**G.L.A UNIVERSITY, MATHURA**

A Project Report

On

# “Quizyfy”



**Submitted by:**

**Aryan Garg Univ. Roll No-181500135**

**Kush Sharma Univ. Roll No-181500343**

**Priyanshu Shukla Univ. Roll No -181500515**

**Vinod Univ. Roll No-181500799**

**Ritik Goyal Univ . Roll No – 181500575**

**Under The Guidance of**

**Mr.Neeraj Khanna**

**Technical Trainer (Department of CSE)**

## 

## 

## 

## 

## GLA UNIVERSITY

## MATHURA, UTTAR PRADESH

**Session: 2020-21**

## 

## DECLARATION

I hereby declare that the work which is being presented in the Mini Project **“QuizyFy”,** in partial fulfillment of the requirements for Mini project Lab is an authentic record of our own work carried under the supervision of **Mr. Neeraj Khanna, Technical Trainer.**

**Aryan Garg**

**Kush Sharma**

**Priyanshu Shukla**

**Ritik Goyal**

### ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the Project Work, undertaken during

B. Tech. Final Year. We owe special debt of gratitude to **Mr. Neeraj Khanna Technical Trainer,** Department of Computer Science &Technology, GLA University, Mathurafor his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature:

Name :

Roll No.:

Date :

Signature:

Name :

Roll No.:

Date :

Signature:

Name :

Roll No.:

Date :

Signature:

Name :

Roll No.:

Date :

## ABBREVIATIONS AND ACRONYMS

**OS O**perating **S**ystem

**UI U**ser **I**nterface

**MCQ M**ultiple **C**hoice **Q**uestion

**RAM R**andom **A**ccess **M**emory

**JDK J**ava **D**evelopment **K**it

**SDK S**oftware **D**evelopmen**t K**it

**API A**pplication **P**rogram **I**nterface

**IDE I**ntegrated **D**evelopment **E**nvironment

**EER E**nhanced **E**ntity **R**elationship

## ANDROID CERTIFICATE



****

****

# INTRODUCTION

## 1 Overview

In today’s world, Smart phones have changed our lives and have become an indispensable part of our lives because of its specialty to simplify our routine work and thereby saving our time. A Smartphone with an Android OS offers excellent functionality to the users offering a distinct experience. Android is a Linux based operating system and it was bought by Google in 2007.There are tons of application available and one of the prime reason for this vast number is android being an open source. On the other hand, android based device like mobile, tab are very user friendly. A survey has done by “Light Castle Partners” research wing which indicates that though other operating system mobile users exist but the majority users are goes with android operating system [1].

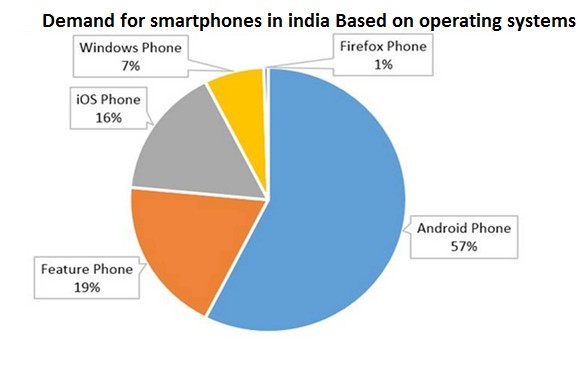


Figure 1.1: A survey result on a random sample of different age groups of mobile phone users

in India, based on their preferred operating systems [1]*.*

In this context, Project application is developed based on android platform. The name of application is define as **‘**QuizyFy”. Aims of this project is to develop an android platform supported Quiz application named “QuizyFy”. It is an online quiz application where user with admin privilege can do administrative task like add, delete and edit from application user interface (UI) and interviewee can participate for MCQ quiz with time limit. So, goals of this project to facilitate users to configure quizzes as well as giving quizzes with this android based smart phone. User friendly environment configuration is also another goal for this project application*.*

## 2Objectives

The main objective of “QuizyFy” 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.

## 3 Motivation

Currently most of the Examination like organizational recruitment, University class test are paper based, which costs time and resources. Questionnaire is developed, printed, and then collect data, entry, editing, cleaning, which time consuming and costly. Proposed application is the starting for avoid those circumstances which are been currently faced by any organization.

## 4Summary

Dramatic breakthroughs in processing power along with the number of extra features included in mobile devices have opened the doors to a wide range of commercial possibilities. In particular, most cell phones regularly include processors comparable to PCs and internet access from a few years ago. With all these added abilities, Online Quiz application is design for

Android based system mobile.

**Proposed Model**

## Purpose of the project

This Project main purpose is to develop Quiz system named ‘QuizyFy. The application (QuizyFy) will provide online based quiz with multiple choice question (MCQ). This quiz application will support android base operating system. With this application, users or any organization can perform actions like

* Administrative Task
* Interview Task

### Quiz Task

Select any Quiz or subject which he /she wants to give Examination. One interviewer can give Quiz only for one time. Once a quiz has finished, it become inactive to that user. Finally, Score can be shown considering the quiz has been taken from any individuals. As a result, the following tasks are define as interviewee task.

• Select Desire Quiz

* Answer questions within set time (by admin)
* Finish Quiz

**Flow Chart:**

The flowchart is a means of visually presenting the flow of data through an information processing system, the operations performed within the system and the sequence in which they are performed. In this lesson, we shall concern ourselves with the program flowchart, which describes what operations (and in what sequence) are required to solve a given problem. The program flowchart can be likened to the blueprint of a building. As we know a designer draws a blueprint before starting construction on a building. Similarly, a programmer prefers to draw a flowchart prior to writing a computer program. As in the case of the drawing of blueprint, the flowchart is drawn according to defined rules and standard flowchart symbols prescribed by the American National Standard

Institute,Inc.

**Symbols used to make flowchart:**

Start or end of the program

Computational steps or processing function of a program

Input or output

Decision making and branching

Connector or joining of two parts of program

Off page connector

Flow lines

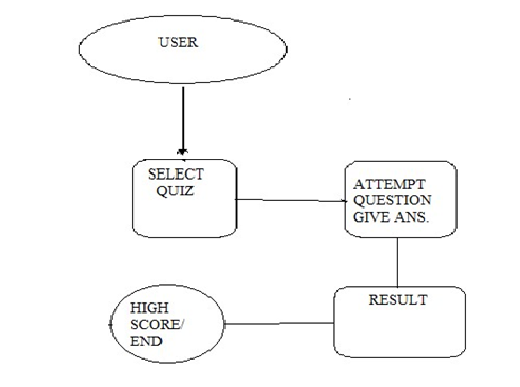


Figure 2.2: Flow Chart for ‘QuizyFy’

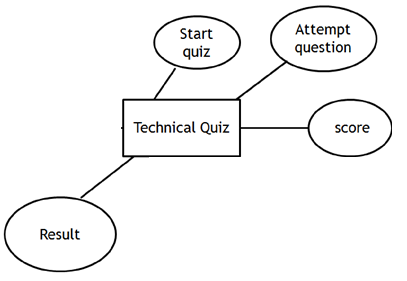
### Description for flow chart

1. Role for that particular user will verify from server database.
2. If user is administrator, following options shall be managed.
   * User
   * Welcome
   * Quiz
   * Question and Answer
   * Score

Above three options can be add, edit and delete by administrator.

1. Score report will show to administrator in read only mode.
2. If user is interviewee, all active quiz list will enable for giving test.
3. Quiz will execute with certain time limit. Quiz will finish after that time or all question answered. .

## E-R Diagram



Entity

Multivalued attribute

Primary key

Attribute

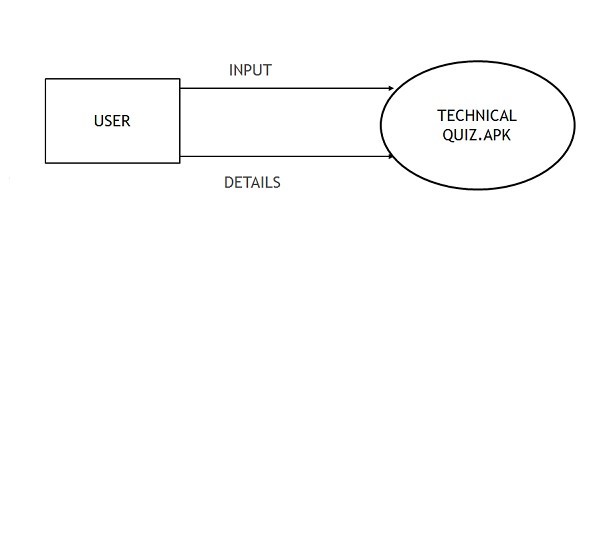
Relation

### Data Flow Diagram (DFD)

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system. DFDs can be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

A DFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart, which shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored.

## DFD Level 0;



**Chapter 3**

**Implementation**

### Technologies Used

* Microsoft Windows 7
* Java Development Kit
* Android SDK
* SQLite Database
* XML

**Microsoft Windows 7**

It should be mentioned that all tools and technology are installed for development work at windows 7 operating system 64 bit platform. Windows 7 is a personal computer operating system developed by Microsoft. It is a part of Windows NT family of operating systems. Development of Windows 7 started as early as 2006 under the codename "Blackcomb."

Windows 7 was released to manufacturing on July 22, 2009, and became generally available on

October 22, 2009, less than three years after the release of its predecessor, Windows Vista.

Minimum hardware requirements for Windows 7 is given below [2].

DVD-ROM drive (Only to install from DVD-ROM media)

Component

Operating system architecture

32-

bit

64-

bit

Processor

1

GHz IA-32 processor

1

GHz x86-64 processor

Memory

)

(

RAM

1

GB

2

GB

Graphics

card

DirectX 9 graphics processor with WDDM driver model 1.0

(

Not absolutely necessary; o nly required for Aero

)

Free hard

drive space

16

GB

20

GB

Optical drive

Table I: Windows 7 Minimum Hardware Requirement [2]

The maximum amount of RAM that Windows 7 supports varies depending on the product

edition and on the processor architecture, as shown below figure

Edition

Processor architecture

IA-32 (32-bit)

x64 (64-bit)

Ultimate

4

GB

192

GB

Enterprise

Professional

Home Premium

16

GB

Home Basic

8

GB

Starter

2

GB

Table II: Physical memory limits of Windows 7 [2]

3.1.2 Java Development Kit (JDK)

A Java Development Kit (JDK) is a program development environment for writing Java applets and applications. It consists of a runtime environment that "sits on top" of the operating system layer as well as the tools and programming that developers need to compile, debug, and run

javac – the Java compiler, which converts source code into Java bytecode javadoc – the documentation generator, which automatically generates documentation from source code comments jar – the archiver, which packages related class libraries into a single JAR file. This tool also helps manage JAR files.

jstack – utility which prints Java stack traces of Java threads (experimental) jstat – Java Virtual Machine statistics monitoring tool (experimental) jstatd – jstat daemon (experimental) keytool – tool for manipulating the keystore pack200 – JAR compression tool policytool – the policy creation and management tool, which can determine policy for a

Java runtime, specifying which permissions are available for code from various sources Visual

VM – visual tool integrating several command-line JDK tools and lightweight performance and

#### Android SDK

The Android SDK (software development kit) is a set of development tools used to develop applications for Android platform. The Android SDK includes the following:

* Required libraries
* Debugger
* An emulator
* Relevant documentation for the Android application program interfaces (APIs)
* Sample source code
* Tutorials for the Android OS

Every time Google releases a new version of Android, a corresponding SDK is also released. To be able to write programs with the latest features, developers must download and install each version’s SDK for the particular phone [5].

API Level is an integer value that uniquely identifies the framework API revision offered by a version of the Android platform. The Android platform provides a framework API that applications can use to interact with the underlying Android system.

The framework API consists of:

* A core set of packages and classes
* A set of XML elements and attributes for declaring a manifest file
* A set of XML elements and attributes for declaring and accessing resources
* A set of Intents
* A set of permissions that applications can request, as well as permission enforcements included in the system The API Level identifier serves a key role in ensuring the best possible experience for users and application developers:
* It lets the Android platform describe the maximum framework API revision that it supports
* It lets applications describe the framework API revision that they require
* It lets the system negotiate the installation of applications on the user's device, such that version-incompatible applications are not installed.

The table below specifies the API Level supported by each version of the Android platform [6].

Platform Version API Level VERSION\_CODE

Android 7.0 25 NOUGAT

Android 6.1 24 MARSHMALLOW

Android 6.0 23 MARSHMALLOW

Android 5.1 22 LOLLIPOP\_MR1 Android 5.0 21 LOLLIPOP

Android 4.4W 20 KITKAT\_WATCH

Android 4.4 19 KITKAT Android 4.3 18 JELLY\_BEAN\_MR2

Android 4.2, 4.2.2 17 JELLY\_BEAN\_MR1 Android 4.1, 4.1.1 16 JELLY\_BEAN

Android 4.0.3, 4.0.4 15 ICE\_CREAM\_SANDWICH\_MR1

Android 4.0, 4.0.1, 4.0.2 14 ICE\_CREAM\_SANDWICH

Android 3.2 13 HONEYCOMB\_MR2

Android 3.1.x 12 HONEYCOMB\_MR1

Android 3.0.x 11 HONEYCOMB

Android 2.3.4

Android 2.3.3

10

GINGERBREAD\_MR1

Android 2.3.2

Android 2.3.1

Android 2.3

9

GINGERBREAD

Android 2.2.x

8

FROYO

Android 2.1.x

7

ECLAIR\_MR1

Android 2.0.1

6

ECLAIR\_0\_1

Android 2.0

5

ECLAIR

Android 1.6

4

DONUT

Android 1.5

3

CUPCAKE

Android 1.1

2

BASE\_1\_1

Android 1.0

1

BASE

Table III: API Level supported by each version of the Android platform [6]

Applications can use a manifest element provided by the framework API — <usessdk> — to describe the minimum and maximum API Levels under which they are able to run, as well as the preferred API Level that they are designed to support. The element offers three key

attributes:

* android:minSdkVersion — Specifies the minimum API Level on which the application is able to run. The default value is "1".
* android:targetSdkVersion — Specifies the API Level on which the application is designed to run. In some cases, this allows the application to use manifest elements or behaviors defined in the target API Level, rather than being restricted to using only those defined for the minimum

API Level.

* android:maxSdkVersion — Specifies the maximum API Level on which the application is able

to run .

**Tools Used**

#### Android Studio

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 .It offer tools custom-tailored for Android developers, including rich code editing, debugging, testing, and profiling tools.

* Operating System
  + A recent version of Windows, OS X or Ubuntu
* Memory
  + 2 GB RAM (available memory, rather than total memory)
* Java Runtime
  + Oracle JDK (no other brand of Java is suitable)

Studio is Appcelerator's free IDE (integrated development environment). It can use to write, test,

and debug mobile applications. Studio also has integrated templates and sample applications to make it even easier to get started creating your own apps. In addition, Studio will help to manage Titanium SDK updates and module usage.

Titanium exists as a bridge between the native operating system and app's code.

The following graphic illustrates this architecture [12]:

At the bottom of the stack is the client operating system: Android, iOS, or the browser (for

Mobile Web applications). At the top is desire app, built JavaScript. In between, is the

Titanium SDK and the APIs it exposes. Application wrote in JavaScript, calling on the Titanium APIs to take actions like drawing buttons, opening windows, showing the camera, etc. The Titanium Bridge (part of the SDK) translates those calls into their native equivalents. In other words, when create a Titanium button, it's actually a proxy for a true native button. When you modify the Titanium button, say to change its label or add an event listener, Kroll applies corresponding changes to the native equivalent.

When events occur in native-land, Kroll bubbles them up to your JavaScript code [l2].

### System Design

#### 1.Admin Panel

Administrator has privilege to access four options. They are

1. Quiz
2. Question and Answer
3. Report
4. User

1. Quiz

Administrator can add quiz with following four options. They are

* Quiz Name

Name of the Quiz will define here.

* Active or Inactive

Currently any specific quiz is active or not. If any quiz is inactive (update by administrator) then

that quiz will not be shown to interviewee.

* Show result or not , If show result option is mark with ‘yes’ then any individual interviewee will see score after finish Quiz. If ‘not’ mark, then can’t see the result.
* Quiz Time

Total time for quiz can be set from here only by administrator.

Application will communicate with webserver API and connect with database to extract all quiz list and will show to administrator. Any quiz can be edit or delete from that list.

## 2. Question and Answers

Four multiple answer (max) can input by the administrator. Administrator can also set less than four multiple (i.e. two, three) answer for any question.

* Question

Question will be written here.

* Answers:

Four multiple answer (max) can input by the administrator. Administrator can also set less than four multiple (i.e. two, three) answer for any question.

* Order No:

Question order can be set here.

For Update or Delete any question, Administrator need to choose any quiz. All questions

will visible to administrator for that particular quiz.

## 3. Report

Report will show for any particular quiz. Username of interviewee and score will be shown to administrator

## 4. Users

Administrator can add quiz with following four options. They are

* User Name

User name will be provided as the user’s email address.

* Role

Role will be define as ‘Admin’ or ‘Interviewee’. If any user has ‘Admin’ role then he/she can set Application parameters. If any user has ‘Interviewee’ role then he/she can sit for any quiz.

After add any user, an automatic email to send to that user by system.

Application will communicate with webserver API and connect with database to extract all user list and will show to administrator. Any user can be edit or delete from that list

### Summary

The Project is developed in Java dynamic programming language by using the “Android

Studio” Integrated Development Environment (IDE). Android development Tool (ADT)

and Android software development kit (SDK) integrated with Titanium studio to develop mobile applications on the Android platform. XML (recursive acronym for XML: eXtensiblemarkup Language) is used as backend which work as application program interface (API). SQLITE Database is use for database configuration.

**User Manual**

### System Requirement

Device

Operating

System (OS)

OS Version

RAM

Disk Space

Android

Mobile/Tab

Android

4.0

.x to

higher

64

MB or

higher

18

Mb

Table V: Minimum System Requirement

### Prerequisite

Internet Connection must active.

### Installation

To install the application in android mobile device or tablet device user needed to run the setup package named as ‘QuizyFy.apk ’. This package can store both in android device memory and external memory card (micro SD card that connect with android device). Package need maximum 9 Mb space. After copy it to android device, following steps should be followed:

 Tap the folder where the ‘QuizyFy’ is store.  Tap the application package ‘QuizyFy’.

After selecting the package, application lets user to install

Tap install

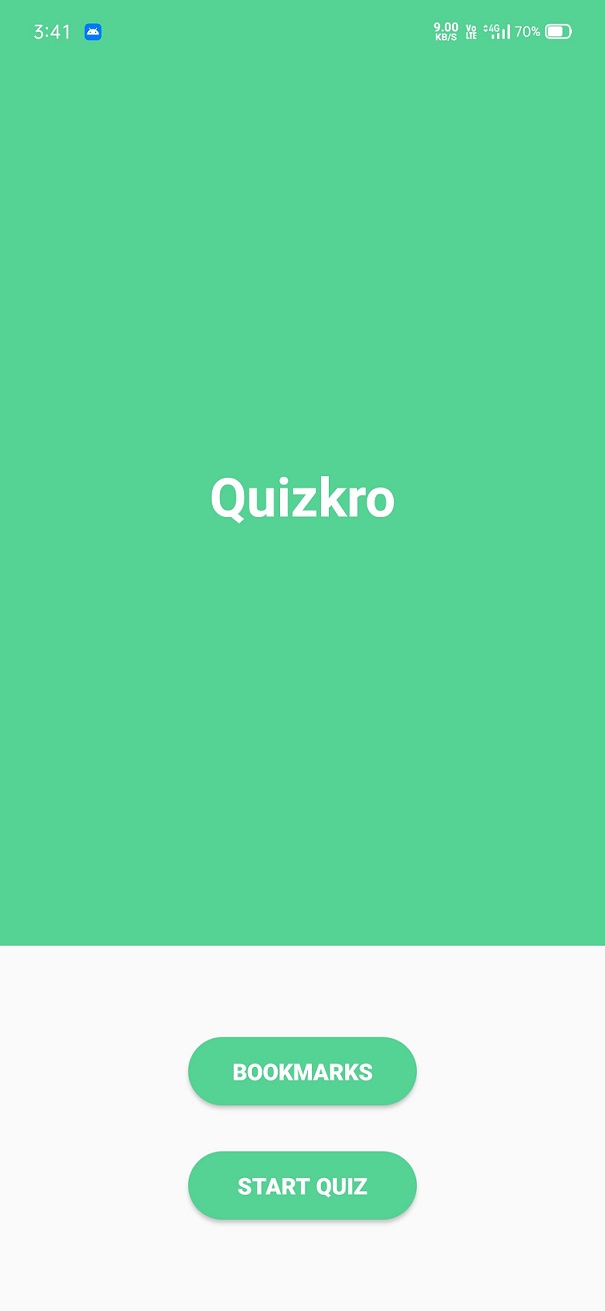
Installation is done and application is ready for use.

### Getting Started

Tap the icon to open the application.

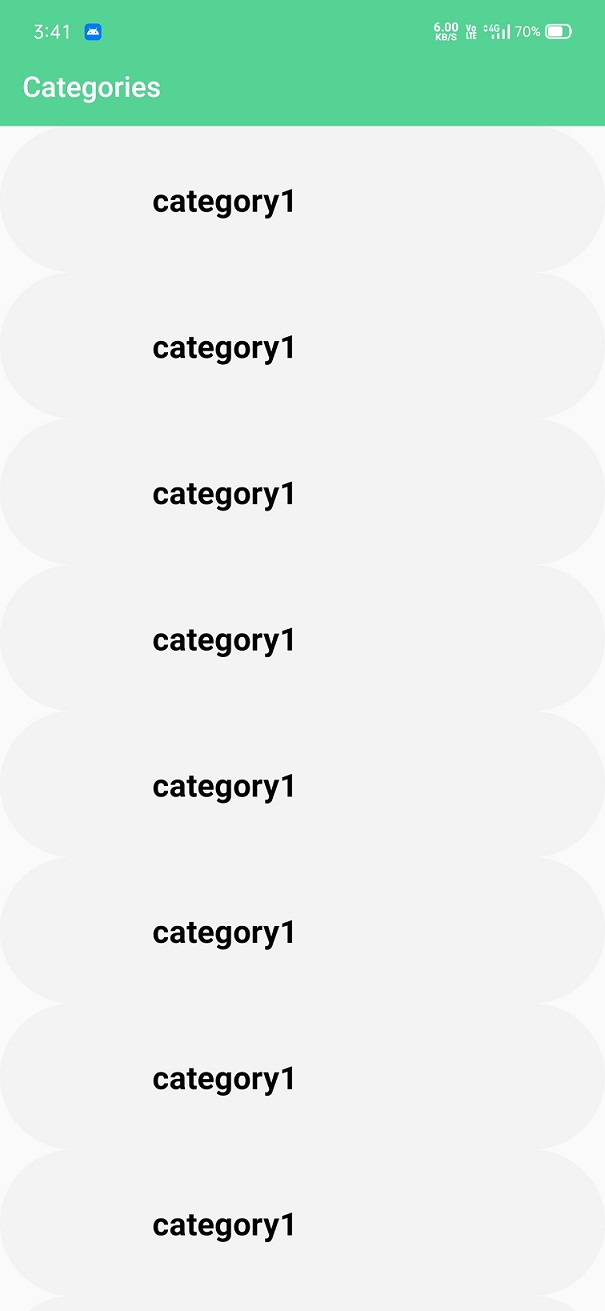
**Home**

Home screen will appear first, when the user start the app & it will appear for 4-5 second.



#### Category

Choose category which category quiz you want to take.

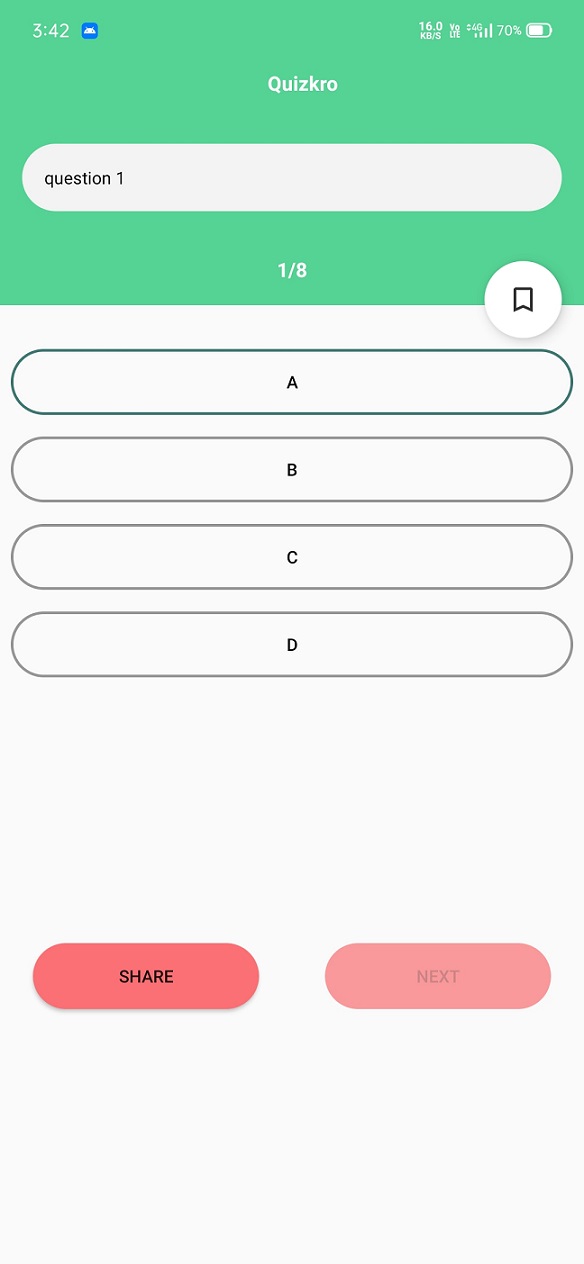


#### Quiz Management

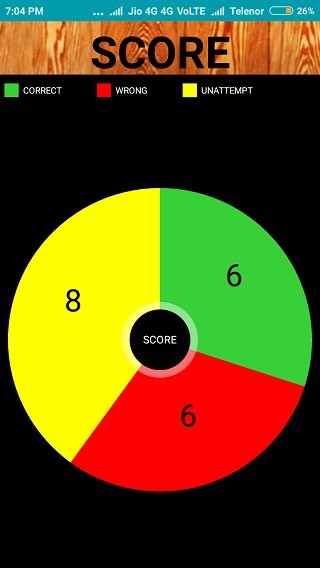
#### User can select the sets , there are different number of sets.

#### C:\Users\kushs\Desktop\2.jpeg

#### Sample questions



#### Result



**Coding**

## Android Lanuch Activity

**packagecom.example.hp.myquiz;**

**import android.app.Activity;**

**import android.content.Intent;**

**import android.os.Bundle;**

**import android.view.animation.Animation; import android.view.animation.AnimationUtils; import android.widget.LinearLayout;**

**public class Android\_Launch extends Activity{**

**LinearLayoutlinearLayout;**

**Animation animLayout; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.android\_launch); linearLayout=(LinearLayout)findViewById(R.id.l\_layout);**

**animLayout=AnimationUtils.loadAnimation(this,**

**R.anim.text\_view);**

**linearLayout.startAnimation(animLayout);**

**Thread timer=new Thread(){**

**public void run() { try { sleep(3000 );**

**}catch (InterruptedException e)**

**{**

**e.printStackTrace(); } finally { Intent i=newIntent("android.example.hp.myquiz.Levels");**

**i.putExtra("subject","Android"); startActivity(i); }**

**}**

**};timer.start();}**

**@Override protected void onPause() { super.onPause(); finish(); }**

**}**

## Login Activity

packagecom.example.hp.myquiz; import android.content.Intent; import android.os.Bundle; import android.support.annotation.NonNull; import android.support.v7.app.AppCompatActivity; import android.text.TextUtils; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.ProgressBar;

importandroid.widget.Toast;

importcom.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 LoginActivity extends AppCompatActivity {

privateEditTextinputEmail, inputPassword; private FirebaseAuthauth; private ProgressBarprogressBar; 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) { String email=auth.getCurrentUser().getEmail();

Intent i=new Intent(LoginActivity.this, Subjects.class);

i.putExtra("id",email); startActivity(i); finish();

}

// set the view now setContentView(R.layout.activity\_login);

inputEmail = (EditText) findViewById(R.id.email); inputPassword = (EditText) findViewById(R.id.password); progressBar = (ProgressBar) findViewById(R.id.progressBar); btnSignup = (Button) findViewById(R.id.btn\_signup); btnLogin = (Button) findViewById(R.id.btn\_login); btnReset = (Button) 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,

SignupActivity.class));

} }); btnReset.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View v) { startActivity(new Intent(LoginActivity.this,

ResetPasswordActivity.class));

} });

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)) {

Toast.makeText(getApplicationContext(), "Enter email address!", Toast.LENGTH\_SHORT).show(); return; }

if (TextUtils.isEmpty(password)) {

Toast.makeText(getApplicationContext(), "Enter password!", Toast.LENGTH\_SHORT).show(); return; }

progressBar.setVisibility(View.VISIBLE);

//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); if (!task.isSuccessful()) { // there was an error if (password.length() < 6) { inputPassword.setError(getStrin g(R.string.minimum\_password)); } else { Toast.makeText(LoginActivity.this,getString(R.string.auth\_failed),

Toast.LENGTH\_LONG).show();

}

} else {

String

email=auth.getCurrentUser().getEmail();

Intent i=new Intent(LoginActivity.this, Subjects.class);

i.putExtra("id",email); startActivity(i); finish(); }

} });

}

});

}

}

## SignUpActivity

packagecom.example.hp.myquiz;

import android.os.Bundle; import android.support.annotation.NonNull; import android.support.v7.app.AppCompatActivity; import android.text.TextUtils; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.ProgressBar; import android.widget.Toast; 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.google.firebase.auth.FirebaseUser; public class SignupActivity extends AppCompatActivity { private EditTextinputEmail, inputPassword; private Button btnSignIn, btnSignUp, btnResetPassword; private ProgressBarprogressBar; private FirebaseAuthauth; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_singnup);

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

btnSignIn = (Button) findViewById(R.id.sign\_in\_button); btnSignUp = (Button) findViewById(R.id.sign\_up\_button); inputEmail = (EditText) findViewById(R.id.email); inputPassword = (EditText) findViewById(R.id.password); progressBar = (ProgressBar) findViewById(R.id.progressBar); btnResetPassword = (Button) findViewById(R.id.btn\_reset\_password);

FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser(); auth = FirebaseAuth.getInstance(); if (auth.getCurrentUser() != null) { // User is logged in

String email=auth.getCurrentUser().getEmail();

Intent i=new Intent(SignupActivity.this, Subjects.class);

i.putExtra("id",email); startActivity(i); finish();

}

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

startActivity(new Intent(SignupActivity.this,

ResetPasswordActivity.class));

} });

btnSignIn.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View v) { startActivity(new Intent(SignupActivity.this, LoginActivity.class)); 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)) {

Toast.makeText(getApplicationContext(), "Enter email address!", Toast.LENGTH\_SHORT).show(); return;

}

if (TextUtils.isEmpty(password)) {

Toast.makeText(getApplicationContext(), "Enter password!", Toast.LENGTH\_SHORT).show(); return; }

if (password.length() < 6) {

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

progressBar.setVisibility(View.VISIBLE); //create user auth.createUserWithEmailAndPassword(email, password)

.addOnCompleteListener(SignupActivity.this, new

OnCompleteListener<AuthResult>() { @Override public void onComplete(@NonNull Task<AuthResult> task) {

Toast.makeText(SignupActivity.this, "createUserWithEmail:onComplete:" + task.isSuccessful(), Toast.LENGTH\_SHORT).show(); progressBar.setVisibility(View.GONE); if (!task.isSuccessful()) {

Toast.makeText(SignupActivity.this, "Authentication failed." + task.getException(), Toast.LENGTH\_SHORT).show();

} else {

String email=auth.getCurrentUser().getEmail();

Intent i=new Intent(SignupActivity.this, Subjects.class);

i.putExtra("id",email); startActivity(i); finish(); }

}

});

}

});

}

@Override protected void onResume() { super.onResume(); progressBar.setVisibility(View.GONE); }

}

## Welcome Activity

ackagecom.example.hp.myquiz; import android.app.Activity; import android.content.Intent; import android.os.Bundle; import android.view.animation.Animation; import android.view.animation.AnimationUtils; import android.widget.LinearLayout;

public class Welcome extends Activity{

LinearLayout; Animation animLayout; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.welcome); linearLayout=(LinearLayout)findViewById(R.id.welcome\_layout); animLayout= AnimationUtils.loadAnimation(this, R.anim.accelerate\_right); linearLayout.startAnimation(animLayout); Intent intent=getIntent(); final String email=intent.getStringExtra("id");

Thread timer=new Thread(){ public void run() { try { sleep(3000 );

}catch (InterruptedException e) {

e.printStackTrace(); } finally { Intent i=new Intent("android.example.hp.myquiz.Subjects");

i.putExtra("id",email); startActivity(i); }

} };timer.start();

} @Override protected void onPause() { super.onPause(); finish();} }

**AndroidManifest.xml**

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

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.quizkro">

<!--Internet permission added-->

<uses-permission android:name="android.permission.INTERNET" />

<application

android:allowBackup="true"

android:fullBackupContent="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity android:name=".questionsactivity" />

<activity android:name=".SetsActivity" />

<activity android:name=".categories\_activity" />

<activity android:name=".MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

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

</intent-filter>

</activity>

</application>

</manifest>