VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018

A PROJECT REPORT
ON

HOSTELMATE

BY

SUGHOSHA S V 4SF20CS158

VINAYAK B KALMANE 4SF20CS175

In the partial fulfillment of the requirement for VI Sem. B. E. (CSE)

MOBILE APPLICATION DEVELOPMENT LABORATORY WITH MINIPROJECT (18CSMP68)

Under the guidance of

Mr.Harisha
Assistant Professor, Dept. of CSE

Department of Computer Science & Engineering
SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
Adyar, Mangaluru-575007
2022-23

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
(Affiliated to Visvesvaraya Technological University, BELAGAVI)
Adyar, Mangaluru – 575007

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled “HOSTELMATE” is submitted in partial fulfillment for the requirement of VI sem. B. E. (Computer Science & Engineering), “MOBILE APPLICATION LABORATORY WITH MINI PROJECT” during the year 2022 – 23 is a result of bonafide work carried out by

SUGHOSHA S V	**4SF20CS158**
VINAYAK B KALMANE	**4SF20CS175**

……………………………...
Mr.Harisha
Assistant. Prof. Dept. of CS&E
SCEM, Mangaluru

……………………………….
Dr. Nagesh H R
HOD, Dept. of CS&E
SCEM, Mangaluru

Signature of the Examiners

1. …………………………..

2. ………………………….

ABSTRACT

The "HostelMate" Mobile Application Development Project focuses on creating a user-friendly and feature-rich mobile application specifically designed for hostel students. HostelMate aims to provide a convenient and efficient platform for students to manage their daily activities. By leveraging the power of mobile technology, HostelMate aims to enhance the overall hostel experience.

HostelMate will offer a range of key features to cater to the needs of hostel students. These features include a login and signup functionality, allowing students to create accounts and access personalized features. Students will be able to apply for leave directly through the application, submitting leave requests with duration and reasons, and tracking the status of their applications. HostelMate will also provide a convenient platform for students to submit complaints or maintenance requests, attaching relevant details and receiving updates on the progress of their complaints.

HostelMate, with its user-friendly interface and essential features, aims to simplify administrative processes and enhance the overall hostel experience for students. By providing a centralized platform for tasks such as leave applications, complaint submissions, fee checking, and viewing notices, HostelMate streamlines communication and empowers students to efficiently manage their hostel-related activities. With its focus on convenience and seamless interaction, HostelMate seeks to foster a positive and engaging environment for hostel students, making their stay more comfortable and enjoyable.

ACKNOWLEDGEMENT

Before we get in-depth with the project, we want to include few expressions of appreciation for the people who has been a part of this project from its inception. The written work of this project has been one of the huge academic challenges we have faced and withoutthe help, patience and guidance of the people involved, this assignment would not have been completed satisfactorily.

It gives us immense pleasure in presenting this project report on "**HOSTELMATE**". It has been our privilege to have a project guide who had assisted us from the commencement of this project. The success of this project is a sheer diligent work, and determination put in by us with the help of our project guide.

We hereby take this chance to include a special note of much obliged for **Mr. Harisha,** Assistant Professor, and Department of Computer Science who guided us in our project.

We are additionally grateful to **Dr.Nagesh H R** Professor & Head of the Department, Computer Science and Engineering for furnishing us with the correct academic atmosphere in the department, whose encouragement and support made our entire undertaking appreciable.

We are extremely thankful to our beloved Principal **Dr. Rajesha S** for encouraging usto come up with new ideas and to express them in a systematic manner.

We would also like to thank all our non-teaching staffs who also were very much supportive to us in building this project.

Last but not the least we want to extend our gratitude to our parents for their continuous love and support for which we are always indebted to them and also thank each and every one of those who helped or provided a helping hand in this project to be carried out.

SUGHOSHA S V (4SF20CS158)
VINAYAK B (4SF20CS175)

PAGE INDEX

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW PROJECT

Welcome to HostelMate App! We're here to enhance your hostel experience with our comprehensive mobile application. From managing your daily routine to staying updated with hostel-related information, HostelMate App is designed to make student life easier. With features like apply for leave, View the hostel fees, Submit issues, View Notices. We offer a complete solution for hostilities.

Should any concerns or complaints arise, our grievance box feature allows you to submit them conveniently. Track the status of your complaints and provide valuable feedback on the resolution process. At HostelMate, we aim to provide a user-friendly platform that simplifies your daily routines, keeps you informed, and ensures your concerns are addressed.

Functionalities of the app:

- Student View
  - Sign-in Page.
  - Sign-up Page
  - Apply for Leave.
  - Report Issues.
  - View Hostel Notice.
  - Leave Status.
  - User Details.

- Warden View
  - View Issues.
  - Upload Notice
  - Apply/Reject Leaves

.

1.2 INTRODUCTION TO JAVA

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers write once, run anywhere. Java is a popular and widely used language throughout the world. Java programming language was originally developed by Sun Microsystems which was initiated by James Gosling and released in 1995 as core component of Sun Microsystems' Java platform (Java 1.0 [J2SE]). After the development of C, C++, Java has come into evolution by addressing their drawbacks . Java is owned by Oracle, and more than billion devices run Java. Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.). It is open-source secure, fast, free and powerful. As Java is close to C++ and C#, it makes it easy for programmers to switch to Java. The language is also easy to learn, understand and implement. Java is used in various kinds of applications like Web, Desktop, Mobile, and Big Data. The language is flexible enough to maintain code complexity, test, implementation, integration and support. Apart from these, there are other key features which make Java more special. It is object-oriented programming language, one of the important hierarchies in the programming languages which is used to implement real time applications, it provides for code reusability.

1.3 INTRODUCTION TO ANDROID STUDIO

Android Studio is the official Integrated Development Environment (IDE) for android application development. Android Studio provides more features that enhance our productivity while building Android apps. The IDE is designed very efficiently which makes the developer’s job easy. It also supports the IntelliJ IDE. Features of Android Studio

- It has a flexible Gradle-based build system.
- It has a fast and feature-rich emulator for app testing.
- Android Studio has a consolidated environment where we can develop for all Android devices
- Apply changes to the resource code of our running app without restarting the app.
- Android Studio provides extensive testing tools and frameworks.
- It provides build-in supports for Google Cloud Platform. It makes it easy to integrate Google Cloud Messaging and App Engine.

APK (Android Package) file, in which we can run that APK file in any device and use the application.

ANDROID SOFTWARE DEVELOPMENT KIT (SDK):

The Android SDK is a collection of software development tools and libraries required to develop Android applications. The Android SDK comprises all the tools necessary to code programs from scratch and even test them. These tools provide a smooth flow of the development process from developing and debugging, through to packaging. The Android SDK is compatible with Windows, Mac OS, and Linux, so you can develop on any of those platforms.

ANDROID DEBUG BRIDGE (ADB):

ADB is something that many Android enthusiasts use, but its full potential is often overlooked. ADB stands for “Android Debug Bridge,” and it is a command line tool that is used to communicate with a smartphone, tablet, smart-watch, set-top box, or any other device that can run the Android operating system (even an emulator). Specific commands are built into the ADB binary and while some of them work on their own, most are commands we send to the connected device.

GRADLE BUILD:

The Android Studio build system is based on Gradle, and the Android Gradle plugin adds several features that are specific to building Android apps. Although the Android plugin is typically updated in lock-step with Android Studio, the plugin (and the rest of the Gradle system) can run independent of Android Studio and be updated separately. It is basically an automated build system which is used to automate the various phases involved in designing an application that includes design, development, test, debug, and publish. We need to configure the project and modules by mentioning all the supported jar files, SDK’s, version name, level, compiled SDK version, build tools version.

ANDROID DEVICE MONITOR:

Android Device Monitor is a stand-alone tool that provides a graphical user interface for several Android application debugging and analysis tools. The Monitor tool does not require installation of a integrated development environment. If we want to access all the hidden files that are generated when we run the application, we can use the monitor. We can select any project and explore the files that are related to that project.

SDK MANAGER:

The Android SDK separates tools, platforms, and other components into packages you can download using the SDK Manager. It is one of the main tools to maintain the updates of all the installed components required to run the project. It also notifies us when the project is not compatible with device or any other compatibility issues and to download any component that is required.

AVD MANAGER:

The AVD Manager provides a graphical user interface in which you can create and manage Android Virtual Devices (AVDs), which are required by the Android Emulator. Each AVD is an emulator configuration that simulates a physical Android device. This makes it possible to run and test your app in a variety of configurations that simulate different physical Android devices.

1.4 INTRODUCTION TO FIREBASE

Firebase is a Backend-as-a-Service (BaaS) which started as a YC11 start-up. It grew up into a next-generation app-development platform on Google Cloud Platform. Firebase is a real-time database that allows storing a list of objects in the form of a tree. We can synchronize data between different devices. It is a software which allows developers to develop Android, IOS, and Web apps. For reporting and fixing app crashes, tracking analytics, creating marketing and product experiments, firebase provides several tools. Firebase has three main services, i.e., a real-time database, user authentication, and hosting. Firebase evolved from Envolve. Envolve is a prior start-up founded by James Tamplin and Andrew Lee in 2011. Firebase Real-time Database was the first product of firebase. It is an API which syncs application data across Android, iOS, and Web devices. It gets stored on Firebase's cloud. Then the firebase real-time database helps the developers to build real-time, collaborative applications. Firebase manages real-time data in the database. So, it easily and quickly exchanges the data to and from the database. Hence, for developing mobile apps such as live streaming, chat messaging, etc., Firebase allows syncing real-time data across all devices - iOS, Android, and Web - without refreshing the screen.

CHAPTER 2

REQUIREMENT ANALYSIS

Requirement specification is focused specially on functioning of the system, functions to be carried out and performance levels to be obtained and corresponding interfaces to be established.

This report gives the description of the roles of users, the functional overviews of the project, input and output characteristics and also the hardware and software for the project.

2.1 Hardware Requirements

The minimum/recommended hardware configuration required for developing the proposed software is given below:

- Processor : Intel(R) Core(TM) i5-8265U CPU @1.60GHz 1.80GHz2GB RAM.
- Processor Speed : 1.80GHz
- RAM – 8 GB or above.
- Monitor resolution - 1280 x 800 minimum screen resolution

2.2 Software Requirements

- Operating System: Windows / LINUX / any operating system that supports the software.
- Language used: Java
- Software used: Android Studio
- Database: Firebase

CHAPTER 3

SYSTEM DESIGN

3.1 Basic Layout

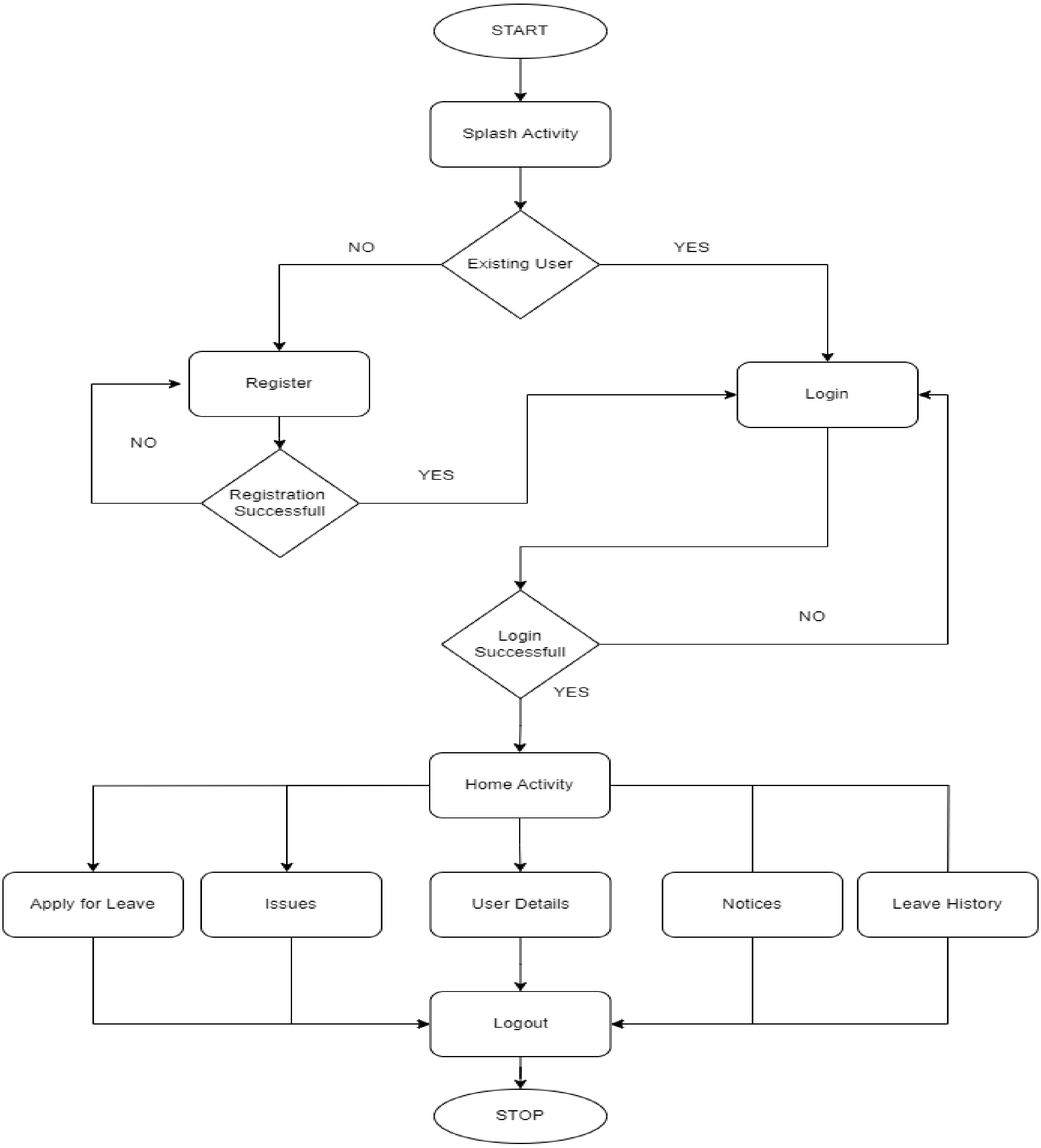
- Splash Screen
- Login Page
- Student Home Page
- Apply for Leave Page
- Notice Page
- Report Issues Page
- User Details Page
- Leave History Page

The above-shown layout is a basic idea of the app design. The app starts with the splash screen i.e., the page displayed when a user opens the app which has the logo of our college. The Login page pops after the splash screen. If the user has already registered, he can simply Login else can register by clicking on sign up. After login user enters the homepage and in homepage there are mainly four sections such as Apply for Leave, Viewing the hostel fees, Complaint page, Notice Page. The whole project runs on the Firebase server, and the data is stored in the firebase real-time database and firestore database.

3.2 Methodology

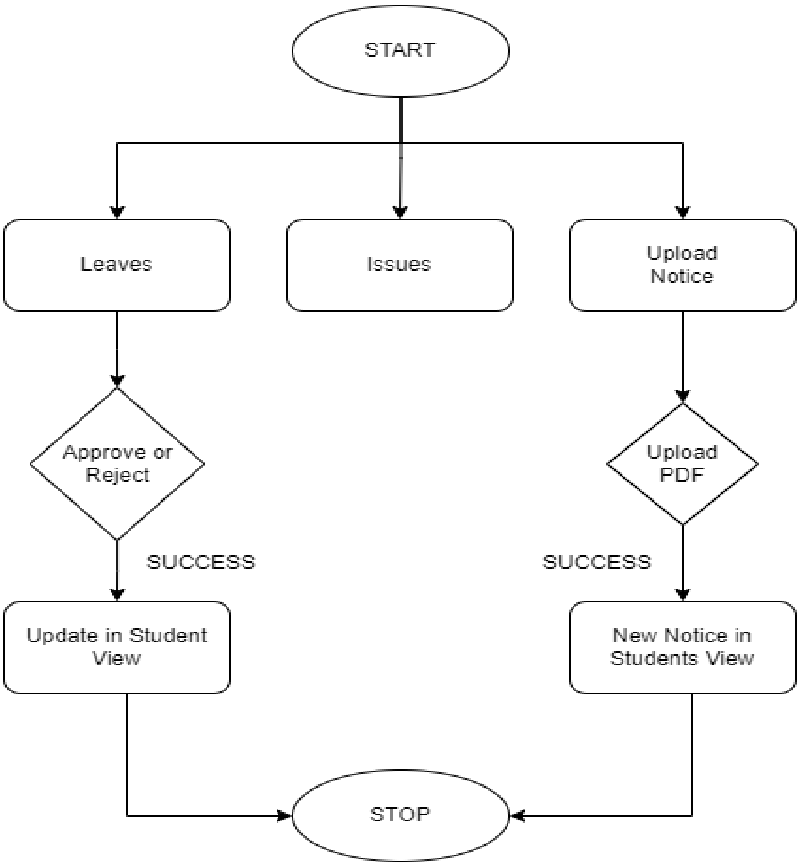
Flowchart is a common type of chart that represents an algorithm or process showing the steps as boxes of various kinds, their order by connecting these with arrows.

Student View :



3.2.1 Student View

Warden View :



3.2.2 Warden View

CHAPTER 4

IMPLEMENTATION

4.1 Algorithm

Warden Page:

```
Step 1: Start

Step 2: Home Screen

Step 3: if (tap on Leaves)
            show leave request list and Approve or Reject the leaves

Step 5:  if (tap on Issues)
            show issues updated reported by the students

Step 6: if (tap on Notice)
            allow the user to upload the PDF notice.
```

Student Page:

```
Step 1: Start (Splash Screen)

Step 2: Home Screen
        if (new user)
            Create Account
    else
            Home Page

Step 3: if (tap on Apply for Leave)
            Shows the Leave Application

Step 4: if (tap on Issues)
            Shows the Issues page which allows user to complaint issues

Step 5: if (tap on Notices)
            Shows notices uploaded by the warden

Step 6: if (tap on Leave Status)
            shows the status of the Leave application
```

```
Step 7: if (tap on MyAccount)
        shows the user details and allows for resetting password

Step 8: if (tap on Logout)
    exit
```

4.2 Code Snippets

Student View:

4.2.1 Signup Code

```
public class SignUp extends AppCompatActivity {
    private Button btn1;
    private Button signup;
    EditText name_var, usn_var, password_var;
    public static final String EXTRA_NAME = "com.example.sahayadriapp.extra.NAME";
    String USN_PATTERN = "^4SF[0-9]{2}[A-Za-z]{2}[0-9]{3}$";
    String PASSWORD_PATTERN = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d).{6,}$";
    FirebaseDatabase firebaseDatabase;
    DatabaseReference reference;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_up);
        //getSupportActionBar().hide();
        Button signup = findViewById(R.id.button7);
        name_var = findViewById(R.id.editname);
        usn_var = findViewById(R.id.editTextTextPersonName5);
        password_var = findViewById(R.id.editTextTextPassword);
        signup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String name = name_var.getText().toString();
                String usn = usn_var.getText().toString();
                String password = password_var.getText().toString();
                if (!name.isEmpty()) {
                    name_var.setError((null));
                    if (!usn.isEmpty()) {
                        usn_var.setError(null);
                        if (!password.isEmpty()) {
                            password_var.setError(null);
                            if (validateUSN(usn)) {
                                if (validatePassword(password)) {
                                    firebaseDatabase = FirebaseDatabase.getInstance();
                                    reference = firebaseDatabase.getReference("datauser");
                                    String name_s = name_var.getText().toString();
                                    String usn_s = usn_var.getText().toString();
                                    String password_s = password_var.getText().toString();
                                    storingdata storingdatass = new storingdata(name_s, usn_s, password_s);
                                    reference.child(usn_s).setValue(storingdatass);
                                    Toast.makeText(SignUp.this, "Successfull", Toast.LENGTH_SHORT).show();
                                    Intent intent = new Intent(SignUp.this, LogIN.class);
                                    intent.putExtra(EXTRA_NAME, name);
                                    startActivity(intent);
                                    finish();
                                else {
                                    password_var.setError("Enter Correct Password");
                                }
                            }
        else {
                usn_var.setError("Enter Correct USN");
              }
            } else {
              password_var.setError("Please enter password");
            }
          } else {
            usn_var.setError("Please enter USN");
          }
        } else {
          name_var.setError("Please Enter the name");
        }
      }
    });
  }
  private boolean validatePassword(String password) {
    Pattern pattern = Pattern.compile(PASSWORD_PATTERN);
    Matcher matcher = pattern.matcher(password);
    return matcher.matches();
  }
  private boolean validateUSN(String usn) {
    Pattern pattern = Pattern.compile(USN_PATTERN);
    Matcher matcher = pattern.matcher(usn);
    return matcher.matches();
  }
}
```

4.2.2 Login Code

```
public class LogIN extends AppCompatActivity {
  private Button btn1;
  private Button login;
  ProgressDialog progressDialog;
  EditText usn_var, password_var;
  public static final String SHARED_PREFS = "sharedPrefs";
  public static final String EXTRA_USN = "com.example.sahayadriapp.extra.USN";
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.login);
    //getSupportActionBar().hide();
    //checkbox();
    Button btn1 = findViewById(R.id.button2);
    Button login = findViewById(R.id.button7);
    usn_var = findViewById(R.id.editusn);
    password_var = findViewById(R.id.editTextTextPassword);
     btn1.setOnClickListener(new View.OnClickListener() {
       @Override
       public void onClick(View v) {
         Intent intent = new Intent(LogIN.this, SignUp.class);
         startActivity(intent);
         finish();
       }
    });
```

```
login.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
    String usn = usn_var.getText().toString();
    String password = password_var.getText().toString();
    progressDialog = new ProgressDialog(LogIN.this);
    progressDialog.setCancelable(false);
    progressDialog.setMessage("Logging in...");
    progressDialog.show();
    if (!usn.isEmpty()) {
      usn_var.setError((null));
      if (!password.isEmpty()) {
        password_var.setError(null);
        final String usn_data = usn_var.getText().toString();
        final String password_data = password_var.getText().toString();
        FirebaseDatabase firebaseDatabase = FirebaseDatabase.getInstance();
        DatabaseReference databaseReference = firebaseDatabase.getReference("datauser");
        Query check_username = databaseReference.orderByChild("usn").equalTo(usn_data);
        check_username.addListenerForSingleValueEvent(new ValueEventListener() {
          @Override
          public void onDataChange(@NonNull DataSnapshot snapshot) {
            if (snapshot.exists()) {
              usn_var.setError(null);
              String passcheck = snapshot.child(usn_data).child("password").getValue(String.class);
              if (passcheck.equals(password_data)) {
                password_var.setError(null);
                if (progressDialog.isShowing())
                  progressDialog.dismiss();
                Intent intent = new Intent(LogIN.this, AfterLogin.class);
                intent.putExtra(EXTRA_USN, usn);
                startActivity(intent);
                finish();
              } else {
                password_var.setError("Wrong Password");
                progressDialog.dismiss();
              }
            } else {
              usn_var.setError("USN Does not exists");
              progressDialog.dismiss();
            }
          }
          @Override
          public void onCancelled(@NonNull DatabaseError error) {}

             });
      } else {
        password_var.setError("Please enter password");
        progressDialog.dismiss();
      }
    } else {
      usn_var.setError("Please enter USN");
      progressDialog.dismiss();
    }
  }
});
```

4.2.3 HomePage

```
public class AfterLogin extends AppCompatActivity {
  private Button applyleave;
  private Button payfees;
  private Button notices;
  private Button issues;
  private TextView textname;
  private TextView textusn;
  private TextView textbranch;
  BottomNavigationView btm;
  public static final String EXTRA_USN = "com.example.sahayadriapp.extra.USN";
  public static String USN = "com.example.sahyadriapp.USN";
  public static final String FEES_USN = "com.example.sahyadriapp.feeUSN";
  public static final String ISSUE_USN = "com.example.sahyadriapp.issueUSN";
  public static final String USER_USN = "com.example.sahyadriapp.userUSN";
  public static final String USER_NAME = "com.example.sahyadriapp.userNAME";
  public static final String USER_BRANCH = "com.example.sahyadriapp.userbranch";
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_after_login);
    // getSupportActionBar().hide();
    Button applyleave = findViewById(R.id.button3);
    Button payfees = findViewById(R.id.button4);
    Button notices = findViewById(R.id.button6);
    Button issues = findViewById(R.id.button5);
    TextView textname = findViewById(R.id.textView2);
    TextView textusn = findViewById(R.id.textView3);
    TextView textbranch = findViewById(R.id.textView4);
    btm = findViewById(R.id.bottom_navigation);
    Intent intent = getIntent();
    String usn2 = intent.getStringExtra(LogIN.EXTRA_USN);
    textusn.setText(usn2);
   String user_usn = textusn.getText().toString();
    String leave_history_usn = textusn.getText().toString();
  if (usn2.contains("CS") || usn2.contains("cs")) {
      textbranch.setText("BE/CSE");
    }
 else if (usn2.contains("EC") || usn2.contains("ec")) {
      textbranch.setText("BE/ECE");
    } else if (usn2.contains("ME") || usn2.contains("me")) {
      textbranch.setText("BE/ME");
    } else if (usn2.contains("IS") || usn2.contains("is")) {
      textbranch.setText("BE/ISE");
    } else {
      textbranch.setText("MBA");
    }
    DatabaseReference reference = FirebaseDatabase.getInstance().getReference("datauser");
        reference.orderByChild("usn").equalTo(usn2).addListenerForSingleValueEvent(new
    ValueEventListener() {
      @Override
```

```
           public void onDataChange(DataSnapshot dataSnapshot) {
             for (DataSnapshot childSnapshot: dataSnapshot.getChildren()) {
                String name1 = childSnapshot.child("name").getValue(String.class);
                textname.setText(name1);
             }
          }
          @Override
          public void onCancelled(DatabaseError databaseError) {
             // Handle error
          }
       });
       String usn11 = textusn.getText().toString();
       applyleave.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View v) {
             Intent intent = new Intent(AfterLogin.this, Applyfroleave.class);
             intent.putExtra(EXTRA_USN, usn11);
             startActivity(intent);
          }
       });
       String usn123 = textusn.getText().toString();
       payfees.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View v) {
             Intent intent = new Intent(AfterLogin.this, PayFees.class);
             intent.putExtra(PayFees.FEES_USN, usn123);
             startActivity(intent);
          }
       });
       String usn1234 = textusn.getText().toString();
       issues.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View v) {
             Intent intent = new Intent(AfterLogin.this, Issues.class);
             intent.putExtra(ISSUE_USN, usn1234);
             startActivity(intent);
          Z
       notices.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View v) {
             Intent intent = new Intent(AfterLogin.this, Notices.class);
startActivity(intent);
          }
       });
       String usn12 = textusn.getText().toString();
       String user_branch = textbranch.getText().toString();
       btm.setSelectedItemId(R.id.home);
       btm.setOnNavigationItemSelectedListener(new
      BottomNavigationView.OnNavigationItemSelectedListener() {
          @Override
          public boolean onNavigationItemSelected(@NonNull MenuItem item) {
             switch (item.getItemId()) {
                case R.id.leave:
                   Intent intent = new Intent(getApplicationContext(), leavehistory.class);
                   intent.putExtra(USN, leave_history_usn);
startActivity(intent);
            overridePendingTransition(0, 0);
          case R.id.home:
            return true;
          case R.id.user:
            Intent intent1 = new Intent(getApplicationContext(), UserProfile.class);
            Bundle bundle = new Bundle();
            bundle.putString("USER_USN", user_usn);
            bundle.putString("USER_BRANCH", user_branch);
            intent1.putExtras(bundle);
            // intent1.putExtra(USER_USN,user_usn);
            //intent1.putExtra(USER_BRANCH,user_branch);
            startActivity(intent1);
            overridePendingTransition(0, 0);
        }
        return false;
      }
    });
  }
}
```

4.2.4 Apply For Leave

```
public class Applyfroleave extends AppCompatActivity {
  private EditText name, usn, reason, parent_contact, dateoa;
  private Button btn;
  FirebaseFirestore db;
  ProgressDialog progressDialog;
  public static final String EXTRA_NAME = "com.example.sahayadriapp.extra.name";
  public static final String EXTRA_USN = "com.example.sahayadriapp.extra.USN";
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_applyfroleave);
    EditText name = findViewById(R.id.editTextTextPersonName);
    EditText usn = findViewById(R.id.editTextTextPersonName1);
    EditText reason = findViewById(R.id.editTextTextPersonName21);
    EditText parent_contact = findViewById(R.id.editTextTextPersonName2);
    EditText dateoa = findViewById(R.id.editTextDate);
    Button btn = findViewById(R.id.button);
    db = FirebaseFirestore.getInstance();
    Intent intent = getIntent();
    String usn2 = intent.getStringExtra(AfterLogin.EXTRA_USN);
    usn.setText(usn2);
    usn.setFocusableInTouchMode(false);
    DatabaseReference reference = FirebaseDatabase.getInstance().getReference("datauser");
       reference.orderByChild("usn").equalTo(usn2).addListenerForSingleValueEvent(new
   ValueEventListener() {
      @Override
      public void onDataChange(DataSnapshot dataSnapshot) {
        for (DataSnapshot childSnapshot: dataSnapshot.getChildren()) {
          String name1 = childSnapshot.child("name").getValue(String.class);
          name.setText(name1);
          name.setFocusableInTouchMode(false);
```

```
            }
          }
          @Override
          public void onCancelled(DatabaseError databaseError) {
            // Handle error
          }
        });
        btn.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View v) {
            progressDialog = new ProgressDialog(Applyfroleave.this);
            progressDialog.setCancelable(false);
            progressDialog.setMessage("Submitting the data...");
            progressDialog.show();
            String name1 = name.getText().toString().trim();
            String usn1 = usn.getText().toString().trim();
            String reason1 = reason.getText().toString().trim();
            String parent_contact1 = parent_contact.getText().toString().trim();
            String dateoa1 = dateoa.getText().toString().trim();
            if (!name1.isEmpty()) {
              name.setError((null));
            } else {
              name.setError("Enter name");
              progressDialog.dismiss();
            }
            if (!usn1.isEmpty()) {
              usn.setError((null));
            } else {
              usn.setError("Enter USN");
              progressDialog.dismiss();
            }
            if (!reason1.isEmpty()) {
              reason.setError((null));
            }
            else {
              reason.setError("Enter reason");
              progressDialog.dismiss();
            }
            if (!parent_contact1.isEmpty()) {
            parent_contact.setError((null));
            } else {
              parent_contact.setError("Enter Phone number");
              progressDialog.dismiss();
            }
            if (!dateoa1.isEmpty()) {
              dateoa.setError((null));
              // progressDialog.dismiss();
              Map < String, Object > user = new HashMap < > ();
              user.put("name", name1);
              user.put("usn", usn1);
              user.put("reason", reason1);
              user.put("parent_contact", parent_contact1);
              user.put("date_of_apply", dateoa1);
              user.put("status", "PENDING");
              user.put("value", "1");
          db.collection("leaves").document(usn1).set(user)
            .addOnCompleteListener(new OnCompleteListener < Void > () {
              @Override
              public void onComplete(@NonNull Task < Void > task) {
                if (task.isSuccessful()) {
                  if (progressDialog.isShowing())
                    progressDialog.dismiss();
                  Intent intent = new Intent(Applyfroleave.this, Confirmation.class);
                  startActivity(intent);
                } else {
                  Toast.makeText(Applyfroleave.this, "Error", Toast.LENGTH_SHORT).show();
                }
              }
            });
        } else {
          dateoa.setError("Enter date");
        }
      }
    });
  }
}
```

4.2.5 Report Issues

```
public class Issues extends AppCompatActivity {
  private EditText name, usn, roomno, description;
  private Button subbtn;
  public static final String ISSUE_USN = "com.example.sahyadriapp.issueUSN";
  FirebaseFirestore db;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_issues);
    EditText name = findViewById(R.id.editTextTextPersonName);
    EditText usn = findViewById(R.id.editTextTextPersonName1);
    EditText roomno = findViewById(R.id.editTextTextPersonName21);
    EditText description = findViewById(R.id.editTextTextPersonName2);
    Button subbtn = findViewById(R.id.button);
    Intent intent = getIntent();
    String usn456 = intent.getStringExtra(AfterLogin.ISSUE_USN);
    usn.setText(usn456);
    usn.setFocusableInTouchMode(false);
    db = FirebaseFirestore.getInstance();
    DatabaseReference reference = FirebaseDatabase.getInstance().getReference("datauser");
       reference.orderByChild("usn").equalTo(usn456).addListenerForSingleValueEvent(new
   ValueEventListener() {
      @Override
      public void onDataChange(DataSnapshot dataSnapshot) {
        for (DataSnapshot childSnapshot: dataSnapshot.getChildren()) {
          String name1 = childSnapshot.child("name").getValue(String.class);
          name.setText(name1);
          name.setFocusableInTouchMode(false);
        }
      }
      @Override
      public void onCancelled(DatabaseError databaseError) {
        // Handle error
      }
    });
  subbtn.setOnClickListener(new View.OnClickListener() {
      @Override
        public void onClick(View v) {
      String name1 = name.getText().toString().trim();
        String usn1 = usn.getText().toString().trim();
        String room1 = roomno.getText().toString().trim();
        String description1 = description.getText().toString().trim();
        Map < String, Object > user = new HashMap < > ();
        user.put("name", name1);
        user.put("usn", usn1);
        user.put("room", room1);
        user.put("description", description1);
        db.collection("issues").document(usn1).set(user)
          .addOnCompleteListener(new OnCompleteListener < Void > () {
            @Override
            public void onComplete(@NonNull Task < Void > task) {
        if (task.isSuccessful()) {
                Intent intent = new Intent(Issues.this, Confirmation.class);
                startActivity(intent);
              } else {
                Toast.makeText(Issues.this, "Error", Toast.LENGTH_SHORT).show();
              }
            }
          });
      }
    });
  }
}
```

4.2.6 View Notice

```
public class Notices extends AppCompatActivity {
   private ListView pdfListView;
   private List < String > pdfList;
   private ArrayAdapter < String > pdfAdapter;
   private FirebaseStorage storage;
   private StorageReference storageRef;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_notices);
      pdfListView = findViewById(R.id.pdf_list_view);
      pdfList = new ArrayList < > ();
      pdfAdapter = new ArrayAdapter < > (this, android.R.layout.simple_list_item_1, pdfList);
      pdfListView.setAdapter(pdfAdapter);
      storage = FirebaseStorage.getInstance("gs://sahayadri-app.appspot.com");
      storageRef = storage.getReference();
      loadPDFsFromFirebase();
      pdfListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView < ? > adapterView, View view, int position, long id) {
                String selectedPDF = pdfList.get(position);
                 openPDF(selectedPDF);
            }
        });
    }
    private void loadPDFsFromFirebase() {
        storageRef.listAll().addOnSuccessListener(listResult - > {
            for (StorageReference item: listResult.getItems()) {
                item.getDownloadUrl().addOnSuccessListener(uri - > {
                    String downloadUrl = uri.toString();
                    String pdfName = item.getName();
                    pdfList.add(pdfName);
                    pdfAdapter.notifyDataSetChanged();
                });
            }
        });
    }
    private void openPDF(String pdfName) {
        StorageReference pdfRef = storageRef.child(pdfName);
        pdfRef.getDownloadUrl().addOnSuccessListener(uri - > {
            String downloadUrl = uri.toString();
            Intent intent = new Intent(Intent.ACTION_VIEW);
            intent.setDataAndType(Uri.parse(downloadUrl), "application/pdf");
            intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            try {
                startActivity(intent);
            } catch (Exception e) {
                // Handle the exception or show an error message
            }
        });
    }
}
```

4.2.7 Leave Status

```
public class leavehistory extends AppCompatActivity {
    BottomNavigationView btm;
    private Button btn;
    public static String USN = "com.example.sahyadriapp.USN";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_leavehistory);
        //  getSupportActionBar().hide();
        btm = findViewById(R.id.bottom_navigation);
        TextView derived_usn = findViewById(R.id.textView21);
        TextView reason_leave = findViewById(R.id.textView13);
        TextView date_leave = findViewById(R.id.textView14);
        TextView status_leave = findViewById(R.id.textView15);
        Button btn = findViewById(R.id.button9);
        Intent intent = getIntent();
        String leave_history_usn = intent.getStringExtra(AfterLogin.USN);
```

```
  derived_usn.setText(leave_history_usn);
FirebaseFirestore db = FirebaseFirestore.getInstance();
        btn.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        DocumentReference leaveRef = db.collection("leaves").document(leave_history_usn);
        leaveRef.get().addOnSuccessListener(new OnSuccessListener < DocumentSnapshot > () {
        @Override
          public void onSuccess(DocumentSnapshot documentSnapshot) {
            if (documentSnapshot.exists()) {
              // Retrieve the values from the document snapshot
              String reason = documentSnapshot.getString("reason");
              String date = documentSnapshot.getString("date_of_apply");
              String status = documentSnapshot.getString("status");
              // Update the corresponding TextViews with the retrieved values
              reason_leave.setText(reason);
              date_leave.setText(date);
              status_leave.setText(status);
            } else {
                         Toast.makeText(leavehistory.this, "Leave application not found",
         Toast.LENGTH_SHORT).show();
            }
          }
        }).addOnFailureListener(new OnFailureListener() {
          @Override
          public void onFailure(@NonNull Exception e) {
                       Toast.makeText(leavehistory.this, "Error retrieving leave application",
         Toast.LENGTH_SHORT).show();
          }
        });
      }
    });
    btm.setSelectedItemId(R.id.leave);
    btm.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
      @Override
      public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
          case R.id.home:
            startActivity(new Intent(getApplicationContext(), AfterLogin.class));
          case R.id.leave:
            return true;
          case R.id.user:
            startActivity(new Intent(getApplicationContext(), UserProfile.class));
        }
        return false;
      }
    });
  }
}
```

Warden View :

4.2.8 View Issues

```
 public class selectissues extends AppCompatActivity {
  RecyclerView recyclerView;
  ArrayList < User2 > userArrayList2;
  MyAdapter2 myAdapter2;
  FirebaseFirestore db;
  ProgressDialog progressDialog;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_selectissues);
    progressDialog = new ProgressDialog(this);
    progressDialog.setCancelable(false);
    progressDialog.setMessage("Fetching the data...");
    progressDialog.show();
    recyclerView = findViewById(R.id.recyclerView2);
    recyclerView.setHasFixedSize(true);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));
    db = FirebaseFirestore.getInstance();
    userArrayList2 = new ArrayList < User2 > ();
    myAdapter2 = new MyAdapter2(selectissues.this, userArrayList2);
    recyclerView.setAdapter(myAdapter2);
    EventChangeListener();
  }
  private void EventChangeListener() {
    db.collection("issues")
      .addSnapshotListener(new EventListener < QuerySnapshot > () {
        @Override
                public void onEvent(@Nullable QuerySnapshot value, @Nullable
        FirebaseFirestoreException error) {
          if (error != null) {
        if (progressDialog.isShowing())
              progressDialog.dismiss();
            Log.e("Firestore Error", error.getMessage());
            return;
          }
          for (DocumentChange dc: value.getDocumentChanges()) {
            if (dc.getType() == DocumentChange.Type.ADDED) {
              userArrayList2.add(dc.getDocument().toObject(User2.class));
            }
            myAdapter2.notifyDataSetChanged();
            if (progressDialog.isShowing())
              progressDialog.dismiss();
          }
        }
      });
  }
}
```

4.2.9 Upload Notice

```
public class Notice extends AppCompatActivity {
  Uri pdfuri = null;
 EditText Title;
 Button upload_btn;
 ProgressBar progressBar;
 @Override
 protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_notice);
    upload_btn = findViewById(R.id.upload_btn);
    progressBar = findViewById(R.id.progressBar);
    progressBar.setVisibility(View.GONE);
    upload_btn.setOnClickListener(new View.OnClickListener() {
       @Override
       public void onClick(View view) {
          openGallery();
       }
    });
 }
 private void openGallery() {
    Intent intent = new Intent();
    intent.setAction(Intent.ACTION_GET_CONTENT);
    intent.setType("application/pdf");
    activityResultLauncher.launch(intent);
 }
 public ActivityResultLauncher < Intent > activityResultLauncher = registerForActivityResult(
    new ActivityResultContracts.StartActivityForResult(),
             new ActivityResultCallback < ActivityResult > () {
       @Override
       public void onActivityResult(ActivityResult activityResult) {
          int result = activityResult.getResultCode();
          Intent data = activityResult.getData();
          if (result == RESULT_OK) {
             pdfuri = data.getData();
           Title = findViewById(R.id.title_text);
             String title = Title.getText().toString();
             final String messagePushID = title;
             FirebaseStorage storage = FirebaseStorage.getInstance("gs://sahayadri-app.appspot.com");
             StorageReference storageRef = storage.getReference();
             final StorageReference filepath = storageRef.child(messagePushID + ".pdf");
             Toast.makeText(Notice.this, filepath.getName(), Toast.LENGTH_SHORT).show();
             progressBar.setVisibility(View.VISIBLE);
             UploadTask uploadTask = filepath.putFile(pdfuri);
                         uploadTask.addOnProgressListener(new OnProgressListener <
          UploadTask.TaskSnapshot > () {
                @Override
                public void onProgress(@NonNull UploadTask.TaskSnapshot snapshot) {
                                           double progress = (100.0 * snapshot.getBytesTransferred()) /
                      snapshot.getTotalByteCount();
                   progressBar.setProgress((int) progress);
                   progressBar.setVisibility(View.VISIBLE);
                }
}).continueWithTask(new Continuation < UploadTask.TaskSnapshot, Task < Uri >> () {
        @Override
                    public Task < Uri > then(@NonNull Task < UploadTask.TaskSnapshot > task)
        throws Exception {

        if (!task.isSuccessful()) {
                throw task.getException();
              }
              return filepath.getDownloadUrl();
            }
          }).addOnCompleteListener(new OnCompleteListener < Uri > () {
            @Override
            public void onComplete(@NonNull Task < Uri > task) {
              progressBar.setVisibility(View.GONE);
              if (task.isSuccessful()) {
                Uri downloadUri = task.getResult();
                                Toast.makeText(Notice.this, "Uploaded Successfully",
                Toast.LENGTH_SHORT).show();
                // Do something with the download URI
              } else {
                Toast.makeText(Notice.this, "Upload Failed", Toast.LENGTH_SHORT).show();
              }
            }
          });
        } else {
          Toast.makeText(Notice.this, "FAILED", Toast.LENGTH_LONG).show();
        }
      }
    }
```

4.2.10 Approve / Reject leaves

```
public class leaves extends AppCompatActivity {
  RecyclerView recyclerView;
  TextView usnView;
  ArrayList < User > userArrayList;
  MyAdapter myAdapter;
  FirebaseFirestore db;
  ProgressDialog progressDialog;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_leaves);
    // Button approvebtn=findViewById(R.id.button);
    // Button rejectbtn=findViewById(R.id.button1);
    //TextView usnView = findViewById(R.id.usnView);
    progressDialog = new ProgressDialog(this);
    progressDialog.setCancelable(false);
    progressDialog.setMessage("Fetching the data...");
    progressDialog.show();
    recyclerView = findViewById(R.id.recyclerView);
    recyclerView.setHasFixedSize(true);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));
```

```
      db = FirebaseFirestore.getInstance();
       userArrayList = new ArrayList < User > ();
       myAdapter = new MyAdapter(leaves.this, userArrayList);
       recyclerView.setAdapter(myAdapter);
       EventChangeListener();
    }

   private void EventChangeListener() {
      db.collection("leaves").orderBy("value", Query.Direction.ASCENDING)
        .addSnapshotListener(new EventListener < QuerySnapshot > () {
          @Override
          public void onEvent(@Nullable QuerySnapshot value, @Nullable FirebaseFirestoreException
error) {
            if (error != null) {
              if (progressDialog.isShowing())
                progressDialog.dismiss();
              Log.e("Firestore Error", error.getMessage());
              return;
            }
            for (DocumentChange dc: value.getDocumentChanges()) {
              if (dc.getType() == DocumentChange.Type.ADDED) {
                userArrayList.add(dc.getDocument().toObject(User.class));
              }
              myAdapter.notifyDataSetChanged();
              if (progressDialog.isShowing())
                progressDialog.dismiss();
            }
          }
        });
}
```

CHAPTER 5

RESULTS

5.1 SPLASH SCREEN

A splash screen is a screen of the software that displays while the application or other item is loading. After the load is complete, the user is generally taken to another functional screen.

Figure 5.1.1: Splash Screen

5.2 SIGNIN PAGE

The Sign-in Page allows users to securely log into their accounts, ensuring personalized access to the application's features and functionalities.

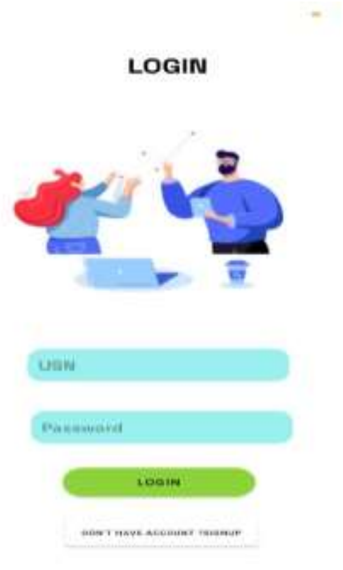


Figure 5.2.1: Sign-in Page

5.3 SIGNUP PAGE

The Sign-up Page allows users to create a new account

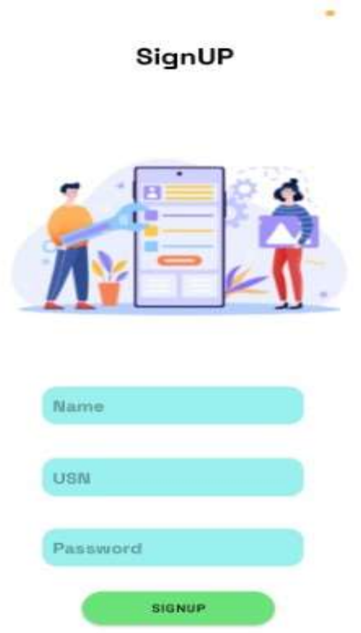


Figure 5.3.1: Sign-Up Page

5.4 HOME PAGE

Home page providing easy access to features, relevant updates, and important information

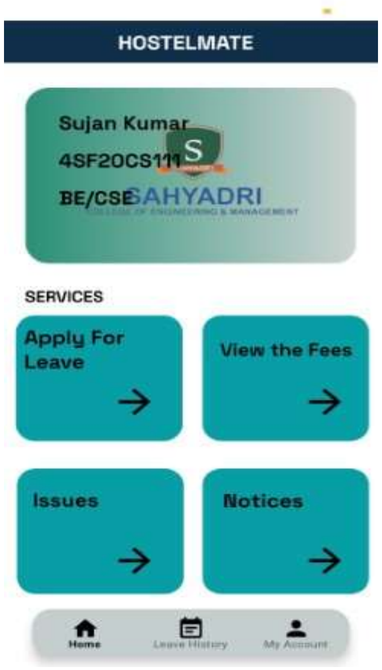
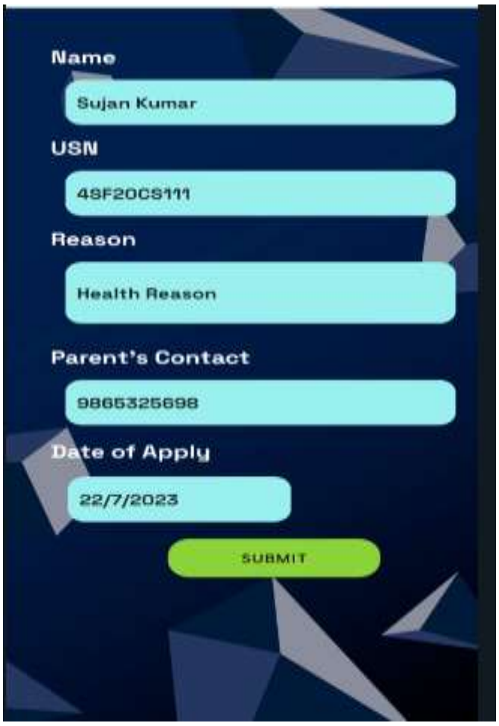


Figure 5.4.1: Student Home Page

5.5 LEAVE APPLICATION PAGE

Enables students to apply for leave by specifying the dates, reason



Name

Sujan Kumar

USN

4SF20CS111

Reason

Health Reason

Parent's Contact

9865325698

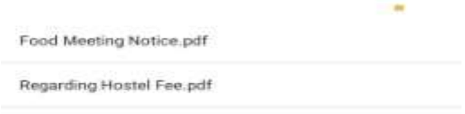
Date of Apply

22/7/2023

Figure 5.5.1:Leave Application Page

5.6 NOTICES PAGE

Provides students with access to view important notices and announcements from the hostel warden**.**



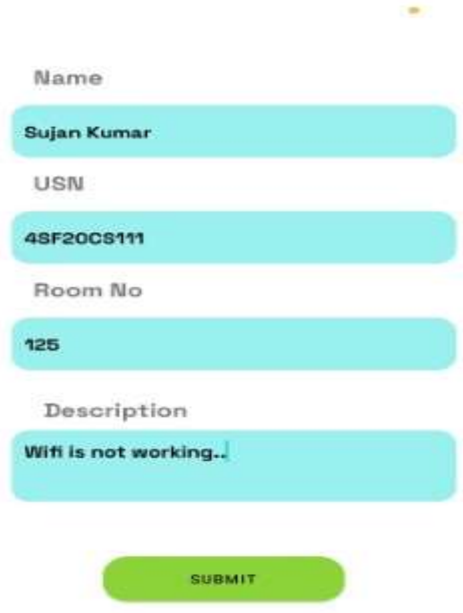
Food Meeting Notice.pdf

Regarding Hostel Fee.pdf

Figure 5.7
5.6.1: View Notice Page

5.7 REPORT ISSUE PAGE

The Report Issue page enables users to submit and communicate their concerns or maintenance requests efficiently in the application.



Name

Sujan Kumar

USN

4SF20CS111

Room No

125

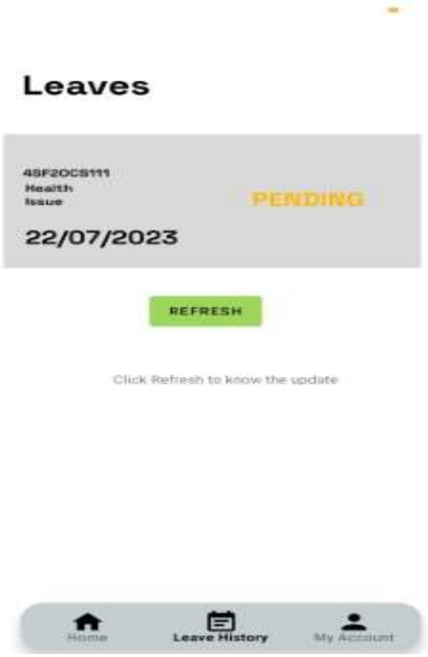
Description

Wifi is not working..

Figure 5.7.1:Report Complaint Page

5.8 LEAVE STATUS PAGE

Allows students to track the status of their leave application, whether it is approved, or rejected.



Leaves

4SF20CS111
Health
Issue

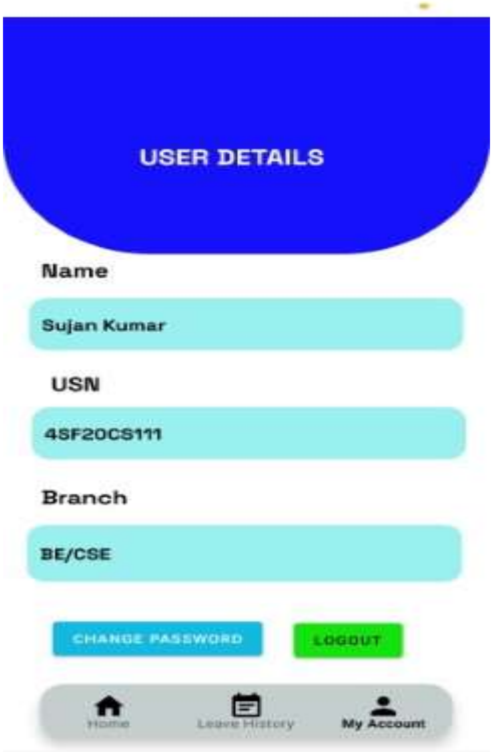
PENDING

22/07/2023

Figure 5.8.1: Leave Status Page

5.9 USER DETAILS PAGE

Displays the student's details, including their name, USN, branch and any other relevant information.



USER DETAILS

Name

Sujan Kumar

USN

4SF20CS111

Branch

BE/CSE

Figure 5.9.1: User Details Page

5.10 WARDEN HOME PAGE

The Warden Home Page offers quick access to leave, issues, and notices pages

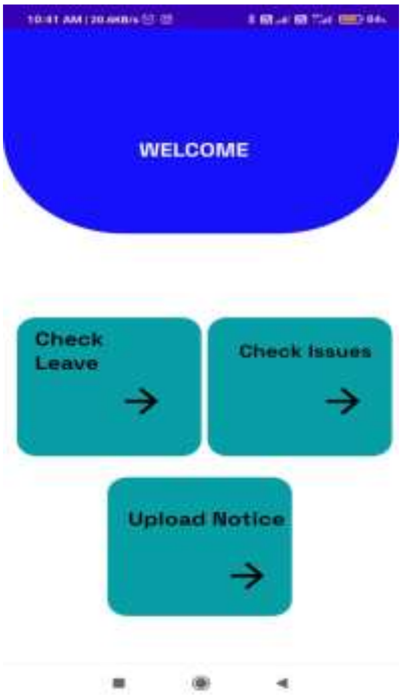
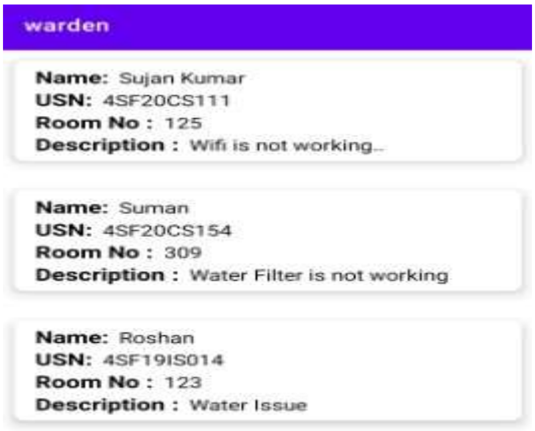


Figure 5.10.1: Warden Home Page

5.11 VIEW ISSUE PAGE

Allows the wardens to view and manage reported issues, such as maintenance requests or complaints, raised by students.



Name: Sujan Kumar
USN: 4SF20CS111
Room No : 125
Description : Wifi is not working..

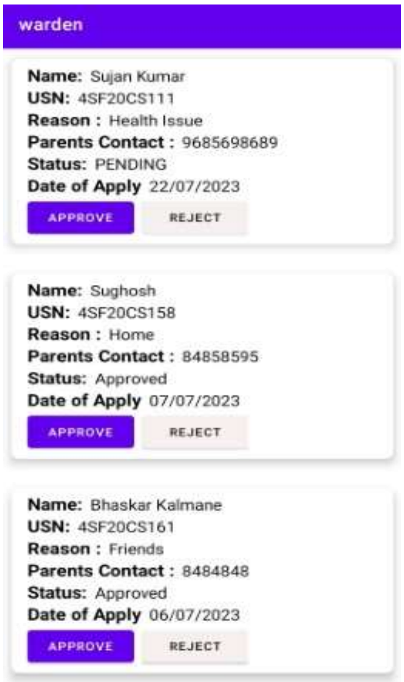
Name: Suman
USN: 4SF20CS154
Room No : 309
Description : Water Filter is not working

Name: Roshan
USN: 4SF19IS014
Room No : 123
Description : Water Issue

Figure 5.11.1: View Issues Page

5.12 APPLY / REJECT LEAVE PAGE

Enables wardens to review and approve or reject leave applications submitted by students



Name: Sujan Kumar
USN: 4SF20CS111
Reason : Health Issue
Parents Contact : 9685698689
Status: PENDING
Date of Apply 22/07/2023

Name: Sughosh
USN: 4SF20CS158
Reason : Home
Parents Contact : 84858595
Status: Approved
Date of Apply 07/07/2023

Name: Bhaskar Kalmane
USN: 4SF20CS161
Reason : Friends
Parents Contact : 8484848
Status: Approved
Date of Apply 06/07/2023

Figure 5.12.1: Approve / Reject Leave

5.13 Upload Notice Page

The Upload Notice page enables wardens to share important information and notices with students.

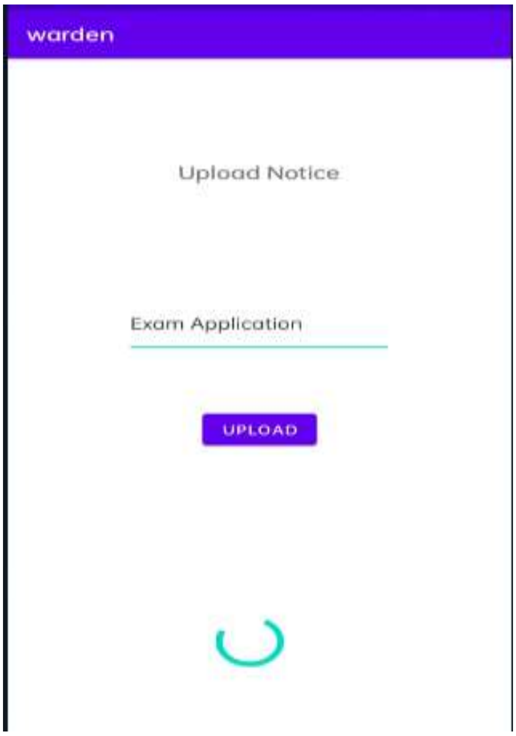
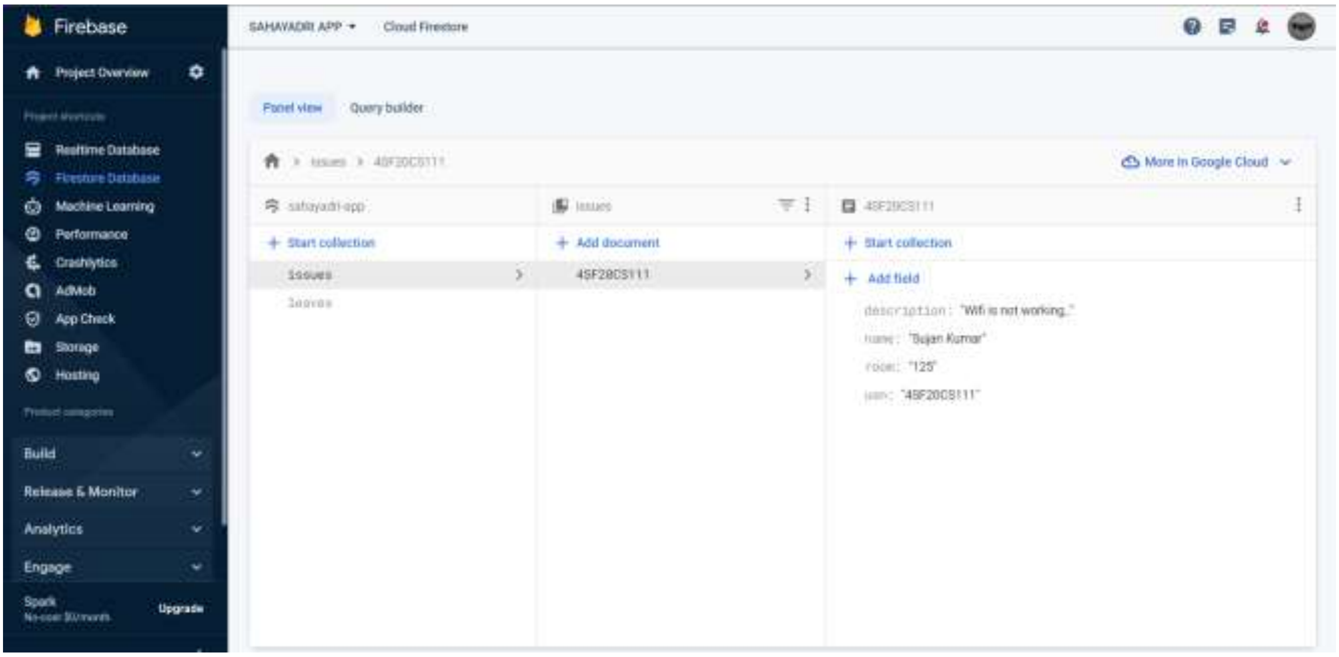


Figure 5.13.1: Upload Notice

DATABASE

5.14: Firestore Database

Firestore is a flexible and scalable NoSQL document database provided by Firebase, a platform for building web and mobile applications. It is designed to store and synchronize data in real-time, making it ideal for collaborative and data-intensive applications.

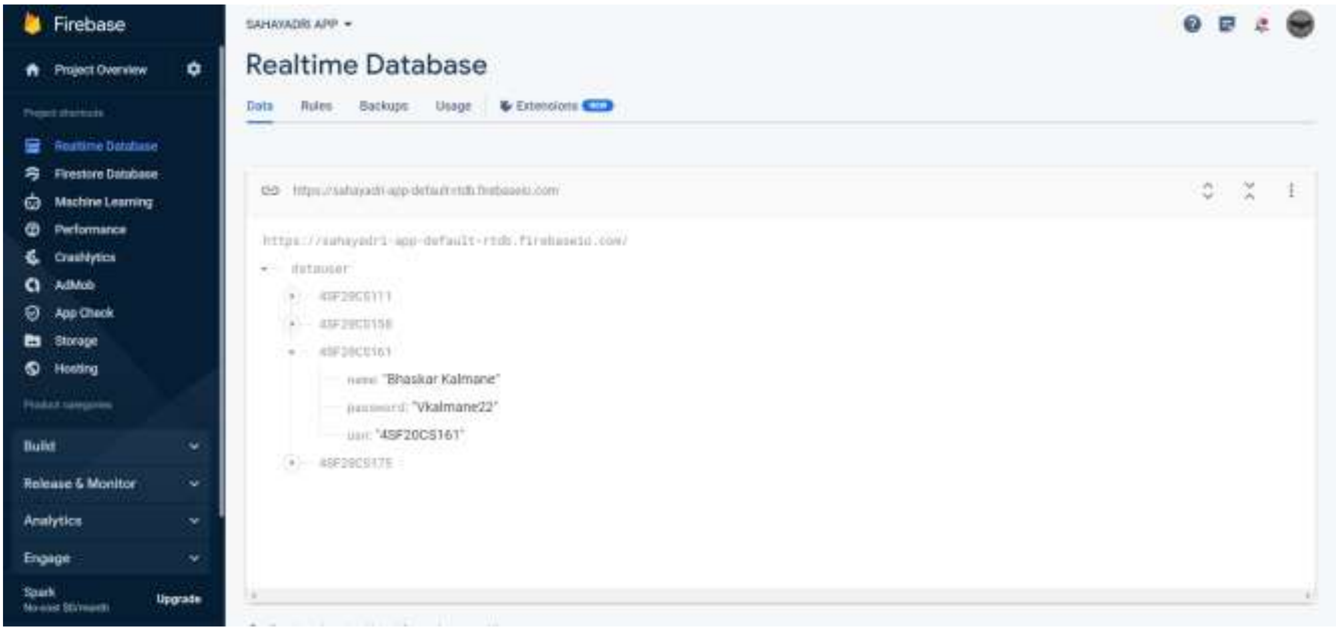


```
description: "Wifi is not working."
name: "Sujan Kumar"
room: "125"
usn: "4SF20CS111"
```

Fig 5.14.1 FireStore Databse

5.15: Realtime database

The Realtime Database is a NoSQL cloud-hosted database provided by Firebase that offers real-time synchronization and data storage for web and mobile applications. It uses a JSON-based data model and provides automatic updates to connected clients whenever data changes.



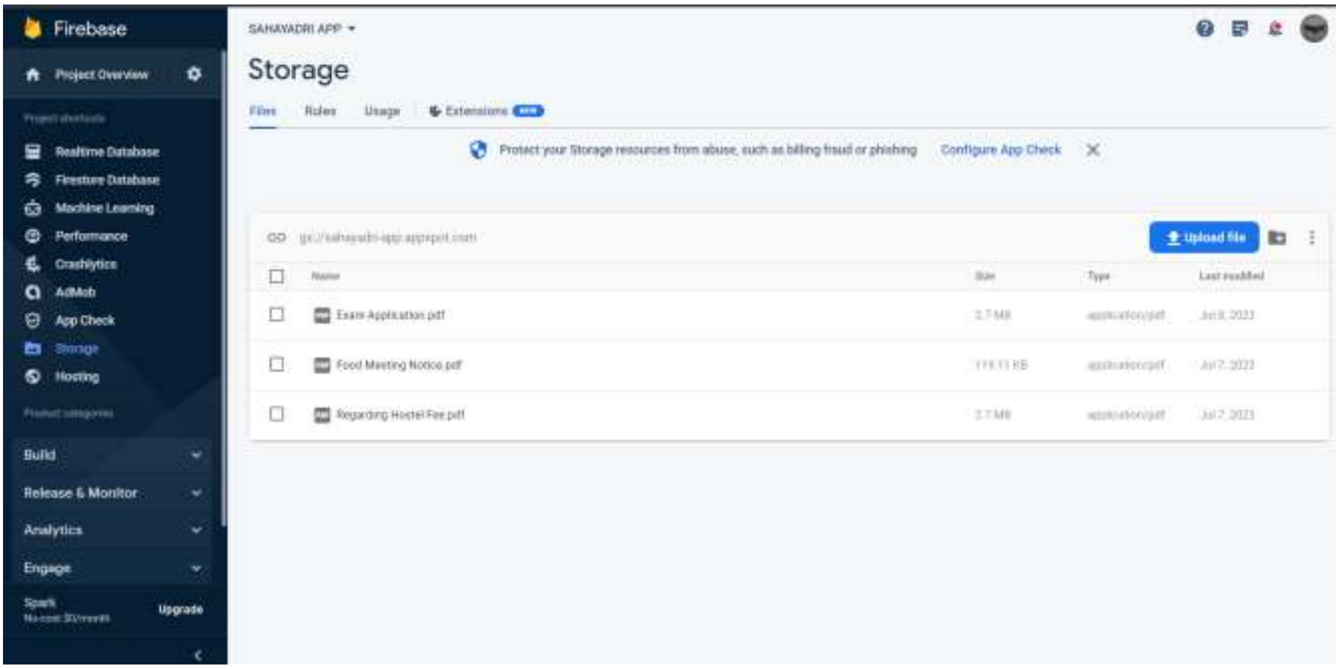
https://sahayadri-app-default-rtdb.firebaseio.com

```
https://sahayadri-app-default-rtdb.firebaseio.com/
- datauser
    + 4SF20CS111
    + 4SF20CS158
    - 4SF20CS161
        name: "Bhaskar Kalmane"
        password: "Vkalmane22"
        usn: "4SF20CS161"
    + 4SF20CS175
```

Fig 5.15.1 Realtime database

5.16 Storage

Firebase Storage is a cloud-based object storage service provided by Firebase. It allows developers to securely upload, download, and manage user-generated content such as images, videos, and other files.



Protect your Storage resources from abuse, such as billing fraud or phishing

gs://sahayadri-app.appspot.com

□	Name	Size	Type	Last modified
□	Exam Application.pdf	2.7 MB	application/pdf	Jul 8, 2023
□	Food Meeting Notice.pdf	119.11 KB	application/pdf	Jul 7, 2023
□	Regarding Hostel Fee.pdf	2.7 MB	application/pdf	Jul 7, 2023

Fig 5.16.1 Storage

CHAPTER 6

CONCLUSION

While developing this project we have learnt a lot about basic concepts of android phone features and capabilities. We understood the basic technologies used by the android platform, relationship between XML and Java for the android platform, uses of AndroidManifest.xml, Main.xml, Strings.xml, grid view, drawable folder, etc. We have learned through our project that Android is a much more diverse operating system than iOS and Windows Phone Mobile. Android has grown rapidly over the past 4 years becoming the most used smartphone operating system in the world. It is unique and incomparable to other mobile operating systems It's no wonder that, the app development process is exhausting, and overwhelming also. There have been lots of steps and processes to follow, and many decision-making mechanisms are involved as well. There are still challenges that could be out of the box, but we have tried our best to cover multiple scenarios and possibilities that might be encountered.

REFERENCES

[1] Sandeep Jain (2023 June 15), "Android Tutorial"
https://www.geeksforgeeks.org/android-tutorial/

[2] Google (2023 June 15) ,"Developer guides".

https://developer.android.com/docs

[3] James Tamplin and Andrew Lee (2023 June 15) ,"Learn the fundamentals".

https://developer.android.com/docs