Mobile Application Development

STRESS AND ANXIETY DETECTOR

A024 KUNJ JARIWALA -70012000200

A041 MEET PATEL - 70012000241

A074 KARAN JHAVERI -70012000258

1. Overview

-User Needs (establish the need/originality of the app).

This app is very useful to track stress/anxiety/depression in daily life.
This app will help the people to track their daily mood and visualize it in the form of graphs and also there are many resources given in the app according to the mood of the person which will help the person to recover from depression or any type of mood swings and this is been recommended by the system according to the user input from the database and the recommendation algo.

-Problem Statement (Application of the project).

Making an app which will help people to track their mood and detect if they are depressed or not . App will also provide you with various functionality like mood tracking , resources library , notes making etc.

2. Description

a. List & explanation of (learned) Tools/Components used.

1. **TextView:**

A TextView is a UI component in Android that is used to display text on the screen. It can be customized with various attributes such as color, size, and font, and can display plain text or HTML-formatted text. TextViews are often used to display labels, headings, or paragraphs of text in an Android app.

2. **EditText:**

An EditText is a UI component in Android that is used to accept user input, such as text entered by the user. It can be customized with various attributes such as input type, hint text, and validation rules. EditTexts are often used in forms, search bars, or chat interfaces to allow users to enter text.

3. **Button:**

A Button is a UI component in Android that the user can click to perform an action. It can be customized with various attributes such as text, color, and size. Buttons are often used to trigger events or navigate to different screens in an Android app.

4. **ImageView:**

An ImageView is a UI component in Android that is used to display images on the screen. It can be customized with various attributes such as scale type, source, and size. ImageViews are often used to display graphics, icons, or photos in an Android app.

5. **LinearLayout:**

A LinearLayout is a layout manager in Android that is used to arrange UI components in a single row or column. It can be customized with various attributes such as orientation, gravity, and weight. LinearLayouts are often used to create simple, linear layouts in an Android app.

6. **RelativeLayout:**

A RelativeLayout is a layout manager in Android that is used to arrange UI components relative to each other. It can be customized with various attributes such as alignment, margins, and padding. RelativeLayouts are often used to create complex, relative layouts in an Android app.

7. **ConstraintLayout:**

A ConstraintLayout is a layout manager in Android that is used to create complex layouts by specifying constraints between UI components. It can be customized with various attributes such as constraints, bias, and chains. ConstraintLayouts are often used to create responsive, flexible layouts in an Android app.

8. **TextUtils:**

TextUtils is a class in Android that provides various utility methods for working with text. Some of its methods include isEmpty(), which checks if a given string is empty or null, and join(), which concatenates an array of strings into a single string.

b. List & explanation of (self-learned) Tools/Components used.

1. **IntroductionActivity**

Activity launched the first time the app is installed in order to introduce the user to the application purpose/features. This activity contains a series of slides that are implemented in layouts and asks for permissions required to use certain features of the app.

2. **ResourceDao**

A Data Access Object (Dao) is the bridge between the user attempting to interact with the lower-level database and the raw database. The access object allows you to perform operations, retrieve data etc.

3. **Resource Fragment class**

This has the code which specifies different parameters according to the mood selected by the user like resource recommendation , changing of backgrounds etc.

4. **Statistical Fragments**

Fragment responsible for displaying the statistics information to the user. This information is mainly the moods graph that displays a user's moods over the past week.

5. **RegisterNotificationChannel function**

Registers a notification channel which is required to post notifications to the user. This is done repeatedly whenever the app is started but there is no problem with calling .createNotificationChannel repeatedly.

6. **setupResourcesDatabase function**

Set up the Resources database with all the articles.

8)Calling Alarm services and notification in the mobile phone

7. Preference. xml file

This gives an option to enable and disable the notification function and resource function.

8. ProgressBar

a UI component used to display progress, such as the progress of a file download.

9. RecyclerView

A UI component used to display a list of items. It is commonly used in chat applications to display a list of chat messages

i.

10. TextWatcher

A TextWatcher is an interface in Android that allows you to listen for changes to an EditText view. It has three methods: beforeTextChanged, onTextChanged, and afterTextChanged, which are called when the user modifies the text in the EditText.

i.

11. SpannableStringBuilder

i.

A SpannableStringBuilder is a class in Android that allows you to build a string with multiple styles or formatting options. It's often used to highlight or emphasize certain parts of a string, such as keywords in a search result.

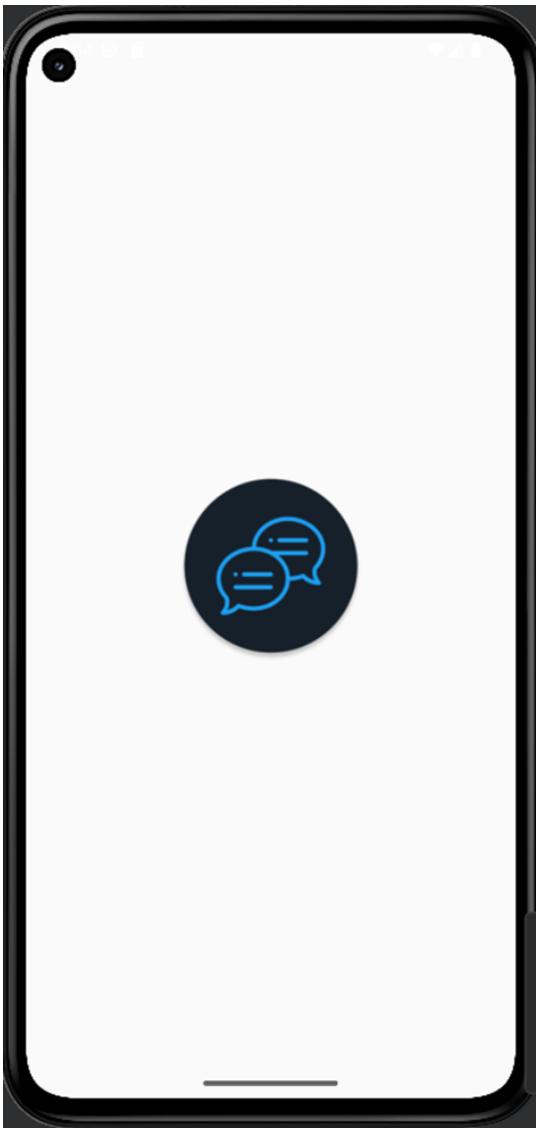
12. ScrollView:

A ScrollView is a UI component in Android that is used to enable scrolling when the content of the screen is too large to fit on the screen. It can be customized with various attributes such as scrollbars, padding, and layout mode. ScrollViews are often used to display long lists or views with large content in an Android app.

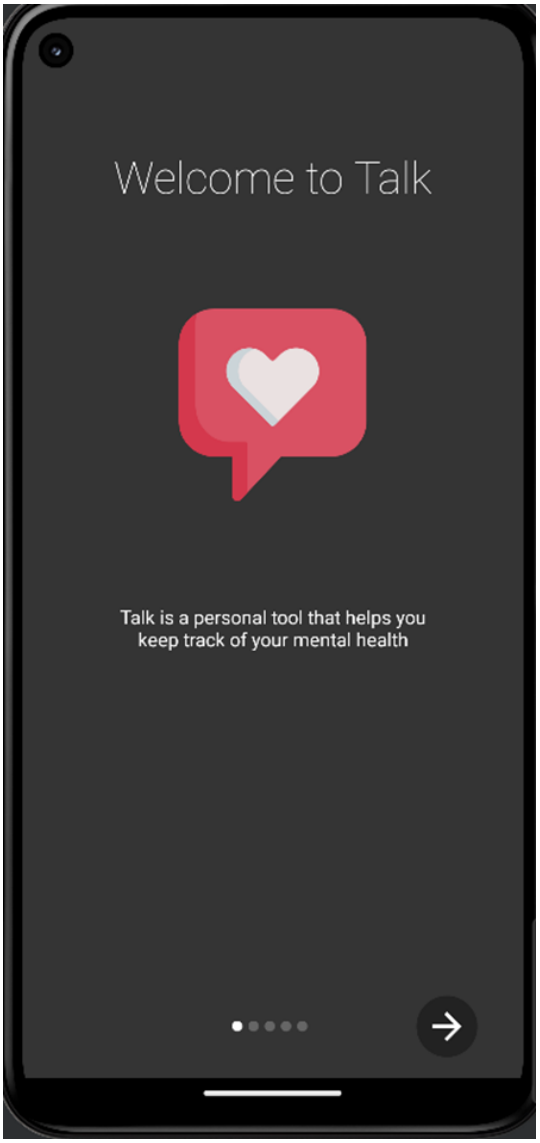
3. Standardized Workable App

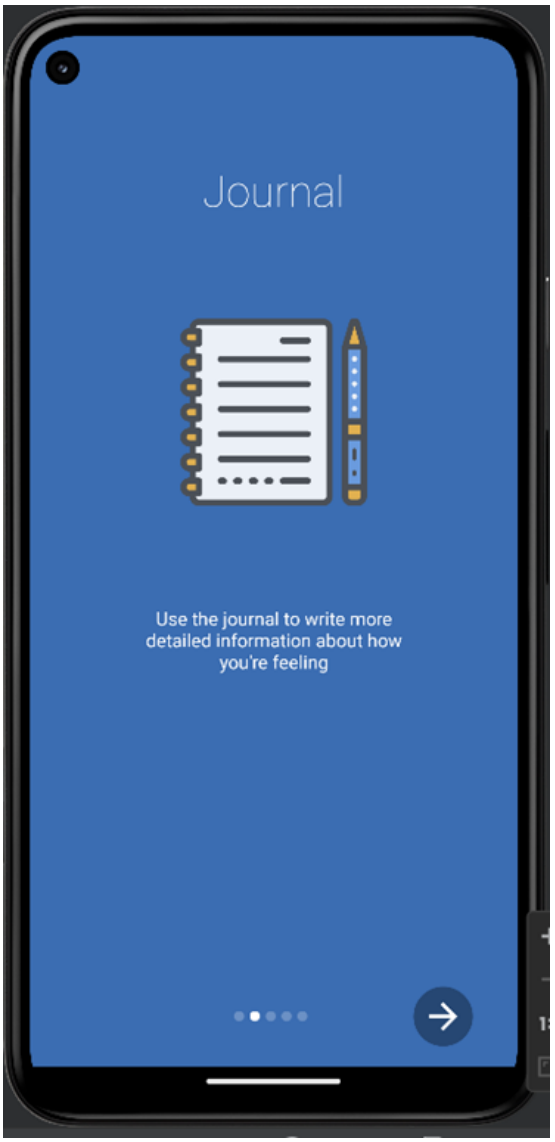
4. User Manual (Step-by-Step screenshot and brief about the working of app).

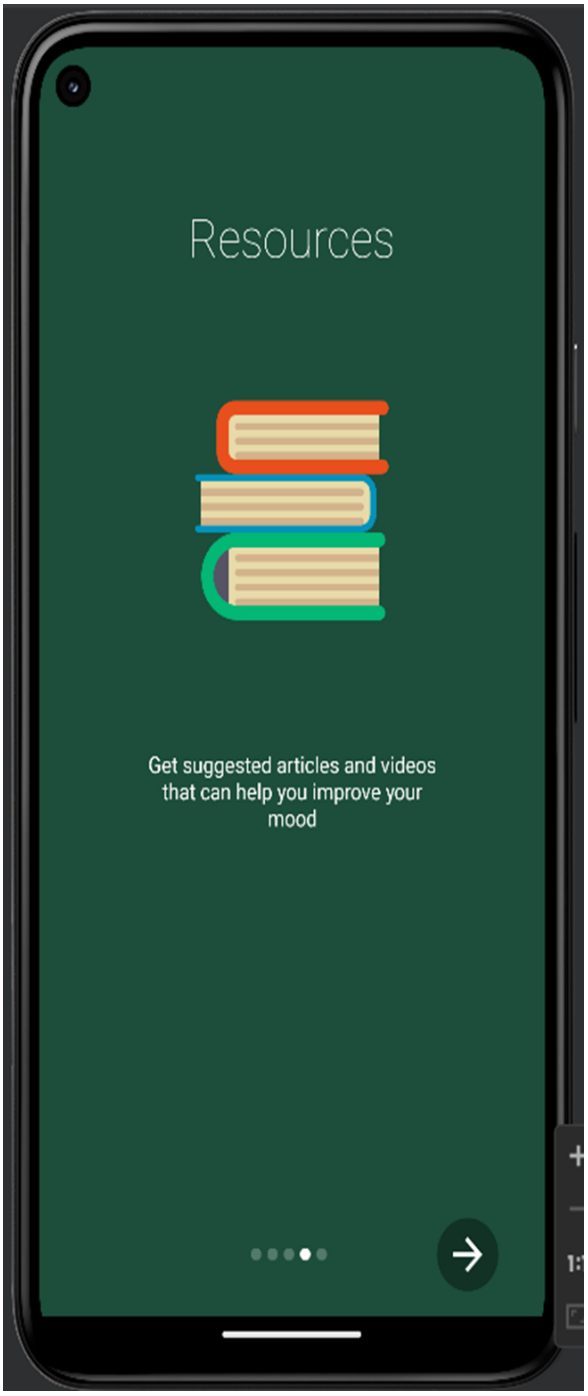
1)THIS IS THE SYMBOL OF OUR APP WHICH IS DISPLAYED IN THE START FOR 1-2 SEC.

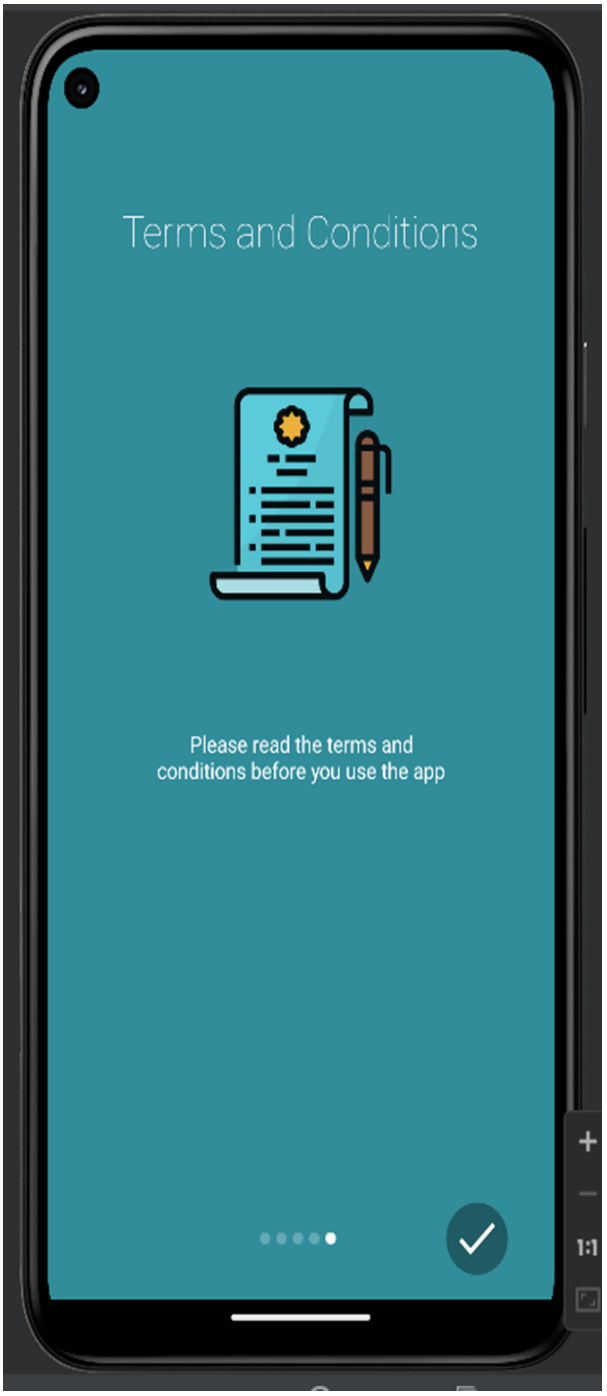


2)THIS IS THE WELCOME SCREEN WHICH IS DISPLAYED AND CHANGES BY CLICKING ON ARROW THIS WILL GUILD THE NEW USER ON HOW TO USE THE APP AND WHAT ARE THE DIFFERENT FUNCTIONALITY THAT THE USERS CAN USE. LIKE : JOURNAL, MOOD GRAPH ,RESOURCES.

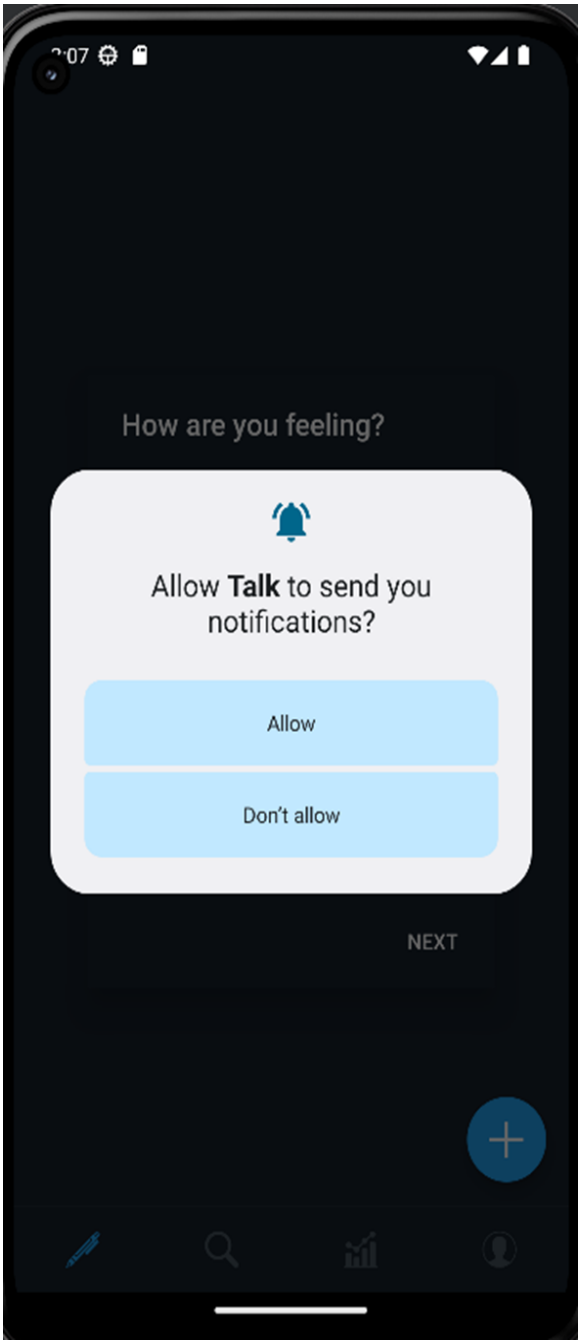




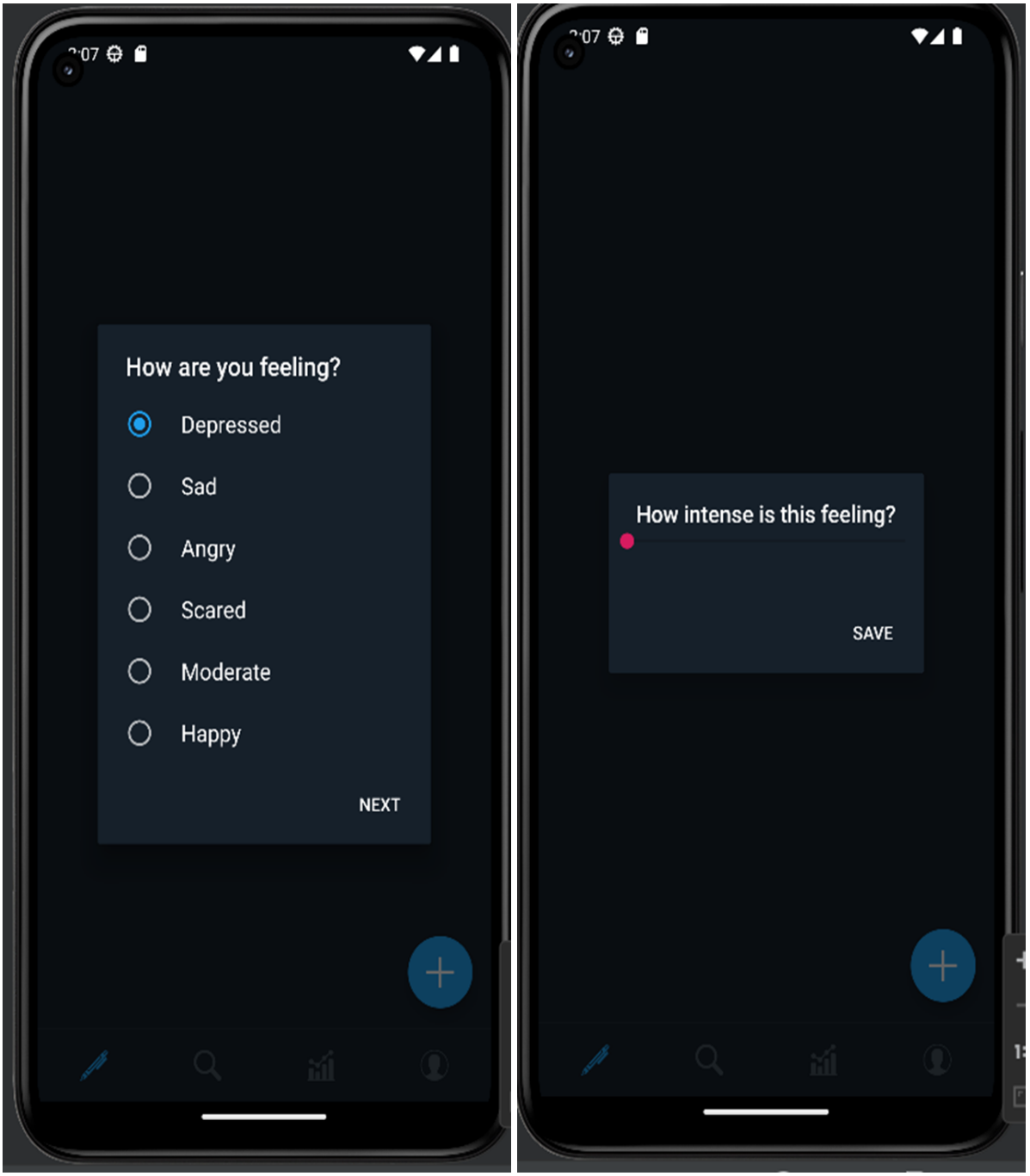

Resources
Get suggested articles and videos that can help you improve your mood


Terms and Conditions
Please read the terms and conditions before you use the app

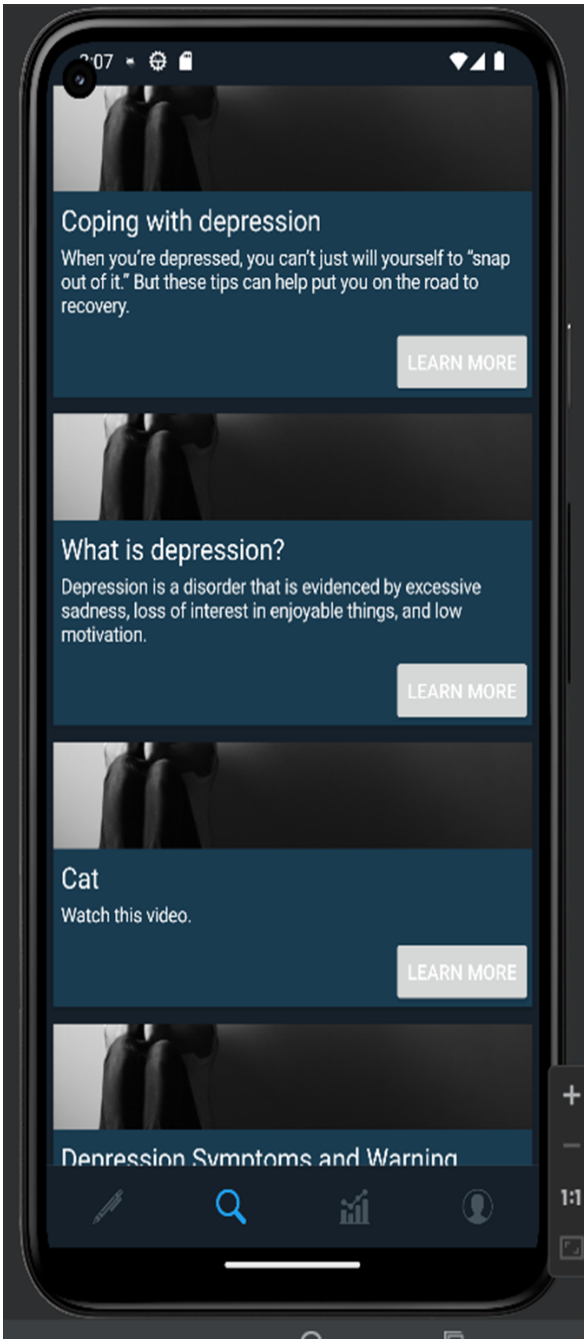
3)THIS ALERT BOX WILL ASK YOU THAT IF YOU WANT TO ALLOW THE NOTIFICATION TO BE RECEIVED ON ON YOUR MOBILE PHONE.

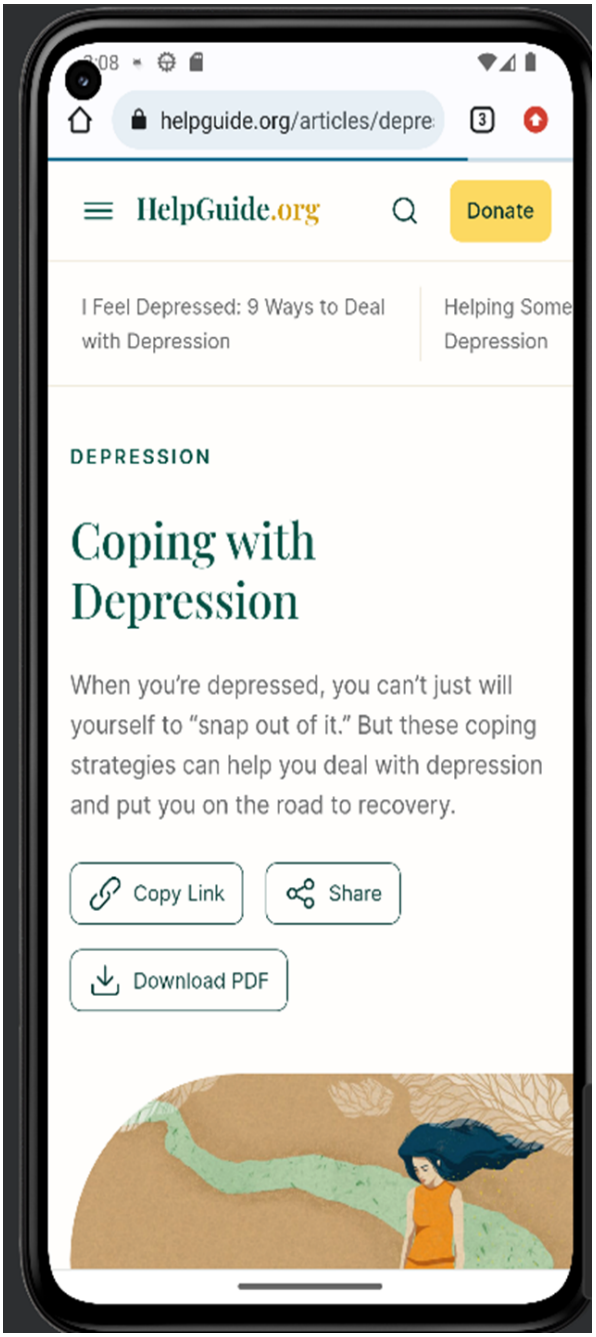


4)THIS ALERT BOX WILL ASK YOU TO ENTER YOUR STATUS ON THE MOOD THAT YOU ARE FEELING ON THE MOMENT AND AFTER THAT IT WILL ASK THE INTENSITY OF THAT FEELING.

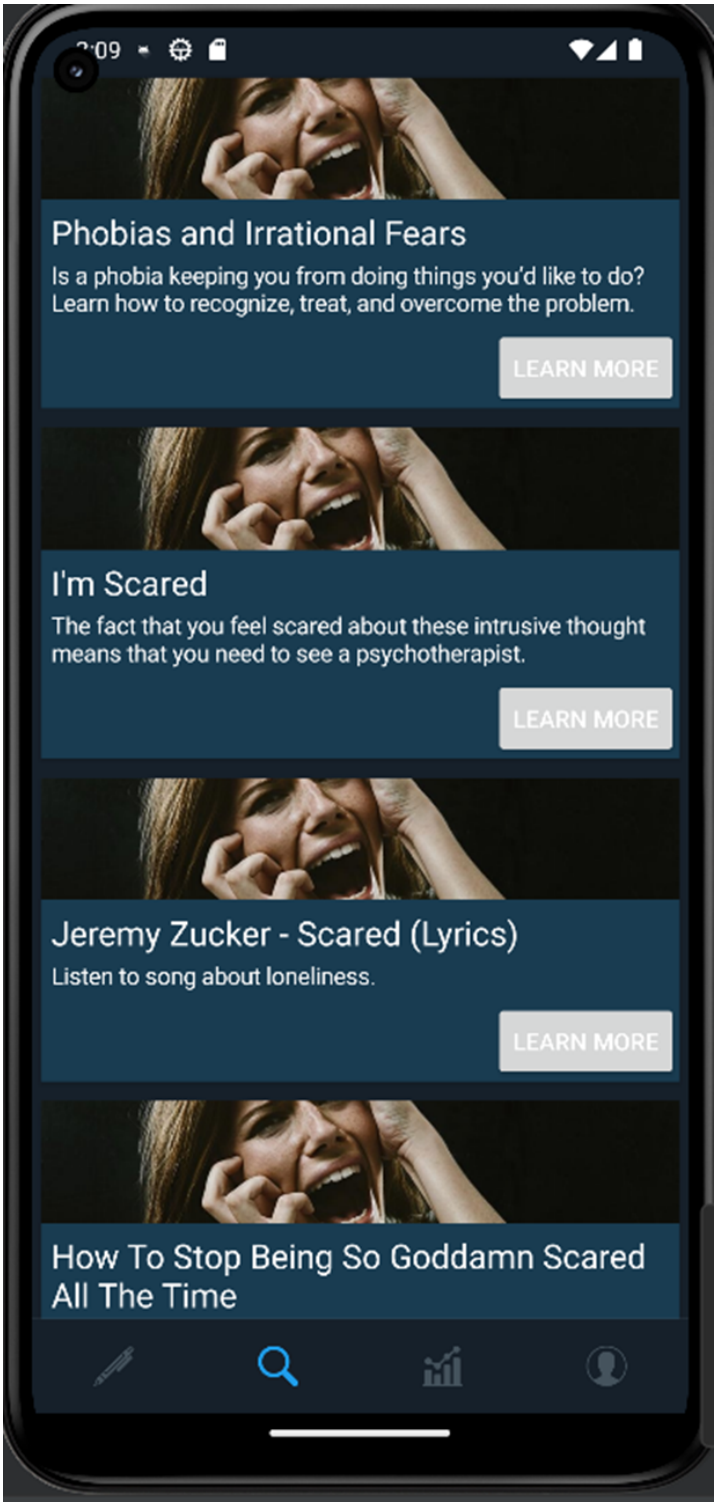


5)THESE ARE THE SOME OF THE RESOURCES WHICH ARE AVAILABLE FOR THE USER ACCORDING TO THE INPUT OF THE USER. YOU CAN SEE THIS BY CLICKING ON THE MAGNIFICATION ICON IN BOTTOM BAR.

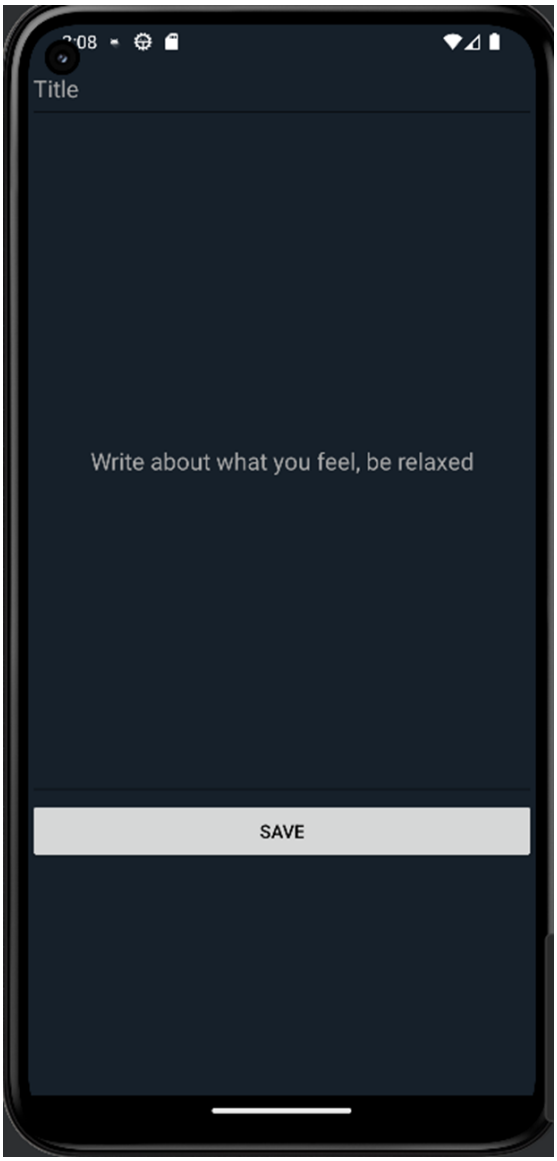


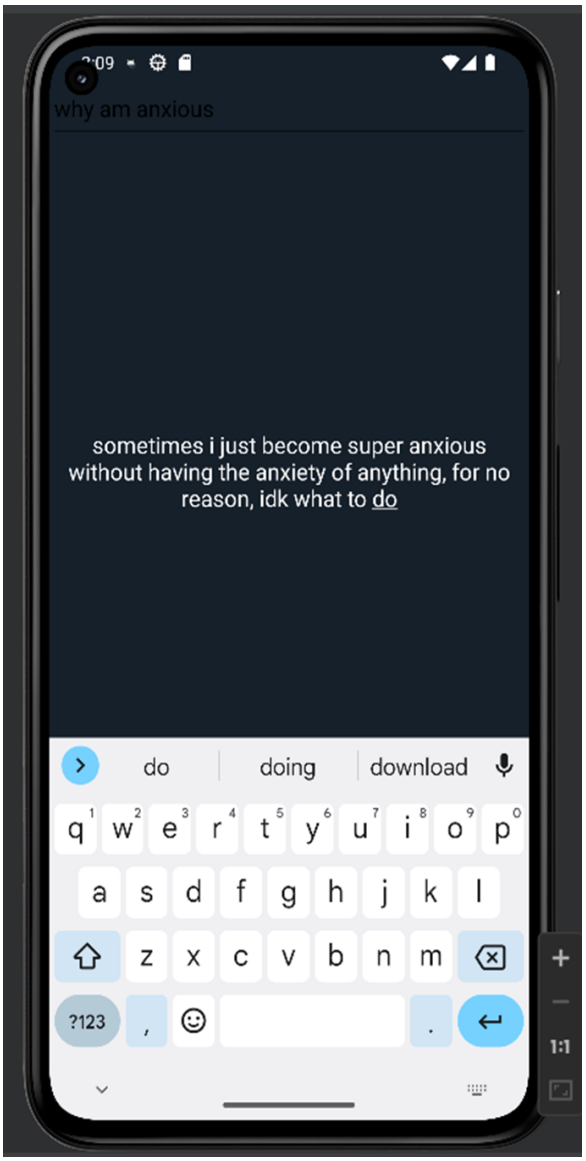


THE WALLPAPER OF THE RESOURCES ALSO CHANGES ACCORDING TO THE MOOD THAT THE USER HAS INPUT IN THE BEGINNING OF THE APP.

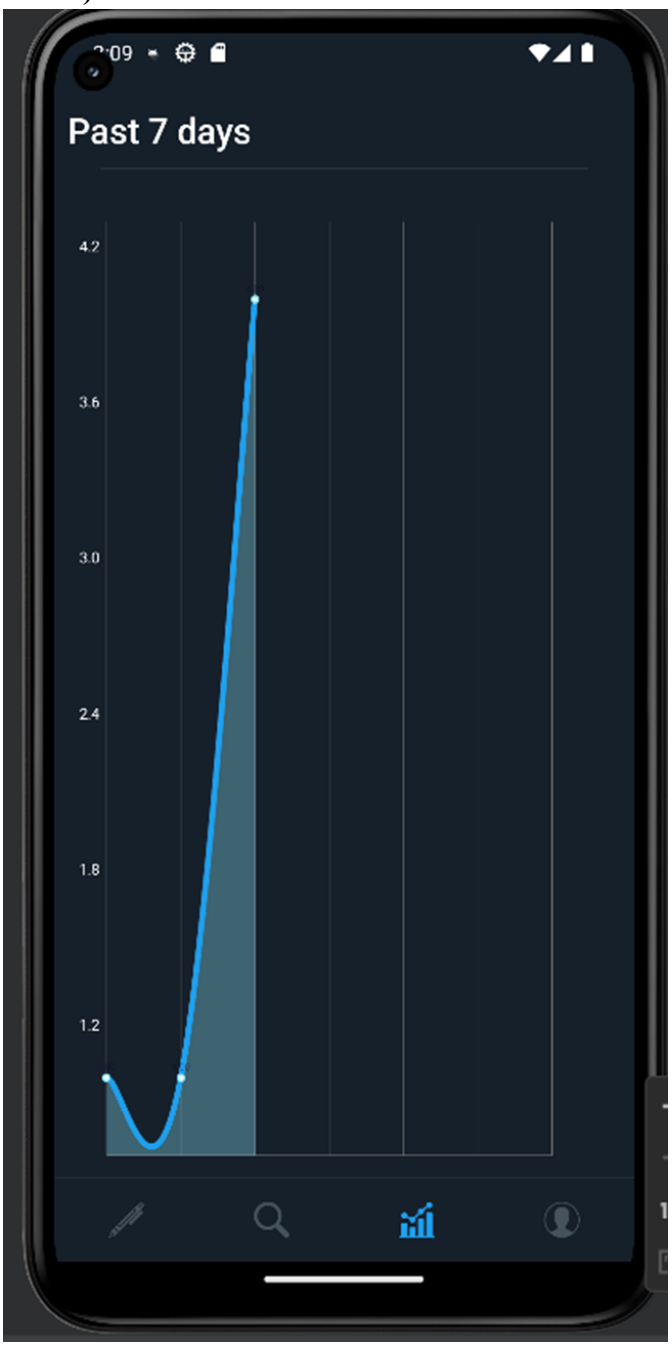


6)YOU CAN CLICK ON PEN ICON TO ACCESS THE JOURNAL AND WRITE DOWN WHATEVER YOU ARE FEELING AND RECORD IT FOR THE FUTURE WHICH CAN BE HELPFUL TO SHOW THE DOCTOR YOUR MENTAL STATE OF MIND IN THE FUTURE.





7)THIS IS THE MOOD GRAPH WHICH IS DISPLAYED AND CHANGES ON MULTIPLE INPUT AND SHOWS YOU YOUR PROGRESS FOR PAST 1 WEEK (7 DAYS).



5)

a. Reference Material Design topics used for which components.

. Material Design concepts are applied to some of these components which we used in the application we made which is talk app:

1. **IntroductionActivity -**

   The use of a series of slides to introduce the app's purpose/features is a common Material Design pattern. The use of layouts and asking for permissions are also typical Material Design practices.

2. **Integration of database for mood detection -**

Material Design doesn't have any specific guidelines for integrating databases, but it does emphasize consistency and ease of use, so it's likely that these principles were taken into account during the implementation.

3. **ResourceDao -**

This component is related to database access, and Material Design doesn't have any specific guidelines for this. However, Material Design principles emphasize clarity and simplicity, so it's likely that these principles were taken into account during the implementation.

4. **Resource Fragment class -**

This component seems to be related to user interface design, and Material Design provides guidelines for designing UI elements like buttons, lists, and cards.

5. **Statistical Fragments -**

Similar to the Resource Fragment, this component is related to user interface design and Material Design provides guidelines for designing UI elements like charts and graphs.

6. **RegisterNotificationChannel function -**

Notifications are an important aspect of Material Design, and there are guidelines for designing notifications to be clear and easy to understand.

7. **setupResourcesDatabase function -**

This is related to database access.

8. **Calling Alarm services and notification in the mobile phone -**

Notifications are an important aspect of Material Design, and there are guidelines for designing notifications to be clear and easy to understand.

9. **Preference. xml file** -

Material Design provides guidelines for designing settings screens, including the use of switches and checkboxes.

10. **ProgressBar -**

Material Design provides guidelines for designing progress bars, including the use of color and animation.

11. **RecyclerView -**

Material Design provides guidelines for designing lists and cards, which are often used with RecyclerView.

12. **TextWatcher -**

This is related to user input, and Material Design provides guidelines for designing text input fields and labels.

13. **SpannableStringBuilder -**

This is related to text formatting, and Material Design provides guidelines for typography and font usage.

c)Database used.

The database that we used for our project is MySQL. This can be seen in the **config/database.php** file, where the database configuration is defined and MySQL is specified as the database driver. Additionally, there are several MySQL-specific functions and queries used throughout the codebase, such as the use of **mysql_real_escape_string()** and the **LIMIT** clause in SQL queries.

5. Modules

    a. Block diagram/description of modules of the app.

Description of modules in our application talk can be provided as:

1. **index.js**:

This is the main entry point of the application, which sets up the Express app, sets up the server to listen for incoming requests, and configures various middleware functions.

2. **app.js**:

This module defines the main application logic, including the routes for handling incoming requests and the functions for creating and managing chat rooms.

3. **routes/**:

This directory contains modules that define the various routes for the application. For example, the **index.js** module handles requests to the root URL, while the **chat.js** module handles requests related to chat rooms.

4. **public/**:

This directory contains static files that are served to clients, such as CSS stylesheets and client-side JavaScript files.

5. **views/**:

This directory contains the templates used to render the HTML pages that are served to clients.

6. **models/**:

This directory contains modules that define the data models used by the application, such as the **User** and **Room** models.

7. **utils/**:

This directory contains various utility functions used by the application, such as functions for generating random room names.

8. **config/**:

This directory contains configuration files used by the application, such as the database connection settings.

b. Distribution of work to each team member.

FRONT END:

1) WELCOME SCREEN
2) JOURNAL (NOTES)
3) MOOD GRAPH
4) RESOURCES PAGE
5) NOTIFICATION

FRONTEND:

KUNJ: HAVE DONE 2 COMPONENTS IN FRONT END OF THE APP:

1) **WELCOME SCREEN:**

WHICH IS DISPLAYED AND CHANGES BY CLICKING ON ARROW THIS WILL GUILD THE NEW USER ON HOW TO USE THE APP AND WHAT ARE THE DIFFERENT FUNCTIONALITY THAT THE USERS CAN USE. LIKE : JOURNAL, MOOD GRAPH ,RESOURCES.

2) **JOURNAL** : THIS IS A NOTE TAKING OPTION.

MEET: HAVE DONE 2 COMPONENTS IN FRONT END OF THE APP:

3) **MOOD GRAPH:**

MOOD GRAPH WHICH IS DISPLAYED AND CHANGES ON MULTIPLE INPUT AND SHOWS YOU YOUR PROGRESS FOR PAST 1 WEEK (7 DAYS).

4) **RESOURCES PAGE**:

RESOURCES ARE THE ONES WHICH ARE AVAILABLE FOR THE USER ACCORDING TO THE INPUT OF THE USER. YOU CAN SEE THIS BY CLICKING ON THE MAGNIFICATION ICON IN BOTTOM BAR.

KARAN: HAVE DONE 1 COMPONENT IN FRONT END OF THE APP :

5) **NOTIFICATION:**

IF WILL DISPLAY AN ALERT BOX WILL ASK YOU THAT IF YOU WANT TO ALLOW THE NOTIFICATION TO BE RECEIVED ON ON YOUR MOBILE PHONE.

BACKEND INTEGRATION: BACKEND INTEGRATION WAS DONE WITH EQUAL AMOUNTS OF INPUT FROM EACH MEMBER, WE LEARNT THE NEW COMPONENTS AND TOOLS WITH THE USE OF KNOWLEDGE ON THE INTERNET, OUR CLASS NOTES AND OTHER RESOURCES.

c. Things tackled/learned in the team (write about your project journey).

We learnt many new things in our project making, we started off with the idea of bringing an applicative purpose of the app where there is a real life use of the application with necessity functionalities that we provided, till the time we had learned different components, layouts and the necessary java objects and methods that we used to implement several functionalities and with the help of it and the new things that we learned with the reference of lab documents, various websites and other sources.

some of the new things that we learned are.:

IntroductionActivity , Integration of database in the app for mood detection, ResourceDao , Resource Fragment class , Statistical Fragments , RegisterNotificationChannel function , setupResourcesDatabase function , Calling Alarm services and notification in the mobile phone , Preference. xml file , ProgressBar ,etc.

Testing and debugging is the phase where we were Developing a complex application like TALK requires thorough testing and debugging to ensure that it works as expected. As a team we have learned how to write unit tests and integration tests to catch bugs and ensure that new features don't break existing functionality.

Overall, developing TALK would has been a good learning experience for us, we group members worked equally with the help of resources and documents already provided to us in our subject, and we thrive to learn and proceed to do more projects which are applicative based where we use our knowledge to develop the same.