ISSN No:-2456-2165

Cloud Based Student Career+ App

Dr. Murugavalli S, Head of the Department, Department of CSE, Panimalar Engineering College
Senthil Kumar G, Associate Professor, Department of CSE, Panimalar Engineering College
Santhosh T, U.G Scholar, Department of CSE, Panimalar Engineering College
Suresh Kannan K, U.G Scholar, Department of CSE, Panimalar Engineering College
Pradeep P, U.G Scholar, Department of CSE, Panimalar Engineering College

Abstract:- Due to the pandemic situation students were completely involved with or were dependent on technology and to communicate, learn and even take exams. Technologies will play a major role in every field in the future. But one thing that does not change is how we study even after the pandemic comes to an end. We will roll back to the traditional way of studying by attending classes physically. Nonetheless we also will be partially depending on technologies to study when we are at home. To improve the experience of how a student can study, we develop a mobile application that acts as a median between the staff and the student. If a simple notification can remind us of several aspects, we can improvise our study experience by a simple notification that arrives directly from the staff to a student mobile mainly focussing on students with backlogs who can manage their study process in a much easier way.

***Keywords**:- On duty (OD), Firebase Cloud Messaging (FCM).*

I. INTRODUCTION

Technology has become a part of our lives as it plays a vital role in enhancing our workflow. From setting reminders to connecting with people from different places, technology could also be used for the betterment of a student's career. Every student today owns their personal smartphone. To improve the communication between students and staff, a mobile application could act as a medium. By implementing cloud functionalities over a mobile application, both students and staff could communicate professionally anywhere anytime. For students with backlogs, this application could help them manage their time by receiving messages every day from the staff who taught them the subject. There are more functionalities added, that would benefit the overall career of a student.

II. EXISTING SYSTEM

Each college could have its and own database of managing student details. Each student can log in with their credentials and see their career progress over there. But when a student moves to next semester with a backlog, he or she will not be getting any advice from the staff for the respective backlog subject on how to manage time and study for clearing the backlog. Time management is what matters for a student where they must have to focus on both their current progress as well as their backlogs. With the help of staff, they can achieve their needs by getting advice on planning their study periods. But, there occurs a situation where one staff cannot handle a large set of students. A staff might teach the same subject for different semesters of different years. One student from a particular semester might have to approach the staff even after a year or two. In such cases the time management for staff becomes hard and they might end up spending extra time after their work hours.

III. PROPOSED SYSTEM

Developing a mobile application that and would enhance the study experience of a student, especially a student with backlogs would find it very easy to study and organize his or her time. Students will gain confidence once the right advice on organizing their study process becomes handy. Like students, teachers will also be able to provide tasks to backlog students with ease. This application has several features such as notify students with backlogs for a particular department or for a set of classes. Students will get notified instantly as the backend functionality is implemented in the cloud. There is a separate login for students and staff. Students could view their backlogs, view notes provided by respective staff, request for an OD on a particular date, and view messages sent by staff in general or as backlog notification. Similarly, staff can log in with their credentials. staff would be able to manage notes, send notifications either generally or specifically to backlog students of certain classes, manage OD requests, modify student backlogs and search for student details to view his progress.

IV. SYSTEM ARCHITECTURE

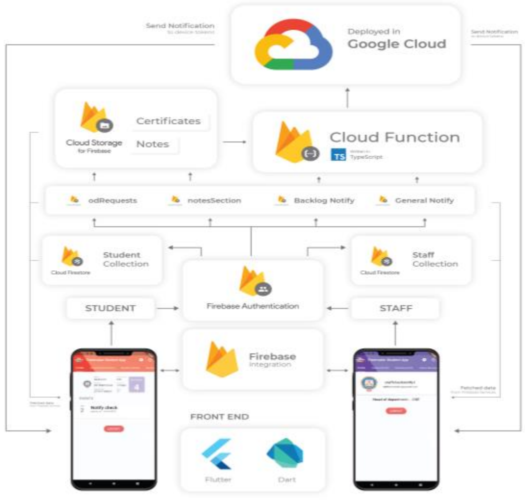

Fig 1: Architecture of the application

All details given by the users will be stored in the firebase firestore in documents and collections format. Any modification that occurs in a collection triggers a function implemented in google cloud to send notifications to all the registered device tokens. This happens instantly and the application is capable of retrieving data from firebase services with ease.

V. FEATURES OF THIS APP

The app consists of two main modules, one for staff and the other for students. Each module is meant to interact with the database and display data in different formats.

Register user

A register module is used to create users with email and password along with their details inside the firebase firestore. Firebase authentication instances and storage instances are used to perform such tasks. After successful creation users will be asked to link their Google account for password recovery purposes. As login is done using Firebase Authentication Services with user-defined email and password, a student or staff can include their domain under the email section.

Login user

A login module is used to log in users with email and password. Firebase authentication instances are used to perform such tasks. After successful login users will be navigated to their respective profile screen.

Storing device tokens

After successful login of users, the device token will be stored in the firestore inside the user document. Using shared preferences the login credentials are stored to make the user stay logged in until the user logs out.

Staff login Features

A. VIEW PROFILE

A staff member who logs in will be able to see their mail id, name, department, and designation under this tab. They will be given an option to unlink their google account and log out from the app on their device.

B. GENERAL NOTIFY

By choosing the department, semester number, and respective classes under the department, staff can send general messages to the selected classes which are stored as separate documents for each message under firestore collection.

C. BACKLOG NOTIFY

By choosing the department, semester number, subject code, and respective classes for which the logged-in staff has taken a class, they can send messages to students with backlogs in chosen subjects which are stored as separate documents for each message under firestore collection.

D. NOTES MANAGEMENT

Staff can upload their own set of notes for selective departments and classes which is stored under documents named after the email id of logged-in staff. Firebase storage instances are used to store data in both firestore and firebase storage. They will be able to upload pdf and view the files they have uploaded as a list of cards.

E. VIEW STUDENT

Staff can search for a student by choosing a department and class id. A list of roll numbers will be fetched from the chosen class. Staff can search for one or more students using their roll number. Student details will be displayed in the form of cards. Student's backlog details could be modified by staff under this tab.

F. PROVIDE OD

OD requests from students are displayed under this tab for the chosen department, semester and classes. A switch is provided to view either all requests or requests waiting for approval. By checking request details, OD can be approved by the staff.

Student login Features

A. VIEW PROFILE

A student who logs in will be able to see every detail they have given during registration. Under events, there will be cards containing details on their approved OD requests. They will be given an option to unlink their google account and log out from the app on their device.

B. GENERAL NOTIFICATIONS

If a staff sends general notifications to a set of classes that matches with the logged-in student's class id, a card will be displayed with the message and sender details including date and time.

C. BACKLOG DETAILS

If the logged-in student has backlogs in certain subjects, it will be displayed as a list of cards along with an option to view notes of that subject.

D. BACKLOG NOTIFICATIONS

If a staff sends backlog notifications to a set of classes for a particular subject that matches with the logged-in student's backlog subjects, a card will be displayed with the message and sender details including subject code. A student can view the respective subject's notes using the "view notes" option.

E. VIEW NOTES

By choosing respective staff mail, class id, and subject, a student will be able to view notes that are uploaded by the chosen staff matching with class id.

F. REQUEST OD

A student can request OD by uploading the photocopy of the proof along with the date, title of the event, and location. These requests are stored as documents under firestore and are linked with firebase storage using the path of the uploaded file.

G. VIEW REQUESTS

After a successful request for OD, a list of cards containing the information provided by the student is displayed. Approval status is shown under each card. If approval is pending, cancellation of requested OD is included. If a request is approved, it gets added under the student's event details.

VI. CLOUD FUNCTIONS

As notifications play a vital role in the application's functionality. Any message that is sent from staff to students or any request that is made by a student needs to be known by the user instantly. To make this possible over the cloud, a cloud function integrated with Firebase Cloud Functions is written and deployed as a backend. There are major three functions that are responsible for notifying the user online on a device. Admin privilege is required to deploy cloud functions.

The functions are written in such a way that whenever a change occurs in a collection under firestore, the function is triggered.

A method named 'sendToDevice()' under FCM is used to send data to devices by passing a list of tokens as arguments.

A. GENERAL NOTIFY

When a staff sends a general notification to a set of classes under a department, the function adds the device token of students to a list which is then passed to an FCM instance. The device tokens are pushed into the list based on certain conditions.

B. BACKLOG NOTIFY

If backlog notification data is added into the collection, the function gets triggered and works similar to the general notify function but with different conditions.

C. OD REQUEST

When a student requests for OD, the cloud function looks out for tokens of staff devices and pushes them to a list based on conditions. The FCM takes care of sending data to those devices in the format of notifications.

VII. LINKING GOOGLE ACCOUNT

To ensure the user's safety, after successful registration the user is required to link their Google account with their current login id. In the future, the firebase dev team might include the feature of making a secondary linked account as a recovery mail to reset password or for any other security purposes.

VIII. CONCLUSION

This project focuses mainly on enhancing communication between the staff and students with the use of trending technologies. A simple notification that can be received anywhere anytime could possibly make an impact on a student's career.

REFERENCES

[1]. Main Course: https://www.udemy.com/course/flutter-bootcamp-with-dart/
[2]. GitHub Repository: https://github.com/londonappbrewery/Flutter-Course-Resources

Reference for basics :

[1]. Flutter documentation: https://flutter.dev/docs
[2]. Firebase documentation: https://firebase.google.com/docs/build
[3]. Firestore documentation: https://firebase.flutter.dev/docs/firestore/usage/
[4]. Cloud function documentation: https://fireship.io/lessons/flutter-push-notifications-fcm-guide/

Package references:

[1]. Google Sign In: https://pub.dev/packages/google_sign_in
[2]. Cupertino Icons: https://pub.dev/packages/cupertino_icons
[3]. Theme Patrol: https://pub.dev/packages/theme_patrol
[4]. Sticky Headers: https://pub.dev/packages/sticky_headers
[5]. Smart Select: https://pub.dev/packages/smart_select
[6]. Simple Animations: https://pub.dev/packages/simple_animations
[7]. Firebase Core: https://pub.dev/packages/firebase_core
[8]. Firebase Authentication: https://pub.dev/packages/firebase_auth
[9]. Firebase Storage: https://pub.dev/packages/firebase_storage
[10]. Firebase Messaging: https://pub.dev/packages/firebase_messaging
[11]. Cloud Firestore: https://pub.dev/packages/cloud_firestore
[12]. Modal Progress HUD: https://pub.dev/packages/modal_progress_hud
[13]. Image Picker: https://pub.dev/packages/image_picker
[14]. Permission Handler: https://pub.dev/packages/permission_handler
[15]. Shared Preferences: https://pub.dev/packages/shared_preferences
[16]. Flutter Toast: https://pub.dev/packages/fluttertoast
[17]. File Picker: https://pub.dev/packages/file_picker
[18]. PDF Viewer: https://pub.dev/packages/flutter_plugin_pdf_viewer
[19]. Carousel Slider: https://pub.dev/packages/carousel_slider

Design Reference:

[1]. Material Design Icons: https://www.materialpalette.com/icons