**CHAPTER – 1**

**INTRODUCTION**

* 1. **Mobile Application**

As we know Android is a cell working system, designed commonly for touchscreen cell gadgets inclusive of smartphones and pills and Android Studio is the authentic included improvement surroundings for Google’s Android working system, constructed on JetBrains’ IntelliJ IDEA software program and designed in particular for Android improvement. Development of android-based Quiz application is especially needed by students and learners to prepare themselves for various examinations directly through good phones and tablets in hands. one among the main goal of our project is to facilitate students in learning, gaining up their data skills. we have a tendency to design the appliance to facilitate the users to be ready to take short quizzes mistreatment transportable devices reminiscent of good phones and tablets. Here in our project we have used this application to build the knowledge about the questions asked it’s a simple android application developed with the use of object oriented Programming language called Java. Designed through XML (eXtensible Markup Language)

Android Quiz Application is a user friendly app we can start the quiz just by clicking on start bar then we will be taken to the quiz questions page were we find some questions with the radio Button after checking the option the user has to click on the submit button then the right answer will be shown. A timer is set using Countdown function. After the number questions is over the credit score will be displayed in the screen later to quit a finish button is been added which will regenerate to the restart page.

* 1. **Aims and Objectives**

Development of Android-based Quiz application is mainly required by students and learners to prepare themselves for different examinations directly through Smart-Phones and tablets in hands. The main aim of this project is to facilitate students in learning, gaining and improving their knowledge skills. The main objective to develop Android project on MCQ Quiz Application is to provide android app on MCQ Quiz Application to customer, from where user can use it from his mobile device. Android project on MCQ Quiz Application is compatible with all android mobiles.

* 1. **About Android Studio**

Android Studio is the official Integrated Development Environment (IDE) for Google’s Android Operating system, built on JetBrains ‘IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

Android Studio supports all the same programming languages of [IntelliJ](https://en.wikipedia.org/wiki/IntelliJ) (and [CLion](https://en.wikipedia.org/wiki/CLion)) e.g. [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B), and more with extensions.  Android Studio 3.0 or later supports [Kotlin](https://en.wikipedia.org/wiki/Kotlin_(programming_language)) and all Java 7 language features and a subset of Java 8 language features that vary by platform version. It also supports all the new versions of Java.

* 1. **About XML**

Extensible Markup Language (XML) is a [markup language](https://en.wikipedia.org/wiki/Markup_language) and [file format](https://en.wikipedia.org/wiki/File_format) for storing, transmitting, and reconstructing arbitrary data. The main purpose of XML is [serialization](https://en.wikipedia.org/wiki/Serialization), i.e. storing, transmitting, and reconstructing arbitrary data. For two disparate systems to exchange information, they need to agree upon a file format. As a [markup language](https://en.wikipedia.org/wiki/Markup_language), XML labels, categorizes, and structurally organizes information. XML tags represent the data structure and contain metadata. What's within the tags is data, encoded in the way the XML standard specifies.

XML has come into common use for the interchange of data over the Internet. Hundreds of document formats using XML syntax have been developed,[[9]](https://en.wikipedia.org/wiki/XML#cite_note-Cover_pages_list-9) including [RSS](https://en.wikipedia.org/wiki/RSS), [Atom](https://en.wikipedia.org/wiki/Atom_(standard)), [Office Open XML](https://en.wikipedia.org/wiki/Office_Open_XML), [OpenDocument](https://en.wikipedia.org/wiki/OpenDocument), [SVG](https://en.wikipedia.org/wiki/Scalable_Vector_Graphics), and [XHTML](https://en.wikipedia.org/wiki/XHTML). XML also provides the base language for [communication protocols](https://en.wikipedia.org/wiki/Communication_protocol) such as [SOAP](https://en.wikipedia.org/wiki/SOAP) and [XMPP](https://en.wikipedia.org/wiki/Extensible_Messaging_and_Presence_Protocol). It is the message exchange format for the [Asynchronous JavaScript and XML (AJAX)](https://en.wikipedia.org/wiki/Ajax_(programming)) programming technique.

* + 1. **Different XML Files used in Android**
* **Layout XML Files:**Layout xml files are used to define the actual UI(User interface) of our application. It holds all the elements(views) or the tools that we want to use in our application. Like the Text View’s, [Button](https://abhiandroid.com/ui/button/)’s and other UI elements.
* **Manifest xml File(Mainfest.xml):**This xml is used to define all the components of our application. It includes the names of our application packages, our Activities, receivers, services  and the permissions that our application needs. For Example – Suppose we need to use internet in our app then we need to define Internet permission in this file.
* **Strings xml File(strings.xml):**This xml file is used to replace the Hard-coded strings with a single string. We define all the strings in this xml file and then access them in our app(Activity or in  Layout XML files) from this file. This file enhance the reusability of the code.
* **Styles xml File(styles.xml):**This xml is used to define different styles and looks for the UI(User Interface) of application. We define our custom themes and styles in this file.
* **Drawable xml Files:**These are those xml files that are used to provide various graphics to the elements or views of application. When we need to create a custom UI we use drawable xml files. Suppose if we need to define a gradient color in the background of [Button](https://abhiandroid.com/ui/button/) or any custom shape for a view then we create a Drawable xml file and set it in the background of View.
* **Color xml File (colors.xml):**This file is used to define the color codes that we used in our app. We simply define the color’s in this file and used them in our app from this file.
* **Dimension xml File(dimens.xml):**This xml file is used to define the dimensions of the View’s. Suppose we need a Button with 50dp(density pixel) height then we define the value 50dp in dimens.xml file and then use it in our app from this file.
  1. **About Java**

Java is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [programming language](https://en.wikipedia.org/wiki/Programming_language) that is designed to have as few implementation [dependencies](https://en.wikipedia.org/wiki/Dependency_(computer_science)) as possible. It is a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_language) programming language intended to let [programmers](https://en.wikipedia.org/wiki/Programmer) write once, run anywhere ([WORA](https://en.wikipedia.org/wiki/Write_once,_run_anywhere)), meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need to recompile.

Java applications are typically compiled to [bytecode](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of the underlying [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture). The [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) of Java is similar to [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B), but has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them. The Java runtime provides dynamic capabilities (such as [reflection](https://en.wikipedia.org/wiki/Reflective_programming) and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most [popular programming languages in use](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity) according to [GitHub](https://en.wikipedia.org/wiki/GitHub), particularly for [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server_model) [web applications](https://en.wikipedia.org/wiki/Web_application), with a reported 9 million developers. Java was originally developed by [James Gosling](https://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) and released in May 1995 as a core component of Sun Microsystems' [Java platform](https://en.wikipedia.org/wiki/Java_(software_platform)). The original and [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) Java [compilers](https://en.wikipedia.org/wiki/Compiler), virtual machines, and [class libraries](https://en.wikipedia.org/wiki/Library_(computing)) were originally released by Sun under [proprietary licenses](https://en.wikipedia.org/wiki/Proprietary_license). As of May 2007, in compliance with the specifications of the [Java Community Process](https://en.wikipedia.org/wiki/Java_Community_Process), Sun had [relicensed](https://en.wikipedia.org/wiki/Software_relicensing) most of its Java technologies under the [GPL-2.0-only](https://en.wikipedia.org/wiki/GNU_General_Public_License) license. [Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation) offers its own [Hotspot](https://en.wikipedia.org/wiki/HotSpot_(virtual_machine)) Java Virtual Machine, however the official [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) is the [OpenJDK](https://en.wikipedia.org/wiki/OpenJDK) JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

**CHAPTER 2**

**SYSTEM REQUIREMENTS**

**2.1 Hardware Requirements**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important. The Web Based Manufacturing System can be efficiently run on Pentium system with at least 128 MB RAM and Hard disk drive having 20 GB. Floppy disk drive of 1.44 MB and 14 inch Samsung color monitor suits the information system operation.(A Printer is required for hard copy output).

Processor : Intel-Core i3, i5, i7

Processor Speed : 2.0. GHz

RAM : 4GB or more

Hard Disk : 40GB to 80GB

**2.2 Software Requirements**

The Android Quiz Application is designed in such a way that the user can easily interact with the screen.

Frontend : XML

Backend : Java

IDE : Android Studio

Emulator : Pixel 2 API Tiramisu

Compile SDK : 32

**CHAPTER 3**

**SYSTEM DESIGN**

The main objective of developing this project is to give friendly experience to the users. The application will greatly simplify and develop the knowledge of technology and would manage to attend the quiz number of times and would also get the right answers to be known instantly. It is a simple quiz application designed for educational purpose. Here in the project the user could start the application and attend the quiz questions that will be displayed on the emulator timer of about 20 seconds is given to the improvement of the project.

**3.1 Data Flow Diagram**

Restart Quiz

**ifs**

**User event**

**yess**

**no**

Finish

**Question=10**

**yes**

**no**

Turn Red

Turn Green

**If Correct**

**Else**

Next question

**On click**

Pop warning

Activate Submit

**Option clicked**

Start Timer

Wait for User Action

**Event Received**

**Fig 3.1 Data Flow Diagram**

**CHAPTER 4**

**IMPLEMENTAION**

**4.1 Study of the System**

We developed an Android project on Quiz Application which is a user friendly app we can start the quiz just by clicking on start button on the front screen then we will be moved to the quiz questions page were we could attend multiple questions and get to know the answers instantly. For these we have used XML and Java as our main languages for designing the app. An Emulator called Tiramisu is used to display the app.

**4.2 Source Code**

**4.2.1 MainActivity.java**

package com.madhan.aqa;  
  
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
  
public class MainActivity extends AppCompatActivity {  
  
 Button btnStart;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 btnStart =findViewById(R.id.*btnStart*);  
  
 btnStart.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent intent = new Intent(MainActivity.this,QuizActivity.class);  
 startActivity(intent);

}  
 });  
 }  
}

**4.2.2 QuestionModel.java**

package com.madhan.aqa;  
  
public class QuestionModel {  
 private String question, option1,option2,option3;  
 private int correctAnsNo;  
  
 public QuestionModel(String question, String option1, String option2, String option3, int correctAnsNo) {  
 this.question = question;  
 this.option1 = option1;  
 this.option2 = option2;  
 this.option3 = option3;  
 this.correctAnsNo = correctAnsNo;  
 }

public String getQuestion() {  
 return question;  
 }  
 public void setQuestion(String question) {  
 this.question = question;  
 }  
 public String getOption1() {  
 return option1;  
 }  
 public void setOption1(String option1) {  
 this.option1 = option1;  
 }  
 public String getOption2() {  
 return option2;  
 }  
 public void setOption2(String option2) {  
 this.option2 = option2;  
 }  
 public String getOption3() {  
 return option3;  
 }  
 public void setOption3(String option3) {  
 this.option3 = option3;  
 }  
 public int getCorrectAnsNo() {  
 return correctAnsNo;  
 }  
 public void setCorrectAnsNo(int correctAnsNo) {  
 this.correctAnsNo = correctAnsNo;  
 }  
}

**4.2.3 QuizActivity.java**

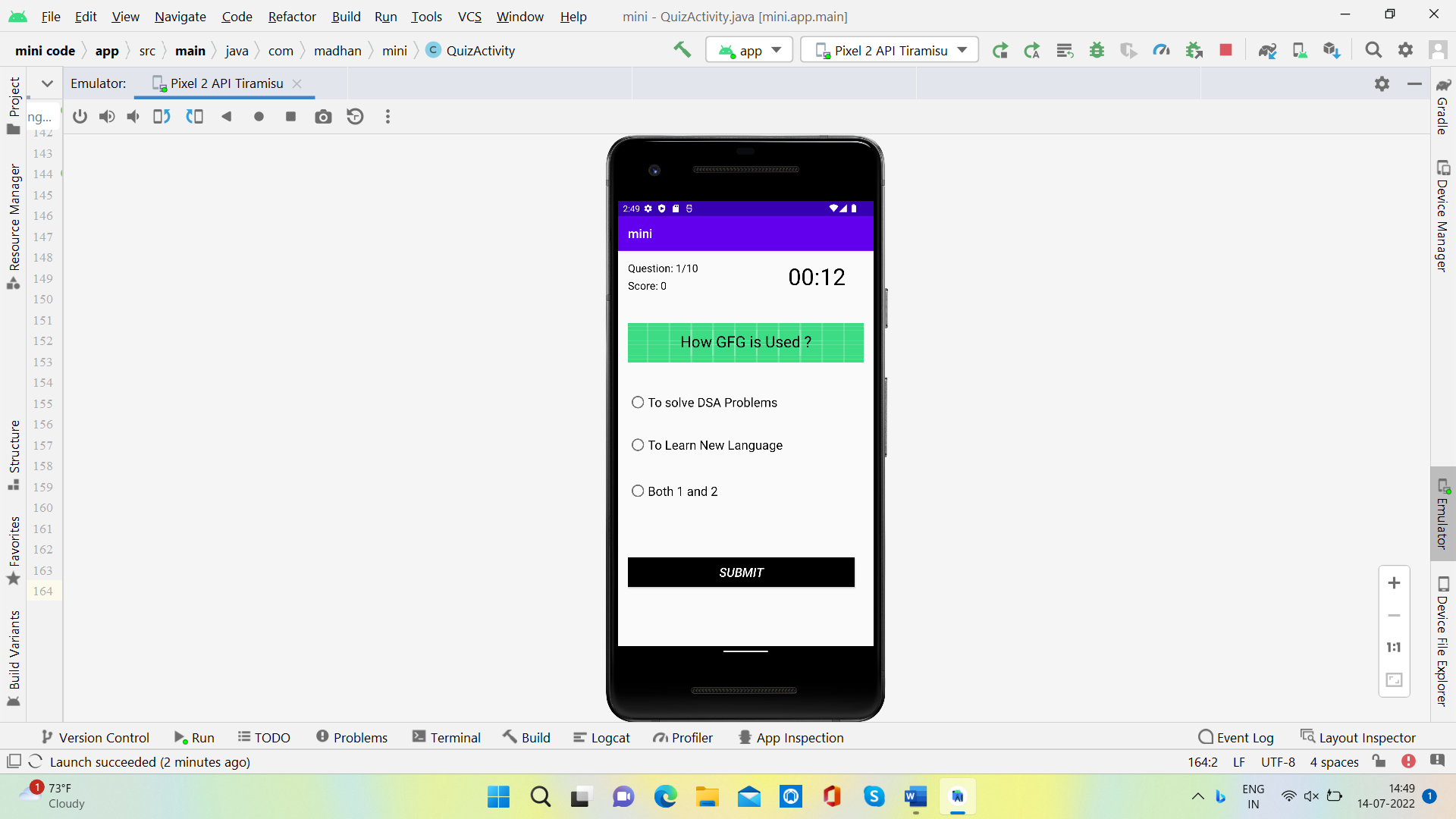
package com.madhan.aqa;  
  
import android.content.res.ColorStateList;  
import android.graphics.Color;  
import android.os.CountDownTimer;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.security.PrivateKey;  
import java.util.ArrayList;  
import java.util.List;  
  
public class QuizActivity extends AppCompatActivity {  
  
 private TextView tvQuestion, tvScore, tvQuestionNo, tvTimer;  
 private RadioGroup radioGroup;  
 private RadioButton rb1, rb2, rb3;  
 private Button btnNext;  
 int totalQuestions;  
 int qCounter = 0;  
 int score;  
  
 ColorStateList dfRbColor;  
 boolean answered;  
 CountDownTimer countDownTimer;  
 private QuestionModel currentQuestion;  
 private List<QuestionModel>questionsList;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_quiz*);  
  
 questionsList =new ArrayList<>();  
 tvQuestion =findViewById(R.id.*textQuestion*);  
 tvScore =findViewById(R.id.*textScore*);  
 tvQuestionNo =findViewById(R.id.*textQuestionNo*);  
 tvTimer =findViewById(R.id.*textTimer*);  
  
 radioGroup =findViewById(R.id.*radioGroup*);  
 rb1 =findViewById(R.id.*rb1*);  
 rb2 =findViewById(R.id.*rb2*);  
 rb3 =findViewById(R.id.*rb3*);  
 btnNext =findViewById(R.id.*btnNext*);  
  
 dfRbColor = rb1.getTextColors();  
  
 addQuestions();  
 totalQuestions=questionsList.size();  
 showNextQuestion();  
  
 btnNext.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 if(answered == false){  
 if(rb1.isChecked() || rb2.isChecked() || rb3.isChecked()){  
 checkAnswer();  
 countDownTimer.cancel();  
 }else{  
 Toast.*makeText*(QuizActivity.this, "Please Select an Option",Toast.*LENGTH\_SHORT*).show();  
 }  
 }else{  
 showNextQuestion();  
 }  
 }  
 });  
 }  
  
 private void checkAnswer() {  
 answered = true;  
 RadioButton rbSelected = findViewById(radioGroup.getCheckedRadioButtonId());  
 int answerNo = radioGroup.indexOfChild(rbSelected) +1;  
 if (answerNo == currentQuestion.getCorrectAnsNo()){  
 score++;  
 tvScore.setText(("Score: "+score));  
 }  
 rb1.setTextColor(Color.*RED*);  
 rb2.setTextColor(Color.*RED*);  
 rb3.setTextColor(Color.*RED*);  
 switch(currentQuestion.getCorrectAnsNo()){  
 case 1:  
 rb1.setTextColor(Color.*GREEN*);  
 break;  
 case 2:  
 rb2.setTextColor(Color.*GREEN*);  
 break;  
 case 3:  
 rb3.setTextColor(Color.*GREEN*);  
 break;  
 }  
 if(qCounter<totalQuestions){  
 btnNext.setText("Next");  
 }else{  
 btnNext.setText("Finish");  
 }  
 }  
  
 private void showNextQuestion() {  
  
 radioGroup.clearCheck();  
 rb1.setTextColor(dfRbColor);  
 rb2.setTextColor(dfRbColor);  
 rb3.setTextColor(dfRbColor);  
  
 if(qCounter<totalQuestions){  
 timer();  
 currentQuestion =questionsList.get(qCounter);  
 tvQuestion.setText(currentQuestion.getQuestion());  
 rb1.setText(currentQuestion.getOption1());  
 rb2.setText(currentQuestion.getOption2());  
 rb3.setText(currentQuestion.getOption3());  
  
 qCounter++;  
 btnNext.setText("Submit");  
 tvQuestionNo.setText("Question: "+qCounter+"/"+totalQuestions);  
 answered = false;  
  
 }else{  
 finish();  
 }  
 }  
  
 private void timer() {  
 countDownTimer = new CountDownTimer(20000,1000) {  
 @Override  
 public void onTick(long l) {  
 tvTimer.setText("00:" + l/1000);  
 }  
  
 @Override  
 public void onFinish() {  
 showNextQuestion();  
 }  
 }.start( );  
 }  
 private void addQuestions(){  
 questionsList.add(new QuestionModel("How GFG is Used ?","To solve DSA Problems","To Learn New Language","Both 1 and 2",3));  
 questionsList.add(new QuestionModel("What is GCM in Android ?","Google Check Message ","Google Cloud Manager","Google Cloud Messaging",3));  
 questionsList.add(new QuestionModel("Abbreviation of API ?","Application Program Interfear","Application Programming Interface","Android Programming Interface",2));  
 questionsList.add(new QuestionModel("Who is the CEO of GOOGLE ?","Sundar Pichai","Satya Nadela","Jeff Bezoz",1));  
 questionsList.add(new QuestionModel("Who is the CEO of Amazon ?","Sundar Pichai","Satya Nadela","Jeff Bezoz",3));  
 questionsList.add(new QuestionModel("Who is the Founder of Apple ?","Tim Cook","Satya Nadela","Jeff Bezoz",1));  
 questionsList.add(new QuestionModel("Who is the CEO of Microsoft ?","Sundar Pichai","Satya Nadela","Jeff Bezoz",2));  
 questionsList.add(new QuestionModel("Who is the Asian no 1 Millanior ?","Suresh Ambani","Mukesh Ambani","Anil Ambani",2));  
 questionsList.add(new QuestionModel("What is the full form of GFG ?","Green For Geeks","Geeks For Geeks","Geeks For Greens",2));  
 questionsList.add(new QuestionModel("Who is the CEO of Alphabets ?","Sundar Pichai","Satya Nadela","Jeff Bezoz",1));  
 }  
}

**CHAPTER 5**

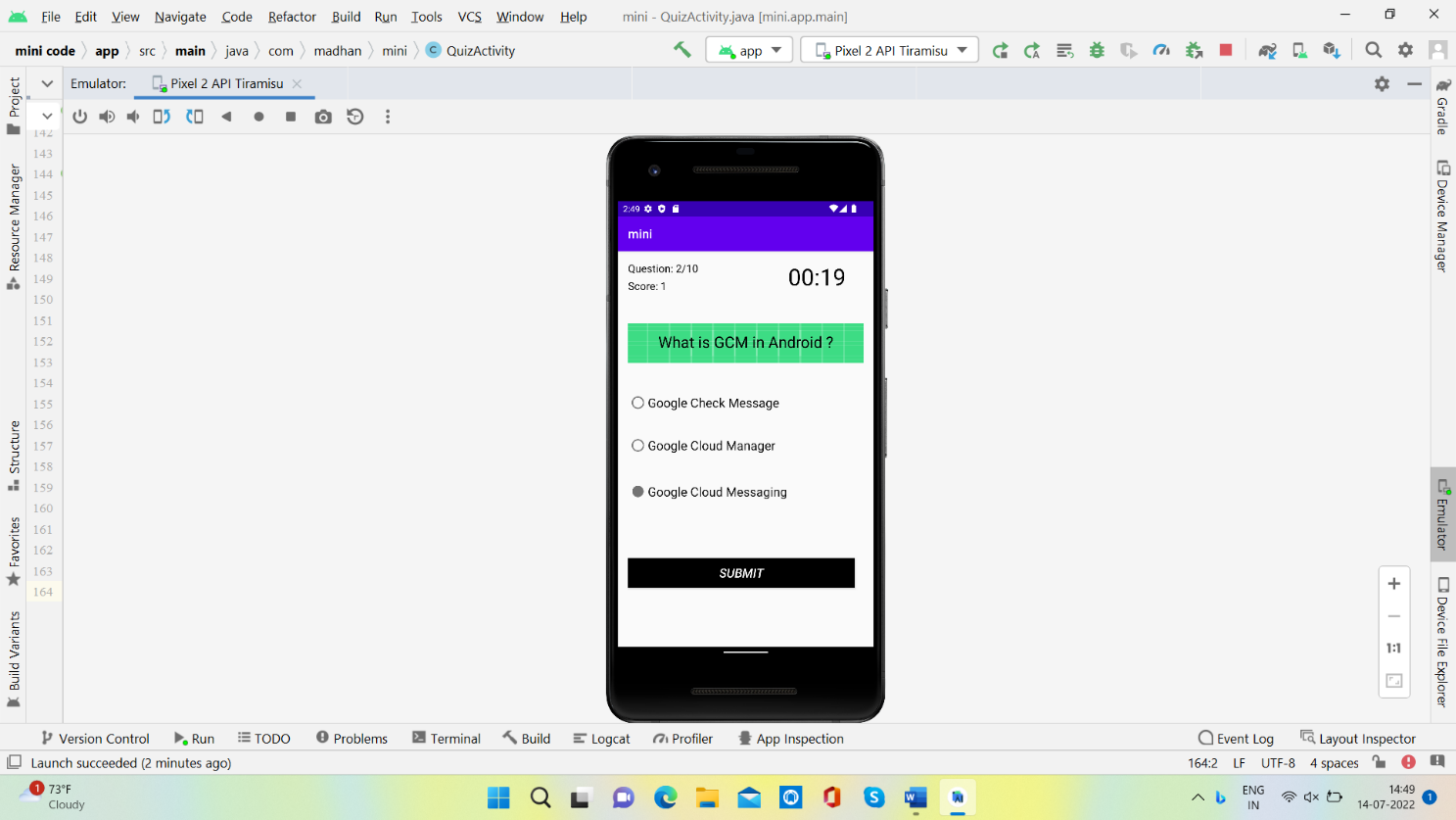
**RESULT**

**5.1 Start Screen of Emulator**

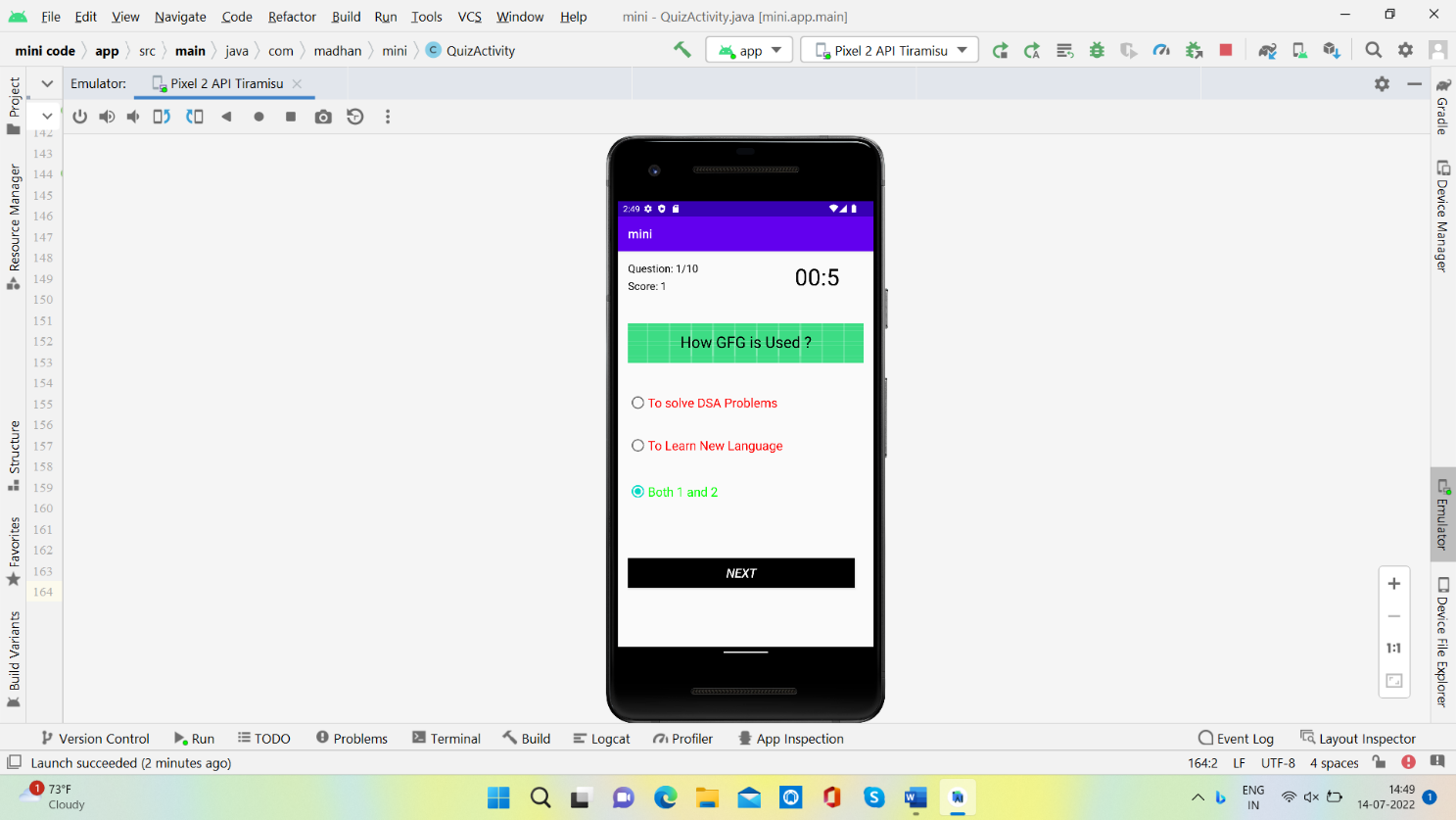
**Fig 5.1 Start Screen of Emulator**

**5.2 Questions Screen**

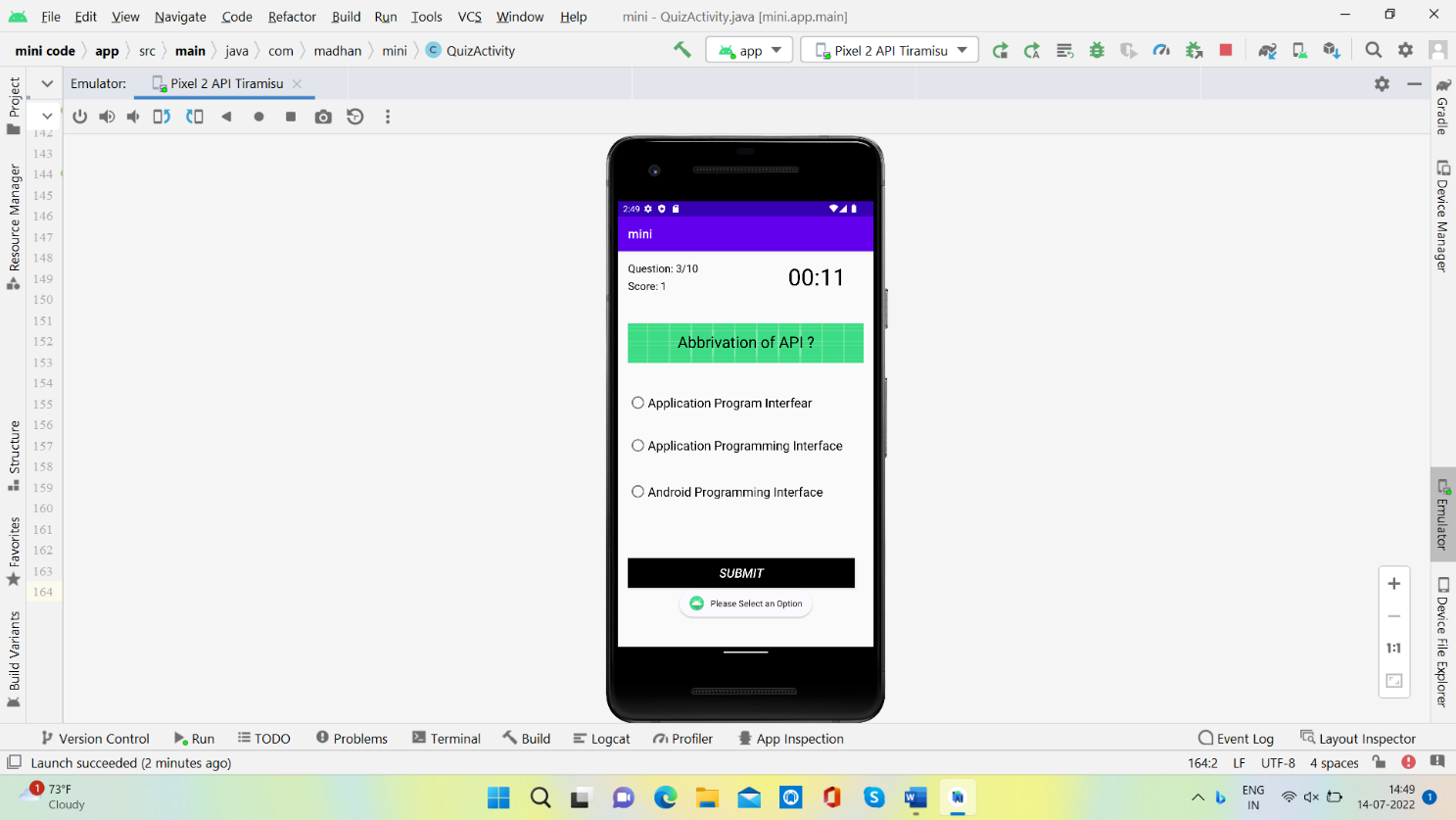
**Fig 5.2 Question Screen**

**5.3 Submit Screen**

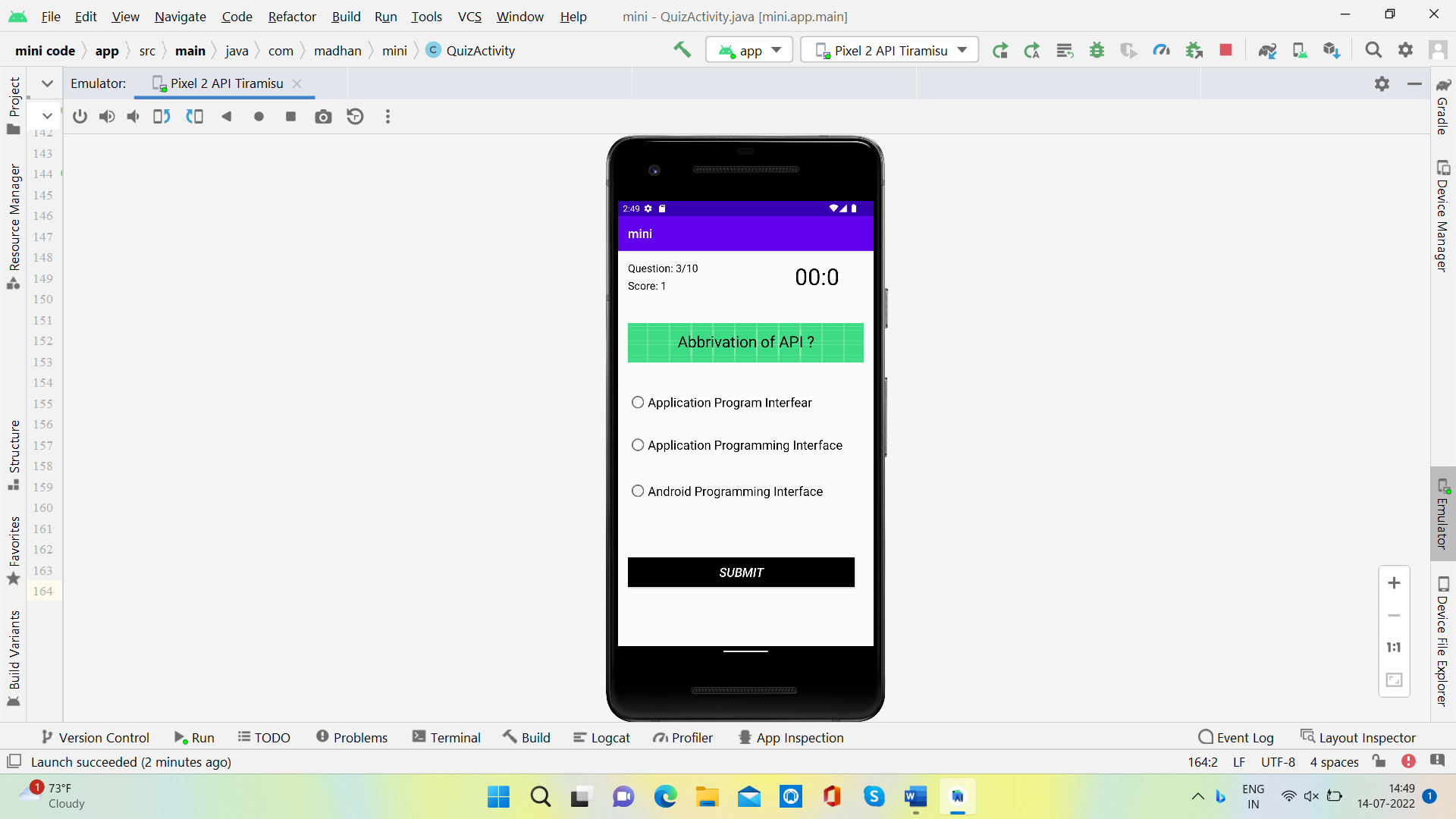
**Fig 5.3 Submit Screen**

**5.4 Answers Display Screen**

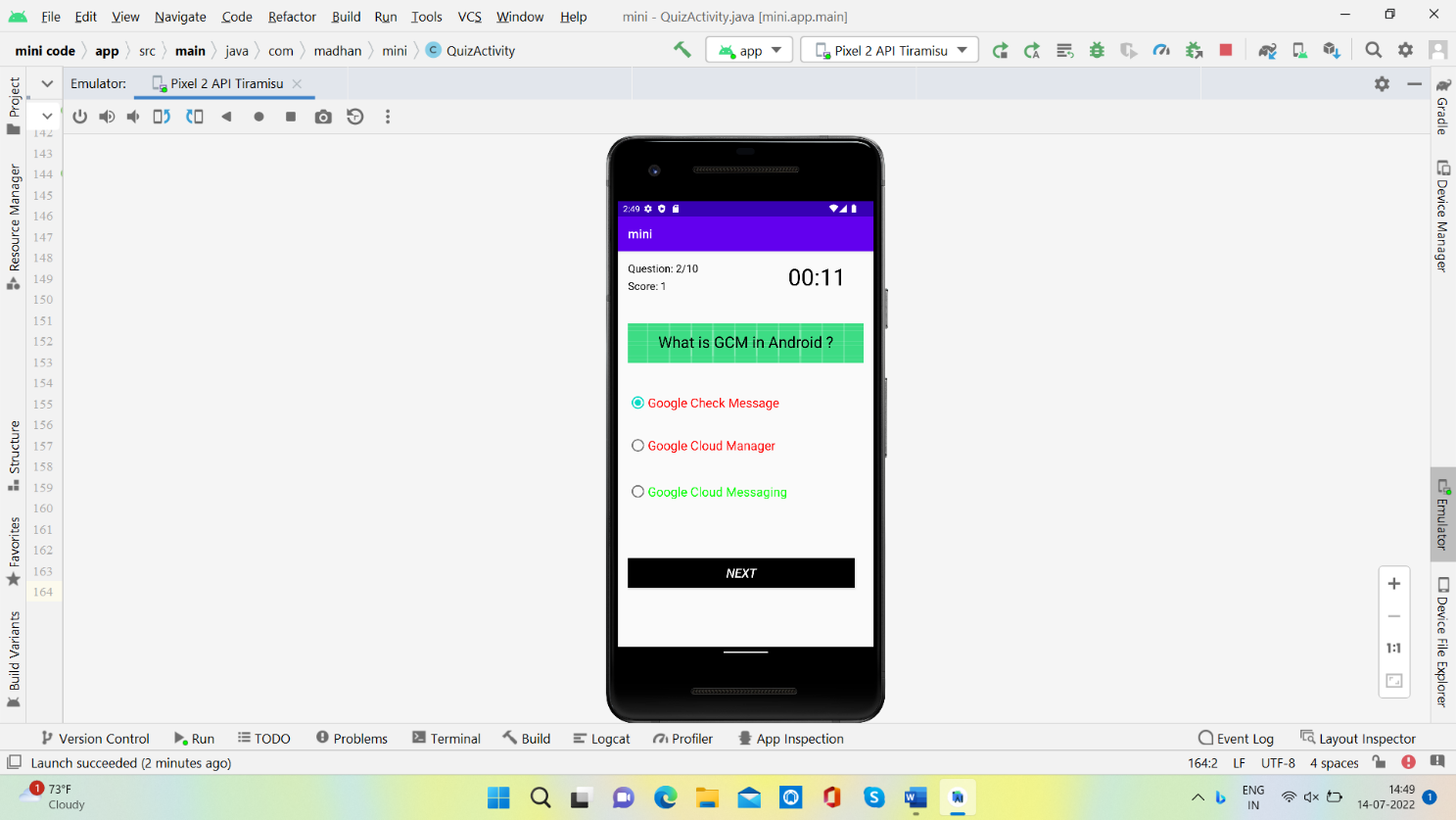
**Fig 5.3 Answers Display Screen**

**5.5 Warning Pop-up Screen**

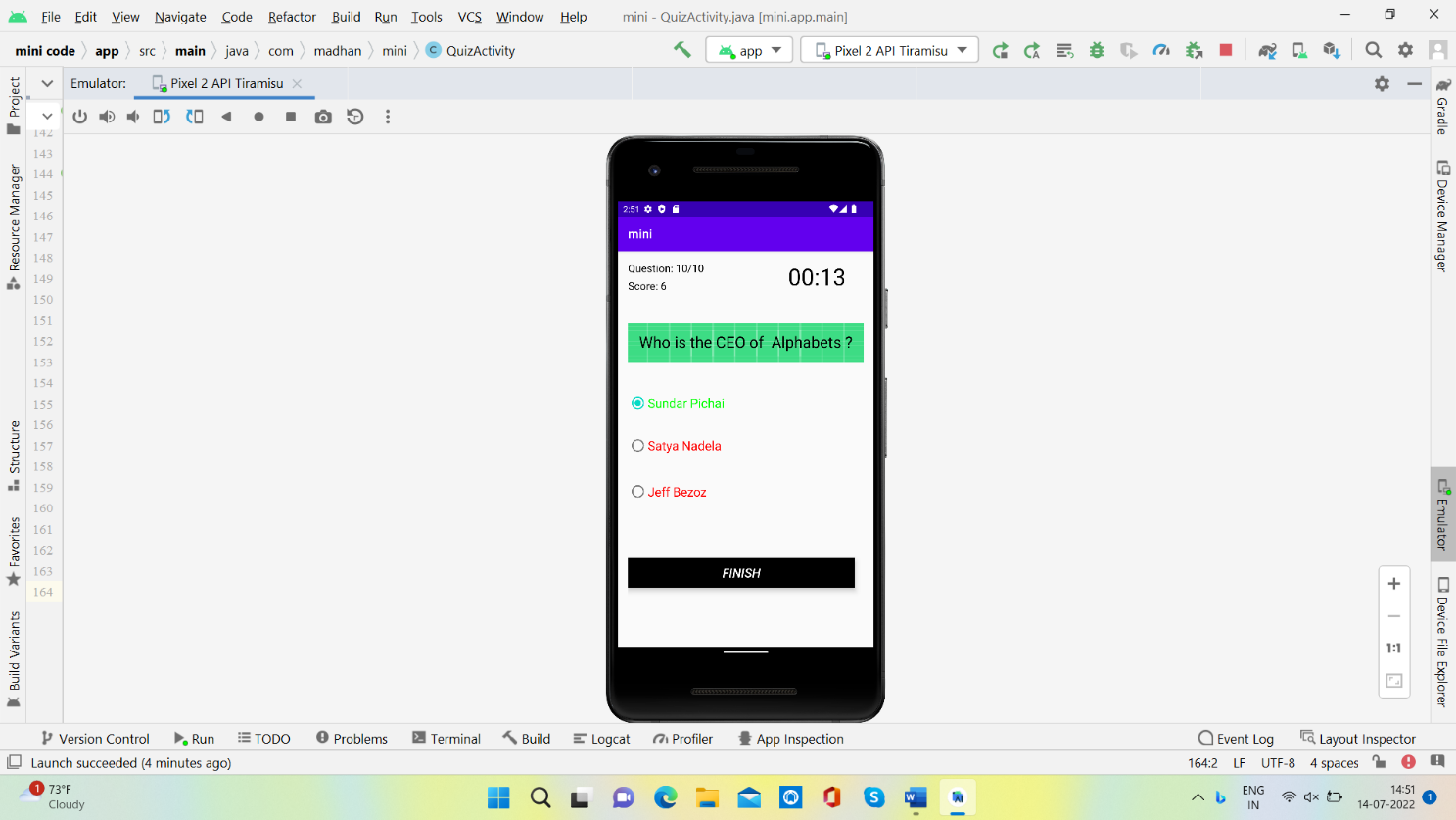
**Fig 5.5 Warning Pop-up Screen**

**5.6 Time Up Screen**

**Fig 5.6 Time Up Screen**

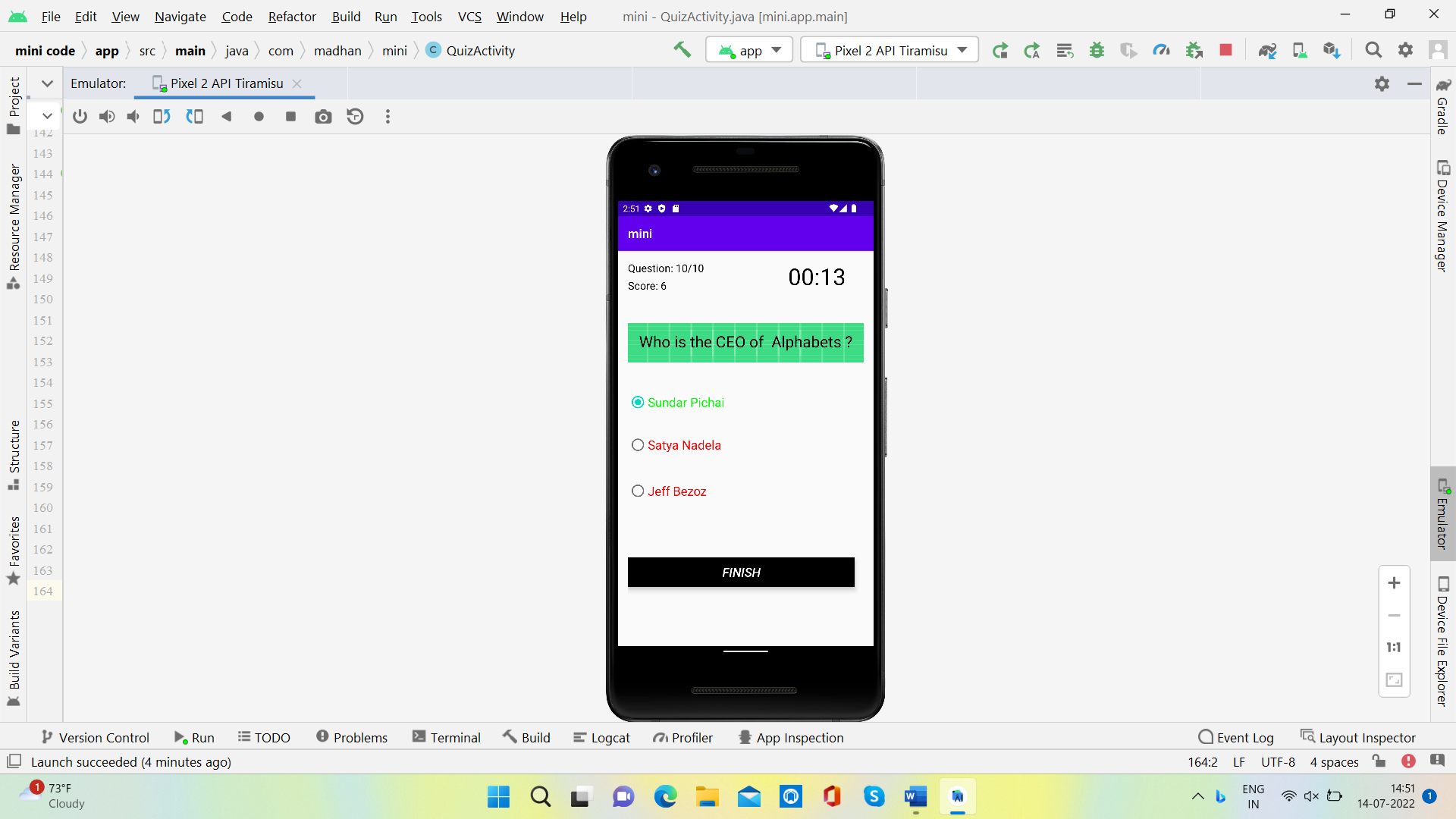
**5.7 Wrong Answer Check Screen**

**Fig 5.7 Wrong Answer Check Screen**

**5.8 Finish Button Display Screen**

**Fig 5.8 Finish Button Display Screen**

**5.9 Final Score Display**

****

**Fig 5.9 Final Score Display**

**CONCLUSION**

An attempt is made to develop an Android Application with the means of satisfactory requirements of the user successfully. Since it is user friendly, it enables the user to interact efficiently and easily. The development of the mini project which has been built by using java as a source, has given a good exposure towards Application through which some of the techniques which help in development of designing, coding was well understood. It helps to understand basics of java and XML. In this project quiz questions are implemented via java code itself which is easily understandable by the very beginner also. This project is both informative and entertaining

**REFERENCES**

1. <https://www.w3schools.com>
2. <https://gradle.org/>
3. <http://google.com>
4. <https://developer.android.com>