VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI-590018, KARNATAKA

A Mini Project Report on

SMART QUIZ

Submitted in partial fulfillment of the requirements as a part of Mobile Application Development Laboratory for the VI Semester of degree of ***Bachelor of Engineering in Information Science and Engineering*** *of VisvesvarayaTechnological University, Belagavi*

by

SHARANYA RP **1RN20IS143**
SIDHARTH S PAI **1RN20IS160**

Under the Guidance of

Ms. PURNIMA S M
Assistant Professor
Department of ISE

Department of Information Science and Engineering

RNS Institute of Technology

Dr. Vishnuvaradhana Road, Rajarajeshwari Nagar post, Channasandra, Bengaluru-560098

2022-2023

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhana Road, Rajarajeshwari Nagar post,

Channasandra, Bengaluru - 560098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled " **SMART QUIZ** "has been successfully completed By **SHARANYA RP (1RN20IS143)** Bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements for the awardof degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2022-2023**. The mini project report has been approved as it satisfies the academic requirements with respect to the Mobile Applications Development laboratory.

Ms. Purnima S M Mini Project Guide Assistant Professor Department of ISE	**Dr. Suresh L** Professor and HOD Department of ISE RNSIT	**Dr. Ramesh Babu H S** Principal RNSIT

External Viva

Name of the Examiners	**Signature with Date**
1. _____	**1.** _____
2. _____	**2.** _____

DECLARATION

We, **SHARANYA RP [USN:1RN20IS143], SIDHARTH S PAI [USN:1RN20IS160]**,

students of VI Semester BE, in Information Science and Engineering, RNS Institute of Technologyhereby declare that the Project entitled "**Smart Quiz App**" has been carried out by us and submitted inpartial fulfillment of the requirements for the *VI Semester of degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2022-2023.

Place: Bengaluru

Date: 19/06/2023

Sharanya RP (1RN20IS143)

Sidharth S Pai (1RN20IS160)

ABSTRACT

This work involves the development of an Android-based application that offers multiple-choice questions in various categories such as Math, Geography, and Literature. The purpose of this application is to assist students in preparing for exams by assessing their knowledge through multiple-choice questions. The application enables users to practice questions in distinct categories and aims to help them improve their understanding.

The main objective of the application is to provide students with a platform to enhance their knowledge and skills. Each category consists of 5 questions, and the score is updated dynamically, with the final score being displayed at the end of each category. The application is designed to be user-friendly, allowing users to identify their weaknesses and focus on specific topics to excel in their exams. Additionally, the application includes features such as the ability to exit or restart a category, allowing users to revisit questions or start over if needed.

ACKNOWLEDGMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

We would like to profoundly thank the **Management** of **RNS Institute of Technology** for providing such a healthy and vibrant environment to conduct the project work.

We would like to express our sincere thanks to our beloved Principal **Dr. Ramesh Babu H S** for his support and inspiration towards the attainment of knowledge.

We wish to place on record our words of gratitude to **Dr. Suresh L,** Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind our project work.

We would also like to thank our mini project guide **Ms. Purnima S M,** Assistant Professor, Department of ISE, RNSIT, Bengaluru, for her valuable input, suggestions, time, and guidance.

We would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped me to conduct the project work.

Place: Bengaluru

Date:19/06/2023

Sharanya RP
(1RN20IS143)
Sidharth S Pai
(1RN20IS160)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION TO ANDRIOD

1.1 History

In the past mobile phones were used only to make calls but with the introduction of smartphones the mobile phone has evolved to a low powered handheld processing system. This evolution was caused by the operating system for mobile phones making them smart that have processing and storage of their own. Now the mobile provides numerous functionalities from calling to texting, multimedia sharing, emails, socializing applications, word processor, excel sheets to various multiplayer games and much more.

The operating system for these handheld devices is iOS by Apple Inc., Windows by Windows Inc., and Android by Google. Among the competitors in the smartphone operatingsystem industry Android holds the largest market share in terms of units shipped worldwide and number of users.

Android is an open-source operating system based on Linux kernel on which applications run on an application framework that controls the activities supported by the libraries and Dalvik virtual machine which compiles and converts all java class files into a single file. There can be a number of virtual machines running simultaneously on a single device managing different applications or instances of an application.

1.2 Android Versions

The development of the Android operating system was started in 2003 by Android, Inc. Later, it was purchased by Google in 2005. The beta version of Android OS was released on November 5, 2007, while the software development kit (SDK) was released on November 12, 2007.

The first Android mobile was officially released with Android 1.0 of the T-Mobile G1 (aka HTC Dream) in October 2008. The first Android version which was released underthe numerical order format was Android 10.

Code name	Version numbers	API level	Release date
No codename	1.0	1	September 23, 2008
No codename	1.1	2	February 9, 2009
Cupcake	1.5	3	April 27, 2009
Donut	1.6	4	September 15, 2009
Eclair	2.0 - 2.1	5 - 7	October 26, 2009
Froyo	2.2 - 2.2.3	8	May 20, 2010
Gingerbread	2.3 - 2.3.7	9 - 10	December 6, 2010
Honeycomb	3.0 - 3.2.6	11 - 13	February 22, 2011
Ice Cream Sandwich	4.0 - 4.0.4	14 - 15	October 18, 2011
Jelly Bean	4.1 - 4.3.1	16 - 18	July 9, 2012
KitKat	4.4 - 4.4.4	19 - 20	October 31, 2013
Lollipop	5.0 - 5.1.1	21- 22	November 12, 2014
Marshmallow	6.0 - 6.0.1	23	October 5, 2015
Nougat	7.0	24	August 22, 2016
Nougat	7.1.0 - 7.1.2	25	October 4, 2016
Oreo	8.0	26	August 21, 2017
Oreo	8.1	27	December 5, 2017
Pie	9.0	28	August 6, 2018
Android 10	10.0	29	September 3, 2019

Code name	Version numbers	API level	Release date
Android 11	11	30	September 8, 2020
Android 12	12	31	October 4, 2021
Android 12L	12.1	32	March 7, 2022
Android 13	13	33	August 15, 2022
Android 14	14	34	Q3, 2023
Android 15	15	TBA	Q3, 2024

Table 1.1: Android Versions

1.3 Android Architecture

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.

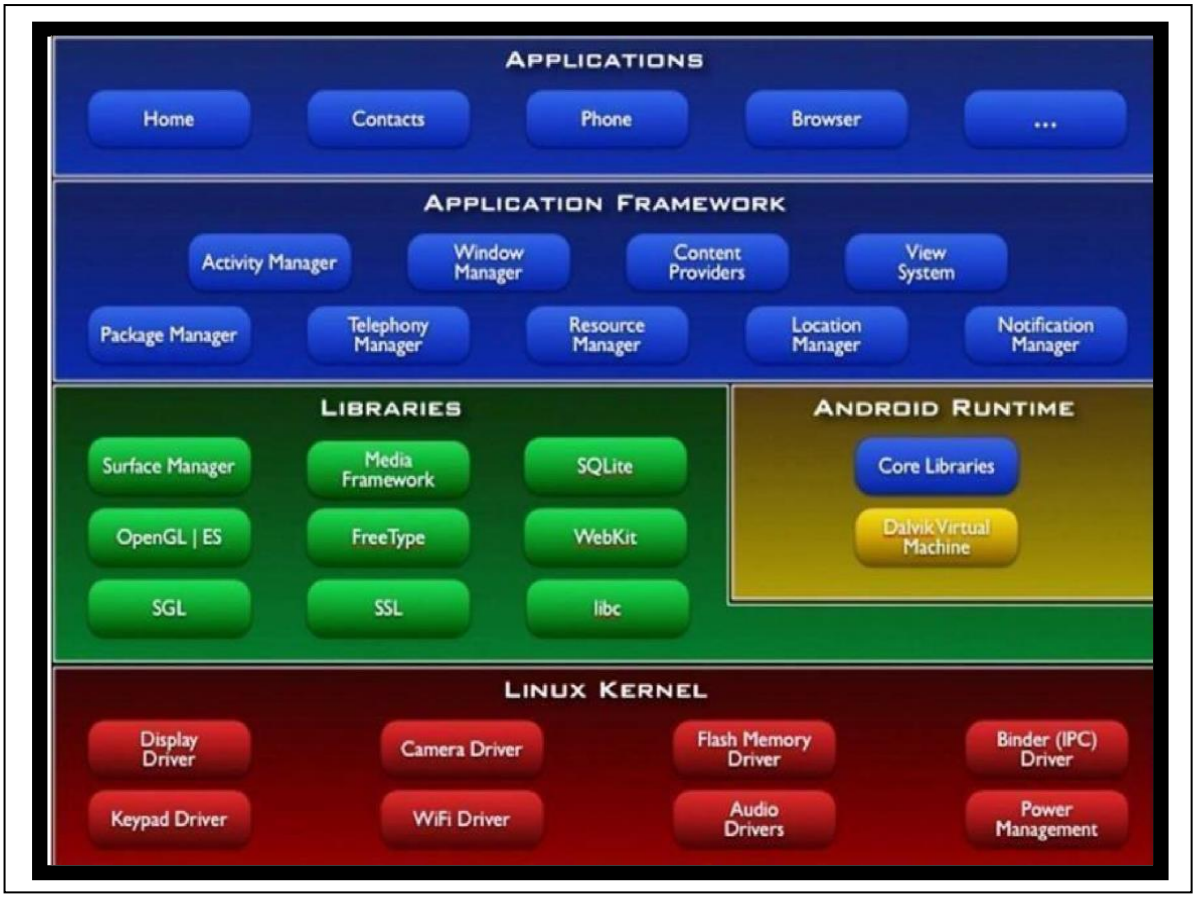

Figure 1.1: Android Architecture diagram

Linux Kernel: This is the layer at the very bottom of the Android architecture.

All other layers run on top of the Linux kernel and rely on this kernel to interact with the hardware. This layer contains all the essential hardware drivers which help to control and communicate with the hardware.

Libraries: This is a set of common functions of the application framework that enables the device to manage several types of data. Some of the most important set of libraries that are included are – Web kit, which is the browser engine to display HTML, OpenGL used to render2- D or 3-D graphics on to the screen, SQLite which is a useful repository for storing and sharing of application data.

A summary of some key core Android libraries available to the Android developer is as follows.

- android.app − Provides access to the application model and is the cornerstone of all Android applications.
- android.content − Facilitates content access, publishing and messaging between applications and application components.
- android.Database − Used to access data published by content providers and includes SQLite database management classes.
- android.opengl − A Java interface to the OpenGL ES 3D graphics rendering API
- android.os − Provides applications with access to standard operating system services including messages, system services and inter-process communication.
- android.text − Used to render and manipulate text on a device display.
- android.view − The fundamental building blocks of application user interfaces.
- android.widget − A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.
- android.webkit − A set of classes intended to allow web-browsing capabilities to be built into applications.

Android Runtime: The Android runtime mainly consists of the Dalvik Virtual Machine (DVM). DVM is very much like the standard Java Virtual Machine (JVM) exceptthat it is optimized for mobile devices that have low processing power and low memory. DVM generates a.dex file from the .class file at compile time and provides higher efficiency in low resources devices. Each application has its own process and an instance of DVM. Android runtime also provides core libraries that enable the Android developers to create applications using the Java language.

Application Framework: The Android runtime mainly consists of the DalvikVirtual Machine (DVM). DVM is very much like the standard Java Virtual Machine (JVM) except that it is optimized for mobile devices that have low processing power and low memory. DVM generates a.dex file from the .class file at compile time and provides higherefficiency in low resources devices. Each application has its own process and an instance of DVM. Android runtime also provides core libraries that enable the Android developersto create applications using the Java language.

Applications: This is the topmost layer in architecture and the layer wherethe application that we develop fits in. This layer provides several pre-installed applications that are default for certain things like Contacts Books, Browser etc.

1.3 Android Studio Installation

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools.

To support application development within the Android operating system, Android Studio uses a Gradle-based build system, emulator, code templates, and GitHub integration. Everyproject in Android Studio has one or more modalities with source code and resource files. These modalities include Android app modules, Library modules, and Google App Enginemodules.

Procedure to be followed to download and install android studio:

STEP 1 : Android Studio and the Software Development Kit can be downloaded directly from any web browser using the below link.

https://developer.android.com/studio

STEP 2 : Android Studio is available for Mac, Windows, and Linux desktopplatforms.

Windows

To install Android Studio on Windows, proceed as follows:

i. If you downloaded an .exe file (recommended), double-click to launch it. If you downloaded a .zip file, unpack the ZIP, copy the android-studio folder into your Program Files folder, and then open the android-studio > bin folder and launch studio64.exe (for 64-bitmachines) or studio.exe (for 32-bit machines).

ii. Follow the setup wizard in Android Studio and install any SDK packages that it recommends.

Mac

To install Android Studio on your Mac, proceed as follows:

i. Launch the Android Studio DMG file.

ii. Drag and drop Android Studio into the Applications folder, then launch it.

iii. Select if you want to import previous Android Studio settings, then press OK .

iv. The Android Studio Setup Wizard guides you through the rest of the setup, which includes downloading Android SDK components that are required for development.

Linux

To install Android Studio on Linux, proceed as follows:

i. Unpack the .zip file you downloaded to an appropriate location for your applications,such as within /usr/local/ for your user profile, or /opt/ for shared users. If you are using a64-bit version of Linux, make sure you first install the required libraries for 64-bit machines.

ii. To launch Android Studio, open a terminal, navigate to the android-studio/bin/ directory, and execute studio.sh.

iii. Select whether you want to import previous Android Studio settings or not, then clickOK.

Chapter 2

INTRODUCTION TO PROJECT

2.1 Overview of the Project

In this Quiz, there are three subject quiz Math , geography, and literature. This application is developed for preparing for Students exams and assessing their knowledge allowing the users to prepare the multiple-choice questions for different categories . The main goal of the application is to enable users to practice for subjective tests conducted by Schools and different institutions.

The user can take one quiz at a time. Each quiz contains five questions. After finishing the quiz, the user can retake the quiz. If the user wants to change the subject,they can also do that. After giving the test the user can check their scores andperformance.

2.2 Aim of the Project

Our Aim is to develop creativity and individuality in problem solving and performing tasks. to prepare students to work in teams or as an individual. to prepare students to improve their skills and knowledge related to specific job positions individually. to enable students to do self-study.

Research has shown that games are essential for healthy development in early childhood and beyond. Play lets children practice what they know, and what they do not. Improve the knowledge of students and upgrade their skills.

Chapter 3

SYSTEM DESIGN

3.1 System Requirements

3.1.1 Hardware Requirements:

- Processor: Pentium IV or above
- RAM: 8 GB or more
- Hard Disk: 2GB or more

3.1.2 Software Requirements:

- Operating System: Windows 7 or above
- IDE: Android Studio (Version : Electric Eel | 2022.1.1)
- API Level: 19 or above

3.2 User Interface

XML Code for the initial UI screen:

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="QuizForFun"
        android:roundIcon="@mipmap/ic_launcher"
        android:supportsRtl="true"
        android:theme="@style/Theme.QuizForFun">
        <activity android:name=".GeographyOrLiteratureQuizActivity"></activity>
        <activity android:name=".FinalResultActivity" />
        <activity android:name=".MathQuizActivity" />
        <activity android:name=".EditPasswordActivity" />
        <activity android:name=".HistoryActivity" />
        <activity android:name=".RuleActivity" />
        <activity android:name=".QuizOptionActivity" />
        <activity android:name=".RegisterActivity" />
```

```
        <activity
            android:name=".LoginActivity"
            android:theme="@style/splashScreenTheme">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Chapter 4

IMPLEMENTATION

In this quiz application, there are three quizzes available: Math, Geography, and Literature. To begin the quiz, users need to sign up on the page by providing their username, email id, and password. Once signed up, users can log in to the home page of the quiz app.

On the home page, users can access the "Rules" tab to familiarize themselves with the guidelines for taking the quizzes. There is also a "Start Quiz" option that allows users to initiate their chosen quiz. After clicking on the "Start Quiz" option, users will be presented with the quiz options, namely Math, Geography, and Literature. They can select any of these quizzes to proceed.

Once a quiz is selected, the quiz session begins, displaying five questions one at a time. Users must answer each question and click the "Submit" button to proceed to the next question. After answering all five questions, the system generates a result based on the user's responses. Each correct answer earns 5 marks, while each wrong answer incurs a deduction of 2 marks.

Upon completion of the quiz, users can view their answers in the "History" section on the home page. This section provides a record of their previous quiz attempts. Additionally, users have the option to change their current password to a new one using the "Change Password" section on the home page.

To ensure data security, users can log out from the app, and their information will be stored securely

Backend Code:

```
package com.sagarkhurana.quizforfun;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.TextView;

import com.sagarkhurana.quizforfun.data.Attempt;
import com.sagarkhurana.quizforfun.data.UserDatabase;
import com.sagarkhurana.quizforfun.data.UserDatabaseClient;
import com.sagarkhurana.quizforfun.other.Constants;
import com.sagarkhurana.quizforfun.other.SharedPref;
import com.sagarkhurana.quizforfun.other.Utils;
import java.util.Calendar;
public class FinalResultActivity extends AppCompatActivity {
    2 usages
    private TextView tvSubject, tvCorrect, tvIncorrect, tvEarned, tvOverallStatus, tvDate;
    sharanya
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_final_result);

        Intent intent = getIntent();
        int correctAnswer = intent.getIntExtra(Constants.CORRECT, defaultValue: 0);
        int incorrectAnswer = intent.getIntExtra(Constants.INCORRECT, defaultValue: 0);
        String subject = intent.getStringExtra(Constants.SUBJECT);
        String email = SharedPref.getInstance().getUser( context: this).getEmail();
        int earnedPoints = (correctAnswer * Constants.CORRECT_POINT) -
                (incorrectAnswer * Constants.INCORRECT_POINT);
        tvSubject = findViewById(R.id.textView16);
        tvCorrect = findViewById(R.id.textView19);
        tvIncorrect = findViewById(R.id.textView27);
        tvEarned = findViewById(R.id.textView28);
        tvOverallStatus = findViewById(R.id.textView29);
        tvDate = findViewById(R.id.textView30);
```

```
findViewById(R.id.imageViewFinalResultQuiz).setOnClickListener(new View.OnClickListener(  7
 sharanya
{   @Override
    public void onClick(View view) {
        Intent intent = new Intent( packageContext: FinalResultActivity.this,MainActivity.class);
        startActivity(intent);
        finish();
    }
});
 sharanya
findViewById(R.id.btnFinishQuiz).setOnClickListener(new View.OnClickListener() {
     sharanya
    @Override
    public void onClick(View view) {
        Intent intent = new Intent( packageContext: FinalResultActivity.this,MainActivity.class);
        startActivity(intent);
        finish();
    }
});
Attempt attempt = new Attempt(
        Calendar.getInstance().getTimeInMillis(),
        subject,
        correctAnswer,
        incorrectAnswer,
          earnedPoints,
          email );
 getOverAllPoints(attempt);
}

5 usages  sharanya
@Override
public void onBackPressed() {
    super.onBackPressed();
    Intent intent = new Intent( packageContext: this,MainActivity.class);
    startActivity(intent);
    finish();
}
1 usage  sharanya
private void getOverAllPoints(Attempt attempt) {
    GetOverallPointsTask getOverallPointsTask = new GetOverallPointsTask(attempt);
    getOverallPointsTask.execute();
}
2 usages  sharanya
class GetOverallPointsTask extends AsyncTask<Void, Void, Void> {
    6 usages
    private final Attempt attempt;
    3 usages
    private int overallPoints = 0;
```

```
        1 usage  sharanya
        public GetOverallPointsTask(Attempt attempt) { this.attempt = attempt; }
        sharanya
        @Override
        protected Void doInBackground(Void... voids) {
            UserDatabase databaseClient = UserDatabaseClient.getInstance(getApplicationContext());
            overallPoints = databaseClient.userDao().getOverAllPoints(attempt.getEmail());
            return null;
        }

        7 usages  sharanya
        @Override
        protected void onPostExecute(Void aVoid) {
            super.onPostExecute(aVoid);

            attempt.setOverallPoints(overallPoints + attempt.getEarned());
            displayData(attempt);
            SaveUserAttemptTask saveUserAttemptTask = new SaveUserAttemptTask(attempt);
            saveUserAttemptTask.execute();

            Log.d( tag: "OVERALL POINTS", String.valueOf(overallPoints));
        }
    }
    1 usage  sharanya
    private void displayData(Attempt attempt) {
        tvIncorrect.setText(String.valueOf(attempt.getIncorrect()));
        tvEarned.setText(String.valueOf(attempt.getEarned()));
        tvOverallStatus.setText(String.valueOf(attempt.getOverallPoints()));
        tvDate.setText(Utils.formatDate(attempt.getCreatedTime()));
    }
    2 usages  sharanya *
    class SaveUserAttemptTask extends AsyncTask<Void, Void, Void> {
        3 usages
        private Attempt attempt;
        1 usage  sharanya
        public SaveUserAttemptTask(Attempt attempt) { this.attempt = attempt; }
        sharanya *
        @Override
        protected Void doInBackground(Void... voids) {
            UserDatabase databaseClient = UserDatabaseClient.getInstance(getApplicationContext());
            databaseClient.userDao().insertAttempt(attempt);
            return null;}
        7 usages  sharanya
        @Override
        protected void onPostExecute(Void aVoid) {
            super.onPostExecute(aVoid);
            Log.d( tag: "Attempt Saved", attempt.toString());
        }
    }
}
```

Chapter 5

TESTING

Software testing is an essential phase in the development life cycle of an application. Testing ensures that the developed system meets its functional and non-functional requirements. Twoimportant terms in software testing are Verification and Validation. Verification is the processof evaluating work-products like requirement specs, design specs and test cases etc. of different development phases to make sure that they meet the requirements for that phase. Itensures that the system is built in the right way. Whereas Validation is the process of evaluating the software at the end of the development phase to make sure that it meets the business requirements. It is used to make sure that the product fulfils its intended use, and thatthe product is built right.

One of the most important tools to test and debug an Android app is the Dalvik debug monitorserver (DDMS) that is part of the Android framework. DDMS helps you to debug your codeas it prints errors, warnings, and other information from your code. It also provides stack tracesfor exceptions on the Logcat output.

Various other testing strategies have been adopted to make sure the correctness of the Smart Quiz app. They are discussed in this chapter.

Test Case No.	Input	Expected Output	Actual Output	Remarks
1.	User getsto choose the categories (Math, Geography and Literature, technical knowledge)	User can try of selecting the questions of their general interest.	User can make an option of selecting the questions by opting their field of interest.	Pass
2.	User gets an option to exit the quiz.	The application basically tends to stop the quiz application with exit command	We basically tend to exit the quiz application	Pass
3.	User can restart the quiz	User basically tend to return to the home page over here using restart functionality.	User can restart the quiz using restart functionality.	Pass

4.	User can go back to the home page	User basically tend to return to the home page over here using go to-home functionality	User can go back to the home page using go to-home functionality	Pass
5.	User can exit the quiz application using the exit functionality	User basically tend to exit the application using exit functionality.	Users tend to exit the application using the exitfunctionality.	Pass
6.	User can view the about-us page under the menu section.	User basically tend to understand the workflow of the application.	User basically tend to understand the work-flow of the application.	Pass

Table 5.1: Testing Table

Chapter 6

RESULTS

All the activities provided in the application and its operation have been presented as Snapshots. A detailed view of all Snapshots of the application is given in this section.

Rules :

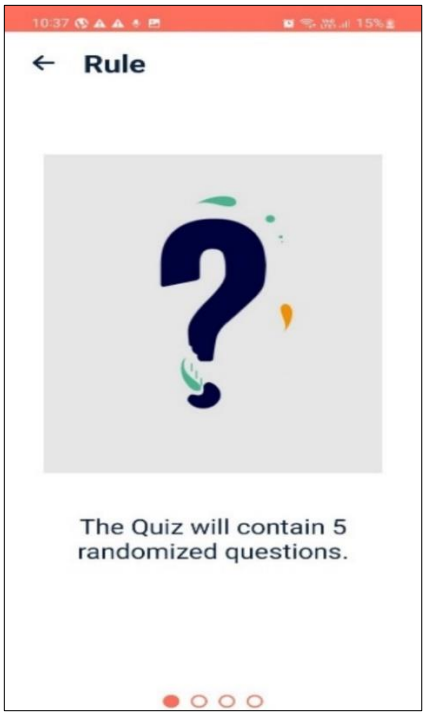

Figure 6.1.1: Rule 1

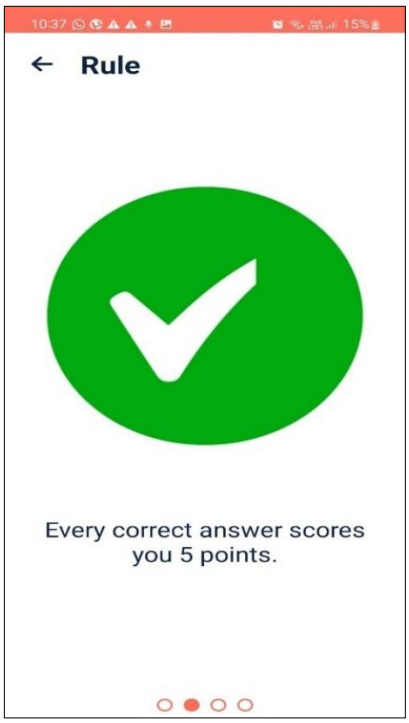

Figure 6.1.2: Rule 2

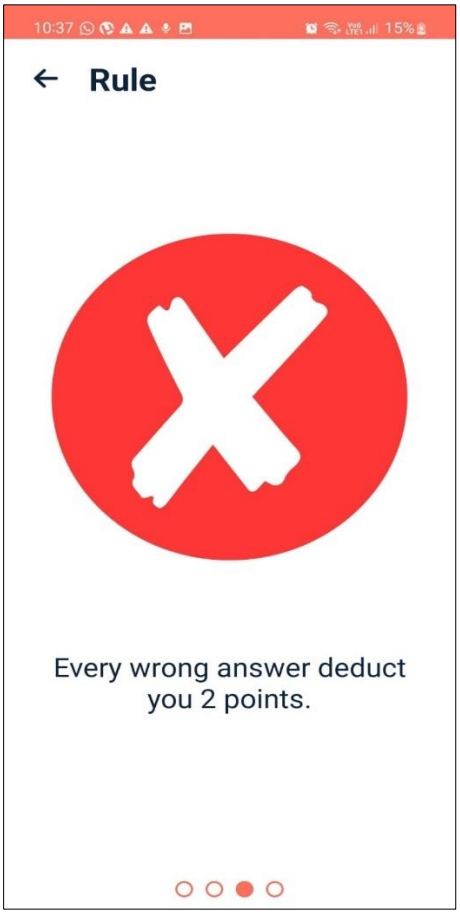

Figure 6.1.3: Rule 3

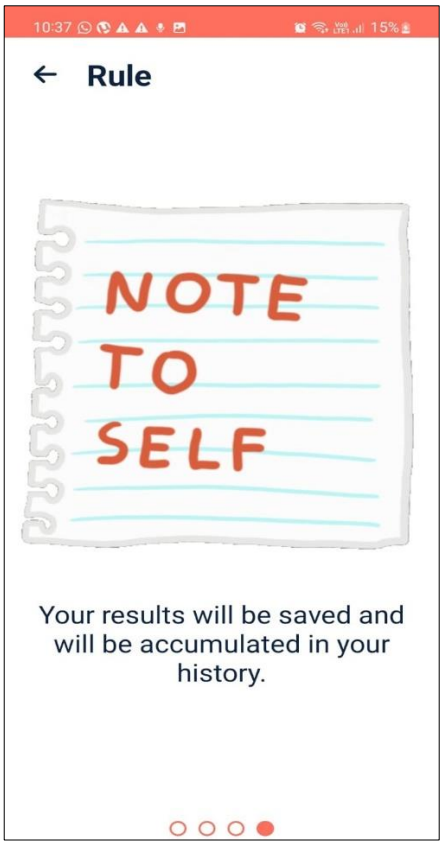

Figure 6.1.4: Rule 4

Steps to Start the quiz:

Step 1: The user must first sign up in the app to start the quiz.

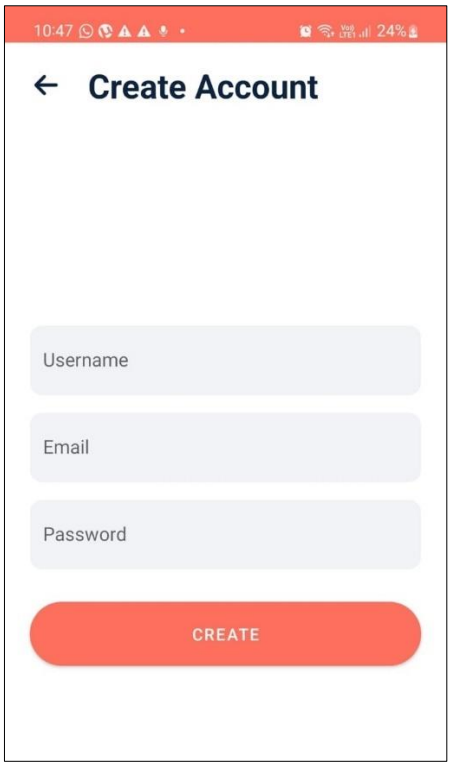

Figure 6.1.5: Sign up page.

Step 2: Now the user has to login into the app with the username and password to start the app.

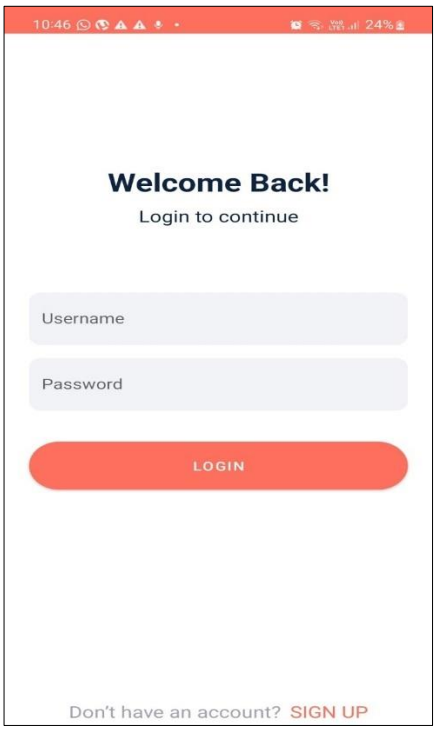

Figure 6.1.6: Login Page

Step 3: Now we move to the home page and "start quiz" option is present.

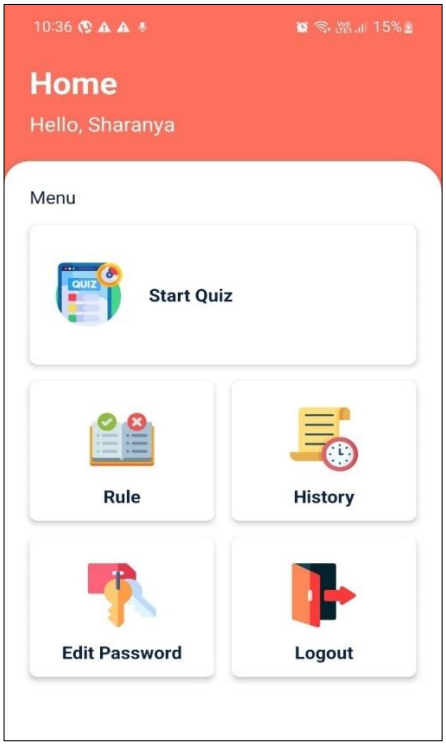

Figure 6.1.7: Home Page

Step 4: After clicking start quiz we move to the quiz option page and there you can start quiz of math, geography, and literature.

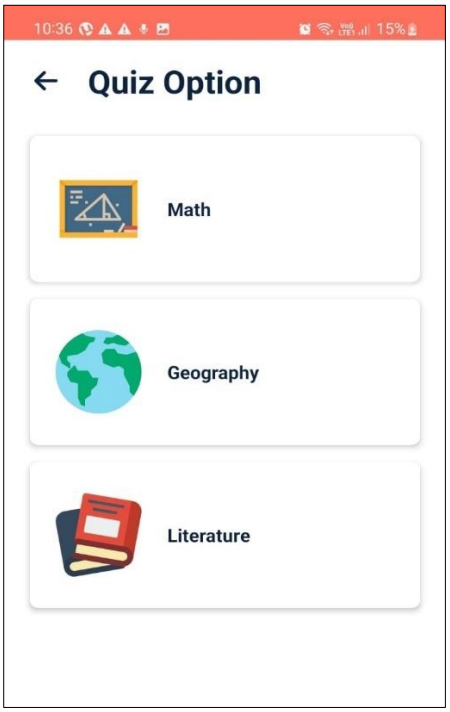

Figure 6.1.8: Quiz Option

Math Quiz: The snapshot shows the front page of the start screen of the math quiz.

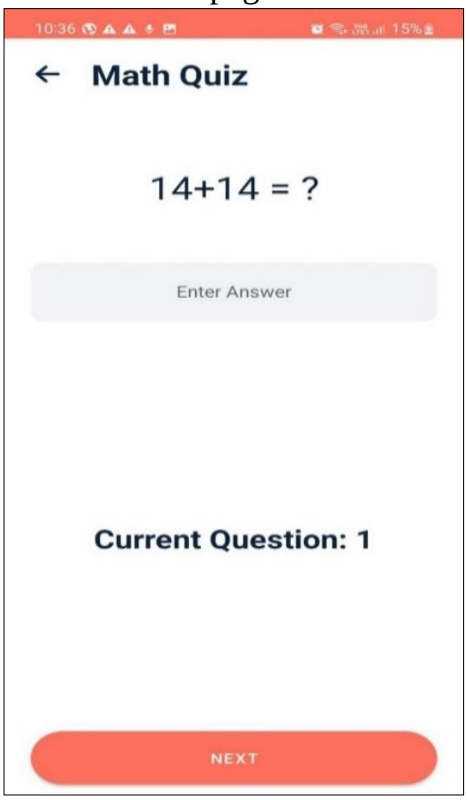

Figure 6.1.9: Start the math quiz.

Geography Quiz: The snapshot shows the front page of the start screen of the geography quiz.

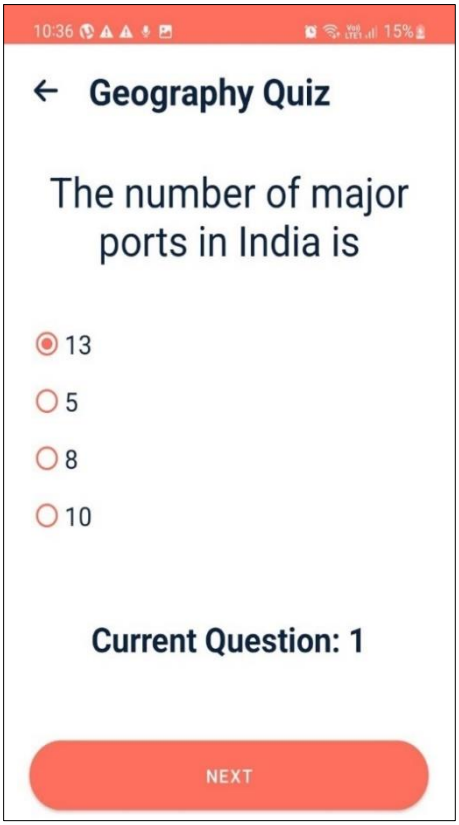

Figure 6.1.10: Start the geography quiz.

Literature Quiz : The snapshot shows the front page of the start screen of the literature quiz.

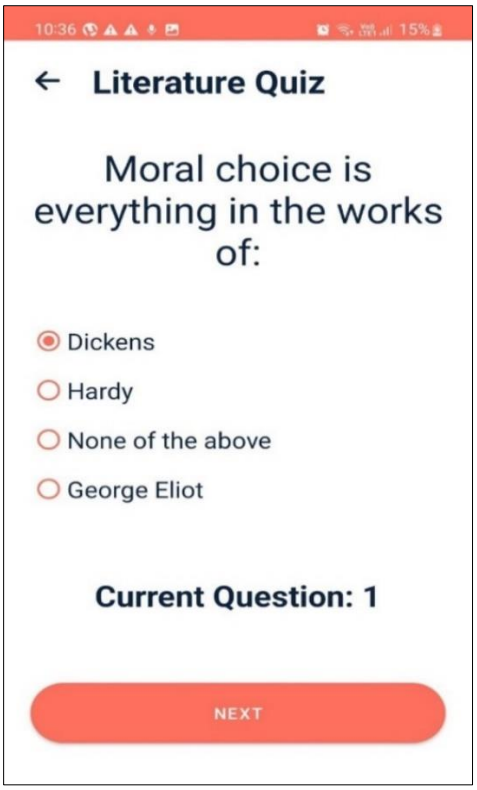

Figure 6.1.11: Start the literature quiz.

Finish the Quiz : This snapshot shows the last question of the quiz and then click "finish" button.

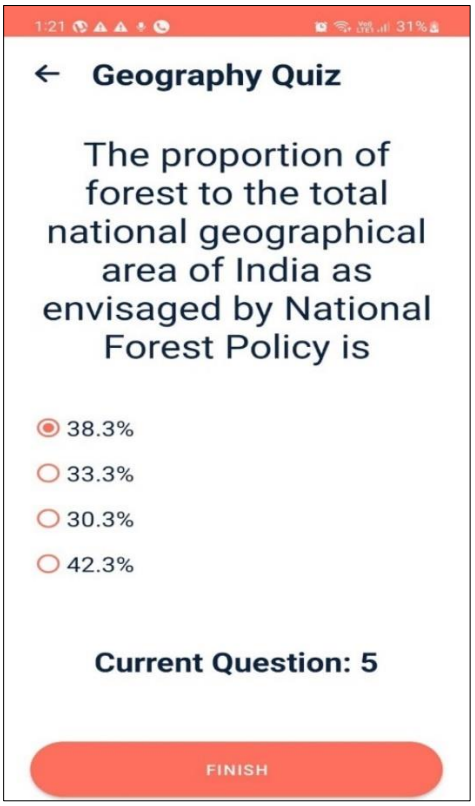

Figure 6.1.12:Finish the quiz.

Final Quiz Result: This page displays the result of the quiz taken after clicking the finish button.

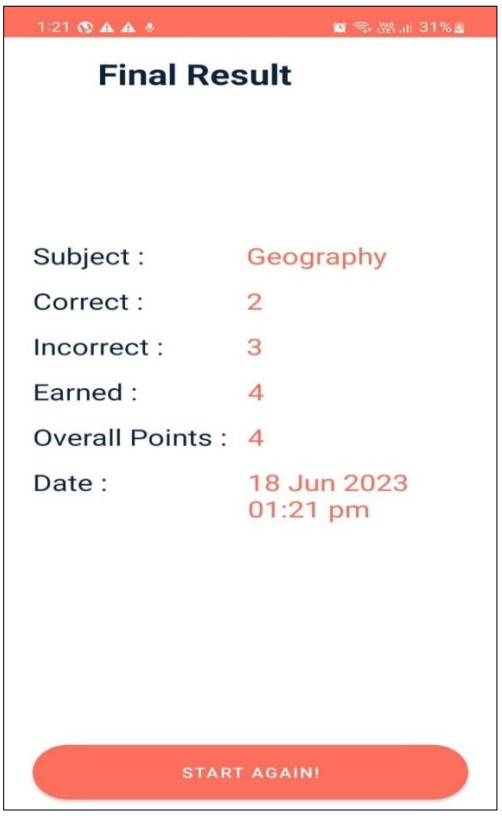

Figure 6.1.13: Final quiz result

History: This page displays the history of all the quizzes taken with the earned score and date.

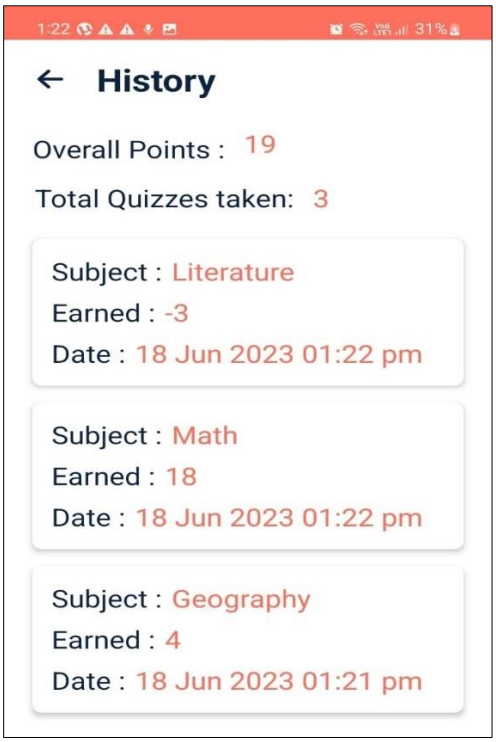

Figure 6.1.14: History

Change password: This page shows that the user after login can change the password again to make the app secure.

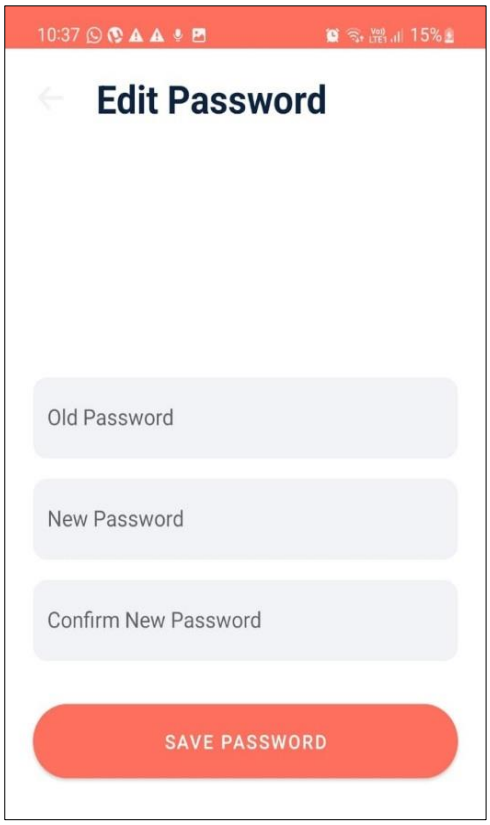

Figure 6.1.15: Change password

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 Conclusion

The ‘Smart Quiz Application’ project is developed to overcome the time-consuming problem of manual system. Apart from that in the current system, checking the answer sheets after taking test wastes the examiners time, so this application will check the correct answer, save the examiner time, and carry the examination in an effective manner. The users who use this system do not need to have high computing knowledge and the system will help them validate their knowledge in the respectivefield. This work deals with the development of android-based multiple-choice questions with various categories namely Math, Geography, Literature Knowledge. This application is developed for preparing for exams, allowing the users to prepare multiple-choice questions for distinct categories. The main goal of the application isto enable users to practice for subjective tests conducted in Schools and College withfocus on Skills and Knowledge.

The idea behind this project is to create an application that would train students for Competitive Exams like Olympiad and Other Exams. In each category there are 5 questions, and the score will be updated dynamically and will be displayed at the end of the particular category. It’s a user-friendly application which helps the user to analyses their weakness and work on those particular topics so that they can work their Knowledge and Skills .We have even considered exit and restart functionality to go back to the starting question of the particular category.

7.2 Future Enhancement

We will improve more features in this game to make it more interesting and fun loving like

- We can add many more Subjects and increase the number of questions into the app for the students to prepare and evaluate themselves for the exams.
- Leaderboards can be implemented to display the top scores of the users for each category.
- Time-based quizzes can be introduced where the users will have a limited amount of time to answer each question.
- Social media sharing can also be enabled where the users can share their quiz scores and achievements on platforms like Instagram or Facebook to promote user engagement.

REFERENCES

1) Google Developer Training, "Android Developer FundamentalsCourse– Concept Reference", Google Developer Training Team, 2017.
2) Dawn Griffiths and David Griffiths, "Head first Android Development", 1st Edition, O'Reilly SPD Publishers, 2015.
3) https://www.javatpoint.com/android-versions
4) https://www.tutorialspoint.com/android/android_architecture.html
5) https://www.geeksforgeeks.org/quiz-application/