MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

ASANSOL ENGINEERING COLLEGE

Major Project Report

On

Coursely

SESSION: 2019-2022

ASANSOL ENGINEERING COLLEGE

Department of Computer Applications

Asansol Engineering College

Kalyanpur, Sen Raleigh Road, Asansol – 713304

CERTIFICATE

This is to certify that the project work entitled **"Coursely"** is a bonafide record of work carried out in the **Department of Computer Application**, Asansol Engineering College, Asansol.

Name	Roll Number	Registration Number
Md. Azmat Ansari	32201219040	036875
Jalmejay Kumar Shaw	32201219037	036885
Md. Fardeen Moiz	32201219045	036131
Md. Samir	32201219042	036412
Nikhil Singh	32201219033	037012

The students of 6th Semester BCA 2021-22 under my / our supervision in requirement of partial fulfillment of the Award of Degree of BCA from Maulana Abul Kalam Azad University of Technology, West Bengal.

Signature of
The Project Guide
Name: Mr. Amitava chakraborty
Designation: Assistant Professor

Recommendation & Signature of
The Principal
Name: Dr. P.P. Bhattacharya

Recommendation &Signature of
The Head of Department
Name: Dr. D. K. PAL

Recommendation &Signature of
Internal / External Examiners

ACKNOWLEDGEMENT

We would like to express my gratitude towards **Mr. Amitava Chakraborty**for guiding me and my Team Members throughout the Project as well as our HOD Sir **Dr. Dibyendu Kumar Pal,** who gave us the golden opportunity to do this wonderful project.

I feel thankful and express my kind gratitude towards my Team Members : Md. Azmat Ansari, Md. Fardeen Moiz, Md. Samir, Jalmejay Kumar Shaw and Nikhil Singh for helping and supporting me , As this is a complete team work so without their cooperation it was not possible to complete our Major Project titled " **Coursely ".**

Our Project is done under the supervision of our Guide **Mr. Amitava Chakraborty.** I thank all participants for their positive support and guidance.

Md. Azmat Ansari
Nikhil Singh
Md. Samir
Md.Fardeen Moiz
Jalmejay Kumar Shaw

PROJECT CONTENTS

INTRODUCTION

The project's name is "**Coursely**." Coursely is an android application that has general questions related to current affairs and computer. It has multiple choice questions with time limit and it also calculate scores of each correct answer. It is good for students of every age group it helps in increasing general knowledge about world Sports and computer etc. Don't need register simply give any user name and password it will saved automatically and you can login again with same user name and password don't have to worry about the past score. The application helps the user to increase his/her knowledge. Since Smartphone mobiles are being widely used by general population and students, the Coursely application can provide on the Student's mobile. Coursely is a application developed to conduct an quiz based on time constraints. Coursely system is accessed by entering the user name and password which is added to the database. Before start of the quiz, the rules and regulations are displayed that includes description of the time limit, number of questions to be answered and scoring methods. Quiz is started by displaying one question with four options each based on computer and general knowledge.

OBJECTIVE

The main objective of "Coursely" is to facilitate a user friendly environment for all users and reduces the manual effort. In past days quiz is conducted manually but in further resolution of the technology we are able to generate the score and pose the queries automatically. The functional requirements include to create users that are going to participate in the quiz, automatic score and report generation and administrative tasks like add, delete, update for admin privilege users. In this application, all the permissions lies with the administrator i.e. specifying the details of the quiz with checking result will show to interviewee or not, addition of question and answers, marks for each question, Set timer for each quiz and generate report with score for each quiz.

SYSTEM REQUIREMENT SPECIFICATION

Hardware Requirement

Processor	: Intel i5 or higher
Main Memory (RAM)	: 4 GB
Cache Memory	:512 KB
Hard disk	: 100GB

SoftwareRequirement

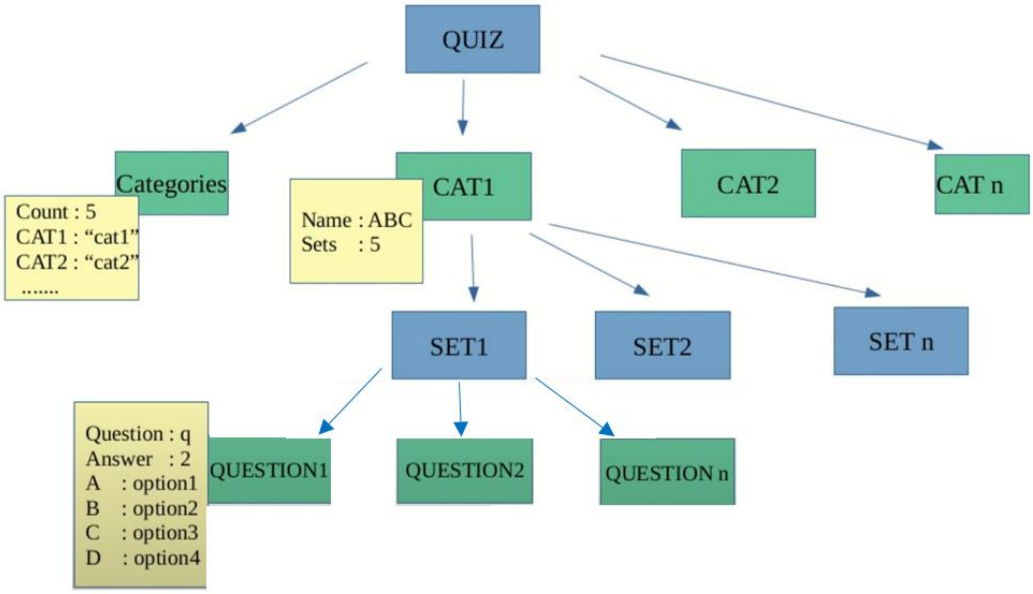
- Windows 10 or higher
- Android Studio
- Android Virtual Device (AVD)

Frontend & Backend development

- XML
- JAVA
- Firebase

E-R DIAGRAM

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects.



FIREBASE

Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and Web apps. Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiment.

Firebase offers a number of services, including:

- Authentication – Firebase Authentication makes it easy for developers to build secure authentication systems and enhances the sign-in and onboarding experience for users. This feature offers a complete identity solution, supporting email and password accounts, phone auth, as well as Google, Facebook, GitHub, Twitter login and more.
- Realtime database – the Firebase Realtime Database is a cloud-hosted NoSQL database that enables data to be stored and synced between users in real time. The data is synced across all clients in real time and is still available when an app goes offline.
- Crashlytics – Firebase Crashlytics is a real-time crash reporter that helps developers track, prioritize and fix stability issues that reduce the quality of their apps. With crashlytics, developers spend less time organizing and troubleshooting crashes and more time building features for their apps.
- Performance – Firebase Performance Monitoring service gives developers insight into the performance characteristics of their iOS and Android apps to help them determine where and when the performance of their apps can be improved.
- Test lab – Firebase Test Lab is a cloud-based app-testing infrastructure. With one operation, developers can test their iOS or Android apps across a variety of devices and device configurations. They can see the results, including videos, screenshots and logs, in the Firebase console.

CODE SHEET AND SCREENSHOTS

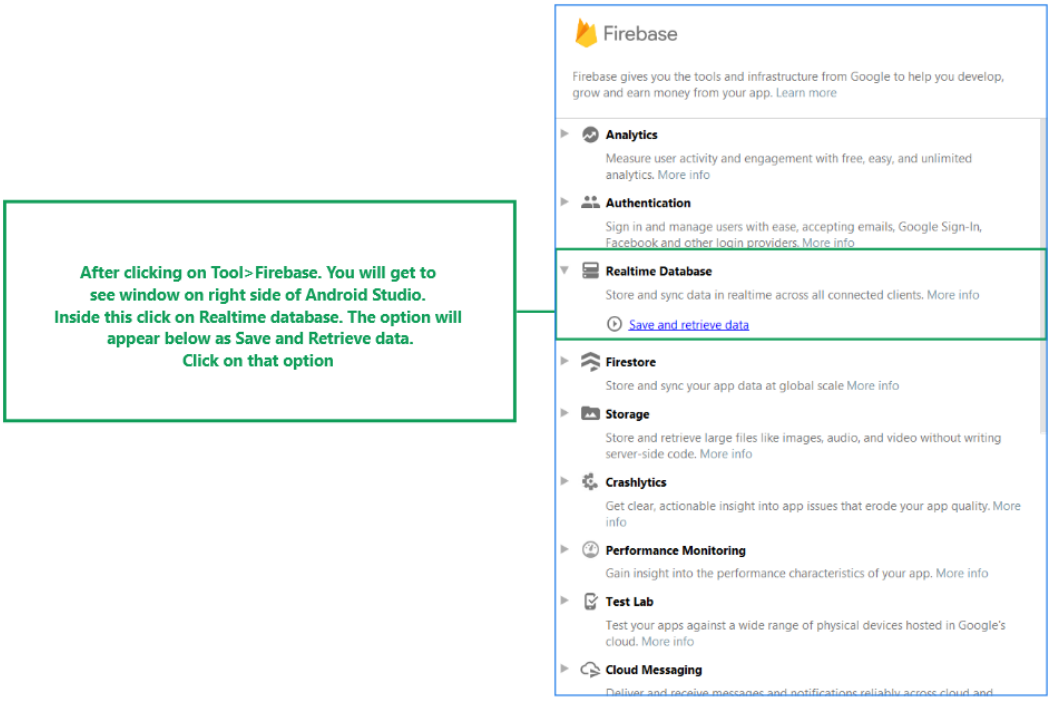
Code CourseLy App:

Step 1: Create a new Project

To create a new project in Android Studio, The next step is to choose the Activity to mobile. Activity in Android refers to a single screen with a user interface. For Beginners, “**Empty Activity**” is recommended .Note that select **Java** as the programming language.

Step 2: Connect your app to Firebase

After creating a new project navigate to the Tools option on the top bar. Inside that click on Firebase. After clicking on Firebase, you can get to see the right column mentioned below in the screenshot.



After clicking on Tool>Firebase. You will get to see window on right side of Android Studio. Inside this click on Realtime database. The option will appear below as Save and Retrieve data. Click on that option

Firebase

Firebase gives you the tools and infrastructure from Google to help you develop, grow and earn money from your app. Learn more

Analytics
Measure user activity and engagement with free, easy, and unlimited analytics. More info

Authentication
Sign in and manage users with ease, accepting emails, Google Sign-In, Facebook and other login providers. More info

Realtime Database
Store and sync data in realtime across all connected clients. More info
Save and retrieve data

Firestore
Store and sync your app data at global scale More info

Storage
Store and retrieve large files like images, audio, and video without writing server-side code. More info

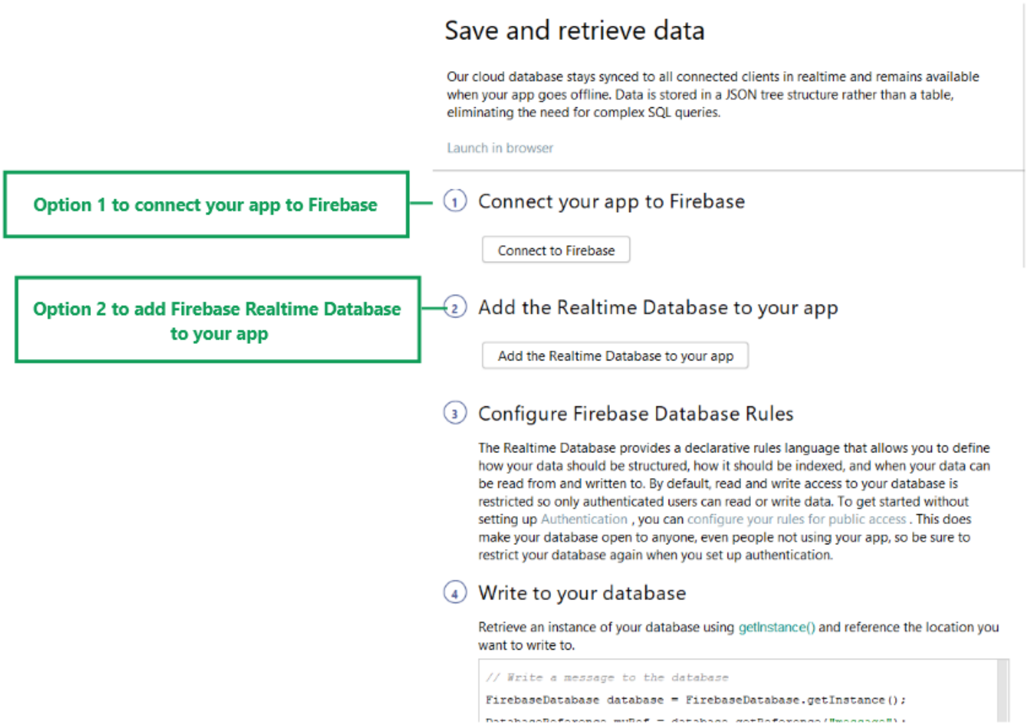
Crashlytics
Get clear, actionable insight into app issues that erode your app quality. More info

Performance Monitoring
Gain insight into the performance characteristics of your app. More info

Test Lab
Test your apps against a wide range of physical devices hosted in Google's cloud. More info

Cloud Messaging
Deliver and receive messages and notifications reliably across cloud and

Inside that column Navigate to Firebase Realtime Database. Click on that option and you will get to see two options on Connect app to Firebase and Add Firebase Realtime Database to your app. Click on Connect now and your app will be connected to Firebase. After that click on the second option and now your App is connected to Firebase.



Save and retrieve data

Our cloud database stays synced to all connected clients in realtime and remains available when your app goes offline. Data is stored in a JSON tree structure rather than a table, eliminating the need for complex SQL queries.

Launch in browser

1. Connect your app to Firebase

   Connect to Firebase

2. Add the Realtime Database to your app

   Add the Realtime Database to your app

3. Configure Firebase Database Rules

   The Realtime Database provides a declarative rules language that allows you to define how your data should be structured, how it should be indexed, and when your data can be read from and written to. By default, read and write access to your database is restricted so only authenticated users can read or write data. To get started without setting up Authentication , you can configure your rules for public access . This does make your database open to anyone, even people not using your app, so be sure to restrict your database again when you set up authentication.

4. Write to your database

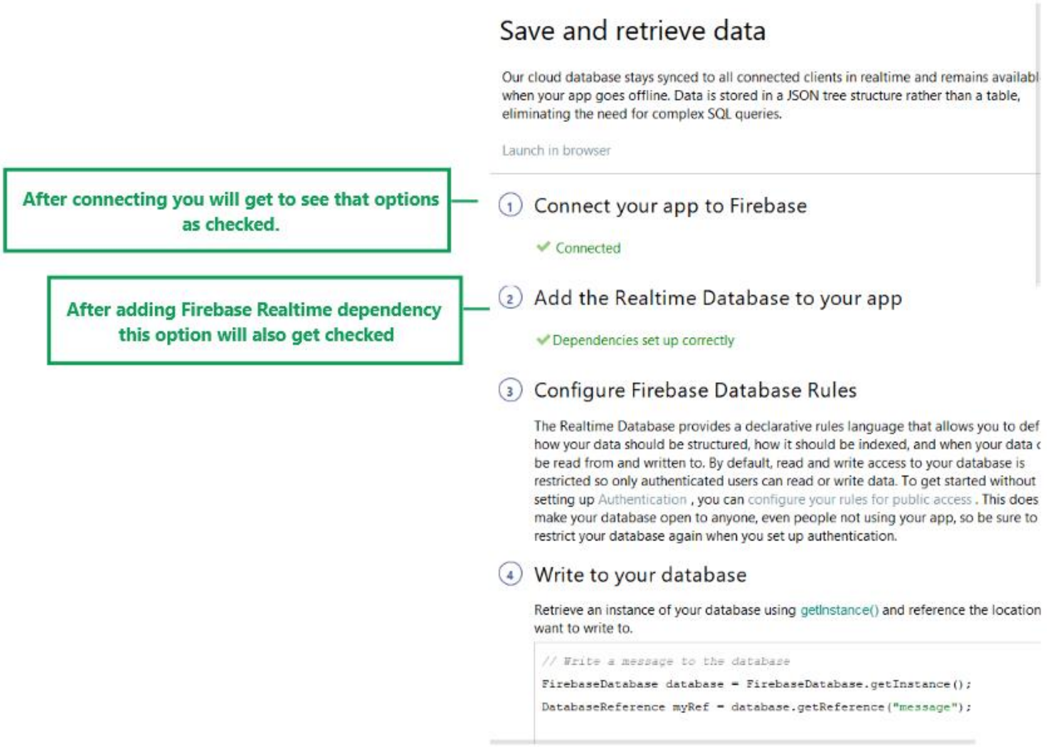
   Retrieve an instance of your database using getInstance() and reference the location you want to write to.

```
// Write a message to the database
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("message");
```

Option 1 to connect your app to Firebase

Option 2 to add Firebase Realtime Database to your app

After connecting your app to Firebase you will get to see the below screen.



Save and retrieve data

Our cloud database stays synced to all connected clients in realtime and remains availabl when your app goes offline. Data is stored in a JSON tree structure rather than a table, eliminating the need for complex SQL queries.

Launch in browser

1. Connect your app to Firebase

   ✓ Connected

2. Add the Realtime Database to your app

   ✓ Dependencies set up correctly

3. Configure Firebase Database Rules

   The Realtime Database provides a declarative rules language that allows you to def how your data should be structured, how it should be indexed, and when your data c be read from and written to. By default, read and write access to your database is restricted so only authenticated users can read or write data. To get started without setting up Authentication , you can configure your rules for public access . This does make your database open to anyone, even people not using your app, so be sure to restrict your database again when you set up authentication.

4. Write to your database

   Retrieve an instance of your database using getInstance() and reference the location want to write to.

```
// Write a message to the database
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("message");
```

After connecting you will get to see that options as checked.

After adding Firebase Realtime dependency this option will also get checked

```
Dependencies {
implementation 'androidx.appcompat:appcompat:1.4.1'
    implementation 'com.google.android.material:material:1.6.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'com.google.firebase:firebase-firestore:24.1.2'
    implementation 'com.google.firebase:firebase-auth:21.0.5'
    implementation 'com.google.firebase:firebase-database:20.0.5'
    implementation platform('com.google.firebase:firebase-bom:28.2.1')
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
}
```

Step 3: Creating different empty activities

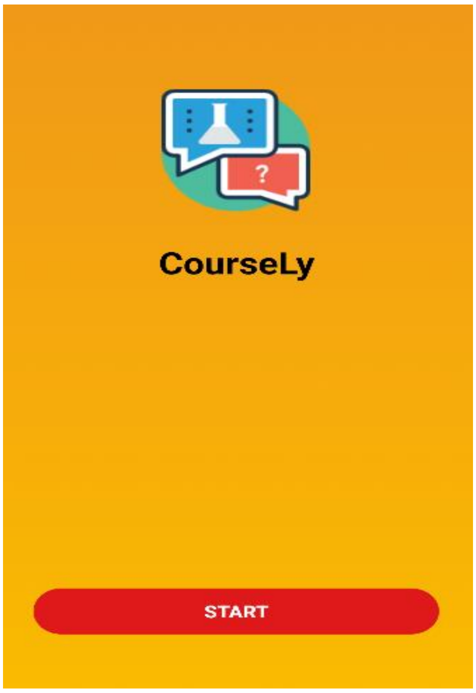
MainActivity.java

```
package com.example.myquiz;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.res.ResourcesCompat;
import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private TextView title;

    private Button start;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        title = findViewById(R.id.main_title);

        start = findViewById(R.id.ma_startB);

        Typeface typeface = ResourcesCompat.getFont(this,R.font.blacklist);

   title.setTypeface(typeface);

        start.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                Intent intent = new Intent(MainActivity.this,CategoryActivity.class);

         startActivity(intent);

            }

        });

    }

}
```

activity_main.xml

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

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:background="@drawable/main_background"

    tools:context=".MainActivity">

<ImageView
```

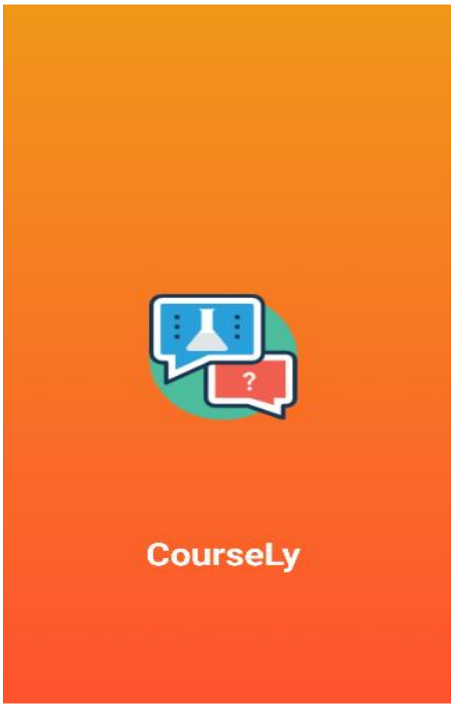
```
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.165"
        app:srcCompat="@drawable/logo" />
    <TextView
        android:id="@+id/main_title"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:text="CourseLy"
        android:gravity="center"
        android:textSize="32sp"
        android:textStyle="bold"
        android:textColor="@color/black"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView2" />
    <Button
        android:id="@+id/ma_startB"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="64dp"
        android:text="Start"
        android:background="@drawable/round_corner"
```



```
    android:textSize="18sp"
    android:textStyle="bold"
    android:textColor="@color/colorAccent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_splash.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="@drawable/gradient_background"
  tools:context=".SplashActivity">
<ImageView
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/logo" />
<TextView
    android:id="@+id/appName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="32dp"
```



```
        android:text="CourseLy"
        android:textColor="@color/white"
        android:textSize="32sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

CategoryActivity.java

```
package com.example.myquiz;
import static com.example.myquiz.SplashActivity.catList;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import android.os.Bundle;
import android.view.MenuItem;
import android.widget.GridView;
public class CategoryActivity extends AppCompatActivity {
    private GridView catGrid;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_category);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setTitle("Categories");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        catGrid = findViewById(R.id.catGridview);
        CatGridAdapter adapter =new CatGridAdapter(catList);
        catGrid.setAdapter(adapter);
    }
    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {

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

activity_category.xml

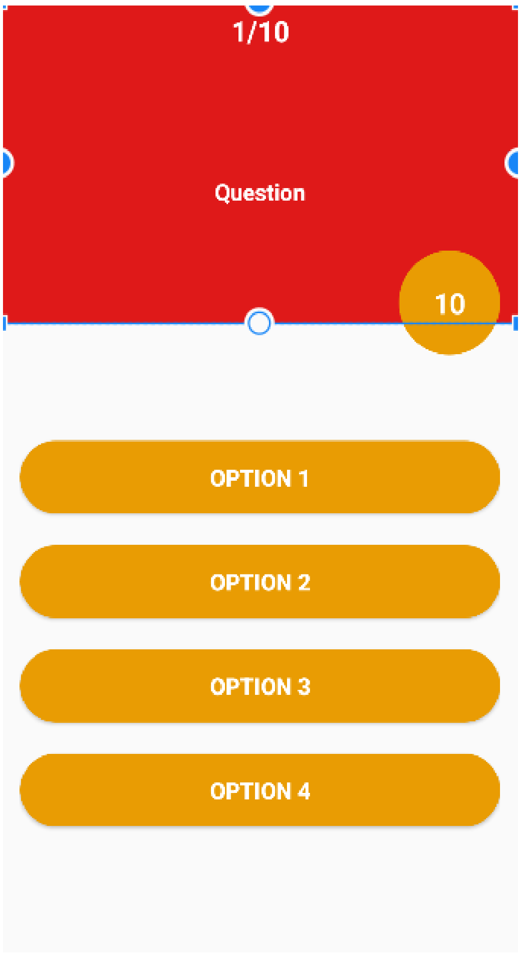
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".CategoryActivity">
<androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/colorPrimary"
        android:theme="@style/ThemeOverlay.AppCompat.Dark">
</androidx.appcompat.widget.Toolbar>
<GridView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:id="@+id/catGridview"
        android:background="@drawable/border"
        android:gravity="center"
        android:columnWidth="100dp"
        android:numColumns="2">
</GridView>
</LinearLayout>
```

activity_question.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".QuestionActivity">
<LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:orientation="vertical"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">
<TextView
            android:id="@+id/quest_num"
            android:layout_width="match_parent"
```

```
      android:layout_height="wrap_content"
      android:gravity="center"
      android:padding="8dp"
      android:text="1/10"
      android:textColor="@color/colorAccent"
      android:textSize="22sp"
      android:textStyle="bold" />
<TextView
      android:id="@+id/question"
      android:layout_width="match_parent"
      android:layout_height="170dp"
      android:layout_margin="16dp"
      android:gravity="center"
      android:text="Question"
      android:textColor="@color/colorAccent"
      android:textSize="18sp"
      android:textStyle="bold" />
</LinearLayout>
<LinearLayout
    android:id="@+id/linearLayout2"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    android:gravity="center"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout">
```

```
<Button
        android:id="@+id/option1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:background="@drawable/round_corner"
        android:backgroundTint="#E99C03"
        android:elevation="3dp"
        android:padding="16dp"
        android:text="option 1"
        android:textColor="@color/colorAccent"
        android:textSize="18sp"
        android:textStyle="bold" />
<Button
        android:id="@+id/option2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:background="@drawable/round_corner"
        android:backgroundTint="#E99C03"
        android:elevation="3dp"
        android:padding="16dp"
        android:text="option 2"
        android:textColor="@color/colorAccent"
        android:textSize="18sp"
        android:textStyle="bold" />
<Button
        android:id="@+id/option3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:background="@drawable/round_corner"
        android:backgroundTint="#E99C03"
        android:elevation="3dp"
        android:padding="16dp"
        android:text="option 3"
        android:textColor="@color/colorAccent"
        android:textSize="18sp"
        android:textStyle="bold" />

<Button
        android:id="@+id/option4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:background="@drawable/round_corner"
        android:backgroundTint="#E99C03"
        android:elevation="3dp"
        android:padding="16dp"
        android:text="option 4"
        android:textColor="@color/colorAccent"
        android:textSize="18sp"
        android:textStyle="bold" />
</LinearLayout>
<TextView
    android:id="@+id/countdown"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="8dp"
    android:text="10"
    android:textColor="@color/colorAccent"
    android:textSize="22sp"
    android:textStyle="bold"
```



```
        android:gravity="center"
        android:background="@drawable/round_corner"
        android:backgroundTint="#E99C03"
        app:layout_constraintBottom_toTopOf="@+id/linearLayout2"
        app:layout_constraintEnd_toEndOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Question.java

```
package com.example.myquiz;

public class Question {
    String question;
    String optionA;
    String optionB;
    String optionC;
    String optionD;
    int correctAns;
    public Question(String question, String optionA, String optionB, String optionC, String
optionD, int correctAns) {
        this.question = question;
        this.optionA = optionA;
        this.optionB = optionB;
        this.optionC = optionC;
        this.optionD = optionD;
        this.correctAns = correctAns;
    }
    public String getQuestion() {
        return question;   }
    public void setQuestion(String question) {
        this.question = question;}
    public String getOptionA() {
        return optionA;  }
    public void setOptionA(String optionA) {
        this.optionA = optionA;  }
    public String getOptionB() {
        return optionB;  }
    public void setOptionB(String optionB) {
        this.optionB = optionB;}
    public String getOptionC() {
        return optionC;  }
    public void setOptionC(String optionC) {
        this.optionC = optionC;  }
    public String getOptionD() {
        return optionD;}
    public void setOptionD(String optionD) {
        this.optionD = optionD;  }
    public int getCorrectAns() {
        return correctAns;  }
    public void setCorrectAns(int correctAns) {
        this.correctAns = correctAns;
    }
}
```

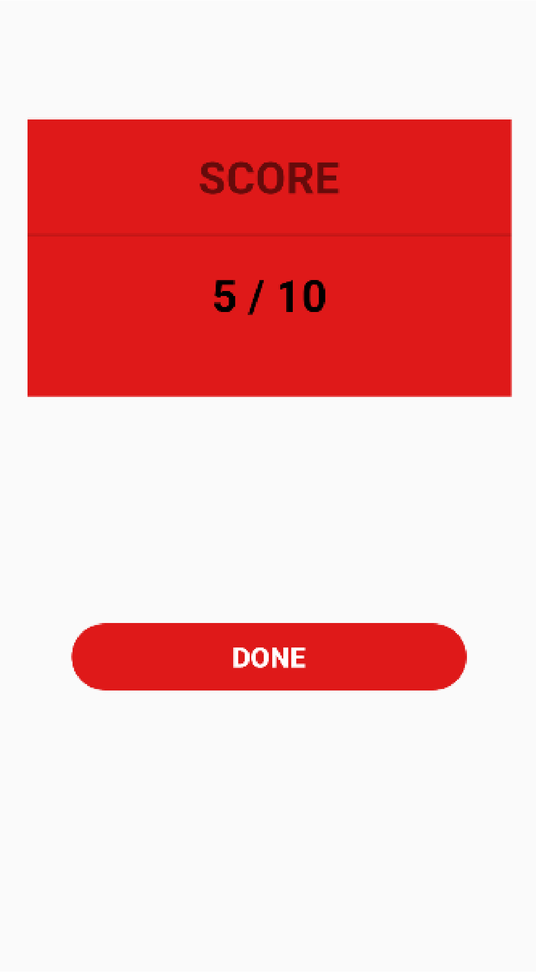
ScoreActivity.java

```
package com.example.myquiz;


import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class ScoreActivity extends AppCompatActivity {
   private TextView score;
   private Button done;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_score);
      score = findViewById(R.id.sa_score);
      done = findViewById(R.id.sa_done);
      String score_str = getIntent().getStringExtra("SCORE");
      score.setText(score_str);
      done.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            Intent intent = new Intent(ScoreActivity.this,MainActivity.class);
            ScoreActivity.this.startActivity(intent);
            ScoreActivity.this.finish();
         }
      });
   } }
```

activity_score.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
    tools:context=".ScoreActivity">
<LinearLayout
        android:id="@+id/linearLayout3"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="100dp"
        android:layout_marginEnd="32dp"
        android:background="@color/colorPrimary"
        android:orientation="vertical"
        android:paddingBottom="32dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">
<TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="20dp"
            android:gravity="center"
            android:text="SCORE"
            android:textSize="32sp"
            android:textStyle="bold" />
<View
            android:id="@+id/divider"
            android:layout_width="match_parent"
            android:layout_height="2dp"
            android:background="?android:attr/listDivider" />
<TextView
            android:id="@+id/sa_score"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="5 / 10"
            android:layout_margin="20dp"
            android:textColor="@android:color/black"
            android:textSize="32sp"
            android:textStyle="bold" />
</LinearLayout>

<Button
        android:id="@+id/sa_done"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="64dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="64dp"
        android:layout_marginBottom="64dp"
        android:background="@drawable/round_corner"
        android:text="Done"
        android:textColor="@color/colorAccent"
        android:textStyle="bold"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/linearLayout3" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



activity_sets.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".SetsActivity">
<androidx.appcompat.widget.Toolbar
        android:id="@+id/set_toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/colorPrimary">
</androidx.appcompat.widget.Toolbar>
<TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Sets"
        android:textSize="26sp"
        android:textStyle="bold"
        android:padding="16dp"/>
<GridView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:id="@+id/sets_gridview"
        android:layout_weight="1"
        android:gravity="center"
        android:horizontalSpacing="16dp"
        android:verticalSpacing="16dp"
        android:padding="16dp"
        android:columnWidth="100dp"
 android:numColumns="auto_fit">
</GridView>
</LinearLayout>
```

SetsAdapter.java

```
package com.example.myquiz;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;
public class SetsAdapter extends BaseAdapter {
    private int numOfSets;
    public SetsAdapter(int numOfSets) {
        this.numOfSets = numOfSets;
    }
    @Override
    public int getCount() {
        return numOfSets;
    }
    @Override
    public Object getItem(int position) {
        return null;
    }
    @Override
    public long getItemId(int position) {
        return 0;
    }
    @Override
    public View getView(final int position, View convertView, final ViewGroup parent) {
        View view;
        if (convertView == null){
```

```
      view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.set_item_layout,parent,false);
    }
    else
    {
      view = convertView;
    }
    view.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        Intent intent = new Intent(parent.getContext(),QuestionActivity.class);
        intent.putExtra("SETNO",position);
        parent.getContext().startActivity(intent);
      }
    });
    ((TextView) view.findViewById(R.id.setNo_tv)).setText(String.valueOf(position+1));
    return view;
  } }
```

loading_progressbar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  xmlns:app="http://schemas.android.com/apk/res-auto">
<ProgressBar
    android:id="@+id/progressBar"
    style="?android:attr/progressBarStyle"
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:layout_marginStart="16dp"
```

```
        android:layout_marginTop="16dp"
        android:layout_marginBottom="16dp"
        android:indeterminateTint="@color/colorPrimary"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
<TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:text="Loading..."
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/progressBar"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

AndroidManifest.xml

```
<?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.myquiz">
<!-- permissions for internet -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyQuiz"
        tools:targetApi="31">
<activity
      android:name=".ScoreActivity"
      android:exported="false" />
<activity
      android:name=".QuestionActivity"
      android:exported="false" />
<activity
      android:name=".SetsActivity"
      android:exported="false" />
<activity
      android:name=".CategoryActivity"
      android:exported="false" />
<activity
      android:name=".SplashActivity"
      android:exported="true">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<activity
      android:name=".MainActivity"
      android:exported="false">
</activity>
</application>
</manifest>
```

Code Admin App:

```
dependencies {
    implementation 'androidx.appcompat:appcompat:1.4.1'
    implementation 'com.google.android.material:material:1.6.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'com.google.firebase:firebase-auth:21.0.5'
    implementation 'com.google.firebase:firebase-firestore:24.1.2'
    implementation 'com.google.firebase:firebase-database:20.0.5'
    implementation platform('com.google.firebase:firebase-bom:28.2.1')
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
}
```

MainActivity.java

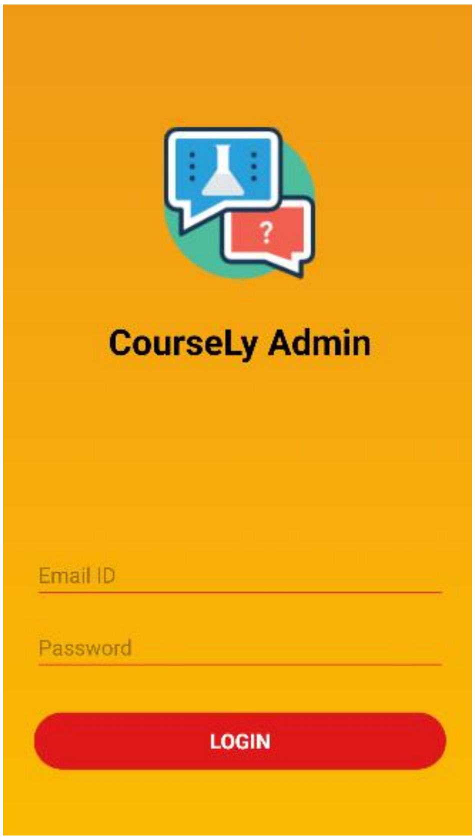
```
package com.example.admin;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.app.Dialog;
import android.content.Intent;
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;

public class MainActivity extends AppCompatActivity {
    private EditText email, pass;
    private Button login;
    private FirebaseAuth firebaseAuth;
    private Dialog loadingDialog;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        email = findViewById(R.id.email);
        pass = findViewById(R.id.password);
        login = findViewById(R.id.loginB);
        loadingDialog = new Dialog(MainActivity.this);
        loadingDialog.setContentView(R.layout.loading_progressbar);
        loadingDialog.setCancelable(false);
loadingDialog.getWindow().setBackgroundDrawableResource(R.drawable.progress_backgro
und);
loadingDialog.getWindow().setLayout(ViewGroup.LayoutParams.WRAP_CONTENT,ViewGro
up.LayoutParams.WRAP_CONTENT);
        firebaseAuth = FirebaseAuth.getInstance();
        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(email.getText().toString().isEmpty()) {
                    email.setError("Enter Email ID");
                    return;
                } else    {
                    email.setError(null);
                }
                if (pass.getText().toString().isEmpty()) {
                    pass.setError("Enter Password");
                    return;
                } else  {
                    pass.setError(null);
                }
                firebaseLogin();
```

```
      }
    });
    if (firebaseAuth.getCurrentUser() !=null)
    {
      Intent intent = new Intent(MainActivity.this,CategoryActivity.class);
      startActivity(intent);
      finish();
    }
  }
  private void firebaseLogin()
  {
    loadingDialog.show();
    firebaseAuth.signInWithEmailAndPassword(email.getText().toString(),
pass.getText().toString())
        .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
          @Override
          public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()){
              // Sign in success, update UI with the signed-in user's information
              Toast.makeText(MainActivity.this, "Success", Toast.LENGTH_SHORT).show();
              Intent intent = new Intent(MainActivity.this,CategoryActivity.class);
              startActivity(intent);
           finish();
            }else {
              // If sign in fails, display a message to the user.
              Toast.makeText(MainActivity.this, "Failure", Toast.LENGTH_SHORT).show();
            }
            loadingDialog.dismiss();
          }
        });
  } }
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="@drawable/main_background"
  tools:context=".MainActivity">
<ImageView
    android:id="@+id/imageView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.216"
    app:srcCompat="@drawable/logo" />

<TextView
    android:id="@+id/main_title"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:gravity="center"
    android:text="CourseLy Admin"
    android:textColor="@android:color/black"
```

```
        android:textSize="30sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView2" />
    <Button
        android:id="@+id/loginB"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="64dp"
        android:background="@drawable/round_corner"
        android:text="Login"
        android:textColor="@color/white"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <EditText
        android:id="@+id/email"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="16dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="Email ID"
        android:backgroundTint="@color/colorPrimary"
        android:textCursorDrawable="@null"
        app:layout_constraintBottom_toTopOf="@+id/password"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <EditText
        android:id="@+id/password"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="32dp"
        android:ems="10"
        android:hint="Password"
        android:textCursorDrawable="@null"
        android:backgroundTint="@color/colorPrimary"
        android:inputType="textPassword"
        app:layout_constraintBottom_toTopOf="@+id/loginB"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>
```



activity_category.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="#F0D6D6"
  tools:context=".CategoryActivity">
<androidx.appcompat.widget.Toolbar
```

```
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/colorPrimary"
        android:theme="@style/ThemeOverlay.AppCompat.Dark"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">
    </androidx.appcompat.widget.Toolbar>
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/cat_recycler"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        app:layout_constraintBottom_toTopOf="@+id/addCatB"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/toolbar">
    </androidx.recyclerview.widget.RecyclerView>
    <Button
        android:id="@+id/addCatB"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:padding="16dp"
        android:text="Add New Category"
        android:textColor="@color/white"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

CategoryModel.java

```
package com.example.admin;
public class CategoryModel {
    private String id;
    private String name;
    private String noOfSets;
    private String setCounter;
    public CategoryModel(String id, String name, String noOfSets, String setCounter) {
        this.id = id;
        this.name = name;
        this.noOfSets = noOfSets;
        this.setCounter = setCounter;   }
  public String getSetCounter() {
     return setCounter;   }
    public void setSetCounter(String setCounter) {
        this.setCounter = setCounter;    }
    public String getId() {
        return id;    }
    public void setId(String id) {
        this.id = id;   }
    public String getName() {
        return name;    }
    public void setName(String name) {
        this.name = name;    }
    public String getNoOfSets() {
        return noOfSets;    }
    public void setNoOfSets(String noOfSets) {
        this.noOfSets = noOfSets; }
}
```

QuestionsActivity.java

```
package com.example.admin;
import static com.example.admin.CategoryActivity.catList;
import static com.example.admin.CategoryActivity.selected_cat_index;
import static com.example.admin.SetsActivity.selected_set_index;
import static com.example.admin.SetsActivity.setsIDs;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.app.Dialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Toast;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QueryDocumentSnapshot;
import com.google.firebase.firestore.QuerySnapshot;

public class QuestionsActivity extends AppCompatActivity {
    private RecyclerView quesView;
    private Button addQB;
    public static List<QuestionModel> quesList = new ArrayList<>();
    private QuestionAdapter adapter;
    private FirebaseFirestore firestore;
    private FirebaseAuth firebaseAuth;
    private Dialog loadingDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_questions);
        Toolbar toolbar = findViewById(R.id.q_toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setTitle("Questions");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        quesView = findViewById(R.id.quest_recycler);
        addQB = findViewById(R.id.addQB);
        firebaseAuth = FirebaseAuth.getInstance();
        loadingDialog = new Dialog(QuestionsActivity.this);
        loadingDialog.setContentView(R.layout.loading_progressbar);
        loadingDialog.setCancelable(false);
loadingDialog.getWindow().setBackgroundDrawableResource(R.drawable.progress_background);
loadingDialog.getWindow().setLayout(ViewGroup.LayoutParams.WRAP_CONTENT,ViewGroup.LayoutParams.WRAP_CONTENT);
        addQB.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(QuestionsActivity.this, QuestionDetailsActivity.class);
                intent.putExtra("ACTION","ADD");
                startActivity(intent);
            }
        });
```

```
  LinearLayoutManager layoutManager = new LinearLayoutManager(this);
  layoutManager.setOrientation(LinearLayoutManager.VERTICAL);
  quesView.setLayoutManager(layoutManager);

  firestore = FirebaseFirestore.getInstance();
  loadQuestions();
}
private void loadQuestions()
{
  quesList.clear();
  loadingDialog.show();
  firestore.collection("QUIZ").document(catList.get(selected_cat_index).getId())
      .collection(setsIDs.get(selected_set_index)).get()
      .addOnSuccessListener(new OnSuccessListener<QuerySnapshot>() {
        @Override
        public void onSuccess(QuerySnapshot queryDocumentSnapshots) {
          Map<String, QueryDocumentSnapshot> docList = new ArrayMap<>();
          for(QueryDocumentSnapshot doc : queryDocumentSnapshots)
          {
            docList.put(doc.getId(),doc);
          }
          QueryDocumentSnapshot quesListDoc  = docList.get("QUESTIONS_LIST");
          String count = quesListDoc.getString("COUNT");
          for(int i=0; i < Integer.valueOf(count); i++)
          {
            String quesID = quesListDoc.getString("Q" + String.valueOf(i+1) + "_ID");
            QueryDocumentSnapshot quesDoc = docList.get(quesID);
            quesList.add(new QuestionModel(
                quesID,
                quesDoc.getString("QUESTION"),
                quesDoc.getString("A"),
                quesDoc.getString("B"),
                              quesDoc.getString("C"),
                              quesDoc.getString("D"),
                              Integer.valueOf(quesDoc.getString("ANSWER"))
                          )); }
                        adapter = new QuestionAdapter(quesList);
  quesView.setAdapter(adapter);
     loadingDialog.dismiss();
                  }
              })
              .addOnFailureListener(new OnFailureListener() {
                  @Override
                  public void onFailure(@NonNull Exception e) {
Toast.makeText(QuestionsActivity.this,e.getMessage(),Toast.LENGTH_SHORT).show();
                    loadingDialog.dismiss();
                  }
              }); }
    @Override
    protected void onResume() {
        super.onResume();
        if(adapter != null) {
            adapter.notifyDataSetChanged();
        }
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if(item.getItemId() == android.R.id.home)
        {
            finish();
        }
        return super.onOptionsItemSelected(item);
    }}
```

activity_questions.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F0D6D6"
    tools:context=".QuestionsActivity">
<androidx.appcompat.widget.Toolbar
        android:id="@+id/q_toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/colorPrimary"
        android:theme="@style/ThemeOverlay.AppCompat.Dark"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">
</androidx.appcompat.widget.Toolbar>
<androidx.recyclerview.widget.RecyclerView
        android:id="@+id/quest_recycler"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        app:layout_constraintBottom_toTopOf="@+id/addQB"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/q_toolbar" >
</androidx.recyclerview.widget.RecyclerView>
<Button
        android:id="@+id/addQB"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:padding="16dp"
        android:text="Add New Question"
        android:textColor="@color/white"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

QuestionModel.java

```
package com.example.admin;


public class QuestionModel {
    private String quesID;
    private String question;
    private String optionA;
    }

    private String optionB;
    private String optionC;
    private String optionD;
    private int correctAns;

    public QuestionModel(String quesID, String question, String optionA, String optionB,
String optionC, String optionD, int correctAns) {
        this.quesID = quesID;
        this.question = question;
        this.optionA = optionA;
        this.optionB = optionB;
        this.optionC = optionC;
        this.optionD = optionD;
        this.correctAns = correctAns;
    }
    public String getQuesID() {
        return quesID;
    }
    public void setQuesID(String quesID) {
        this.quesID = quesID;
    }
    public String getQuestion() {
        return question;
    }
    public void setQuestion(String question) {
        this.question = question;
    }
    public String getOptionA() {
        return optionA;
    }
    public void setOptionA(String optionA) {
        this.optionA = optionA;
    }
    public String getOptionB() {
        return optionB;
    }
    public void setOptionB(String optionB) {
        this.optionB = optionB;
    }
    public String getOptionC() {
        return optionC;
    }
    public void setOptionC(String optionC) {
        this.optionC = optionC;
    }
    public String getOptionD() {
        return optionD;
    }
    public void setOptionD(String optionD) {
        this.optionD = optionD;
    }
    public int getCorrectAns() {
        return correctAns;
   }
    public void setCorrectAns(int correctAns) {
        this.correctAns = correctAns;
    }
}
```

QuestionDetailsActivity.java

```
package com.example.admin;

import static com.example.admin.CategoryActivity.catList;
import static com.example.admin.CategoryActivity.selected_cat_index;
import static com.example.admin.QuestionsActivity.quesList;
import static com.example.admin.SetsActivity.selected_set_index;
import static com.example.admin.SetsActivity.setsIDs;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import android.app.Dialog;
import android.os.Bundle;
import android.util.ArrayMap;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.firestore.FirebaseFirestore;
import java.util.Map;

public class QuestionDetailsActivity extends AppCompatActivity {
   private EditText ques, optionA, optionB, optionC, optionD, answer;
   private Button addQB;
   private String qStr, aStr, bStr, cStr, dStr, ansStr;
   private Dialog loadingDialog;
   private FirebaseFirestore firestore;
```

```
    private String action;
    private int qID;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_question_details);
        Toolbar toolbar = findViewById(R.id.qdetails_toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        ques = findViewById(R.id.question);
        optionA = findViewById(R.id.optionA);
        optionB = findViewById(R.id.optionB);
        optionC = findViewById(R.id.optionC);
        optionD = findViewById(R.id.optionD);
        answer = findViewById(R.id.answer);
        addQB = findViewById(R.id.addQB);
        loadingDialog = new Dialog(QuestionDetailsActivity.this);
        loadingDialog.setContentView(R.layout.loading_progressbar);
        loadingDialog.setCancelable(false);
loadingDialog.getWindow().setBackgroundDrawableResource(R.drawable.progress_backgro
und);

loadingDialog.getWindow().setLayout(ViewGroup.LayoutParams.WRAP_CONTENT,ViewGro
up.LayoutParams.WRAP_CONTENT);
        firestore = FirebaseFirestore.getInstance();
        action = getIntent().getStringExtra("ACTION");
        if(action.compareTo("EDIT") == 0)
        {  qID = getIntent().getIntExtra("Q_ID",0);
            loadData(qID);
            getSupportActionBar().setTitle("Question " + String.valueOf(qID + 1));
            addQB.setText("UPDATE");
        }else{
```

```
        getSupportActionBar().setTitle("Question " + String.valueOf(quesList.size() + 1));
        addQB.setText("ADD");
      }
      addQB.setOnClickListener(new View.OnClickListener() {
        @Override
public void onClick(View v) {
          qStr = ques.getText().toString();
          aStr = optionA.getText().toString();
          bStr = optionB.getText().toString();
          cStr = optionC.getText().toString();
          dStr = optionD.getText().toString();
          ansStr = answer.getText().toString();
          if(qStr.isEmpty()) {
            ques.setError("Enter Question");
            return; }
          if(aStr.isEmpty()) {
            optionA.setError("Enter option A");
            return; }
          if(bStr.isEmpty()) {
            optionB.setError("Enter option B ");
            return;  }
          if(cStr.isEmpty()) {
            optionC.setError("Enter option C");
            return; }
          if(dStr.isEmpty()) {
            optionD.setError("Enter option D");
            return; }
          if(ansStr.isEmpty()) {
            answer.setError("Enter correct answer");
            return; }
          if(action.compareTo("EDIT") == 0)
          {
```

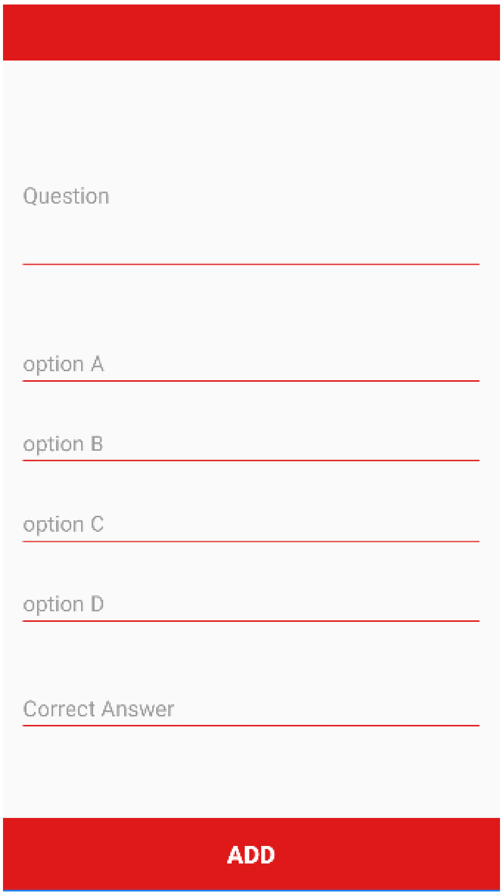
```
          editQuestion();
        }
        else {
          addNewQuestion();
        }
      }
    });
  }
  private void addNewQuestion()
  {
    loadingDialog.show();
    Map<String,Object> quesData = new ArrayMap<>();
    quesData.put("QUESTION",qStr);
    quesData.put("A",aStr);
    quesData.put("B",bStr);
    quesData.put("C",cStr);
    quesData.put("D",dStr);
 quesData.put("ANSWER",ansStr);
    final String doc_id =
firestore.collection("QUIZ").document(catList.get(selected_cat_index).getId())
        .collection(setsIDs.get(selected_set_index)).document().getId();
    firestore.collection("QUIZ").document(catList.get(selected_cat_index).getId())
        .collection(setsIDs.get(selected_set_index)).document(doc_id)
        .set(quesData)
        .addOnSuccessListener(new OnSuccessListener<Void>() {
          @Override
          public void onSuccess(Void aVoid) {
            Map<String,Object> quesDoc = new ArrayMap<>();
            quesDoc.put("Q" + String.valueOf(quesList.size() + 1) + "_ID", doc_id);
            quesDoc.put("COUNT",String.valueOf(quesList.size() + 1));
            firestore.collection("QUIZ").document(catList.get(selected_cat_index).getId())
                .collection(setsIDs.get(selected_set_index)).document("QUESTIONS_LIST")
        .update(quesDoc)
        .addOnSuccessListener(new OnSuccessListener<Void>() {
         @Override
         public void onSuccess(Void aVoid) {
          Toast.makeText(QuestionDetailsActivity.this, " Question Added
Successfully", Toast.LENGTH_SHORT).show();
          quesList.add(new QuestionModel(
            doc_id,
            qStr,aStr,bStr,cStr,dStr, Integer.valueOf(ansStr)
       ));
          loadingDialog.dismiss();
          QuestionDetailsActivity.this.finish();
         }
        })
        .addOnFailureListener(new OnFailureListener() {
         @Override
         public void onFailure(@NonNull Exception e) {
Toast.makeText(QuestionDetailsActivity.this,e.getMessage(),Toast.LENGTH_SHORT).show();
          loadingDialog.dismiss();
         } });
     } })
    .addOnFailureListener(new OnFailureListener() {
     @Override
     public void onFailure(@NonNull Exception e) {
Toast.makeText(QuestionDetailsActivity.this,e.getMessage(),Toast.LENGTH_SHORT).show();
      loadingDialog.dismiss();
     }
    });
 }
 private void loadData(int id)
 {
  ques.setText(quesList.get(id).getQuestion());
```

```
        optionA.setText(quesList.get(id).getOptionA());
        optionB.setText(quesList.get(id).getOptionB());
        optionC.setText(quesList.get(id).getOptionC());
        optionD.setText(quesList.get(id).getOptionD());
        answer.setText(String.valueOf(quesList.get(id).getCorrectAns()));
    }
    private void editQuestion()
    {
        loadingDialog.show();
        Map<String,Object> quesData = new ArrayMap<>();
        quesData.put("QUESTION", qStr);
        quesData.put("A",aStr);
        quesData.put("B",bStr);
        quesData.put("C",cStr);
        quesData.put("D",dStr);
 quesData.put("ANSWER",ansStr);
        firestore.collection("QUIZ").document(catList.get(selected_cat_index).getId())
.collection(setsIDs.get(selected_set_index)).document(quesList.get(qID).getQuesID())
                .set(quesData)
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        Toast.makeText(QuestionDetailsActivity.this,"Question updated
successfully",Toast.LENGTH_SHORT).show();
                        quesList.get(qID).setQuestion(qStr);
                        quesList.get(qID).setOptionA(aStr);
                        quesList.get(qID).setOptionB(bStr);
                        quesList.get(qID).setOptionC(cStr);
                        quesList.get(qID).setOptionD(dStr);
                        quesList.get(qID).setCorrectAns(Integer.valueOf(ansStr));
                        loadingDialog.dismiss();
                        QuestionDetailsActivity.this.finish();
                }
            })
            .addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
Toast.makeText(QuestionDetailsActivity.this,e.getMessage(),Toast.LENGTH_SHORT).show();
                    loadingDialog.dismiss();
                }
            });
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if(item.getItemId() == android.R.id.home)
        {
            finish();
        }
        return super.onOptionsItemSelected(item);
    }
}
```

activity_question_details.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".QuestionDetailsActivity">
<androidx.appcompat.widget.Toolbar
        android:id="@+id/qdetails_toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/colorPrimary"
        android:theme="@style/ThemeOverlay.AppCompat.Dark"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"/>
    <Button
        android:id="@+id/addQB"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Add"
        android:padding="16dp"
        android:background="@color/colorPrimary"
        android:textColor="@color/white"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="8dp"
        android:orientation="vertical"
        android:gravity="center"
        app:layout_constraintBottom_toTopOf="@+id/addQB"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/qdetails_toolbar">
<EditText
      android:id="@+id/question"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:ems="10"
      android:layout_marginBottom="50dp"
      android:backgroundTint="@color/colorPrimary"
      android:textColor="@android:color/black"
      android:textCursorDrawable="@null"
      android:inputType="textMultiLine"
      android:maxLines="7"
      android:lines="5"
      android:hint="Question" />
<EditText
      android:id="@+id/optionA"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:ems="10"
      android:layout_marginBottom="20dp"
      android:backgroundTint="@color/colorPrimary"
      android:textColor="@android:color/black"
      android:textCursorDrawable="@null"
      android:inputType="textPersonName"
      android:hint="option A" />
<EditText
      android:id="@+id/optionB"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:ems="10"
      android:layout_marginBottom="20dp"
      android:backgroundTint="@color/colorPrimary"
```

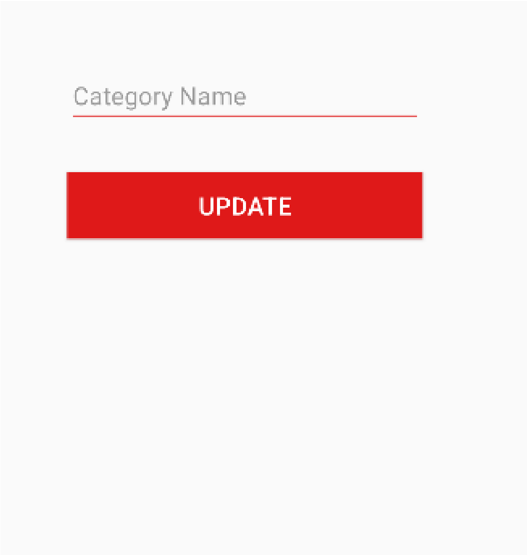
```
        android:textColor="@android:color/black"
        android:textCursorDrawable="@null"
        android:inputType="textPersonName"
        android:hint="option B" />
<EditText
        android:id="@+id/optionC"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:layout_marginBottom="20dp"
        android:backgroundTint="@color/colorPrimary"
        android:textColor="@android:color/black"
        android:textCursorDrawable="@null"
        android:inputType="textPersonName"
        android:hint="option C" />
<EditText
        android:id="@+id/optionD"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:layout_marginBottom="20dp"
        android:backgroundTint="@color/colorPrimary"
        android:textColor="@android:color/black"
        android:textCursorDrawable="@null"
        android:inputType="textPersonName"
        android:hint="option D" />
<EditText
        android:id="@+id/answer"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:layout_marginBottom="20dp"
```



```
            android:layout_marginTop="20dp"
            android:backgroundTint="@color/colorPrimary"
            android:textColor="@android:color/black"
            android:textCursorDrawable="@null"
            android:inputType="textPersonName"
            android:hint="Correct Answer" />
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

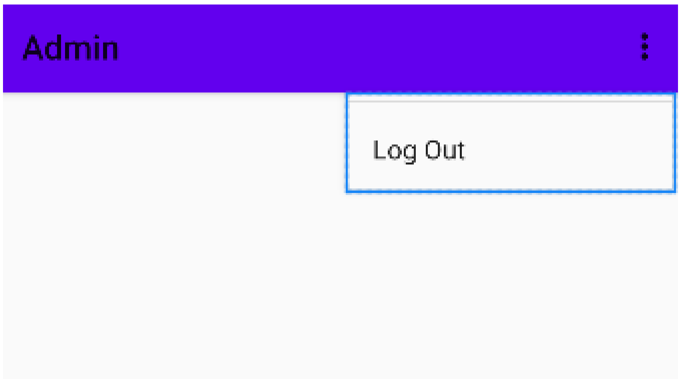
edit_category_dialog.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="350dp"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_margin="16dp"
    android:padding="16dp"
    android:layout_height="wrap_content">
<EditText
        android:id="@+id/ec_cat_name"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="32dp"
        android:layout_marginEnd="32dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="Category Name"
        android:textColor="@android:color/black"
        android:backgroundTint="@color/colorPrimary"
        android:textCursorDrawable="@null"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/ec_add_btn"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="32dp"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="32dp"
        android:background="@color/colorPrimary"
        android:text="UPDATE"
        android:textColor="@color/white"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/ec_cat_name" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



menu_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
<item
   android:id="@+id/idLogOut"
  android:title="Log Out"
  android:icon="@drawable/ic_logout"/>
</menu>
```



AndroidManifest.xml

```xml
<?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.admin">
<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.Admin"
    tools:targetApi="31">
<activity
      android:name=".QuestionDetailsActivity"
      android:exported="false" />
<activity
      android:name=".QuestionsActivity"
      android:exported="false" />
<activity
      android:name=".SetsActivity"
      android:exported="false" />
<activity
      android:name=".CategoryActivity"
      android:exported="false" />
<activity
      android:name=".MainActivity"
      android:exported="true">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
</manifest>
```

TESTING:

Testing is a process of executing a program with the indent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding.

System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

TESTING OBJECTIVES:

1. Testing is a process of executing a program with the intent of finding an error

2. A good test case is one that has a probability of finding an as yet undiscovered error

3. A successful test is one that uncovers an undiscovered error

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are

- White box testing.
- Black box testing.

WHITE-BOX TESTING:

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

BLACK-BOX TESTING:

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

UNIT TESTING:

Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important paths are tested to uncover errors within the boundary of the modules. These tests were carried out during the programming stage itself. All units of Vienna SQL were successfully tested.

INTEGRATION TESTING:

Integration testing focuses on unit tested modules and build the program structure that is dictated by the design phase.

SYSTEM TESTING:

System testing tests the integration of each module in the system. It also tests to find discrepancies between the system and its original objective, current specification and system documentation. The primary concern is the compatibility of individual modules. Entire system is working properly or not will be tested here. Top-down testing implementing here.

ACCEPTANCE TESTING:

This testing is done to verify the readiness of the system for the implementation. Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

ALPHA TESTING:

Alpha testing is a type of acceptance testing; performed to identify all possible issues/bugs before releasing the product to everyday users or public. The focus of this testing is to simulate real users by using black box and white box techniques. The aim is to carry out the tasks that a typical user might perform. Alpha testing is carried out in a lab environment and usually the testers are internal employees of the organization. To put it as simple as possible, this kind of testing is called alpha only because it is done early on, near the end of the development of the software, and before beta testing. Alpha Testing performed at developer's site.

BETA TESTING:

Beta Testing of a product is performed by "real users" of the software application in a "real environment" and can be considered as a form of external user acceptance testing. Beta version of the software is released to a limited number of end-users of the product to obtain feedback on the product quality. Beta testing reduces product failure risks and provides increased quality of the product through customer validation. It is the final test before shipping a product to the customers. Direct feedback from customers is a major advantage of Beta Testing. This testing helps to tests the product in real time environment. Beta testing is performed at client location or end user of the product.

LIMITATIONS

This is not a real life project. I just did it as my student project to add in my course. Like the things this projects also has some limitations and can further be enhances by someone, because there are certain drawbacks that do not permit the system to be 100% accurate.

CONCLUSION AND FUTURE ENHANCEMENT

In the conclusion I would like to say that I am thankful to my teacher for completion of the project. This is a big idea in small package. In future I will do more work to give it real life view. I think this project will help me to guide in future career to step ahead.

REFERENCES

1. CMC study material

WEBSITES

www.geeksforgeeks.org

www.codementor.io

www.stackoverflow.com