

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi – 590 018



Submitted in partial fulfillment of requirements for the
MOBILE APPLICATION DEVELOPMENT (18CSMP68)
Mini Project Report
On

“KIDS LEARNING APP”

Submitted by

SANIL M M (1GD20CS038)

VARUN REDDY B (1GD20CS052)

VIGNESH R (1GD20CS055)

Under the Guidance of

Mr. GIRISH M

Assistant Professor,

Dept. of Computer Science & Engineering

Bangalore-560048



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GOPALAN COLLEGE OF ENGINEERING AND
MANAGEMENT
BANGALORE-560048

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

This is to certify that the mini project work entitled “**Kids Learning App**” is a bonafied work carried out by by **SANIL M M(1GD20CS038)** ,**VARUN REDDY B (1GD20CS052)**, **VIGNESH R (1GD20CS055)** in partial fulfillment for the Mobile Application Development Laboratory prescribed by the Visvesvaraya Technological University, Belgavi during the year 2022-2023 for the sixth semester B.E. Computer Science and Engineering. The Mini project report has been approved as it satisfies the academic requirements with respect to the mini project work prescribed for the sixth semester **Mobile Application Development LAB**.

Signature of Guide

Mr. GIRISH M

Assistant Professor, CSE

Signature of HOD

Dr. Shailesh Kumar

Professor & Head, CS&E

Name of the Examiners

1. _____

2. _____

Signature with date

DECLARATION

We **SANIL M M (1GD20CS038)**, **VARUN REDDY B(1GD20CS052)**, **VIGNESH R (1GD20CS055)** students of VI semester B.E in Computer Science and Engineering, Gopalan College of Engineering and Management, Bangalore - 560 048 hereby declare that the mini-project entitled “ **KIDS LEARNING APP** “ submitted to the Visvesvaraya Technological University during the academic year 2022-2023, is a record of a work done by us under the guidance of **Mr. Girish M, Assistant Professor**, Department of Computer Science and Engineering, Gopalan College of Engineering and Management, Bangalore. This work is submitted towards the academic requirement of Mobile Application Development (18CSMP68).

Date :

SANIL M M (1GD20CS038)

Place :

VARUN REDDY B(1GD20CS052)

VIGNESH R(1GD20CS055)

ACKNOWLEDGEMENT

We sincerely owe our gratitude to all the persons who helped and guided us in completing this mini project work.

We are thankful to **Dr. Basavaraju C, Principal, Gopalan College of Engineering and Management, Bangalore** , for having supported us in our academic endeavors.

We are extremely thankful to **Dr. Shailesh Kumar , Professor & Head, Department of Computer Science and Engineering**, for his valuable support and his timely inquiries into the progress of the work.

We are greatly indebted to our guide **Mr.Girish M , Asst. Professor, Department of Computer Science and Engineering**, for his consistent co-operation and support.

We are obliged to all **teaching and non-teaching staff members of Department of Computer Science and Engineering, Bangalore** for the valuable support provided by them in their respective fields we are grateful for their co-operation during the period of our mini project.

Lastly we thank almighty, our parents and friends for their constant encouragement without which this mini project would not be possible.

SANIL MM (1GD20CS038)

VARUN REDDY B (1GD20CS052)

VIGNESH R (1GD20CS055)

ABSTRACT

The Kids Learning App is an android based application using Mobile Application based on earliest exposure of education system which will be alphabets, numbers, colours, week days and shapes . Instead of hard copy materials and now computer as the learning and teaching medium, parents and teachers can use this app to teach their kids and students wherever and whenever they want. The primary used for this system is pre-schooler and the secondary users are the elder people such as parents, teachers, and siblings to guide and teach the primary user. Teacher, parents or guidance can apply this application into their mobile and teach the kids from time to time wherever and whenever.

In addition, this application may help the kids to learn interactively with fun and safe as well. They can also learn at home and not only in school or kindergarten in which through parent's mobile phone with parent's guide. This system is focusing on the importance of getting the kid's attention towards studying by implementing sound from various source, attractive pictures and colours

Table of contents

Chapter 1.	Introduction	1
1.1	Project Summary	1
1.2	Project Purpose.....	1
1.3	Project Scope	1
1.4	Problem Statement.....	2
1.5	Objective Of the Project	2
1.6	Organisation Of the Project.....	3
Chapter 2.	System Requirements,	4
2.1	User Characteristics	4
2.2	Hardware And Software Requirements	4
Chapter 3.	Design.....	5
3.1	System Architecture Design	5
3.2	Data flow Graph	6
Chapter 4.	Implementation	7
4.1	Built In Functions.....	7
4.2	Android Manifest	7
4.3	Source code.....	9
Chapter 5.	Snapshots	23
Chapter 6.	Testing	26
Chapter 7.	Conclusion.....	28
7.1	Future Enhancement.....	28
7.2	Bibliography.....	28

CHAPTER 1

INTRODUCTION

1.1 Project Summary

The project is a single web-page containing different list items and a text area to add custom tasks. The “Add” button lets the user add an object to the list. We have the facility to check the items from the list or remove them if we want. If we wish, we can clear all the list items with a single button. It is a very user-friendly and user-interactive project that makes the work of a user a lot easier. To-Do List project is an application specially built to keep track of errands or tasks that need to be done. This application will be like a task keeper where the user would be able to enter the tasks that they need to do. Once they are done with their tasks, they can also remove them from the list.

1.2 Project Purpose

The project has been implemented in modules so as to make it user friendly i.e., if a code viewer is finding trouble with a particular function it can go to a particular module and make corresponding changes in it instead of searching in the program. Also, it is more advantageous for future modifications as when modifications are to be made instead of searching the whole program module search can be useful as it decreases the time consumption.

1.3 Project Scope

- Now since we are amidst a pandemic, this job has gotten more difficult and since the learning process has switched over to online mode there is very little both parties can do.
- We need a more fun and interactive way to teach children, using current world technology that makes enjoyable for children and easier for teachers.

- It's often hard to grasp the attention of younger children which makes it difficult for pre-primary teachers to carry out their duties of imparting the basic and fundamental knowledge which acts as the foundation for their learning process.

1.4 Problem Statement

“To implement To-Do List project application specifically to build and keep track of errands or tasks that need to be done.”

1.5 Objective of the Project

This application will be like a task keeper where the user would be able to enter the tasks that they need to do. Once they are done with their tasks, they can also remove them from the list.

Let us see what is there in the application:

- You can add the tasks that are to be done in a descriptive way.
- You will be able to add as many tasks as you have.
- Once the task is completed, you will be able to remove it by clicking on “Task Completed” button. Following are the user interface components that will be shown in our application
- There will be an add task button on the right top of the application
- There will be a button “Task Completed” for each task added, so you will be able to delete it.
- The tasks that will be added by the users are stored in the list view.

We have developed a mobile application that addresses the above issue. We introduce you to 'Kiplay' kids learning app which acts as an educational friend to all children. It provides a fun and interactive way to learn various topics.

This app aims at incorporating the orthodox teaching with modern teaching and with the help of current technology, this is brought to every household to help all children grow their foundation as strong as possible.

The app has the following features :

- Fun way to learn the English Alphabet and Days of the week via androids text-to-speech module
- Activities that helps the child to learn animals and fruits names in a very interesting way.
- The child can also listen to the poems through the app.
- An activity to visually and phonically learn colors and their spellings
- The Child can shape his imagination and creativity through the paint tools provided within the app.

1.6 Organisation of the Project

The project was organised in a systematic way. First, we analysed what are the basic features to be included in the project to make it acceptable. As it is a Android- oriented project, we made the sketches prior, so as to have an idea like how our output must look like. After all these, the source code was formulated as a paper work. All the required software were downloaded. Finally, the successful implementation of the project.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 User Characteristics

Dependability: The dependability of a computer system is a property of the system that equates to its trustworthiness.

Availability: The ability of the system to deliver services when requested. There is no error in the program while executing the program.

Reliability: The ability of the system to deliver services as specified. The program is compatible with all types of operating system without any failure.

Safety: The ability of the system to operate without catastrophic failure. This program is user friendly and it will never affect the system.

Security: The ability of the system to protect itself against accidental or deliberate intrusion.

The various methods used in this project are as follows: -

- Emulator To perform and display the functionality of the project.
- Android studio to create, design, test, debug and run the android project.
- Mouse To navigate through the emulator.
- Keyboard To give inputs to the project.

2.2 Hardware And Software Requirements

Basic system requirements for Android Studio			
	Microsoft Windows	Mac	Linux
Operating System Version	Microsoft Windows 7/8/10 (32- or 64-bit) The Android Emulator only supports 64-bit Windows.	Mac OS X 10.10 (Yosemite) or higher, up to 10.14 (macOS Mojave)	GNOME or KDE desktop Tested on gLinux based on Debian (4.19.67-2rodete2).
Random Access Memory (RAM)	4 GB RAM minimum; 8 GB RAM recommended.		
Free digital storage	2 GB of available digital storage minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image).		
Minimum required JDK version	Java Development Kit 8		
Minimum screen resolution	1280 x 800		

CHAPTER 3

DESIGN

3.1 System Architecture Design

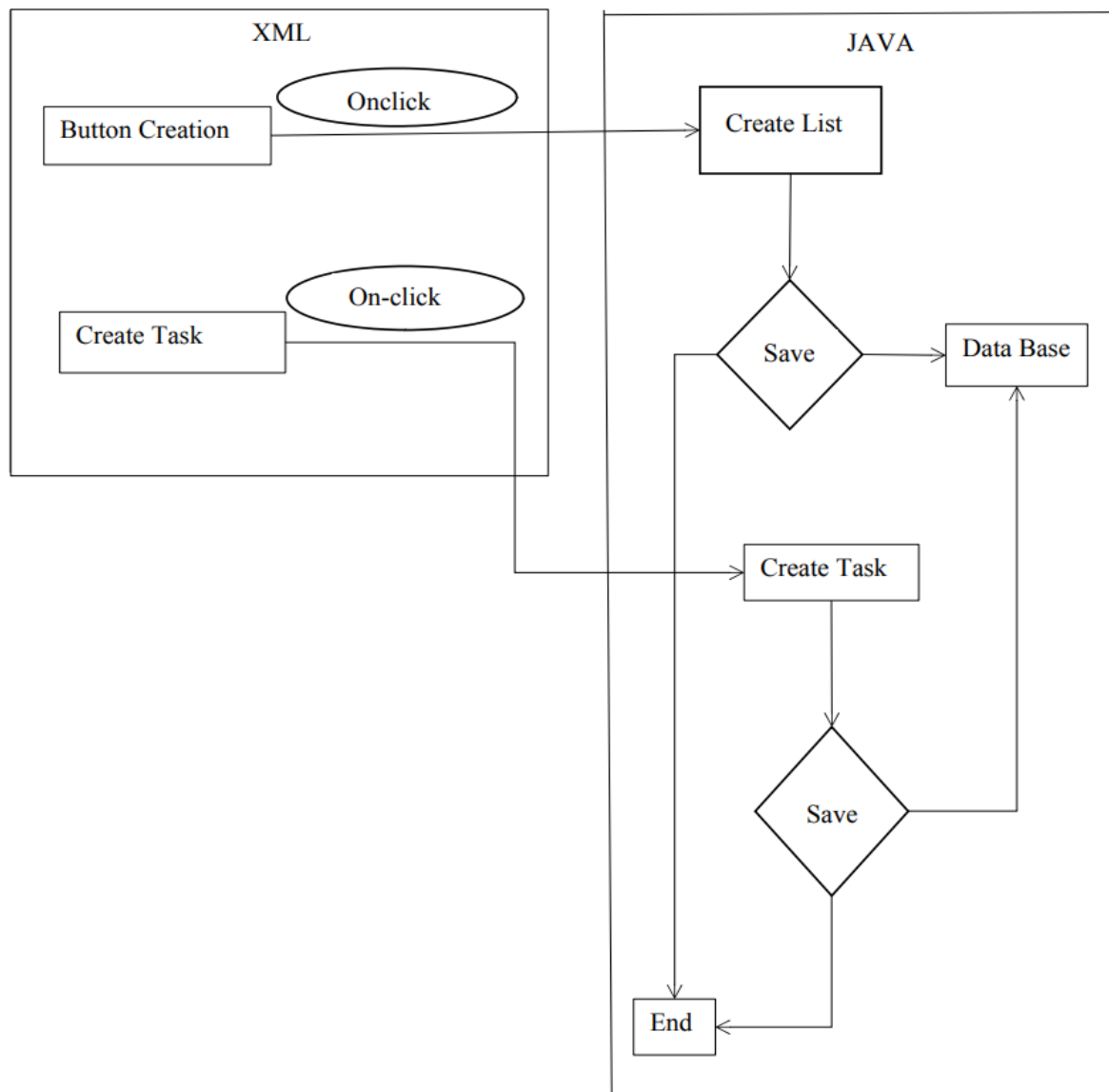


Fig. 3.1: System Architecture Design

3.2 Data Flow Graph

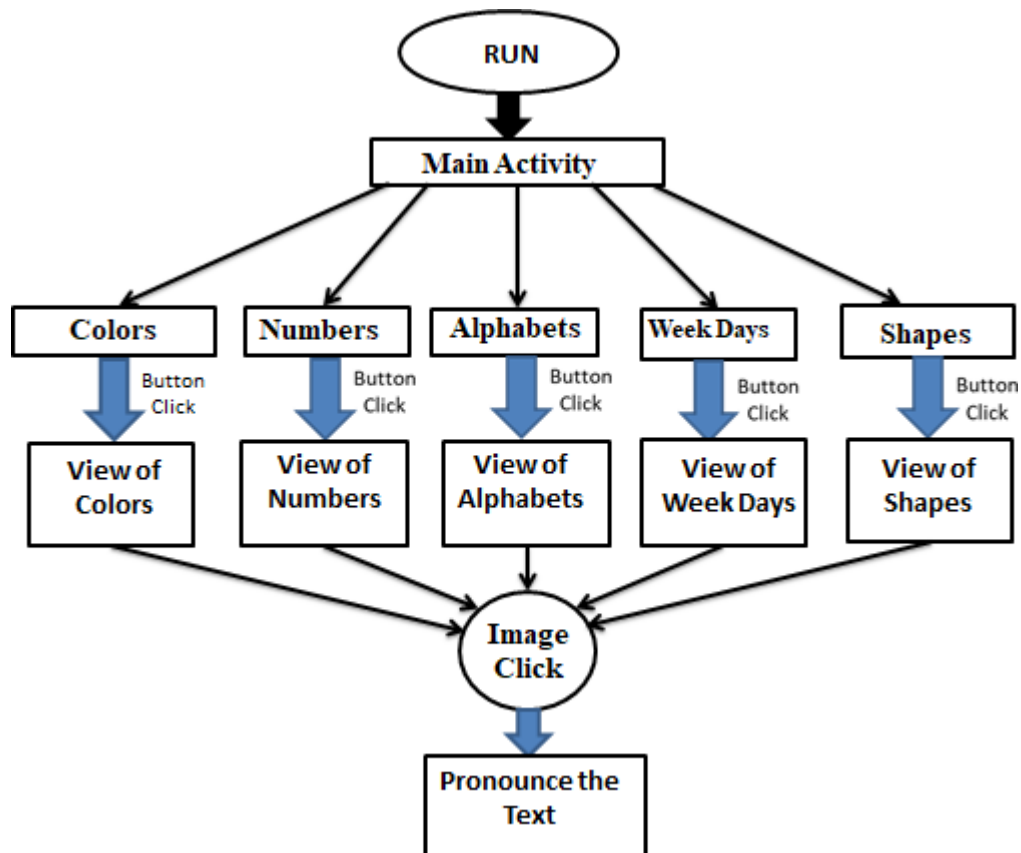


Fig. 3.2: Data Flow Graph

CHAPTER 4

IMPLEMENTATION

4.1 Built in Functions

sqrt() – Used to calculate the square root of a number.

print() – Used to print a message to standard output.

rem() – To find the remainder of one number when divided by another.

toInt() - To convert a number to integer value.

readline() – Used for standard input.

compareTo() – To compare two numbers and return Boolean value.

println() – Outputs data on the terminal.

Random.nextInt() – returns a random value from the

specified range. **toCharArray()** – returns all chars from the string in an array. **forEach()** – iterates over the values of the array.

4.2 AndroidManifest.xml

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

<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="ideanity.oceans.kidslearning">
<application android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/AppTheme">
<activity android:name=".ShapesActivity"></activity>
```

```
<activity android:name=".WeekActivity" />

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

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

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

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

<activity android:name=".MainActivity">

<intent-filter>

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

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

</intent-filter>

</activity>

<meta-data
android:name="preloaded_fonts"
android:resource="@array/preloaded_fonts" />

</application>

</manifest>
```

4.3 SOURCE CODE**4.3.1 : Alphabet Activity**

```
package ideanity.oceans.kidslearning;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.media.MediaPlayer;import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;

import ideanity.oceans.kidslearning.adapter.AlphabetAdapter;
import ideanity.oceans.kidslearning.adapter.NumberAdapter;
import ideanity.oceans.kidslearning.helpers.AlphabetHelper;
import ideanity.oceans.kidslearning.helpers.AlphabetHelper;

public class AlphabetActivity extends AppCompatActivity implements RecyclerViewAction{

    RecyclerView recyclerViewAlphabet;

    RecyclerView.Adapter adapter;

    static MediaPlayer mpone;

    static MediaPlayer mptwo;
```

```
static MediaPlayer mpthree;
static MediaPlayer mpfour;
static MediaPlayer mpfive;
ImageView backMenu;
TextView colorName;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);
    setContentView(R.layout.activity_alphabet);

    recyclerViewAlphabet = findViewById(R.id.recycler_alphabet);
    backMenu = findViewById(R.id.menu_nav);
    colorName = findViewById(R.id.alphabet_number);

    mppone = MediaPlayer.create(AlphabetActivity.this, R.raw.a);
    mptwo= MediaPlayer.create(AlphabetActivity.this, R.raw.b);
    mpthree= MediaPlayer.create(AlphabetActivity.this, R.raw.c);
    mpfour= MediaPlayer.create(AlphabetActivity.this, R.raw.d);
    mpfive= MediaPlayer.create(AlphabetActivity.this, R.raw.e);
    backMenu.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            AlphabetActivity.super.onBackPressed();
        }
    });
    featuredAlphabet();
}
private void featuredAlphabet()
{
```



```
ArrayList<AlphabetHelper> questionLocations = new ArrayList<>();
questionLocations.add(new AlphabetHelper("A"));
questionLocations.add(new AlphabetHelper("B"));
questionLocations.add(new AlphabetHelper("C"));
questionLocations.add(new AlphabetHelper("D"));
questionLocations.add(new AlphabetHelper("E"));
adapter = new AlphabetAdapter(questionLocations, this, this);
GridLayoutManager gridLayoutManager = new GridLayoutManager(this, 2,
GridLayoutManager.VERTICAL, false);
recyclerViewAlphabet.setLayoutManager(gridLayoutManager);
recyclerViewAlphabet.setAdapter(adapter);
}

@Override
public void onClicked(int clickedViewId, int clickedItemPosition) {
try {
switch (clickedItemPosition) {
case 0: mpone.start();
        colorName.setText("A");
        break;

case 1: mptwo.start();
        colorName.setText("B");
        break;

case 2: mpthree.start();
        colorName.setText("C");
        break;

case 3: mpfour.start();
        colorName.setText("D");
        break;
```

```
case 4: mpfive.start();
        colorName.setText("E");
        break;
default: Toast.makeText(this, "Wrong index", Toast.LENGTH_SHORT).show();
        colorName.setText("1 2 3");
    }
}
catch (IllegalStateException e) {
    e.printStackTrace();
}
}
@Override
public void onViewLongClicked(int clickedViewId, int clickedItemPosition) {

}
@Override
public void onDestroy() {
    super.onDestroy();
    mpone.release();
    mptwo.release();
    mpthree.release();
    mpfour.release();
    mpfive.release();
}
}
```

4.3.2 : Colors Activity

```
package ideanity.oceans.kidslearning;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.graphics.drawable.GradientDrawable;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;
import android.widget.GridLayout;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;
import ideanity.oceans.kidslearning.adapter.ColorAdapter;
import ideanity.oceans.kidslearning.helpers.ColorHelper;
public class ColorsActivity extends AppCompatActivity implements RecyclerViewAction{
    RecyclerView recyclerViewColor;
    RecyclerView.Adapter adapter;
    private GradientDrawable red, green, purple, blue, pink, yellow, orange, brown, black, white;
    //Media Creation
    static MediaPlayer mppurple;
    static MediaPlayer mpwhite;
    static MediaPlayer mpgreen;
    static MediaPlayer mppink;
    static MediaPlayer mpred;
    ImageView backMenu;
    TextView colorName;

    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,WindowManager.LayoutParams.FLAG_FULLSCREEN); setContentView(R.layout.activity_colors);
recyclerViewColor = findViewById(R.id.recycler_color);
backMenu = findViewById(R.id.menu_nav);
colorName = findViewById(R.id.color_name);
mpred = MediaPlayer.create(ColorsActivity.this, R.raw.red);
mpgreen= MediaPlayer.create(ColorsActivity.this, R.raw.green);
mppink= MediaPlayer.create(ColorsActivity.this, R.raw.pink);
mppurple= MediaPlayer.create(ColorsActivity.this, R.raw.purple);
mpwhite= MediaPlayer.create(ColorsActivity.this, R.raw.white);
featuredColors();
```

```
backMenu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ColorsActivity.super.onBackPressed();
    }
});
private void featuredColors() {
    //All Gradients
    red = new GradientDrawable(GradientDrawable.Orientation.LEFT_RIGHT, new
    int[]{0xccf44336, 0xccf44336});
    green = new GradientDrawable(GradientDrawable.Orientation.LEFT_RIGHT, new
    int[]{0xcc4caf50, 0xcc4caf50});
    purple = new GradientDrawable(GradientDrawable.Orientation.LEFT_RIGHT, new
    int[]{0xcc9c27b0, 0xcc9c27b0});
    blue = new GradientDrawable(GradientDrawable.Orientation.LEFT_RIGHT, new
    int[]{0xcc2196f3, 0xcc2196f3});
    pink = new GradientDrawable(GradientDrawable.Orientation.LEFT_RIGHT, new
```

```
int[] {0xccec407a, 0xccec407a});
ArrayList<ColorHelper> questionLocations = new ArrayList<>();
questionLocations.add(new ColorHelper(red, "Red"));
questionLocations.add(new ColorHelper(green, "Green"));
questionLocations.add(new ColorHelper(purple, "Purple"));
questionLocations.add(new ColorHelper(blue, "Blue"));
questionLocations.add(new ColorHelper(pink, "Pink"));
adapter = new ColorAdapter(questionLocations, this, this);

GridLayoutManager gridLayoutManager = new GridLayoutManager(this, 2,
GridLayoutManager.VERTICAL, false);
recyclerViewColor.setLayoutManager(gridLayoutManager);
recyclerViewColor.setAdapter(adapter);
}
@Override
public void onViewClicked(int clickedViewId, int clickedItemPosition) {

try {
switch (clickedItemPosition) {
case 0: mpred.start();
        colorName.setText("Red");
        break;
case 1: mpgreen.start();
        colorName.setText("Green");
        break;
case 2: mppurple.start();
        colorName.setText("Purple");
        break;
case 3: mpblue.start();
        colorName.setText("Blue");
        break;
```

```
case 4: mppink.start();
        colorName.setText("Pink");
        break;
case 5: mpyellow.start();
        colorName.setText("Yellow");
        break;
default: Toast.makeText(this, "Wrong index", Toast.LENGTH_SHORT).show();
colorName.setText("Color");
}
}
catch (IllegalStateException e) {
e.printStackTrace();
}
}
@Override
public void onViewLongClicked(int clickedViewId, int clickedItemPosition) {

}

@Override
public void onDestroy() {
super.onDestroy();
mpblue.release();
mpgreen.release();
mppink.release();
mppurple.release();
mpwhite.release();
}
}
```

4.3.3 : Numbers Activity

```
package ideanity.oceans.kidslearning;
```

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;
import ideanity.oceans.kidslearning.adapter.ColorAdapter;
import ideanity.oceans.kidslearning.adapter.NumberAdapter;
import ideanity.oceans.kidslearning.helpers.ColorHelper;
import ideanity.oceans.kidslearning.helpers.NumberHelper;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);
    setContentView(R.layout.activity_numbers);
    recyclerViewNumber = findViewById(R.id.recycler_numbers);
    backMenu = findViewById(R.id.menu_nav);
    colorName = findViewById(R.id.number_name);

    mpone = MediaPlayer.create(NumbersActivity.this, R.raw.kid_1);
    mptwo= MediaPlayer.create(NumbersActivity.this, R.raw.kid_2);
    mpthree= MediaPlayer.create(NumbersActivity.this, R.raw.kid_3);
```

```
mpfour= MediaPlayer.create(NumbersActivity.this, R.raw.kid_4);
mpfive= MediaPlayer.create(NumbersActivity.this, R.raw.kid_5);
backMenu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        NumbersActivity.super.onBackPressed();
    }
});
featuredNumbers();
}
private void featuredNumbers() {
    ArrayList<NumberHelper> questionLocations = new ArrayList<>();
    questionLocations.add(new NumberHelper("1"));
    questionLocations.add(new NumberHelper("2"));
    questionLocations.add(new NumberHelper("3"));
    questionLocations.add(new NumberHelper("4"));
    questionLocations.add(new NumberHelper("5"));
}
@Override
public void onViewLongClicked(int clickedViewId, int clickedItemPosition) {
}
@Override
public void onDestroy() {
    super.onDestroy();
    mpone.release();
    mptwo.release();
    mpthree.release();
    mpfour.release();
    mpfive.release();
}
}
```


4.3.4 : Shapes Activity

```
package ideanity.oceans.kidslearning;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle; import android.view.View;
import android.view.WindowManager;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;
import ideanity.oceans.kidslearning.adapter.NumberAdapter;
import ideanity.oceans.kidslearning.adapter.ShapeAdapter;
import ideanity.oceans.kidslearning.helpers.ShapeHelper;
import ideanity.oceans.kidslearning.helpers.ShapeHelper;

public class ShapesActivity extends AppCompatActivity implements RecyclerViewAction{
    RecyclerView recyclerViewShape;
    RecyclerView.Adapter adapter;
    static MediaPlayer mpone;
    static MediaPlayer mptwo;
    static MediaPlayer mpthree;
    static MediaPlayer mpfour;
    static MediaPlayer mpfive;
    ImageView backMenu;
    TextView colorName;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_shapes);
        recyclerViewShape = findViewById(R.id.recycler_shape);
```

```
backMenu = findViewById(R.id.menu_nav);
colorName = findViewById(R.id.shape_number);

mpone = MediaPlayer.create(ShapesActivity.this, R.raw.circle);
mptwo= MediaPlayer.create(ShapesActivity.this, R.raw.square);
mpthree= MediaPlayer.create(ShapesActivity.this, R.raw.triangle);
mpfour= MediaPlayer.create(ShapesActivity.this, R.raw.star);
mpfive= MediaPlayer.create(ShapesActivity.this, R.raw.rectangle);
backMenu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ShapesActivity.super.onBackPressed();
    }
});
featuredShapes();
}

private void featuredShapes() {
    ArrayList<ShapeHelper> questionLocations = new ArrayList<>();
    questionLocations.add(new ShapeHelper(R.drawable.circle));
    questionLocations.add(new ShapeHelper(R.drawable.square));
    questionLocations.add(new ShapeHelper(R.drawable.triangle));
    questionLocations.add(new ShapeHelper(R.drawable.star));
    questionLocations.add(new ShapeHelper(R.drawable.rectangle));
    adapter = new ShapeAdapter(questionLocations, this, this);

    GridLayoutManager gridLayoutManager = new GridLayoutManager(this, 2,
    GridLayoutManager.VERTICAL, false);
    recyclerViewShape.setLayoutManager(gridLayoutManager);
    recyclerViewShape.setAdapter(adapter);
}
}
```

4.3.5 : Weeks Activity

```
package ideanity.oceans.kidslearning;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;
import ideanity.oceans.kidslearning.adapter.AlphabetAdapter;
import ideanity.oceans.kidslearning.adapter.WeekAdapter;
import ideanity.oceans.kidslearning.helpers.WeekHelper;
import ideanity.oceans.kidslearning.helpers.WeekHelper;

public class WeekActivity extends AppCompatActivity implements RecyclerViewAction{
    RecyclerView recyclerViewWeek;
    RecyclerView.Adapter adapter;
    static MediaPlayer mpone;
    static MediaPlayer mptwo;
    static MediaPlayer mpthree;
    static MediaPlayer mpfour;
    static MediaPlayer mpfive;
    ImageView backMenu;
    TextView colorName;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_week);
        recyclerViewWeek = findViewById(R.id.recycler_week);
        backMenu = findViewById(R.id.menu_nav);
        colorName = findViewById(R.id.week_number);
    }
}
```

```
mpone = MediaPlayer.create(WeekActivity.this, R.raw.sunday);
mptwo= MediaPlayer.create(WeekActivity.this, R.raw.monday);
mpthree= MediaPlayer.create(WeekActivity.this, R.raw.tuesday);
mpfour= MediaPlayer.create(WeekActivity.this, R.raw.wednesday);
mpfive= MediaPlayer.create(WeekActivity.this, R.raw.thursday);
mpsix= MediaPlayer.create(WeekActivity.this, R.raw.friday);
mpseven= MediaPlayer.create(WeekActivity.this, R.raw.saturday);
backMenu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        WeekActivity.super.onBackPressed();
    }
});
featuredWeek();
}
@Override
public void onDestroy() {
    super.onDestroy();
    mppone.release();
    mptwo.release();
    mpthree.release();
    mpfour.release();
    mpfive.release();
    mpsix.release();
    mpseven.release();
}
}
```

CHAPTER 5

SNAPSHOTS



Figure 5.1 : MainActivity



Figure 5.2 : Alphabet Activity



Figure 5.3 : Numbers Activity

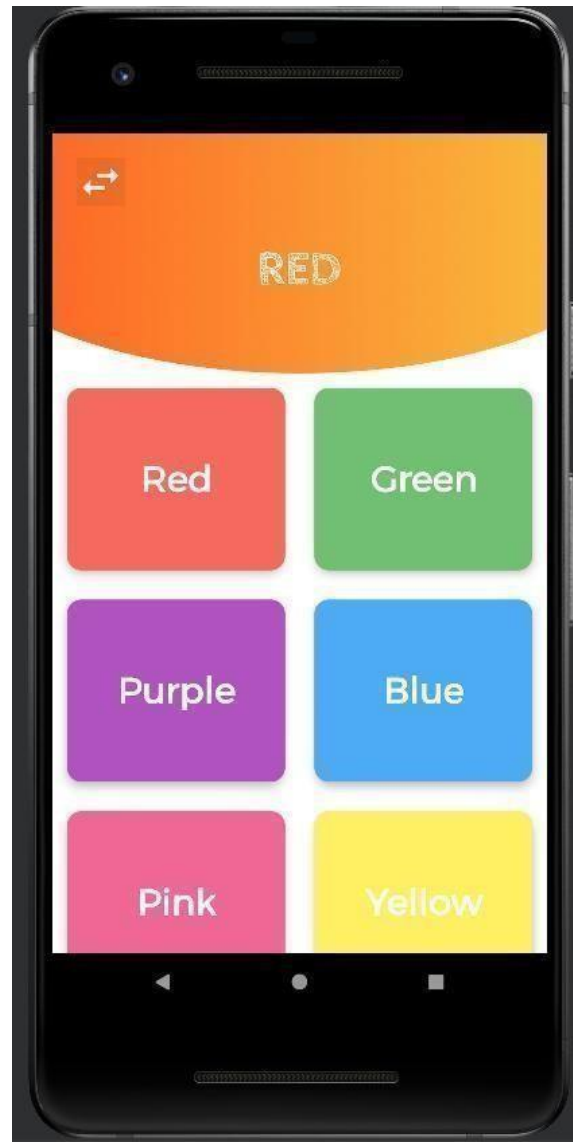


Figure 5.4 : Colors Activity



Figure 5.5 : Week Activity



Figure 5.6 : Shapes Activity

CHAPTER 6

TESTING

In software engineering, a test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement. A battery of test cases can be built to produce the desired coverage of the software being tested. Formally defined test cases allow the same tests to be run repeatedly against successive versions of the software, allowing for effective and consistent regression testing.

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. Testing is the process of executing a program with the explicit intention of finding errors that is making the program fail. The tester may analysts, programmer or a specialist trained for software testing, is actually trying to make the program fail. Analysts know that an effective testing program does not guarantee system reliability. Therefore, reliability must be designed into the system.

In order to fully test that all the requirements of an application are met, there must be at least two test cases for each requirement: one positive test and one negative test. If a requirement has sub-requirements, each sub-requirement must have at least two test cases. Keeping track of the link between the requirement and the test is frequently done using a traceability matrix. Written test cases should include a description of the functionality to be tested, and the preparation required to ensure that the test can be conducted. A formal written test case is characterized by a known input and by an expected output, which is worked out before the test is executed. The known input should test a precondition and the expected output should test a postcondition.

For applications or systems without formal requirements, test cases can be written based on the accepted normal operation of programs of a similar class. In some schools of testing, test

cases are not written at all but the activities and results are reported after the tests have been run.

In scenario testing, hypothetical stories are used to help the tester to think through a complex problem or system. These scenarios are usually not written down in any detail. They can be as simple as a diagram for a testing environment or they could be a description TIC-TAC-TOE APP Dept. of CSE 2020-21 Page 14 written in prose. The ideal scenario test is a story that is motivating, credible, complex, and easy to evaluate.

SL NO.	DESCRIPTION	EXPECTED OUTPUT	ACTUAL OUTPUT	REMARKS
1.	Pass the program	MainActivity page is displayed	MainActivity page is displayed	PASS
2.	When Colors button is clicked	View of different Colors	View of different Colors	PASS
3.	When Numbers button is clicked	View of Numbers	View of Numbers	PASS
4.	When Alphabets button is clicked	View of Alphabets	View of Alphabets	PASS
5.	When Week Days button is clicked	View of Week days	View of Week days	PASS
6.	When Shapes button is clicked	View of different Shapes	View of different Shapes	PASS

Table 6.1: Test Case

CHAPTER 7

CONCLUSION

Our application has provided many features that helps children learn with minimal work and maximum fun. It's children friendly design not only makes it attractive but also helps in holding the attention of these children. It helps children learn outside of their school syllabus and introduces them to learning data from all around the world. This application acts as your own personal tutor who's there for you at all times ready to impart knowledge as your convenience. We would further like to do more research to make our app more easier to learn from, include learning techniques that has been proven to help the children learn in the best way possible. We would also like to include information for each categories from different parts of the world so kids would be more exposed to these information from a earlier age.

7.1 Future Enhancement

The development of the project is not an easy process as it involves lot of challenges in different stages of software analysis, design, coding and testing. Having understood the requirements properly and implementing the solutions as per the expectation as brought to the closure of the project. We have tried our best to make this project very realistic, so that the user does not face any trouble when switching over from many real-life android project to this highly useful one.

7.2 Bibliography

Websites:

- www.Data-Flair.Training.com
- www.Github.com
- www.wikipedia.com
- www.Youtube.com