e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:04/Issue:03/March-2022 Impact Factor- 6.752 www.irjmets.com

MENTAL HEALTH TRACKER APPLICATION

T Sai Smrutik*1, Y Naveen Kumar*2, M Devraj*3,

Y Sai Krishna*4, G Uday Kiran*5

*1,2,3,4Student, Department Of Computer Science And Engineering, B V Raju Institute Of Technology, Narsapur, Telangana, India.

*5Assistant Professor, Department Of Computer Science And Engineering, B V Raju Institute Of Technology, Narsapur, Telangana, India.

ABSTRACT

Mental Health is a very important issue in today's world . As in these days, a lot of people now working from home and even companies are preferring work from home, the mental health situation of the people has nose-dived to unbelievable levels. The main objective of the proposed project focuses on building an application that will try to collect the mental state of the user and provide close to accurate remedies – assigning tasks, connecting a doctor(or psychiatrist) etc. The application tries to find if an app user is suffering from any mental issue by asking a few psychological questions and then based on those answers, it suggests measures that the app users can believe to recover from their issue and all this happens based on how the user answers the questions. Based on the answers that the users provide, the application will suggest tasks to them. The application will also take timely updates from the user and will update the data to the dashboard.

Keywords: Mental Health, Flutter, Dart, Android Studio, Firebase.

I. INTRODUCTION

Mobile applications are the current trend because of the exploding popularity and usage of smart phones and tablets. This prolific and highly growing industry has attracted many corporate firms from every corner of the market as consumers these days are avoiding desktop computing and switching to their mobile devices as they are really handy. The mobile application development has most of its roots from the traditional software development. Android Studio is the platform or tool that helps to create the mobile applications in a much easier way. Also mobile applications are built mostly using two languages – Java and Dart. Java being the most common software language, is most choice of the developers. Dart is named the official language for mobile application development by Google. We also have many frameworks that help to build a cleaner user interface. As mental health is one of the main issues in today's world, we are merging the problem with a solution of mobile application.

Objective

The main objective of the project is to include the basic aid of a mental issue into a mobile application that can be operated on user's fingertips. Also, keeping in count the user's state of mind our attempt/objective is to build a really user-friendly application that might not stress the user's mind. Our attempt is to build an application that will try to address the problem by asking a few questions to assess the mental health of the user and conclude that the user has a genuine mental health issue and then the application will provide the required remedies.

II. METHODOLOGY

Existing System

In the existing System, we can observe many applications that probably focus on physical health, but we rarely see an application that tries to remedy the psychological health of a person.

Applications like myfitnessPal, cult-fit, etc. focus on the physical health of a person, but there are existing applications like MindMantra et. Which don't have a proper stack of questions that will try to assess the user accurately. Also, the existing applications lack the basic features like a good and flexible user interface.

Proposed System

To facilitate a more efficient and effective health tracker application, the system should have a clean and flexible UI and the proposed system provides exactly that. Also the proposed system will hae a good stack of psychological questions, that will assess a user accurately, and ensures if the user has a genuine mental issue. Proper profiling of the users will be useful to correctly assess the users and update the user dashboard. Flutter is an accurate framework that greatly fits our requirements and hence used. It is a smooth and flexible UI framework developed by Google. Firebase is a Real-time database and is a cloud-based noSQL database that lets you store and sync the data of the users in real-time.

III. MODELING AND ANALYSIS

In this, we designed the system with three modules which are described below,

Visualize and Create Screens: It mainly includes two modules – Creating Custom Appbar and Create full-screen Questions interface page.

Set-up Auth flow: This stage includes modelling of login page, setting up Google sign-in (future work) and then retrieving user data from the database.

Plug data from app to Firebase: This stage includes the modules like – connecting local state-variables to firebase storage, setup and debug the transactions of Firebase data.

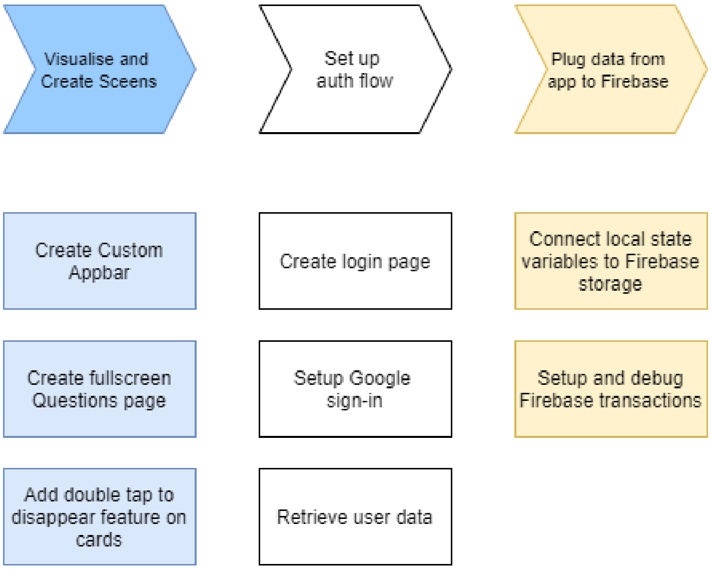

Visualise and Create Sceens
Set up auth flow
Plug data from app to Firebase
Create Custom Appbar
Create login page
Connect local state variables to Firebase storage
Create fullscreen Questions page
Setup Google sign-in
Setup and debug Firebase transactions
Add double tap to disappear feature on cards
Retrieve user data

Figure 1: System Architecture

IV. MODULE DESIGN AND ORGANIZATION

Our application is composed of five main modules, which can be illustrated by the below figure.

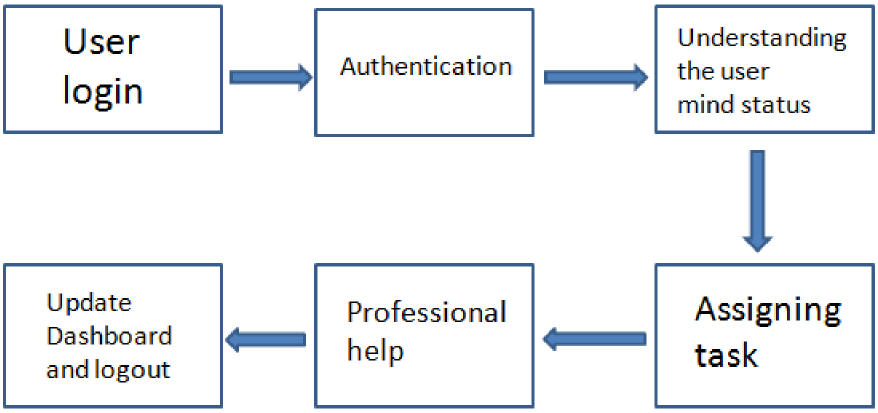

User login
Authentication
Understanding the user mind status
Assigning task
Professional help
Update Dashboard and logout

Figure 2: Data Flow Diagram

User Login:

The user will login/register which may be a Google sign-in or which may include logging into the application with the user personal credentials.

Knowing User's Mental Status:

Truffle After logging in, the application will have a user flexible prompt to answer to a few psychological questions.

After answering those questions, the correct mind-state of the user is determined and used for later purposes.

Update Dashboard:

After all the questions are answered, the answers are saved and the dashboard of the user will be updated.

Assigning tasks/ Getting professional's help:

The tasks are nothing but the remedies provided by the application for a given mind-state, be it listening to music or playing games or asking the user to do what they are particularly interested in doing. And if the health issue is severe or out of order, then the application may also suggest to consult a psychiatrist.

Updating Dashboard and Logout:

After all tasks are performed by the user, the dashboard changes and updates time to time.

V. IMPLEMENTATION AND TESTING

The key functions used in the project implementation include:

Building Screens:

We set-up the basic skeleton of the application first. Building an intuitive application and designing the elements in such a way that they look appealing.

Default Tab Controller along with Tab Bar View and Drawer are the dart classes along with flutter framework are mainly used to build the screens.

Questions Screen and UI Refinement:

This focuses on making the app friendly for target audience, as the potential users may be suffering from some mental illness and asking too many questions won't be a good decision.

So, we build the screens and functionality using the variables that are state managed.

Few of the psychological questions may include:

1. Do you ever feel that you've been affected by feelings of edginess, anxiety or nerves?
2. How frequently have you been doing things that mean something to you or your life?
3. Do you ever feel discomfort or stressed around other people?

Simple Algorithm for Classification:

Once we have the answers from the questions that the user answers, we will determine whether the person needs any help or not. In other words, whether the application needs to suggest some tasks for them to feel better. The complexity of the algorithm can range from simple comparative check to more advanced classification techniques. So, for now we are using simple comparisons to reduce the complexity and display the tasks to user according to threshold (say, if ans1 > some threshold) suggest the respective task.

Set-up Firebase Project:

Firebase is Google-backed database used to provide the backend services like authentication, cloud storage etc. Firebase is one of the easiest ways to get started with building a backend, which is a Google-backed application development.

Firebase is a real-time noSQL cloud-hosted database that helps its customers to store data of the users in real-time. Firebase can manage backend without the utmost concern of the users. The basic services of firebase can be shown from the below picture after creating and setting up the project.

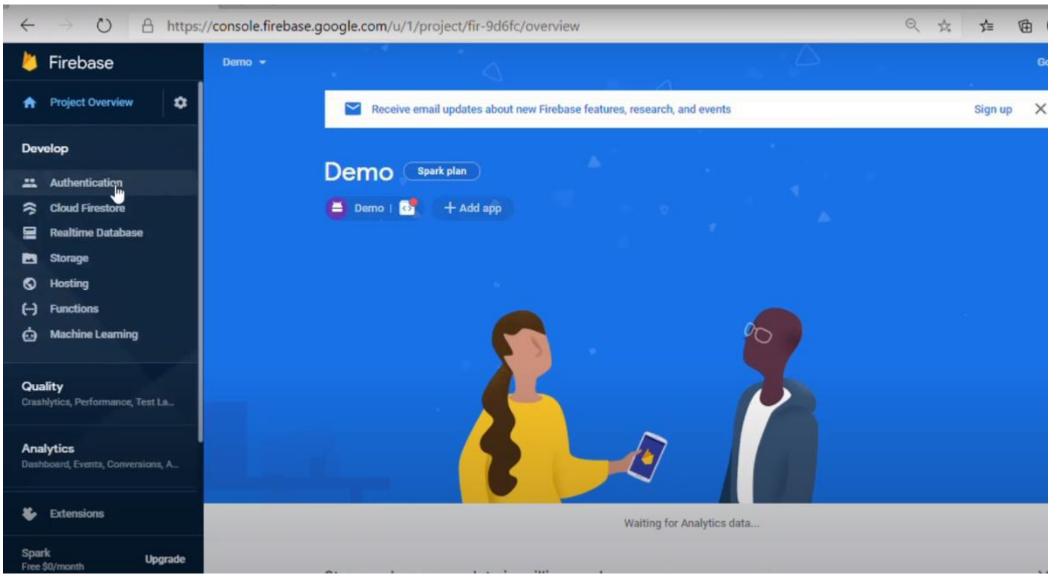
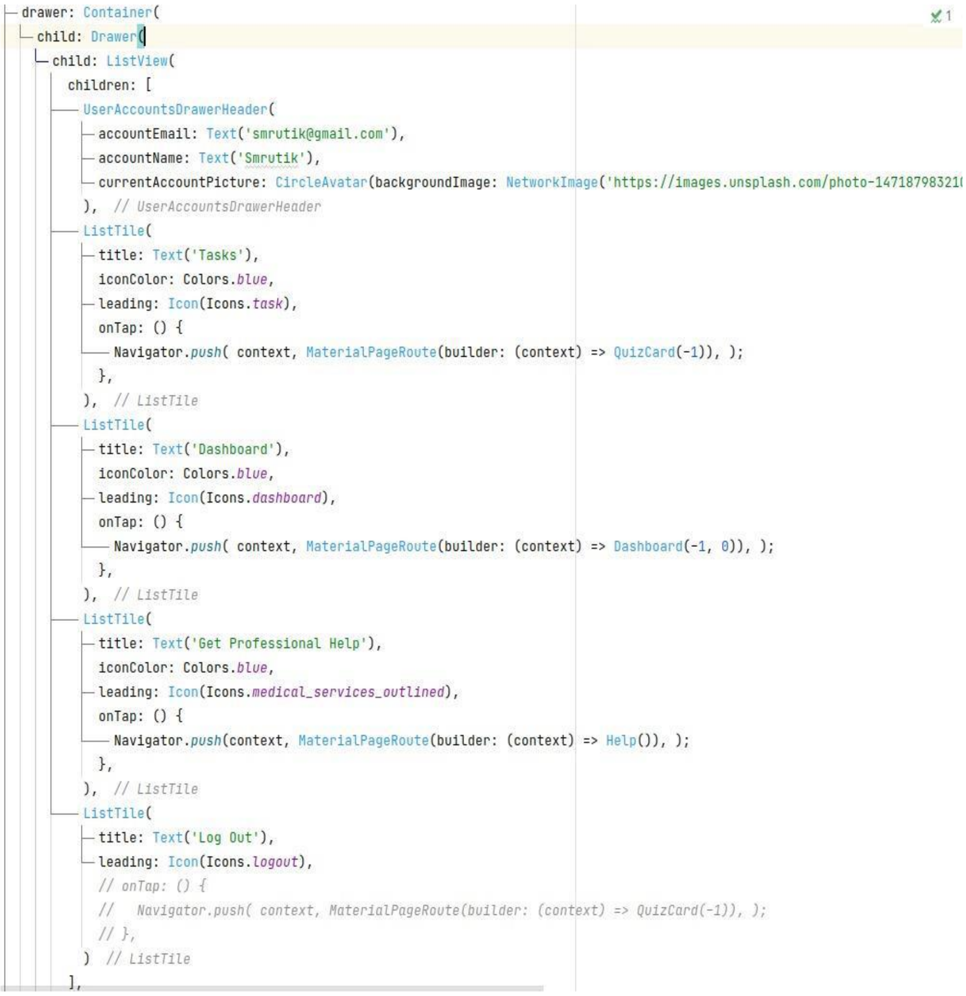

https://console.firebase.google.com/u/1/project/fir-9d6fc/overview
Firebase
Project Overview
Develop
Authentication
Cloud Firestore
Realtime Database
Storage
Hosting
Functions
Machine Learning
Quality
Crashlytics, Performance, Test La...
Analytics
Dashboard, Events, Conversions, A...
Extensions
Spark
Free $0/month
Upgrade
Demo
Receive email updates about new Firebase features, research, and events
Sign up
Demo
Spark plan
Demo
+ Add app
Waiting for Analytics data...

Figure 3: Firebase Console

Code:

Drawer with options:

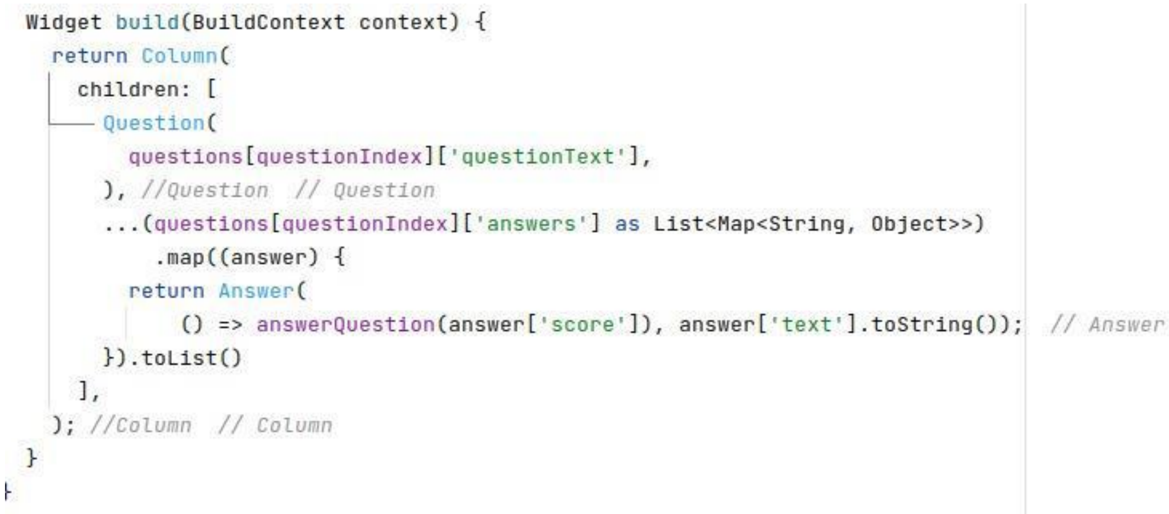
This code represents the implementation of side-drawer which mainly represents the account/profile details of the person, the task's view shortcut, dashboard shortcut, and the logout option screen.



```
drawer: Container(
  child: Drawer(
    child: ListView(
      children: [
        UserAccountsDrawerHeader(
          accountEmail: Text('smrutik@gmail.com'),
          accountName: Text('Smrutik'),
          currentAccountPicture: CircleAvatar(backgroundImage: NetworkImage('https://images.unsplash.com/photo-14718798321
        ), // UserAccountsDrawerHeader
        ListTile(
          title: Text('Tasks'),
          iconColor: Colors.blue,
          leading: Icon(Icons.task),
          onTap: () {
            Navigator.push( context, MaterialPageRoute(builder: (context) => QuizCard(-1)), );
          },
        ), // ListTile
        ListTile(
          title: Text('Dashboard'),
          iconColor: Colors.blue,
          leading: Icon(Icons.dashboard),
          onTap: () {
            Navigator.push( context, MaterialPageRoute(builder: (context) => Dashboard(-1, 0)), );
          },
        ), // ListTile
        ListTile(
          title: Text('Get Professional Help'),
          iconColor: Colors.blue,
          leading: Icon(Icons.medical_services_outlined),
          onTap: () {
            Navigator.push(context, MaterialPageRoute(builder: (context) => Help()), );
          },
        ), // ListTile
        ListTile(
          title: Text('Log Out'),
          leading: Icon(Icons.logout),
          // onTap: () {
          //   Navigator.push( context, MaterialPageRoute(builder: (context) => QuizCard(-1)), );
          // },
        ) // ListTile
      ],
```

Screenshot 1: code for Side Drawer

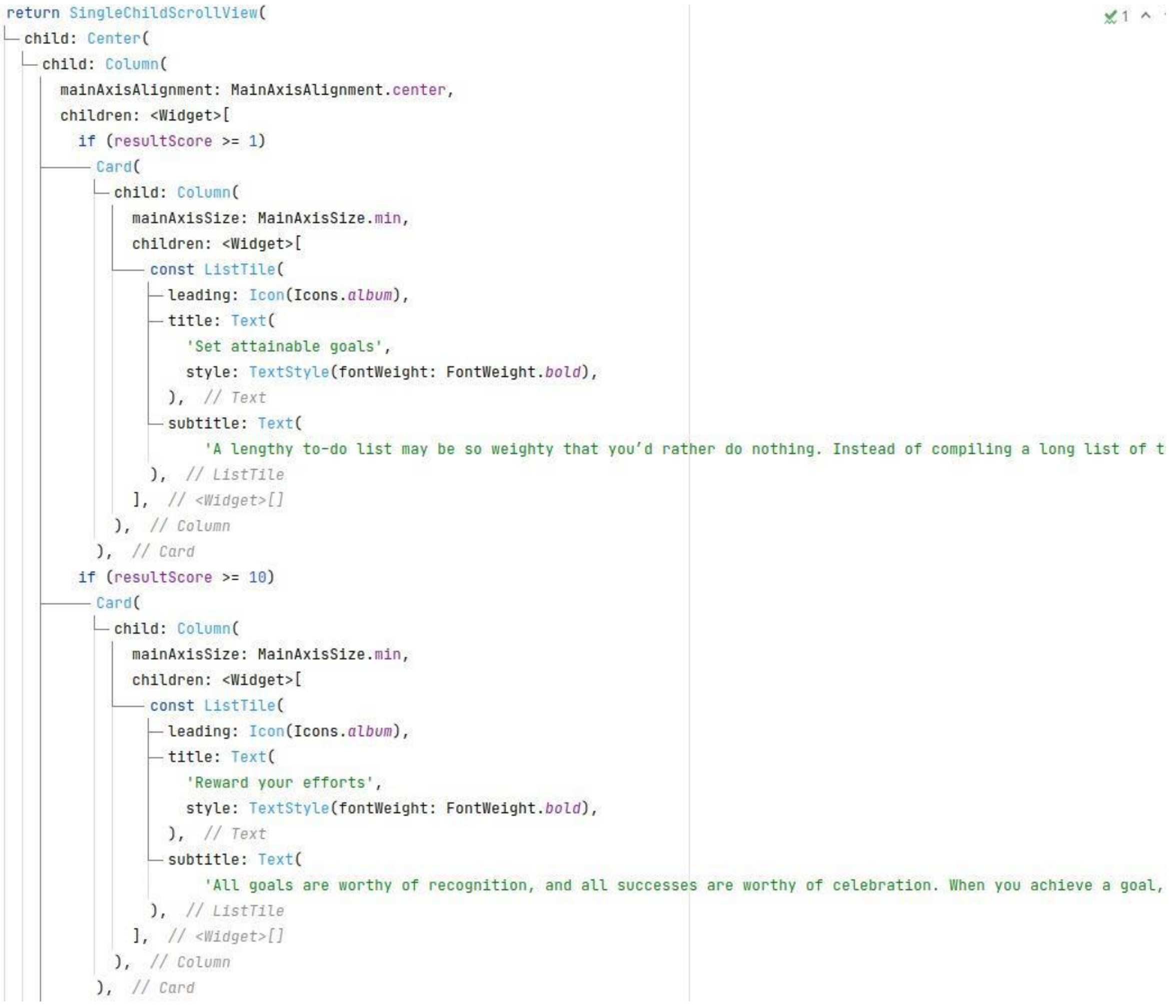
Question view:



```
Widget build(BuildContext context) {
  return Column(
    children: [
      Question(
        questions[questionIndex]['questionText'],
      ), //Question  // Question
      ...(questions[questionIndex]['answers'] as List<Map<String, Object>>)
          .map((answer) {
        return Answer(
            () => answerQuestion(answer['score']), answer['text'].toString());  // Answer
      }).toList()
    ],
  ); //Column  // Column
}
}
```

Screenshot 2: code for Questions view

Classifying Algorithm:



```
return SingleChildScrollView(
  child: Center(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: <Widget>[
        if (resultScore >= 1)
          Card(
            child: Column(
              mainAxisSize: MainAxisSize.min,
              children: <Widget>[
                const ListTile(
                  leading: Icon(Icons.album),
                  title: Text(
                    'Set attainable goals',
                    style: TextStyle(fontWeight: FontWeight.bold),
                  ), // Text
                  subtitle: Text(
                      'A lengthy to-do list may be so weighty that you'd rather do nothing. Instead of compiling a long list of t
                ), // ListTile
              ], // <Widget>[]
            ), // Column
          ), // Card
        if (resultScore >= 10)
          Card(
            child: Column(
              mainAxisSize: MainAxisSize.min,
              children: <Widget>[
                const ListTile(
                  leading: Icon(Icons.album),
                  title: Text(
                    'Reward your efforts',
                    style: TextStyle(fontWeight: FontWeight.bold),
                  ), // Text
                  subtitle: Text(
                      'All goals are worthy of recognition, and all successes are worthy of celebration. When you achieve a goal,
                ), // ListTile
              ], // <Widget>[]
            ), // Column
          ), // Card
```

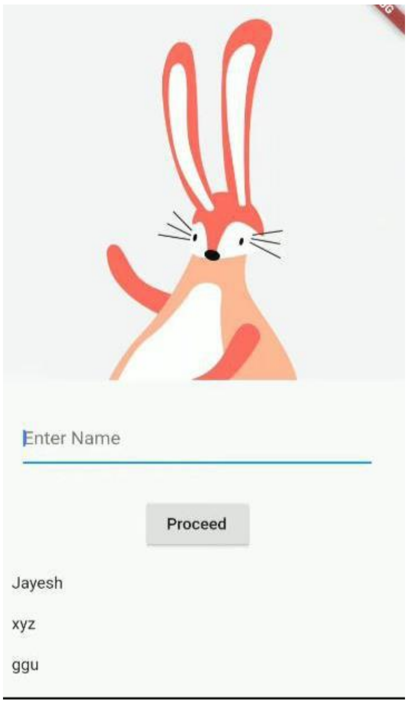
Screenshot 3: code for Algorithm

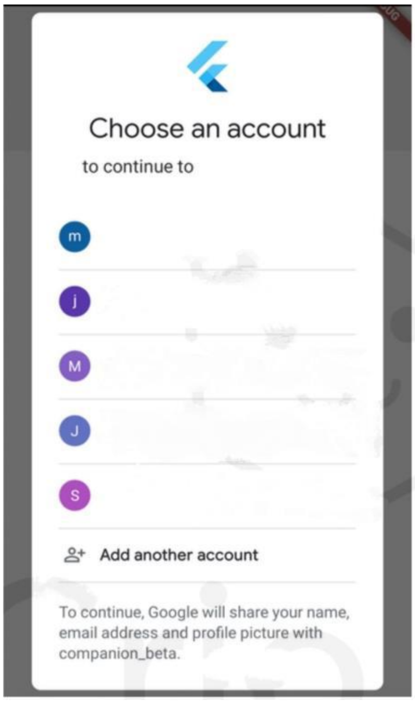
Output Screens:

Login page:

The process by which an individual gains access by proper authentication and identification of the user in the database. The user credentials will most probably consists of a username/email and its matching password.

Logging into any mobile/web application is usually used to get inside of a specific page, which won't be visible to hackers/exploiters. Once the user logs in, the login token can be used to track the user's actions while connected to the site. The users can perform the operations inside the application only after proper authentication using username and password or else the user should register first to get login credentials.

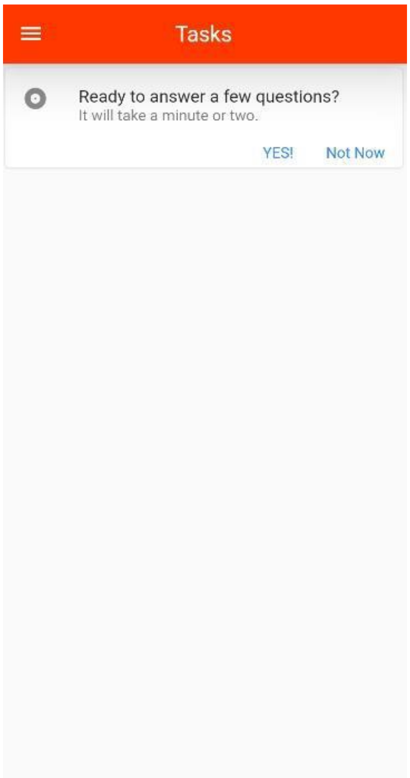




To continue, Google will share your name, email address and profile picture with companion_beta.

Tasks page:

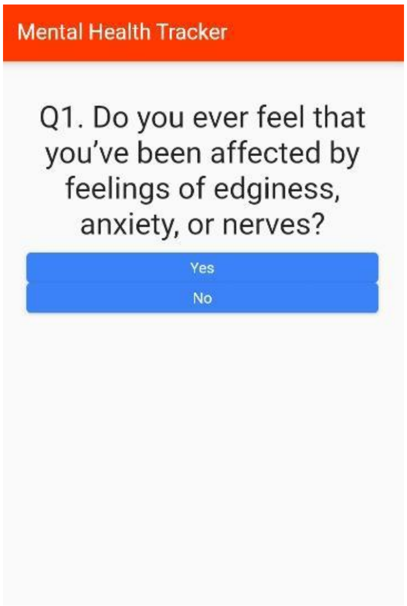
This page displays the tasks and Dashboard of the user controlled by a DefaultTabController and TabBarView. The DefaultTabController is the controller of the tabs that is written in the AppBar part of the application screen, while TabBarView is mainly responsible for the view of the tasks that is visible in the UI to the user.



Ready to answer a few questions?
It will take a minute or two.

Display of Questions:

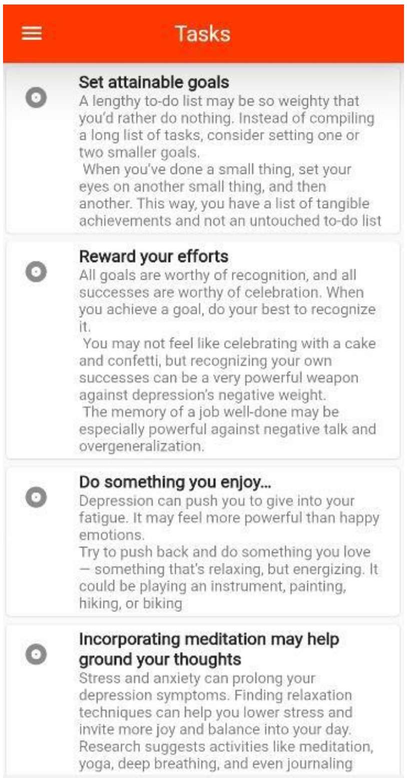
The psychological questions are displayed in full-screen mode one after another whose answers are being stored for further formulation of tasks and dashboard.



Q1. Do you ever feel that you've been affected by feelings of edginess, anxiety, or nerves?

Assigning tasks based on answers:

After the user answers all the questions displayed, the answers are saved and then used to analyze the state of mind of the user, and respective tasks are assigned to the user to freshen up his mind.



○ **Set attainable goals**
A lengthy to-do list may be so weighty that you'd rather do nothing. Instead of compiling a long list of tasks, consider setting one or two smaller goals.
When you've done a small thing, set your eyes on another small thing, and then another. This way, you have a list of tangible achievements and not an untouched to-do list

○ **Reward your efforts**
All goals are worthy of recognition, and all successes are worthy of celebration. When you achieve a goal, do your best to recognize it.
You may not feel like celebrating with a cake and confetti, but recognizing your own successes can be a very powerful weapon against depression's negative weight.
The memory of a job well-done may be especially powerful against negative talk and overgeneralization.

○ **Do something you enjoy...**
Depression can push you to give into your fatigue. It may feel more powerful than happy emotions.
Try to push back and do something you love — something that's relaxing, but energizing. It could be playing an instrument, painting, hiking, or biking

○ **Incorporating meditation may help ground your thoughts**
Stress and anxiety can prolong your depression symptoms. Finding relaxation techniques can help you lower stress and invite more joy and balance into your day.
Research suggests activities like meditation, yoga, deep breathing, and even journaling

VI. RESULTS AND DISCUSSION

Result Analysis:

After building the application, the main result of the application is to display the questions to be answered and after the questions are answered, displaying the tasks based on those answers. The questions' screen and the tasks screen can be viewed from the above figures.

VII. CONCLUSION

Here we have created a mobile application using Dart language and flutter framework as an addition. This project involves building a simple application using Flutter that tracks the mental health of its users and tries to help them get over their mental condition by suggesting tasks and keeping records of their progress.

The user answers some questions and based on those answers, the tasks are assigned.

VIII. FUTURE WORK

Further improvements can include the following:

1. Diversify many other mental health issues including bipolar disorder, child autism, etc.
2. Bring out the Google sign-in feature to make the authentication process more robust.
3. Build a more softer and flexible UI.
4. Bring out more accurate classifying algorithms to differentiate the users based on their state of mind.

IX. REFERENCES

[1] https://flutter.dev/?ds&gclsrc=ds
[2] https://dart.dev/guides
[3] https://developer.android.com/docs
[4] https://firebase.google.com/?gclsrc=ds&gclsrc=ds&gclid=CPuo0Mfbn_MCFcXejgodF-MJ8Q