iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** V **Month of publication:** May 2021

DOI: https://doi.org/10.22214/ijraset.2021.34712

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

A Research Paper on a Progress Tracking Application using Flutter and Firebase

Parth Jindal[1], Piyush Sharma[2], Mohit Kumar[3]
[1, 2, 3]*B. Tech Students, Department of Computer Science and Engineering, Global Institute of Technology, Jaipur, Rajasthan, India*

Abstract: ***Task Management applications are readily used by many individuals; however, various clients utilize the applications distinctively relying upon their specific necessities. To recognize the issues faced by clients of task applications, I directed exploration to build up a comprehension of the market and clients' necessities. Web-based media and other easily available online interruptions make it difficult for us to remain fixed on our tasks and make it hard for us to tackle our job proficiently. Additionally, continually switching between tasks may give us the false inclination that we are being gainful when we are, indeed, not. It's more significant for us to focus on assignments and work on those that are generally significant, instead of zeroing in on erasing little things from our daily agenda only to feel good.***
The objective of this application is to assist us with getting mindful of how we invest energy and time doing those tasks and how profitable that time is. It can help in improving our productivity by giving is a descriptive, pictorial view of our daily timeline and how to improve it. When we have a description of the assessed time we'll have to spend on our undertakings, alongside the approved time spent on the things for reference or individual/group reviews, we can deal with our day-by-day schedules all the more proficiently.
Keywords: Flutter 2.0, Firebase, Dart, Android Studio, Application Development, Cross-Platform Development

I. INTRODUCTION

The world is in chaos and we are confined to our homes because of the ongoing pandemic and all of the working industry and students are confined to their homes. The classes have shifted to the online mode of teaching and the working classes are either out of work or are working from home. Considering current times and situations we carried out several interviews among students and people working tech. and business jobs, and found out that most people find it difficult to manage the sudden paradigm change as they suddenly find themselves in a situation in which they have no monitoring system set in place to maintain or even measure their productivity. The excess amount of free time due to the lockdown situation has made people less productive than they ever have been before. We have reached the conclusion that we are in need of a tool which can show us our daily expenditure of time and how much of it was expended judiciously. There are a few existing applications that address the issue of productivity but offer nothing in terms of analysis or means to make the day more productive. Hence, the main objective of this application is to provide the user with a modern interface with all the latest tools of productivity such as Pomodoro timers and Baroque music libraries which have been shown to increase concentration and aid in productivity. Most importantly the application will map the daily tasks of the user and provide the analysis of how the day was spent thus helping the user to better understand where they must improve. The application in question will be developed in Flutter, a cross-platform UI framework developed by Google and will be released on various platforms such as Android, iOS, and Web. The application will be an ongoing process as it leaves much room for future additions as the situation changes. As the development technology is a UI Framework, the UX and UI components of the application will be especially refined due to the nature of the technology in use and this is precisely one of the major reasons for choosing this technology as in the modern market, the aesthetics of the application has taken a slight edge over even the performance of the application.

II. CONCEPTS (THEORY)

A. Flutter

Flutter is a simple but high-performance UI framework based on the Dart language developed by Google, which reduces the time overhead of the development cycle of an application by rendering the UI directly into the OS's canvas rather than through a native framework. It offers a large and well-maintained library of UI components which not only make the development process extremely fast as compared to other frameworks but also produces high fidelity applications which not only look beautiful but perform very well. Every UI component in Flutter is called a Widget and each Widget is optimized for the mobile environment and designing an application using these widgets is as simple as arranging HTML elements.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

Flutter being a cross-platform framework can produce applications for various frameworks using a single codebase, these platforms include Android, iOS, Linux, Mac, Windows, Google Fuchsia, and the web. Flutter uses the Material.dart package which contains all the designs for the UI components we know and love from the Android family.With the release of Flutter 2.0, it now also provides features like NULL safety which was very rare in the mobile application development environment. The most amazing feature of Flutter is the hot reload and hot restart options which when run after changes in the code, allows the app to show immediate changes in mere seconds as opposed to other native development tools which take up most of the development time in rerunning the application after the changes are made.

B. Dart

Dart is a programming language developed by Google and it is client-optimized and is used for building applications that can run on multiple platforms. It can be used to build desktop, server, web, or mobile applications. Dart has a C-style syntax and is a garbage-collected, object-oriented, class-based language. It is developed and maintained by Google and has enormous official and community support as it is the primary language for Flutter development. With the newest version of Dart, rare but very useful features like sound NULL safety are also provided which bolsters the overall ability of the Dart programming language.

C. Firebase

Firebase is viewed as a web application platform. It assists developers with building high-fidelity and high-quality applications. It stores the data JSON format which doesn't utilize a query for inserting, updating, deleting, or adding data to it. It is the backend of a framework that is utilized as a database for putting away data. Firebase is a platform created by Google for making portable and web applications. Firebase Storage was intended for application developers who need to store and serve client-produced content, for instance, photographs or some other record. It gives secure archive shares and downloads for Firebase applications, paying little heed to network quality.

D. Android Studio

Built on JetBrains's IntelliJ IDEA integrated development environment software and designed purposely for Android development, Android Studio is the official IDE for the Android operating system. It is a fast, powerful, and feature-rich IDE with a fast Emulator which is very helpful in installing and running Android applications at a remarkably higher speed and testing the application under various Android configurations. Due to the proprietary nature of the iOS environment and various security differences as compared to Windows the IDE of choice for MacBook i.e., iOS testing and development is handled by XCode.

E. Widgets

In Android, the View becomes the basis for every component that is rendered and displayed on the screen. Toolbars, Buttons, and Inputs, everything is a View. In Flutter, View is a Widget. To start, widgets have alternate futures: they are perpetual and simply exist until they ought to be changed. At whatever widgets and their case change, Flutter's structure makes another plant of widget event. Widgets are lightweight, partly as a result of their immutability. Since they are not seeing themselves and are not clearly drawing anybody, however rather are a portrayal of the UI and it is semantics that brings "expanded" into certifiable view inquiries in the engine. Widgets are the major structure squares of a Flutter application UI. Each widget is an invariable attestation of a part of the UI. Unlike various structures or nearby stage contraptions that alternate points of view, see regulators, designs, and various properties, Flutter has a consistent, united article model. Widgets are the segments that affect and particularly the view, feeling of usage, and its appearance.

III.CASE STUDY AND SURVEY

A. Problem Statement

Web-based media and other effectively available online interruptions make it difficult for us to remain fixed on our undertakings and make it hard for us to manage our job productively. Likewise, continually exchanging between undertakings may give us the bogus inclination that we are being beneficial when we are, indeed, not. It's more significant for us to focus on undertakings and work on those that are generally significant, as opposed to zeroing in on erasing little things from our daily agenda only to look good. The objective of this application is to assist us with getting mindful of how we invest energy during the time spent doing those undertakings and how gainful that time is. It can help set a few limitations via web-based media to decrease interruption and track the time we spend chipping away at the to-do things. At the point when we have a superior feeling of the assessed time we'll have to spend on our assignments, alongside the approved time spent on the things for reference or individual/group reviews, we can deal with our day-by-day schedules all the more proficiently.

©IJRASET: All Rights are Reserved

B. Interviews and Survey

We were Hoping to utilize interviews to get familiar with task application clients who either work full-time or as freelancers. We picked these classifications on the grounds that these clients are well on the way to work in a climate that requires productive time assignments and in which a task manager application could be generally advantageous.

1) Interview Questions

a) How would you deal with your tasks consistently and how you do focus on them?
b) What is your experience with utilizing a task management application and why you pick them?
c) How many assignments would you say you are able to complete in a day? How does that influence your daily habits and routine?
d) What do you do when you have incomplete assignments, and why they haven't been completed?
e) How would you assign your time to the tasks you are on?
f) What is the data that you record on your to-do list, and how does this help you in your tasks?
g) How would you assess you have done the tasks efficiently?

From our interviews and study, we had the option to separate subjects among clients and recognize where we think the major issue lies. The gathered information assisted us with framing a client persona whose characteristics line up with the crowd we are focusing on, and I alluded to this persona through our product design process.

We found the following common queries and traits among the clients interviewed and surveyed:

- They appreciate the application being easily set up and get up and running quickly.
- The functionality of the application should be quick and hassle-free.
- They focus too much on a single task and thus lose track of time, therefore, sacrificing efficiency.
- The biggest problem with the currently available applications is that they are either too simple and do not provide all the tools in one place or are too complex, even to the point that they become a minor distraction and a hassle to use.

It is quite tempting for us to add a number of different tools and functionalities in our attempt to provide for the user but as pointed out by the survey, the applications which did this, became too cluttered and complex to use and thus failed to achieve their primary goal. We have adopted the MoSCoW method for the scope planning for this project which will help us in trimming down the scope for the preparation of an MVP (minimum viable product). Our main goal at this point is to release an alpha version of the application which will be made available to the user for free and will be used to gather data as feedback for the further iterations of the product.

Fig. 1 MoSCoW Methodology

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

C. High-Level Application Planning

The below maps convey a high-level action plan for the application.

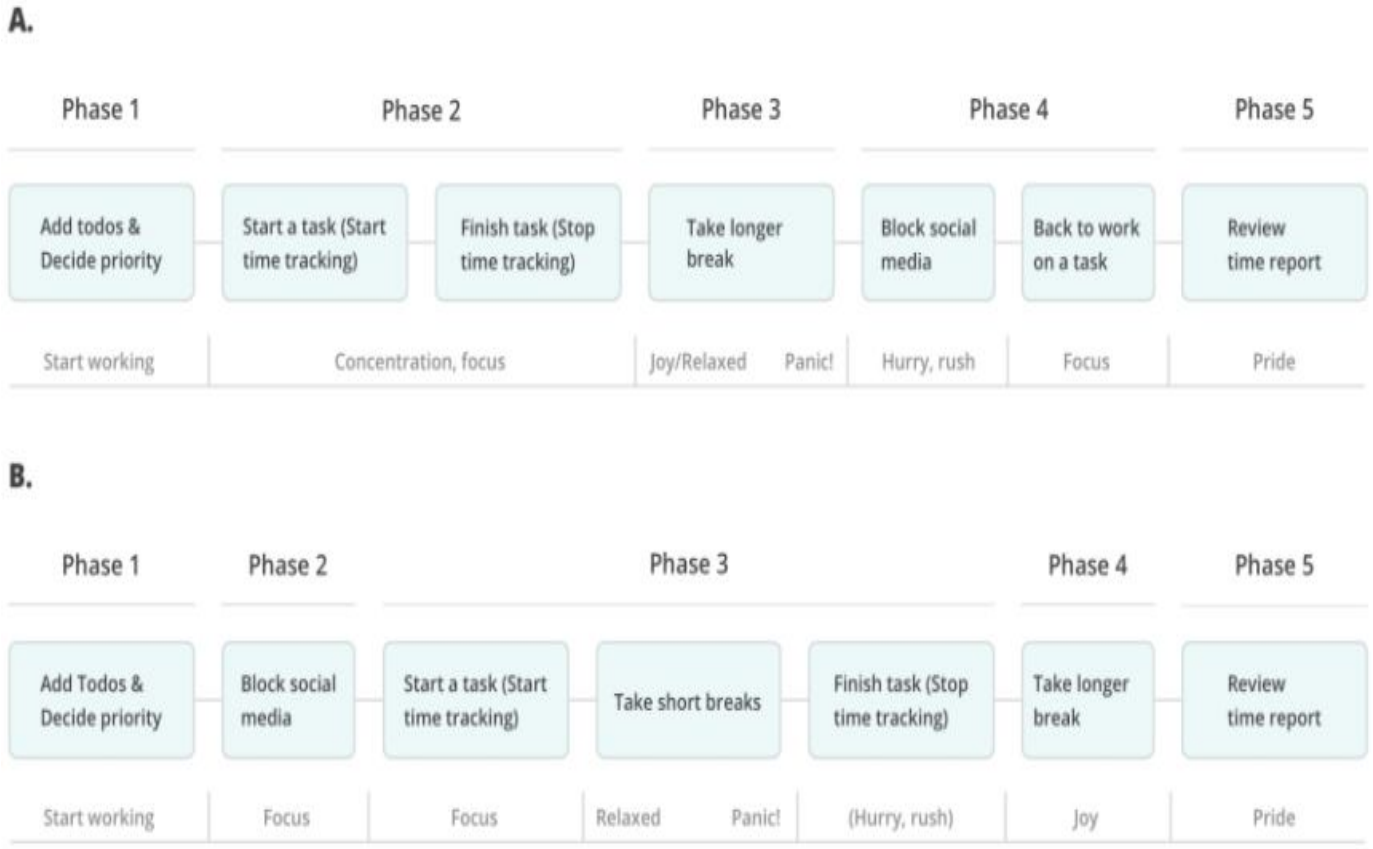


Fig. 2 Phase Map

Fig. 3 Action Map

The site map is a high-level flow structure that depicts all the options the user will have while using the product, we have purposefully taken measures to not give the user too many tools and options as our survey showed that an excess of functionality has a counterproductive effect on the efficiency of the user.

©IJRASET: All Rights are Reserved

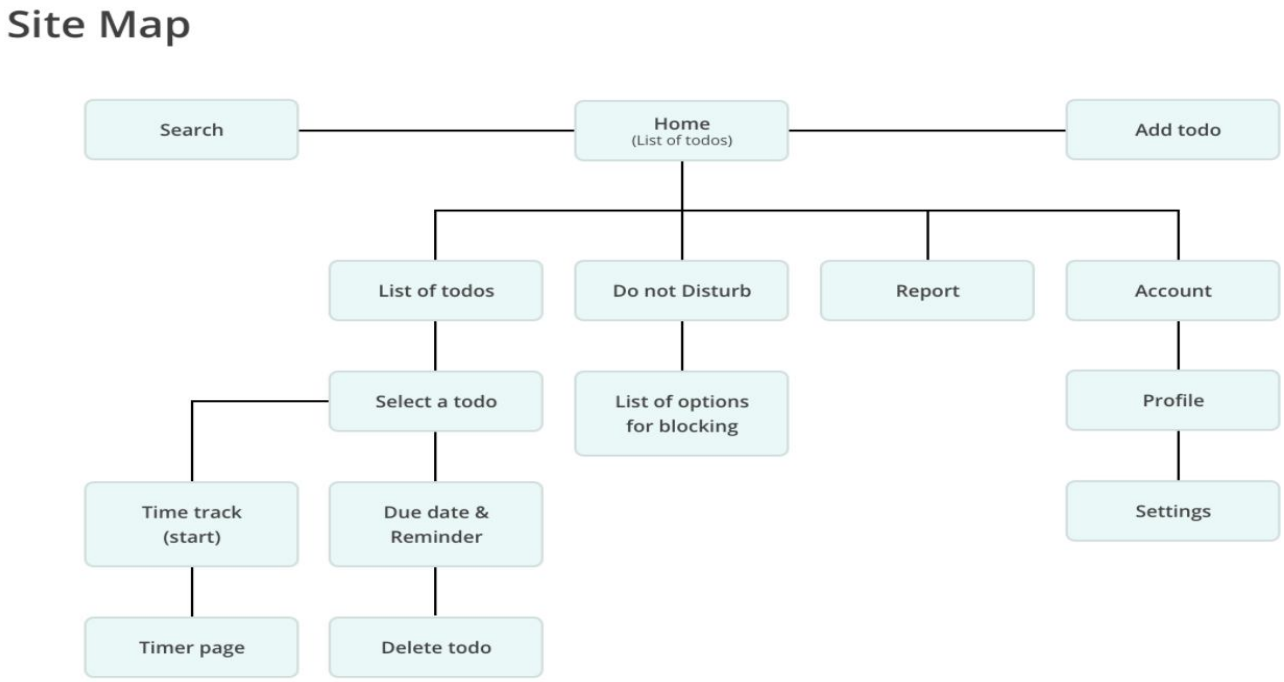


Fig. 4 Site Map

Apart from the core functionality aspect of the application, there are several additional development aspects which are common in most application such as UI design, Registration/Login functionality, personalized data storage, payment options to name a few, all these will be implemented using Flutter and Firebase utilizing the tools provided by the frameworks to speed up the process and promise security that these frameworks innately provide. In the future versions of the application, we are planning on including features that are fairly uncommon and are gaining popularity such as accepting payments via cryptocurrencies such as Bitcoin and Ethereum, and options to design custom routines and timer patterns, and playlists which the creator can sell as NFTs (non-fungible tokens).

IV. CONCLUSIONS

As is clear from the above text, we are very eager to implement this project concept using Flutter 2.0 and Firebase. As this app was planned to be released on both Android and iOS as well as on the Web, our original inclination was to develop the app using Flutter for the mobile platforms and React for a web application but with the formal release of Flutter 2.0 which now officially provides support for Flutter Web and features like sound null safety we have decided to undertake this project with Flutter for all platform use-cases. Therefore, in conclusion, we will be developing and releasing a new and fresh take on progress tracking apps with our version of the idea which will be implemented completely in Flutter and use Firebase for a database along with REST APIs using Python.

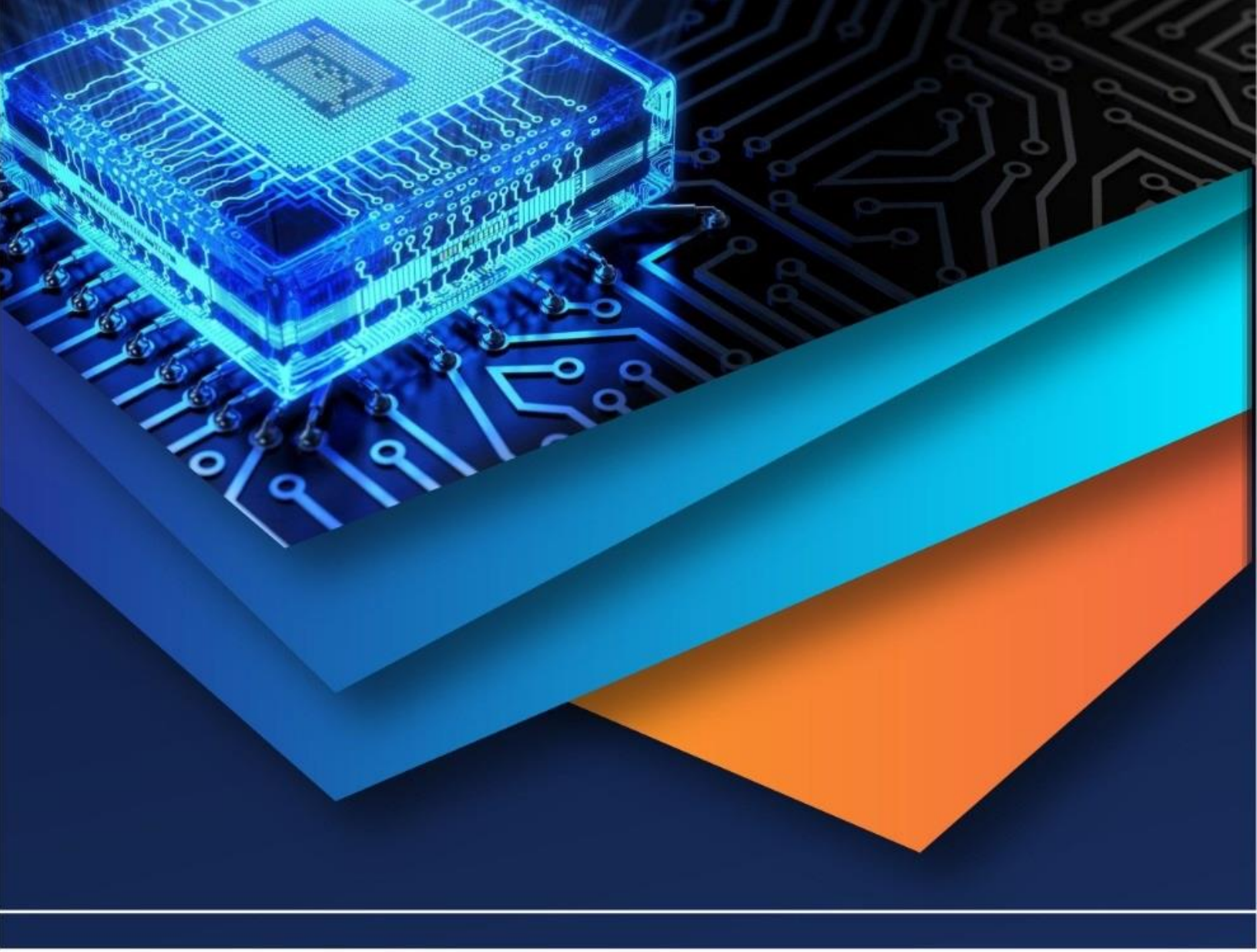
V. ACKNOWLEDGMENT

It brings us great pleasure in writing this paper. Firstly, we would like to extend our thanks to our guide “Ms. Pooja Sharma” who provided us with her experience and expertise, and informative insights and critiques when we were in need of them. We would also like to thank Mr. S. S. Shekhawat for allowing us to move forward with this idea for our project. Also, a great and well-deserving thanks to Google for creating and maintaining these high-end, production-level technologies which allowed us to make this project a reality in the most innovative and efficient way possible.

REFERENCES

[1] https://flutter.dev/
[2] https://dart.dev/
[3] https://flutter.dev/docs/resources/
[4] https://firebase.google.com/

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)