

# **Online Examination App**

## **(E-Test)**

Submitted in partial fulfillment of the requirements

For the degree of

**B.E. Information Technology**

By

**Lavia Suvarna                      64**

**Shweta Pai                              65**

**Malcolm D'souza                  66**

Supervisor:

**Alvina Alphonso**

Assistant Professor



Department of Information Technology

St. Francis Institute of Technology

(Engineering College)

University of Mumbai

2020-2021

## DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources.

We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in this submission.

We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1. -----  
(Signature)

-----  
(Name of student and Roll No.)

2. -----  
(Signature)

-----  
(Name of student and Roll No.)

3. -----  
(Signature)

-----  
(Name of student and Roll No.)

Date:17/12/2020

## **CERTIFICATE**

This Android Lab Mini-project E-Test by Lavia Suvarna, Shweta Pai and Malcolm D'Souza is complete in all respects and was successfully demonstrated on 17<sup>th</sup> December 2020.

Name : -----

Signature :-----

(Internal examiner)

Name : -----

Signature :-----

(External examiner)

Name : -----

Signature :-----

(Head of the Department)

Date: 17/12/2020.

Place: Mumbai.

# CONTENTS

## 1. Introduction

1. Project Description .....	1
2. Need of Application .....	1
3. Problem Statement .....	1
4. Proposed Solution .....	2
5. Scope of Application .....	2

## 2. Requirement Analysis

1. Target Audience .....	3
2. Features required .....	3
3. Hardware and Software requirements .....	3

## 3. Prototype .....

4

## 4. Design

1. Architectural Design .....	5
2. User Interface Design .....	6

## 5. Implementation

1. Working of Project .....	11
-----------------------------	----

## 6. Results and Discussion .....

51

## 7. Conclusion and Future Scope .....

56

## 8. References/Bibliography .....

58

# **Chapter 1**

## **Introduction**

### **1.1 *Project Description:***

E-Test is an online Examination platform which will be used by teachers and students in order to conduct objective type examinations efficiently when offline exams could not be conducted in a pandemic situation.

It aims to provide a personalized application where students get a dynamic test environment and teachers can upload the test as well as declare the result for the test and keep a record of all the test results.

### **1.2 *Need of Application***

To conduct online objective exams for students when conducting offline examinations is not possible.

### **1.3 *Problem Statement:***

In this rising pandemic situation, it is difficult to conduct physical examinations. There is a lot of confusion on how to test students on a regular basis and cope up with the physical examinations that took place when schools and colleges were functioning. Teachers face a lot of problems in conducting fair examinations through an online platform. It becomes difficult to coordinate the test questions and results in an efficient manner. Thus an online MCQ app will make online examinations easy and smooth.

#### **1.4 *Proposed Solution:***

The aim of this project is to develop an application that will be used to take online objective exams for students when giving an exam physically is not possible. This will also provide a platform where teachers can upload the questions and view the marks of students.

#### **1.5 *Scope of Application:***

- To conduct an online MCQ test.
- To provide timed tests for students to attempt.
- To keep a track of score in every test and provide results of these attempted tests.

## Chapter 2

### Requirement Analysis

#### **2.1 Target Audience:**

- Students
- Teachers

#### **2.2 Features Required:**

- User Login
- Authentication
- MCQ test for every Subject
- Timed Tests
- Download results
- View Score of every test
- Location API
- Camera Access

#### **2.3 Hardware and Software requirements:**

##### **Hardware Requirements:**

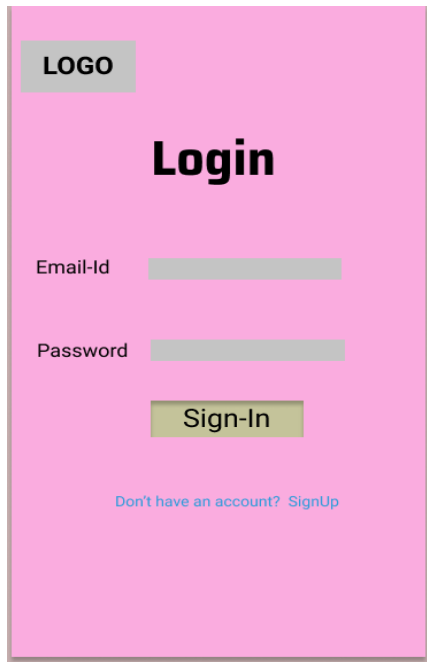
1.	RAM	4GB or above
2.	Intel processor	i5/i7
3.	CPU	2.5GHz
4.	Internet connection	1Mbps or more

##### **Software Requirements:**

1.	Adobe XD	11.0.23 version
2.	Operating System	Windows 7 or above
3.	Android Studio	Version 4.1
4.	Web browser(google chrome)	V6 and above

## Chapter 3

### Prototype



LOGO

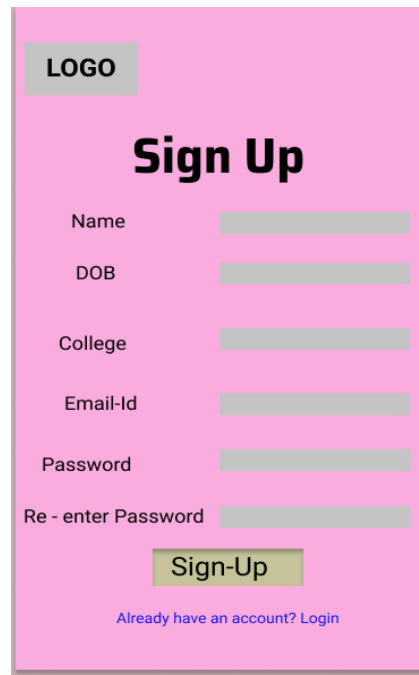
# Login

Email-Id

Password

[Don't have an account? SignUp](#)

Figure 1 Login Page



LOGO

# Sign Up

Name

DOB

College

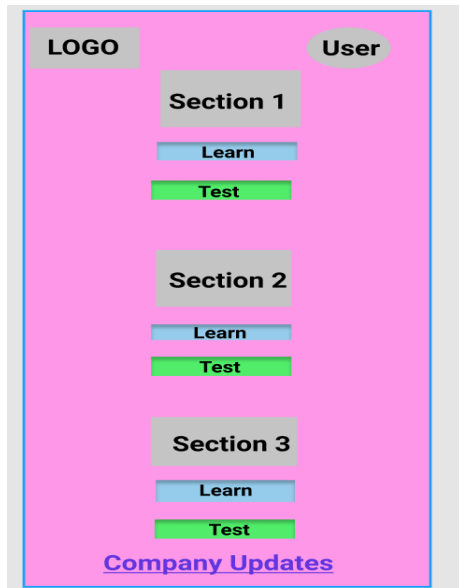
Email-Id

Password

Re - enter Password

[Already have an account? Login](#)

Figure 2 Signup Page



LOGO User

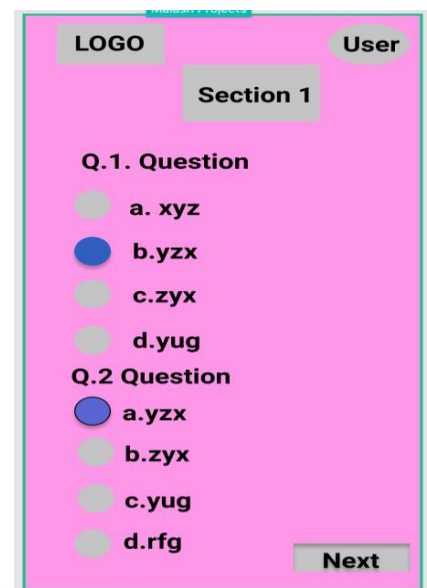
## Section 1

## Section 2

## Section 3

[Company Updates](#)

Figure 3 Home Page



LOGO User

## Section 1

**Q.1. Question**

☐ a. xyz

☒ b. yzx

☐ c. zyx

☐ d. yug

**Q.2 Question**

☒ a. yzx

☐ b. zyx

☐ c. yug

☐ d. rfg

Figure 4 Test



# Chapter 4

## Design

### 1. Architectural Design

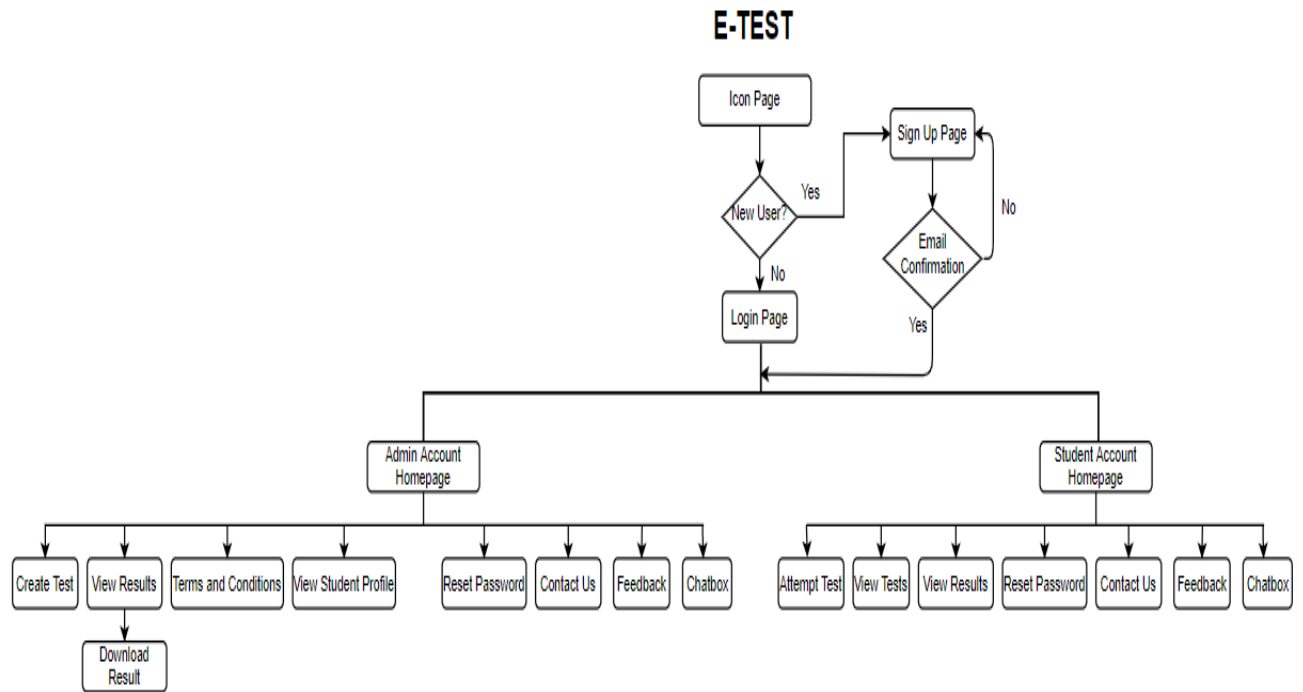


Figure 5 E-Test Architectural Diagram

## 2. User Interface Design

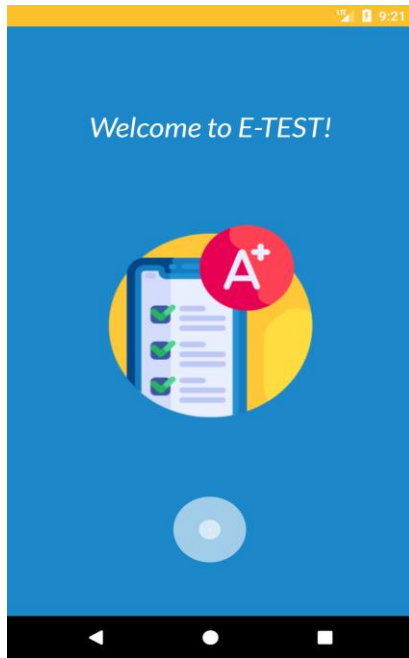


Figure 6 Splash Activity Page

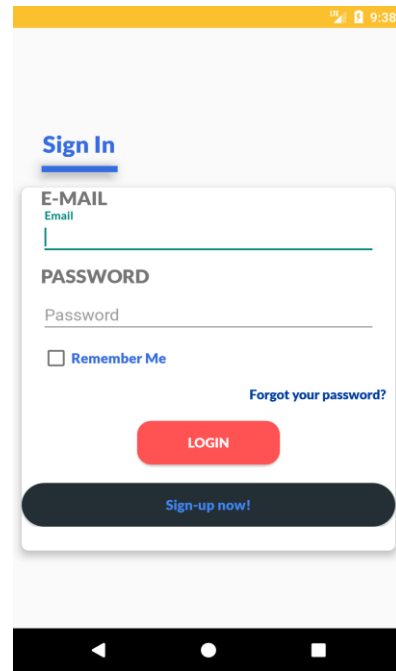


Figure 7 Sign-In Page

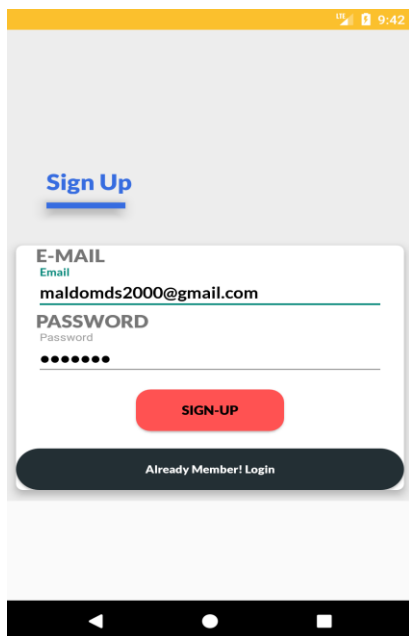


Figure 8 Sign Up page

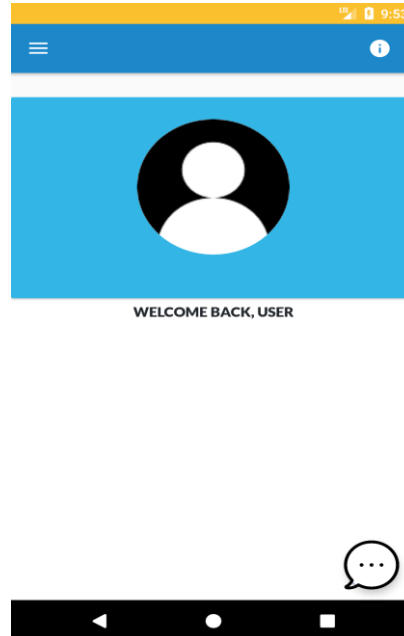


Figure 9 Student Home page on login

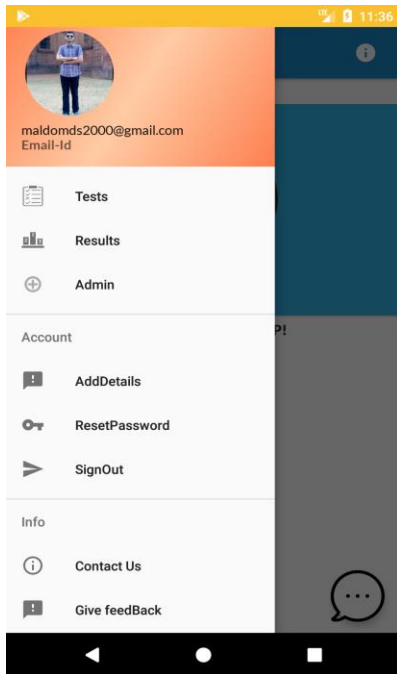


Figure 10 Drawer with various tabs

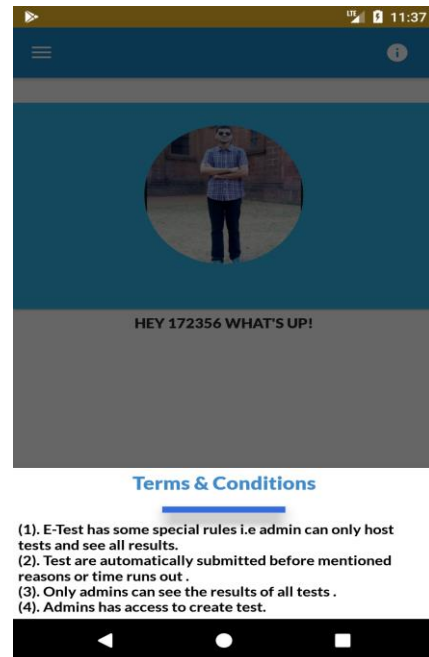


Figure 11 Terms and conditions popup

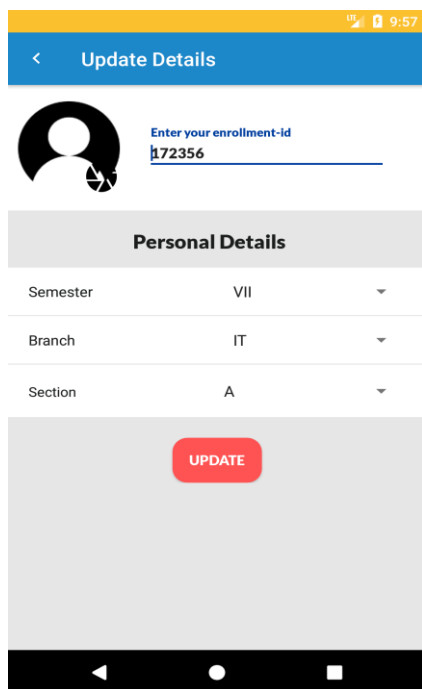


Figure 12 Update details page

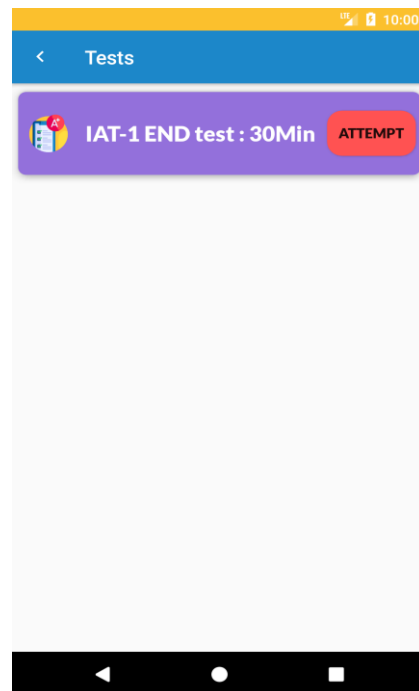


Figure 13 Tests page

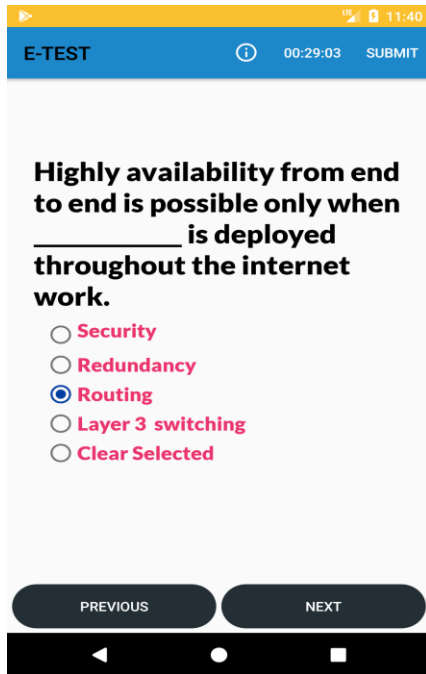


Figure 14 Questions of the test

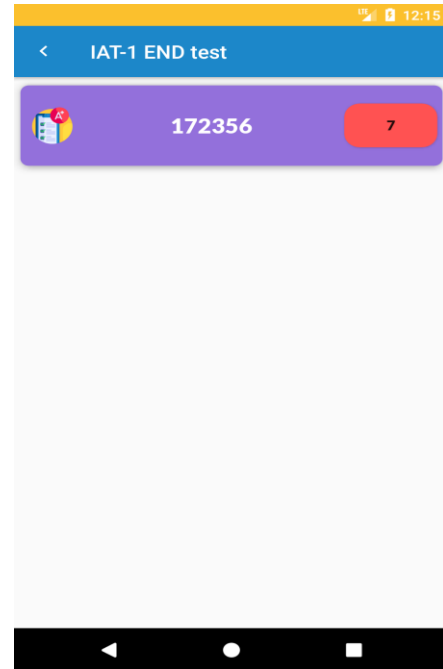


Figure 15 Results leaderboard

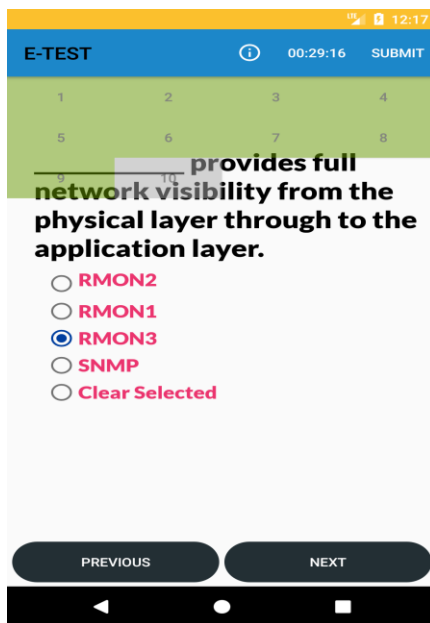


Figure 16 List of Attempted questions.

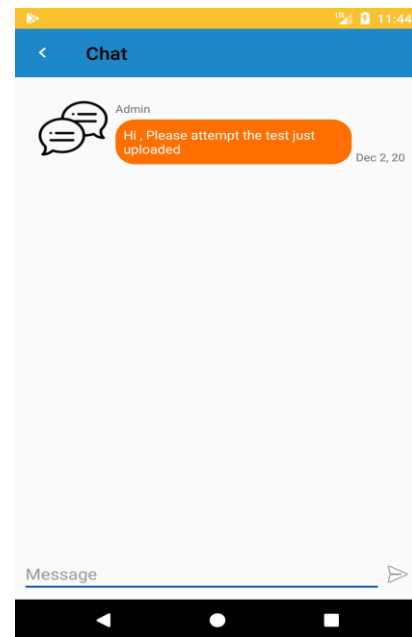


Figure 17 Chat page

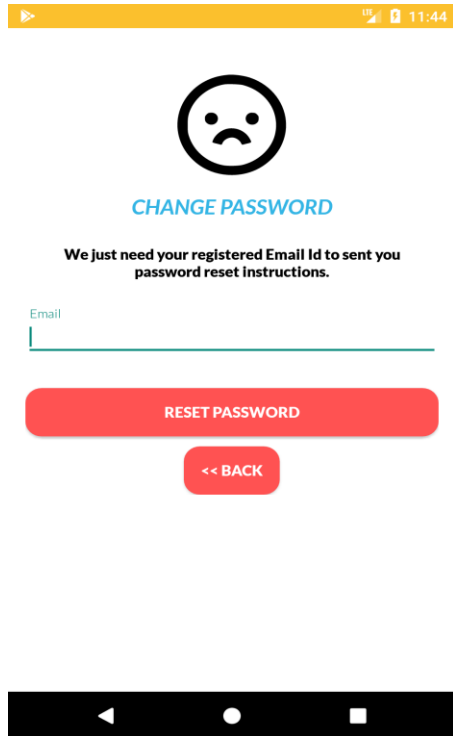


Figure 18 Change Password page

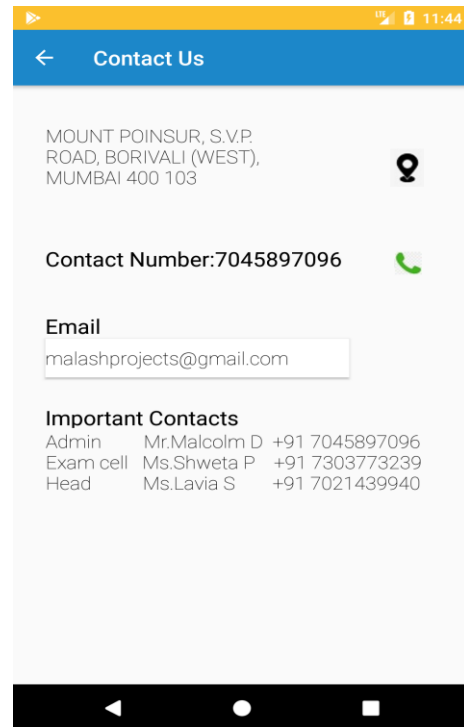


Figure 19 Contact us page

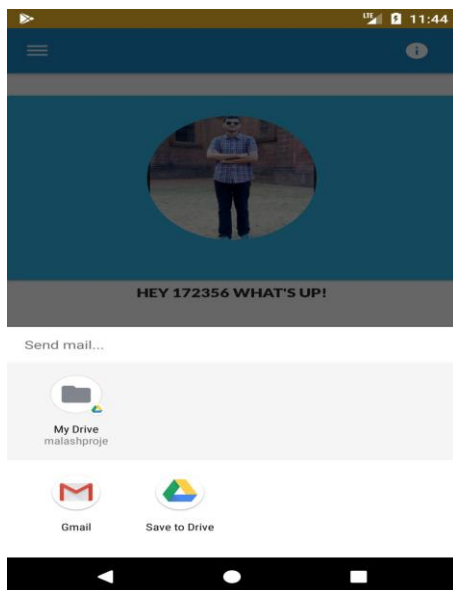


Figure 20 Send feedback options

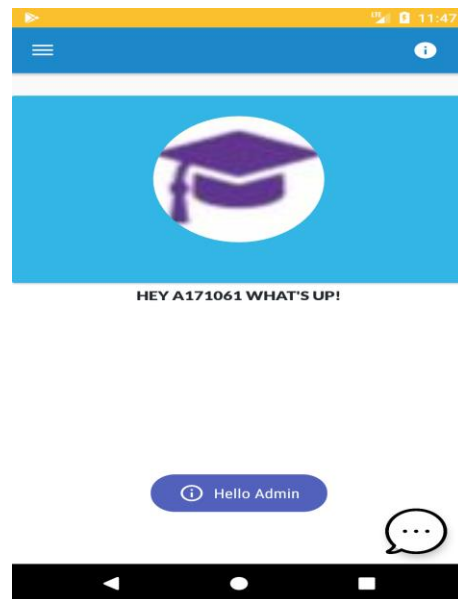


Figure 21 Admin Home page

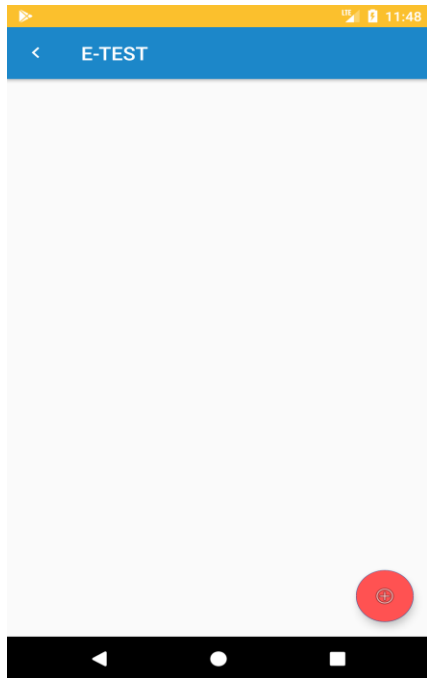


Figure 22 Create test option

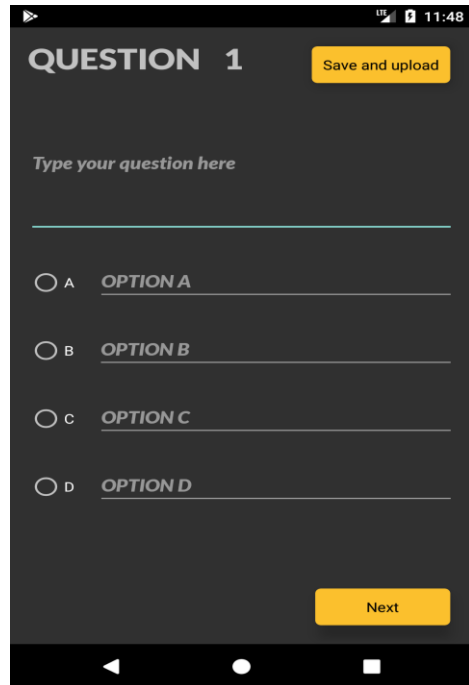


Figure 23 Test questions upload page

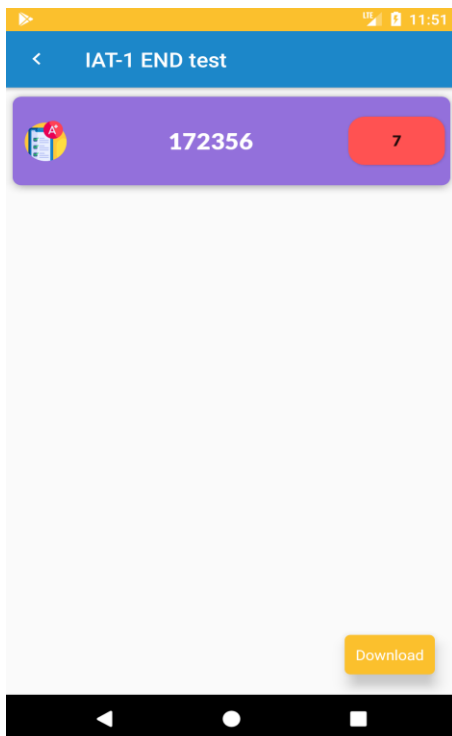


Figure 24 Admin result page with download option

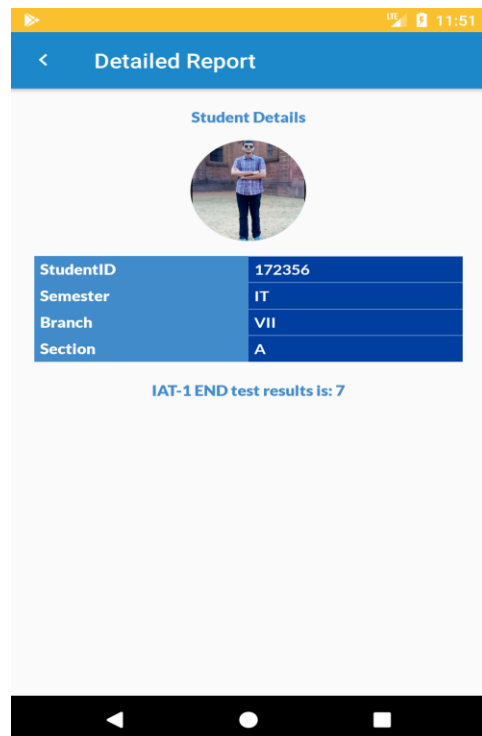


Figure 25 Student profile viewed by admin

## Chapter 5

### Implementation

#### 1. Working of Project

Our Project consist of an android application which acts a user interface to the student and admins/teachers and a firebase database which stores data, authenticates users and fetches data on our app at various places. Our app has two parts, one for students/users and the other part is admin part which can be used by teachers. To locate our app on the phone the user has to look for our E-Test app logo and click on it. On clicking the splash activity page get's loaded with our app icon and welcomes the user to our application. Then appears sign-in module of our app, if the user does not have a account on our app he need to click on the bottom sign-up button. On clicking in the sign-up button the sign-up page is displayed, the user needs to enter his email-id and password of his choice but with a minimum length of 6 characters. An verification email is sent to the email id used during registration, the user needs to click on the link sent by the mail and on clicking the mail, registration successful message is displayed and the data gets stored in the firebase database. Now the user can use his credentials and sign-in to the system. On successful sign-in the user is taken to the home page of our app.

As a prerequisite the user needs to enter his pid/enrollment number in the add details tab also he needs to update his semester, class and section, if the user wishes to he can use put a profile picture which can be clicked from camera or can be uploaded from the gallery, this image gets stored in the firebase and it appears on the home page with our pid and on the navigation drawer.

Both admin and students can chat with each other using the chatbot, admin can pass on a message to students also students facing issues can reach out to the admin. We also have a terms and condition tab where the terms and conditions of our app are mentioned.

Just a previously mentioned our app has two parts admin and student:

- **Student part of our app:**

After logging in and updating details the student can navigate to the test tab and attempt the test, which has timer of certain time and can also view the number of visited and unvisited questions.

The Student can also view their results from the result module immediately after the completion of the test also he can check the marks of his previous examination.

Students can also call the admin incase of major issues or mail the admin from the contact us page and also get the location from home to the institute.

Students can also change their password if they want to change or in case if they forget as password reset link is sent on registered email-id.

- **Admin part of our app:**

Admin has the most access rights, he can upload the test, save questions and he can also view the results and the profiles of all the candidates and download the result into a excel sheet just after the test is complete, this makes the evaluation of marks easy with also marks entered. Also the result will be displayed immediately after the test.

## Splash Activity module

### 1.SplashActivity.java

```
package com.example.test_quiz.Splash_Activity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.TextView;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;

import com.example.test_quiz.R;
import com.example.test_quiz.Auth_Controller.LoginActivity;
import com.wang.avi.AVLoadingIndicatorView;

public class SplashActivity extends AppCompatActivity {

    private AVLoadingIndicatorView avLoadingIndicatorView;

    /** Called when the activity is first created. */
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        avLoadingIndicatorView = findViewById(R.id.avi);
        avLoadingIndicatorView.smoothToShow();
        Animation anim = AnimationUtils.loadAnimation(this,R.anim.alpha);
        anim.reset();
        ConstraintLayout l = findViewById(R.id.lin_lay);
        l.clearAnimation();
        anim=AnimationUtils.loadAnimation(this,R.anim.translate);
        anim.reset();
        TextView tv= findViewById(R.id.textView2);
        tv.clearAnimation();
        tv.startAnimation(anim);
        //fillableLoaders.setProgress(40,1000);
        /* New Handler to start the Menu-Activity
         * and close this Splash-Screen after some seconds.*/

        /** Duration of wait */
        int SPLASH_DISPLAY_LENGTH = 2090;
        new Handler().postDelayed(new Runnable(){
            @Override
            public void run() {
                /* Create an Intent that will start the Menu-Activity. */
                Intent mainIntent = new Intent(SplashActivity.this, LoginActivity.class);
                mainIntent.setFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
                SplashActivity.this.startActivity(mainIntent);
                avLoadingIndicatorView.smoothToHide();
            }
        }, SPLASH_DISPLAY_LENGTH);
    }
}
```



```

        SplashActivity.this.finish();
        overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
    }
}, SPLASH_DISPLAY_LENGTH);
}
}

```

## 2.Splash\_create\_quiz.java

```

package com.example.test_quiz.Splash_Activity;

import android.content.Intent;

import com.daimajia.androidanimations.library.Techniques;
import com.example.test_quiz.Create_Quiz.Custom_quiz;
import com.example.test_quiz.R;
import com.viksaa.sssplash.lib.activity.AwesomeSplash;
import com.viksaa.sssplash.lib.cnst.Flags;
import com.viksaa.sssplash.lib.model.ConfigSplash;

public class Splash_create_quiz extends AwesomeSplash {

    private boolean isAdmin = false;
    @Override
    public void initSplash(ConfigSplash configSplash) {

        isAdmin = getIntent().getBooleanExtra("ChatAdmin",false);
        configSplash.setBackgroundColor(R.color.btn_logout_bg); //any color you want form colors.xml
        configSplash.setAnimCircularRevealDuration(500); //int ms
        configSplash.setRevealFlagX(Flags.REVEAL_RIGHT); //or Flags.REVEAL_LEFT
        configSplash.setRevealFlagY(Flags.REVEAL_BOTTOM); //or Flags.REVEAL_TOP

        //Choose LOGO OR PATH; if you don't provide String value for path it's logo by default

        //Customize Logo
        configSplash.setLogoSplash(0); //or any other drawable
        configSplash.setAnimLogoSplashDuration(0); //int ms
        configSplash.setAnimLogoSplashTechnique(Techniques.Bounce); //choose one form Techniques (ref:
https://github.com/daimajia/AndroidViewAnimations)

        //Customize Path
        // configSplash.setPathSplash(Constants.DROID_LOGO); //set path String
        configSplash.setOriginalHeight(400); //in relation to your svg (path) resource
        configSplash.setOriginalWidth(400); //in relation to your svg (path) resource
        configSplash.setAnimPathStrokeDrawingDuration(0);
        configSplash.setPathSplashStrokeSize(3); //I advise value be <5
        configSplash.setPathSplashStrokeColor(R.color.btn_logout_bg); //any color you want form colors.xml
        configSplash.setAnimPathFillingDuration(50);
        configSplash.setPathSplashFillColor(R.color.strokeColor); //path object filling color

        //Customize Title
        configSplash.setTitleSplash("");
    }
}

```

```

        configSplash.setTitleTextColor(R.color.btn_logout_bg);
        configSplash.setTitleTextSize(30f); //float value
        configSplash.setAnimTitleDuration(0);
        configSplash.setAnimTitleTechnique(Techniques.FlipInX);
//        configSplash.setTitleFont("fonts/ttf"); //provide string to your font located in assets/fonts/
    }

    @Override
    public void animationsFinished() {

        startActivity(new Intent(Splash_create_quiz.this, Custom_quiz.class));
        finish();

    }
}

```

## Authentication control Module.

### 1.LoginActivity.java

```

package com.example.test_quiz.Auth_Controller;

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

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

import es.dmoral.toasty.Toasty;

public class LoginActivity extends AppCompatActivity {

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

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);

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

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

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

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

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

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

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

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

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

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

```

```

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

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

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

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

//authenticate user
auth.signInWithEmailAndPassword(email, password)
    .addOnCompleteListener(LoginActivity.this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            // If sign in fails, display a message to the user. If sign in succeeds
            // the auth state listener will be notified and logic to handle the
            // signed in user can be handled in the listener.
            progressBar.setVisibility(View.GONE);
            avLoadingIndicatorView.setVisibility(View.GONE);
            avLoadingIndicatorView.smoothToHide();
            if (!task.isSuccessful()) {
                // there was an error
                if (password.length() < 6) {
                    inputPassword.setError(getString(R.string.minimum_password));
                } else {
                    Toasty.warning(LoginActivity.this, getString(R.string.auth_failed),
Toasty.LENGTH_LONG).show();
                }
            } else {
                if (auth.getCurrentUser().isEmailVerified()) {
                    startActivity(new Intent(LoginActivity.this, MainActivity.class));
                    overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
                    finish();
                } else {
                    Toasty.error(LoginActivity.this, R.string.email_unverified,
Toasty.LENGTH_SHORT).show();
                    FirebaseAuth.getInstance().signOut();
                }
            }
        }
    });
}
}
}
}

```

## 2.ResetPasswordActivity.java

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

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

import es.dmoral.toasty.Toasty;

public class ResetPasswordActivity extends AppCompatActivity {

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

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

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

        auth = FirebaseAuth.getInstance();

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

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

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

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

```

        avLoadingIndicatorView.setVisibility(View.VISIBLE);
        avLoadingIndicatorView.show();
        auth.sendPasswordResetEmail(email)
            .addOnCompleteListener(new OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {
                    if (task.isSuccessful()) {
                        Toasty.success(ResetPasswordActivity.this, "We have sent you instructions to reset your
password!",
                                Toasty.LENGTH_SHORT).show();
                    } else {
                        Toasty.success(ResetPasswordActivity.this, "Failed to send reset email!",
                                Toasty.LENGTH_SHORT).show();
                    }

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

```

### 3.SignupActivity.java

```

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

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

import java.util.Objects;

import es.dmoral.toasty.Toasty;

public class SignUp extends AppCompatActivity {

    private EditText inputEmail, inputPassword;
    private Button btnSignIn, btnSignUp, btnResetPassword;
    private AVLoadingIndicatorView avLoadingIndicatorView;

```

```

private FirebaseAuth auth;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_signup);

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

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

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

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

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

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

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

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

            if (password.length() < 6) {
                Toast.makeText(getApplicationContext(), "Password too short, enter minimum 6 characters!",
                Toast.LENGTH_SHORT).show();
                return;
            }

```

```

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

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

```

## Attempt Test Module

### 1. AttemptTestActivity.java

```

package com.example.test_quiz.Attempt_Quiz_Section;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Point;
import android.os.Bundle;
import android.os.CountDownTimer;
import androidx.annotation.ColorInt;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

```



```

import androidx.appcompat.widget.Toolbar;
import androidx.recyclerview.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.TranslateAnimation;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.GridView;
import android.widget.LinearLayout;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;

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

import org.jetbrains.annotations.NotNull;

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

import es.dmoral.toasty.Toasty;

public class AttemptTest extends AppCompatActivity {
    ArrayList<Question> questions;
    String []answers;
    Toolbar toolbar;
    DiscreteScrollView scrollView;
    LinearLayout indexLayout;
    GridView quesGrid;
    ArrayList<String> list;
    ArrayList<String> arrayList;
    int flag_controller = 1;
    long timer;//=((Test) getIntent().getExtras().get("Questions")).getTime()*60*1000;
    popGridAdapter popGrid;
    Button next,prev;
    TextView textView;
    private DatabaseReference mDatabase;
    private FirebaseAuth auth;
    private String TESTNAME;
    private RadioGroup group;
    private int countPaused = 0;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
mDatabase = FirebaseDatabase.getInstance().getReference();
auth= FirebaseAuth.getInstance();
setContentView(R.layout.activity_attempt);
questions=((Test) getIntent().getExtras().get("Questions")).getQuestions();
TESTNAME = (String) getIntent().getExtras().get("TESTNAME");
toolbar=findViewById(R.id.toolbar);
toolbar.setTitleTextColor(getResources().getColor(android.R.color.black));
answers=new String[questions.size()];
setSupportActionBar(toolbar);
toolbar.setTitleTextColor(getResources().getColor(android.R.color.black));
scrollView = findViewById(R.id.discrete);
final QuestionAdapter questionAdapter=new QuestionAdapter(questions);
scrollView.setAdapter(questionAdapter);

next=findViewById(R.id.next);
next.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(scrollView.getCurrentItem()==questions.size()-1){
            showPopUp();
        }else {
            //setNextPrevButton(scrollView.getCurrentItem() + 1);
            scrollView.smoothScrollToPosition(scrollView.getCurrentItem() + 1);
        }
    }
});

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

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

```

```

        setNextPrevButton(newPosition);
    }
});

timer=((Test) getIntent().getExtras().get("Questions")).getTime()*60*1000;
}
void showPopUp(){
    AlertDialog.Builder builder=new AlertDialog.Builder(AttemptTest.this);
    builder.setMessage("Do you want to submit?");
    builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            submit();
            // setAlertDialog(answerText);
            dialogStart();
        }
    });

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

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

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

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

void dialogStart() {

```

```

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

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

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

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

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

```

```

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

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

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

@Override
public void onBackPressed() {
    showPopUp();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {

    super.onCreateOptionsMenu(menu);

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

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

```

```

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

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

return true;
}

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

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

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

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

    private int itemHeight;
    private ArrayList<Question> data;

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

    @Override
    public void onAttachedToRecyclerView(RecyclerView recyclerView) {
        super.onAttachedToRecyclerView(recyclerView);
        Activity context = (Activity) recyclerView.getContext();
        Point windowDimensions = new Point();

```

```

context.getWindowManager().getDefaultDisplay().getSize(windowDimensions);
itemHeight = Math.round(windowDimensions.y * 0.6f);
}

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

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

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

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

```

```

        holder.radioGroup.check(R.id.radioButton4);
    }
}

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

class ViewHolder extends RecyclerView.ViewHolder {

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

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

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

    public void unCheck() {

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

class popGridAdapter extends BaseAdapter{
    Context mContext;
    popGridAdapter(Context context){

```



```

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

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

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

    @Override
    public View getView(final int i, View view, ViewGroup viewGroup) {
        View convertView;
        if(view==null){
            convertView=new Button(mContext);
        }else convertView=view;
        if(answers[i]==null)
            (convertView).setBackgroundColor(getResources().getColor(android.R.color.darker_gray));
        else
            (convertView).setBackgroundColor(getResources().getColor(android.R.color.holo_green_dark));

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

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

public void slideUp(View view){
    TranslateAnimation animate = new TranslateAnimation(
        0,          // fromXDelta
        0,          // toXDelta
        0, // fromYDelta
        -view.getHeight(), // toYDelta
        animate.setDuration(500);
        view.startAnimation(animate);
        view.setVisibility(View.GONE);
    }

    // slide the view from its current position to below itself
    public void slideDown(View view){
        view.setVisibility(View.VISIBLE);
        TranslateAnimation animate = new TranslateAnimation(

```

```

        0,          // fromXDelta
        0,          // toXDelta
        -view.getHeight(),      // fromYDelta
        0); // toYDelta
    animate.setDuration(500);
    view.startAnimation(animate);

}

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

```

## 2.Tests.java

```

package com.example.test_quiz.Attempt_Quiz_Section;

import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.content.ContextCompat;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.Filterable;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import com.example.test_quiz.NotificationActivity.NotificationService;
import com.example.test_quiz.R;
import com.example.test_quiz.Model.Question;
import com.example.test_quiz.Model.Test;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.wang.avi.AVLoadingIndicatorView;

import java.util.ArrayList;
import java.util.Objects;

```

```

public class Tests extends AppCompatActivity {
    private FirebaseDatabase database;
    private DatabaseReference myRef;
    private AVLoadingIndicatorView avLoadingIndicatorView;
    private ListView listView;
    private TestAdapter testAdapter;
    private int lastPos = -1;

    ArrayList<Test> tests=new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tests);
        Toolbar toolbar = findViewById(R.id.toolbar);
        toolbar.setTitleTextColor(getResources().getColor(android.R.color.white));
        setSupportActionBar(toolbar);
        avLoadingIndicatorView = findViewById(R.id.loader1);
        avLoadingIndicatorView.setVisibility(View.VISIBLE);
        avLoadingIndicatorView.show();
        setSupportActionBar(toolbar);
        Objects.requireNonNull(getSupportActionBar())
            .setDisplayHomeAsUpEnabled(true);
        database= FirebaseDatabase.getInstance();
        myRef=database.getReference();
        listView=findViewById(R.id.test_listview);
        testAdapter=new TestAdapter(Tests.this,tests);
        listView.setAdapter(testAdapter);
        getQues();
    }

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

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

        int id = item.getItemId();

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

    public void getQues(){
        //addListenerForSingleValueEvent
        myRef.child("tests").addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                tests.clear();
            }
        });
    }
}

```

```

        for(DataSnapshot snapshot:dataSnapshot.getChildren()){
            Test t=new Test();
            t.setName(snapshot.getKey());
            t.setTime(Long.parseLong(snapshot.child("Time").getValue().toString()));
            ArrayList<Question> ques=new ArrayList<>();
            for (DataSnapshot qSnap:snapshot.child("Questions").getChildren()){
                ques.add(qSnap.getValue(Question.class));
            }
            t.setQuestions(ques);
            tests.add(t);
        }
        testAdapter.dataList=tests;
        testAdapter.notifyDataSetChanged();
        avLoadingIndicatorView.setVisibility(View.GONE);
        avLoadingIndicatorView.hide();
        Log.e("The read success: ", "su"+tests.size());
    }

    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {
        avLoadingIndicatorView.setVisibility(View.GONE);
        avLoadingIndicatorView.hide();
        Log.e("The read failed: ", databaseError.getMessage());
    }
});
}

class TestAdapter extends ArrayAdapter<Test> implements Filterable {
    private Context mContext;
    ArrayList<Test> dataList;
    public TestAdapter( Context context,ArrayList<Test> list) {
        super(context, 0 , list);
        mContext = context;
        dataList = list;
    }

    @SuppressWarnings("SetTextI18n")
    @NonNull
    @Override
    public View getView(final int position, @Nullable View convertView, @NonNull ViewGroup parent) {
        View listItem = convertView;
        if(listItem == null)
            listItem = LayoutInflater.from(mContext).inflate(R.layout.test_item,parent,false);

        ((ImageView)listItem.findViewById(R.id.item_imageView)).
            setImageDrawable(ContextCompat.getDrawable(mContext,R.drawable.logo));

        ((TextView)listItem.findViewById(R.id.item_textView))
            .setText(dataList.get(position).getName()+" : "+dataList.get(position).getTime()+"Min");

        ((Button)listItem.findViewById(R.id.item_button)).setText("Attempt");

        (listItem.findViewById(R.id.item_button)).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

```

        Intent intent=new Intent(mContext, AttemptTest.class);
        intent.putExtra("Questions",dataList.get(position));
        intent.putExtra("TESTNAME",dataList.get(position).getName());
        startActivity(intent);
    }
});

    Animation animation = AnimationUtils.loadAnimation(mContext(),
        (position > lastPos) ? R.anim.up_from_bottom : R.anim.down_from_top);
    (listItem).startAnimation(animation);
    lastPos = position;

    return listItem;
}
}
}

```

## 2.Create\_quiz\_main.java

```

package com.example.test_quiz.Create_Quiz;

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

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

import java.util.Objects;

public class create_quiz_main extends AppCompatActivity {

    private FloatingActionButton fab1;
    private Toolbar toolbar;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.content_main_1);
        toolbar = findViewById(R.id.toolbartst);
        toolbar.setTitleTextColor(getResources().getColor(android.R.color.white));
        setSupportActionBar(toolbar);
        Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);
        fab1 = findViewById(R.id.fab);
        fab1.setOnClickListener(new View.OnClickListener() {
            @Override

```

```

        public void onClick(View v) {
            startActivity(new Intent(create_quiz_main.this, Splash_create_quiz.class));
            finish();
        }
    });
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {

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

```

## Result Module

### 1.GetDetailedReport.java

```

package com.example.test_quiz.Create_Quiz;

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

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

import java.util.Objects;

public class create_quiz_main extends AppCompatActivity {

    private FloatingActionButton fab1;
    private Toolbar toolbar;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.content_main_1);
        toolbar = findViewById(R.id.toolbartst);
        toolbar.setTitleTextColor(getResources().getColor(android.R.color.white));
        setSupportActionBar(toolbar);
        Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);
        fab1 = findViewById(R.id.fab);
        fab1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

```

```

        startActivity(new Intent(create_quiz_main.this, Splash_create_quiz.class));
        finish();
    }
});
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {

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

```

## 2.Resultsadmin.java

```

package com.example.test_quiz.Results_section;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.content.ContextCompat;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;

import com.example.test_quiz.R;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.wang.avi.AVLoadingIndicatorView;

import java.util.ArrayList;
import java.util.Objects;

```

```

public class ResultsAdmin extends AppCompatActivity {

    private FirebaseDatabase database;
    private FirebaseAuth auth;
    private DatabaseReference myRef;
    private AVLoadingIndicatorView avLoadingIndicatorView;
    private ListView listView;
    private ResultsAdmin.TestAdapter testAdapter;
    ArrayList<String> result=new ArrayList<>();
    private int lastPos = -1;
    private boolean isAdmin = false;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        auth= FirebaseAuth.getInstance();
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tests);
        isAdmin = getIntent().getBooleanExtra("ISADMIN",false);
        if(!isAdmin)
            setTitle("Results");
        Toolbar toolbar = findViewById(R.id.toolbar);
        toolbar.setTitleTextColor(getResources().getColor(android.R.color.white));
        setSupportActionBar(toolbar);
        Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);
        avLoadingIndicatorView = findViewById(R.id.loader1);
        avLoadingIndicatorView.setVisibility(View.VISIBLE);
        avLoadingIndicatorView.show();
        database= FirebaseDatabase.getInstance();
        myRef=database.getReference();
        listView=findViewById(R.id.test_listview);
        testAdapter=new ResultsAdmin.TestAdapter(ResultsAdmin.this,result);
        listView.setAdapter(testAdapter);
        getResults();
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

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

    public void getResults(){

        if(isAdmin) {

            myRef.child("Results").addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                    result.clear();
                    for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
                        result.add(snapshot.getKey());
                    }
                    testAdapter.dataList = result;
                }
            });
        }
    }
}

```



```

        testAdapter.notifyDataSetChanged();
        avLoadingIndicatorView.setVisibility(View.GONE);
        avLoadingIndicatorView.hide();
        Log.e("The read success: ", "su" + result.size());
    }

    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {
        avLoadingIndicatorView.setVisibility(View.GONE);
        avLoadingIndicatorView.hide();
        Log.e("The read failed: ", databaseError.getMessage());
    }
});
}
else {

myRef.child("Results").addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        result.clear();
        for(DataSnapshot snapshot:dataSnapshot.getChildren()){
            if(snapshot.hasChild(Objects
                .requireNonNull(auth.getUid())))
                result.add(snapshot.getKey());
        }
        testAdapter.dataList=result;
        testAdapter.notifyDataSetChanged();
        avLoadingIndicatorView.setVisibility(View.GONE);
        avLoadingIndicatorView.hide();
        Log.e("The read success: ", "su"+result.size());
    }

    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {
        avLoadingIndicatorView.setVisibility(View.GONE);
        avLoadingIndicatorView.hide();
        Log.e("The read failed: ",databaseError.getMessage());
    }
});
}
}

class TestAdapter extends ArrayAdapter<String> {

    private Context mContext;
    ArrayList<String> dataList;

    public TestAdapter( Context context,ArrayList<String> list) {
        super(context, 0 , list);
        mContext = context;
        dataList = list;
    }

    @NonNull
    @Override

```

```

public View getView(final int position, @Nullable View convertView, @NonNull ViewGroup parent) {

    View listItem = convertView;
    if(listItem == null)
        listItem = LayoutInflater.from(mContext).inflate(R.layout.test_item,parent,false);
    ((ImageView)listItem.findViewById(R.id.item_imageView))
        .setImageDrawable(ContextCompat.getDrawable(mContext,R.drawable.ranking));
    ((ImageView)listItem.findViewById(R.id.item_imageView)).setPadding(10,0,0,0);
    ((TextView)listItem.findViewById(R.id.item_textView)).setText(dataList.get(position));
    ((Button)listItem.findViewById(R.id.item_button)).setText("View");

    Animation animation = AnimationUtils.loadAnimation(getContext(),
        (position > lastPos) ? R.anim.up_from_bottom : R.anim.down_from_top);

    (listItem).startAnimation(animation);
    lastPos = position;
    ((Button)listItem.findViewById(R.id.item_button)).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent=new Intent(ResultsAdmin.this, ResultsAdminDetailed.class);
            intent.putExtra("test",dataList.get(position));
            intent.putExtra("ISAdmin",isAdmin);
            startActivity(intent);
        }
    });
    return listItem;
}
}
}

```

## Implementation Screenshots:

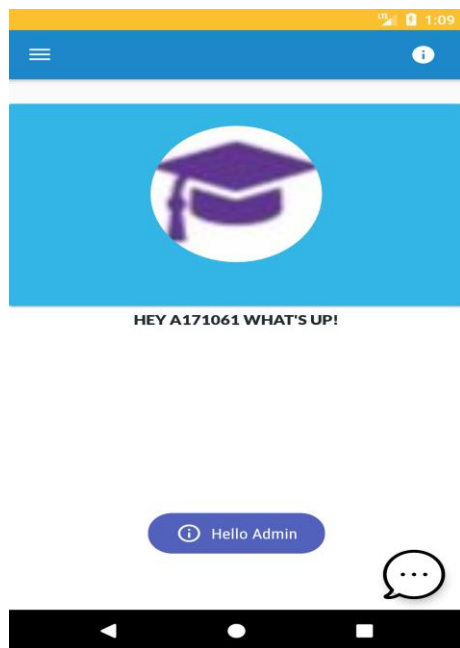


Figure 26 Successful Admin Login

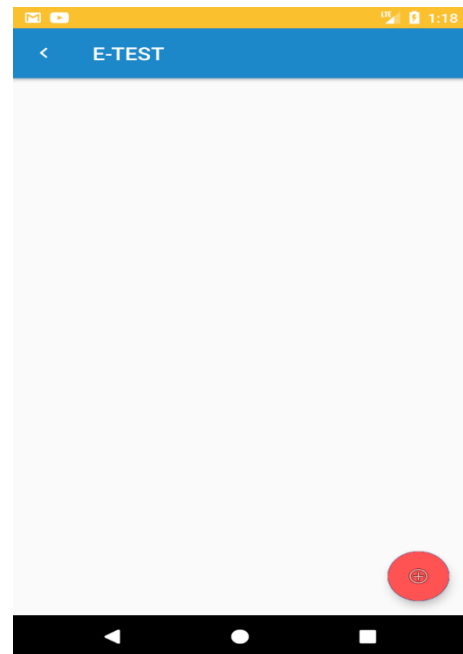


Figure 27 Create test page for admin

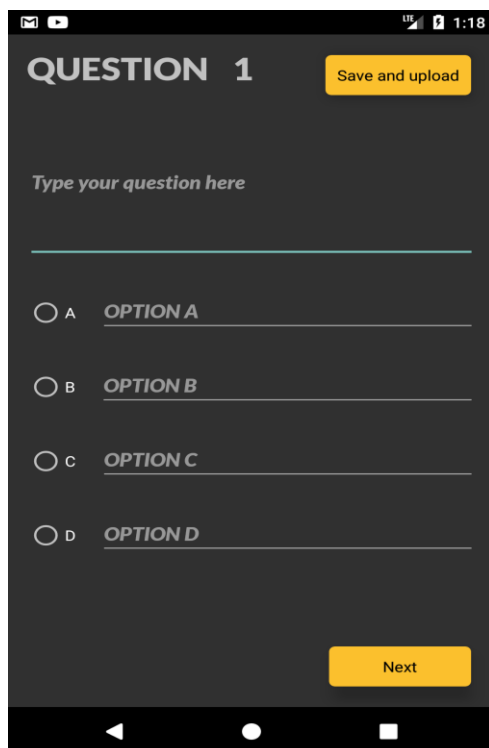


Figure 28 Enter the question page

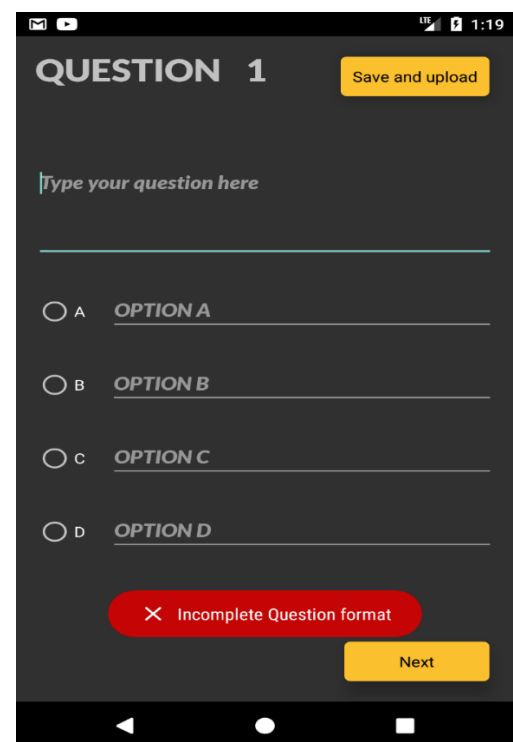


Figure 29 Toast on uploading without options

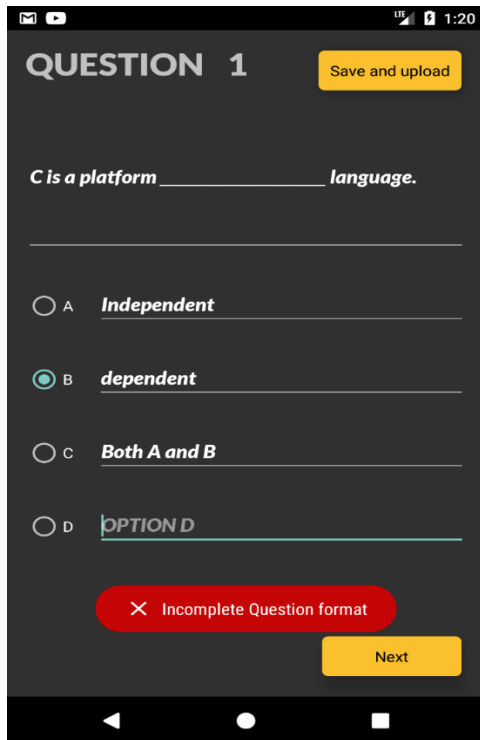


Figure 30 Toast on not filling one option

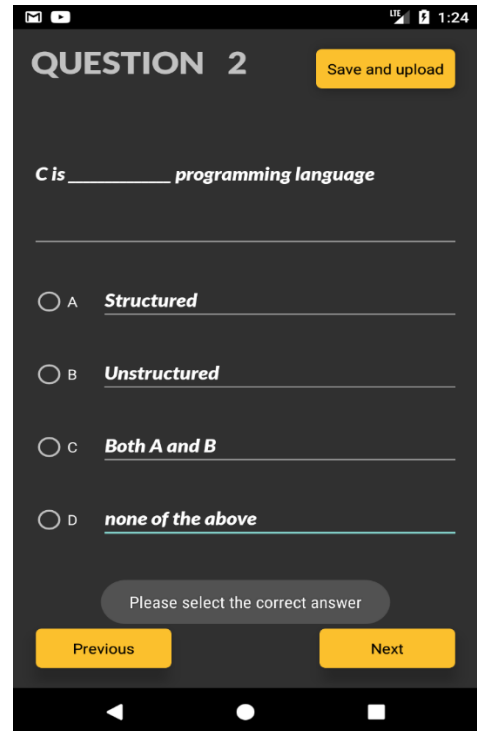


Figure 31 Toast when correct answer is not ticked

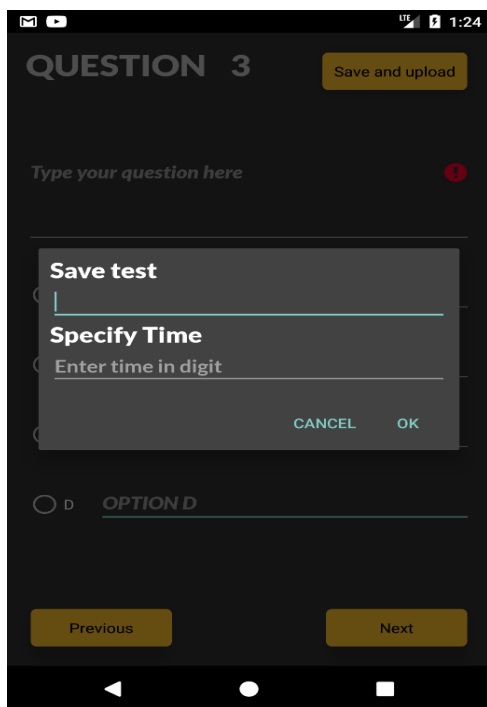


Figure 32 Pop up after clicking on save& upload.

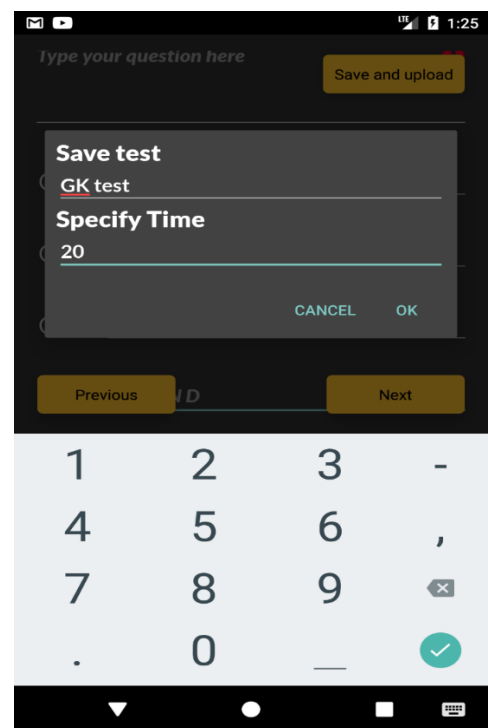


Figure 33 Entering the name and time for test

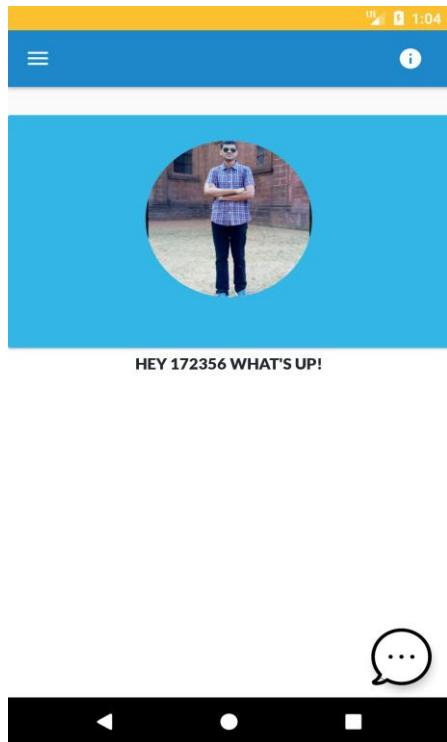


Figure 34 Successful Student Login

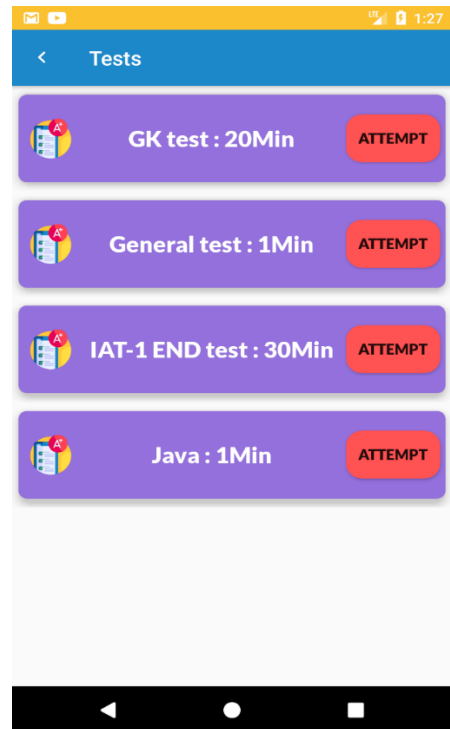


Figure 35 Students can view the test .

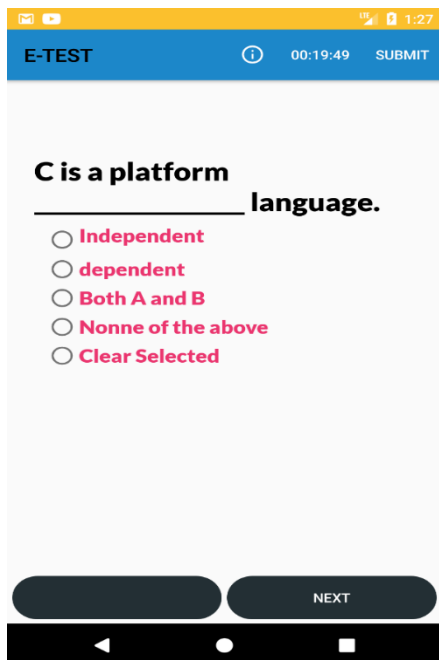


Figure 36 Test with timer and submit button.

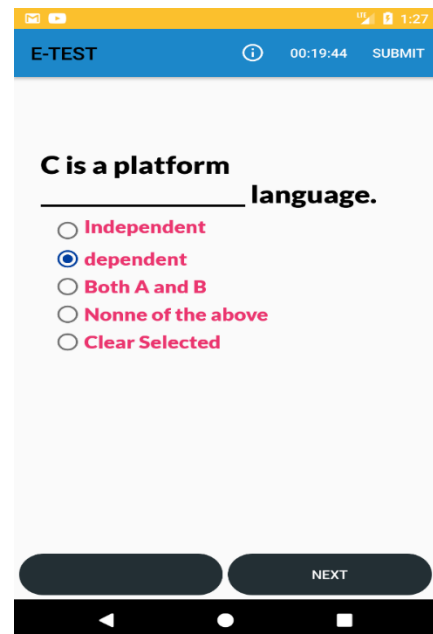


Figure 37 Students can select option

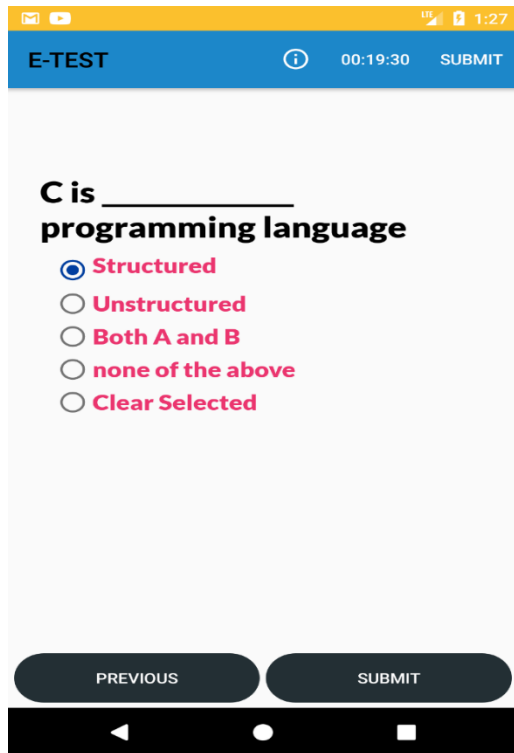


Figure 38 Previous button to navigate back.

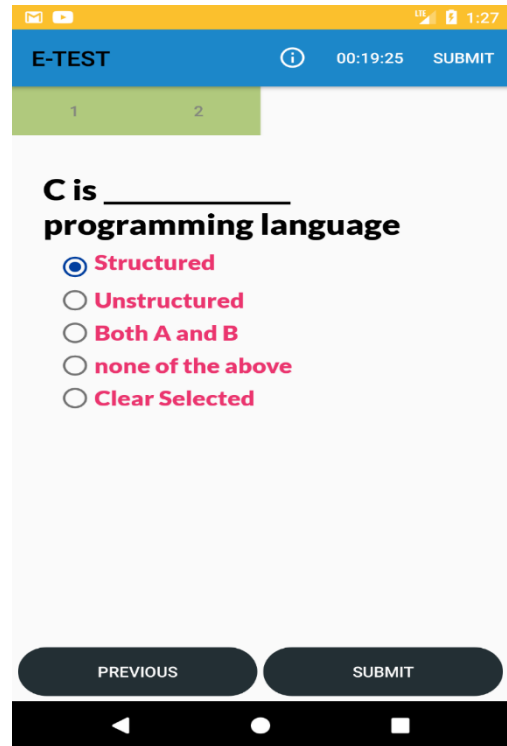


Figure 39 After clicking on information tab

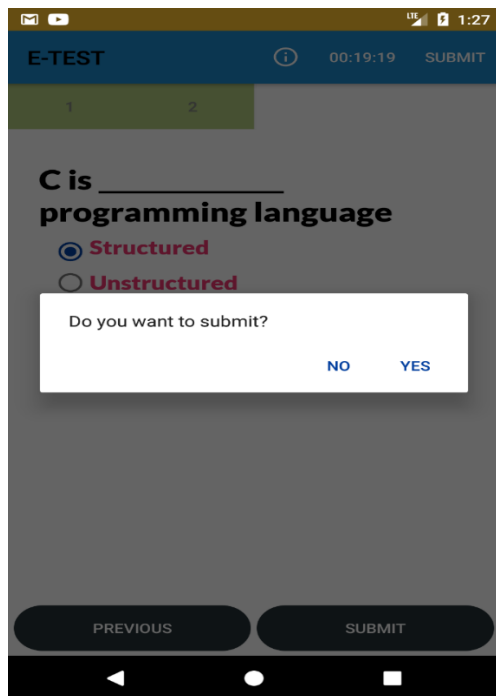


Figure 40 Submit the test popup.

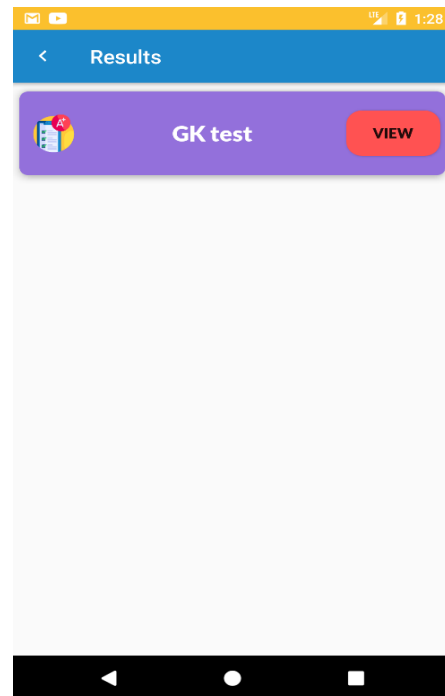


Figure 41 Student result tab after test

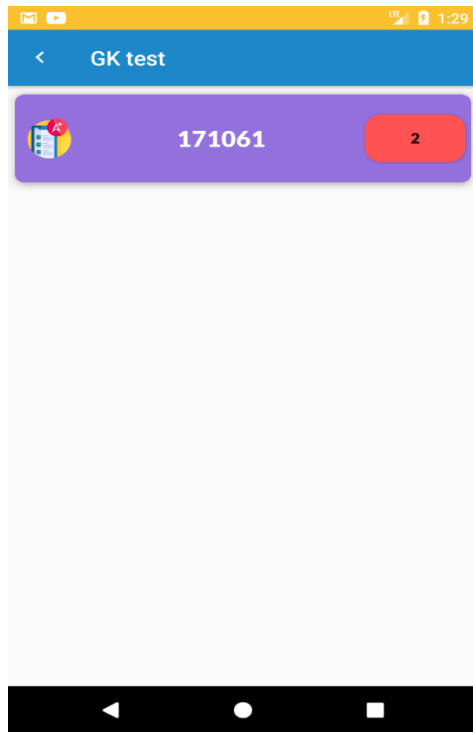


Figure 42 Test marks displayed on scoreboard

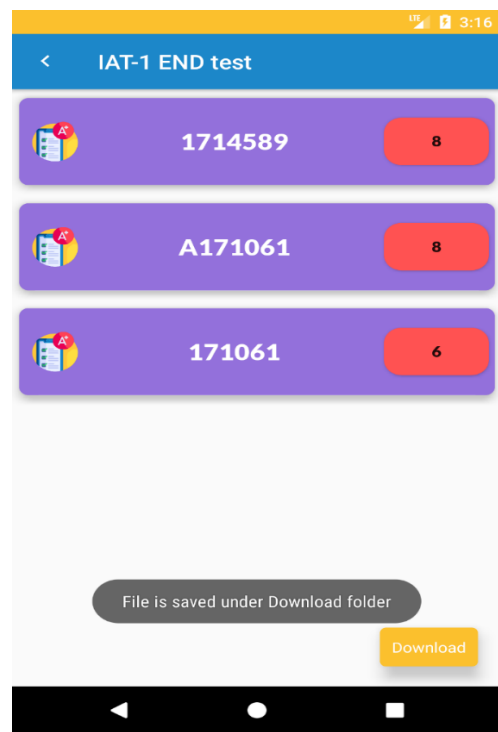


Figure 43 Admin can download the result.

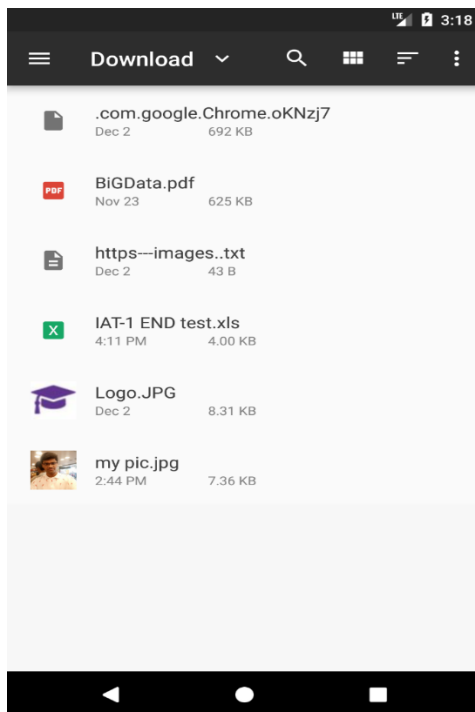


Figure 44 Excel sheet with the name of the test .

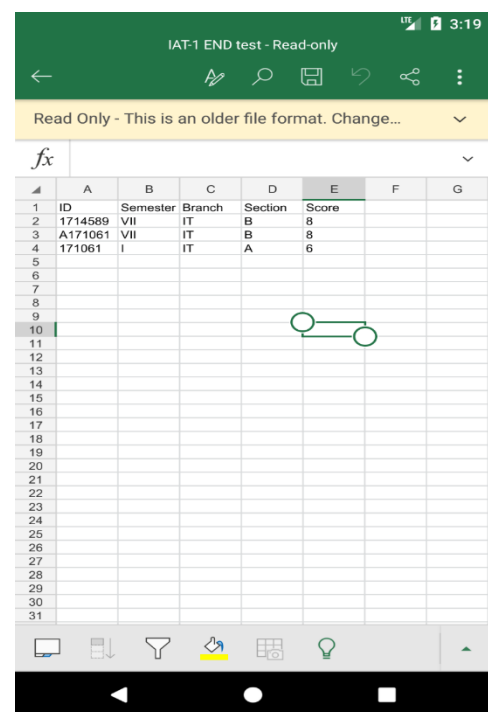


Figure 45 Excel sheet with student marks

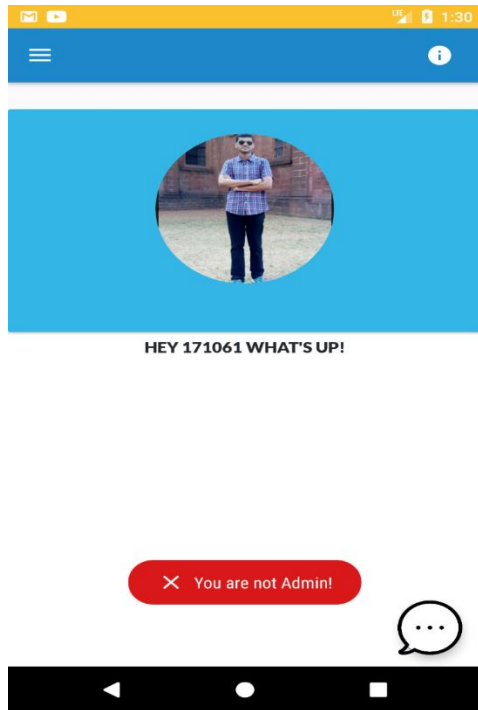


Figure 46 Toast after clicking on admin by student

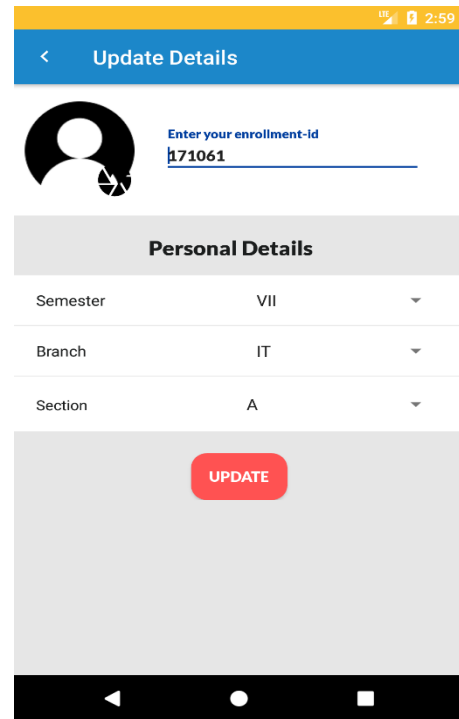


Figure 47 Update details page

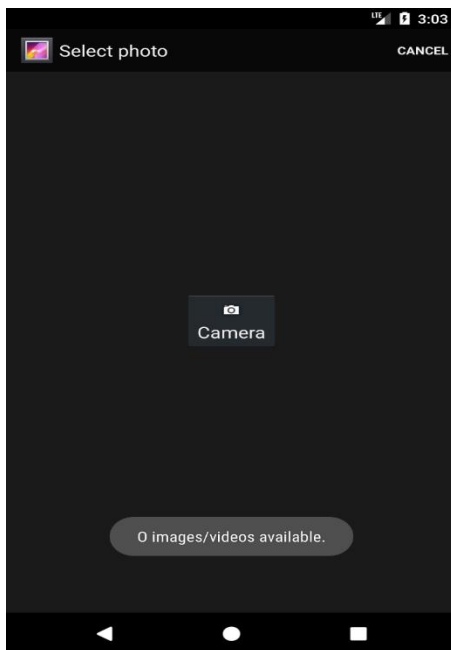


Figure 48 Camera/Gallery Page

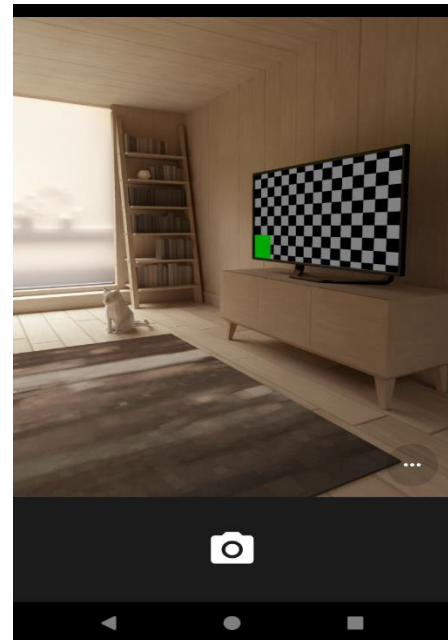


Figure 49 Clicking picture(Emulator)



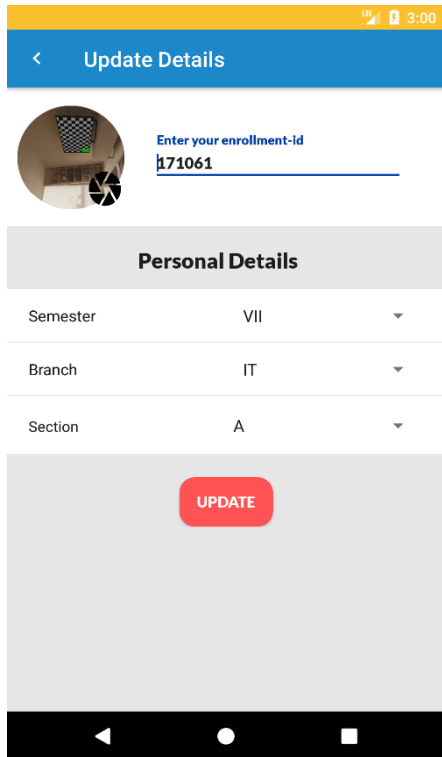


Figure 50 The captured image uploaded

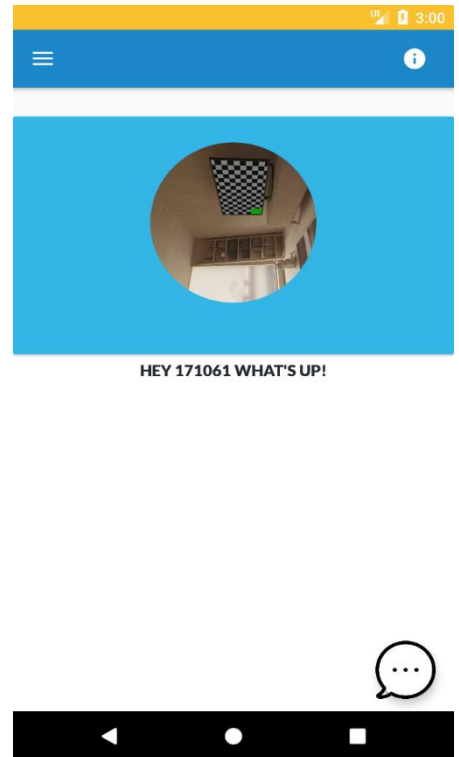


Figure 51 The capture image on home page

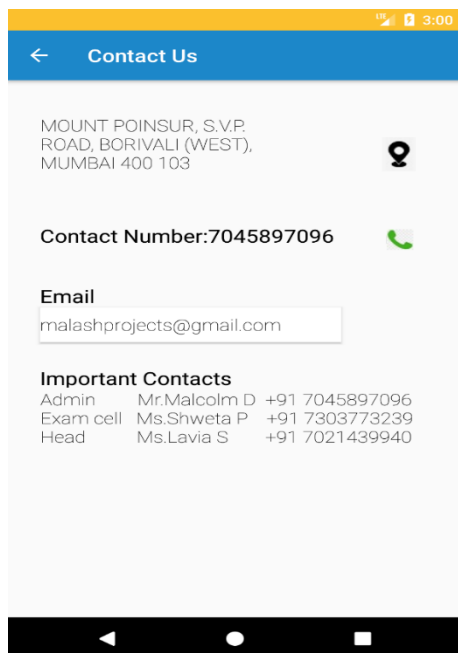


Figure 52 Contact us page

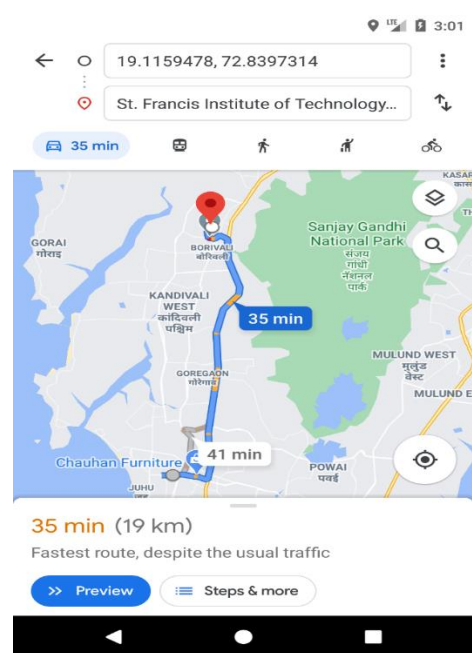


Figure 53 Location displayed

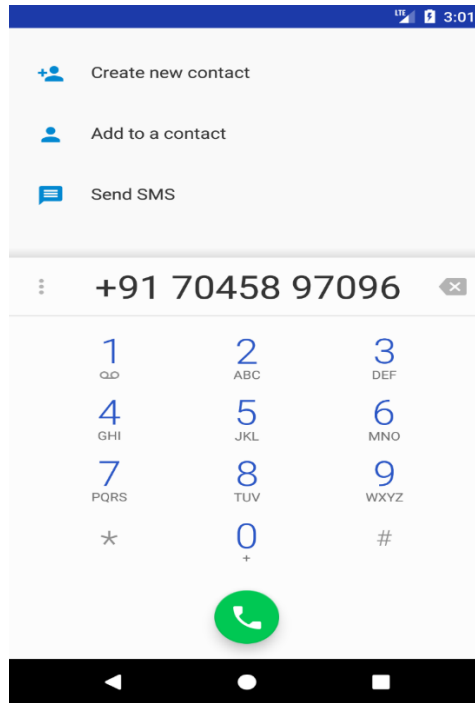


Figure 54 Calling tab with number pre-entered displayed as on the contact us page

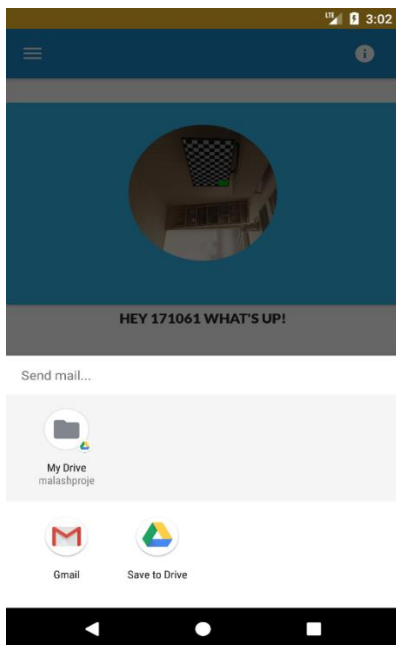


Figure 55 Feedback drawer

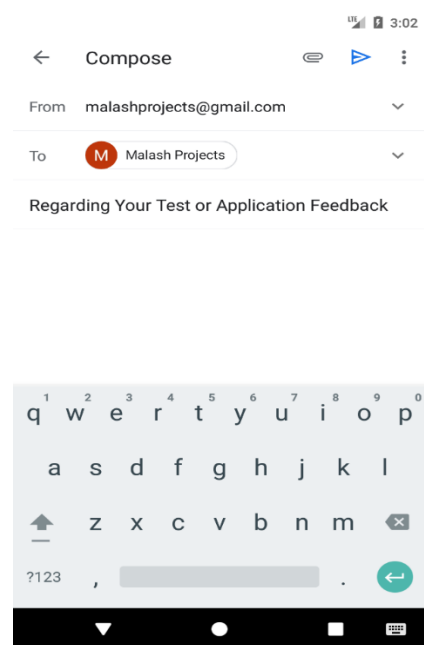


Figure 56 Email feedback

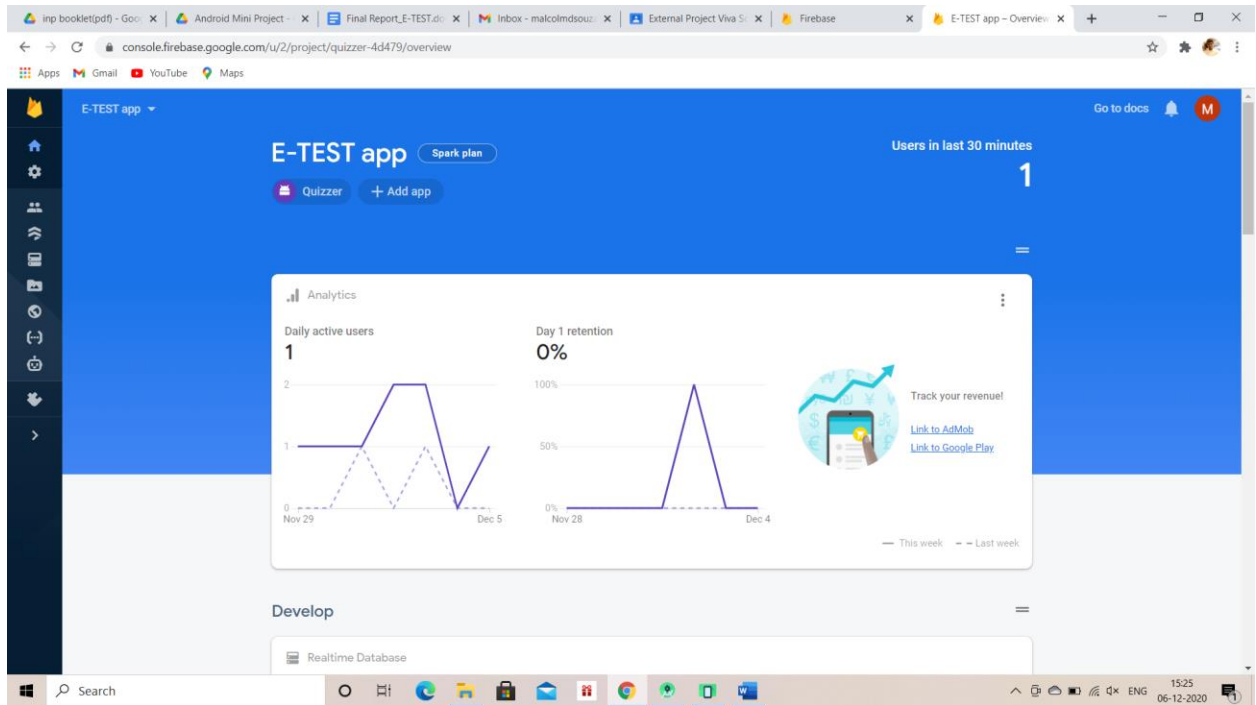


Figure 57 Firebase console of our app, It handles our database and authentication activity

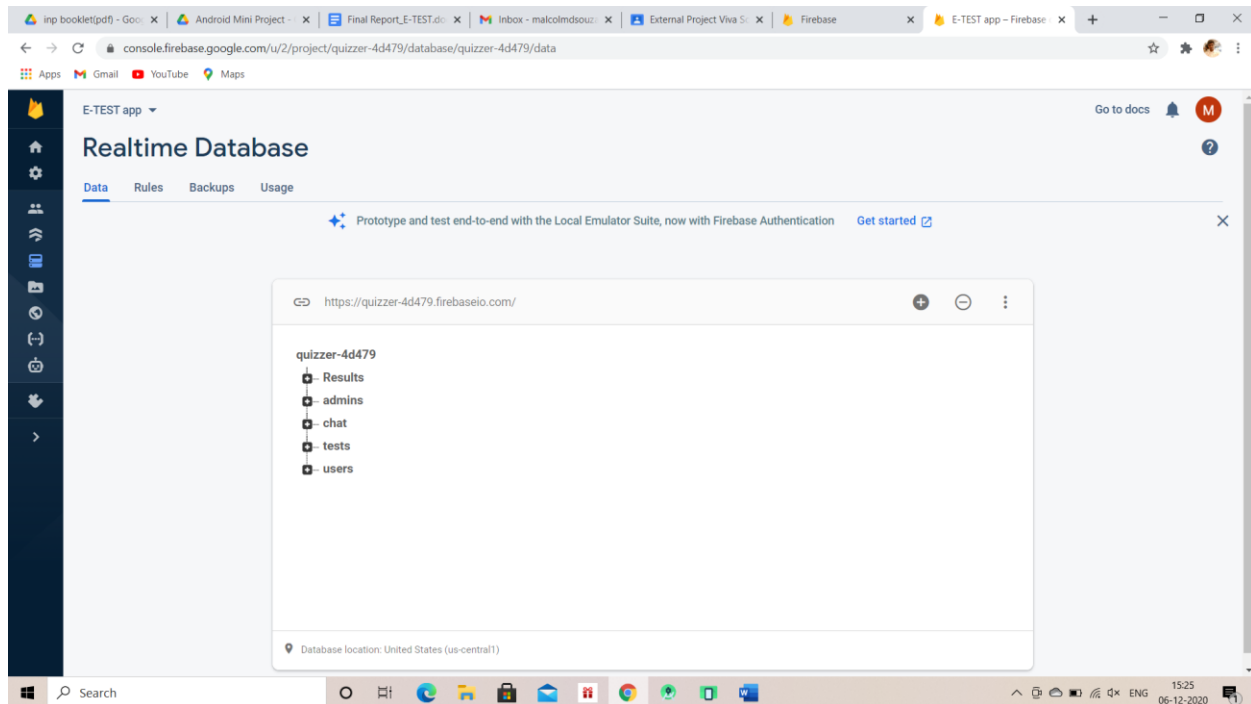


Figure 58 Our real time database, our results, admin id, chats, tests, and users information are stored here in tree like structure

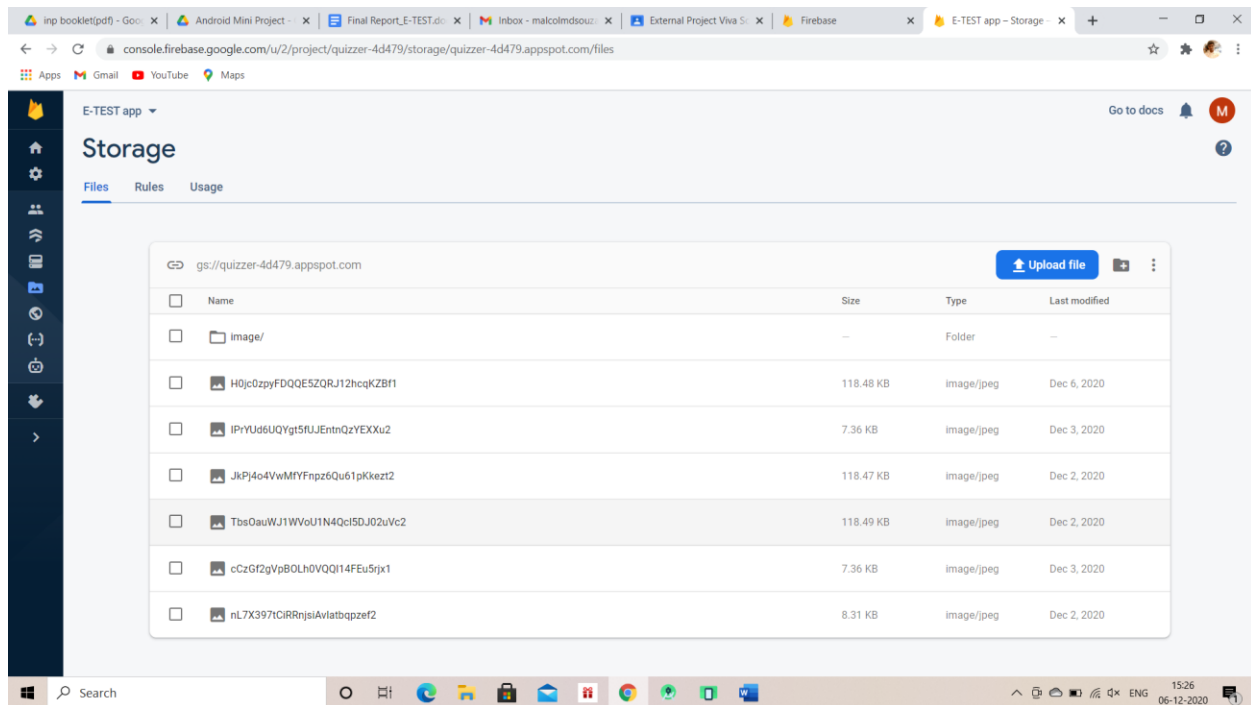


Figure 59 Our uploaded profile picture are stored here in storage section of firebase

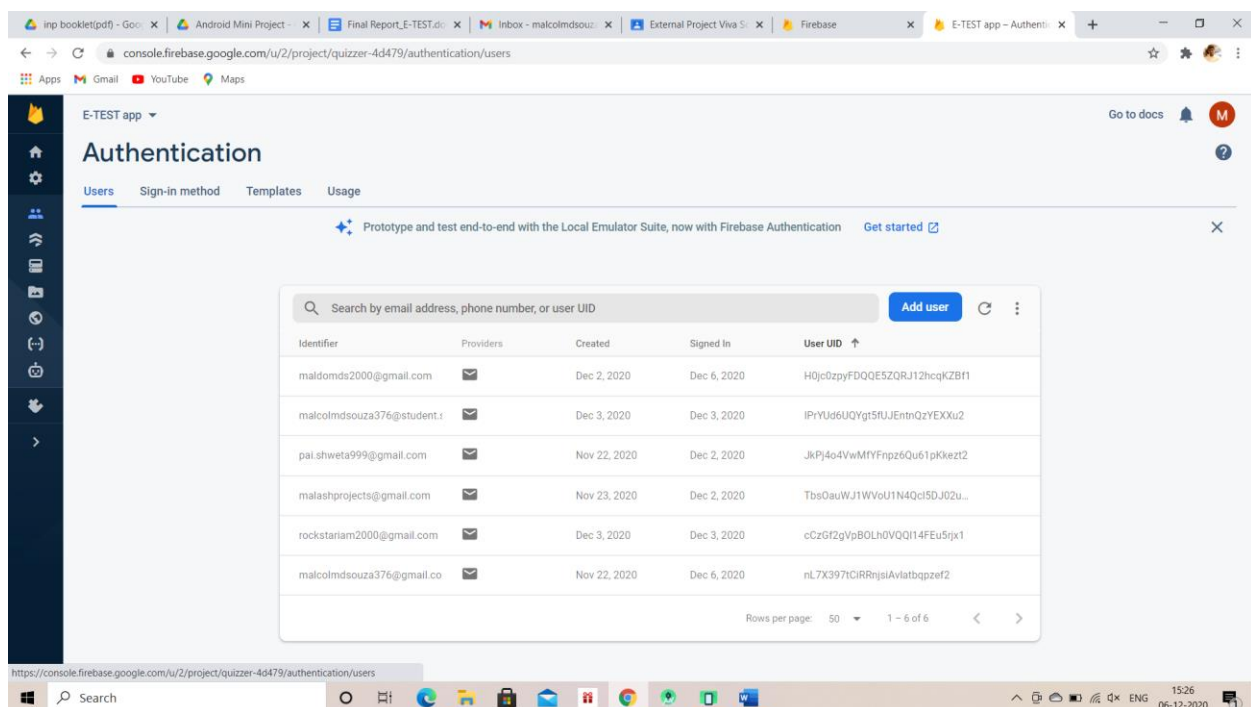


Figure 60 Our user authentication information like email and password are stored here and password is encrypted and stored and user is only registered if user authenticates his email id.

## Chapter 6

### Results and Discussion

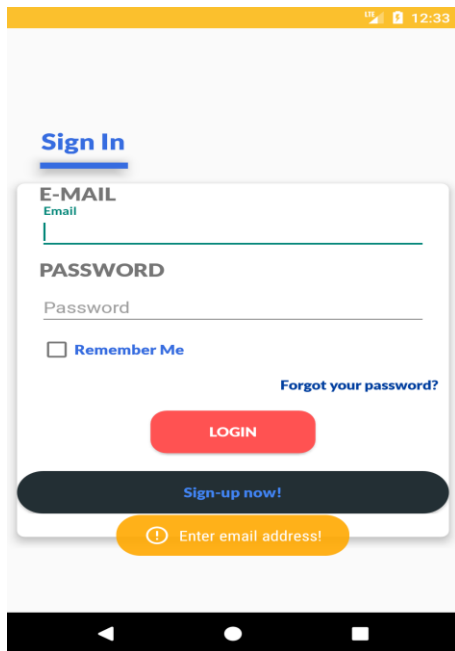


Figure 61 No Input Login Error

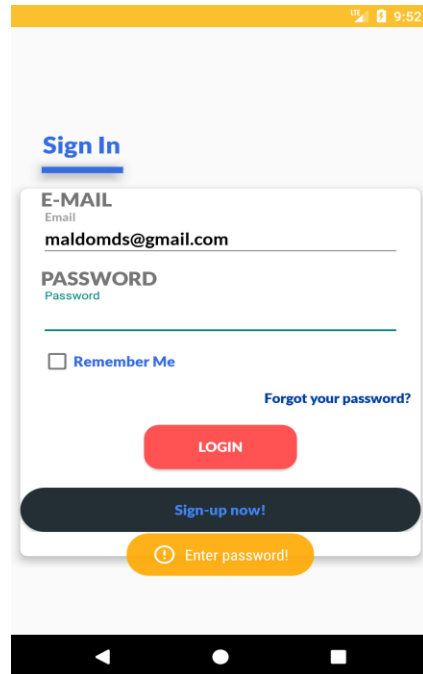


Figure 62 No password Input Login Error

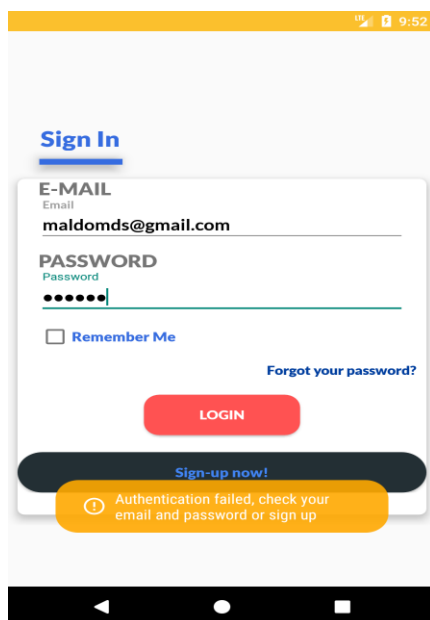


Figure 63 Non Registered User Login Error

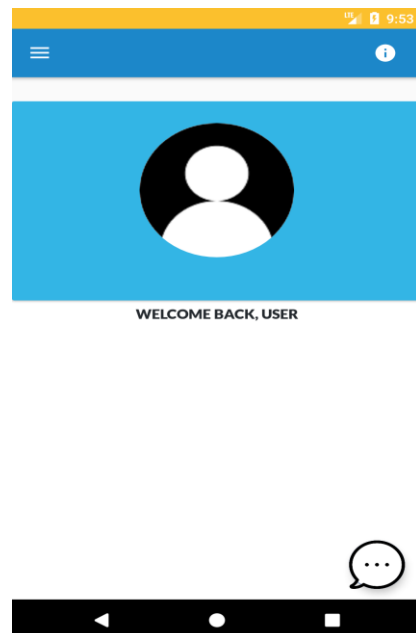


Figure 64 Successful Login into the system

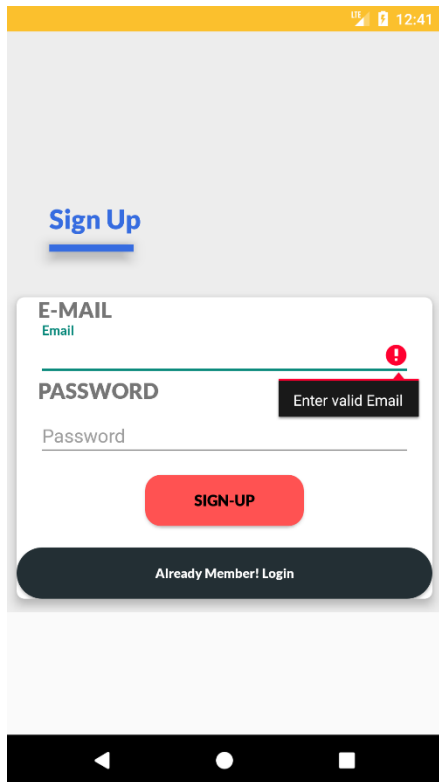


Figure 65 No Input Registration Error

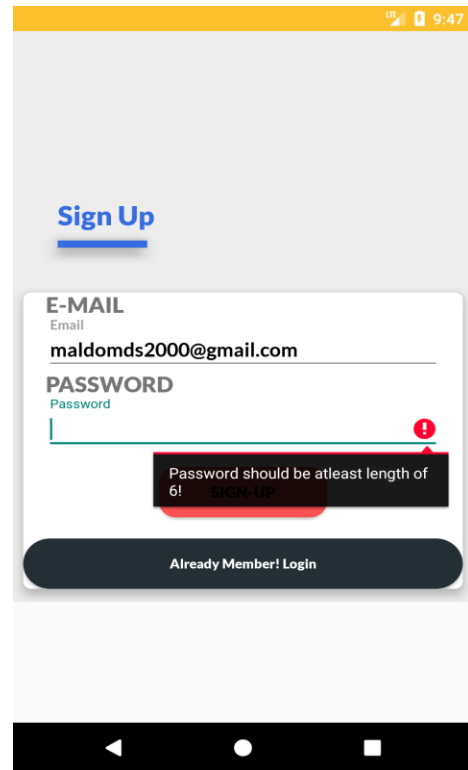


Figure 66 Password size small Registration Error

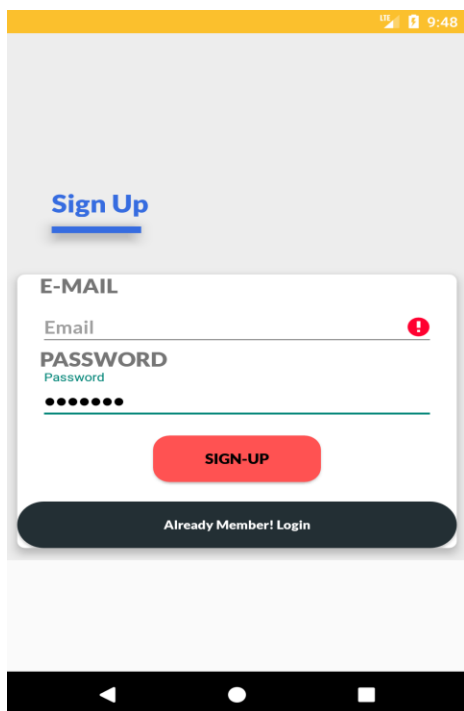


Figure 67 Already registered.

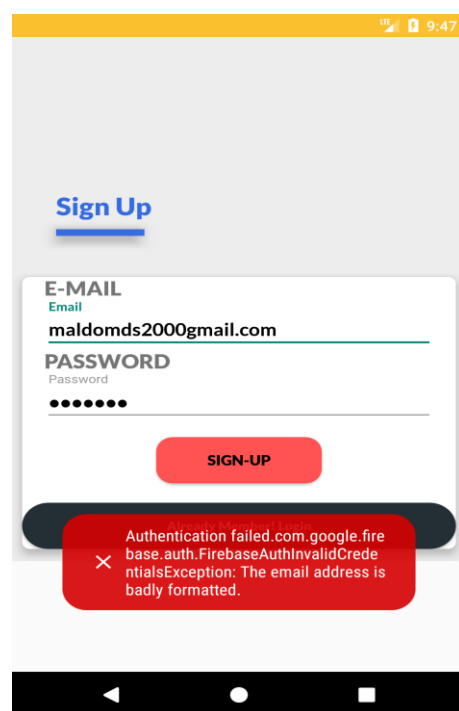


Figure 68 Invalid Email error signup page

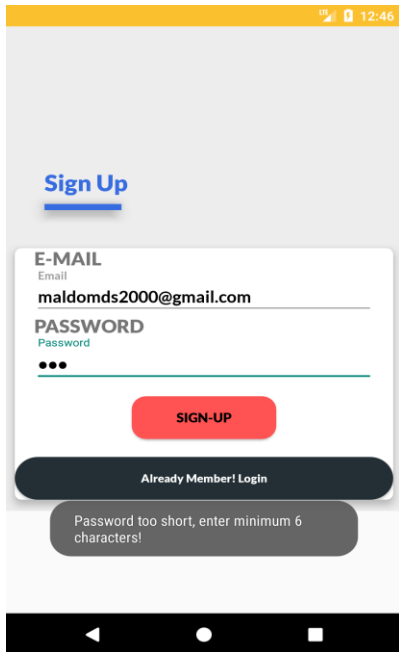


Figure 69 Invalid/ No Input Verification Error 2

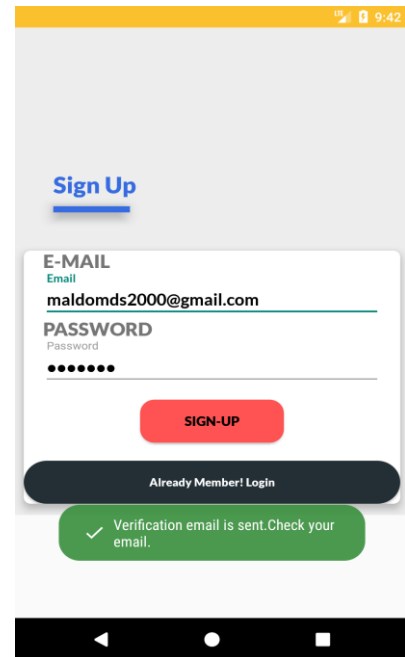


Figure 70 Need to verify the email

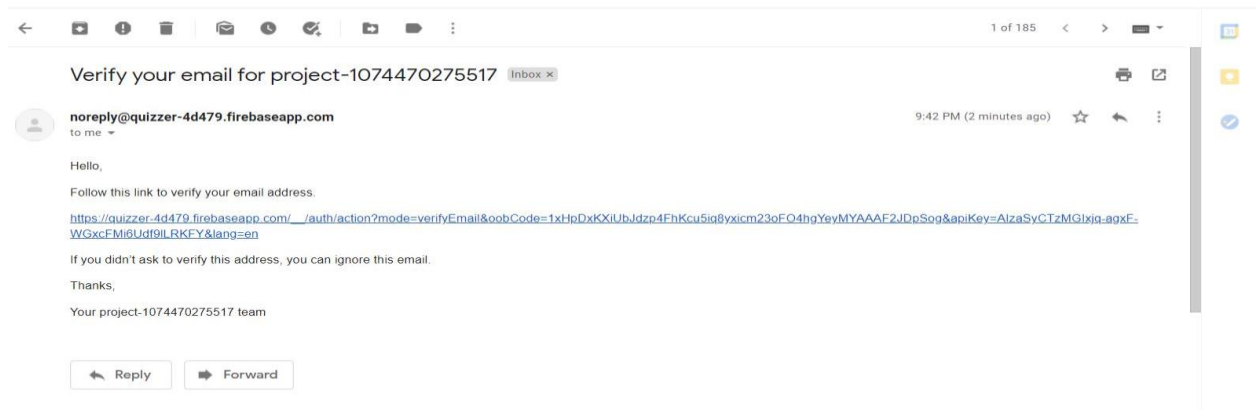


Figure 71 Email verification link sent by mail.

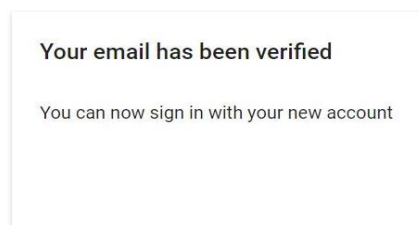


Figure 72 Email verification successful

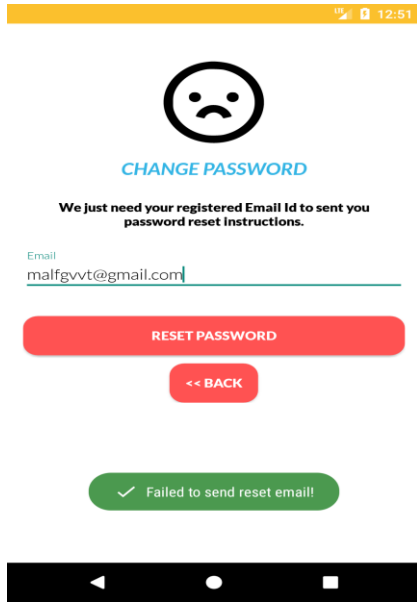


Figure 73 Change password page

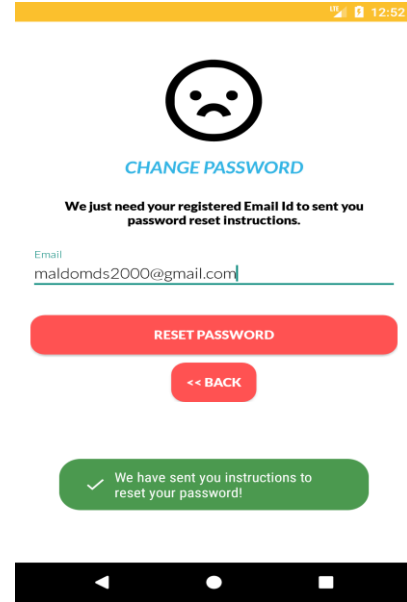


Figure 74 Toast on reset request successful

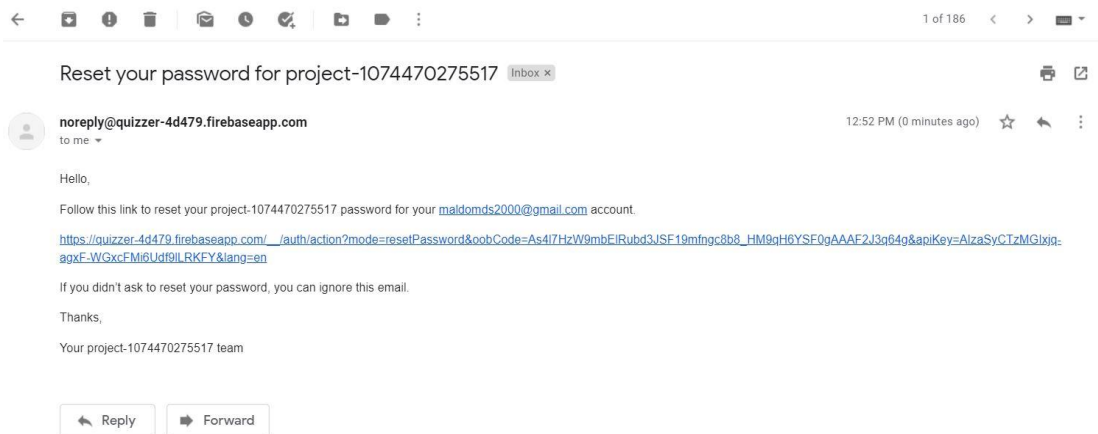


Figure 75 Password reset link sent on mail

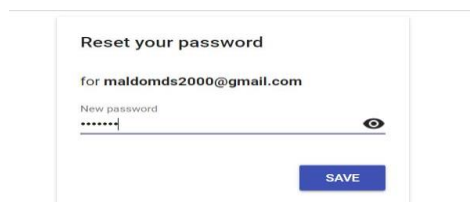


Figure 76 Entering new password



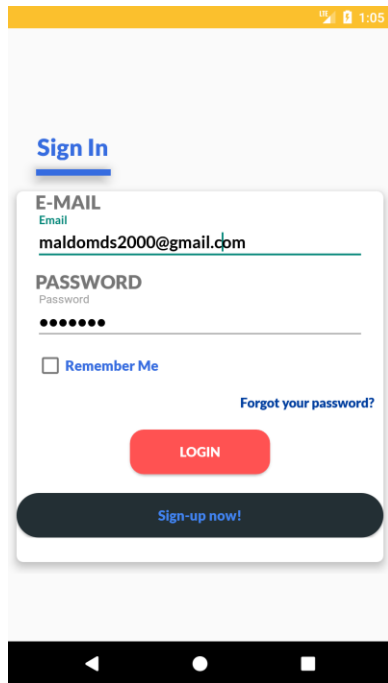


Figure 77 Correct Password and email

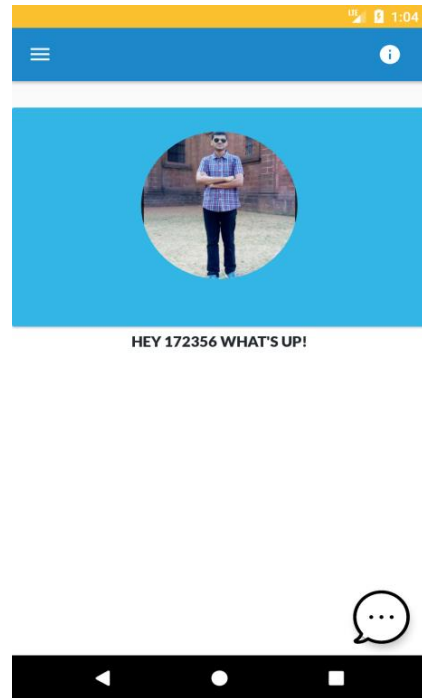


Figure 78 Successful Login after reset

## Chapter 7

### **Conclusion**

By this application we aim to provide ease of conducting examinations for students so that the absence of physical exams does not affect the examination process. We also authenticate the user by email authentication and also allow the user to change his/her password.

This android app will provide everyone with an online platform to attempt a test from their Android Mobile device itself, since all can't afford a laptop. Our test will be equipped with a timer which keeps a track of when the user starts the test and ends the test when the time limit set by admin is reached, thereby ensuring that all of the students got equal amount of time for the test not more nor less. We have successfully implemented the access modules: login and registration user pages, user specific GUI pages, Contact us page and T&C page. Account specific functions in their respective pages, adding location feature, adding camera feature was successfully implemented.

## **Future Scope**

In addition to the features imbibed in the E-Test, one can add the following features to increase the functionality:

- Proctoring System.
- Daily Test Alerts.
- Facility to upload answers for a descriptive test.

## References

- [1] Siti Aisyah, "Development of Continuous Authentication System on Android-Based Online Exam Application", International Conference on Information Technology Systems and Innovation, 2018.
- [2] Shrikrushna S. Kure, "Online Examination System Based on Android Mobile", International Journal of Engineering Research in Computer Science and Engineering, Volume 4, Issue 4, April 2017.
- [3] Jay Lakhani, "E-Learning and Online Examination Android Application using Smart Adaptive Algorithm", International Journal for Research in Engineering Application & Management, Vol-02, Issue 01, APR 2016.
- [4] Rishish Mohan Bhatnagar, "Online Examination System for MCQ's", IJCSMC, Vol. 6, Issue. 5, May 2017.