ACKNOWLEDGEMENT

While presenting this MOBILE APPLICATION DEVELOPMENT mini project on "**Online Examination Application",** we feel that it is our duty to acknowledge the help rendered to us by various persons.

Firstly, we thank God for showering his blessings on us. We are grateful to our institution PES institute of technology and management for providing us a congenial atmosphere to carry out the project successfully. We would like to express our heartfelt gratitude to Mr. Chaitanya Kumar **M V,** Principal, PESITM, Shivamogga, for extending his support.

We would also like to express our heartfelt gratitude to **Dr. Arjun U,** HOD, Computer Science and Engineering whose guidance and support was truly valuable.

We are very grateful to our guide, Ms. Vinutha H M, and our co guides, Department of Computer Science, for their able guidance and valuable advice at every stage of our project which helped me in the successful completion of our project.

We would also have indebted to our parent and friends for their continued moral and material support throughout the course of project and helping me in finalize the presentation.

Our hearty thanks to all those have contributed bits, bytes and words to accomplish this project.

Shaishav (4PM20CS090)
Sumit Raj (4PM20CS115)
Supritha S (4PM20CS116)
Swathi J Gowda (4PM20CS117)

DECLARATION

We student of 6th semester BE, Computer Science and Engineering department hereby declare that project work entitled “Online Examination Application” has been carried out by us at PESITM, Shivamogga and submitted in partial fulfilment of the course requirement for the award of the degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum, during the academic year 2022-2023.

We also declare that, to the best of our knowledge and belief, the work reported here does not form the part of dissertation on the basis of which a degree or award was conferred on a earlier occasion on this by any other student.

Date:

Place: Shimoga.

SHAISHAV (4PM20CS090)
SUMIT RAJ (4PM20CS115)
SUPRITHA S (4PM20CS116)
SWATHI J GOWDA (4PM20CS117)

ABSTRACT

This project proposes the development of an Android mobile application for conducting online examinations. The application will allow users to create and administer exams, as well as take exams themselves. The application will be designed to be user-friendly and secure, and it will be able to support a variety of question types, including multiple choice, true/false, and fill-in-the-blank. The application will also be able to track the progress of students and generate reports on their performance.

The development of this application will have a number of benefits. First, it will allow for more convenient and efficient testing. Second, it will make it easier to track student progress and identify areas where they need additional help. Third, it will provide a more secure and reliable way to conduct exams.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

ONLINE EXAMINATION APPLICATION targets all the android phones. The mobile application development project aims to create an innovative online examination application that revolutionizes the traditional approach to conducting exams. With the increasing use of mobile devices and the need for remote learning solutions, this application offers a convenient and efficient platform for students, educators, and institutions to conduct exams seamlessly.

The application will be designed with a user-centric approach, focusing on providing a user-friendly interface and intuitive navigation. It will support a wide range of exam types, including multiple choice, subjective, and time-based exams. The application will also incorporate features for adding multimedia elements to enhance the exam experience.

1.1 Introduction to Mobile application development

Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources. Hence, the mobile development process involves creating installable software bundles (code, binaries, assets, etc.) , implementing backend services such as data access with an API, and testing the application on target devices. There are two dominant platforms in the modern smartphone market. One is the iOS platform from Apple Inc. The iOS platform is the operating system that powers Apple's popular line of iPhone smartphones. The second is Android from Google. The Android operating system is not only used by Google devices but also by many other OEMs to built their own smartphones and other smart devices.

Figure 1.1 Devices running on Android Operating System

1.1.1 ANDROID

Android is an operating system and programming platform for smart phones and other mobile devices (such as tablets). It includes a software development kit for writing original code and assembling software modules to create apps for Android users. It also provides a marketplace to distribute apps. Altogether, Android represents an ecosystem for mobile apps. Android is a widely-adopted opensource project.

Android Studio is Android's official IDE. It is purpose-built for Android to accelerate your development and help you build the highest-quality apps for every Android device

1.1.2 History:

This was owned by Open Handset Alliance. Google formed a group of hardware, software, and telecommunication companies called the Open Handset Alliance with the goal of contributing to Android development. They then made up of 84 companies. The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008.

On June 27, 2012, at the Google I/O conference, Google announced the next Android version,

4.1 Jelly Bean. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance.

1.1.3 Android Architecture:

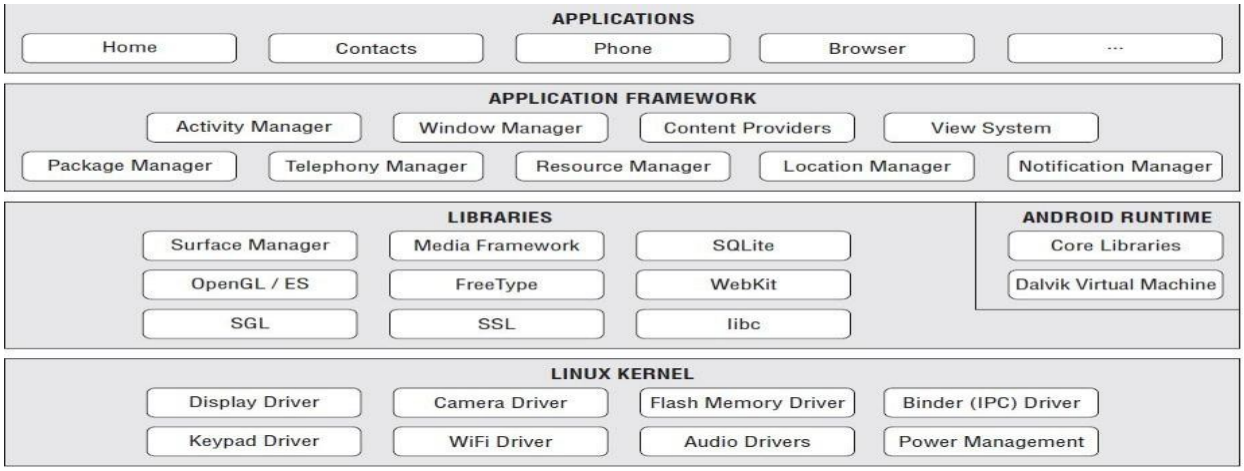


Figure 1.2 Android Architecture

Android Applications: Your apps live at this level, along with core system apps for email, SMS messaging, calendars, Internet browsing, or contacts. Android applications are usually

developed in the Java language using the Android Software Development Kit. Once developed, Android applications can be packaged easily and sold out either through a store such as GooglePlay, SlideME, Opera Mobile Store, Mobango, F-droid and the Amazon Appstore.

Java API Framework/Android Framework

All features of Android are available to developers through application programming interfaces (APIs) written in the Java language. Like:

- View System: used to build an app's UI, including lists, buttons, and menus.
- Resource Manager: used to access to non-code resources such as localized strings, graphics, and layout files.
- Notification Manager: used to display custom alerts in the status bar.
- Activity Manager: that manages the lifecycle of apps, etc.

Libraries and Android Runtime

Each app runs in its own process and with its own instance of the Android Runtime, which enables multiple virtual machines on low-memory devices. Android also includes a set of core runtime libraries. Many core Android system components and services are built from native code that require native libraries written in C and C++. These native libraries are available to apps through the Java API framework.

Linux Kernel

At its base, Android runs on a Linux kernel for interacting with the device's processor, memory, etc. Thus an Android device can be seen as a Linux computer.

1.2 Features of Android

Android is a powerful open-source operating system that open-source provides immense features and some of these are listed below.

Android Open-Source Project so we can customize the OS based on our requirements.

Android supports different types of connectivity for GSM, CDMA, Wi-Fi,

Bluetooth, etc. for telephonic conversation or data transfer.

Using wifi technology we can pair with other devices while playing games or using other applications.

It contains multiple APIs to support location-tracking services such as GPS.

We can manage all data storage related activities by using the file manager.

It contains a wide range of media supports like AVI, MKV, FLV, MPEG4, etc. to play or record a variety of audio/video.

It also supports different image formats like JPEG, PNG, GIF, BMP, MP3, etc.

It supports multimedia hardware control to perform playback or recording using a camera and microphone.

Android supports multi-tasking means we can run multiple applications at a time and can switch in between them.

It provides support for virtual reality or 2D/3D Graphics.

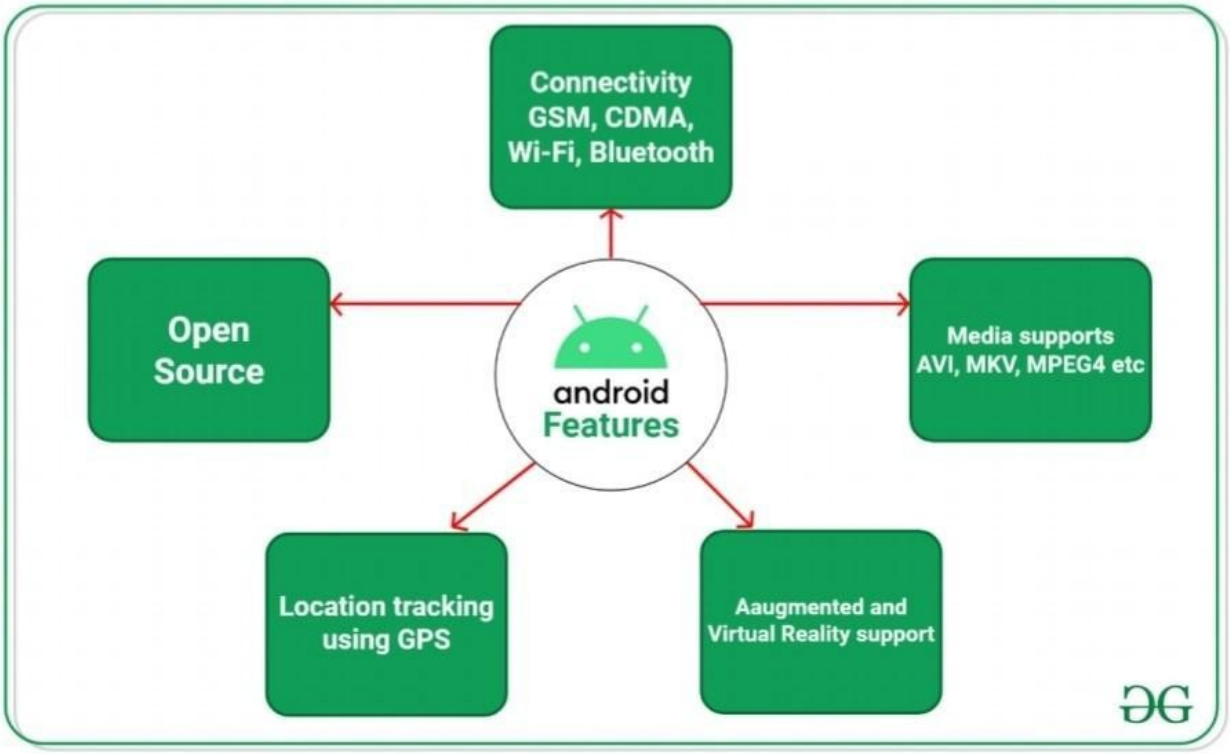


Figure 1.3 Features of Android

1.5 Project Goal

The project goal is to develop a user-friendly mobile application for online examinations. It aims to streamline the exam process, ensure secure authentication and data encryption, provide comprehensive exam management features, support various question types, and generate accurate and timely results. The application will be accessible on major mobile platforms, offering convenience and flexibility for students, educators, and institutions.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 Software and Hardware Requirements:

Basic system requirements for installing Android Studio:

Operating System Version

Microsoft Windows Microsoft Windows 7/8/10 (32- or 64-bitm

Random Access Memory (RAM)

4 GB RAM minimum; 8 GB RAM recommended.

Free digital storage

2 GB of available digital storage minimum, 4 GB Recommended (500 MB for IDE

1.5 GB for Android SDK and emulator system image).

Minimum required JDK Version

Java Development Kit 8

Minimum screen resolution 1280 x 800.

2.2 About Android Studio

Android Studio is exclusively designed for developing Android applications. It consists of all android SDK tools to design, develop, maintain, test, debug and publish our app. The IDE is very efficiently which makes the developer's job easy. It also supports the IntelliJ IDE, the main idea behind this IDE is that it automatically senses the variables, methods, classes, built-in functions or it could be anything else when we press the first letter of it. Say, suppose we declared few variables or methods that starts with an 'S', it automatically senses everything that starts with an 'S' and makes suggestions. It also supports Git as a version control system to maintain the app changes and push them into github. All java files, layout files (for design) are i ntegrated into a single project easily. After the completion of project, the whole application could be put as an .APK (Android Package) file, in which we can run that APK file in any d evice and use the application. Other main tools include Android SDK, ADB, and Gradle Build.

2.2.1Android Software Development Kit (SDK):

One of the main tools used in developing android applications, as it packages many core features into one SDK and it can be used in the application easily. This helps us to avoid writing lot of code, and building applications faster.

2.2.2 Android Debug Bridge (ADB):

Android SDK uses ADB tool as a connection device which allows us to connect the Android Devices or Emulator with the machine via USB. After developing or while developing applications, we can connect with the device to check how the application runs. Later, we can debug and run the applications.

2.2.3 Gradle Build:

Gradle Scripts are the recent feature that is added to Android Studio. 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. to ensure that the developed app is compatible with the testing device/emulator. Gradle is also similar to Ant and Maven which helps in maintaining java projects (repositories).

2.2.4 Android Device Monitor:

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. But, as they are hidden files, we need root permissions to access them. Suppose, if we run the app in device, we need to root the device and run commands in adb shell to get permissions.

2.2.5 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.

2.2.6 AVD Manager:

It is used to create virtual devices of any desired API level to support higher level SDK's incase our device does not support. Using emulators to test the application is difficult as it might be little slower when compared to real device.

2.2.7 SQLite Database:

Android also supports inbuilt database which is Android SQLite to develop any small applications and perform any CRUD (Create, Update, and Delete) operations. As it is not flexible enough to support substantial number of data, for complex applications we are using other external databases.

2.3 Programming Languages used in Developing Android Applications

Java

XmL

2.3.1 Java

As the project is developing an Android Application, the default programming language is Java. All Android applications are built using Java in Android Studio or Eclipse or both. Java is a popular and widely used language throughout the world. As mentioned in, Java is one of the powerful programming languages like C, C++. developed by Sun Microsystems which has many powerful features as described below. After the development of C, C++, Java has come into evolution by addressing their drawbacks. It is one of the open source projects that could be easily installed in our machine. The language is also easy to l earn, understand and implement. Java is used in various kinds of applications like Web, Desktop, Mobile, and Big Data. Many powerful features are supported by Java including various libraries, application services, graphics library for 2D/3D applications. The language is flexible enough to maintain code complexity, test, implementation, integration and support. Apart from these, one of the important hierarchies in the programming languages which is used to implement real time applications, it provides for code reusability, it has a platform independence feature including any virtual machines (Write Once Read Everywhere), as in no need to write the code for different OS as the Java Compliers convert the java source files to bytecode and this could be interpreted by any machine and the actual code is compiled irrespective of any machine, OS. It is more secured as the compilers are designed efficiently to figure out any kind of errors. Android code (program control and logic, as well as data storage and manipulation) is written in Java. Writing Android code will feel a lot writing any other Java program: you create classes, define methods, instantiate objects, and call methods on those objects. But because you're working within a framework.

Java is a popular programming language, created in 1995.

It is owned by Oracle, and more than **3 billion** devices run Java. It is used for:

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection

2.3.2 XML

XML (Extensible Markup Language) is a markup language similar to HTML, but without predefined tags to use. Instead, you define your own tags designed specifically for your needs. This is a powerful way to store data in a format that can be stored, searched, and shared. Most importantly, since the fundamental format of XML is standardized, if you share or transmit XML across systems or platforms, either locally or over the internet, the recipient can still parse the data due to the standardized XML syntax. The design goals of XML focus on simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

XML Properties

XML is a markup language that focuses on data rather than how the data looks.

XML is designed to send, store, receive and display data. In simple words you can say that XML is used for storing and transporting data.

XML became a W3C (W3C stands for World Wide Web Consortium, the main international s tandards organization for the World Wide Web) recommendation on February 10, 1998.

XML is different from HTML. XML focuses on data while HTML focuses on how the data looks.

XML does not depend on software and hardware, it is platform and programming language independent.

Unlike HTML where most of the tags are predefined, XML doesn't have predefined tags, rather you have to create your own tags.

2.3.4 Firebase

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets organizations store and sync data in real time across all of their users' devices. This makes it easy to build apps that are always up to date, even when users are offline. The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in real-time to every connected client. When you build cross-platform apps with our Apple platforms, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data. Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices.

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.

When you upgrade to Firebase Authentication with Identity Platform, you unlock additional features, such as multi-factor authentication, blocking functions, user activity and audit logging, SAML and generic OpenID Connect support, multi-tenancy, and enterprise-level support.

Key capabilities

You can sign in users to your Firebase app either by using FirebaseUI as a complete drop-in auth solution or by using the Firebase Authentication SDK to manually integrate one or several sign-in methods into your app.

Firebase Authentication with Identity Platform

Firebase Authentication with Identity Platform is an optional upgrade that adds several new features to Firebase Authentication.

This upgrade does not require any migration—your existing client SDK and admin SDK code will continue to work as before, and you'll gain immediate access to features such as enhanced logging and enterprise-grade support and SLAs. With some additional code, you'll be able to add multi-factor auth, blocking functions, and support for SAML and OpenID Connect providers.

Firebase Authentication with Identity Platform has a different pricing scheme compared to the base product. When upgraded, no-cost (Spark) plan projects will be limited to 3,000 daily active users, and pay-as-you-go (Blaze) plan projects will be charged for usage beyond the free tier of 50,000 monthly active users. Be sure you understand the billing implications before you upgrade.

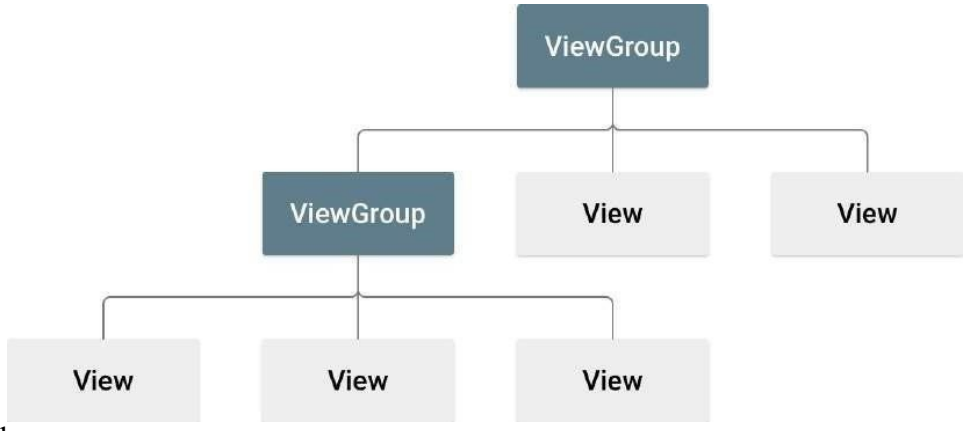
CHAPTER 3

SYSTEM DESIGN AND IMPLEMENTATION

3.1 DESIGN

USER INTERFACE:

The user interface (UI) for an Android app is built as a hierarchy of layouts and widgets. The layouts are View Group objects, containers that control how their child views are positioned on the screen. Widgets are View objects, UI components such as buttons and text



boxes.

Android provides an XML vocabulary for View Group and View classes, so most of your UI is defined in XML files. However, rather than teach you to write XML, this lesson shows you how to create a layout using Android Studio's Layout Editor. The Layout Editor writes the XML for you as you drag and drop views to build your layout.

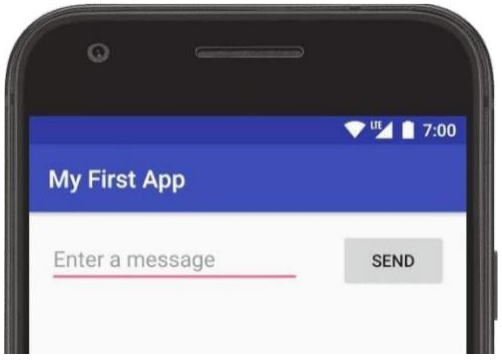
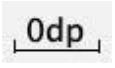


Fig no 5: user interface

3.2 LAYOUT EDITOR:

1. In the Project window, open app > res > layout > activity_main.xml.
2. To make room for the Layout Editor, hide the Project window. To do so, select View > Tool Windows > Project, or just click Project on the left side of the Android Studio screen.
3. If your editor shows the XML source, click the Design tab at the top right of the window.
4. Click  (Select Design Surface) and select Blueprint.
5. Click  (View Options) in the Layout Editor Toolbar and make sure that Show All Constraints is checked.
6. Make sure Auto connect is off. A tooltip in the toolbar displays  (Enable Auto connection to Parent) when Auto connects is off.
7. Click 

(Default Margins) in the toolbar and select. If needed, you can adjust the margins for each view later.
8. Click  (Device for Preview) in the toolbar and select 5.5, 1440 × 2560, 560 dpi (Pixel XL).

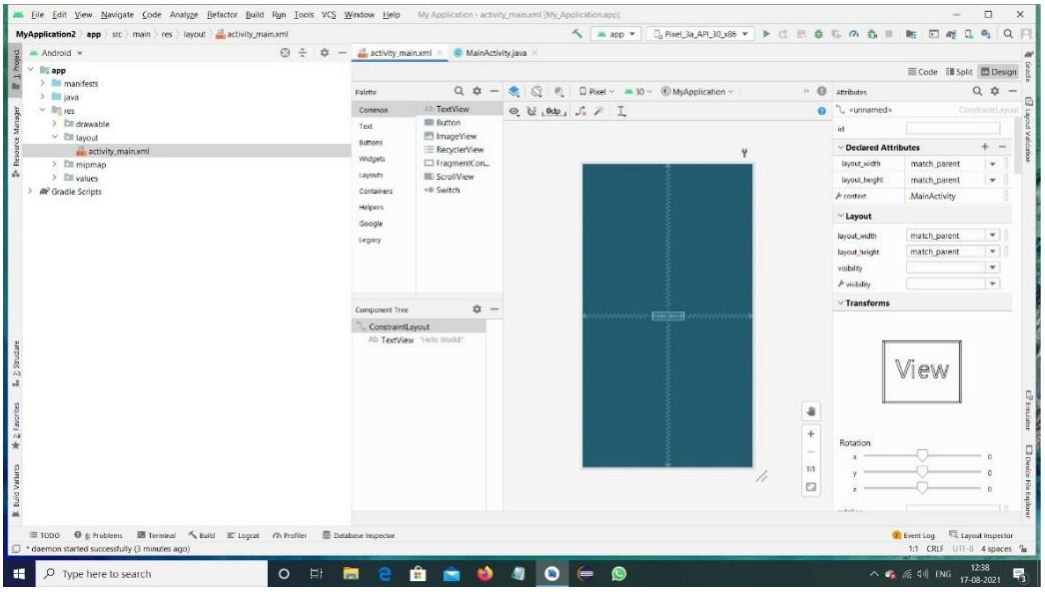


Fig no 6: Layout Editor

3.2.1 ADD A TEXTBOX:

First, you need to remove what's already in the layout. Click Text View in the Component Tree panel and then press the Delete key.

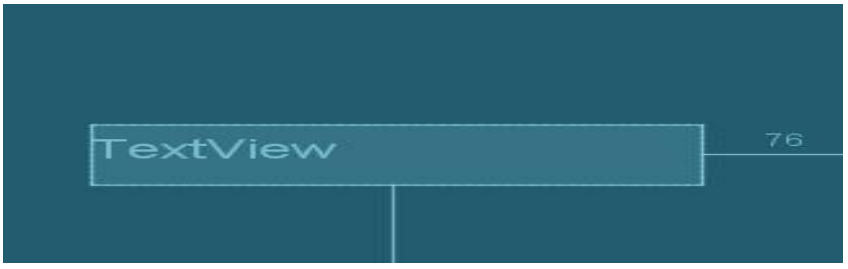


Fig no 7: Add a textbox.

1. In the Palette panel, click Text to show the available text controls.
2. Drag the Plain Text into the design editor and drop it near the top of the layout. This is an Edit Text widget that accepts plain text input.

3.2.2 ADD A BUTTON

1. In the Palette panel, click Buttons.
2. Drag the Button widget into the design editor and drop it near the right side.
3. Create a constraint from the left side of the button to the right side of the text box.

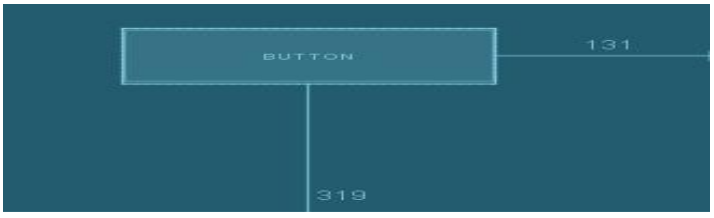


Fig no 8: Add a Button

3.2.3 CHANGE UI STRING

1.Open the Project window and then open app > res > values > strings.xml.

2.This is a string resources file, where you can specify all of your UI strings. It allows you to manage all of your UI strings in a single location, which makes them easier to find, update, and localize.

3. Click Open editor at the top of the window. This opens the Translations Editor, which provides a simple interface to add and edit your default strings. It also helps you keep all of your translated strings organized.

a. Click + (Add Key) to create a new string as the "hint text" for the text box. At this point, the window showed in figure 7 opens. a. Enter "edit message" in the Key field.

b. Enter "Enter a message" in the Default Value field.

c. Click ok.

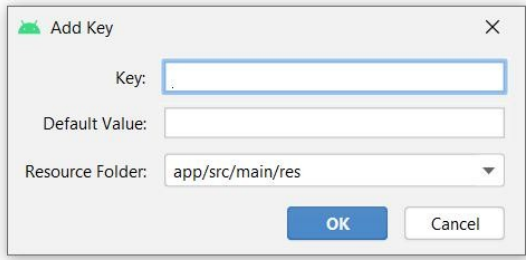


Fig no 9: Change UI String

3.2 IMPLEMENTATION

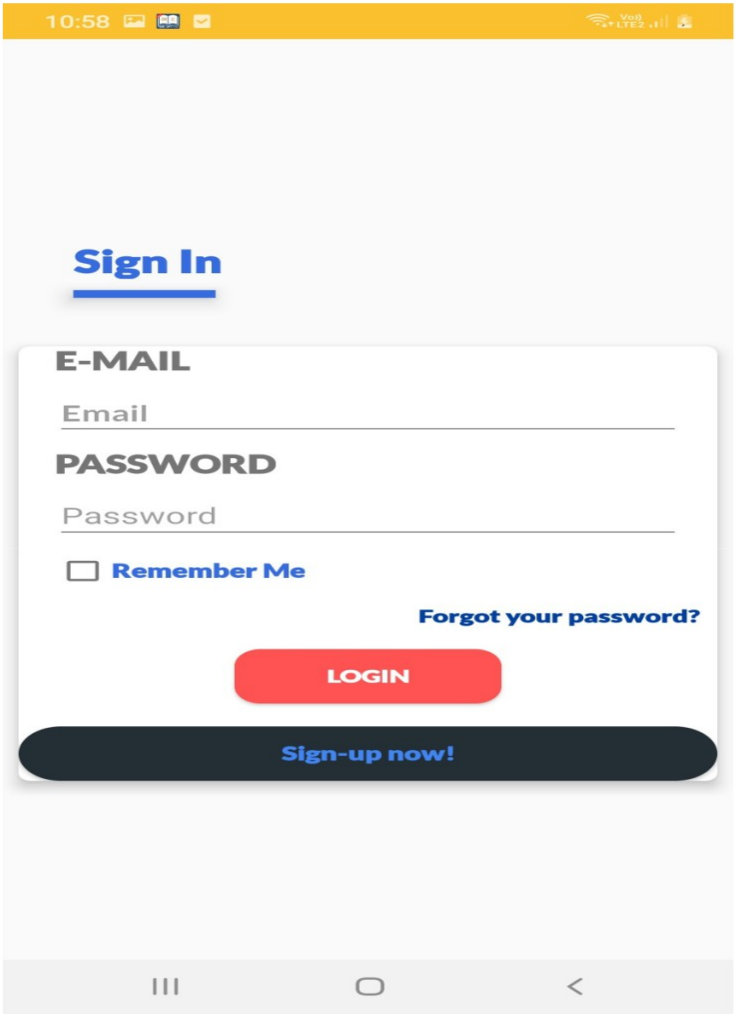
3.2.1 USER INTERFACE

For a login page in a Hotel Booking System application, you can design a user interface that includes the following components and features like,

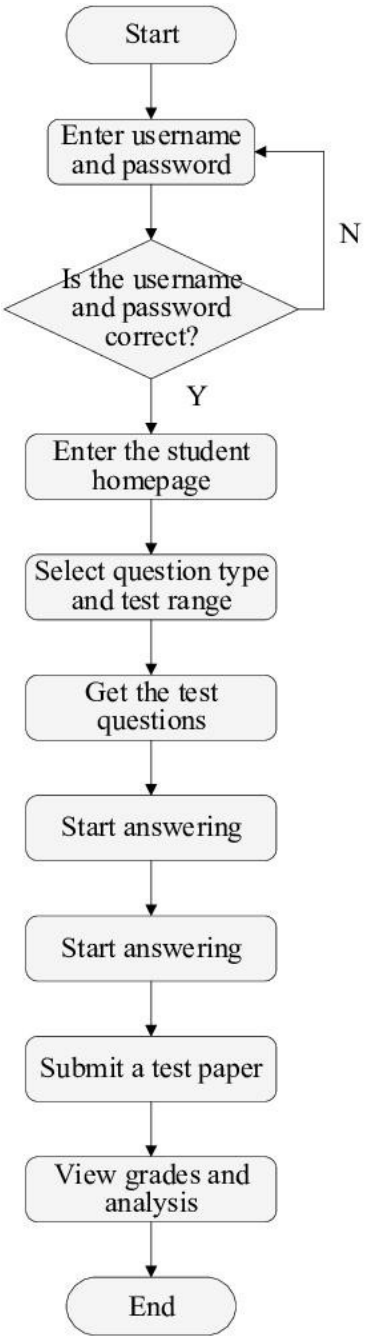
Input Fields: Include several input fields for users to enter their registration information. Common fields include:

- Email: Provide a field to enter a valid email address.
- Password: Include a password field for users to create a secure password and at last login button.

Android provides an XML vocabulary for View Group and View classes, so most of your UI isdefined in XML files. However, rather than teach you to write XML, this lesson shows you howto create a layout using Android Studio's Layout Editor. The Layout Editor writes the XML for you as you drag and drop views to build



3.2.2 Flow Chart



3.2.2 About the Functions used in Java

- **void onCreate(Bundle savedInstanceState)**

Android provides another elegant way of achieving this. To achieve this, we have to override a method called onSaveInstanceState(). Android platform allows the users to save any instance state. Instance state can be saved in the Bundle. Bundle is passed as argument to the onSaveInstanceState method.

We can load the saved instance state from the Bundle passed as argument to the onCreate method. We can also load the saved instance state in onRestoreInstanceState method. But I will leave that for the readers to figure out.

- **void basicNotify(View view)**

View is one of the most general class in Android. It holds references to single piece of UI. See link above. Passing View view as argument in methods in most cases gives you opportunity to call method associated with this view. For example if you click on some elements, view is passed in listener so you can know which view was clicked and what attributes it have. Please read Android documentation.

- **void useAppContext()**

useContext is a hook that allows you to access and consume a given Context in a React app.

- **void addition_isCorrect():**

checks for correct addition done by the particular code

- **void headsUpNotify(View view)**

There are multiple ways of how the notification should appear on screen and leave the screen. A set of default animator helper classes are provided and can be switched out by overriding config_headsUpNotificationAnimationHelper.

com.android.car.notification.headsup.animationhelper.CarHeadsUpNotificationTopAnimationHelper

- **expandableNotify(View view)**

enabling the user to get more data out of the notification system. This new ability is called expanded notification, watch the short tutorial below to know more about hoe to expand / collapse your notifications.

- **createNotificationChannel(String CHANNEL)**

Channel importance affects the interruption level of all notifications posted in the channel, and you must specify it in the NotificationChannel constructor. Creating an existing notification channel with its original values performs no operation, so it's safe to call this code when starting an app.

By default, all notifications posted to this channel use the visual and auditory behaviors defined by the importance level from the NotificationManagerCompat class

3.3.3 Source code

XML Code:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.test_quiz">

    <uses-permission android:name="android.permission.INTERNET" /> <uses-permission
    android:name="android.permission.READ_EXTERNAL_STORAGE"/> <uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"/> <uses-permission
    android:name="android.permission.ACCESS_WIFI_STATE" /> <uses-permission
    android:name="android.permission.ACCESS_NETWORK_STATE" /> <uses-permission
    android:name="android.permission.CAMERA" />
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
```

```
<uses-permission android:name="com.google.android.c2dm.permission.SEND"/>

<supports-screens android:resizeable="true"
    android:anyDensity="true"
    android:largeScreens="true"
    android:normalScreens="true"
    android:smallScreens="true"
    android:xlargeScreens="true">
</supports-screens>


<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher_round"
    android:label="Quizzer"
    tools:replace="label"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:usesCleartextTraffic="true"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">

    <activity
        android:name=".View.MainActivity"
        android:label="@string/app_name"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity
        android:name=".Auth_Controller.SignUp"
        android:theme="@style/AppTheme.NoActionBar" />

    <activity
        android:name=".Auth_Controller.LoginActivity"
        android:theme="@style/AppTheme.NoActionBar"/>
    <activity
        android:name=".Splash_Activity.SplashActivity"
        android:theme="@style/AppTheme.NoActionBar">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category
        android:name="android.intent.category.LAUNCHER" /> </intent-
        filter>
    </activity>


    <activity
        android:name=".Auth_Controller.ResetPasswordActivity"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity
```

```
    android:name=".Attempt_Quiz_Section.AttemptTest"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
    android:name=".Attempt_Quiz_Section.Tests"
    android:label="@string/title_activity_tests"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
    android:name=".View.AddDetails"
    android:label="@string/title_activity_UPdate"
    android:theme="@style/AppTheme.NoActionBar"/>
<activity
    android:name=".Results_section.ResultsAdmin"
    android:label="Results:ADMIN"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
    android:name=".Results_section.ResultsAdminDetailed"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
    android:name=".View.AboutUsActivity"
    android:label="About Us!">
</activity>

<activity
    android:name=".Splash_Activity.SplashAnimation"
    android:theme="@style/AppTheme.NoActionBar"/>

<activity android:name=".Splash_Activity.Splash_create_quiz"
    android:theme="@style/AppTheme.NoActionBar"/>

<activity
    android:name=".Create_Quiz.create_quiz_main"
    android:theme="@style/AppTheme.NoActionBar">
</activity>

<activity
    android:name=".Create_Quiz.Custom_quiz"
    android:theme="@style/Theme.AppCompat.NoActionBar">
</activity>

<activity
    android:name=".Chat_Section.Chat_Handler"
    android:label="Chat"
    android:theme="@style/AppTheme.NoActionBar">
</activity>

<service
    android:name=".NotificationActivity.MyFirebaseInstanceIDService">
```

```
        <intent-filter>
            <action
        android:name="com.google.firebase.INSTANCE_ID_EVENT" /> </intent-
        filter>
      </service>

      <service
        android:name=".NotificationActivity.MyFirebaseMessagingService">
        <intent-filter>
          <action android:name="com.google.firebase.MESSAGING_EVENT" />
          <action android:name="com.google.android.c2dm.intent.RECEIVE" />
        </intent-filter>
      </service>

      <service android:name=".NotificationActivity.NotificationService" />

      <activity android:name=".Results_section.GetDetailReport"
        android:label="Detailed Report"
        android:theme="@style/AppTheme.NoActionBar">
      </activity>

    </application>
</manifest>
```

JAVA Code:

MainActivity.java

```
package com.example.test_quiz.View;

import android.app.Dialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.ColorDrawable;
import android.net.ConnectivityManager; import
android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import android.util.DisplayMetrics;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
```

```
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.TextView;

import com.example.test_quiz.Attempt_Quiz_Section.Tests;
import com.example.test_quiz.Create_Quiz.create_quiz_main;
import com.example.test_quiz.Model.User;
import com.example.test_quiz.Results_section.ResultsAdmin;
import com.example.test_quiz.Splash_Activity.SplashAnimation;
import com.example.test_quiz.R;
import com.example.test_quiz.Auth_Controller.ResetPasswordActivity;
import com.example.test_quiz.Auth_Controller.LoginActivity; import
com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.navigation.NavigationView;
import com.google.firebase.auth.FirebaseAuth; import
com.google.firebase.auth.FirebaseUser; 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
com.google.firebase.storage.FirebaseStorage; import
com.google.firebase.storage.StorageReference; import
java.util.Objects;
import de.hdodenhof.circleimageview.CircleImageView;
import es.dmoral.toasty.Toasty;

public class MainActivity extends AppCompatActivity
        implements NavigationView.OnNavigationItemSelectedListener {

    private FirebaseDatabase database;
    private FirebaseAuth auth;
    private DatabaseReference myRef;
    public TextView USer_email;
    public ImageButton imageButton;
    private StorageReference firebaseStorage;
    private boolean isAdmin=false;
    private TextView userID;
    private CircleImageView imageView;
    public CircleImageView imageView1;
    final long ONE_MEGABYTE = 1024 * 1024;
    private FloatingActionButton floatingActionButton;

    @Override
```

```
protected void onCreate(Bundle savedInstanceState)
    { super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main); Toolbar
    toolbar = findViewById(R.id.toolbar);
    imageButton = findViewById(R.id.userImage2);
    setSupportActionBar(toolbar);
    imageView = findViewById(R.id.card1);
    floatingActionButton = findViewById(R.id.chatHead);
    auth= FirebaseAuth.getInstance();

    //get user image if it's available
                                                                    firebaseStorage =
FirebaseStorage.getInstance().getReference(Objects.requireNonNull(auth.getUid()));
    firebaseStorage.getBytes(ONE_MEGABYTE)
        .addOnSuccessListener(new OnSuccessListener<byte[]>()
          { @Override
          public void onSuccess(byte[] bytes) {
            Bitmap bm = BitmapFactory.decodeByteArray(bytes, 0,
            bytes.length); DisplayMetrics dm = new DisplayMetrics();
            getWindowManager().getDefaultDisplay().getMetrics(dm);

            imageView.setMinimumHeight(150);
            imageView.setMinimumWidth(150);
            imageView.setMaxHeight(150);
            imageView.setMaxWidth(150);
            imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
            imageView.setImageBitmap(bm);
          }
        }).addOnFailureListener(new OnFailureListener() {
      @Override
      public void onFailure(@NonNull Exception e) {

      }
    });
    //fragment for term & conditions!
    imageButton.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        BottomSheetFragment sheetFragment = new BottomSheetFragment();
        sheetFragment.show(getSupportFragmentManager(),sheetFragment.getTag());
      }
    });

    auth = FirebaseAuth.getInstance();
    database= FirebaseDatabase.getInstance();
    myRef=database.getReference();
```

```
        /*check if user is admin or not**/
        checkForAdmin();

        DrawerLayout drawer = findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                                        this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();

        /*set profile_pic of user on navigation drawer**/
        NavigationView navigationView = findViewById(R.id.nav_view);
        View header = navigationView.getHeaderView(0);
        imageView1 = (navigationView.getHeaderView(0)).findViewById(R.id.imageView);
        USer_email = header.findViewById(R.id.text_user_name);
        setTextOnUser();
        navigationView.setNavigationItemSelectedListener(this);
        userID = findViewById(R.id.text_user_card);
        setUserEmail();
        setImageOnNavHeader();

        floatingActionButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,
                SplashAnimation.class); intent.putExtra("ChatAdmin",isAdmin);
                startActivity(intent);
            }
        });
    }

    public void checkForAdmin() {

        myRef.child("admins").addListenerForSingleValueEvent(new ValueEventListener()
            { @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot)
                { if(dataSnapshot.child(Objects.requireNonNull(auth.getUid()))
                        .exists()&& Objects.requireNonNull(dataSnapshot.child(auth.getUid())
                        .getValue()).toString().equals("true")){
                    isAdmin=true;
                                                Toasty.info(getApplicationContext(),"Hello  Admin",
Toasty.LENGTH_LONG).show();
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {
```

```
        }
    });
}

public void setImageOnNavHeader() {

                                                                    firebaseStorage =
FirebaseStorage.getInstance().getReference(Objects.requireNonNull(auth.getUid()));
    firebaseStorage.getBytes(ONE_MEGABYTE)
            .addOnSuccessListener(new OnSuccessListener<byte[]>()
              { @Override
              public void onSuccess(byte[] bytes) {
                 Bitmap bm = BitmapFactory.decodeByteArray(bytes, 0,
                 bytes.length); DisplayMetrics dm = new DisplayMetrics();
                 getWindowManager().getDefaultDisplay().getMetrics(dm);

                 imageView1.setMinimumHeight(90);
                 imageView1.setMinimumWidth(90);
                 imageView1.setMaxHeight(100);
                 imageView1.setMaxWidth(100);
                 imageView1.setScaleType(ImageView.ScaleType.CENTER_CROP);
                 imageView1.setImageBitmap(bm);
              }
            }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {

        }
    });
}

@Override
public void onBackPressed() {
    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    /  Inflate the menu; this adds items to the action bar if it is
    present. getMenuInflater().inflate(R.menu.main, menu);
    return true;
```

```
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        /  Handle action bar item clicks here. The action bar will
        /  automatically handle clicks on the Home/Up button, so long
        /  as you specify a parent activity in
        AndroidManifest.xml. //noinspection
        SimplifiableIfStatement
        return super.onOptionsItemSelected(item);
    }

    @SuppressWarnings("")
    @Override
    public boolean onNavigationItemSelected(MenuItem item) {

        /  Handle navigation view item clicks
            here. int id = item.getItemId();

            if (id == R.id.nav_test) {
                if (isNetworkAvailable(MainActivity.this)) { startActivity(new
                    Intent(MainActivity.this, Tests.class));
                    overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
                }
                else
                    alertNoConnection();
            } else if (id == R.id.nav_result) {
                if ( isNetworkAvailable(MainActivity.this)) {
                    Intent intent = new Intent(MainActivity.this, ResultsAdmin.class);
                    intent.putExtra("ISADMIN",isAdmin);
                    startActivity(intent);
                    overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
                }
                else
                    alertNoConnection();
            } else if (id == R.id.create_test) {
                if (isAdmin && isNetworkAvailable(MainActivity.this)) { startActivity(new
                    Intent(MainActivity.this, create_quiz_main.class));
                    overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
                }
                else if (isNetworkAvailable(MainActivity.this))
                                        Toasty.error(getApplicationContext(), "You are not Admin!",
Toasty.LENGTH_SHORT).show();
                else
                    alertNoConnection();
            } else if (id == R.id.nav_respass) {
                if (isNetworkAvailable(MainActivity.this)) {
                    startActivity(new Intent(MainActivity.this, ResetPasswordActivity.class));
```

```
            overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
          }
          else
            alertNoConnection();
        } else if (id == R.id.nav_signout)
          { auth.signOut();
          startActivity(new Intent(MainActivity.this,
          LoginActivity.class)); finish();
          overridePendingTransition(android.R.anim.fade_in,
        android.R.anim.fade_out); } else if (id == R.id.nav_details) {
          startActivity(new Intent(MainActivity.this, AddDetails.class));
          overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
        } else if (id == R.id.about_details) {
          startActivity(new Intent(MainActivity.this, AboutUsActivity.class));
          overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
        } else if (id == R.id.feedback_id) {

          Intent intent = new Intent(Intent.ACTION_SEND);
          intent.setType("message/rfc822");
          intent.putExtra(Intent.EXTRA_EMAIL, new String[]{"Enter your email here"});
                intent.putExtra(Intent.EXTRA_SUBJECT, "Regarding Your Test or Application
Feedback");
          intent.putExtra(Intent.EXTRA_TEXT, "Put your subject
          here!"); try {
            startActivity(Intent.createChooser(intent, "Send mail...")); }
          catch (android.content.ActivityNotFoundException ex) {
                  Toasty.error(getApplicationContext(), "There are no email clients installed.",
Toasty.LENGTH_SHORT).show();
          }
        }

      DrawerLayout drawer = findViewById(R.id.drawer_layout);
      drawer.closeDrawer(GravityCompat.START);
      return true;
    }


    //set textView on MainActivity
    public void setUserEmail() {
        DatabaseReference mDatabase;
        FirebaseAuth auth;
        mDatabase = FirebaseDatabase.getInstance().getReference();
        auth= FirebaseAuth.getInstance();
       mDatabase.child("users").child(Objects.requireNonNull(auth.getUid()))
            .addValueEventListener(new ValueEventListener() {
          @Override
          public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
```

```
            if(dataSnapshot.exists()){
                User user=dataSnapshot.getValue(User.class);
                String temp = "Hey "+ user.name +" what's up!";
                userID.setText(temp);
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });

}

public void setTextOnUser(){
    FirebaseUser usero = FirebaseAuth.getInstance().getCurrentUser();
    USer_email.setText(Objects.requireNonNull(usero).getEmail());
}

/*method to handle network connection**/
public boolean isNetworkAvailable(Context context) {
    ConnectivityManager connectivityManager = ((ConnectivityManager) context
            .getSystemService(Context.CONNECTIVITY_SERVICE));
    return connectivityManager.getActiveNetworkInfo() != null
            && connectivityManager.getActiveNetworkInfo().isConnected();
}

public void alertNoConnection() {

    /*final AlertDialog.Builder builder = new
    AlertDialog.Builder(MainActivity.this); builder.setIcon(R.drawable.nowifi);
    builder.setCancelable(true);
    builder.setTitle("No Connection Available!");
    builder.setPositiveButton("Ok", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which)
            { dialog.dismiss();
        }
    });
    builder.show();*/

    Dialog builder = new Dialog(this);
    builder.requestWindowFeature(Window.FEATURE_NO_TITLE);
    builder.getWindow().setBackgroundDrawable(
            new ColorDrawable(android.graphics.Color.TRANSPARENT));
```

```
        builder.setOnDismissListener(new DialogInterface.OnDismissListener() {
            @Override
            public void onDismiss(DialogInterface dialogInterface) {
                //nothing;
            }
        });

        ImageView imageView = new ImageView(this);
        imageView.setImageResource(R.drawable.nowifi);
/         imageView.getLayoutParams().height = 100;
/         imageView.getLayoutParams().width = 100;
/         imageView.requestLayout();
        //imageView.setImage(R.drawable.nowifi);
        builder.addContentView(imageView, new RelativeLayout.LayoutParams(
                400,
                400));
        builder.show();
    }

}
```

Create Quiz

```
package com.example.test_quiz.Create_Quiz;

import android.content.Intent;
import android.os.Bundle;
import androidx.annotation.Nullable;
import
androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar; import
android.view.MenuItem; import android.view.View;

import com.example.test_quiz.R;
import com.example.test_quiz.Splash_Activity.Splash_create_quiz;
import com.google.android.material.floatingactionbutton.FloatingActionButton;

import java.util.Objects;

public class create_quiz_main extends AppCompatActivity {

    private FloatingActionButton fab1;
    private Toolbar toolbar;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState)
        { super.onCreate(savedInstanceState);
        setContentView(R.layout.content_main_1);
```

```
        toolbar = findViewById(R.id.toolbartst);
        toolbar.setTitleTextColor(getResources().getColor(android.R.color.black));
        setSupportActionBar(toolbar);
        Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);
        fab1 = findViewById(R.id.fab);
        fab1.setOnClickListener(new View.OnClickListener()
            { @Override
            public void onClick(View v) {
                startActivity(new Intent(create_quiz_main.this,
                Splash_create_quiz.class)); finish();
            }
        });
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

        if(item.getItemId() == android.R.id.home) {
            finish();
        }
        return super.onOptionsItemSelected(item);
    }
}}
```

Auth_Controller

```
package com.example.test_quiz.Auth_Controller;

import android.content.Intent;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar; import
android.text.TextUtils; import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import com.example.test_quiz.View.MainActivity;
import com.example.test_quiz.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task; import
com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.wang.avi.AVLoadingIndicatorView;

import es.dmoral.toasty;
```

```
public class LoginActivity extends AppCompatActivity {

    private EditText inputEmail, inputPassword;
    private FirebaseAuth auth;
    private AVLoadingIndicatorView avLoadingIndicatorView;
    private Button btnSignup, btnLogin, btnReset;


    @Override
    protected void onCreate(Bundle savedInstanceState)
        { super.onCreate(savedInstanceState);

        //Get Firebase auth instance
        auth = FirebaseAuth.getInstance();

        if (auth.getCurrentUser() != null) {

            if(auth.getCurrentUser().isEmailVerified()) {
                Intent intent = new Intent(LoginActivity.this, MainActivity.class);
                startActivity(intent);
                overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
                finish();
            }else {
                Toasty.warning(getApplicationContext(), R.string.email_unverified,
Toasty.LENGTH_SHORT).show();
                FirebaseAuth.getInstance().signOut();
                finish();
            }
        }

        /  set the view now
        setContentView(R.layout.activity_login);

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        inputEmail = findViewById(R.id.email); inputPassword
        = findViewById(R.id.password);
        avLoadingIndicatorView = findViewById(R.id.loader1);
        btnSignup = findViewById(R.id.btn_signup); btnLogin
        = findViewById(R.id.btn_login); btnReset =
        findViewById(R.id.btn_reset_password);

        //Get Firebase auth instance
        auth = FirebaseAuth.getInstance();
```

```
        btnSignup.setOnClickListener(new View.OnClickListener()
            { @Override
            public void onClick(View v) {
                startActivity(new Intent(LoginActivity.this, SignUp.class));
                overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
            }
        });

        btnReset.setOnClickListener(new View.OnClickListener()
            { @Override
            public void onClick(View v) {
                startActivity(new Intent(LoginActivity.this, ResetPasswordActivity.class));
                overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
            }
        });

        btnLogin.setOnClickListener(new View.OnClickListener()
            { @Override
            public void onClick(View v) {
                String email = inputEmail.getText().toString();
                final String password = inputPassword.getText().toString();

                if (TextUtils.isEmpty(email)) {
                    Toasty.warning(getApplicationContext(),"Enter email address!",
Toasty.LENGTH_SHORT).show();
                    return;
                }

                if (TextUtils.isEmpty(password))
                    { Toasty.warning(getApplicationContext(),"Enter password!",
Toasty.LENGTH_SHORT).show();
                    return;
                }

                avLoadingIndicatorView.setVisibility(View.VISIBLE);
                avLoadingIndicatorView.smoothToShow();

                //authenticate user
                auth.signInWithEmailAndPassword(email, password)
                        .addOnCompleteListener(LoginActivity.this,
new OnCompleteListener<AuthResult>() {
                            @Override
                            public void onComplete(@NonNull Task<AuthResult> task) {
                                /  If sign in fails, display a message to the user. If sign in succeeds
                                /  the auth state listener will be notified and logic to handle the
```

```
                        /  signed in user can be handled in the listener.
//progressBar.setVisibility(View.GONE);
                        avLoadingIndicatorView.setVisibility(View.GONE);
                        avLoadingIndicatorView.smoothToHide();
                        if (!task.isSuccessful()) {
                            /  there was an error
                            if (password.length() < 6) {
                                inputPassword.setError(getString(R.string.minimum_password));
                            } else {
                                Toasty.warning(LoginActivity.this,getString(R.string.auth_failed),
Toasty.LENGTH_LONG).show();
                            }
                        } else {
                            if(auth.getCurrentUser().isEmailVerified()) {
                                startActivity(new Intent(LoginActivity.this, MainActivity.class));
                                overridePendingTransition(android.R.anim.fade_in,
android.R.anim.fade_out);
                                finish();
                            }else {
                                Toasty.error(LoginActivity.this,R.string.email_unverified,
Toasty.LENGTH_SHORT).show();
                                FirebaseAuth.getInstance().signOut();
                            }

                        }
                    }
                });
        }
    });
  }
}
package com.example.test_quiz.Auth_Controller;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.text.TextUtils; import
android.view.View;
import android.widget.Button;
import android.widget.EditText;

import com.example.test_quiz.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task; import
com.google.firebase.auth.FirebaseAuth;
import com.wang.avi.AVLoadingIndicatorView;

import es.dmoral.toasty.Toasty;
```

```
public class ResetPasswordActivity extends AppCompatActivity {

    private EditText inputEmail;
    private Button btnReset, btnBack;
    private FirebaseAuth auth;
    private AVLoadingIndicatorView avLoadingIndicatorView;

    @Override

    protected void onCreate(Bundle savedInstanceState)
        { super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_reset_password);

        inputEmail = findViewById(R.id.email);
        btnReset = findViewById(R.id.btn_reset_password);
        btnBack = findViewById(R.id.btn_back);
        avLoadingIndicatorView = findViewById(R.id.loader1);

        auth = FirebaseAuth.getInstance();

        btnBack.setOnClickListener(new View.OnClickListener()
            { @Override
            public void onClick(View v) {
                finish();
            }
        });

        btnReset.setOnClickListener(new View.OnClickListener()
            { @Override
            public void onClick(View v) {

                String email = inputEmail.getText().toString().trim();

                if (TextUtils.isEmpty(email)) {
                    Toasty.info(getApplication(),"Enter your registered email id",
                            Toasty.LENGTH_SHORT).show();
                    return;
                }

                avLoadingIndicatorView.setVisibility(View.VISIBLE);
                avLoadingIndicatorView.show();
                auth.sendPasswordResetEmail(email)
                        .addOnCompleteListener(new OnCompleteListener<Void>() {
                            @Override
                            public void onComplete(@NonNull Task<Void> task)
                                { if (task.isSuccessful()) {
                                    Toasty.success(ResetPasswordActivity.this,"We have sent you instructions
```

```
to reset your password!",
                                    Toasty.LENGTH_SHORT).show();
                        } else {
                            Toasty.success(ResetPasswordActivity.this,"Failed to send reset
                                email!", Toasty.LENGTH_SHORT).show();
                        }

                        avLoadingIndicatorView.setVisibility(View.GONE);
                        avLoadingIndicatorView.hide();
                    }
                });
            }
        });
    }

}
package com.example.test_quiz.Auth_Controller;
import android.content.Intent;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.text.TextUtils; import
android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.example.test_quiz.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task; import
com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.wang.avi.AVLoadingIndicatorView;

import java.util.Objects;

import es.dmoral.toasty.Toasty;

public class SignUp extends AppCompatActivity {

    private EditText inputEmail, inputPassword;
    private Button btnSignIn, btnSignUp, btnResetPassword;
    private AVLoadingIndicatorView avLoadingIndicatorView;
    private FirebaseAuth auth;
    @Override
    protected void onCreate(Bundle savedInstanceState)
        { super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_signup);

//Get Firebase auth instance
auth = FirebaseAuth.getInstance();

btnSignIn = findViewById(R.id.sign_in_button); btnSignUp =
findViewById(R.id.sign_up_button); inputEmail =
findViewById(R.id.email); inputPassword =
findViewById(R.id.password); avLoadingIndicatorView =
findViewById(R.id.loader1); btnResetPassword =
findViewById(R.id.btn_reset_password);


btnResetPassword.setOnClickListener(new View.OnClickListener()
   { @Override
   public void onClick(View v) {
      startActivity(new Intent(SignUp.this, ResetPasswordActivity.class));
      overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
   }
});

btnSignIn.setOnClickListener(new View.OnClickListener()
   { @Override
   public void onClick(View v) {
      overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
      finish();
   }
});

btnSignUp.setOnClickListener(new View.OnClickListener()
   { @Override
   public void onClick(View v) {

      String email = inputEmail.getText().toString().trim();
      String password = inputPassword.getText().toString().trim();

      if (TextUtils.isEmpty(email)) {
         inputEmail.setError("Enter valid Email");
         return;
      }

      if (TextUtils.isEmpty(password))
         { inputPassword.setError("Password should be atleast length of
         6!"); return;
      }

      if (password.length() < 6) {
```

```
                Toast.makeText(getApplicationContext(), "Password too short, enter minimum 6
characters!", Toast.LENGTH_SHORT).show();
                return;
            }

            avLoadingIndicatorView.setVisibility(View.VISIBLE);
            avLoadingIndicatorView.smoothToShow();
            //create user
            auth.createUserWithEmailAndPassword(email, password)
                  .addOnCompleteListener(SignUp.this, new OnCompleteListener<AuthResult>()
                     { @Override
                     public void onComplete(@NonNull Task<AuthResult> task)
                        { Toast.makeText(SignUp.this, "createUserWithEmail:onComplete:" +
task.isSuccessful(), Toast.LENGTH_SHORT).show();
                        avLoadingIndicatorView.setVisibility(View.GONE);
                        avLoadingIndicatorView.smoothToHide();
                        /  If sign in fails, display a message to the user. If sign in succeeds
                        /  the auth state listener will be notified and logic to handle the
                        /  signed in user can be handled in the listener.
                        if (!task.isSuccessful()) {
                           Toasty.error(SignUp.this,"Authentication failed."+ task.getException(),
                                 Toasty.LENGTH_SHORT).show();
                        } else {
                           Objects.requireNonNull(auth.getCurrentUser()).sendEmailVerification();

Toasty.success(SignUp.this,R.string.email_sent,Toasty.LENGTH_SHORT).show();
                        }
                     }
                  });

         }
      });
   }

   @Override
   protected void onResume() {
      super.onResume();
      avLoadingIndicatorView.setVisibility(View.GONE);
      avLoadingIndicatorView.hide();
   }
}
```

Results Section

```
package com.example.test_quiz.Results_section;

import android.graphics.Bitmap;
```

```
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.util.DisplayMetrics;
import android.view.MenuItem;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

import com.example.test_quiz.R;
import      com.google.android.gms.tasks.OnFailureListener;
import     com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;

import java.util.Objects;

import de.hdodenhof.circleimageview.CircleImageView;

public class GetDetailReport extends AppCompatActivity {

    CircleImageView circleImageView;
    TextView textView1;
    TextView textView2;
    TextView textView3;
    TextView textView4;
    TextView textView5;
    private FirebaseDatabase database;
    private FirebaseAuth auth;
    private DatabaseReference myRef;
    private StorageReference firebaseStorage;
    final long ONE_MEGABYTE = 1024 * 1024;


    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.complete_student_details);
        Toolbar toolbar = findViewById(R.id.toolbartst);
        setSupportActionBar(toolbar);
        toolbar.setTitleTextColor(getResources().getColor(android.R.color.black));
```

```
    Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);
    circleImageView = findViewById(R.id.USERIMAGE);
    textView1 = findViewById(R.id.StudentID);
    textView2 = findViewById(R.id.SemID); textView3
    = findViewById(R.id.Branchid); textView4 =
    findViewById(R.id.SectionID); textView5 =
    findViewById(R.id.this_Section); String temp =
    getIntent().getStringExtra("USERID");

    firebaseStorage = FirebaseStorage.getInstance().getReference().child(temp);
    firebaseStorage.getBytes(ONE_MEGABYTE)
            .addOnSuccessListener(new OnSuccessListener<byte[]>()
               { @Override
               public void onSuccess(byte[] bytes) {
                  Bitmap bm = BitmapFactory.decodeByteArray(bytes, 0,
                  bytes.length); DisplayMetrics dm = new DisplayMetrics();
                  getWindowManager().getDefaultDisplay().getMetrics(dm);

                  circleImageView.setMinimumHeight(100);
                  circleImageView.setMinimumWidth(100);
                  circleImageView.setMaxHeight(100); circleImageView.setMaxWidth(100);
                  circleImageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
                  circleImageView.setImageBitmap(bm);

               }
            }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {

        }
    });

    textView1.setText(getIntent().getStringExtra("DetailID"));
    textView2.setText(getIntent().getStringExtra("DetailBranch"));
    textView3.setText(getIntent().getStringExtra("DetailSem"));
    textView4.setText(getIntent().getStringExtra("DetailSec"));
    String Temp = getIntent().getStringExtra("TestNAME") + " results is: "
            + getIntent().getStringExtra("Marks");
    textView5.setText(Temp);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {

    if(item.getItemId() == android.R.id.home) {
        finish();
    }
```

```
        return super.onOptionsItemSelected(item);
    }
}
```

Attempt Quiz Section

```
package com.example.test_quiz.Attempt_Quiz_Section;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Point;
import android.os.Bundle;
import android.os.CountDownTimer;
import androidx.annotation.ColorInt;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.recyclerview.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.TranslateAnimation;
import android.widget.AdapterView; import
android.widget.ArrayAdapter;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.GridView;
import android.widget.LinearLayout;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;

import com.example.test_quiz.NotificationActivity.NotificationService;
import com.example.test_quiz.R;
import com.example.test_quiz.Model.Question;
import com.example.test_quiz.Model.Test;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.yarolegovich.discretescrollview.DiscreteScrollView;

import org.jetbrains.annotations.NotNull;
```

```
import java.util.ArrayList;
import java.util.concurrent.TimeUnit;

import es.dmoral.toasty.Toasty;

public class AttemptTest extends AppCompatActivity {
    ArrayList<Question> questions;
    String []answers;
    Toolbar toolbar;
    DiscreteScrollView scrollView;
    LinearLayout indexLayout;
    GridView quesGrid;
    ArrayList<String> list;
    ArrayList<String> arrayList;
    int flag_controller = 1;
    long timer;// =((Test) getIntent().getExtras().get("Questions")).getTime()*60*1000;
    popGridAdapter popGrid;
    Button next,prev;
    TextView textView;
    private DatabaseReference mDatabase;
    private FirebaseAuth auth;
    private String TESTNAME;
    private RadioGroup group;
    private int countPaused = 0;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState)
        { super.onCreate(savedInstanceState);
        mDatabase = FirebaseDatabase.getInstance().getReference();
        auth= FirebaseAuth.getInstance();
        setContentView(R.layout.activity_attempt);
        questions=((Test) getIntent().getExtras().get("Questions")).getQuestions();
        TESTNAME = (String) getIntent().getExtras().get("TESTNAME");
        toolbar=findViewById(R.id.toolbar);
        toolbar.setTitleTextColor(getResources().getColor(android.R.color.black));
        answers=new String[questions.size()];
        setSupportActionBar(toolbar);
        toolbar.setTitleTextColor(getResources().getColor(android.R.color.black));
        scrollView = findViewById(R.id.discrete);
        final QuestionAdapter questionAdapter=new QuestionAdapter(questions);
        scrollView.setAdapter(questionAdapter);

        next=findViewById(R.id.next);
        next.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view)
                { if(scrollView.getCurrentItem()==questions.size()-1){
```

```
                showPopUp();
            }else {
                //setNextPrevButton(scrollView.getCurrentItem() + 1);
                scrollView.smoothScrollToPosition(scrollView.getCurrentItem() + 1);
            }
        }
    });

    prev=findViewById(R.id.prev);
    prev.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(scrollView.getCurrentItem()!=0){
                //setNextPrevButton(scrollView.getCurrentItem()-1);
                scrollView.smoothScrollToPosition(scrollView.getCurrentItem()-1);
            }
        }
    });

    setNextPrevButton(scrollView.getCurrentItem());
    indexLayout=findViewById(R.id.index_layout);
    indexLayout.setAlpha(.5f);
    quesGrid=findViewById(R.id.pop_grid);
    popGrid=new popGridAdapter(AttemptTest.this);
    quesGrid.setAdapter(popGrid);
    quesGrid.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
            scrollView.smoothScrollToPosition(i+1);
            slideUp(indexLayout);
        }
    });
    scrollView.addScrollListener(new
DiscreteScrollView.ScrollListener<RecyclerView.ViewHolder>() {
        @Override
        public void onScroll(float scrollPosition, int currentPosition, int newPosition, @Nullable
RecyclerView.ViewHolder currentHolder, @Nullable RecyclerView.ViewHolder newCurrent) {
            setNextPrevButton(newPosition);
        }
    });

    timer=((Test) getIntent().getExtras().get("Questions")).getTime()*60*1000;
  }
  void showPopUp(){
    AlertDialog.Builder builder=new AlertDialog.Builder(AttemptTest.this);
    builder.setMessage("Do you want to submit?");
    builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
```

```
        @Override
        public void onClick(DialogInterface dialogInterface, int i)
            { submit();
/             setAlertDialog(answerText);
            dialogStart();
        }
    });

    builder.setNegativeButton("No", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i)
            { dialogInterface.cancel();
        }
    });
    builder.show();

}

/*submit result to database**/
void submit(){
    flag_controller = 0;
    int score=0;
    list = new ArrayList<>();
    arrayList = new ArrayList<>();
    for(int i=0;i<answers.length;i++){
        if(answers[i]!=null&&answers[i].equals(questions.get(i).getAnswer()))
            { score++;
        }
        String temp = (answers[i]!=null) ? answers[i]+") ":"null) ";

        list.add("Your choice ("+
                temp +
                "Right choice is("+ questions.get(i).getAnswer()+")");
        arrayList.add(questions.get(i).getQuestion());
    }

    try {
        mDatabase.child("Results").child(((Test)
getIntent().getExtras().get("Questions")).getName())
                .child(auth.getUid()).setValue(score);
    }catch (Exception e){
        Log.e("Result Update Failed " ,e.getMessage());
    }
}

void dialogStart() {
```

```
        final AlertDialog.Builder builderSingle = new AlertDialog.Builder(AttemptTest.this);
        builderSingle.setIcon(R.mipmap.ic_launcher_round);
        builderSingle.setTitle(TESTNAME+" Answers");
        final ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>
                (AttemptTest.this, android.R.layout.select_dialog_singlechoice);
        final ArrayAdapter<String> arrayAdapter1 = new ArrayAdapter<>
                (AttemptTest.this,android.R.layout.select_dialog_singlechoice);

        for(String y : arrayList) {
           arrayAdapter1.add(y);
        }
        for(String x: list){
           arrayAdapter.add(x);
        }

        builderSingle.setCancelable(false);
        builderSingle.setNegativeButton("Done!", new DialogInterface.OnClickListener() {
           @Override
           public void onClick(DialogInterface dialog, int which)
              { finish();
              dialog.dismiss();
           }
        });

        builderSingle.setAdapter(arrayAdapter1, new DialogInterface.OnClickListener() {
           @Override
           public void onClick(DialogInterface dialog, int which)
              { String strName = arrayAdapter.getItem(which);
              AlertDialog.Builder builderInner = new AlertDialog.Builder(AttemptTest.this);
              builderInner.setMessage(strName);
              builderInner.setCancelable(false);
              builderInner.setTitle("Your Selected Question Answer is");
              builderInner.setPositiveButton("Ok", new DialogInterface.OnClickListener() {
                 @Override
                 public void onClick(DialogInterface dialog,int which) {
//                    finish();
                    builderSingle.show();
//                    dialog.dismiss();
                 }
              });
              builderInner.show();
           }
        });
        builderSingle.show();

    }
```

```
    @Override
    protected void onPause() {
        super.onPause();
        if(countPaused<=2 && countPaused >=0 && flag_controller == 1)
            startService(new Intent(AttemptTest.this,
                    NotificationService.class));
        countPaused++;
    }

    @Override
    protected void onResume() {
        super.onResume();
        stopService(new Intent(AttemptTest.this, NotificationService.class));
        if(countPaused>2) {
            Toasty.success(AttemptTest.this,"Thank you! Your response has been
                    submitted.", Toasty.LENGTH_SHORT).show();
            countPaused = -1000;
            submit();
            dialogStart();
        }
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        stopService(new Intent(AttemptTest.this, NotificationService.class));
}

    void setNextPrevButton(int pos){
        if(pos==0){
/           prev.setBackgroundColor(getResources().getColor(android.R.color.darker_gray));
            prev.setText("");
        }else {
/           prev.setBackgroundColor(getResources().getColor(R.color.colorPrimaryDark));
            prev.setText("Previous");
        }
        if(pos==questions.size()-1){
            next.setText("Submit");
/       next.setBackgroundColor(getResources().getColor(android.R.color.holo_red_dark)); }else {
            next.setText("Next");
/             next.setBackgroundColor(getResources().getColor(R.color.colorPrimaryDark));
        }
    }

    @Override
    public void onBackPressed() {
```

```
        showPopUp();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

        super.onCreateOptionsMenu(menu);

        getMenuInflater().inflate(R.menu.attempt_menu, menu);
        final MenuItem counter = menu.findItem(R.id.counter);

        new CountDownTimer(timer, 1000) {
            public void onTick(long millisUntilFinished) {
                long millis = millisUntilFinished;
                long
hr=TimeUnit.MILLISECONDS.toHours(millis),mn=(TimeUnit.MILLISECONDS.toMinutes(milli
s)-
                        TimeUnit.HOURS.toMinutes(TimeUnit.MILLISECONDS.toHours(millis))),
                        sc=TimeUnit.MILLISECONDS.toSeconds(millis) -

TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millis));


                String hms =format(hr)+":"+format(mn)+":"+format(sc) ;
                counter.setTitle(hms);
                timer = millis;
            }
            String format(long n){
                if(n<10)
                    return "0"+n;
                else return ""+n;
            }

            public void onFinish() {
                submit();
                dialogStart();
            }
        }.start();

        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if(id==R.id.submit){
            showPopUp();
```

```
        return true;
    }else if(id==R.id.info){
        togglePopUp();
    }
    return super.onOptionsItemSelected(item);
}


void togglePopUp(){
    if(indexLayout.getVisibility()==View.GONE){
        slideDown(indexLayout);
    }else slideUp(indexLayout);
}

class QuestionAdapter extends RecyclerView.Adapter<QuestionAdapter.ViewHolder> {

    private int itemHeight;
    private ArrayList<Question> data;

    QuestionAdapter(ArrayList<Question> data) {
        this.data = data;
    }


    @Override
    public void onAttachedToRecyclerView(RecyclerView recyclerView) {
        super.onAttachedToRecyclerView(recyclerView);
        Activity context = (Activity) recyclerView.getContext();
        Point windowDimensions = new Point();
        context.getWindowManager().getDefaultDisplay().getSize(windowDimensions);
        itemHeight = Math.round(windowDimensions.y * 0.6f);
    }

    @NotNull
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType)
        { LayoutInflater inflater = LayoutInflater.from(parent.getContext());
        View v = inflater.inflate(R.layout.frag_test, parent, false);
        ViewGroup.LayoutParams params = new ViewGroup.LayoutParams(
                ViewGroup.LayoutParams.MATCH_PARENT,
                itemHeight);
        v.setLayoutParams(params);
        return new ViewHolder(v);
    }

    @Override
```

```
    public void onBindViewHolder(final ViewHolder holder, final int position)
        { holder.questionText.setText(data.get(position).getQuestion());
        holder.r1.setText(data.get(position).getOpt_A());
        holder.r2.setText(data.get(position).getOpt_B());
        holder.r3.setText(data.get(position).getOpt_C());
        holder.r4.setText(data.get(position).getOpt_D()); holder.r5.setText("Clear
        Selected");

        holder.radioGroup.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup radioGroup, int i) {
                final int selectedId = holder.radioGroup.getCheckedRadioButtonId();
                if(i==R.id.radioButton){
                    answers[position]="A";
                } else if(i==R.id.radioButton2){
                    answers[position]="B";
                }else if(i==R.id.radioButton3){
                    answers[position]="C";
                }else if(i==R.id.radioButton4){
                    answers[position]="D";
                }
                else if(i==R.id.radioButton5) {
                    holder.radioGroup.clearCheck();
                    answers[position] = null;
                }
                popGrid.notifyDataSetChanged();
            }
        });


        if(answers[position]==null) {
            holder.radioGroup.clearCheck();
        }else if(answers[position].equals("A")) {
            holder.radioGroup.check(R.id.radioButton);
        }else if(answers[position].equals("B")) {
            holder.radioGroup.check(R.id.radioButton2);
        }else if(answers[position].equals("C")) {
            holder.radioGroup.check(R.id.radioButton3);
        }else if(answers[position].equals("D")) {
            holder.radioGroup.check(R.id.radioButton4);
        }
    }

    @Override
    public int getItemCount() {
        return data.size();
```

```
    }

    class ViewHolder extends RecyclerView.ViewHolder {

        private View overlay;
        private TextView questionText;
        private RadioGroup radioGroup;
        private RadioButton r1,r2,r3,r4,r5;

        ViewHolder(View itemView) {
            super(itemView);
            questionText = itemView.findViewById(R.id.questionTextView);
            radioGroup=itemView.findViewById(R.id.radioGroup);
            r1=itemView.findViewById(R.id.radioButton);
            r2=itemView.findViewById(R.id.radioButton2);
            r3=itemView.findViewById(R.id.radioButton3);
            r4=itemView.findViewById(R.id.radioButton4); r5 =
            itemView.findViewById(R.id.radioButton5);
        }

        public void setOverlayColor(@ColorInt int color)
            { overlay.setBackgroundColor(color);
        }

        public void unCheck() {

            int selectedId = radioGroup.getCheckedRadioButtonId();
            if(radioGroup.getCheckedRadioButtonId() == R.id.radioButton) {
                r1.setChecked(true);
            }
            else if(radioGroup.getCheckedRadioButtonId() == R.id.radioButton2)
                { r2.setChecked(true);
            }
            else if(radioGroup.getCheckedRadioButtonId() == R.id.radioButton3)
                { r3.setChecked(true);
            }
            else if(radioGroup.getCheckedRadioButtonId() == R.id.radioButton4)
                { r4.setChecked(true);
            }
            else if(radioGroup.getCheckedRadioButtonId() ==R.id.radioButton5)
                { r5.setChecked(true);
            }
        }
    }
}

    class popGridAdapter extends BaseAdapter{
```

```
        Context mContext;
        popGridAdapter(Context context){
            mContext=context;
        }
        @Override
        public Object getItem(int i) {
            return null;
        }

        @Override
        public long getItemId(int i) {
            return i;
        }

        @Override
        public int getCount() {
            return questions.size();
        }

        @Override
        public View getView(final int i, View view, ViewGroup viewGroup)
            { View convertView;
            if(view==null){
                convertView=new Button(mContext);
            }else convertView=view;
            if(answers[i]==null)

(convertView).setBackgroundColor(getResources().getColor(android.R.color.darker_gray));
            else

(convertView).setBackgroundColor(getResources().getColor(android.R.color.holo_green_dark));

            ((Button)convertView).setText(""+(i+1));

            (convertView).setOnClickListener(new View.OnClickListener()
                { @Override
                public void onClick(View view) {
                    //setNextPrevButton(i);
                    scrollView.smoothScrollToPosition(i);
                }
            });
            return convertView;
        }
    }

    public void slideUp(View view){
        TranslateAnimation animate = new TranslateAnimation(
```

```
            0,              // fromXDelta
            0,              // toXDelta
            0, // fromYDelta
           -view.getHeight());              // toYDelta
      animate.setDuration(500);
      view.startAnimation(animate);
      view.setVisibility(View.GONE);
   }

   /  slide the view from its current position to below
   itself public void slideDown(View view){
      view.setVisibility(View.VISIBLE); TranslateAnimation
      animate = new TranslateAnimation(
            0,              // fromXDelta
            0,              // toXDelta
            -view.getHeight(),              // fromYDelta
            0); // toYDelta
      animate.setDuration(500);
      view.startAnimation(animate);

   }

   @Override
   protected void onDestroy() {
      submit();
      super.onDestroy();
   }
}
```

Chapter 4

RESULT

4.1 SnapShots

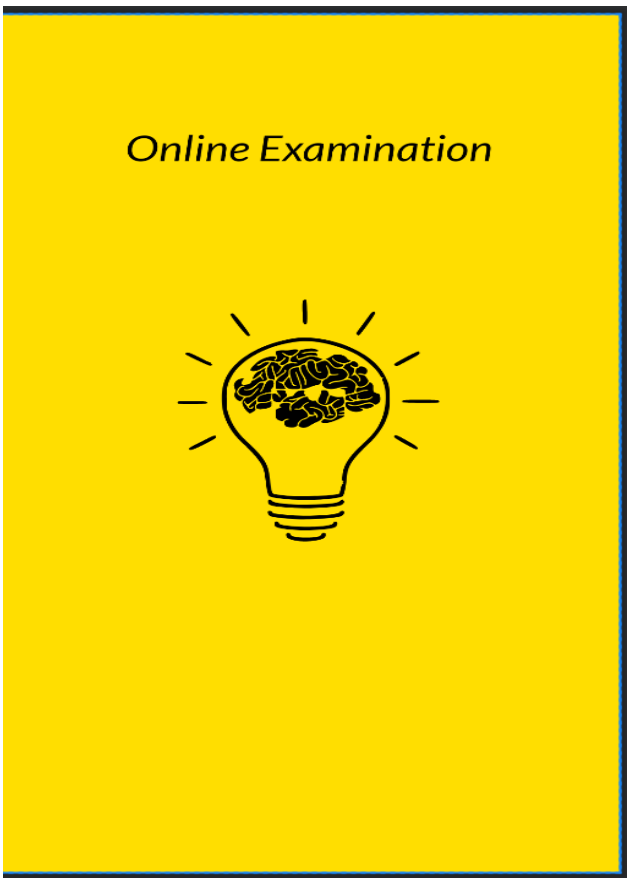


Figure 4.1 Start page of the application

This is the starting page of the application with name of the application on it.

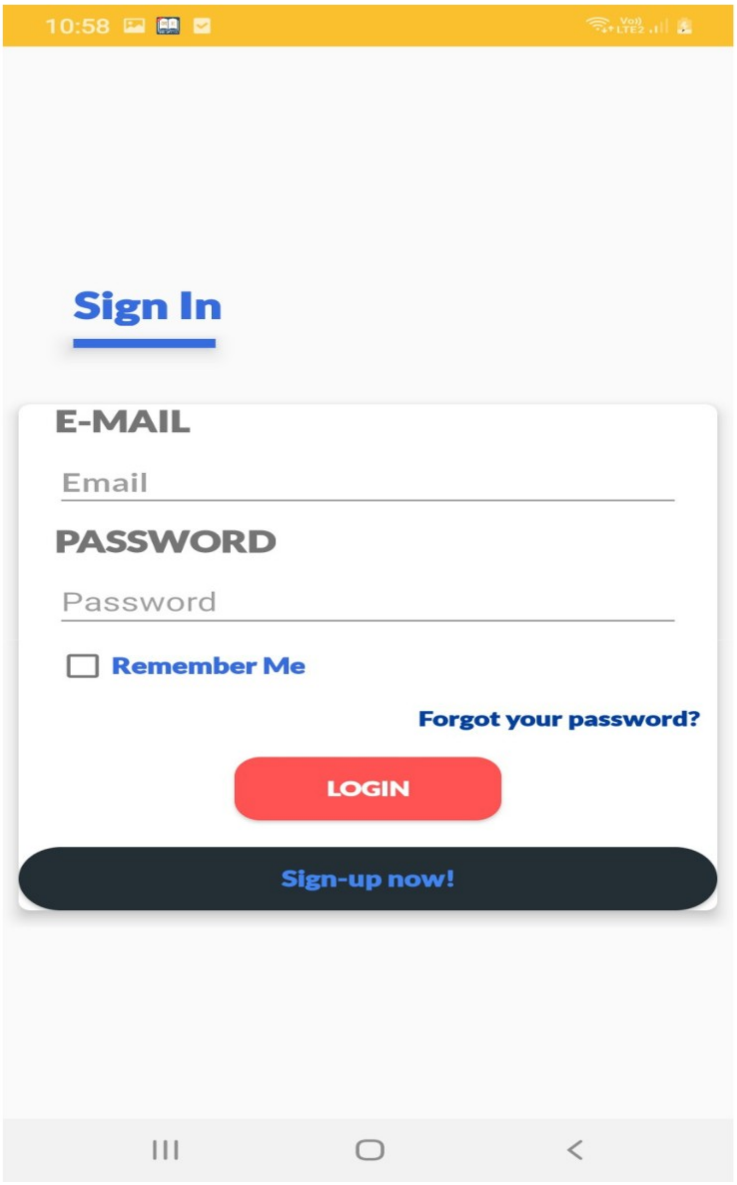


Figure 4.2 Signin interface for users

The interface provided for Sign in facility for the user by providing their registered email and registered password .If user have forgotten their password then they can set new password from "Forgot Your password" option.

Figure 4.3 Detailed report of the student

This is used for detailed description about the student such as student id, branch, section ,semester.

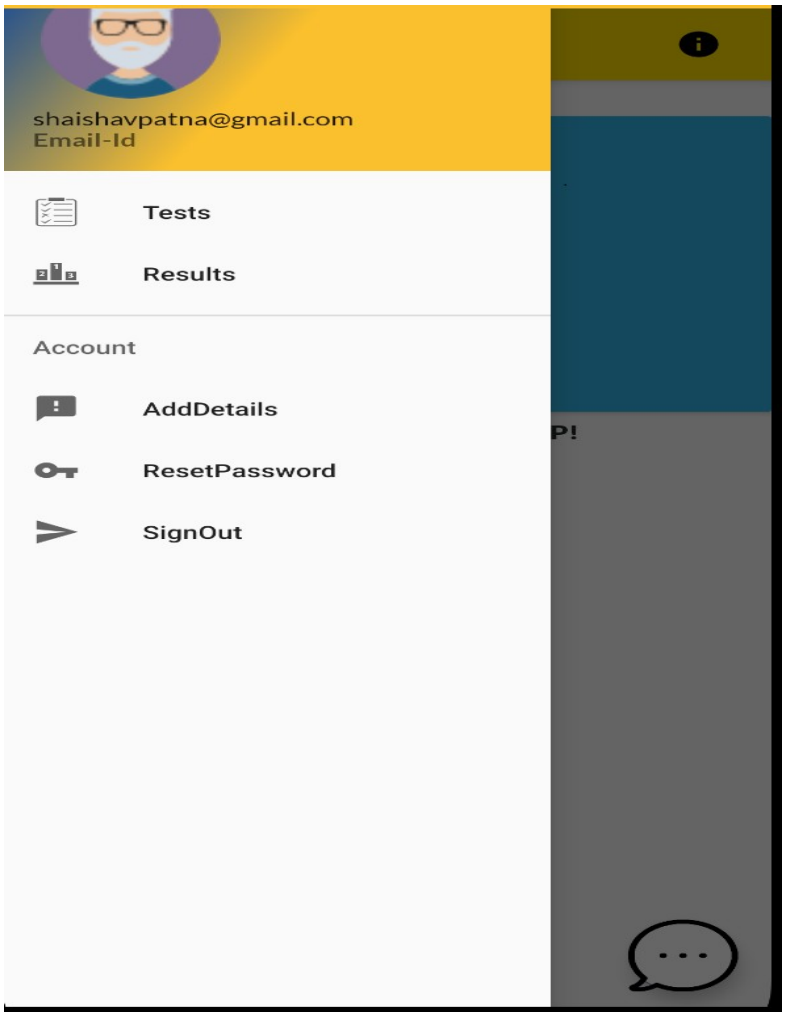


Figure 4.4 Navigation menu bar

Interface for naviagtion menu through which we can directly navigate on any implemented features of the application.

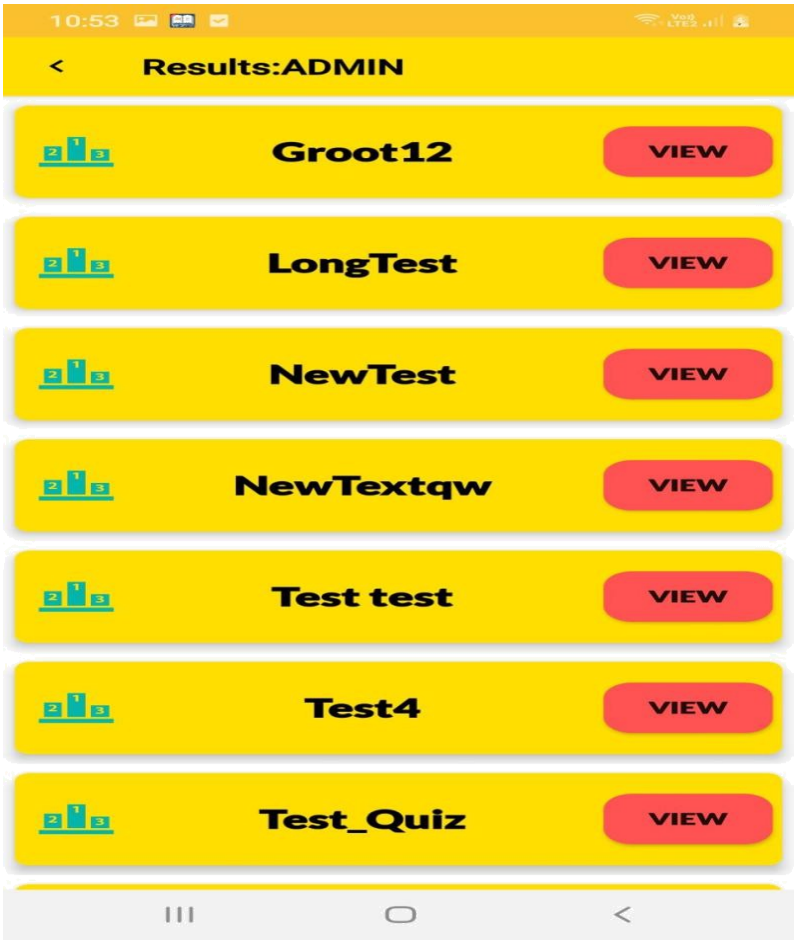


Figure 4.5 Results of the examinations

Feature implemented for providing results of the student after completion of their online examination.
It is showing a list of examinations which has been appeared by the student.

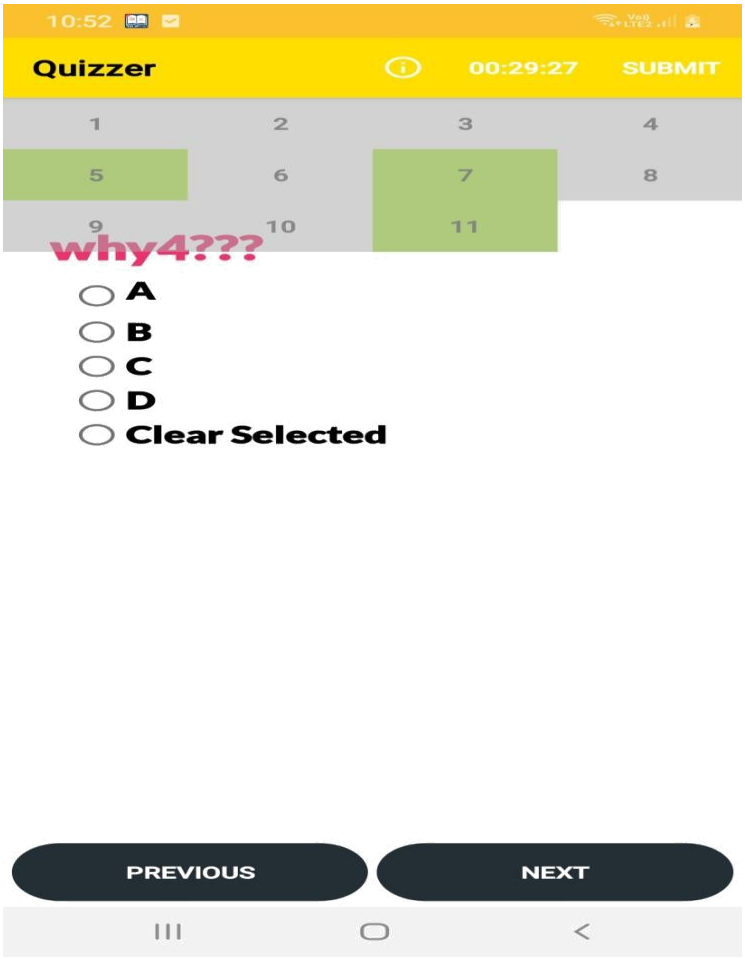


Figure 4.6 Starting of Examination

It show after starting of the examination ,the questions are assembled in number preview and the question appeared as we click on the number of question.

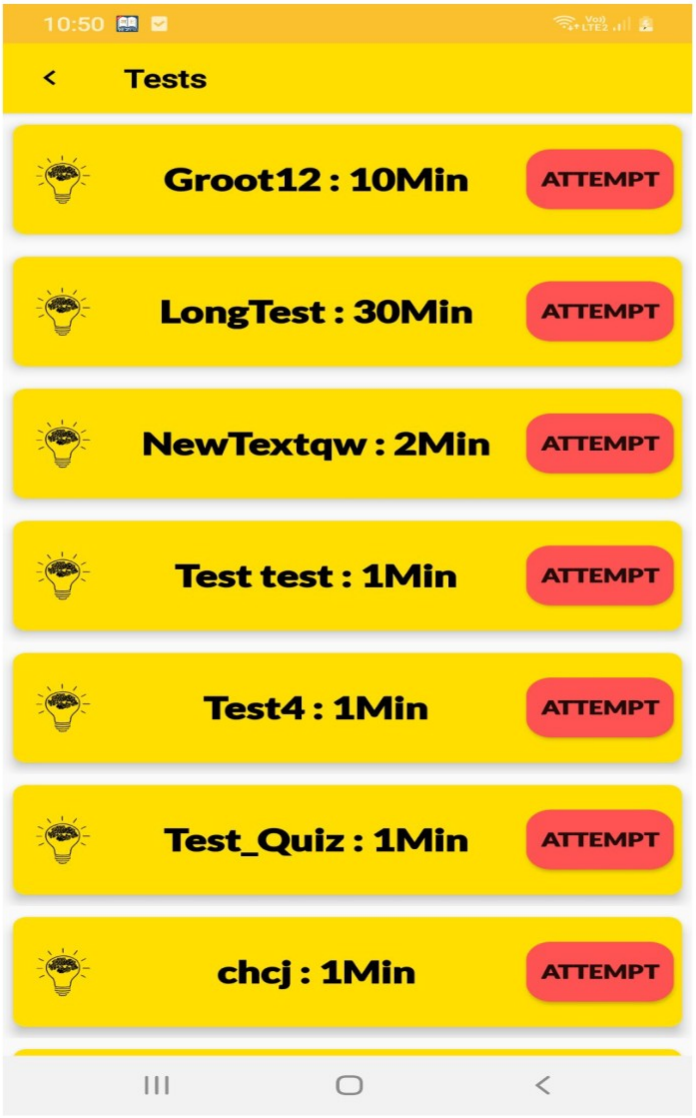


Figure 4.7 Available tests for the students

It is showing the list of available examinations for which student can attempt by clicking on the ATTEMPT button provided in extreme right side.

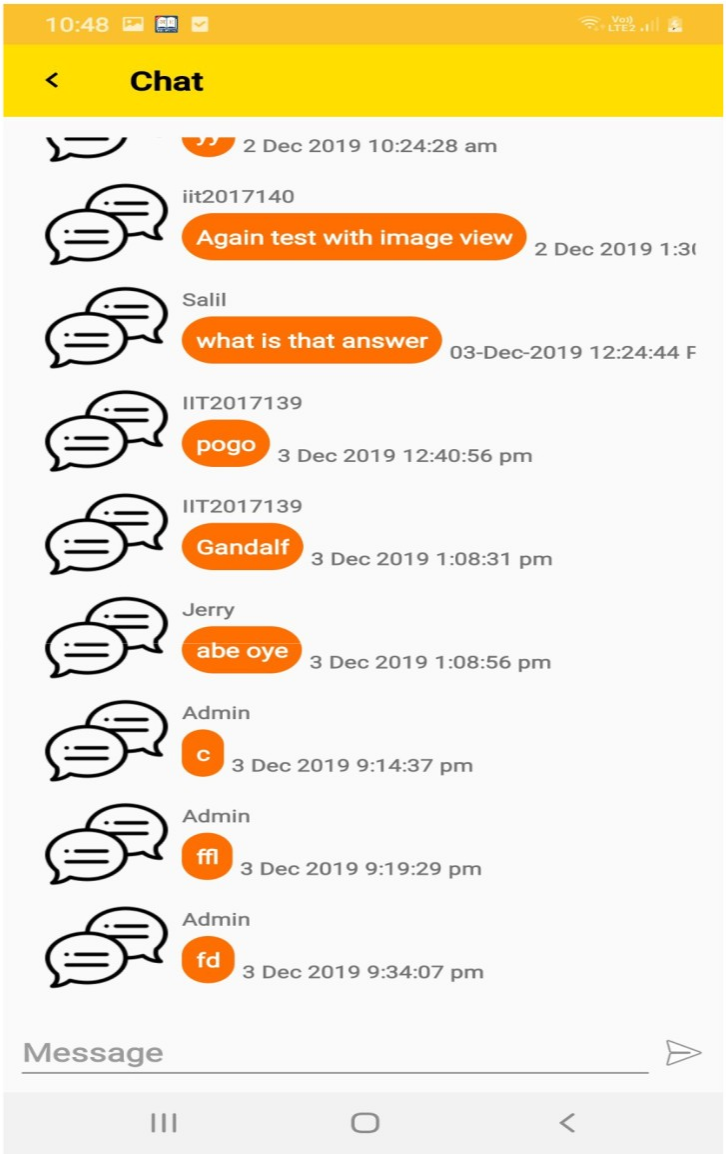


Figure 4.8 Chat feature for communication

Chat facility provided for the students, so they can do conversation with other students in order to take help and solve any problems.

Chapter 5

CONCLUSION AND FUTURE SCOPE

The development of an online examination application is a complex and challenging project, but it can be a very rewarding one. The application can provide a number of benefits for both students and instructors, and it can help to make the process of taking exams more convenient and efficient.

The future scope for mobile application development project on an online examination application is very promising. As the use of mobile devices continues to grow, more and more people will be able to access online exams. This will make it easier for schools and universities to conduct online examinations, and it will provide a more convenient and efficient way for students to take exams.

Future Scope:

The future scope for mobile application development project on an online examination application includes the following:

•Support for more question types**:** The application could be expanded to support more question types, such as drag-and-drop, matching, and essay questions.

•Integration with learning management systems**:** The application could be integrated with learning management systems (LMS) to make it easier for instructors to create and administer exams.

•Personalization: The application could be personalized to the individual student, such as by providing feedback on the student's performance and suggesting additional resources.

•Adaptive testing: The application could use adaptive testing to adjust the difficulty of the exam based on the student's performance.

REFERENCES

[1] "Agile Methodology," 4 July 2013

[2] P. Jain, "What Makes Java a Powerful Programming Language," 11 February 2013.

[3] A. Rongala, "Benefits of Java over Other Programming Languages," 7 May 2015.

[4] www.wikipedia.com

[5] www.youtube.com

[6] www.Github.com

[7] "Android Developers," ,"SoapUI,"