot shows the "Post Your Experience" screen. At the top, there is a back arrow icon and the text "കമാത Kamatha". Below that, there is a section labeled "Title" with a placeholder "Enter your title". Underneath it is a section labeled "Experience / Research" with a placeholder "Enter your feeling". At the bottom right, there is a large green button with the text "Post Your Experience".







## Screenshots of Contributions



## References

<https://firebase.google.com/docs>

<https://flutter.dev/learn>

<https://dart.dev/guides>

## Post News, Articles and Schemas

In this component, admin can log in to the system and post their news, articles, and schemas. Users can see the news and give their reviews. There is a tab called See Posted News in the Drawer view. Admin can edit / delete their news.

### CODE SNIPPETS

### Screens

#### AddNews.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/widgets/add_item_form.dart';
import 'package:Kamataha/src/widgets/add_news_form.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class AddNews extends StatefulWidget {
    const AddNews({Key? key});

    @override
    State<AddNews> createState() => _AddNewsState();
}

class _AddNewsState extends State<AddNews> {
    final FocusNode _titleFocusNode = FocusNode();
    final FocusNode _descriptionFocusNode = FocusNode();

    late User _user;

    @override
    Widget build(BuildContext context) {
        return GestureDetector(
            onTap: () {
                _titleFocusNode.unfocus();
                _descriptionFocusNode.unfocus();
            }
        );
    }
}
```

```

        },
        child: Scaffold(
            backgroundColor: CustomColors.firebaseioNavy,
            appBar: AppBar(
                elevation: 0,
                backgroundColor: CustomColors.firebaseioNavy,
                title: AppBarTitle(),
                centerTitle: true,
            ),
            body: SafeArea(
                child: ListView(children: [
                    Padding(
                        padding: const EdgeInsets.only(
                            left: 16.0,
                            right: 16.0,
                            bottom: 20.0,
                        ),
                        child: AddNewsForm(),
                    ),
                ]),
            ),
        ),
    );
}
}

```

## AddReview.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/widgets/add_item_form.dart';
import 'package:Kamataha/src/widgets/add_review_form.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class AddReview extends StatefulWidget {
    const AddReview({Key? key, required this.currentDescription})
        : super(key: key);

    final String currentDescription;
    @override
    State<AddReview> createState() => _AddReviewState();
}

```

```
class _AddReviewState extends State<AddReview> {
    final FocusNode _titleFocusNode = FocusNode();
    final FocusNode _descriptionFocusNode = FocusNode();

    @override
    Widget build(BuildContext context) {
        return GestureDetector(
            onTap: () {
                _titleFocusNode.unfocus();
                _descriptionFocusNode.unfocus();
            },
            child: Scaffold(
                backgroundColor: CustomColors.firebaseioNavy,
                appBar: AppBar(
                    elevation: 0,
                    backgroundColor: CustomColors.firebaseioNavy,
                    title: AppBarTitle(),
                    centerTitle: true,
                ),
                body: SafeArea(
                    child: ListView(children: [
                        Padding(
                            padding: const EdgeInsets.only(
                                left: 16.0,
                                right: 16.0,
                                bottom: 20.0,
                            ),
                            child: AddReviewForm(
                                currentDescription: widget.currentDescription,
                            ),
                        ),
                    ]),
                ),
            );
    }
}
```

## AdminDashboard.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/all_news_screen.dart';
```

```
import 'package:Kamataha/src/screens/all_review.dart';
import 'package:Kamataha/src/screens/permission_screen.dart';
import 'package:Kamataha/src/screens/posts.dart';
import 'package:Kamataha/src/screens/questions.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:Kamataha/src/widgets/item_list.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'add_screen.dart';

class AdminDashboard extends StatefulWidget {
  const AdminDashboard({Key? key, required User user})
      : _user = user,
        super(key: key);

  final User _user;

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

class _AdminDashboardState extends State<AdminDashboard> {
  late User _user;

  @override
  void initState() {
    _user = widget._user;
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return DefaultTabController(
      length: 2,
      child: Scaffold(
        backgroundColor: CustomColors.firebaseioNavy,
        appBar: AppBar(
          title: AppBarTitle(),
          backgroundColor: CustomColors.firebaseioNavy,
          elevation: 0,
          centerTitle: true,
        ),
        // floatingActionButton: FloatingActionButton(
        //   child: const Icon(Icons.add),
        //   backgroundColor: CustomColors.firebaseioOrange,
        
```

```
//    onPressed: () {
//        Navigator.of(context).push(
//            MaterialPageRoute(
//                builder: (context) => AddScreen(user: _user),
//            ),
//        );
//    },
// ),
body: ListView(
    children: [
        Padding(
            padding: EdgeInsets.all(15),
            child: Card(
                color: Colors.white,
                child: ListTile(
                    title: Column(
                        children: [
                            Padding(
                                padding: EdgeInsets.all(0.8),
                                child: Image.asset(
                                    "assets/news.jpeg",
                                    width: 100,
                                    height: 100,
                                ),
                            ),
                            const Text(
                                'News',
                                style: TextStyle(
                                    fontSize: 24.0,
                                    color: Colors.black,
                                ),
                            ),
                            Padding(
                                padding: EdgeInsets.fromLTRB(150, 0, 0, 0),
                                child: Text("view more"),
                            )
                        ],
                    ),
                    onTap: () {
                        Navigator.of(context).push(MaterialPageRoute(
                            builder: (context) => AllNewsScreen(user: _user),
                        ));
                    },
                ),
            ),
        ),
    ],
),
```

```
Padding(
  padding: EdgeInsets.all(15),
  child: Card(
    color: Colors.white,
    child: ListTile(
      title: Column(
        children: [
          Padding(
            padding: EdgeInsets.all(0.8),
            child: Image.asset(
              "assets/schemas.jpeg",
              width: 100,
              height: 100,
            ),
          ),
          const Text(
            'Schemas',
            style: TextStyle(
              fontSize: 24.0,
              color: Colors.black,
            ),
          ),
        ],
      ),
      onTap: () {
        Navigator.of(context).push(MaterialPageRoute(
          builder: (context) => AdminDashboard(user: _user),
        ));
      },
    )));
Padding(
  padding: EdgeInsets.all(15),
  child: Card(
    color: Colors.white,
    child: ListTile(
      title: Column(
        children: [
          Padding(
            padding: EdgeInsets.all(0.8),
            child: Image.asset(
              "assets/articles.jpeg",
              width: 100,
              height: 100,
            ),
          ),
        ],
      ),
    )));

```

```
        ),
        const Text(
            'Articles',
            style: TextStyle(
                fontSize: 24.0,
                color: Colors.black,
            ),
            ),
            ],
        ),
        onTap: () {
            Navigator.of(context).push(MaterialPageRoute(
                builder: (context) => AdminDashboard(user: _user),
            )));
        },
    ),
),
Padding(
    padding: EdgeInsets.all(15),
    child: Card(
        color: Colors.white,
        child: ListTile(
            title: Column(
                children: [
                    Padding(
                        padding: EdgeInsets.all(0.8),
                        child: Image.asset(
                            "assets/comment.jpeg",
                            width: 100,
                            height: 100,
                        ),
                    ),
                ],
            ),
            const Text(
                'Comments',
                style: TextStyle(
                    fontSize: 24.0,
                    color: Colors.black,
                ),
                ),
            ],
        ),
        onTap: () {
            Navigator.of(context).push(MaterialPageRoute(
                builder: (context) => Reviews(user: _user),
            )));
        },
    ),
);
```

```

        },
        ),
        )),
    ],
),
floatingActionButton: FloatingActionButton(
    child: const Icon(Icons.add),
    backgroundColor: CustomColors.firebaseioOrange,
    onPressed: () {
        Navigator.of(context).push(
            MaterialPageRoute(
                builder: (context) => Permission(),
            ),
        );
    },
),
),
);
}
}

```

## AddNewsScreen.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/add_screen.dart';
import 'package:Kamataha/src/screens/permission_screen.dart';
import 'package:Kamataha/src/widgets/all_add_list.dart';
import 'package:Kamataha/src/widgets/all_news_display_list.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:Kamataha/src/widgets/item_list.dart';
import 'package:Kamataha/src/widgets/my_add.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class AllNewsScreen extends StatefulWidget {
    const AllNewsScreen({Key? key, required User user})
        : _user = user,
        super(key: key);

    final User _user;

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

```

```

}

class _AllNewsScreenState extends State<AllNewsScreen> {
    late User _user;

    @override
    void initState() {
        _user = widget._user;
        super.initState();
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: AppBarTitle(),
                backgroundColor: CustomColors.firebaseioNavy,
            ),
            backgroundColor: CustomColors.firebaseioNavy,
            body: AllNewsDisplayList(user: _user),
            floatingActionButton: FloatingActionButton(
                child: const Icon(Icons.add),
                backgroundColor: CustomColors.firebaseioOrange,
                onPressed: () {
                    Navigator.of(context).push(
                        MaterialPageRoute(
                            builder: (context) => Permission(),
                        ),
                    );
                },
            ),
        );
    }
}

```

## AllReview.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/add_screen.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:Kamataha/src/widgets/item_list.dart';
import 'package:Kamataha/src/widgets/review_list.dart';
import 'package:firebase_auth/firebase_auth.dart';

```

```

import 'package:flutter/material.dart';

class Reviews extends StatefulWidget {
  const Reviews({Key? key, required User user})
      : _user = user,
        super(key: key);

  final User _user;

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

class _ReviewsState extends State<Reviews> {
  late User _user;

  @override
  void initState() {
    _user = widget._user;
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: AppBarTitle(),
        backgroundColor: CustomColors.firebaseioNavy,
      ),
      backgroundColor: CustomColors.firebaseioNavy,
      body: ReviewList(user: _user),
    );
  }
}

```

## EditNewsScreen.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/widgets/edit_item_form.dart';
import 'package:Kamataha/src/widgets/edit_news_form.dart';
import 'package:flutter/material.dart';

```

```
class EditNewsScreen extends StatefulWidget {
    final String currentTitle;
    final String currentDescription;
    final String documentId;

    // ignore: use_key_in_widget_constructors
    const EditNewsScreen({
        required this.currentTitle,
        required this.currentDescription,
        required this.documentId,
    });

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

class _EditNewsScreenState extends State<EditNewsScreen> {
    final FocusNode _titleFocusNode = FocusNode();

    final FocusNode _descriptionFocusNode = FocusNode();

    bool _isDeleting = false;

    @override
    Widget build(BuildContext context) {
        return GestureDetector(
            onTap: () {
                _titleFocusNode.unfocus();
                _descriptionFocusNode.unfocus();
            },
            child: Scaffold(
                backgroundColor: CustomColors.firebaseioNavy,
                appBar: AppBar(
                    elevation: 0,
                    backgroundColor: CustomColors.firebaseioNavy,
                    title: Row(
                        mainAxisAlignment: MainAxisAlignment.start,
                        mainAxisSize: MainAxisSize.min,
                        children: [
                            const SizedBox(width: 8),
                            Text(
                                ' Edit News ',
                                style: TextStyle(
                                    color: CustomColors.firebaseioOrange,
                                    fontSize: 18,
                                ),
                            ),
                        ],
                    ),
                ),
            ),
        );
    }
}
```

```
        ),
        ],
),
),
centerTitle: true,
actions: [
_isDeleting
? const Padding(
padding: EdgeInsets.only(
top: 10.0,
bottom: 10.0,
right: 16.0,
),
child: CircularProgressIndicator(
valueColor: AlwaysStoppedAnimation<Color>(
Colors.redAccent,
),
strokeWidth: 3,
),
),
),
: IconButton(
icon: const Icon(
Icons.delete,
color: Colors.redAccent,
size: 32,
),
onPressed: () async {
setState(() {
_isDeleting = true;
});;

await Database.deleteItem(
docId: widget.documentId,
);

setState(() {
_isDeleting = false;
});;

Navigator.of(context).pop();
},
),
],
),
body: SafeArea(
```

```

        child: Padding(
            padding: const EdgeInsets.only(
                left: 16.0,
                right: 16.0,
                bottom: 20.0,
            ),
            child: EditNewsForm(
                documentId: widget.documentId,
                titleFocusNode: _titleFocusNode,
                descriptionFocusNode: _descriptionFocusNode,
                currentTitle: widget.currentTitle,
                currentDescription: widget.currentDescription,
            ),
        ),
    ),
),
),
),
),
),
);
}
}
}

```

## PermissionScreen.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/add_news.dart';
import 'package:Kamataha/src/utils/adminAuth.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:Kamataha/src/widgets/add_item_form.dart';
import 'package:Kamataha/src/widgets/add_news_form.dart';
import 'package:Kamataha/src/widgets/custom_form_field.dart';
import 'package:Kamataha/src/widgets/list_comments.dart';
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';

class Permission extends StatefulWidget {
    const Permission({Key? key}) : super(key: key);

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

class _PermissionState extends State<Permission> {
    final TextEditingController _managerBudgetTextEditingController =
        TextEditingController();

```

```
final FocusNode fNode = FocusNode();
bool _isDeleting = false;
@Override
void initState() {
    super.initState();
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: CustomColors.firebaseioNavy,
        appBar: AppBar(
            elevation: 0,
            backgroundColor: CustomColors.firebaseioNavy,
            title: Row(
                mainAxisAlignment: MainAxisAlignment.start,
                mainAxisSize: MainAxisSize.min,
                children: [
                    const SizedBox(width: 8),
                    Text(
                        'Admin Dashboard : ',
                        style: TextStyle(
                            color: CustomColors.firebaseioYellow,
                            fontSize: 18,
                        ),
                    ),
                    ],
                ),
            centerTitle: true,
            // leading: IconButton(
            //     icon: const Icon(Icons.menu),
            //     onPressed: () => {_buildDrawer(context)},
            // ),
        ),
        body: SafeArea(
            child: Column(
            children: [
                Center(
                    child: Text(
                        "Please Enter your Admin Credentials for access to the admin dashboard",
                        style: TextStyle(
                            color: Colors.white,
                            fontSize: 18,
                        ),
                    ),
                ),
            ],
        ),
    );
}
```

```
        ),
    )),
    Center(
        child: CustomFormField(
            isLabelEnabled: false,
            controller: _managerBudgetTextEditingController,
            focusNode: fNode,
            keyboardType: TextInputType.text,
            inputAction: TextInputAction.next,
            validator: (value) => Validator.validateField(
                value: value,
            ),
            label: 'Company Budget',
            hint: 'Enter your access key',
        ),
    ),
    Center(
        child: ElevatedButton(
            child: Text("Submit"),
            onPressed: () async {
                if (Auth.getPermission(
                    _managerBudgetTextEditingController.text)) {
                    Navigator.of(context)
                        .push(MaterialPageRoute(builder: (context) => AddNews()));
                } else
                    Fluttertoast.showToast(
                        msg: "Access Key Incorrect",
                        toastLength: Toast.LENGTH_SHORT,
                        gravity: ToastGravity.BOTTOM,
                        timeInSecForIosWeb: 1,
                        backgroundColor: Colors.red,
                        textColor: Colors.yellow));
            },
        ),
    ),
),
],
)),
);
});
```

## UserinfoScreen.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/add_screen.dart';
import 'package:Kamataha/src/screens/admin_dashboard.dart';
import 'package:Kamataha/src/screens/credits_screen.dart';
import 'package:Kamataha/src/screens/dashboard.dart';
import 'package:Kamataha/src/screens/private_question_answer.dart';
import 'package:Kamataha/src/screens/private_screen.dart';
import 'package:Kamataha/src/screens/seller_dashboard.dart';
import 'package:Kamataha/src/screens/sign_in_screen.dart';
import 'package:Kamataha/src/utils/auth.dart';
import 'package:Kamataha/src/widgets/app_bar_title.dart';
import 'package:Kamataha/src/widgets/item_list.dart';
import 'package:Kamataha/src/widgets/search_input_widget.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class UserInfoScreen extends StatefulWidget {
  const UserInfoScreen({Key? key, required User user})
      : _user = user,
        super(key: key);

  final User _user;

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

class _UserInfoScreenState extends State<UserInfoScreen> {
  late User _user;
  bool _isSigningOut = false;

  final String time = DateTime.now().toString().substring(0, 4);

  final FocusNode _nameFocusNode = FocusNode();
  final FocusNode _emailFocusNode = FocusNode();
  final FocusNode _passwordFocusNode = FocusNode();

  Drawer _buildDrawer(BuildContext context) {
    return Drawer(
      child: Container(
        color: CustomColors.firebaseioNavy,
        child: ListView(
          padding: EdgeInsets.zero,
          children: [
            Row(),
```



```
        child: ListTile(
            leading: const Icon(
                Icons.account_box_outlined,
                color: Colors.white,
            ),
            trailing: const Icon(Icons.arrow_forward_ios_outlined),
            title: const Text(
                'Account',
                style: TextStyle(
                    fontSize: 24.0,
                    color: Colors.white,
                ),
            ),
            onTap: () {
                Navigator.of(context).push(MaterialPageRoute(
                    builder: (context) => PrivateScreen(user: _user),
                ));
            },
            subtitle: Text(
                "See your posts | Manage your preferences",
                style: TextStyle(color: CustomColors.firebaseioGrey),
            ),
        ),
    ),
    const SizedBox(height: 8.0),
Card(
    color: CustomColors.firebaseioOrange,
    child: ListTile(
        leading: const Icon(
            Icons.question_answer,
            color: Colors.white,
        ),
        trailing: const Icon(Icons.arrow_forward_ios_outlined),
        title: const Text(
            'Questionnaire',
            style: TextStyle(
                fontSize: 24.0,
                color: Colors.white,
            ),
        ),
        onTap: () {
            Navigator.of(context).push(MaterialPageRoute(
                builder: (context) => PrivateQuestionScreen(user: _user),
            ));
        },
    ),
}
```

```
        subtitle: Text(
            "See your Questions",
            style: TextStyle(color: CustomColors.firebaseioGrey),
        ),
    ),
),
),
const SizedBox(height: 8.0),
Card(
    color: CustomColors.firebaseioOrange,
    child: ListTile(
        leading: const Icon(
            Icons.shopping_cart,
            color: Colors.white,
        ),
        trailing: const Icon(Icons.arrow_forward_ios_outlined),
        title: const Text(
            'Items',
            style: TextStyle(
                fontSize: 24.0,
                color: Colors.white,
            ),
        ),
        onTap: () {
            Navigator.of(context).push(MaterialPageRoute(
                builder: (context) => SellerDashboard(user: _user),
            ));
        },
        subtitle: Text(
            "Sell your items | Buy your needs",
            style: TextStyle(color: CustomColors.firebaseioGrey),
        ),
    ),
),
const SizedBox(height: 8.0),
Card(
    color: CustomColors.firebaseioOrange,
    child: ListTile(
        leading: const Icon(
            Icons.schema_rounded,
            color: Colors.white,
        ),
        trailing: const Icon(Icons.arrow_forward_ios_outlined),
        title: const Text(
            'See Posted News',
            style: TextStyle(
```

```
        fontSize: 24.0,
        color: Colors.white,
    ),
),
onTap: () {
    Navigator.of(context).push(MaterialPageRoute(
        builder: (context) => AdminDashboard(user: _user),
    )));
},
subtitle: Text(
    "See your news ",
    style: TextStyle(color: CustomColors.firebaseioGrey),
),
),
),
// Card(
//     color: Colors.lightGreen,
//     child: ListTile(
//         leading: const Icon(
//             Icons.credit_score_outlined,
//             color: Colors.white,
//         ),
//         trailing: const Icon(Icons.arrow_forward_ios_outlined),
//         title: const Text(
//             'Credits',
//             style: TextStyle(
//                 fontSize: 24.0,
//                 color: Colors.white,
//             ),
//         ),
//         onTap: () {
//             Navigator.of(context).push(MaterialPageRoute(
//                 builder: (context) => const CreditsScreen()));
//         },
//     ),
//     subtitle: Text(
//         "About US | Technologies",
//         style: TextStyle(color: CustomColors.firebaseioGrey),
//     ),
// ),
// ),
Center(
    child: Padding(
        padding: const EdgeInsets.only(top: 50),
        child: Text(
            "v2.0",
        ),
    ),
)
```

```
        style: TextStyle(color: CustomColors.firebaseioGrey),
    ),
),
),
),
_isSigningOut
? const CircularProgressIndicator(
    valueColor: AlwaysStoppedAnimation<Color>(Colors.white),
)
: Padding(
    padding:
        const EdgeInsets.only(left: 70, right: 70, top: 180),
child: ElevatedButton(
    style: ButtonStyle(
        backgroundColor: MaterialStateProperty.all(
            Color(0xFF9CF29A),
        ),
        shape: MaterialStateProperty.all(
            RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(45),
            ),
        ),
    ),
    onPressed: () async {
        setState(() {
            _isSigningOut = true;
        });
        await Authentication.signOut(context: context);
        setState(() {
            _isSigningOut = false;
        });
        Navigator.of(context)
            .pushReplacement(_routeToSignInScreen());
    },
),
child: const Padding(
    padding: EdgeInsets.only(top: 5.0, bottom: 8.0),
    child: Text(
        'Sign Out',
        style: TextStyle(
            fontSize: 20,
            fontWeight: FontWeight.bold,
            color: Colors.white,
            letterSpacing: 2,
        ),
    ),
),
),
```

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

Route _routeToSignInScreen() {
return PageRouteBuilder(
pageBuilder: (context, animation, secondaryAnimation) => SignInScreen(),
transitionsBuilder: (context, animation, secondaryAnimation, child) {
var begin = const Offset(-1.0, 0.0);
var end = Offset.zero;
var curve = Curves.ease;

var tween =
Tween(begin: begin, end: end).chain(CurveTween(curve: curve));

return SlideTransition(
position: animation.drive(tween),
child: child,
);
},
);
}

@Override
void initState() {
_user = widget._user;

super.initState();
}

@Override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: CustomColors.firebaseioNavy,
appBar: AppBar(
elevation: 0,
backgroundColor: CustomColors.firebaseioNavy,
title: AppBarTitle(),
centerTitle: true,
// leading: IconButton(
// icon: const Icon(Icons.menu),
```

```

        // onPressed: () => {_buildDrawer(context)},
        // ),
    ),
    // floatingActionButton: FloatingActionButton(
    //   child: const Icon(Icons.add),
    //   backgroundColor: CustomColors.firebaseioOrange,
    //   onPressed: () {
    //     Navigator.of(context).push(
    //       MaterialPageRoute(
    //         builder: (context) => AddScreen(user: _user),
    //       ),
    //     );
    //   },
    // ),
    floatingActionButtonAnimator: FloatingActionButtonAnimator.scaling,
    drawer: _buildDrawer(context),
    body: SafeArea(
      child: Padding(
        padding: const EdgeInsets.only(
          left: 16.0,
          right: 16.0,
          bottom: 20.0,
        ),
        child: Dashboard(
          user: _user,
        ),
      ),
    ),
    bottomNavigationBar: Padding(
      padding: const EdgeInsets.only(left: 90, bottom: 5),
      child: Text(
        "Copyright " + time + " @" + "Kamatha",
        style: TextStyle(color: CustomColors.firebaseioOrange),
      ),
    ),
  );
}
}

```

## SplashScreen.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/sign_in_screen.dart';

```

```

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

class SplashScreenHome extends StatefulWidget {
  const SplashScreenHome({Key? key}) : super(key: key);

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

class _SplashScreenHomeState extends State<SplashScreenHome> {
  @override
  Widget build(BuildContext context) {
    return SplashScreen(
      seconds: 5,
      navigateAfterSeconds: SignInScreen(),
      title: const Text(
        'Welcome',
        style: TextStyle(
          fontWeight: FontWeight.bold, fontSize: 20.0, color: Colors.green),
      ),
      backgroundColor: CustomColors.firebaseioNavy,
      loaderColor: Colors.green,
    );
  }
}

```

## Widgets

### AddNewsForm.dart

```

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'custom_form_field.dart';

// ignore: must_be_immutable
class AddNewsForm extends StatefulWidget {
  const AddNewsForm({
    Key? key,
  });

```

```
@override
_AddNewsFormState createState() => _AddNewsFormState();
}

class _AddNewsFormState extends State<AddNewsForm> {
final _addItemFormKey = GlobalKey<FormState>();

late User _user;

bool _isProcessing = false;

final TextEditingController _titleController = TextEditingController();
final TextEditingController _descriptionController = TextEditingController();

@Override
Widget build(BuildContext context) {
    return Form(
        key: _addItemFormKey,
        child: Column(
            children: [
                Padding(
                    padding: const EdgeInsets.only(
                        left: 8.0,
                        right: 8.0,
                        bottom: 24.0,
                    ),
                    child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                            const SizedBox(height: 24.0),
                            Text(
                                'Title',
                                style: TextStyle(
                                    color: CustomColors.firebaseioGrey,
                                    fontSize: 22.0,
                                    letterSpacing: 1,
                                    fontWeight: FontWeight.bold,
                                ),
                            ),
                            const SizedBox(height: 8.0),
                            CustomFormField(
                                isLabelEnabled: false,
                                controller: _titleController,
                                focusNode: FocusNode(),
                            ),
                            const SizedBox(height: 24.0),
                            Text(
                                'Description',
                                style: TextStyle(
                                    color: CustomColors.firebaseioGrey,
                                    fontSize: 22.0,
                                    letterSpacing: 1,
                                    fontWeight: FontWeight.bold,
                                ),
                            ),
                            const SizedBox(height: 8.0),
                            CustomFormField(
                                isLabelEnabled: false,
                                controller: _descriptionController,
                                focusNode: FocusNode(),
                            ),
                        ],
                    ),
                ),
            ],
        ),
    );
}
```

```
        keyboardType: TextInputType.text,
        inputAction: TextInputAction.next,
        validator: (value) => Validator.validateField(
            value: value,
        ),
        label: 'Title',
        hint: 'Enter your title',
    ),
    const SizedBox(height: 24.0),
    Text(
        'Description',
        style: TextStyle(
            color: CustomColors.firebaseioGrey,
            fontSize: 22.0,
            letterSpacing: 1,
            fontWeight: FontWeight.bold,
        ),
    ),
    const SizedBox(height: 8.0),
    CustomFormField(
        maxLines: 10,
        isLabelEnabled: false,
        controller: _descriptionController,
        focusNode: FocusNode(),
        keyboardType: TextInputType.text,
        inputAction: TextInputAction.done,
        validator: (value) => Validator.validateField(
            value: value,
        ),
        label: 'Description',
        hint: 'Enter your Description',
    ),
],
),
),
),
_isProcessing
? Padding(
    padding: const EdgeInsets.all(16.0),
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
    ),
)
:
ElevatedButton(
```

```
        style: ButtonStyle(
            backgroundColor: MaterialStateProperty.all(
                CustomColors.firebaseioOrange,
            ),
            shape: MaterialStateProperty.all(
                RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(45),
                ),
            ),
        ),
        onPressed: () async {
            if (_addItemFormKey.currentState!.validate()) {
                setState(() {
                    _isProcessing = true;
                });

                await Database.addNews(
                    title: _titleController.text,
                    description: _descriptionController.text,
                );

                setState(() {
                    _isProcessing = false;
                });

                Navigator.of(context).pop();
            }
        },
    ],
);
});
```

```
}
```

## AddReviewForm.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'custom_form_field.dart';

// ignore: must_be_immutable
class AddReviewForm extends StatefulWidget {
    final String currentDescription;
    const AddReviewForm({Key? key, required this.currentDescription});

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

class _AddReviewFormState extends State<AddReviewForm> {
    final _addItemFormKey = GlobalKey<FormState>();

    late User _user;

    bool _isProcessing = false;

    final TextEditingController _titleController = TextEditingController();
    final TextEditingController _descriptionController = TextEditingController();

    @override
    Widget build(BuildContext context) {
        return Form(
            key: _addItemFormKey,
            child: Column(
                children: [
                    Padding(
                        padding: const EdgeInsets.only(
                            left: 8.0,
                            right: 8.0,
                            bottom: 24.0,
                        ),
                        child: Column(
```



```
        borderRadius: BorderRadius.circular(45),
    ),
),
),
),
),
onPressed: () async {
    if (_addItemFormKey.currentState!.validate()) {
        setState(() {
            _isProcessing = true;
        });
    }

    await Database.addReview(
        description: _descriptionController.text,
    );

    setState(() {
        _isProcessing = false;
    });

    Navigator.of(context).pop();
}
},
child: Padding(
    padding: const EdgeInsets.only(top: 16.0, bottom: 16.0),
    child: Text(
        'Post Your Review',
        style: TextStyle(
            fontSize: 24,
            fontWeight: FontWeight.bold,
            color: CustomColors.firebaseioGrey,
            letterSpacing: 2,
        ),
    ),
),
),
),
),
),
],
),
);
}
}
```

## AllNewsDisplayList.dart

```
// ignore_for_file: use_key_in_widget_constructors
```

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/display_all.dart';
import 'package:Kamataha/src/screens/display_news.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class AllNewsDisplayList extends StatefulWidget {
    const AllNewsDisplayList({required User user}) : _user = user;
    final User _user;

    @override
    State<AllNewsDisplayList> createState() => _AllNewsDisplayListState();
}

class _AllNewsDisplayListState extends State<AllNewsDisplayList> {
    late User _user;

    @override
    void initState() {
        _user = widget._user;
        super.initState();
    }

    @override
    Widget build(BuildContext context) {
        return StreamBuilder<QuerySnapshot>(
            stream: Database.readNews(),
            builder: (context, snapshot) {
                if (snapshot.hasError) {
                    return const Text('Something went wrong');
                } else if (snapshot.hasData || snapshot.data != null) {
                    if (snapshot.data!.docs.isEmpty) {
                        return Center(
                            child: Text(
                                "No news yet",
                                style: TextStyle(color: CustomColors.firebaseioGrey),
                            ),
                        );
                    }
                    return ListView.separated(
                        separatorBuilder: (context, index) => const SizedBox(height: 16.0),
                        itemCount: snapshot.data!.docs.length,
                    );
                }
            },
        );
    }
}
```

```
itemBuilder: (context, index) {
    var noteInfo = snapshot.data!.docs[index].data() as dynamic;
    String docID = snapshot.data!.docs[index].id;
    String title = noteInfo["title"];
    String description = noteInfo['description'];

    String time = noteInfo['time'];

    return Padding(
        padding: const EdgeInsets.all(8.0),
        child: Ink(
            decoration: BoxDecoration(
                color: Color(0xFF9CF29A),
                borderRadius: BorderRadius.circular(8.0),
            ),
            child: ListTile(
                shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(8.0),
                ),
                onTap: () => Navigator.of(context).push(
                    MaterialPageRoute(
                        builder: (context) => DisplayNews(
                            currentTitle: title,
                            currentDescription: description,
                            documentId: docID,
                            description: '',
                        ),
                    ),
                ),
                title: Text(
                    title,
                    maxLines: 1,
                    overflow: TextOverflow.ellipsis,
                ),
                subtitle: Column(
                    children: [
                        Text(
                            description,
                            maxLines: 2,
                            overflow: TextOverflow.ellipsis,
                        ),
                        const Text(""),
                        // Text(
                        //     user +
                        //     " : " +

```

```
//           time.substring(0, 20) +
//           " ❤️ " +
//           likes.toString(),
//           maxLines: 1,
//           overflow: TextOverflow.ellipsis,
//           style: const TextStyle(fontSize: 10),
//           ),
//         ],
//       ),
//       trailing: const Padding(
//         padding: EdgeInsets.only(top: 5),
//         child: Icon(Icons.arrow_forward_ios_sharp),
//       ),
//     ),
//   )));
},
);
}
}

return Center(
  child: CircularProgressIndicator(
    valueColor: AlwaysStoppedAnimation<Color>(
      CustomColors.firebaseioOrange,
    ),
  ),
);
},
);
}
});
```

## EditNewsForm.dart

```
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:flutter/material.dart';
import 'custom_form_field.dart';

class EditNewsForm extends StatefulWidget {
  final FocusNode titleFocusNode;
  final FocusNode descriptionFocusNode;
  final String currentTitle;
```

```
final String currentDescription;
final String documentId;

const EditNewsForm({
  required this.titleFocusNode,
  required this.descriptionFocusNode,
  required this.currentTitle,
  required this.currentDescription,
  required this.documentId,
});

@Override
_EditNewsFormState createState() => _EditNewsFormState();
}

class _EditNewsFormState extends State<EditNewsForm> {
  final _editNewsFormKey = GlobalKey<FormState>();

  bool _isProcessing = false;

  late TextEditingController _titleController;
  late TextEditingController _descriptionController;

  @override
  void initState() {
    _titleController = TextEditingController(
      text: widget.currentTitle,
    );

    _descriptionController = TextEditingController(
      text: widget.currentDescription,
    );
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return Form(
      key: _editNewsFormKey,
      child: ListView(
        children: [
          Padding(
            padding: const EdgeInsets.only(
              left: 8.0,
              right: 8.0,
            ),
          ),
        ],
      ),
    );
  }
}
```

```
        bottom: 24.0,
    ),
    child: Column(
        mainAxisAlignment: CrossAxisAlignment.start,
        children: [
            const SizedBox(height: 24.0),
            Text(
                'Title',
                style: TextStyle(
                    color: CustomColors.firebaseioGrey,
                    fontSize: 22.0,
                    letterSpacing: 1,
                    fontWeight: FontWeight.bold,
                ),
            ),
            const SizedBox(height: 8.0),
            CustomFormField(
                isLabelEnabled: false,
                controller: _titleController,
                focusNode: widget.titleFocusNode,
                keyboardType: TextInputType.text,
                inputAction: TextInputAction.next,
                validator: (value) => Validator.validateField(
                    value: value,
                ),
                label: 'Title',
                hint: 'Enter your news title',
            ),
            const SizedBox(height: 24.0),
            Text(
                'Description',
                style: TextStyle(
                    color: CustomColors.firebaseioGrey,
                    fontSize: 22.0,
                    letterSpacing: 1,
                    fontWeight: FontWeight.bold,
                ),
            ),
            const SizedBox(height: 8.0),
            CustomFormField(
                maxLines: 10,
                isLabelEnabled: false,
                controller: _descriptionController,
                focusNode: widget.descriptionFocusNode,
                keyboardType: TextInputType.text,
```

```
        inputAction: TextInputAction.done,
        validator: (value) => Validator.validateField(
            value: value,
        ),
        label: 'Description',
        hint: 'Enter your news description',
    ),
],
),
),
),
_isProcessing
? Padding(
    padding: const EdgeInsets.all(16.0),
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
    ),
)
: Padding(
    padding: const EdgeInsets.only(
        left: 30,
        right: 30,
    ),
    child: ElevatedButton(
        style: ButtonStyle(
            backgroundColor: MaterialStateProperty.all(
                CustomColors.firebaseioOrange,
            ),
            minimumSize: MaterialStateProperty.all(Size.zero),
            shape: MaterialStateProperty.all(
                RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(45),
                ),
            ),
        ),
        onPressed: () async {
            widget.titleFocusNode.unfocus();
            widget.descriptionFocusNode.unfocus();

            if (_editNewsFormKey.currentState!.validate()) {
                setState(() {
                    _isProcessing = true;
                });
            }
        },
    ),
)
```

```

        await Database.updateNews(
            docId: widget.documentId,
            title: _titleController.text,
            description: _descriptionController.text,
        );

        setState(() {
            _isProcessing = false;
        });

        Navigator.of(context).pop();
    },
),
child: Padding(
    padding: const EdgeInsets.only(top: 16.0, bottom: 16.0),
    child: Text(
        'Update News',
        style: TextStyle(
            fontSize: 24,
            fontWeight: FontWeight.bold,
            color: CustomColors.firebaseioGrey,
            letterSpacing: 2,
        ),
        ),
        ),
        ),
        ),
        ),
        ),
        ),
        ],
),
);
);
}
}

```

## GoogleSignInButton.dart

```

// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/screens/user_info_screen.dart';
import 'package:Kamataha/src/utils/auth.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class GoogleSignInButton extends StatefulWidget {

```

```
@override
_GoogleSignInButtonState createState() => _GoogleSignInButtonState();
}

class _GoogleSignInButtonState extends State<GoogleSignInButton> {
bool _isSigningIn = false;

@Override
Widget build(BuildContext context) {
return Padding(
padding: const EdgeInsets.only(bottom: 16.0),
child: _isSigningIn
? const CircularProgressIndicator(
valueColor: AlwaysStoppedAnimation<Color>(Colors.green),
)
: OutlinedButton(
style: ButtonStyle(
backgroundColor: MaterialStateProperty.all(Colors.white),
shape: MaterialStateProperty.all(
RoundedRectangleBorder(
borderRadius: BorderRadius.circular(40),
),
),
),
onPressed: () async {
setState(() {
_isSigningIn = true;
});
User? user =
await Authentication.signInWithGoogle(context: context);

setState(() {
_isSigningIn = false;
});

if (user != null) {
Navigator.of(context).pushReplacement(
MaterialPageRoute(
builder: (context) => UserInfoScreen(
user: user,
),
),
);
}
},
),
},
```

```

        child: Padding(
            padding: const EdgeInsets.fromLTRB(0, 10, 0, 10),
            child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: const <Widget>[
                    Image(
                        image: AssetImage("assets/google_logo.png"),
                        height: 35.0,
                    ),
                    Padding(
                        padding: EdgeInsets.only(left: 10),
                        child: Text(
                            'Sign in with Google',
                            style: TextStyle(
                                fontSize: 20,
                                color: Colors.black54,
                                fontWeight: FontWeight.w600,
                            ),
                        ),
                    ),
                ],
            ),
        ),
    );
}
}

```

## ReviewList.dart

```

// ignore_for_file: use_key_in_widget_constructors

import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/display_all_question.dart';
import 'package:Kamataha/src/screens/display_all.dart';
import 'package:Kamataha/src/utils/database.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class ReviewList extends StatefulWidget {
    const ReviewList({required User user}) : _user = user;

```

```
final User _user;
@Override
State<ReviewList> createState() => _ReviewListState();
}

class _ReviewListState extends State<ReviewList> {
late User _user;

@Override
void initState() {
    _user = widget._user;
    super.initState();
}

@Override
Widget build(BuildContext context) {
    return StreamBuilder<QuerySnapshot>(
        stream: Database.readReviews(),
        builder: (context, snapshot) {
            if (snapshot.hasError) {
                return const Text('Something went wrong');
            } else if (snapshot.hasData || snapshot.data != null) {
                if (snapshot.data!.docs.isEmpty) {
                    return Center(
                        child: Text(
                            "No reviews yet",
                            style: TextStyle(color: CustomColors.firebaseioGrey),
                        ),
                    );
                }
                return ListView.separated(
                    separatorBuilder: (context, index) => const SizedBox(height: 16.0),
                    itemCount: snapshot.data!.docs.length,
                    itemBuilder: (context, index) {
                        var noteInfo = snapshot.data!.docs[index].data() as dynamic;
                        String docID = snapshot.data!.docs[index].id;
                        String description = noteInfo["description"];

                        return Padding(
                            padding: const EdgeInsets.all(8.0),
                            child: Ink(
                                decoration: BoxDecoration(
                                    color: Color(0xFF9CF29A),
                                    borderRadius: BorderRadius.circular(8.0),
                            ),
                        ),
                    },
                );
            }
        },
    );
}
```

```

        child: ListTile(
            shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(8.0),
            ),
            title: Text(
                description,
                maxLines: 1,
                overflow: TextOverflow.ellipsis,
            ),
        ),
    )));
},
);
}

return Center(
    child: CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(
            CustomColors.firebaseioOrange,
        ),
    ),
);
},
);
});
}
}
}

```

## SearchInputWidget.dart

```

// ignore_for_file: use_key_in_widget_constructors
import 'package:Kamataha/src/res/custom_colors.dart';
import 'package:Kamataha/src/screens/search_screen.dart';
import 'package:Kamataha/src/utils/validator.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

// ignore: duplicate_ignore
class SearchInputWidget extends StatefulWidget {
    const SearchInputWidget({required User user}) : _user = user;

    final User _user;

    @override

```

```
    _SearchInputWidgetState createState() => _SearchInputWidgetState();
}

class _SearchInputWidgetState extends State<SearchInputWidget> {
    late User _user;

    @override
    void initState() {
        _user = widget._user;
        super.initState();
    }

    final _addItemFormKey = GlobalKey<FormState>();

    final TextEditingController _commentController = TextEditingController();

    @override
    Widget build(BuildContext context) {
        return Padding(
            padding: const EdgeInsets.only(
                left: 10,
                right: 10,
            ),
            child: Form(
                key: _addItemFormKey,
                child: TextFormField(
                    controller: _commentController,
                    style: TextStyle(
                        color: CustomColors.firebaseioYellow,
                    ),
                    validator: (value) => Validator.validateField(
                        value: _commentController.text,
                    ),
                    // controller: controller,
                    decoration: InputDecoration(
                        prefixIcon: const Icon(Icons.search),
                        labelText: "Search by author's name ",
                        hintText: "Note: case sensitive",
                        hintStyle: const TextStyle(color: Colors.red),
                        labelStyle: TextStyle(
                            color: CustomColors.firebaseioGrey,
                        ),
                        suffixIcon: IconButton(
                            icon: const Icon(Icons.send),
                            splashColor: Colors.blue,
                        ),
                    ),
                ),
            ),
        );
    }
}
```

```

        tooltip: "Post comment",
        onPressed: () async {
            if (_addItemFormKey.currentState!.validate()) {
                Navigator.of(context).push(MaterialPageRoute(
                    builder: (context) => SearchScreen(
                        input: _commentController.text,
                        user: _user,
                    )));
            }
        },
    )));
}
}

```

## CRUD OPERATIONS

```

static Future<void> addNews({
    required String title,
    required String description,
}) async {
    DocumentReference documentReferencer = _firestore.collection('news').doc();

    Map<String, dynamic> data = <String, dynamic>{
        "title": title,
        "description": description,
        "time": time,
    };

    await documentReferencer
        .set(data)
        .whenComplete(() => print("Note question added to the database"))
        .whenComplete(() => print("Note news added to the database"))
        .catchError((e) => print(e));
}

```

```

static Stream<QuerySnapshot> readNews() {
    Query<Map<String, dynamic>> notesItemCollection =
        _firestore.collection('news').orderBy("time", descending: true);
    print('fetched');
}

```

```
        return notesItemCollection.snapshots();
    }
```

```
static Future<void> updateNews({
    required String docId,
    required String title,
    required String description,
}) async {
    DocumentReference documentReferencer =
        _firestore.collection('news').doc(docId);

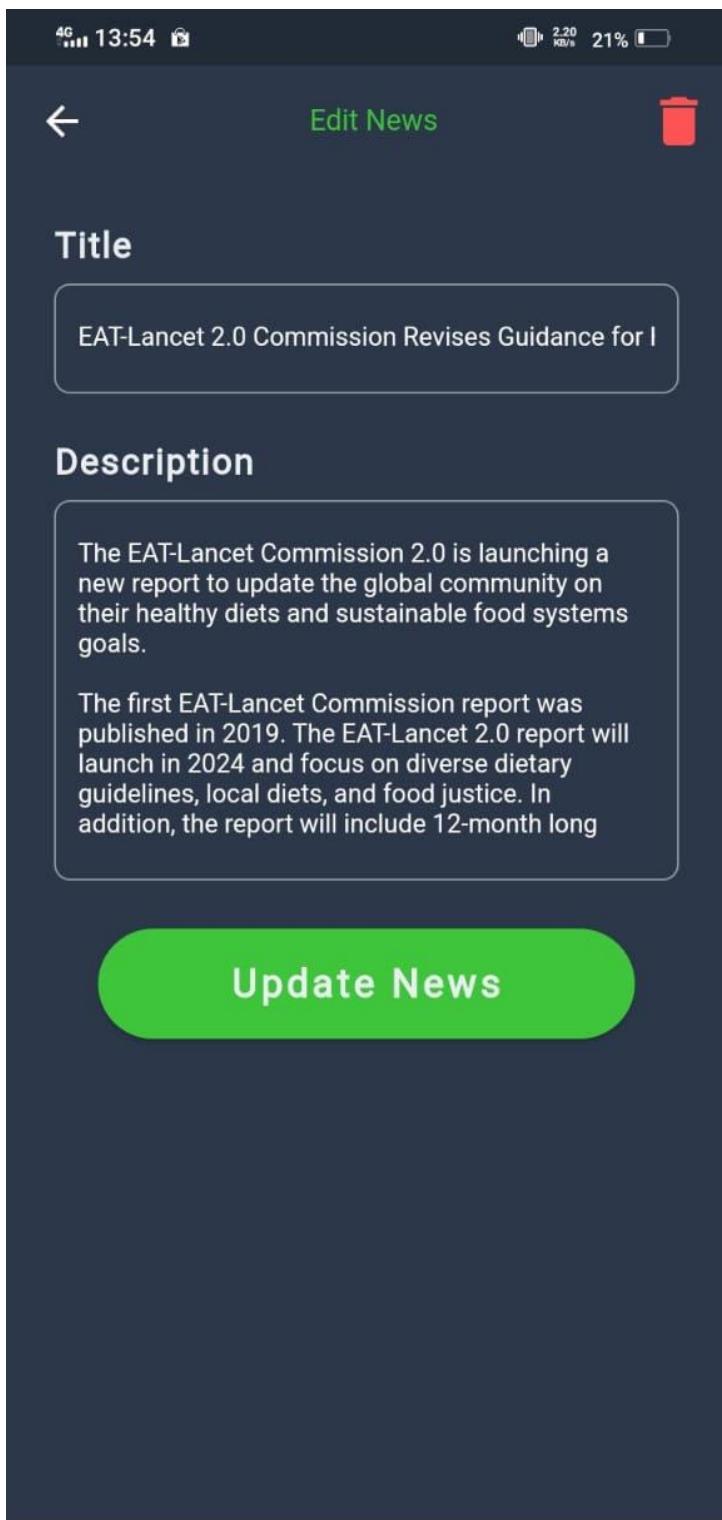
    Map<String, dynamic> data = <String, dynamic>{
        "title": title,
        "description": description
    };

    await documentReferencer
        .update(data)
        .whenComplete(() => print("Note news updated in the database"))
        .catchError((e) => print(e));
}
```

```
static Future<void> deleteNews({
    required String docId,
}) async {
    DocumentReference documentReferencer =
        _firestore.collection('news').doc(docId);

    await documentReferencer
        .delete()
        .whenComplete(() => print('Note news deleted from the database'))
        .catchError((e) => print(e));
}
```

## UI Screenshots



4G 13:54

5.60 KB/s 21%



කමත Kamatha



News

[view more](#)



Schemas



Articles



Comments



4G 13:54

0.00 KB/s 21%



## Admin Dashboard :

Please Enter your Admin Credentials for access to the admin dashboard

Enter your access key

Submit



4G 13:54

0.20 KB/s 21%



කමත Kamatha

### Title

Enter your title

### Description

Enter your Description

Post Your News



4G 13:55

0.90 KB/s 21%



කමත Kamatha

Testing 1

Testing review





കമത Kamatha

Title EAT-Lancet 2.0 Commission Revises Guidance for Healthy, Diverse Diets, and Sustainable Food Systems

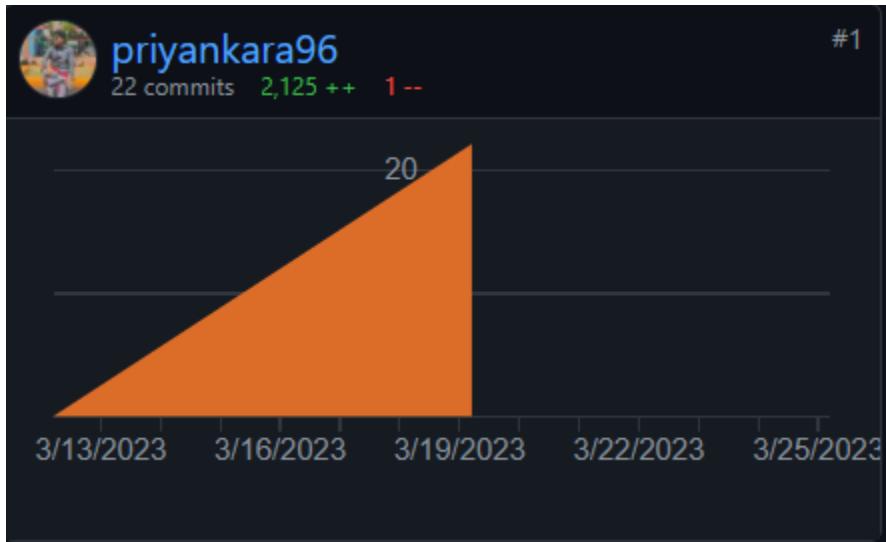
Posted by : Raveena

The EAT-Lancet Commission 2.0 is launching a new report to update the global community on their healthy diets and sustainable food systems goals.

The first EAT-Lancet Commission report was published in 2019. The EAT-Lancet 2.0 report will launch in 2024 and focus on diverse dietary guidelines, local diets, and food justice. In addition, the report will include 12-month long global consultations for the public and other interested global food systems stakeholders to share their thoughts on a transition to sustainable food systems, and Intergovernmental Panel on Climate Change (IPCC) modeling efforts to evaluate multiple pathways to sustainable food systems.

EDITREVIEWDELETE

## Contribution



## References

<https://firebase.google.com/docs>

<https://flutter.dev/learn>

<https://dart.dev/guides>