

# **MENTAL HEALTH TRACKER**

## **PROJECT SYNOPSIS**

OF MAJOR PROJECT

### **BACHELOR OF TECHNOLOGY**

Branch: Computer Science & Technology

SUBMITTED BY:

SAHIL GUPTA 2000290100124

NIMISH VASHIST 2000290100093

PRASHANT KUMAR PAL 2000290100102

August 2022



**KIET Group of Institutions, Delhi-NCR,  
Ghaziabad (UP)  
Department of Computer Science and  
Engineering**

### **Table of contents**

<b><u>Content</u></b>	<b><u>Page no</u></b>
INTRODUCTION	3
OBJECTIVES	4
LITERATURE REVIEW	5
METHODOLOGY	10
REQUIREMENT	12
EXPECTED OUTCOMES	13
REFERENCES	14

## **INTRODUCTION:**

Mental health is an important issue in the world today. With a large population now working from home and staying away from loved ones, the mental health situation has deteriorated. As such, it becomes important to track and remedy any problems before they get too serious. We try achieving this using the Companion App. Keeping in mind that users might be suffering from mental illness and wouldn't want to engage much with an app, you'll have to design the app to be very friendly and welcoming. By the end of this project, you'll have a beautiful and fast app that is fun to use and also serves your goal. Try implementing the best practices while building the app; the following sections will detail the implementation goals and suggest some ways to achieve them. This project involves building a simple app in Flutter that tracks the mental health of its users and tries to help them get through their condition by suggesting tasks and keeping record of their progress. Beginners in Flutter will find this project challenging to complete, intermediate folks might learn new things about accessibility, design practices and fine-tuning the app for the audience in question and will have fun building the app. Folks who are at an advanced stage of development might find this project a decent addition to their portfolio.

## **OBJECTIVES**

The project focuses on building a mental health tracker. You will try to get an idea of the mental state of your user (in the least intrusive ways), find out if they are suffering and then suggest measures, they can take to get out of their present condition.

A user answers some questions and based on the answers that they provide, you will suggest tasks to them and maintain a record of their mental state for displaying on a dashboard

# **LITERATURE REVIEW**

## **Research Paper 1**

Title: Scaling Up Machine Learning: Introduction

Author: Ron Beckerman

- Distributed and parallel processing of very large datasets has been employed for decades in specialized, high-budget settings, such as financial and petroleum industry applications.

- The current rise in interest in scaling up machine learning applications can be partially attributed to the evolution of hardware architectures and programming frameworks that make it easy to exploit the types of parallelism realizable in many learning algorithms.

- Several platforms make it convenient to implement concurrent processing of data instances or their features. This allows straight forward parallelization of many learning algorithms that view input as an unordered batch of examples and aggregate isolated computations over each of them.

- Increased attention to large-scale machine learning is also due to the spread of very large datasets across many modern applications. Such datasets are often accumulated on distributed storage platforms, motivating the development of learning algorithms that can be distributed appropriately.

Finally, the proliferation of sensing devices that perform real-time inference based on high-dimensional, complex feature representations drives additional demand for utilizing parallelism in learning centric applications.

## Research Paper 2

Title: Data Structures

Author: Jean-Daniel Bissonnet

Data structures are the keystone on which all algorithmic techniques rely. The definition of basic yet high-level data structures, with precise features and a well-studied implementation, allows the designer of an algorithm to concentrate on the core issues of the problem

Throughout this book, we describe data structures especially designed for representing geometric objects and dealing with them. But computational geometers also make extensive use of data structures that represent subsets or sequences of objects.

These structures can be used directly by the algorithms or modified and augmented for geometric use. The first part of this chapter recalls the terminology and features of each basic data structure used in this book. It is useful to know how these structures can be implemented and what their performances are.

The most delicate problem is undoubtedly the one addressed by dictionaries and priority queues, which treat finite subsets of a totally ordered set (the universe). To achieve better efficiency, these structures are usually encoded as balanced binary trees.

Finally, when the universe is finite, dictionaries and priority queues can be even more efficiently implemented by other more sophisticated techniques, the characteristics of which are given without proof in the third part of this chapter.

## Research Paper 3

Title: Introduction to Python

Author: Jaan Kiusalas

This chapter is not a comprehensive manual of Python. Its sole aim is to provide sufficient information to give you a good start if you are unfamiliar with Python

Python is an object-oriented language that was developed in the late 1980s as a scripting language (the name is derived from the British television show Monty Python's Flying Circus). Although Python is not as well known in engineering circles as some other languages, it has a considerable following in the programming community

In fact, Python is used by more programmers than Fortran. Python may be viewed as an emerging language because it is still being developed and refined. In the current state, it is an excellent language for developing engineering applications - Python's facilities for numerical computation are as good as those of Fortran MATLAB.

Python programs are not compiled into machine code but are run by an interpreter. The great advantage of an interpreted language is that programs can be tested and debugged quickly, allowing the user to concentrate more on the principles behind the program and less on programming itself.

Because there is no need to compile, link, and execute after each correction, Python programs can be developed in a much shorter time than equivalent Fortran or C programs.

## **Research Paper 4**

Title: Python- A Comprehensive yet Free Programming Language for statistician

Author: Xitij U. Shukla & Dinesh J.Parmar

Due to availability of tools for efficient numerical and statistical analysis and visualization of voluminous data with least programming syntaxes.

Python being a modular programming language is accepted by academic and scientific community on account of its clean and readable coding style. Code indented with whitespace for function definition, loop, and control structures rather than curly braces of C, C++, or Java.

Python is a platform independent programming language available for windows and Linux. It can run with 500MHz processor and 256 MB RAM. The manual installation may distract the researcher.

Packages such as Pandas and Seaborn compensate further needs of advanced data analysis and visualization respectively for financial and statistical computations.



## **Research Paper 5**

Title- Data Science: Fundamental Principles

Author: - Alisa Bilal Zorić

With the enormous increase in data, there is a constant need for analyzing such a large amount of data. Data Science can manage this data and develop beneficial machine learning models that predict future results.

We can conclude that Data Science is emerging multidisciplinary field with roots in mathematics, statistics, and computer science.

The main goal of Data Scientists is to recognize and use meaningful insights from data to help organizations in taking smarter decisions.

During that process, they use different tools and methods to identify redundant patterns and hidden knowledge within the data. They also use the most powerful hardware, most efficient algorithms, and programming systems to solve the data related problems.

## **METHODOLOGY**

### **Start building your screens**

We will set up the basic skeleton of the app first. Use your creativity to build an intuitive app and design the elements in such a way that they are appealing to people of all backgrounds. Make sure the workflow is easy to understand and follow. Ditch any design choice that would require the user to take a complex route to achieving a goal that can be done in a simpler way. If you are building a Flutter app for the first time, refer this page. I recommend installing VSCode and the Flutter and Dart plugins to start developing apps.

### **Questions Screen and UI refinement**

This task focuses on making the app friendly to its target audience. You have to remember that potential users of this app would be suffering from mental illness and as such it won't be wrong to assume that asking a lot of questions won't be the best choice. In addition, the mechanism adopted for asking said questions need to be the least intrusive one you can think of. That way, we get the required information at minimum inconvenience to the end-user. Another detail worth mentioning is the double-tap to disappear feature. Once the user is done with a task, the user can get rid of it from the screen using just a double-tap. Easy, intuitive and satisfying - just the way we want it! An important thing to note while building the screens and using the variables that power your app, is state management.

### **Write a simple algorithm**

Once you have all the answers to the questions, you will need to determine whether the person needs help or not. In other words, whether you need to suggest some tasks for them to feel better. How you achieve this algorithm is largely up to you; the complexity can range from a simple comparative check to more advanced classification techniques.

### **Set up your Firebase project**

We need a backend for our application to provide services like authentication, cloud storage etc. Firebase is one of the easiest ways to get started with building a backend. You can learn more about Firebase from the links provided in the References section.

### **Add Google sign-in feature**

We are using Firebase for providing authentication services like Google sign-in. Much of the workflow is very easy to follow and build, and a little research online can help you get started.

**Connect local variables to Firebase storage.** Local variables like the answers to the questions, scores and suggestions for the users should be tied to the current user account. This can be done by creating cloudstore documents for the answers. You should be tracking parameters like when the last answer was provided (recall that you have to show questions only once in a day), which tasks have been completed by the user and a record of the mental states and tasks completed over a period of time (say, a week) to show on the Dashboard. All files that have variables which need to be persisted or to be made available across the app should have code implemented to connect these variables to Cloud Firestore. Developers use the Firebase SDKs for Cloud Storage to upload and download files directly from clients. Cloud Storage stores your files in a Google Cloud Storage bucket, making them accessible through both Firebase and Google Cloud.

### **Build for release**

Once you have run the app in debug mode on your phone or emulator and you feel that it is ready to be deployed, we can go ahead with deployment. An optional step before building your app for release is obfuscating your code. Obfuscation hides function and class names in your compiled Dart code, making it difficult for an attacker to reverse engineer your proprietary app. Your application backend is already deployed on Firebase.

## **REQUIREMENTS**

- Research about good design practices. Learn how you can make your app more accessible and user friendly.
- Create a full-screen experience for the Questions UI. This is done to avoid any distractions and make the user focus on just the question on the screen.
- Find out the different ways in which you can predict an outcome from a set of known variables (here, the user answers).
- Research about Firebase and the features it offers.
- Explore Cloud Firestore in Firebase. Find out how it's different from a Realtime Database.
- Design a structure for the documents to be stored on the database.
- Make sure to link the data with the current user.
- Research about using the Firebase library in Flutter to handle Cloud firestore transactions.

## **EXPECTED OUTCOME**

- to return relevant information or perform specific functions as requested by the user
- helps to predict accuracy of the various datasets
- able to track the mental health of individual on basis of data provided

## **REFERENCES**

- [1] Scaling up Machine Learning Algorithms for Large Datasets
- [2] Data Structures and Algorithms in Pen-based Computing Environments
- [3] Introduction to Python
- [4] Python-A comprehensive yet free programming language for Statisticians
- [5] Fundamentals of Data Science for Future Data Scientist
- [6] Got idea from [BOB Microsite Online Contest at Techgig.com](#)
- [7] Used Wikipedia for relevant information regarding machine learning algorithms and voice assistant
- [8] Used information from blogs of Analytics Vidhya