ONLINE ATTENDANCE REGISTER APP USING ANDROID STUDIO

A PROJECT REPORT

Submitted by

PRABHU.R (311620104040)

KUMAR.S.N (311620104302)

in partial fulfillment for the award of degree of

BACHELOR OF ENGINEERING IN COMPUTER SCIENCEAND ENGINEERING

MISRIMAL NAVAJEE MUNOTH JAIN ENGINEERING COLLEGE

THORAIPAKKAM, CHENNAI-60009

ANNA UNIVERSITY: CHENNAI 600025APRIL

2023

BONAFIDE CERTIFICATE

Certified that this project report

"ONLINE ATTENDANCE REGISTER APP USING ANDROID STUDIO "

is the bonafide work of

"**PRABHU.R AND KUMAR.S.N "**

who carried out the project work under my supervision.

SIGNATURE

Dr. N. Saravanan B.E.,ME., Ph. D

HEAD OF THE DEPARTMENT

Department of Computer Scienceand Engineering

Misrimal Navajee Munoth Jain Engineering College,

Thoraipakkam,

Chennai-600097

SIGNATURE

Mrs. P.Asha B.E.,M.E., (Ph. D)

SUPERVISOR

ASSOCIATE PROFESSOR

Department of Artificial Intelligence and Data Science

Misrimal Navajee Munoth Jain Engineering College,

Thoraipakkam,

Chennai-600097

Submitted for the Project Viva-Voce Examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

S.NO	ABBREVIATIONS	EXPANSION
1	JDK	JAVA DEVELOPMENT KIT
2	XML	EXTENSIBLE MARKUP LANGUAGE
3	SDK	SOFTWARE DEVELOPMENT KIT
4	API	APPLICATION PROGRAMMING INTERFACE

ACKNOWLEDGEMENT

We express our sincere gratitude to our honorable **Secretary** (Administration) **Dr. Harish L Metha** and Secretary (Academic) **Shri L. Jaswant S Munoth** for providing the infrastructure facilities to do this project during our course period.

We thank our **Principal**, **Dr. C. C. Christopher** for his support and motivation for the development and completion of this project.

We express profound sense of gratitude and thanks to our **Head Of The Department**, **Dr.N. Saravanan** for his valuable suggestions and guidance forthe development and completion of this project.

We thank our project coordinator, **Mrs.P.Asha, Associate Professor**, for her valuable suggestions and constant encouragement that led us to the successful completion of our project.

We thank our supervisor **Mrs. P.Asha Associate Professor** for her constant help and valuable suggestions.

We thank all the Teaching and Non-Teaching Staff members of our Department who helped us to complete our project.

Above all we thank the Almighty, and our parents and for their constant supportand encouragement for completing this project.

MISRIMAL NAVAJEE MUNOTH JAIN ENGINEERING COLLEGE,CHENNAI-97

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

Producing competent Computer Engineers with a strong background in the latest trends and technology to achieve academic excellence and to become pioneer in software and hardware products with an ethical approach to serve the society.

MISSION

To provide quality education in Computer Science and Engineering with the state- of-the-art facilities. To provide the learning audience that helps the students to enhance problem solving skills and to inculcate in them the habit of continuous learning in their domain of interest. To serve the society by providing insight solutions to the real-world problems by employing the latest trends of computing technology with strict adherence to professional and ethical responsibilities.

ABSTRACT

The rapid growth of technology has revolutionized various aspects of our lives, including the way educational institutions manage attendance records. In this project, we present the development of an Attend My Class App using Android Studio, a powerful integrated development environment (IDE) for Android application development. The aim of this app is to provide a convenient and efficient solution for educational institutions to manage attendance records digitally.

The Attend My Class App leverages the capabilities of Android devices, such as smartphones and tablets, to streamline the attendance-taking process. The app incorporates features that enable teachers or instructors to record attendance digitally, eliminating the need for traditional manual methods. Students can also access the app to view their attendance records and stay updated on their attendance status.

The app utilizes various technologies and functionalities provided by Android Studio to ensure a robust and user-friendly experience. These include user authentication and authorization, database integration for storing attendance data securely, and intuitive user interfaces for seamless navigation and interaction. Additionally, the app utilizes network connectivity to synchronize attendance data between devices and the central server, ensuring real-time updates and data integrity.

The development process involves designing and implementing a database schema to store attendance-related information efficiently. The user interface is designed to be intuitive and visually appealing, providing an enjoyable user experience. The app also incorporates advanced features, such as notifications for upcoming classes or low attendance thresholds, enhancing its functionality and usefulness.

The Attend My Class App is expected to provide significant benefits to educational institutions, teachers, and students. By automating the attendance-taking process, it reduces the administrative burden on teachers and improves overall efficiency. Students can easily access their attendance records, enabling them to monitor their attendance and take necessary actions to maintain satisfactory attendance levels.

In conclusion, the development of the Attend My Class App using Android Studio provides a practical and efficient solution for managing attendance records in educational institutions. By leveraging the capabilities of Android devices, the app simplifies the attendance-taking process, enhances data accuracy, and promotes a digital approach to attendance management. Future enhancements may include additional features such as analytics and reporting capabilities to further optimize attendance monitoring and analysis.

CHAPTER 1

INTRODUCTION

In today's technologically advanced era, the traditional methods of manual attendance tracking in educational institutions are being replaced by digital solutions. One such solution is the "Attend My Class" app, developed using Android Studio. This app leverages the power of Android devices and the comprehensive features of Android Studio, an integrated development environment (IDE), to offer a convenient and efficient way to manage attendance records.

The "Attend My Class" app is designed to simplify the attendance-taking process for teachers and provide students with easy access to their attendance data. By harnessing the capabilities of Android devices, the app offers a range of features and functionalities that streamline the entire attendance management workflow.

Gone are the days of maintaining traditional attendance registers and manually calculating attendance percentages. With the "Attend My Class" app, teachers can effortlessly mark attendance electronically, eliminating errors and reducing administrative burdens. The app provides a user-friendly interface for teachers to record attendance, view class schedules, and access student profiles. Additionally, advanced features such as notifications for low attendance or upcoming classes empower teachers to take timely actions and improve attendance monitoring.

For students, the "Attend My Class" app serves as a convenient platform to access their attendance records anytime, anywhere. They can easily view their attendance history, check their current attendance status, and receive alerts if their attendance falls below a specified threshold. This promotes a sense of accountability among students and enables them to proactively manage their attendance, ensuring they stay on track with

their academic progress.

The development of the "Attend My Class" app is based on the robust Android Studio IDE, which provides a comprehensive set of tools and resources for Android application development. The app incorporates secure database integration to store attendance data, authentication mechanisms to ensure authorized access, and seamless network connectivity for real-time synchronization between devices and servers. The user interface of the app is designed to be intuitive and visually appealing, enhancing the overall user experience for both teachers and students.

By embracing digital attendance management through the "Attend My Class" app, educational institutions can streamline their administrative processes, reduce paperwork, and improve accuracy. This app promotes efficiency and transparency, benefiting teachers, administrators, students, and parents alike. With the widespread use of Android devices, the "Attend My Class" app offers a practical and accessible solution to attendance tracking in the digital age.

In the subsequent sections of this project, we will delve into the technical aspects of the development process, including database design, user authentication, and the implementation of key features. Furthermore, we will discuss potential future enhancements and the multitude of benefits that the "Attend My Class" app can bring to educational institutions.

CHAPTER 2

LITERATURE SURVEY

The development of the "Attend My Class" app using Android Studio requires a comprehensive understanding of existing research and literature related to attendance management and mobile application development. The following literature survey provides an overview of relevant studies, research papers, and existing applications that have contributed to the understanding and implementation of similar systems.

"Mobile-Based Attendance Management System for Educational Institutions" by Akash Kandpal et al. (International Journal of Computer Applications, 2017):
This research paper discusses a mobile-based attendance management system that utilizes Android devices to simplify attendance tracking in educational institutions. It highlights the advantages of a mobile-based approach, such as convenience, real-time updates, and improved data accuracy. The study explores design considerations and implementation strategies for developing such a system.

"Development of Mobile Attendance Management System using Android Application" by Sanju Verma et al. (International Journal of Advanced Research in Computer Science and Software Engineering, 2016):
This paper focuses on the development of a mobile attendance management system using Android technology. It highlights the benefits of mobile applications in attendance tracking, including accessibility and ease of use. The study emphasizes the importance of incorporating secure database management and authentication mechanisms to ensure data privacy and security.

"Design and Implementation of a Web-based Mobile Attendance Management

System" by Jing-Wen Huang et al. (2012 International Conference on Cloud and Service Computing):
This research paper presents a web-based mobile attendance management system that allows users to access attendance records through web and mobile interfaces. It discusses the significance of cross-platform compatibility and user-friendly interfaces for an enhanced user experience. The study explores the integration of cloud computing technologies for data storage and synchronization.

"Attendance Management System using Android and QR Code" by Hemanth Kumar P and Harsha N (International Journal of Engineering Research & Technology, 2014):
This paper proposes an attendance management system that utilizes Android devices and QR code technology for efficient and accurate attendance tracking. The study highlights the advantages of QR codes in capturing attendance data and discusses the implementation challenges and considerations. It also emphasizes the importance of data backup and recovery mechanisms.

"A Review on Mobile Attendance Application" by Venkata Prasad M and Rajesh G (International Journal of Innovative Research in Computer and Communication Engineering, 2015):
This review article provides an overview of various mobile attendance applications available in the market. It discusses the features and functionalities offered by these applications, such as real-time updates, notifications, and data analysis. The study emphasizes the need for user-friendly interfaces and seamless integration with existing systems in educational institutions.

These studies and research papers demonstrate the significance of mobile-based attendance tracking systems and offer valuable insights into various aspects of their development, including design considerations, database management, authentication

mechanisms, and integration with existing infrastructures. By leveraging this existing knowledge, the development of the "Attend My Class" app using Android Studio can benefit from proven methodologies and best practices, leading to the creation of an efficient and effective attendance management solution.

CHAPTER 3

SYSTEM ANALYSIS

EXISTING SYSTEM

3.1.1 OVERVIEW

The Online Attendance Tracking App developed using Android Studio is a convenient and efficient solution for managing attendance in educational institutions. It offers a user-friendly interface for teachers to mark attendance and access student profiles. Students can easily view their attendance records, check their current attendance status, and receive notifications. The app utilizes secure database integration, authentication mechanisms, and real-time synchronization for accurate and accessible attendance data. With its streamlined features and seamless functionality, the app enhances the attendance management process for both teachers and students.

DRAWBACK

The "Attend My Class" app, developed using Android Studio, offers numerous benefits for attendance management. However, it's essential to consider potential drawbacks. These include technical dependencies on stable internet connectivity and device compatibility, which may lead to delays or inaccuracies in attendance recording. Privacy and security concerns arise from storing attendance data in a database, necessitating measures such as encryption and secure authentication. Device limitations and user adoption challenges can affect the app's functionality and user engagement. Additionally, reliance on technology introduces the risk of technical issues and system failures. Addressing these concerns through thorough testing, robust security measures, user training, and contingency plans can help ensure a successful implementation of the "Attend My Class" app.

REQUIREMENT ANALYSIS

A requirements specification is a document that outlines the necessary requirements for a software product. The purpose of a software

requirements specification is to provide a comprehensive understanding of the software project, including its objectives and parameters. It should detail the project's intended audience, user interface, and any hardware or software requirements that must be met to ensure the software functions as intended.

SOFTWARE REQUIREMENT

The software requirements give a detailed description of the system and all itsfeatures.

- JDK
- ANDROID STUDIO
- PROGRAMMING LANGUAGES (JAVA OR KOTLIN)
- DATABASE MANAGEMENT(FIREBASE OR SQLITE)
- REQUIRED API AND LIBRARISE

HARDWARE REQUIREMENT

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete engineer as the starting point for the system design.

Ram : 4GB Ram or more

Processor : Any ARM-based Processor

Storage : 10GB or more

Processor Speed : 1.4GHZ or more

Operating System : Android 6.0 or above

TOOLS USED

JAVA DEVELOPMENT KIT

The JDK is required to compile, run, and debug Java code. It includes the Java Runtime Environment (JRE), necessary libraries, and development tools.

ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for Android app development. It provides a complete set of tools, including a code editor, debugger, emulator, and visual layout editor, to streamline the app development process.

JAVA

Java is the primary programming language used for Android app development. It offers a robust and mature ecosystem with extensive libraries, frameworks, and community support. Java is widely used for building Android apps due to its performance, versatility, and compatibility

.3.3.4.DATABASE

Firebase is a cloud-based platform provided by Google that offers a variety of services for building and managing mobile and web applications.

Firebase provides backend functionality, including real-time database, authentication, cloud storage, push notifications, and analytics. In the context of an online medicines ordering app, Firebase can be utilized for user authentication, real-time data synchronization, and storing order information securely.

CHAPTER 4

SYSTEM DESIGN

SYSTEM ARCHITECTURE

The diagram shows the system architecture of the project. Each block shows their structural and functional components of the project.

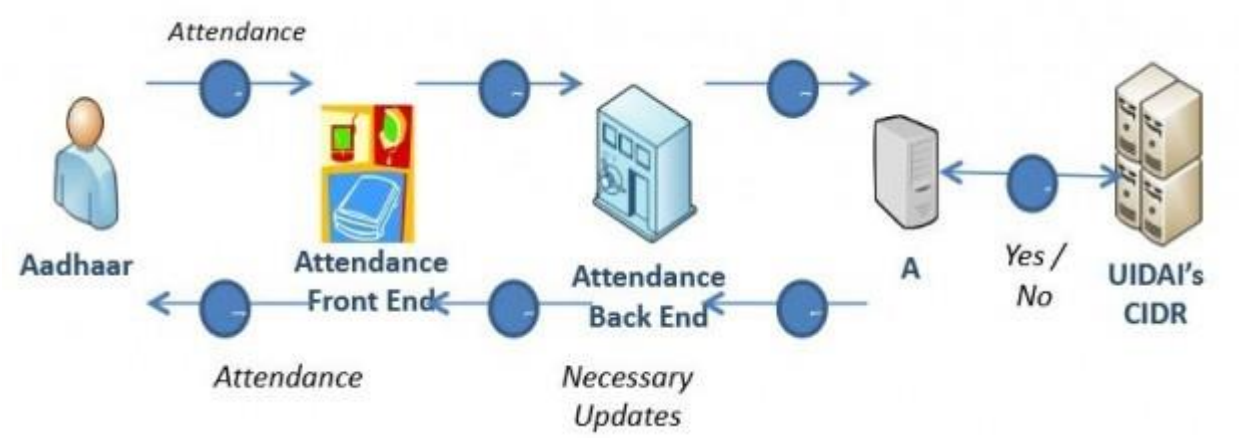


Figure 4.1 System Architecture

Figure 4.1 shows the system architecture diagram for training and testing of the project.

USE CASE DIAGRAM

This diagram shows the user such as actor, system and the role of developer in this project. This behavior diagrammodels the functionality of the system using use cases.

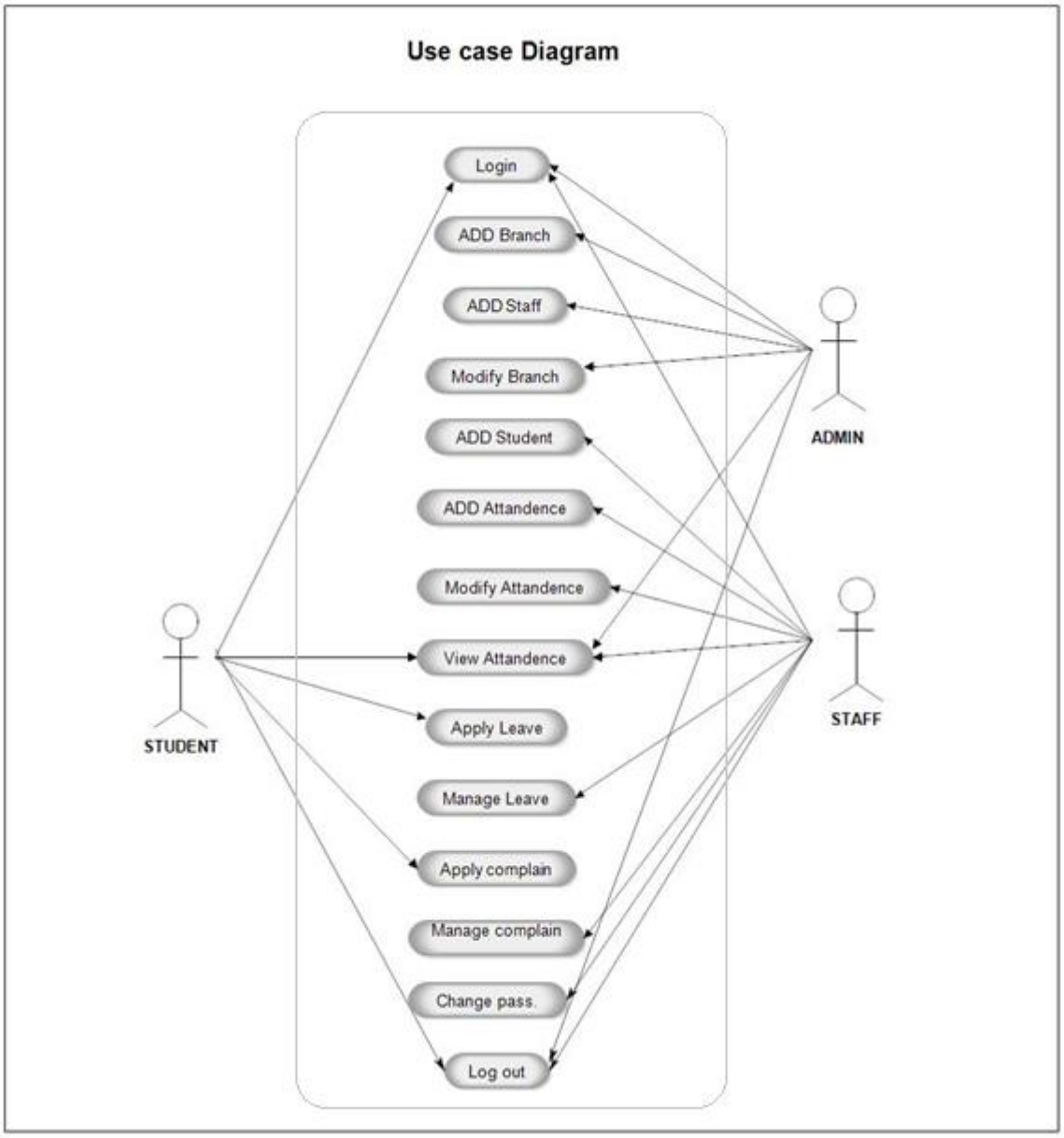


Figure 4.2 Use Case Diagram

Figure 4.2 shows the role of user such as actor, system and developer in this project.

CLASS DIAGRAM

The structure of the application id described in the class diagram by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. Each class has their attributes and the operations defined with the relationship between the classes.

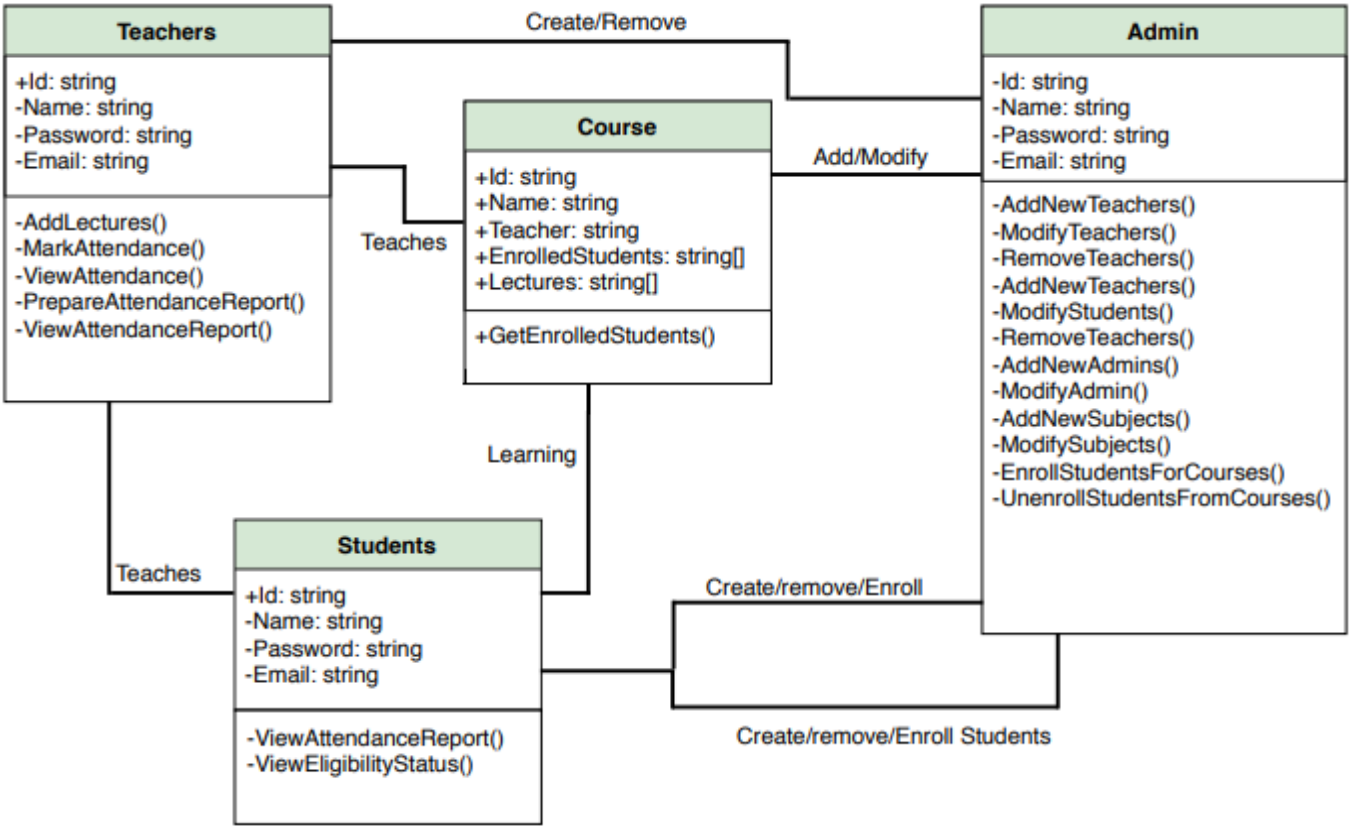


Figure 4.3 CLASS DIAGRAM

Figure 4.3 shows the system's classes, their attributes, operations (or methods), and the relationships among objects.

CHAPTER 5

IMPLEMENTATION

User Interface Design:

Design the user interface (UI) for the app, ensuring it is intuitive, user-friendly, and visually appealing. Consider the different screens and functionalities needed for teachers and students, such as attendance marking, attendance history, and notifications.

Data Management:

Set up a database to store attendance data securely. Decide on the appropriate database management system, such as SQLite or Firebase Realtime Database, and design the database schema to efficiently store and retrieve attendance records.

Authentication and Authorization:

Implement a secure authentication system to verify the identity of users, such as teachers and administrators. This ensures that only authorized individuals can access and modify attendance records.

Attendance Marking:

Develop the functionality for teachers to mark attendance. This may involve options for manual entry or integration with other technologies such as QR code scanning or NFC tags for automated attendance capture.

Real-time Synchronization:

Implement real-time synchronization between the app and the database to ensure immediate updates and accurate attendance records across devices. Utilize technologies like Firebase Cloud Messaging or WebSocket for real-time data exchange.

Notifications and Alerts:

Enable the app to send notifications and alerts to teachers and students regarding attendance status, upcoming classes, or any changes related to attendance. Implement push notification services like Firebase Cloud Messaging or local notifications to deliver timely updates.

Reporting and Analytics:

Develop features to generate attendance reports and analytics. This can include generating statistical data, visualizing attendance patterns, and exporting reports in various formats for further analysis or record-keeping purposes.

Testing and Debugging:

Perform comprehensive testing and debugging to identify and fix any issues or bugs in the app. Conduct testing on different Android devices and screen sizes to ensure optimal performance and compatibility.

Throughout the implementation process, it is important to follow best practices for code organization, maintainability, and documentation. Regularly update the app to address user feedback, add new features, and improve overall performance and user experience.

CHAPTER 6

RESULT AND ANALYSIS

Result:

Improved Efficiency:

The app significantly improves the efficiency of the attendance management process. Teachers can quickly mark attendance using their Android devices, eliminating the need for manual paper-based registers. This saves time and reduces administrative burdens, allowing teachers to focus more on teaching and less on administrative tasks.

Enhanced Accuracy:

The app improves the accuracy of attendance tracking. With real-time synchronization between devices and the database, attendance records are updated instantly, reducing the chances of errors or discrepancies. The use of automated attendance capture methods like QR code scanning or NFC tags further enhances accuracy and minimizes the risk of human error.

Accessible Attendance Data:

The app provides easy access to attendance data for both teachers and students. Teachers can view and analyze attendance records, generate reports, and identify attendance patterns. Students can check their attendance status, view their attendance history, and receive notifications to stay updated. This accessibility empowers teachers and students to monitor attendance effectively and take necessary actions when needed.

Analysis:

Data Insights and Analytics:

The app enables the generation of attendance reports and analytics, providing valuable insights into attendance patterns. Teachers can identify trends, track individual student attendance, and make data-driven decisions to improve attendance rates. The ability to visualize attendance data through charts or graphs enhances data analysis and helps in identifying areas that require attention or improvement.

User Feedback and Satisfaction:

User feedback plays a crucial role in evaluating the app's performance. Gathering feedback from teachers and students helps identify any usability issues, bugs, or feature requests. Regular updates and addressing user feedback contribute to user satisfaction and ensure that the app meets the evolving needs of the users.

Scalability and Adaptability:

The app can be scaled and adapted to suit the specific requirements of different educational institutions. Whether it's a small school or a large university, the app can accommodate multiple classes, subjects, and student groups, making it versatile and adaptable for various educational settings.

In conclusion, the Online Attendance Tracking App developed using Android Studio delivers positive results in terms of improved efficiency, enhanced accuracy, and accessibility of attendance data. The analysis highlights the value of data insights, user feedback, and the app's scalability and adaptability. These benefits contribute to a streamlined attendance management process and ultimately improve the overall academic experience.

CHAPTER 7

SYSTEM TESTING

User Interface Testing:

Validate that all user interface elements (buttons, forms, menus) are displayed correctly and are interactive.

Test the app's responsiveness across different screen sizes and orientations.

Verify that the user interface adheres to design guidelines, such as consistent branding, proper layout, and appropriate use of colors and fonts.

Functionality Testing:

Test each module's functionality, such as user registration, medicine listing, cart management, order placement, and user profile management.

Verify that users can search for medicines, add items to the cart, proceed to checkout, and complete the payment process without any errors.

Ensure that the app properly handles various scenarios, such as invalid inputs, out-of-stock items, and order cancellation.

Integration Testing:

Test the integration of different components, such as the app's frontend with the backend server and APIs.

Verify that data is correctly retrieved from and stored in the backend database.

Test the integration with external services, such as payment gateways, to ensure smooth communication and proper handling of transactions.

Performance Testing:

Assess the app's performance under different conditions, including varying

network speeds, device capabilities, and simultaneous user loads.

Measure and optimize the app's response time, load time, and resourceutilization.

Test the app's ability to handle high traffic and peak usage scenarios.

Security Testing:

Conduct security testing to identify vulnerabilities and potential security risks.Verify that user data is securely transmitted, stored, and protected.

Test authentication mechanisms to ensure secure user login and session management.

Usability Testing:

Evaluate the app's usability by conducting user tests and gathering feedback on the app's ease of use, intuitiveness, and overall user experience.

Identify areas for improvement in terms of navigation, layout, and clarity of instructions.

Ensure that the app is accessible to users with disabilities, adhering to accessibility guidelines.

Compatibility Testing:

Test the app on different Android devices with varying screen sizes, resolutions, and operating system versions.

Verify that the app functions correctly and appears as expected across a range of devices and Android versions.

Error Handling and Recovery Testing:

Test the app's ability to handle and recover from errors, such as networkfailures,

server timeouts, or invalid inputs.

Validate that appropriate error messages are displayed, and users are guided to take necessary actions to resolve issues.

Localization Testing:

Verify the app's compatibility with different languages and cultures.

Test that all app content, including text, labels, and notifications, is correctly localized and displayed in the desired language.

Regression Testing:

Perform regression testing to ensure that new feature implementations or bug fixes do not negatively impact existing functionality.

Re-test previously identified issues to ensure they have been resolved.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEME

8.1.CONCLUSION

In conclusion, the Online Attendance Tracking App developed using Android Studio offers a user-friendly and efficient solution for attendance management in educational institutions. The app improves efficiency by simplifying the attendance marking process and reducing administrative burdens. It ensures enhanced accuracy through real-time synchronization and automated attendance capture methods. With accessible attendance data, teachers and students can easily monitor attendance and make informed decisions. The app's data insights and analytics enable teachers to track attendance patterns and improve attendance rates. User feedback plays a vital role in refining the app and ensuring user satisfaction. The app's scalability makes it suitable for various educational settings. Overall, the Online Attendance Tracking App optimizes attendance management, streamlines processes, and enhances the academic experience.

8.2.Future Enhancements:

Biometric Authentication:

Implement biometric authentication options such as fingerprint or face recognition to enhance security and streamline the login process for teachers and students. This provides an additional layer of protection and convenience for users.

Integration with Learning Management Systems:

Integrate the app with popular Learning Management Systems (LMS) to synchronize attendance data with other educational platforms. This allows seamless data exchange between the attendance app and the LMS, providing a comprehensive solution for teachers and students.

Automated Attendance Analysis:

Develop advanced algorithms to automatically analyze attendance data and generate insights. This can include identifying patterns, trends, and correlations between attendance and academic performance, enabling teachers to make data-driven interventions and improve student outcomes.

Gamification Elements:

Introduce gamification elements to increase student engagement with attendance tracking. Incorporate features such as rewards, badges, or leaderboards based on attendance records, encouraging students to maintain high attendance rates and fostering a sense of healthy competition.

Geolocation Tracking:

Utilize geolocation services to track attendance based on students' physical presence in designated areas, such as classrooms or lecture halls. This can provide an additional layer of verification and accuracy in attendance marking.

Integration with Calendar Systems:

Integrate the app with popular calendar systems to automatically populate class schedules and reminders for both teachers and students. This helps in reducing manual data entry and ensures that everyone stays informed about upcoming classes and attendance requirements.

Multi-platform Compatibility:

Expand the app's compatibility to support multiple platforms, such as iOS and web browsers, to cater to a wider range of users. This allows users to access attendance data and perform necessary actions across different devices and platforms.

Data Visualization and Reporting Enhancements:

Enhance the app's reporting capabilities by providing more advanced data visualization options, such as interactive charts and graphs. This makes it easier for teachers and administrators to interpret attendance data and identify patterns or trends more effectively.

Student Engagement Features:

Introduce features to enhance student engagement and participation, such as in-app communication channels or discussion forums related to attendance or class-related announcements. This promotes student interaction and collaboration within the app.

Integration with Attendance Beacons:

Explore the use of Bluetooth Low Energy (BLE) beacons or RFID technology to automate attendance capture. This eliminates the need for manual attendance marking and provides a seamless experience for both teachers and students.

By incorporating these future enhancements, the Online Attendance Tracking App can continue to evolve, providing an even more robust and comprehensive solution for efficient attendance management in educational institutions.

APPENDICES

APPENDIX 1

SAMPLE CODING

Java

```
packagecom.example.myappla tion;
import static android.content.ContentValues.TAG;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.text.TextUtils;
import android.util.Log;
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.OnCompleteListener;
import com.google.android.gms.tasks.Task;
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.sql.Connection;
import java.sql.ResultSet;
public class MainActivity extends AppCompatActivity {
   //private FirebaseAuth mAuth;

   // DatabaseReference usersRef =
FirebaseDatabase.getInstance().getReferenceFromUrl("https://android-app-98dcd-
default-rtdb.firebaseio.com/");
   DatabaseReference usersRef1 =
FirebaseDatabase.getInstance().getReference("users/staff");
   DatabaseReference usersRef2 =
FirebaseDatabase.getInstance().getReference("users/password");
   DatabaseReference usersRef3 =
FirebaseDatabase.getInstance().getReference("users/student");
   DatabaseReference usersRef4 =
FirebaseDatabase.getInstance().getReference("users/password1");

   DatabaseReference usersRef5 =
FirebaseDatabase.getInstance().getReference("users/admin");

   DatabaseReference usersRef6 =
FirebaseDatabase.getInstance().getReference("users/password2");
   String otpCode = String.valueOf((int) (Math.random() * 900000) + 100000);

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);



      EditText USERNAME = findViewById(R.id.Idno);
      EditText Password = findViewById(R.id.passWord);
      EditText Regno = findViewById(R.id.Regno1);
```

```
EditText Dob = findViewById(R.id.dob1);
EditText Regno1 = findViewById(R.id.Idno1);
EditText Dob1 = findViewById(R.id.passWord1);

Button button = findViewById(R.id.button1);
Button button1 = findViewById(R.id.button2);
Button button2 = findViewById(R.id.button3);

button.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      String username = USERNAME.getText().toString();
      String userPass = Password.getText().toString();


      Handler handler = new Handler();
      Runnable r = new Runnable() {
         public void run() {


            if (username.isEmpty() && userPass.isEmpty()) {
               Toast.makeText(getApplicationContext(), "please enter your username
and password",
                     Toast.LENGTH_LONG).show();
            } else {
               usersRef1.addListenerForSingleValueEvent(new ValueEventListener() {

                  @Override
                  public void onDataChange(@NonNull DataSnapshot snapshot) {

                     for (DataSnapshot userSnapshot : snapshot.getChildren()) {

                        String storedEmail = userSnapshot.getValue(String.class);
                        String key1 = userSnapshot.getKey();
```

```
                    Log.d(TAG, "Value is:" + storedEmail);

                    if (storedEmail.equals(username)) {
                      Toast.makeText(getApplicationContext(), "username is
correct",
                          Toast.LENGTH_LONG).show();
                      usersRef2.addValueEventListener(new ValueEventListener() {
                        @Override
                        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
                          for (DataSnapshot userSnapshot :
dataSnapshot.getChildren()) {
                            String password = userSnapshot.getValue(String.class);
                            String key2 = userSnapshot.getKey();
                            Log.d(TAG, "Value is:" + password);
                            System.out.println(userPass);
                            // System.out.println(password);
                            if (password.equals(userPass)) {
                              Toast.makeText(getApplicationContext(), "password is
correct",
                                  Toast.LENGTH_LONG).show();
                              if (key1.equals(key2)) {
                                //Toast.makeText(getApplicationContext(), "login
successful",
                                //  Toast.LENGTH_LONG).show();
                                Toast.makeText(getApplicationContext(), "login
successful",
                                    Toast.LENGTH_LONG).show();
                                Intent i = new Intent(MainActivity.this,
MainActivity3.class);
                                i.putExtra("name",username);
                                startActivity(i);
```

```
                                } else {
                                    Toast.makeText(getApplicationContext(), "keys are
not equal",
                                            Toast.LENGTH_LONG).show();
                                }

                            } else {
                                Toast.makeText(getApplicationContext(), "invalid
password",
                                        Toast.LENGTH_LONG).show();

                            }
                        }
                    }

                    @Override
                    public void onCancelled(@NonNull DatabaseError
databaseError) {
                        // Handle error
                    }
                });


            } else {
                Toast.makeText(getApplicationContext(), "user not found",
                        Toast.LENGTH_LONG).show();
            }
        }


    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
        Log.e(TAG, "onCancelled", error.toException());
```

```
                    }
                });


            }

        }
        /*usersRef.addListenerForSingleValueEvent(new ValueEventListener() {

            String username = USERNAME.getText().toString();
            String userPass = Password.getText().toString();
            public void onDataChange(DataSnapshot dataSnapshot) {
               for (DataSnapshot userSnapshot: dataSnapshot.getChildren()) {
                  String storedEmail =
userSnapshot.child("users").child("staff").child("name").getValue(String.class);
                  String password =
userSnapshot.child("users").child("staff").child("password").getValue(String.class);
                  if (username.equals(storedEmail) && userPass.equals(password)) {
                     Toast.makeText(getApplicationContext(), "login successful!",
                           Toast.LENGTH_LONG).show();

                  }
               }
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {
               Log.e(TAG, "onCancelled", databaseError.toException());
            }*/
        };
        handler.postDelayed(r, 1000);
    }

});
```

```
button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String username = Regno.getText().toString();
        String userPass = Dob.getText().toString();


        Handler handler = new Handler();
        Runnable r = new Runnable() {
            public void run() {


                if (username.isEmpty() && userPass.isEmpty()) {
                    Toast.makeText(getApplicationContext(), "please enter your username
and password",
                            Toast.LENGTH_LONG).show();
                } else {
                    usersRef3.addListenerForSingleValueEvent(new ValueEventListener() {

                        @Override
                        public void onDataChange(@NonNull DataSnapshot snapshot) {

                            for (DataSnapshot userSnapshot : snapshot.getChildren()) {

                                String storedEmail = userSnapshot.getValue(String.class);
                                String key1 = userSnapshot.getKey();



                                Log.d(TAG, "Value is:" + storedEmail);

                                if (storedEmail.equals(username)) {
                                    Toast.makeText(getApplicationContext(), "username is
correct",
                                            Toast.LENGTH_LONG).show();
```

```
                usersRef4.addValueEventListener(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
                        for (DataSnapshot userSnapshot :
dataSnapshot.getChildren()) {
                            String password = userSnapshot.getValue(String.class);
                            String key2 = userSnapshot.getKey();
                            Log.d(TAG, "Value is:" + password);
                            System.out.println(userPass);
                            // System.out.println(password);
                            if (password.equals(userPass)) {
                                Toast.makeText(getApplicationContext(), "password is
correct",
                                        Toast.LENGTH_LONG).show();
                                if (key1.equals(key2)) {
                                    //Toast.makeText(getApplicationContext(), "login
successful",
                                    //  Toast.LENGTH_LONG).show();
                                    Toast.makeText(getApplicationContext(), "login
successful",
                                            Toast.LENGTH_LONG).show();
                                    Intent i = new Intent(MainActivity.this,
MainActivity6.class);
                                    i.putExtra("name",username);
                                    startActivity(i);


                                } else {
                                    Toast.makeText(getApplicationContext(), "keys are
not equal",
                                            Toast.LENGTH_LONG).show();
                                }

                            } else {
```

```
                                    Toast.makeText(getApplicationContext(), "invalid
password",
                                            Toast.LENGTH_LONG).show();

                                }
                            }
                        }

                        @Override
                        public void onCancelled(@NonNull DatabaseError
databaseError) {
                            // Handle error
                        }
                    });


                } else {
                    Toast.makeText(getApplicationContext(), "user not found",
                            Toast.LENGTH_LONG).show();
                }
            }


        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            Log.e(TAG, "onCancelled", error.toException());
        }
    });


}

}
```

```
        /*usersRef.addListenerForSingleValueEvent(new ValueEventListener() {

            String username = USERNAME.getText().toString();
            String userPass = Password.getText().toString();
            public void onDataChange(DataSnapshot dataSnapshot) {
              for (DataSnapshot userSnapshot: dataSnapshot.getChildren()) {
                String storedEmail =
userSnapshot.child("users").child("staff").child("name").getValue(String.class);
                String password =
userSnapshot.child("users").child("staff").child("password").getValue(String.class);
                if (username.equals(storedEmail) && userPass.equals(password)) {
                  Toast.makeText(getApplicationContext(), "login successful!",
                      Toast.LENGTH_LONG).show();

                }
              }
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {
              Log.e(TAG, "onCancelled", databaseError.toException());
            }*/
          };
          handler.postDelayed(r, 1000);
        }
      });

      button2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
          String username = Regno1.getText().toString();
          String userPass = Dob1.getText().toString();


          Handler handler = new Handler();
```

```
        Runnable r = new Runnable() {
            public void run() {


                if (username.isEmpty() && userPass.isEmpty()) {
                    Toast.makeText(getApplicationContext(), "please enter your username
and password",
                            Toast.LENGTH_LONG).show();
                } else {
                    usersRef5.addListenerForSingleValueEvent(new ValueEventListener() {

                        @Override
                        public void onDataChange(@NonNull DataSnapshot snapshot) {

                            for (DataSnapshot userSnapshot : snapshot.getChildren()) {

                                String storedEmail = userSnapshot.getValue(String.class);
                                String key1 = userSnapshot.getKey();



                                Log.d(TAG, "Value is:" + storedEmail);

                                if (storedEmail.equals(username)) {
                                    Toast.makeText(getApplicationContext(), "username is
correct",
                                            Toast.LENGTH_LONG).show();
                                    usersRef6.addValueEventListener(new ValueEventListener() {
                                        @Override
                                        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
                                            for (DataSnapshot userSnapshot :
dataSnapshot.getChildren()) {
                                                String password = userSnapshot.getValue(String.class);
                                                String key2 = userSnapshot.getKey();
                                                Log.d(TAG, "Value is:" + password);
```

```
                    System.out.println(userPass);
                    // System.out.println(password);
                    if (password.equals(userPass)) {
                      Toast.makeText(getApplicationContext(), "password is
correct",
                          Toast.LENGTH_LONG).show();
                      if (key1.equals(key2)) {
                        //Toast.makeText(getApplicationContext(), "login
successful",
                        //  Toast.LENGTH_LONG).show();
                        Toast.makeText(getApplicationContext(), "login
successful",
                            Toast.LENGTH_LONG).show();
                        Intent i = new Intent(MainActivity.this,
MainActivity4.class);
                        startActivity(i);


                      } else {
                        Toast.makeText(getApplicationContext(), "keys are
not equal",
                            Toast.LENGTH_LONG).show();
                      }

                    } else {
                      Toast.makeText(getApplicationContext(), "invalid
password",
                          Toast.LENGTH_LONG).show();

                    }
                  }
                }                                @Override
                public void onCancelled(@NonNull DatabaseError
databaseError) {
                  // Handle error
```

```
                    }
                });


            } else {
                Toast.makeText(getApplicationContext(), "user not found",
                        Toast.LENGTH_LONG).show();
            }
        }


    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
        Log.e(TAG, "onCancelled", error.toException());
    }
});


}

}
/*usersRef.addListenerForSingleValueEvent(new ValueEventListener() {

    String username = USERNAME.getText().toString();
    String userPass = Password.getText().toString();
    public void onDataChange(DataSnapshot dataSnapshot) {
        for (DataSnapshot userSnapshot: dataSnapshot.getChildren()) {
            String storedEmail =
userSnapshot.child("users").child("staff").child("name").getValue(String.class);
            String password =
userSnapshot.child("users").child("staff").child("password").getValue(String.class);
            if (username.equals(storedEmail) && userPass.equals(password)) {
                Toast.makeText(getApplicationContext(), "login successful!",
```

```
                        Toast.LENGTH_LONG).show();

                    }
                }
            }
            @Override
            public void onCancelled(DatabaseError databaseError) {
                Log.e(TAG, "onCancelled", databaseError.toException());
            }*/
        };
        handler.postDelayed(r, 1000);
      }
    });
  }
}
```

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"
   android:background="@drawable/image1"

   tools:context=".MainActivity">

   <TextView
      android:id="@+id/textView"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_alignParentStart="true"
      android:layout_alignParentTop="true"
      android:layout_alignParentEnd="true"
      android:layout_centerHorizontal="true"
      android:layout_marginStart="5dp"
      android:layout_marginTop="12dp"
      android:layout_marginEnd="5dp"
      android:background="#2FB5D0"
      android:text="MNMJEC online attendance register"
      android:textAlignment="center"
      android:textColor="#0E0E0D"
      android:textSize="25dp"
      tools:ignore="TextSizeCheck" />

   <ScrollView
      android:layout_width="410dp"
      android:layout_height="683dp"
      android:layout_alignParentStart="true"
      android:layout_alignParentTop="true"
      android:layout_marginStart="1dp"
```

```
android:layout_marginTop="48dp">

<RelativeLayout
   android:layout_width="match_parent"
   android:layout_height="1200dp"
   android:layout_alignParentEnd="true"
   android:layout_alignParentBottom="true"
   android:layout_marginEnd="0dp"
   android:layout_marginBottom="0dp">


   <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_alignParentStart="true"
      android:layout_alignParentTop="true"
      android:layout_alignParentEnd="true"
      android:layout_marginTop="54dp"
      android:layout_marginStart="5dp"
      android:layout_marginEnd="5dp"
      android:text="STAFF LOGIN"
      android:textAlignment="center"
      android:textColor="@color/black"
      android:textSize="25dp"
      android:textStyle="bold" />

   <TextView
      android:layout_width="113dp"
      android:layout_height="36dp"
      android:layout_alignParentStart="true"
      android:layout_alignParentTop="true"
      android:layout_alignParentEnd="true"
      android:layout_marginStart="50dp"
      android:layout_marginTop="111dp"
      android:layout_marginEnd="196dp"
```

```
    android:text="ID no"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="14sp" />

<!--   <EditText
    android:id="@+id/enter_c"
    android:layout_width="150dp"
    android:layout_height="41dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_marginTop="194dp"
    android:layout_marginEnd="56dp"
    android:background="@drawable/background_shape"
    android:hint="enter the captcha"
    tools:ignore="TouchTargetSizeCheck,DuplicateSpeakableTextCheck" />-->

<EditText
    android:id="@+id/Idno"
    android:layout_width="172dp"
    android:layout_height="40dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_marginStart="137dp"
    android:layout_marginTop="99dp"
    android:layout_marginEnd="50dp"
    android:background="@drawable/background_shape"
    android:hint="enter your ID no"
    android:textSize="20dp"
    tools:ignore="TouchTargetSizeCheck" />

<TextView
    android:layout_width="107dp"
    android:layout_height="41dp"
```

```
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_marginStart="38dp"
    android:layout_marginTop="171dp"
    android:inputType="text"
    android:text="Password"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="14sp"
    tools:ignore="TouchTargetSizeCheck" />


<EditText
    android:id="@+id/passWord"
    android:layout_width="wrap_content"
    android:layout_height="38dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_marginStart="137dp"
    android:layout_marginTop="165dp"
    android:layout_marginEnd="41dp"
    android:background="@drawable/background_shape"
    android:hint="enter your password"
    android:textSize="20dp"
    tools:ignore="TouchTargetSizeCheck" />

<Button
    android:id="@+id/button1"
    android:layout_width="165dp"
    android:layout_height="68dp"
    android:layout_alignParentStart="false"
    android:layout_alignParentTop="true"
    android:layout_centerInParent="true"
    android:layout_marginTop="225dp"
```

```
    android:backgroundTint="@color/white"
    android:gravity="center"
    android:text="login"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="20dp"
    tools:ignore="DuplicateSpeakableTextCheck,TextSizeCheck" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="false"
    android:layout_alignParentTop="true"
    android:layout_centerInParent="true"
    android:layout_marginStart="5dp"
    android:layout_marginTop="325dp"
    android:layout_marginEnd="5dp"
    android:text="STUDENT LOGIN"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="25dp"
    android:textStyle="bold" />

<TextView
    android:layout_width="113dp"
    android:layout_height="36dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_marginStart="45dp"
    android:layout_marginTop="380dp"
    android:layout_marginEnd="196dp"
    android:text="Reg no"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="14sp" />
```

```
<EditText
   android:id="@+id/Regno1"
   android:layout_width="wrap_content"
   android:layout_height="38dp"
   android:layout_alignParentTop="true"
   android:layout_alignParentEnd="true"
   android:layout_alignParentStart="true"
   android:layout_marginTop="370dp"
   android:layout_marginStart="137dp"
   android:layout_marginEnd="54dp"
   android:background="@drawable/background_shape"
   android:hint="enter your Reg no"
   android:textSize="20dp"
   android:inputType="number"
   tools:ignore="TouchTargetSizeCheck" />

<!--  <TextView
   android:id="@+id/captcha"
   android:layout_width="128dp"
   android:layout_height="wrap_content"
   android:layout_alignParentStart="true"
   android:layout_alignParentTop="true"
   android:layout_centerVertical="true"
   android:layout_marginStart="59dp"
   android:layout_marginTop="820dp"
   android:background="@drawable/background_shape"
   android:backgroundTint="@color/black"
   android:text="hello"
   android:textAlignment="center"
   android:textColor="@color/white"
   android:textSize="25dp" />

<TextView
   android:layout_width="104dp"
```

```
    android:layout_height="5dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_marginStart="68dp"
    android:layout_marginTop="226dp"
    android:background="@color/black" />

<EditText
    android:id="@+id/enter_c1"
    android:layout_width="150dp"
    android:layout_height="41dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_marginTop="501dp"
    android:layout_marginEnd="67dp"
    android:background="@drawable/background_shape"
    android:hint="enter the captcha"
    tools:ignore="TouchTargetSizeCheck" />

<TextView
    android:layout_width="104dp"
    android:layout_height="5dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_marginStart="55dp"
    android:layout_marginTop="531dp"
    android:background="@color/black" />

<TextView
    android:id="@+id/captcha1"
    android:layout_width="128dp"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_centerVertical="true"
```

```
    android:layout_marginStart="41dp"
    android:layout_marginTop="506dp"
    android:background="@drawable/background_shape"
    android:backgroundTint="@color/black"
    android:text="hello"
    android:textAlignment="center"
    android:textColor="@color/white"
    android:textSize="25dp"
    tools:ignore="TextContrastCheck,TextSizeCheck" />


<TextView
    android:layout_width="104dp"
    android:layout_height="5dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_marginStart="56dp"
    android:layout_marginTop="822dp"
    android:background="@color/black" />-->


<TextView
    android:layout_width="107dp"
    android:layout_height="41dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_marginStart="56dp"
    android:layout_marginTop="441dp"
    android:text="DOB"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="14sp"
    tools:ignore="TouchTargetSizeCheck" />

<EditText
```

```
    android:id="@+id/dob1"
    android:layout_width="146dp"
    android:layout_height="38dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentStart="true"
    android:layout_marginTop="426dp"
    android:layout_marginStart="137dp"
    android:layout_marginEnd="56dp"
    android:background="@drawable/background_shape"
    android:hint="enter your dob"
    android:inputType="date"
    android:textSize="20dp"
    tools:ignore="TouchTargetSizeCheck,TextSizeCheck" />

<Button
    android:id="@+id/button2"
    android:layout_width="165dp"
    android:layout_height="68dp"
    android:layout_alignParentStart="false"
    android:layout_alignParentTop="true"
    android:layout_centerInParent="true"
    android:layout_marginTop="486dp"
    android:backgroundTint="@color/white"
    android:gravity="center"
    android:text="login"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="20dp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="false"
    android:layout_alignParentTop="true"
```

```
    android:layout_centerHorizontal="true"
    android:layout_marginStart="5dp"
    android:layout_marginTop="586dp"
    android:layout_marginEnd="5dp"
    android:gravity="center"
    android:text="ADMIN LOGIN"
    android:textColor="@color/black"
    android:textSize="25dp"
    android:textStyle="bold" />


<TextView
    android:layout_width="113dp"
    android:layout_height="36dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_marginStart="30dp"
    android:layout_marginTop="641dp"
    android:text="Usrename"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="14sp" />

<EditText
    android:id="@+id/Idno1"
    android:layout_width="wrap_content"
    android:layout_height="38dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true"
    android:layout_marginStart="137dp"
    android:layout_marginTop="632dp"
    android:layout_marginEnd="49dp"
    android:background="@drawable/background_shape"
    android:hint="enter your user name"
```

```
    android:textSize="20dp" />

<!-- <EditText
    android:id="@+id/enter_c"
    android:layout_width="150dp"
    android:layout_height="41dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_marginTop="818dp"
    android:layout_marginEnd="51dp"
    android:background="@drawable/background_shape"
    android:hint="enter the captcha" />-->

<TextView
    android:layout_width="107dp"
    android:layout_height="41dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_marginStart="31dp"
    android:layout_marginTop="696dp"
    android:inputType="text"
    android:text="Password"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="14sp" />

<EditText
    android:id="@+id/passWord1"
    android:layout_width="wrap_content"
    android:layout_height="38dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true"
    android:layout_marginTop="688dp"
    android:layout_marginStart="137dp"
```

```
            android:layout_marginEnd="40dp"
            android:background="@drawable/background_shape"
            android:hint="enter your password"
            android:textSize="20dp" />

        <Button
            android:id="@+id/button3"
            android:layout_width="165dp"
            android:layout_height="68dp"
            android:layout_alignParentStart="true"
            android:layout_alignParentTop="true"
            android:layout_centerInParent="true"
            android:layout_marginStart="93dp"
            android:layout_marginTop="753dp"
            android:backgroundTint="@color/white"
            android:gravity="center"
            android:text="login"
            android:textAlignment="center"
            android:textColor="@color/black"
            android:textSize="20dp" />

    </RelativeLayout>
  </ScrollView>


</RelativeLayout>
```

APPENDIX 2

SCREEN SHOTS

Figure A INDEX PAGE

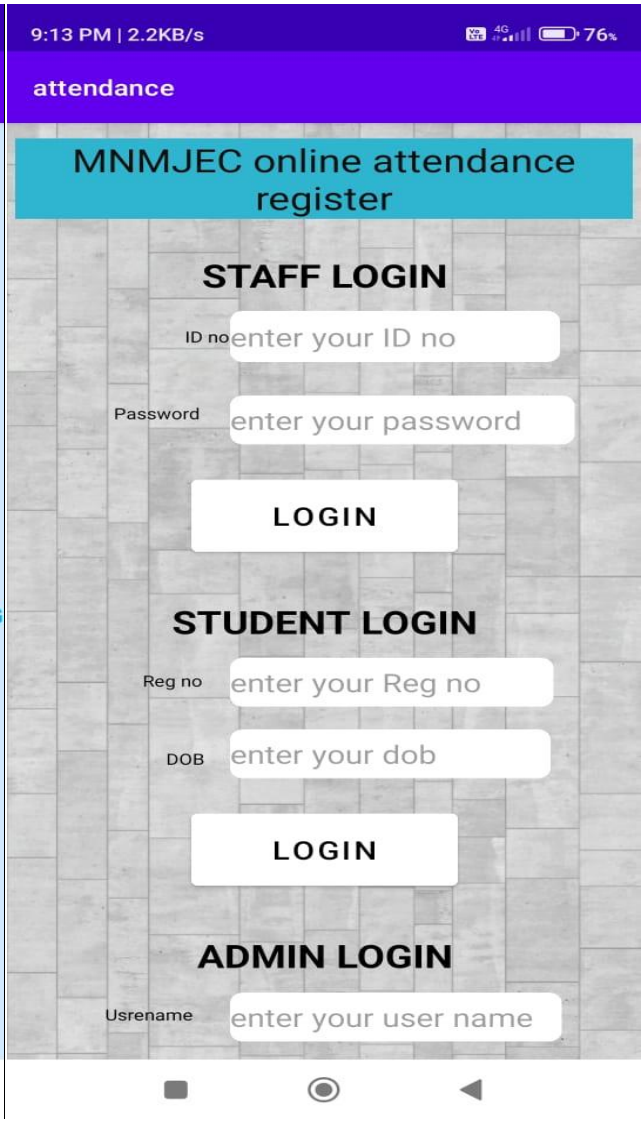
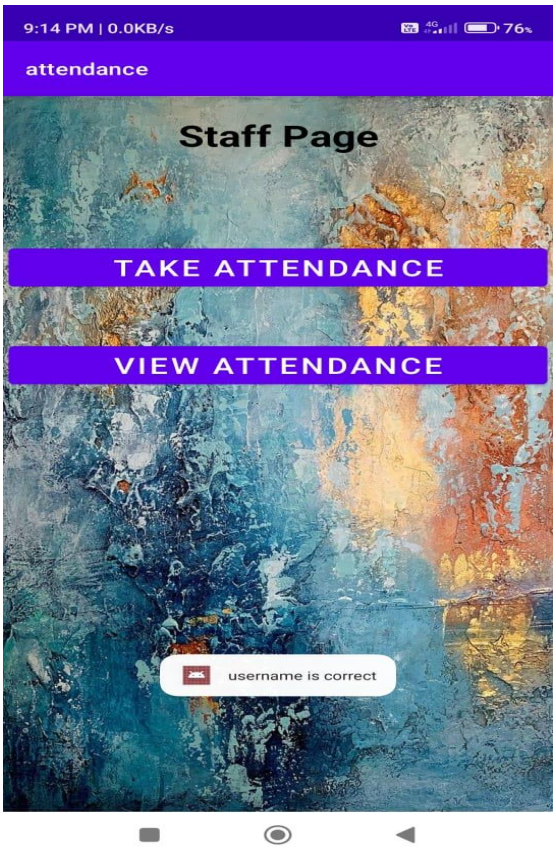
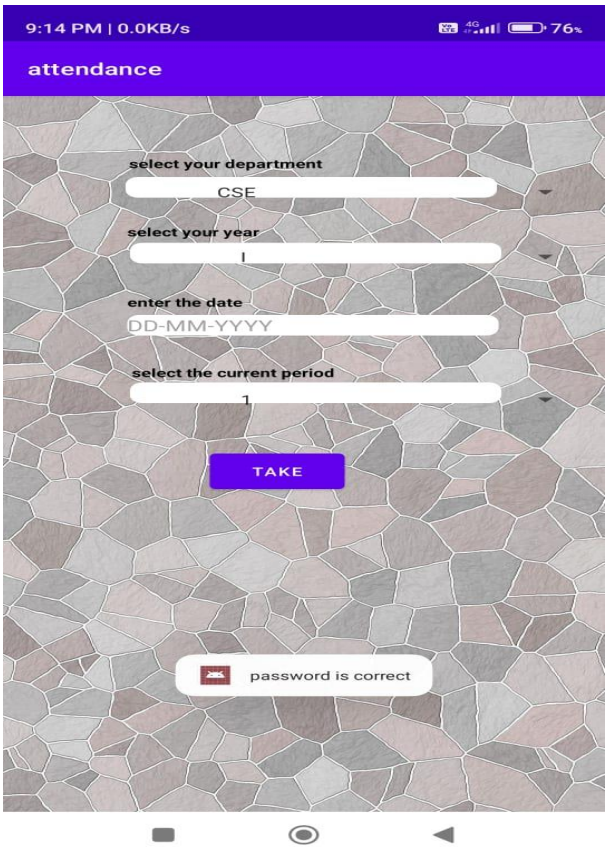
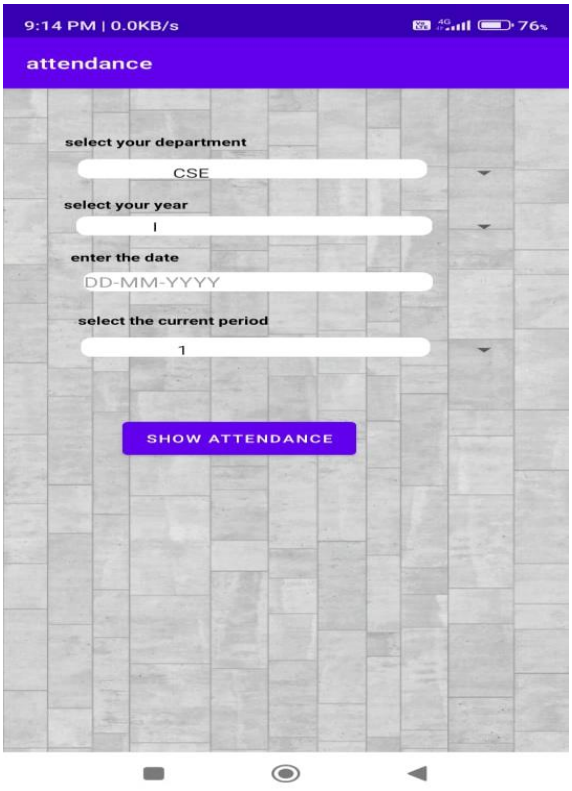


Figure B HOME PAGE







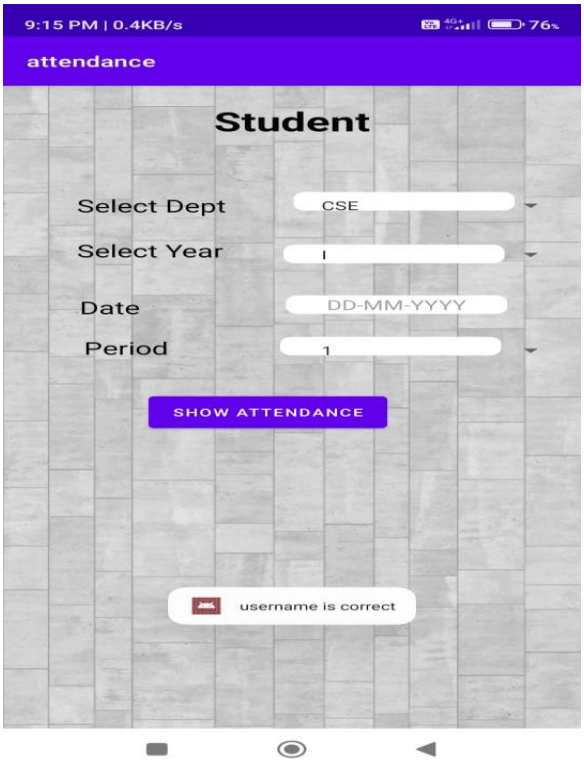


Figure C VIEW &TAKE ATTENDANCE

Figure D STUDENT LOGIN

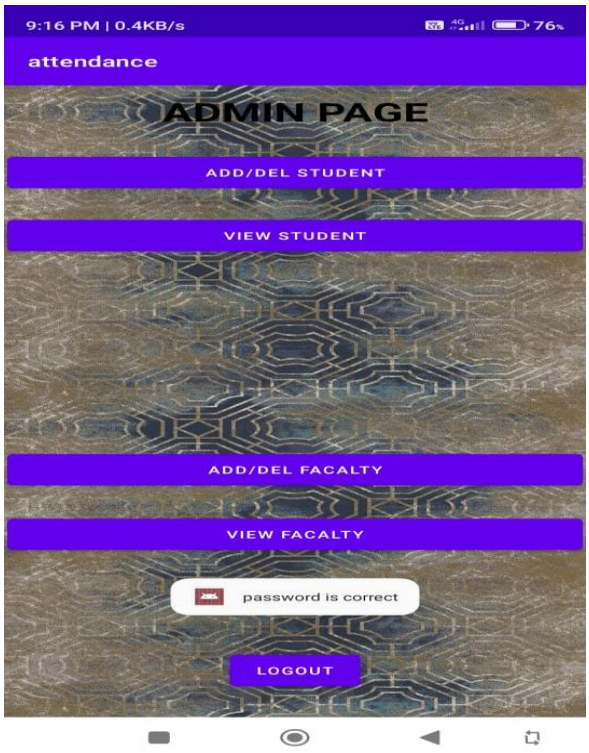


Figure E ADMIN LOGIN

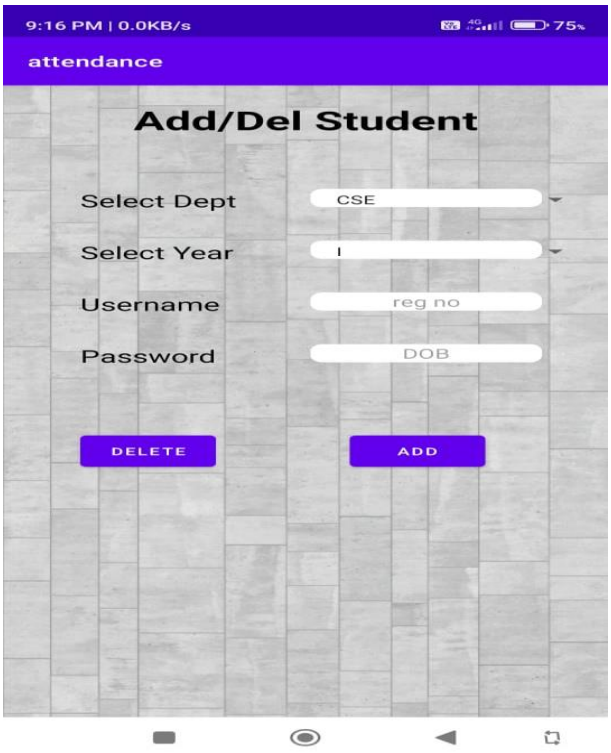
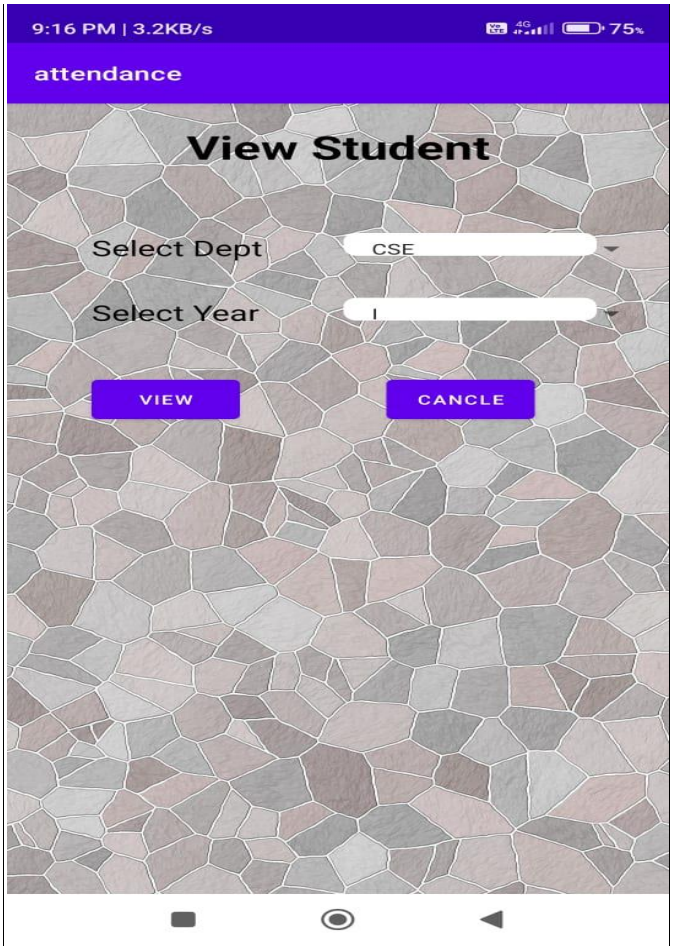
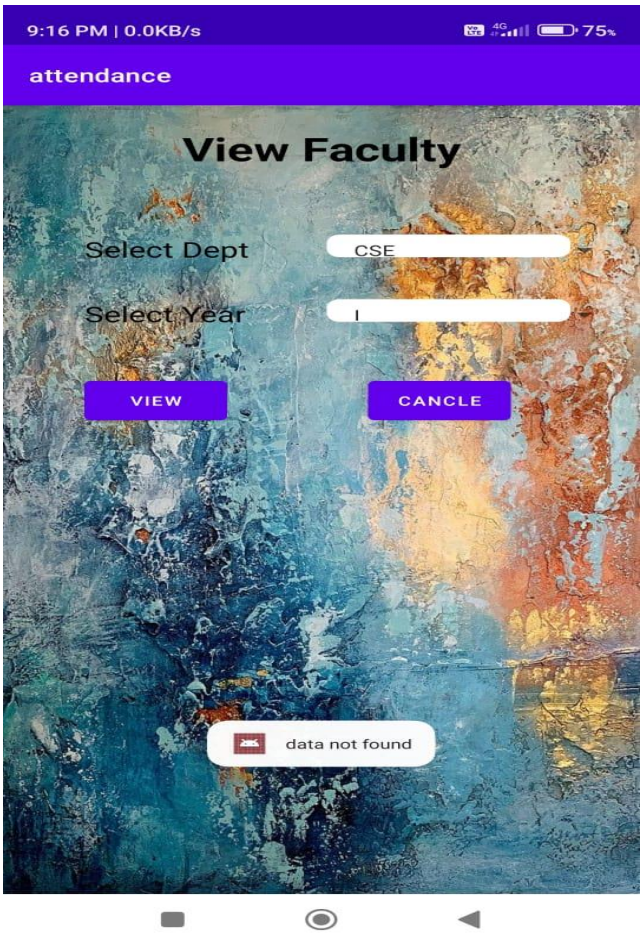


Figure F ADD/DEL STUDEND





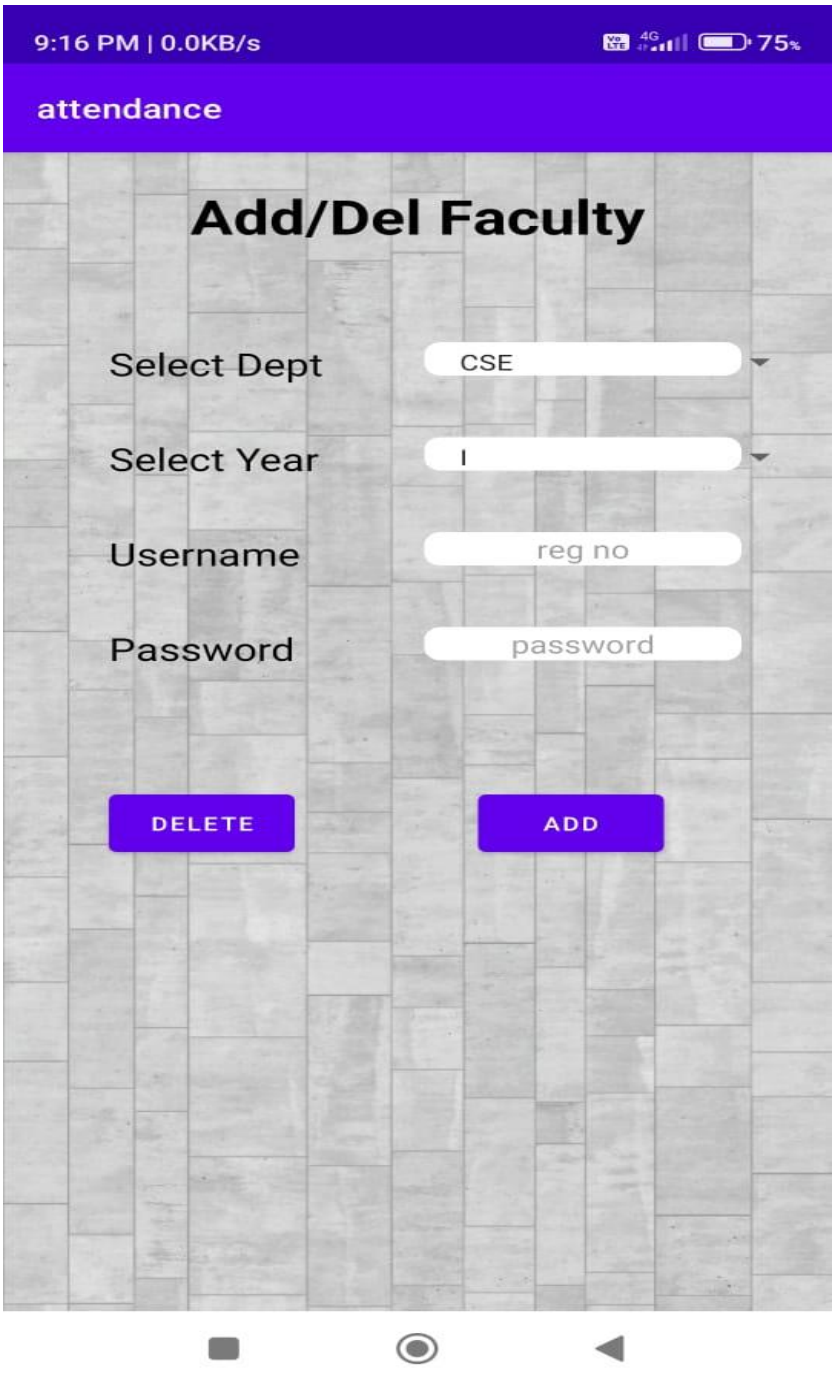


Figure G ADD/DEL FACULTY

REFERENCES

[1] M. A. Gerosa and C. Lucena, “The Development and Application of Distance Learning Courses on the Internet RDL-Reuse Representation View project An Agent-based Framework for the Internet of Things View project,” Taylor Fr., vol. 17, no. 1, pp. 23–38, 2002, doi: 10.1080/02680510120110157.

[2] C. Series, “Development of physics learning e-module based on local culture wisdom in Pontianak , West Kalimantan Development of physics learning e-module based on local culture wisdom in Pontianak , West Kalimantan,” 2019, doi: 10.1088/1742-6596/1381/1/012045.

[3] T. Partovi and M. R. Razavi, “The effect of game-based learning on academic achievement motivation of elementary school students,” Learn. Motiv., vol. 68, no. August, p. 101592, 2019, doi: 10.1016/j.lmot.2019.101592.

[4] A. Hussein and M. Hussain, “Social-media based assessment of academic programs,” Stud. Educ. Eval., vol. 62, no. June, pp. 149–157, 2019, doi: 10.1016/j.stueduc.2019.06.003.

COs, PO Mapping, PSO Mapping

Course Code &Name : CS8611 Mini project Lab
REGULATION: R2017 YEAR/SEM:III/VI

COURSE OUTCOMES

CS8611.1	Identify the problem by applying acquired knowledge.
CS8611.2	Analyze and categorize executable project modules after considering risks.
CS8611.3	Choose efficient tools for designing project modules.
CS8611.4	Combine all the modules through effective team work after efficient testing.
CS8611.5	Elaborate the completed task and compile the project report.

CORRELATION LEVELS

Substantial/ High	3
Moderate/ Medium	2
Slight/ Low	1
No correlation	

CO – PSO CORRELATION LEVEL MATRIX

COs	PSO1	PSO2
CS8611.1	3	1
CS8611.2		2
CS8611.3		3
CS8611.4		3
CS8611.5		

CO-PO CORRELATION LEVEL MATRIX

COs	POs											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CS8611.1	3	3		1			1	2	3			

CS8611.2		3	2	3		3	1		3	3		2
CS8611.3			3	3	3			3	3	3	1	2
CS8611.4							3	3	3	3	2	3
CS8611.5									3	3		3