

DIGITAL BULLETIN BOARD

A MINI PROJECT REPORT (15CS72C)

Submitted by

DONALD. P

SAKTHIVELRAJAN. K

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



NATIONAL ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University – Chennai)

K.R. NAGAR, KOVILPATTI – 628 503

December 2021

BONAFIDE CERTIFICATE

Certified that this mini project report “**DIGITAL BULLETIN BOARD**” is the bonafide work of **DONALD P (REGISTER NO: 1812111), SAKTHIVELRAJAN K (REGISTER NO: 1812114)** who carried out the project work under my supervision.

SIGNATURE

Dr.V.GOMATHI M.Tech., Ph.D.,

Head of the Department

Professor and Head

Department of CSE,
National Engineering College

(An Autonomous Institution)

K.R.Nagar, Kovilpatti: 628503.

SIGNATURE

Mr.K.VIVEKRABINSON ME.,

Guide/Supervisor

Assistant Professor

Department of CSE,
National Engineering College

(An Autonomous Institution)

K.R.Nagar, Kovilpatti: 628503.

Submitted to the **MINI PROJECT (15CS72C)** Viva-Voce Examination held at
National Engineering College on

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

First and foremost, we would like to thank God Almighty for showering his blessings throughout our life. He has been the tower of our strength in each step of our work. We take the privilege to express hearty thanks to our parents for their valuable support and effort to complete the project work.

We would like to express our deep sense of gratitude and respectful regards to our director **Dr. S. Shanmugavel B.Sc., D.M.I.T., Ph.D.**, for giving an opportunity to do this work.

We have great pleasure in acknowledging our Principal **Dr. K. Kalidasa Murugavel, M.E., Ph.D.**, for extending his full support to undergo this work.

We express our profound thanks to our beloved Head of the Department **Dr. V. Gomathi., M.Tech., Ph.D.**, for extending her full support and providing various facilities during the project work.

We would like to thank our project guide **Mr K.Vivekrabinson M.E.**, Assistant Professor, Department of Computer Science and Engineering, whose valuable guidance, technical support and suggestions helped us for doing the project work.

We express our gratitude to our project coordinator **Dr. J. Naskath M.E., Ph.D.**, Assistant Professor (Sr. Grade) Department of Computer Science and Engineering for his valuable guidance at each and every stage of the project.

We extend our hearty thanks to our tutors and class in-charges for their valuable guidance. We are grateful to all the staff members and our dear friends for their valuable suggestion and co-operation for this project work.

ABSTRACT

Most of the institutions/universities switch to the online system because of its necessity due to COVID-19, so we need an efficient medium to convey the information. Digital Bulletin Board is an application that provides a platform to access notice/bulletin from anywhere at any time. To access our service, the user should have registered as either student or staff. The corresponding role for the users will be provided by the admin upon successful verification (using OTP). The staff members have the only privileges to post a notice. The students can receive information from their respective departments only. For example, the student studying computer engineering can receive a message from the computer department only. Still, the placement coordinator of each department can share the information with any other department. There will be an admin panel to control the Staff/student's account. This application would be a new way of displaying a notice rather than the conventional method. This android application can be used by staff, students. The proposed application is quick, easy to maintain, and has a rich UI. A well-designed mobile app can perform actions much quicker than a mobile website. Applications usually store their data locally on mobile devices, unlike websites that generally use web servers. Mobile technology will be the core of consumer-brand interaction and offer excess customization, communication, and loyalty opportunities. We used the Android studio and Google firebase platform for this

mobile application. Mobile technology will growingly be the core of the consumer-brand interaction and offers significant changes for customization, communication, and loyalty. In short using this application the staff shares the information or notice to the students, and it can be in text format and the students can only view the information shared by the staff.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1.	INTRODUCTION	1
2.	LITERATURE SURVEY	3
	2.1 LITERATURE REVIEW	3
	2.2 LIMITATIONS OF EXISTING SYSTEM	5
	2.2.1 Order of data	5
	2.2.2 Complexity	5
	2.2.3 Inconsistency of data	5
	2.2.4 Damage	6
	2.2.5 Editing and Communication	6
	2.3 SUMMARY	6
3.	PROPOSED METHODOLOGY	7
	3.1 ARCHITECTURE DIAGRAM	7
	3.2 FRONT END OF THE APPLICATION	8
	3.2.1 Activities	9
	3.3 BACK END OF THE APPLICATION	9
	3.3.1 Services	9
	3.4 ADDING FIREBASE TO THE APPLICATION	9
	3.5 SYSTEM REQUIREMENTS	10
	3.6 MODULE DESCRIPTION	11
	3.6.1 User Module	11
	3.6.2 Admin Module	11
	3.6.2.1 Login Component	11
	3.6.2.2 Batch Details	11
	3.6.2.3 Notice Details	11
	3.7 Database Design	12
4.	RESULT ANALYSIS	13

4.1	RESULT	13
4.2	SCREENSHOTS	13
4.2.1	User Module	13
4.2.2	Admin Module	15
4.2.3	Firebase	18
4.3	SAMPLE CODE	21
4.3.1	XML	21
4.3.1.1	ACTIVITY_MAIN.XML	21
4.3.1.2	ACTIVITY_ADMIN_UPLOAD.XML	23
4.3.2	JAVA	27
4.3.2.1	MAINACTIVITY.JAVA	27
4.3.2.2	ADMIN_UPLOAD.JAVA	31
5.	CONCLUSION	35
6.	EXPECTED OUTCOME PROOF	36
	APPENDICES	37
	REFERENCES	38

LIST OF FIGURES

FIGURE NO.	FIGURE DESCRIPTION	PAGE NO.
3.1	ARCHITECTURE DIAGRAM	7
3.2	DATABASE DESIGN	12
4.1	LOGIN PAGE	13
4.2	NOTICE WINDOW	14
4.3	ADMIN LOGIN PAGE	15
4.4	BATCH DETAILS	16
4.5	NOTICE TEXT AREA	17
4.6	AUTHENTICATION	18
4.7	ADMIN CREDENTIALS	19
4.8	NOTICE DATABASE	20

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
API	Application Programming Interface
XML	Extensible Markup Language
OS	Operating System
GUI	Graphical User Interface
SQL	Structured Query Language
QR	Quick Response
DOS	Disk Operating System
APK	Android Package
UI	User Interface
GB	Giga Byte
IDE	Integrated Development Environment
SDK	Software Development Kit
DBB	Digital Bulletin Board

CHAPTER 1

INTRODUCTION

The importance of mobile phones in our activities and routine life is undeniably never-ending. This is so owing to there is an ongoing massive transformation in that mobile phones are no longer the ordinary communication device they used to be. It has become an extremely large point of attention for individuals and businesses alike, courtesy of the various incredible features and chances that mobile phones offer. The access to high-speed internet and the cumulative progress of mobile technology, the availability and the remarkable communicative interface in these devices result in a whole level of new and innovative experience in mobile computing (mobile apps). This is made possible through the progress of mobile applications and development. Today, the availability of mobile applications is on the larger such that it produces experience computing and a remarkable change in the way humans feel. A few years ago, to access the internet, read and checkout emails one who needs a computer, but nowadays this is different because computing is now carried all places in their cell phones. Imagine buying a bus ticket on the go. This is something our forefathers never thought of it or did. Think not going to the bank but still transferring money to friends and family. All thanks to the top application development companies and application developers. No matter which, they have come to save, enabling their easy life. Various leading application development companies and developers have designed educational apps that have profoundly impacted lives across the globe. These apps are dear to all educationists, faculty, and students. Bright things play an essential role in the real world. Today all things around us are becoming smart.

Mobile development is a process for building mobile applications that run on mobile devices. These applications can either be pre-installed or

downloaded. It is installed by the user later. They use the network capabilities of the device to work computing resources remotely. Hence, the mobile application development process requires creating software installed on the device, enabling backend services for data access through APIs, and testing the application on target devices. To develop scalable mobile apps, you also need to consider screen sizes, hardware requirements, and many other aspects of the app development process.

Each screen or window in an application is known as an activity. Activities are classes that you enlarge from a base activity class. Each activity is bound to an XML layout file which provides the front-end visuals. Activities are created, resumed, and terminated by the Android OS through a series of call back methods called automatically by the OS at the appropriate times. Some examples of call back methods: on-Create, on-Pause, on-Resume, on-Stop. These methods must be overwritten for your activity to progress correctly.

In this 21st century, the mobile app plays a vital role in many fields for satisfying the needs of the people. Our plan is to develop a mobile application for the digital bulletin board for the college/institutions. This application reduces the time compared to manual work. Staff members of the college/institutions can share their notices through this application. This mobile application consists of staff who can post their messages by logging in with their username and password and then redirect them to select the year via buttons. In this staff module, they can share their information in the text format, and there are checkboxes to share the information to the particular classes. In the student's module, the students can view the information posted by the staff. There were some existing systems for the digital bulletin board, but our application provides a simple user interface and is easy for anyone to use. This system works with a database that holds the users' data, Admin, and the information will be send and received with the date and time.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE REVIEW

1.Display Message on Notice Board using GSM - Electronic

This paper deals with an SMS-based notice board incorporating the widely used GSM to facilitate displaying messages on the notice board via the user's mobile phone. Its operation is based on microcontroller ATMEGA32 programmed in assembly language. A SIM300 GSM modem with a SIM card is interfaced to the ports of the microcontroller with the help of commands. When the user sends an SMS via a registered number from his mobile phone, it is received by SIM300 GSM modem at the receiver sends.

Limitations:

- SMS Based system.
- No Scheduling.

2.Electronic Display board Using GSM Technology - Wireless

This paper discussed the design of SMS driven automatic display board which can conventionally notice boards and return the currently used programmable electronic display. It is proposed to design a receive cum display toolkit, which can be programmed and operated from an authorized mobile phone.

Limitations:

- Using LED Board.
- Limit the number of characters.

3.Design and Implementation of Digital Notice Board Using Power Line Communication - Electronic

This paper proposes one such application for automating an educational institution by circulars or replacing manual notice boards with digital notice boards. With frequent updating, a centralized database is highly possible. The system uses existing power lines to transfer the data to a particular node or broadcast it to various nodes. The address is allocated to each receiver, and its response is based on their appropriate commands.

Limitations:

- Need Power Line Communication.
- A limited number of Character.

4.GSM Based Wireless Electronic Notice board

B. Ghazal, et al developed a notice Board. This paper proposes a GSM-based wireless electronic notice board system that can replace the current programmable electronic Display. It is about to write the message on a cell phone and send it as SMS, which will display the news on the LCD.

Limitations:

- Using GSM & LCD

5.Raspberry Pi Wireless Electronic Notice board

V. M. Ibrahim, et al present that notice boards play a significant role in our day-to-day life. Here the admin can control the notice board through the internet. So, information can be sent anywhere in the world and can be displayed within seconds. We can send Information through Mobile and connect the Raspberry Pi to the internet at the receiving side. First, we have to install an application on the admin Mobile.

Limitations:

- Using Raspberry Pi
- Using Electronic Display.

2. 2 LIMITATIONS OF EXISTING SYSTEM

Currently, our college has a manual system of putting notices on notice boards. It's outdated now. Nobody has time to stand in a rush to read the messages on the noticeboard.

2.2.1 Order of data

Notice can get out of charge in the traditional notice board system. If someone accidentally puts some data in the wrong place, it can lead to lost data. An automated Notice management system allows users to quickly check whether information already exists somewhere in the design, which helps avoid problems like redundant data.

2.2.2 Complexity

Automated system is less complex than a manual system of handling notices, making it easier for untrained people to manipulate and access data. Anyone having the basic knowledge of mobiles can work on the automated system.

2.2.3 Inconsistency of data

There will be no access for future use since notice might get misplaced during manual notices management. So notice won't be appropriately preserved for future use.

2.2.4 Damage

Manual notices stacks are vulnerable to damage, destruction, and the ways that digital databases are not. A company may backup its digital data both onsite and at office locations, ensuring its security if the office building suffered a flood, fire, or similar disaster. A manual database, however, may exist in one place without any copies. As a result, manual database would be very vulnerable to a fire or any other natural disaster. In addition, while requesting a manual database system, information must be found by hand instead of electronically. While a digital database will typically allow users to search the entire database for specific information within seconds, someone looking for information in a manual system may have to spend hours surfing for a particular piece of data.

2.2.5 Editing and communication

Manual notices do not allow users to edit data or information quickly. Manual notifications often cannot be edited directly, forcing users to make new copies. To circulate notice on paper, users must require peons and other staff. This app allows users to edit information fields directly, and because data is stored digitally, it is already in a form that can be easily transmitted.

2.3 SUMMARY

We have developed an Android-based Digital Bulletin System to provide college-related notices directly on your android device. In this way, Staff get meeting-related notices and students get class details, event particulars, etc. Just with one click, every piece of information can be shared instantly with less effort.

CHAPTER 3

PROPOSED METHODOLOGY

The proposed system for this digital bulletin board needs an initial registration process for both the staff and students is usually required before posting/sharing information privileges are granted.

Our proposed system has three modules, namely the Admin Module Staff module and Student module.

- In the admin module, the admin manages all the processes like adding users removing the users and maintaining the application.
- In the staff module, the staff shares the information or notice to the students, and it can be in text format.
- In user module, the students can only view the information shared by the staff.
- Whenever the staff sends a message/information, a notification will be shared with the respective students.

3.1 ARCHITECTURE DIAGRAM

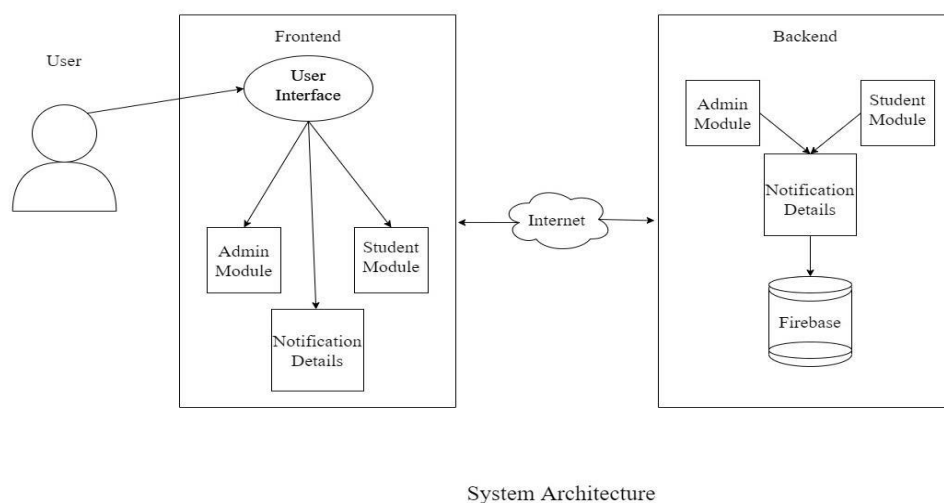


Figure 3.1 System Architecture

The overall architecture of the application is shown in figure 3.1. The front end of the application is a User Interface that is designed in Java and XML. The backend of the application was highly developed in Google Firebase. On initial update, staff's details are uploaded into the local database like username, password, and email are stored in a local database, and further information and data are stored in firebase. In the google firebase where the staff details, department data, and other data are stored in online storage, and an Internet connection is required to perform the retrieve and update activities in the firebase.

3.2 Front-end of the application

The front end is written using XML. Android is an OS that is built basically for mobile phones. It was developed by Google and wholly based on the Linux Kernel and other open-source software. It is used for touch screens on mobile devices such as tablets and smartphones, but nowadays, these are used in Android Auto cars, TV, watches, cameras, etc. It has been one of the best-selling OS in markets for smartphones. Android OS was developed by Android Inc., and Google bought in 2005. Various applications (apps) like games, music players, cameras, etc., are made for these smartphones for running on the android platform. Google play store features more than 3.5 million applications.

The application is developed on a software called Android Studio. These executable applications are installed through a bundle or package called APK (Android Package Kit). In Android, basically, programming is handled in two languages C++ or JAVA and XML (Extension Mark-up Language). Nowadays, KOTLIN is also liked better. The XML file deals with the design, layouts, blueprint, presentation, etc., (a s afront-end)

3.2.1 Activities

It deals with the User Interface and the user interactions to the window. In other words, it is a UI that contains activities. These can be two or more depending upon the application. It starts when the application is launched. At least one action is always present, which is known as Main Activity.

3.3 The backend of the application

The back end is expressed in Java, which is great because anyone can easily adopt it nowadays. You can use the Java standard library (JSL) in addition to the Android library when coding your application. This gives you access to many pre-made objects, and the APIs are thoroughly documented online by Oracle and Google.

3.3.1 Services

Services are the background actions done by the application; these might be continuing operations like a user playing game while surfing the internet. A service might need another sub-service to perform specific tasks. The primary purpose of the Services is to provide non-stop working of the app without breaking any interaction with the user.

3.4 Adding Firebase to Android App

Several services are offered online, such as online processing, storage, real-time database, authorization of the user, etc. Google developed a platform called firebase that provides all these online services. It also gives a daily analysis of the usage of these services and the details of the user using them.

To reduce, it can be said that firebase is a mobile and web application development platform. It provides services that a mobile application might require a web application. Anyone can easily include firebase in this

application, and it will make their online work way more accessible than it was used to be.

The steps followed to add the firebase with android project in android studio:

1. Update the latest android studio (≥ 2.3)
2. Let create a new project in the firebase by selecting the Add project.
3. then, open the android studio and click on Tools in the up left corner.
4. Now select on the firebase option in the drop-down menu.
5. A list will appear on the right side of the screen. It will show different services that firebase offers. Choose the desired service.
6. Now click on the Connection Firebase option in the menu which service we want.
7. Add the dependencies of your services by clicking on the Add [YOUR SERVICE NAME] to the application. In the below image, the firebase cloud messaging assistance is chosen.

3.5 SYSTEM REQUIREMENTS

Hardware and Software requirements

The proposed work has been developed with the system configuration of:

RAM – 4GB

Processor – intel core i5

Hard Disk – 500GB

Operating System – Windows

Tools- Android device, Android Studio IDE, Android SDK, Android emulator

Language – Java, XML

3.6 MODULE DESCRIPTION

3.6.1 User Module

This module contains the notice details which can be viewed by the user. In this module, the students can only view the information shared by the staff. Whenever the team sends a message/information, a notification will be transmitted to the respective students which is shown in figure 4.2 given below.

3.6.2 Admin Module

3.6.2.1 Login

Login is the initial process of the application. For the initial login process, the internet is required. The firebase database will be updated during login. After the password validation, the database will be corrected, and the notice details window will be opened. Figure 4.1 shows the screenshot of the login process.

3.6.2.2 Batch Details

The staff can select the required batch students or year to share the notice/information in this window. Suppose if the staff needs to send the information to all the batches, they can use the option called “All” to share the information.

3.6.2.3 Notice Details

In this application window staff can enter the title of the notice and can add explicit content in the below text box. There is a unique feature here. If the

user wants to share the content only to Class A, they can check the Class A check box and hit the send button. In the same way, for Class B also the notice can be shared privately. If the user wants the student of both Classes to see the message, then they can check both the Class A and Class B checkboxes and hit the send button.

3.7 DATABASE DESIGN

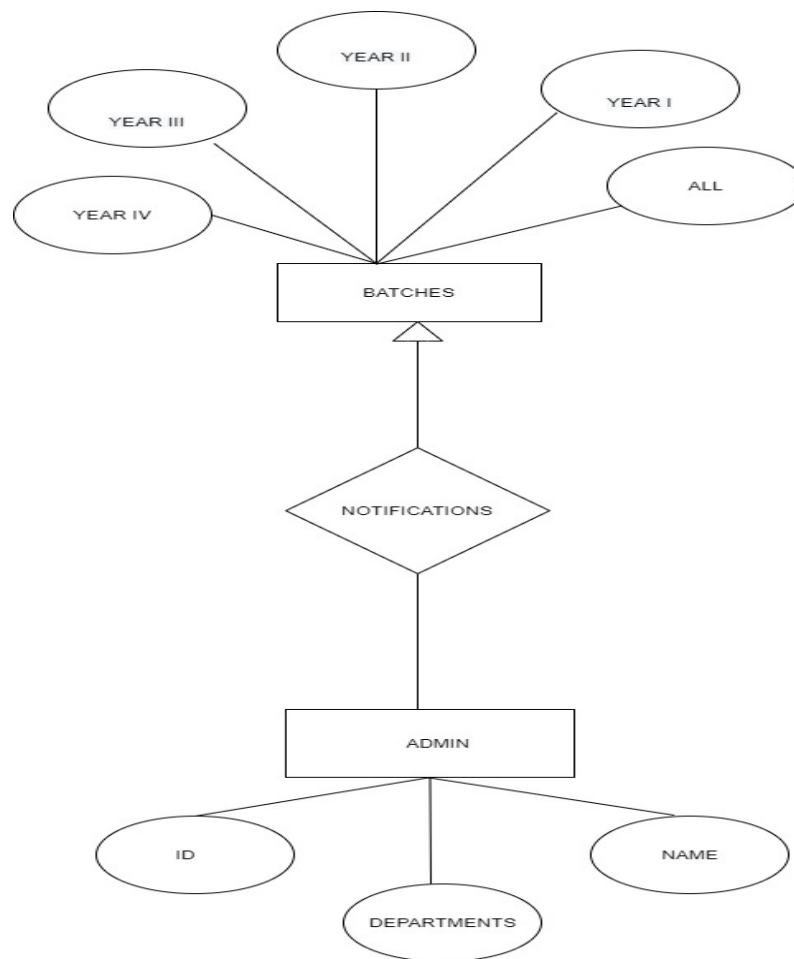


Figure 3.2 Database Design

The figure 3.2 represents the Relationship diagram for the user database schema. It has few relations, and each relationship has a vital attribute to identify rows unique and avoid duplication of data where these schemas can be applied to the firebase database.

CHAPTER 4

RESULT ANALYSIS

4.1 RESULT

All the modules mentioned above are completed and tested successfully. In a period in history the mobile technology opens the windows to the android app. The smartphones are emerging, and the websites are vanishing. It is the time to change from conventional websites to apps that have become part of our daily routine. We are introducing the “Digital Bulletin Board,” the android application the software, which would be a college application.

4.2 SCREENSHOTS

4.2.1 User Module

4.2.1.1 Login

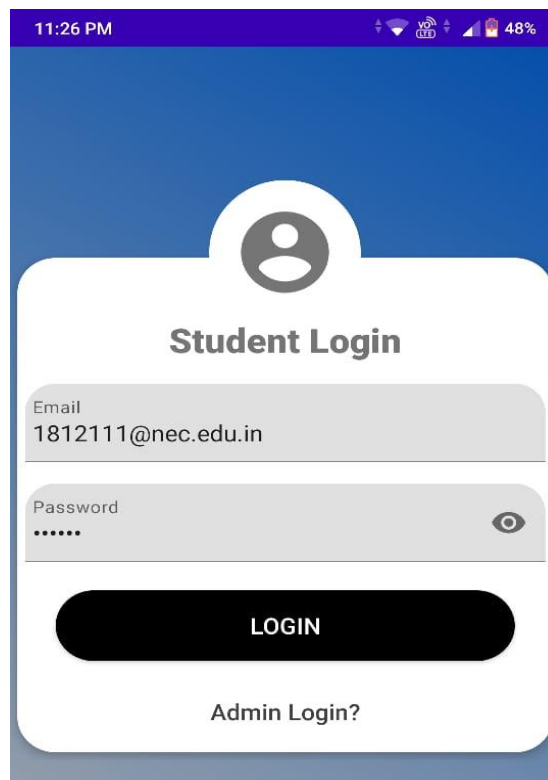


Figure 4.1 User Module

The figure 4.1 represents a login, logging in, or logging on is the entering of registered username, password into a system by a user to access that system. A login generally requires the user to enter two pieces of information, first a username and then a password.

4.2.1.2 Notice Window

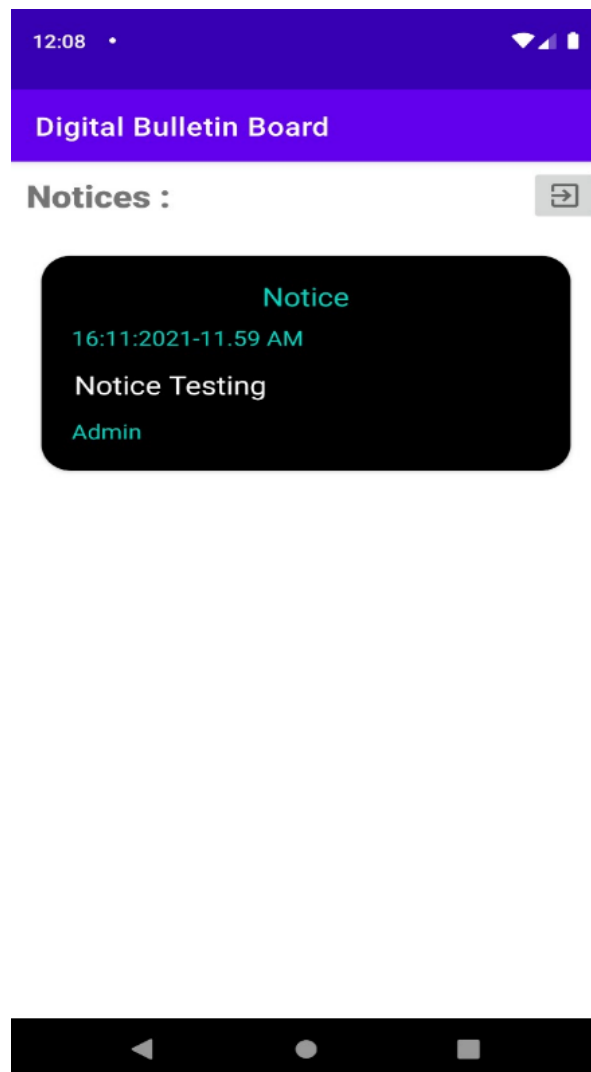


Figure 4.2 Notice Window

In this window figure 4.2 users can view their notices from the admin with the title ,date,time,notices and the sender's name.

4.2.2 Admin Module

4.2.2.1 Admin Login

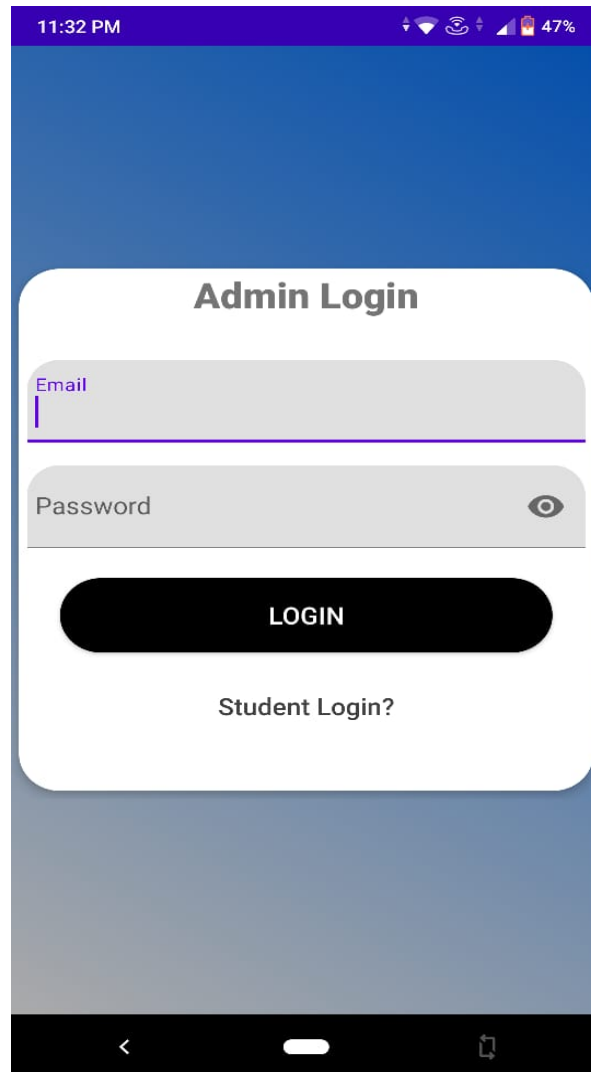


Figure 4.3 Admin Login

The figure 4.3 represents a login, logging in, or logging on is the entering of registered username, password into a system by a user to access that system. Login or entry available to the user with special rights to control or restrict the activity of other users. Here this application, login generally requires the user to enter two pieces of information, first a username and then a password. After clicking the login button, it will be signing into the application.

4.2.2.3 Batch Details

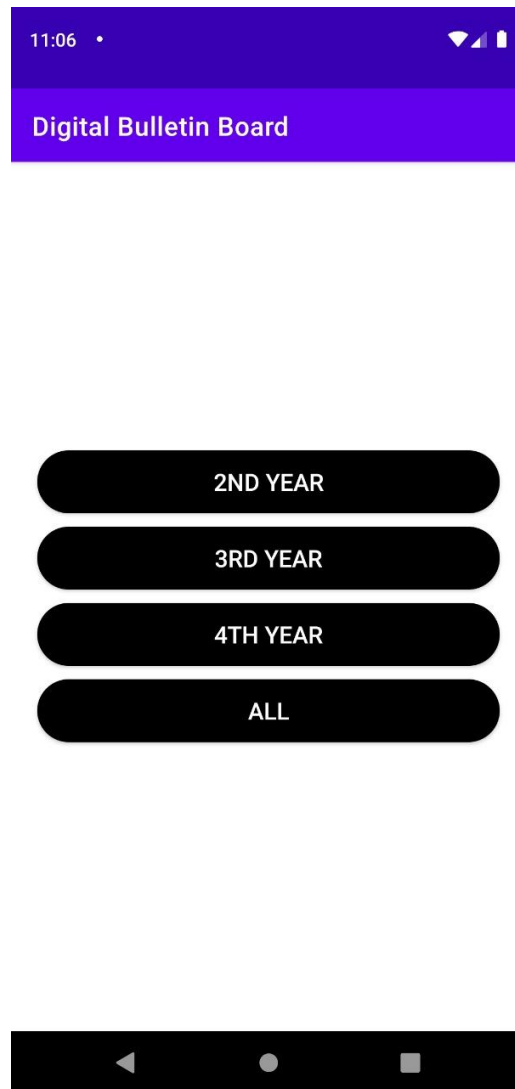


Figure 4.4 Batch Details

- The figure 4.4 shows that the digital bulletin board. After the completion of the login process, we must choose the corresponding year/batch and enter the information in the given text area.
- In the text area, you have to choose the section by clicking the given checkboxes.
- Once our message is typed completely you are ready to share the messages by clicking the send button.

4.2.2.4 NOTICE TEXT AREA

12:10 •

Digital Bulletin Board

☐ Class A ☐ Class B

Send By

Title

Enter you're message

SEND

Figure 4.5 Notice Text Area

Figure 4.5 represents the notice text area where staff can enter the title of the notice and can add explicit content in the below text box. There is a unique feature here. If the user wants to share the content only to Class A, they can check the Class A check box and hit the send button. In the same way, for Class B also the notice can be shared privately. If the user wants the student of both Classes to see the message, then they can check both the Class A and Class B checkboxes and hit the send button.

4.2.3 FIREBASE

4.2.3.1 Authentication

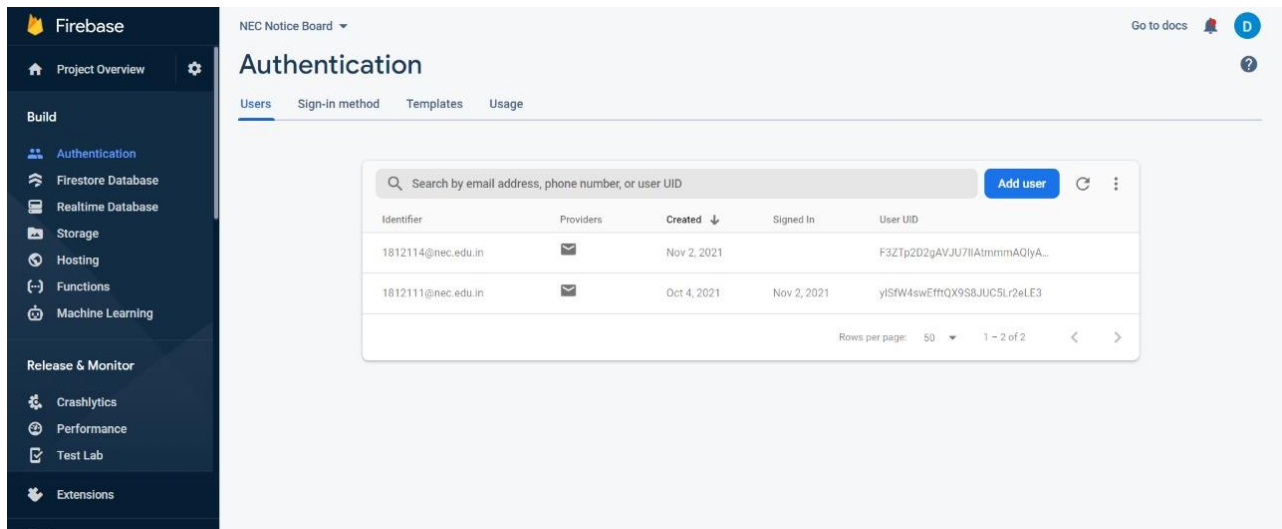


Figure 4.6 Authentication

Figure 4.6 represents the authentication process of recognizing a user's identity. It is the mechanism of associating an incoming request with a set of identifying credentials.

The authentication process always runs at the start of the application, before the permission and throttling checks occur, and before any other code is allowed to proceed.

4.2.3.2 Admin Credentials

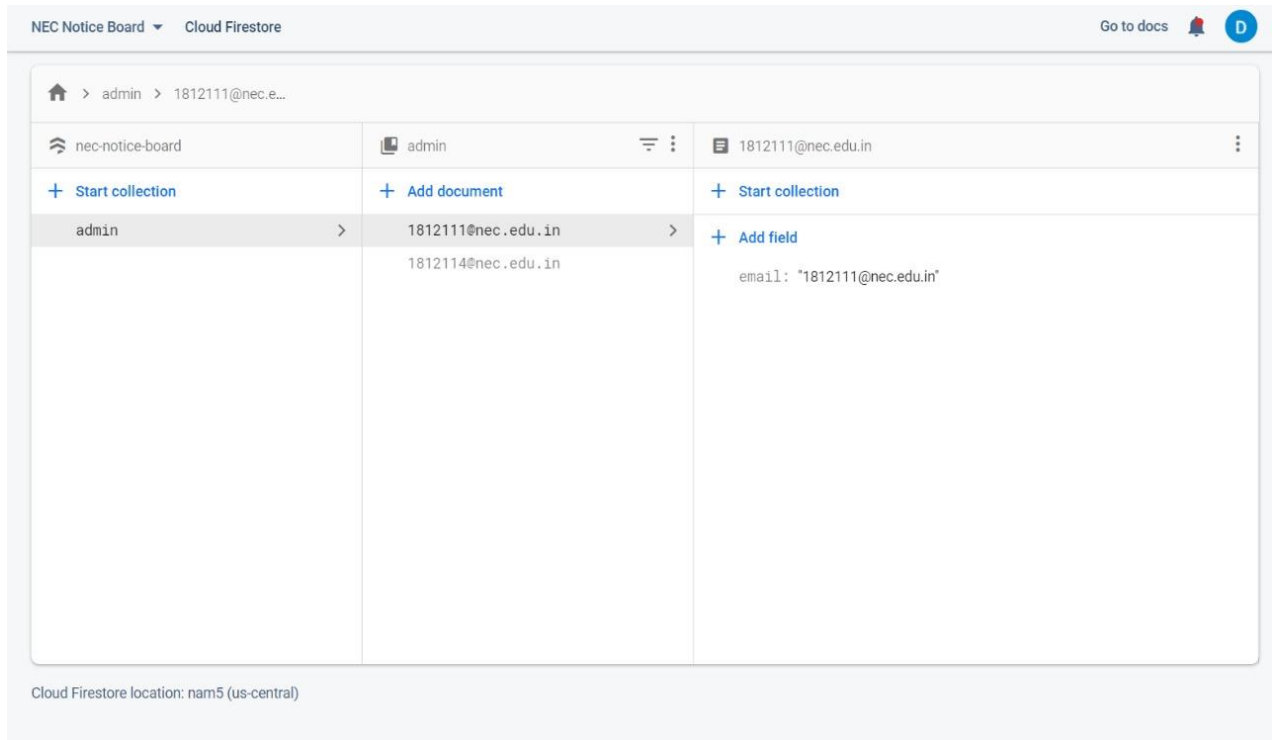


Figure 4.7 Admin Credentials

The figure 4.7 shows the login credentials of the admin like username,password also admin can add the users and remove the users in the database.

4.2.3.3 Notice Database

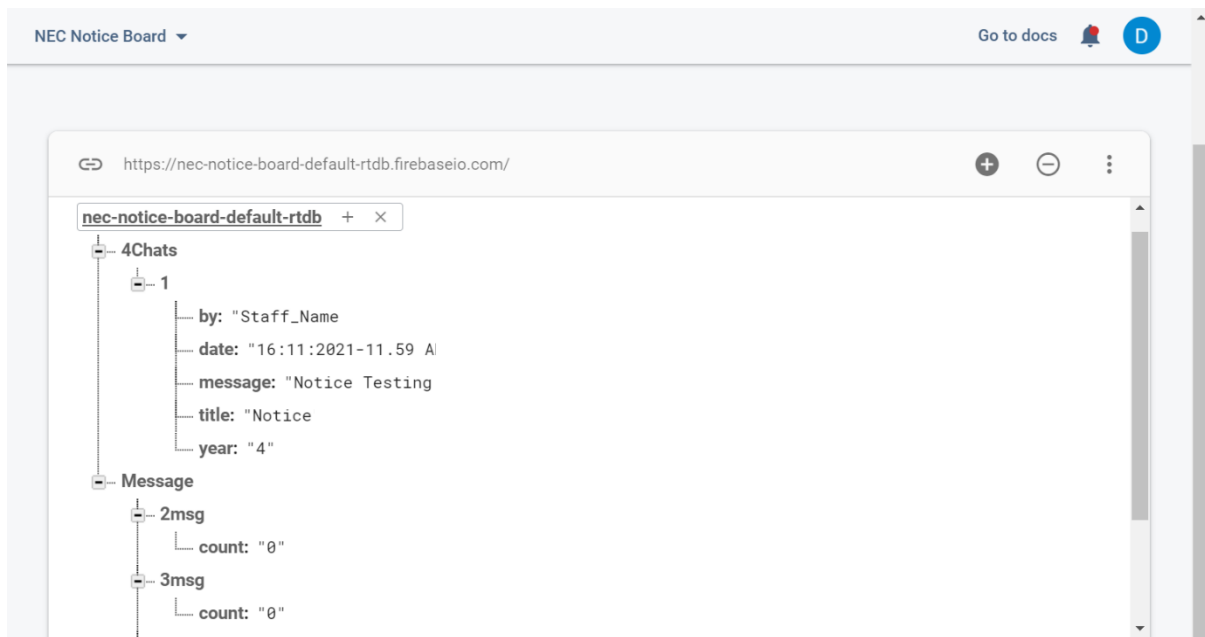


Figure 4.8 Notice Database

The figure 4.8 shows that the Notice or Chats are stored in the format of data under the realtime database. The only data types supported by the Firebase Realtime Database (Android) are string, long, double, Boolean, maps, lists.

4.3 SAMPLE CODE

4.3.1 XML

4.3.1.1 ACTIVITY_MAIN.XML

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".UserHomeActivity">

    <LinearLayout

        android:layout_width="match_parent"
        android:orientation="horizontal"
        android:id="@+id/heading_notices"
        android:layout_height="wrap_content">

        <TextView

            android:layout_width="wrap_content"
            android:layout_alignParentStart="true"
            android:padding="5dp"
            android:text="Notices : "
            android:layout_margin="5dp"
            android:layout_weight="1"
```

```

        android:fontFamily="sans-serif-black"

        android:textSize="24sp"

        android:layout_height="wrap_content"/>
    <ImageButton

        android:layout_width="wrap_content"

        android:layout_gravity="center"

        android:id="@+id/signout"

        android:src="@drawable/ic_baseline_exit_to_app_24"

        android:layout_height="wrap_content"/>
</LinearLayout>

<androidx.recyclerview.widget.RecyclerView

    android:id="@+id/rv"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:layout_below="@+id/heading_notices"

    android:layout_marginStart="5dp"

    android:layout_marginTop="5dp"

    android:layout_marginEnd="5dp"

    android:layout_marginBottom="5dp"

    android:padding="5dp" />
</RelativeLayout>

```

4.3.1.2 ACTIVITY_ADMIN_UPLOAD.XML

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".Admin_Upload">

    <androidx.appcompat.widget.LinearLayoutCompat
        android:layout_width="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:orientation="horizontal"
        android:layout_height="wrap_content">

        <CheckBox
            android:layout_width="wrap_content"
            android:text="Class A"
            android:layout_weight="1"
            android:gravity="center"
            android:layout_gravity="center"
            android:textSize="16sp"
```



```

        android:fontFamily="sans-serif-medium"
        android:layout_height="wrap_content"/>
<CheckBox
    android:layout_marginLeft="15dp"
    android:layout_width="wrap_content"
    android:text="Class B"
    android:textSize="16sp"
    android:fontFamily="sans-serif-medium"
    android:layout_gravity="center"
    android:layout_weight="1"
    android:gravity="center"
    android:layout_height="wrap_content"/>
</androidx.appcompat.widget.LinearLayoutCompat>
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:textSize="16sp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    app:boxCornerRadiusTopEnd="20dp"
    app:boxCornerRadiusTopStart="20dp">
    <EditText

```

```

        android:id="@+id/title_send"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:fontFamily="sans-serif-black"

        android:hint="Title"

        android:textSize="16sp" />
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:layout_marginTop="5dp"

        android:layout_gravity="start"

        android:gravity="start"

        android:textSize="16sp"

        android:layout_marginLeft="20dp"

        android:layout_marginRight="20dp"

        app:boxCornerRadiusTopEnd="20dp"

        app:boxCornerRadiusTopStart="20dp">
<EditText

        android:id="@+id/message_sent"

        android:layout_width="match_parent"

        android:hint="Enter you're message"

        android:textSize="16sp"

```

```

        android:gravity="start"

        android:minHeight="300dp"

        android:maxLines="18"

        android:fontFamily="sans-serif-black"

        android:layout_height="wrap_content"/>
</com.google.android.material.textfield.TextInputLayout>

<androidx.appcompat.widget.AppCompatButton

    android:layout_width="wrap_content"

    android:textSize="18sp"

    android:background="@drawable/button"

    android:text="Send"

    android:textColor="@color/white"

    android:id="@+id/sendbtn"

    android:layout_gravity="center"

    android:layout_marginTop="40dp"

    android:layout_height="wrap_content"/>

</LinearLayout>

```

4.3.2 JAVA

4.3.2.1 MAINACTIVITY.JAVA

```
package com.example.necnoticeboard;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QuerySnapshot;

public class AdminLoginActivity extends AppCompatActivity {

    private FirebaseAuth fbauth;

    private TextView studentlogin ;
```

```

private Button login;

private String str_email,str_password;

private EditText email,password;

private FirebaseFirestore firestore;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_admin_login);

    fbauth = FirebaseAuth.getInstance();

    firestore = FirebaseFirestore.getInstance();

    studentlogin = findViewById(R.id.stafflogin);

    email = findViewById(R.id.email);

    login = findViewById(R.id.login);

    password = findViewById(R.id.password);

    login.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View view) {

            fbauth.signOut();

            str_password = password.getText().toString();

            str_email = email.getText().toString();

            if(str_password.length()>1 && str_email.length() >1){

                ProgressDialog loadingbar = new
ProgressDialog(AdminLoginActivity.this);

```

```

        loadingbar.setTitle("Signing In");

        loadingbar.setMessage("Please wait while we check your
credentials");

        loadingbar.setCanceledOnTouchOutside(false);

        loadingbar.show();

fbauth.signInWithEmailAndPassword(str_email,str_password).addOnS
uccessListener(new OnSuccessListener<AuthResult>() {

    @Override

    public void onSuccess(AuthResult authResult) {

        firestore.collection("admin").document(str_email+"")

            .get().addOnSuccessListener(new

OnSuccessListener<DocumentSnapshot>() {

        @Override

        public void onSuccess(DocumentSnapshot

documentSnapshot) {

            if(documentSnapshot.exists()) {

                Toast.makeText(AdminLoginActivity.this,

"User Admin ", Toast.LENGTH_SHORT).show();

                Intent intent = new

Intent(AdminLoginActivity.this, AdminHomeActivity.class);

                loadingbar.dismiss();

intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |

Intent.FLAG_ACTIVITY_NEW_TASK );

                startActivity(intent);

```

```

        finish();
    }

    else{

        loadingbar.dismiss();

Toast.makeText(AdminLoginActivity.this,"User not a
Admin",Toast.LENGTH_SHORT).show();

    }

}

}).addOnFailureListener(new OnFailureListener() {

    @Override

    public void onFailure(@NonNull Exception e) {

        loadingbar.dismiss();

        Toast.makeText(AdminLoginActivity.this,"Try
again",Toast.LENGTH_SHORT).show();

    }

});

}

}).addOnFailureListener(new OnFailureListener() {

    @Override

    public void onFailure(@NonNull Exception e) {

        loadingbar.dismiss();

        Toast.makeText(AdminLoginActivity.this,"Check
Email and Password",Toast.LENGTH_SHORT).show();

```

```

        }

    });

}

}

});

}

}

```

4.3.2.2 ADMIN_UPLOAD.JAVA

```

package com.example.necnoticeboard;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;

```



```

import java.util.Calendar;

import java.util.Map;

public class Admin_Upload extends AppCompatActivity {

    private FirebaseDatabase database ;

    String str_message, str_title;

    EditText title,message;

    private int count;

    Button send;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_admin_upload);

        database = FirebaseDatabase.getInstance();

        title = findViewById(R.id.title_send);

        message = findViewById(R.id.message_sent);

        send = findViewById(R.id.sendbtn);

        Calendar now = Calendar.getInstance();

        now.add(Calendar.DATE,1);

        int m=now.get(Calendar.MONTH)+1;

        int d=now.get(Calendar.DATE) ;

        int y=now.get(Calendar.YEAR);

        int hr = now.get(Calendar.HOUR);

        int mm =now.get(Calendar.MINUTE);

```

```

String date = d+":"+m+":"+y+"-"+hr+":"+mm;

send.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        str_message = message.getText().toString();

        str_title = title.getText().toString();

        if(str_message != null && str_title!= null){

DatabaseReference dataref =
FirebaseDatabase.getInstance().getReference("Chats");

        DatabaseReference dataref1 =
FirebaseDatabase.getInstance().getReference("Message");

        list_data_notices rf = new
list_data_notices(str_message,date,str_title);
        database.getReference("Message").child("msg").get().addOnSuccessLi
stener(new OnSuccessListener<DataSnapshot>() {

            @Override

            public void onSuccess(DataSnapshot dataSnapshot) {

                if(dataSnapshot.exists()){

                    message_count msgc =
dataSnapshot.getValue(message_count.class);

                    int count = Integer.valueOf(msgc.getCount());

                    count = count+1;

                    dataref.child(String.valueOf(count)).setValue(rf);

                    dataref1.child("msg").child("count").setValue(""+count);

```

```
        }  
    }  
    });  
}  
message.setText("");  
title.setText("");  
}  
});
```

CHAPTER 5

CONCLUSION

Nowadays, Smartphones are become to known. It is time to change from the conventional websites to applications that have become a part of our daily routine. This "Digital Bulletin Board" is the android application software that would be a college application. Digital Bulletin Board is beneficial in an institution or college. This system reduces the paperwork; also, it is easy to maintain the records. Supervision can be done from anywhere. This project mainly minimizes the human effort necessary. The college handles this application, so there is no information leak, and data will be secured. Since it is an android application, anyone can use the system in a very efficient manner and anywhere at any time. It is easy to get the necessary information without latency.

CHAPTER 6

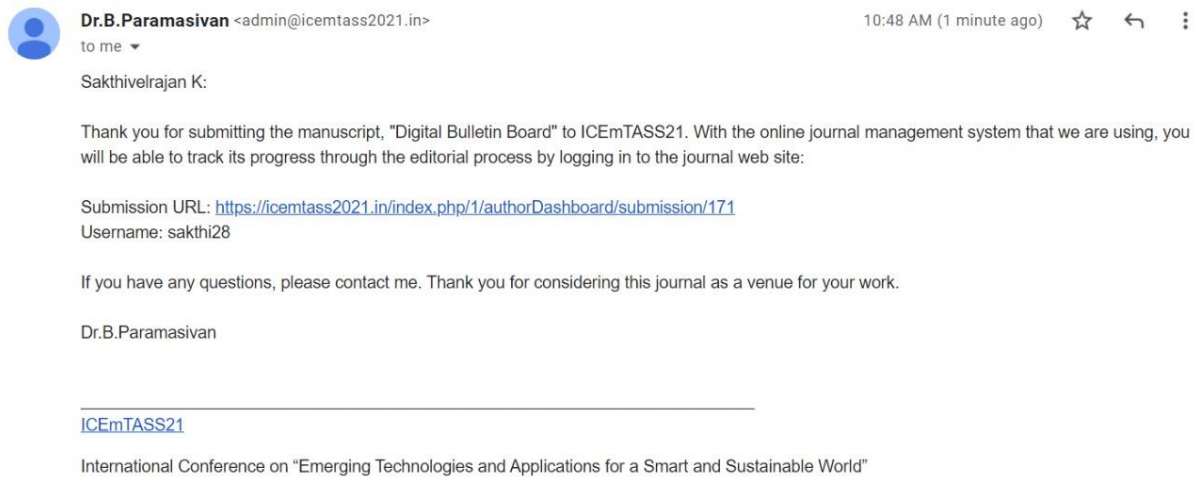
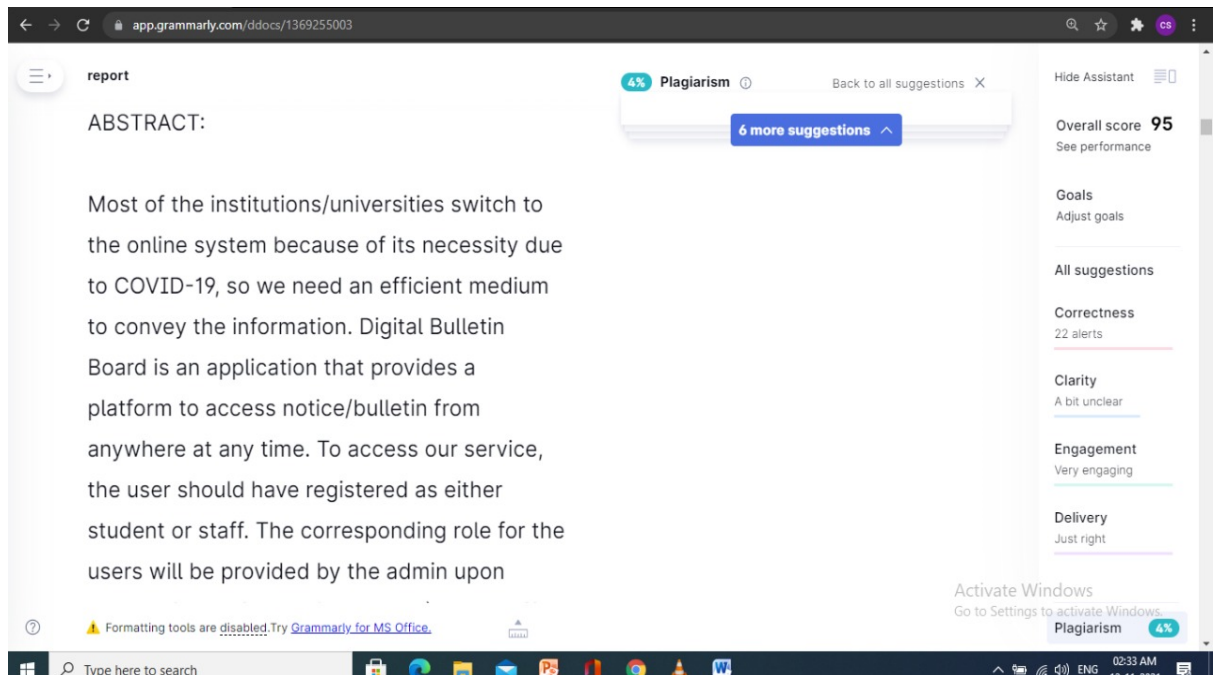


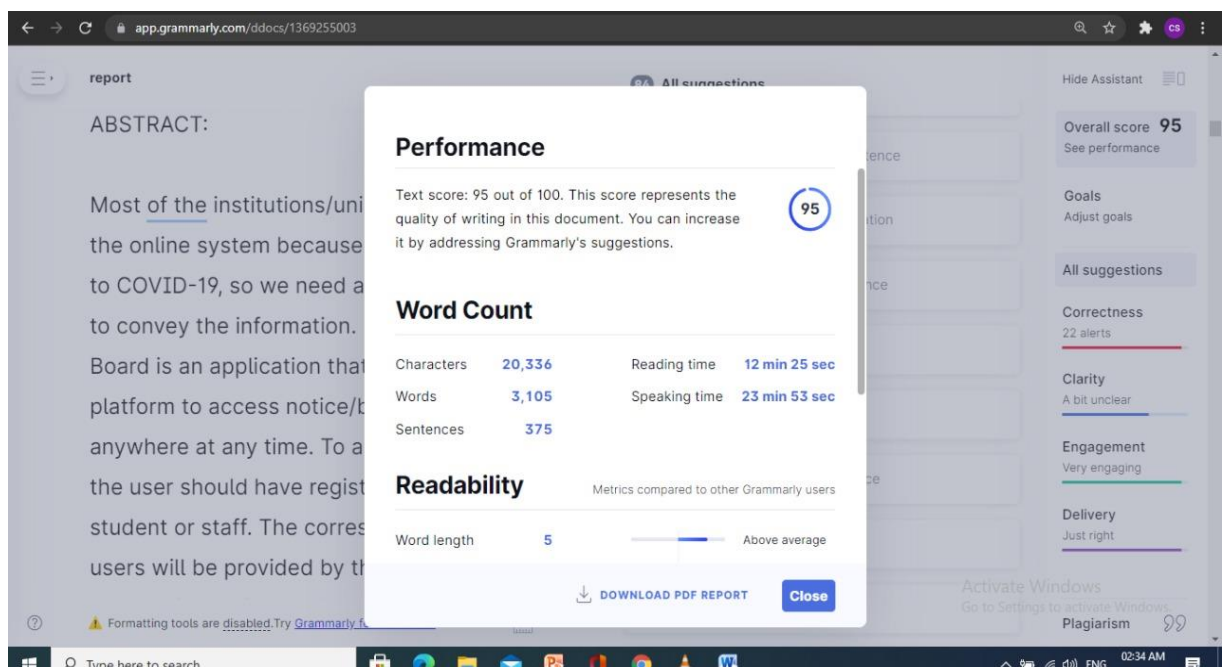
Figure 6.1 Expected Outcome Proof

Applied For International Conference on “Emerging Technologies and Application for a Smart and Sustainable World”.

APPENDICES



Plagiarism Report



Performance and Grammarly check

REFERENCES

1. Pawan Kumar, Kiran Pal, Vikas Bhardwaj GSM Based e-Notice Board Wireless Communication, International Journal of Advanced – 3rd June-2012.
2. Payal Mishra, Pinki Singh, Shivani Gupta SMS Based Wireless Notice Board Design GSM Mobile, International Journal of Advance Research In Science And Engineering IJARSE, Vol. No.2, Issue No.10, 10-Oct-2013.
3. Foram Kamdar, Anubhav Malhotra and Pritish Mahadik Display Message on Notice Board using GSM Advance in Electronic and Electric Engineering. ISSN 2231-1297, Volume 3, Number 7 (2013), pp. 827-832 © Research India Publications.
4. N.Jagan Mohan Reddy and G.Venkeshwaralu Wireless Electronics Display Board Using GSM Technology, International Journal of Electrical, Electronics and Data Communication, 2017 IJEDR Volume 5, Issue 2 ISSN: 2320-2084.
5. Modi Tejal, Prakash Kureshi, Noshin Ayaz, Ostwal Pratiksha Sumtilal Design and Implementation of Digital Notice Board Using Power Line Communication Guifen Gu and Guili Pengthe Survey of GSM Wireless Communication System, International Conference on Computer and Information Application (ICCIA 2010) ISSN: 2321-9939.
6. Kanimozhi.M, Kiruthika shree.K, Kousalya.K , Kesavamoorthy.T , Kalaiyarasi.V Digital Data Displaying Board Using GSM, Digital Data Displaying Board Using GSM International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Issue 3, March 2017, ISSN (Online): 2278 – 8875.

7. Rao H. Di-Fa Chang, Yi-Bing Lin, I SMS: an integration platform for short message service and IP networks in Network IEEE. 2002; 2(15):48-55.2.
8. Swiatkowski Michal, Klaudiusz Wozniak. Olczyk Lukasz. Student Notice Board Based on LED Matrix System Controlled over TCP/IP Protocol, in International Students and Young Scientists Workshop, Photonics and Microsystems IEEE. 2006, 59-60.3.
9. Shruthi K. Smart Notice Board, in Research & Technology in the Coming Decades (CRT 2013), National Conference on Challenges, IET. 2013, 1-4.4.
10. Memon Azam Rafique, Chowdhry Bhawni Shankar, Syed M. Shehram Shah, Memon Tarique Rafique, Syed. MZ Abbas Shah. An Electronic Information Desk System for Information Dissemination in Educational Institutions, computing for Sustainable Global Development (INDIA Com), 2015 2nd International Conference. 2015, 1275-1280.5.
11. Dr. Himani Goyal Wireless Notice Board using UWB with monitoring System in International Journal of Advance Research in Computer Science and Management Studies. 2014; 2(11):390-395.6.
12. Ziya Alkar Ali. An Internet Based Wireless Home Automation System for Multifunctional Devices in IEEE TRANSACTIONS ON Consumer Electronics. 2005; 51(4):1169-1174.7.
13. Vishal Deshmukh S. Design and Implementation of Multiple LED Notice Boards by Using ZIGBEE Technology”, in International Journal Of Engineering and Computer Science. 2015; 4(2):10282-10285.8.

14. Kumar Pawan, Bhardwaj Vikas, Pal Kiran Singh Narayan, Mishra Amit. GSM based e-Notice Board: Wireless Communication, in International Journal of Soft Computing and Engineering. 2012; 2(3):601-605. Xin-kan Mu, Yong-hong Chen. SMS-based Remote Computer Supervisory Control System Design, in Journal of Software. 2014; 9(3):621-626.10.
15. Mayur Bhoyar R, Suraj Chavhan. Secure Method of Updating Digital Notice Board through SMS with PC Monitoring System, in IOSR Journal of Computer Science, e-ISSN: 2278-0661. 2014, 24-29.11.
16. Smt. M. Baby, Harini P. SMS Based Wireless E-Noticeboard in, International Journal of Emerging Technology and Advanced Engineering. 2013;3(3):181-185.12.
17. Gowtham R. Multiuser Short Message Service Based Wireless Electronic Notice Board in International Journal of Engineering and Computer Science. 2013; 2(4):1035-1041.13.
18. Km. Singh Ritu. Keyboard Driven Electronic Notice Board in International Journal of Advanced Technology in Engineering and Science. 2015; 3(4):46-50.14.
19. Gupta Hardik, Shukla Puja, Nagwekar Ankita. GSM Based LED Scrolling Display Board International Journal of Students Research in Technology and Management. 2013; 1(3):278-291.15.
20. Jeff Brown. SMS: The Short Message Service, in IEEE Computer Society. 2007, 106-111.16.
21. Km. Singh Ritu Keyboard Driven Electronic Notice Board in International Journal of Advanced Technology in Engineering and Science. 2015; 3(4):46-50.